

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



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

PHÁT TRIỂN BOARD ĐIỀU KHIỂN VÀ
HỆ THỐNG HỖ TRỢ NÔNG NGHIỆP ĐÔ THỊ

HỘI ĐỒNG: Công nghệ phần mềm

GVHD: ThS. Nguyễn Cao Trí

GVPB: ThS. Lê Đình Thuận

SVTH 1: Huỳnh Bá Thạch (51303742)

SVTH 2: Tạ Chí Tây (51303574)

SVTH 3: Nguyễn Đình Dũng (51300665)

SVTH 4: Nguyễn Lê Minh Khôi (51301906)

TP Hồ Chí Minh, 01/2018

LỜI CAM ĐOAN

Nhóm xin cam đoan đây là công trình thực hiện của nhóm và được sự hướng dẫn của ThS. Nguyễn Cao Trí. Ngoài các thông tin thu thập được có ghi rõ nguồn trong phần Tài liệu, các phần thiết kế và hiện thực khác đều do chính nhóm thực hiện và chưa được công bố trước đây. Nhóm xin hoàn toàn chịu trách nhiệm về mặt nội dung luận văn tốt nghiệp của mình. Trường đại học Bách Khoa – Đại Học Quốc Gia Thành Phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do nhóm gây ra trong quá trình thực hiện (nếu có).

Nhóm thực hiện đề tài

LỜI CẢM ƠN

Lời đầu tiên, chúng tôi xin được gửi lời cảm ơn chân thành đến ThS. Nguyễn Cao Trí – người thầy đã cho chúng tôi cơ hội được tham gia nghiên cứu và hiện thực đề tài này. Trong quá trình hiện thực, thầy đã thường xuyên trao đổi và hỗ trợ về mặt kiến thức, ý tưởng, tài liệu và kinh phí để nhóm có thể hoàn thiện được đề tài từ giai đoạn thực tập tốt nghiệp đến giai đoạn luận văn tốt nghiệp.

Nhóm cũng xin gửi lời cảm ơn đến các thầy cô trường Đại Học Bách Khoa – Đại Học Quốc Gia Thành phố Hồ Chí Minh nói chung và các thầy cô trong khoa Khoa Học và Kỹ Thuật Máy Tính nói riêng, đã truyền đạt những kiến thức quý báu, làm nền tảng cho quá trình thực hiện đề tài. Xin gửi lời cảm ơn chân thành đến gia đình, bạn bè, những người đã luôn hỗ trợ và động viên chúng tôi trong quá trình thực hiện luận văn tốt nghiệp.

Báo cáo được thực hiện trong khoảng ba tuần, không thể tránh khỏi những sai sót, nhóm rất mong được sự đóng góp của các thầy cô để nhóm có thể hoàn thiện hơn.

Sau cùng, nhóm xin kính chúc các thầy cô trường Đại Học Bách Khoa – Đại Học Quốc Gia Thành phố Hồ Chí Minh luôn dồi dào sức khỏe để có thể tiếp tục trên con đường truyền đạt kiến thức cho các thế hệ sinh viên.

Nhóm thực hiện đề tài

TÓM TẮT

Sự phát triển của công nghệ trong thời đại hiện nay đã mang lại nhiều thay đổi tích cực trong tất cả các lĩnh vực. Và lĩnh vực nông nghiệp cũng có những bước tiến lớn trong thời đại của cuộc cách mạng công nghiệp 4.0. Hòa cùng với xu thế của thời đại, sinh viên nhóm chúng tôi đã tìm hiểu và thực hiện dự án về hệ thống công nghệ hỗ trợ dành cho nông nghiệp, đặc biệt đối với nông nghiệp trong đô thị. Sự phát triển mạnh mẽ của Internet of Things và sự cần thiết của nó dành cho nông nghiệp trong môi trường đô thị, nơi mà không có nhiều không gian dành cho nông nghiệp, đã được thực hiện ở giai đoạn Thực tập tốt nghiệp, để tài tiếp tục được phát triển ở giai đoạn Luận văn tốt nghiệp. Tại giai đoạn này, nhóm tiếp tục xây dựng hệ thống hoàn chỉnh hơn từ hình mẫu đã được thiết kế và xây dựng từ giai đoạn trước, đầy đủ hơn các chứng năng cơ bản để hỗ trợ giám sát và chăm sóc cho một hệ thống thủy canh trong môi trường đô thị. Hệ thống bao gồm phần cứng và phần mềm trên nền tảng ứng dụng web và ứng dụng di động được cải thiện về mặt giao diện, tương tác thân thiện hơn với người dùng và cung cấp các API hữu ích cho những nhà phát triển.

Mục lục

Danh sách hình vẽ	ix
Danh sách bảng	xv
Danh sách chữ viết tắt	xvii
1 GIỚI THIỆU ĐỀ TÀI	1
1.1 Ý tưởng	1
1.2 Mục tiêu	1
1.2.1 Phân nghiên cứu	1
1.2.2 Phân kết quả	2
2 CƠ SỞ LÝ THUYẾT	3
2.1 Giới thiệu về Internet of Things	3
2.1.1 Internet of Things (IoT)	3
2.1.2 Định danh	4
2.1.3 Cấu trúc của hệ thống IoT [3]	4
2.1.4 Ứng dụng	5
2.2 Giới thiệu về hệ thống thủy canh	6
2.2.1 Hệ thống dạng bắc (Wick System)	7
2.2.2 Hệ thống thủy canh tĩnh (Water Culture)	7
2.2.3 Hệ thống ngập và rút định kỳ (EBB and Flow System)	8
2.2.4 Hệ thống nhỏ giọt (Drip System)	9
2.2.5 Hệ thống màng dinh dưỡng NFT (Nutrient Film Technique)	9
2.2.6 Khí canh (Aeroponics)	10
2.2.7 Các thông số trong dung dịch thủy canh, thiết bị đo ppm	11
2.3 Hệ thống thủy canh với IoT	14
2.4 Các hệ thống thủy canh tự động hiện có trên thị trường	14
2.5 Giới thiệu về các công nghệ được sử dụng	15
2.5.1 RESTful web service [7]	15
2.5.2 Nodejs [9]	16

2.5.3	Framework Expressjs [10]	17
2.5.4	Sequelizejs [11]	17
2.5.5	Angularjs [12]	18
2.5.6	Angular 2 [13]	18
2.5.7	Framework Ionic 2 [14]	19
2.5.8	TypeScript [15]	19
2.5.9	JSON Web Token [16]	19
2.5.10	Postgresql [17]	21
2.5.11	NodeBB [18]	21
2.5.12	Git [19]	21
2.5.13	XXTEA [20]	22
2.5.14	ARM [22]	22
2.5.15	Mbed [23]	23
2.5.16	Arduino [24]	23
2.5.17	MQTT [26]	23
3	THIẾT KẾ HỆ THỐNG	25
3.1	Kiến trúc mô hình hệ thống	25
3.2	Hệ thống board điều khiển	25
3.2.1	Tổng quan mô hình board điều khiển	25
3.2.2	Hoạt động, chức năng	30
3.2.3	Kiến trúc phần mềm	45
3.3	Ứng dụng web	53
3.3.1	Sơ đồ use case	53
3.3.2	Thiết kế database	54
3.3.3	Thiết kế API	56
3.4	Ứng dụng di động	57
3.5	Giao tiếp giữa Board điều khiển và Web	58
3.5.1	Giao thức	58
3.5.2	Cấu trúc các gói tin	58

4 HIỆN THỰC HỆ THỐNG	62
4.1 Hệ thống Board điều khiển	62
4.1.1 Môi trường phát triển	62
4.1.2 Mạch chức năng đo ppm	62
4.1.3 Hệ thống board điều khiển	69
4.2 Ứng dụng web	85
4.2.1 Môi trường phát triển	85
4.2.2 Các chức năng chính	85
4.2.3 Hiện thực server API	89
4.2.4 Cấu trúc thư mục	90
4.2.5 Xây dựng database	91
4.2.6 Một số hình ảnh	93
4.3 Ứng dụng di động	102
4.3.1 Môi trường phát triển	102
4.3.2 Các chức năng chính	102
4.3.3 Một số hình ảnh	103
5 KIỂM THỬ HỆ THỐNG	107
5.1 Web server	107
5.2 Ứng dụng di động	107
5.3 Board điều khiển	107
5.3.1 Thiết bị	107
5.3.2 Các chức năng	109
5.4 Kết nối giữa ứng dụng web và board điều khiển	113
5.4.1 Dữ liệu nhận từ cảm biến	113
5.4.2 Điều khiển các actuator	113
6 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	115
6.1 Tổng kết	115
6.1.1 Kết quả đạt được	115
6.1.2 Những hạn chế của hệ thống	116

6.1.3	Thuận lợi, khó khăn trong quá trình thực hiện đề tài	116
6.1.4	Kiến thức thu thập được	117
6.2	Hướng mở rộng hệ thống	117
6.2.1	Ứng dụng web	117
6.2.2	Ứng dụng di động	118
6.2.3	Board điều khiển	118
6.3	Quá trình làm việc của nhóm	118
Tài liệu		123
A PHỤ LỤC A: CHI TIẾT CÁC USE CASE		126
B PHỤ LỤC B: CHI TIẾT CÁC API		143

Danh sách hình vẽ

1	Mô hình tương tác của mạng lưới thiết bị kết nối Internet [1].	3
2	Cả thế giới chìm trong Internet [1].	6
3	Mô hình hệ thống dạng bắc [4].	7
4	Mô hình hệ thống thủy canh tĩnh [4].	8
5	Mô hình hệ thống ngập và rút định kỳ [4].	8
6	Mô hình hệ thống nhỏ giọt [4].	9
7	Mô hình hệ thống màng dinh dưỡng NFT [4].	10
8	Mô hình hệ thống khí canh [4].	10
9	Giới hạn với cây trồng [6].	12
10	Các thành phần ion [33].	13
11	Trường hợp sử dụng dòng AC [33].	13
12	Trường hợp sử dụng dòng DC [33].	13
13	Cấu trúc một JSON [8].	16
14	Cơ chế hoạt động của Nodejs [9].	17
15	Quá trình đăng nhập bằng JWT [16].	20
16	Một ví dụ mô tả các thiết bị liên lạc với nhau thông qua MQTT broker [27].	24
17	Kiến trúc hệ thống.	25
18	Cảm biến nhiệt độ, độ ẩm [28].	26
19	Module BH1750 [29].	26
20	Module Realtime Clock DS1307 [30].	27
21	Relay [32].	28
22	Thanh ghi dịch [32].	28
23	Board ARM STM32F03C8T6 [32].	29
24	Board NodeMCU ESP8266 [32].	29
25	Mô hình tổng quan.	30
26	Hoạt động độc lập, định thời đọc dữ liệu.	31
27	Máy bơm nước.	32
28	Máy sục oxy.	32
29	Đèn led.	33

30	Hoạt động của thanh ghi dịch.	34
31	Hoạt động của các actuators.	35
32	Hoạt động của tín hiệu điều khiển relay.	35
33	Hoạt động Thời gian thực.	36
34	Actuator lifetime.	37
35	Hoạt động lưu trữ thông tin actuator.	39
36	MQTT client.	40
37	Wifi station.	41
38	Wifi AP.	41
39	Wifi application.	42
40	Cấu trúc giao thức UART.	44
41	Cách hoạt động của giao thức UART.	45
42	Cấu hình wifi.	45
43	Hoạt động của web-server hoạt động trên board Node MCU.	46
44	Truyền nhận dữ liệu.	46
45	WifiStation.	47
46	MqttClient.	47
47	UserEEPROM.	48
48	Sensor.	48
49	SerialTransport.	49
50	Webserver.	49
51	SwitchAP.	50
52	Application.	50
53	SerialTransport.	51
54	Aplication.	51
55	Actuator.	52
56	Các chức năng của khách.	53
57	Các chức năng của thành viên.	54
58	Thiết kế database.	55
59	Mô hình cụm server.	56

60	Mô hình RESTful API [37].	57
61	Mô hình các giao thức trong hệ thống.	58
62	PPM schematic.	63
63	PPM PCB mô hình 2D.	63
64	PPM PCB mô hình 3D.	64
65	Oscillator cầu Wein [34].	64
66	Cầu diode [35].	65
67	Nguồn đôi 12V - Trước [32].	66
68	Nguồn đôi 12V - Sau [32].	66
69	Jack DC [32].	67
70	IC TL074 [32].	67
71	TL074 PIN [36].	67
72	Diode Zener [32].	68
73	Tụ 0.015 μ F [32].	68
74	Tụ 0.22 μ F [32].	68
75	Cầu Diode [32].	69
76	Thư viện DHT.	69
77	Đường dẫn thư viện DHT.	70
78	Kết quả đo nhiệt độ, độ ẩm DHT.	70
79	Thư viện BH1750	71
80	Đường dẫn thư viện BH1750.	71
81	Kết quả đo cường độ ánh sáng BH1750.	71
82	Thư viện thời gian thực.	72
83	Đường dẫn thư viện thời gian thực.	72
84	Thư viện thanh ghi dịch.	73
85	Đường dẫn thư viện thanh ghi dịch.	73
86	Nguồn cho các actuator.	73
87	Thư viện mã hóa xxtea.	73
88	Đường dẫn thư viện xxtea.	74
89	Thư viện MQTT.	74

90	Đường dẫn thư viện MQTT.	74
91	Thư viện cấu trúc dữ liệu.	75
92	Đường dẫn thư viện cấu trúc dữ liệu.	75
93	Nút nhấn.	76
94	Node MCU phát ra wifi.	76
95	Truy cập vào địa chỉ để kết nối với webserver.	77
96	Thiết lập kết nối.	78
97	Thiết lập kết nối.	78
98	Cấu hình giao thức MQTT.	79
99	Schematic.	81
100	PCB mô hình 2D	82
101	PCB mô hình 3D	82
102	Mô hình thử nghiệm - 1.	83
103	Mô hình thử nghiệm - 2.	83
104	Mô hình thử nghiệm - 3.	84
105	Thao tác đăng nhập vào hệ thống sử dụng JWT.	86
106	Nhóm các thao tác tương tác với board điều khiển.	87
107	Nhóm các thao tác CRUD.	88
108	Cây thư mục mã nguồn.	90
109	Thiết kế sơ đồ lớp.	92
110	Trang chủ.	93
111	Trang tìm kiếm mùa vụ.	93
112	Trang đăng ký.	94
113	Trang đăng nhập.	94
114	Trang cá nhân.	95
115	Thêm một board mới.	95
116	Danh sách các board điều khiển.	96
117	Danh sách các actuator của một board.	96
118	Thêm một actuator.	97
119	Danh sách các mùa vụ.	97

120	Thêm một mùa vụ.	98
121	Thông tin mùa vụ.	98
122	Chi tiết mùa vụ.	99
123	Nguồn dữ liệu.	99
124	Lịch trình hoạt động của các actuator.	100
125	Thêm lịch trình cho actuator.	100
126	Trang forum.	101
127	Trang quản lý dành cho administrator/moderator.	101
128	Trang đăng nhập.	103
129	Trang chủ sau khi đăng nhập.	104
130	Thông tin mùa vụ và dữ liệu.	105
131	Danh sách các actuator.	106
132	Board prototype.	107
133	Relay prototype.	108
134	Đóng hộp.	108
135	PPM prototype.	109
136	PPM đóng vỏ.	109
137	Đo ppm dung dịch muối ăn 1.	110
138	Đo ppm dung dịch muối ăn 2.	110
139	Đo ppm dung dịch muối ăn 3.	111
140	Kết quả đo ppm 1.	111
141	Kết quả đo ppm 2.	112
142	Dữ liệu gửi lên từ cảm biến.	113
143	Đăng ký.	126
144	Đăng nhập.	126
145	Use case thay đổi thông tin.	128
146	Use case thêm thiết bị.	130
147	Use case tương tác với diễn đàn.	131
148	Use case tìm kiếm mùa vụ.	133
149	Use case xem thông tin thiết bị.	134

150	Use case sửa thông tin thiết bị	135
151	Use case tạo relay.	136
152	Use case thêm mùa vụ mới.	138
153	Use case sửa thông tin mùa vụ.	139
154	Use case cài đặt chương trình mùa vụ.	140
155	Use case cài đặt ngưỡng dữ liệu.	141

Danh sách bảng

1	Các gói tin trao đổi giữa STM và Node MCU.	43
2	Cấu trúc chung của các gói tin.	58
3	Cấu hình.	59
4	Lịch trình cho actuator.	59
5	Điều khiển actuator.	59
6	Dữ liệu gửi từ các cảm biến.	60
7	Đồng bộ thời gian thực.	60
8	Thêm bớt actuator.	60
9	Gói tin phản hồi.	61
10	Thêm, bớt một board điều khiển.	61
11	Nối chân DHT-NodeMCU	69
12	Nối chân BH1750 – Node MCU	70
13	Nối chân PPM – Node MCU	72
14	Nối chân DS1307 – STM32F103C8T6	72
15	Nối chân 74HC595 – STM32F103C8T6	72
16	Các trường hợp giá trị dữ liệu.	74
17	Bảng công việc của Huỳnh Bá Thạch.	119
18	Bảng công việc của Tạ Chí Tây.	120
19	Bảng công việc của Nguyễn Lê Minh Khôi.	121
20	Bảng công việc của Nguyễn Đình Dũng.	122
21	Mô tả use case đăng ký.	127
22	Mô tả use case đăng nhập.	127
23	Mô tả use case hay đổi thông tin cá nhân.	128
24	Use case thay đổi hình ảnh cá nhân.	129
25	Mô tả use case thay đổi mật khẩu.	129
26	Mô tả use case thêm thiết bị.	130
27	Mô tả use case tạo bài viết mới.	131
28	Mô tả use case xóa bài viết.	132
29	Mô tả use case thêm bình luận bài viết.	132

30	Mô tả use case xóa thiết bị.	132
31	Mô tả use case tìm kiếm mùa vụ.	133
32	Mô tả use case gửi bình luận mùa vụ.	134
33	Mô tả use case sửa thông tin thiết bị.	135
34	Mô tả use case tạo relay.	136
35	Mô tả use case xóa relay.	137
36	Mô tả use case sửa relay.	137
37	Mô tả use case tạo mùa vụ mới.	138
38	Mô tả use case sửa thông tin mùa vụ.	139
39	Mô tả use case xóa mùa vụ.	140
40	Mô tả use case cài đặt chương trình cho mùa vụ.	141
41	Mô tả use case cài đặt ngưỡng dữ liệu.	142

Danh sách chữ viết tắt

1. **IoT** Internet of Things
2. **RFID** Radio Frequency Identification
3. **NFC** Near Field Communication
4. **IP** Internet Protocol
5. **API** Application Programming Interface
6. **CSDL** Cơ Sở Dữ Liệu
7. **HTML** Hypertext Markup Language
8. **MVC** Model - Controller - View
9. **SPA** Single Page Application
10. **UI** User Interface
11. **IDE** Integrated Development Environment
12. **CRUD** Create, Read, Update, Delete

1 GIỚI THIỆU ĐỀ TÀI

1.1 Ý tưởng

Công nghệ thủy canh đã được nghiên cứu từ thế kỷ 17. Đến nay, công nghệ này đã hoàn thiện và hướng tới việc sản xuất ra những sản phẩm nông nghiệp sạch. Công nghệ thủy canh có thể áp dụng với quy mô lớn trong các nông trại hoặc với quy mô nhỏ trong các hộ gia đình. Nhóm đang hướng tới đối tượng là các hộ gia đình ở thành phố, nơi diện tích chật hẹp và không có đất để trồng trọt. Họ có thể tận dụng các khoảng không nhỏ trong nhà như sân thượng, lan can, v.v... để đặt một giàn thủy canh nhỏ, cung cấp nguồn nông sản sạch cho gia đình. Đồng thời hướng đến việc tự động hóa các công việc chăm sóc, người dùng không phải mất nhiều thời gian để chăm sóc cho cây trồng nhưng chúng vẫn có thể tự phát triển và cho năng suất cao.

Trên thế giới hiện đã có những ứng dụng sử dụng IoT để tự động hóa hoàn toàn quy trình trồng rau sạch thủy canh nhà phố, trong khi ở Việt Nam còn khá mới mẻ. Với những yếu tố trên, nhóm đã hiện thực và phát triển hệ thống hỗ trợ cho mô hình thủy canh theo hướng kết hợp IoT với mục đích giúp người dùng tạo ra nông sản sạch, không ô nhiễm.

1.2 Mục tiêu

1.2.1 Phân nghiên cứu

- Tìm hiểu và thực hành quá trình thiết kế và xây dựng một dự án.
- Tìm hiểu nắm vững các nền tảng, thư viện cũng như các công cụ để thực hiện dự án này.
- Áp dụng các kiến thức về thiết kế mềm, kỹ thuật lập trình vào dự án.
- Tổ chức dự án cho phép dễ dàng bảo trì, mở rộng.
- Tìm hiểu về hệ thống các mô hình thủy canh:
 - Tổng quan, khái niệm về thủy canh.
 - Các mô hình thủy canh và cách vận hành của từng mô hình.

- Học hỏi và biết cách sử dụng các công nghệ mới hiện nay như Nodejs, Angularjs, Angular 2, Ionic, ...

1.2.2 Phân kết quả

- Hiện thực mạch đo giá trị ppm (parts per million) trong môi trường dung dịch.
- Hệ thống board điều khiển với các cảm biến (nhiệt độ, độ ẩm, ánh sáng, ppm) và các relay điều khiển (máy bơm nước, quạt, đèn) có thể đáp ứng các tiêu chí:
 - Vận hành trong thời gian dài, ổn định.
 - Chức năng thu thập dữ liệu các cảm biến phải hoạt động tốt, ổn định, sai số chấp nhận được.
 - Chức năng điều khiển các thiết bị như máy bơm nước, sục oxy, đèn chiếu sáng, v.v... đáp ứng thời gian thực.
 - Các kết nối về sóng wifi ổn định, cung cấp chức năng cấu hình cho module wifi.
 - Protocol giao tiếp giữa board điều khiển và web-server hoạt động chính xác.
 - Thiết kế hiện thực board điều khiển về mặt phần cứng, phần khung, vỏ bảo vệ.
- Xây dựng website cho người dùng với các chức năng:
 - Thu nhận các thông tin từ thiết bị gửi lên từ các cảm biến.
 - Phân tích các thông tin gửi lên, giám sát, đưa ra cảnh báo và giải pháp xử lý đối với trường hợp các thông tin vượt ngưỡng cho phép.
 - Tạo một lịch trình và nạp xuống cho thiết bị.
 - Theo dõi thường xuyên, upload hình ảnh về các mùa vụ đang trồng.
 - Chia sẻ thông tin của mình cho các người dùng khác, xây dựng một cộng đồng chung để mọi người trao đổi và chia sẻ thông tin trong lĩnh vực thủy canh.
- Hiện thực mô hình thủy canh thu nhỏ cho demo.

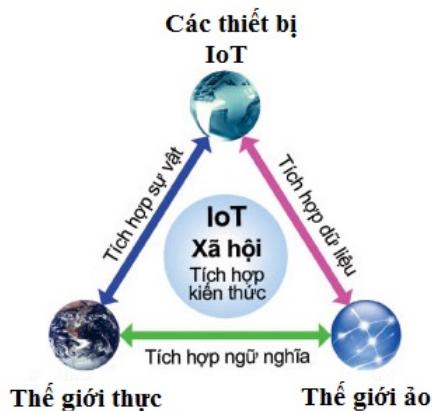
2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Internet of Things

2.1.1 Internet of Things (IoT)

Internet of Things là khái niệm về một hệ thống mạng mà ở đó mọi vật đều được cung cấp một định danh và có khả năng trao đổi thông tin, dữ liệu qua hệ thống mạng đó [1]. Hệ thống mạng đó chính là mạng Internet, tất cả mọi thứ được kết nối với nhau, người dùng có thể quản lý các thiết bị thông qua mạng bằng các thiết bị như PC, smartphone, tablet, v.v... Các thiết bị sẽ giao tiếp với nhau, hạn chế sự tác động của con người. Theo số liệu dự đoán của Gartner (công ty nghiên cứu và tư vấn công nghệ thông tin hàng đầu thế giới) thì đến năm 2020 sẽ có khoảng 20,4 tỷ thiết bị được kết nối được sử dụng [2]. Với sự phát triển của các thiết bị thông minh ngày càng gia tăng, IoT được xem như tương lai của thế giới.

Chúng ta có thể hình dung rằng mọi thiết bị trong hệ thống kết nối, hoạt động cùng với nhau. Khi ta bước gần về đến cửa nhà, cơ chế điều khiển sẽ tự động mở cửa từ xa, đèn cửa và hành lang được kích hoạt. Hệ thống điều hòa đang từ trạng thái chờ sẽ chuyển sang trạng thái hoạt động. Mọi thứ điều hoạt động một cách tự động, hài hòa nhờ sự kết nối với nhau thông qua mạng Internet. Đó chính là những gì IoT mang lại cho chúng ta.



Hình 1: Mô hình tương tác của mạng lưới thiết bị kết nối Internet [1].

2.1.2 Định danh

Để các thiết bị có thể kết nối, chúng cần có một định danh riêng để phân biệt với nhau [1]. Trong môi trường Internet, việc truyền nhận thông tin giữa các thiết bị cần thiết phải có địa chỉ IP dành cho mỗi thiết bị. Hiện tại địa chỉ IP phổ biến thường dùng là IPv4 (mỗi địa chỉ có 32 bit). Với sự phát triển và tăng nhanh về mặt số lượng của các thiết bị thông minh kết nối với Internet làm cho địa chỉ IPv4 ngày càng ít. Sự xuất hiện của địa chỉ IPv6 (mỗi địa chỉ có 128 bit) đã giải quyết vấn đề về việc định danh cho cái thiết bị tham gia vào mạng Internet.

Khi mọi đối tượng được định danh sẽ làm cho việc quản lý các đối tượng trong mạng IoT trở nên dễ dàng hơn.

2.1.3 Cấu trúc của hệ thống IoT [3]

Cảm biến

Các cảm biến có chức năng thu thập các tín hiệu analog từ môi trường thông qua việc quét các thông số sang tín hiệu digital.

Xử lý cục bộ và lưu trữ

Dữ liệu được lưu trữ và xử lý cục bộ thông qua các vi điều khiển, board mạch nhúng. Xử lý các tín hiệu thông tin nhận được từ cảm biến. Các thiết bị phần cứng có khả năng giao tiếp theo các quy tắc đã được định sẵn. Chúng rất đa dạng tùy vào mục đích sử dụng.

Vi điều khiển là những thiết bị nhỏ, dễ dàng kết nối, có thể lập trình, nhiều mã nguồn mở đơn giản. Tuy nhiên vì có kích thước nhỏ nên khả năng, tốc độ xử lý không cao, RAM chỉ khoảng 2KB đối với Arduino. Chúng không chạy hệ điều hành, phần mềm chạy trên đó được gọi là Firmware, giống như một chương trình. Đối với loại máy tính mini như Raspberry Pi có thể chạy hệ điều hành đồng nghĩa với việc sức mạnh và tốc độ xử lý cao hơn, mạnh mẽ hơn.

Có nhiều thiết bị được phát triển dành cho công nghệ IoT, tùy nhu cầu mà chọn thiết bị cho phù hợp với mục đích, chi phí sử dụng. Một số thiết bị như: STM32, ARM, Arduino, ChipKIT, v.v...

Network và Internet

Các thiết bị phần cứng thu thập dữ liệu cục bộ và đẩy dữ liệu lên cloud để lưu trữ. Những

giao thức giao tiếp giữa thiết bị phần cứng và cloud thường được sử dụng:

- CoAP (Constrained Application Protocol)
- MQTT (Message Queuing Telemetry Transport)
- HTTP (HyperText Transfer Protocol)
- XMPP (Extensible Messaging and Presence Protocol)

Cloud (đám mây)

Nơi dữ liệu được thu thập, bao gồm hệ thống các máy chủ, hệ thống lưu trữ.

2.1.4 Ứng dụng

IoT có ứng dụng vô cùng rộng, trong rất nhiều lĩnh vực, có thể kể ra một số ứng dụng nổi bật hiện nay như sau:

- Nhà thông minh.
- Thành phố thông minh.
- Xe hơi tự hành.
- Vườn tự chăm sóc.
- Máy bay không người lái.
- Thiết bị theo dõi sức khỏe.
- Hệ thống chiếu sáng tự động.
- Chuỗi cung ứng thông minh.



Hình 2: Cả thế giới chìm trong Internet [1].

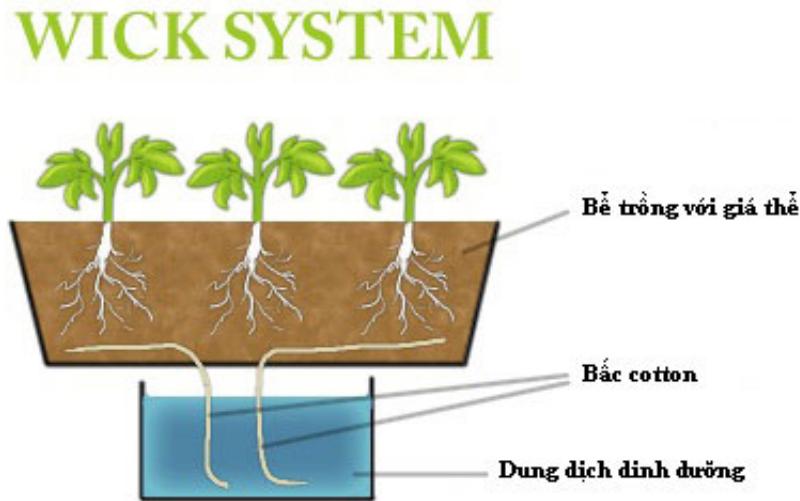
2.2 Giới thiệu về hệ thống thủy canh

Thủy canh là một kỹ thuật trồng cây trong môi trường dung dịch dinh dưỡng, có thể hiểu là trồng cây trong nước, mà không cần sử dụng đất trồng [4]. Rễ cây sẽ được giữ trong các giá thể như xơ dừa, trúu, sỏi nhẹ, mút xốp, v.v... Các chất dinh dưỡng cần thiết và các yếu tố khác như ánh sáng cho quá trình quang hợp, oxy cho quá trình hô hấp vẫn được cung cấp đầy đủ cho sự phát triển của cây.

Phương pháp thủy canh giúp có thể điều chỉnh nồng độ dinh dưỡng một cách nhanh chóng và dễ dàng tùy vào từng giai đoạn phát triển của cây. Thủy canh thích hợp cho các khu vực có diện tích nhỏ hẹp, chật chội như sân thượng, ban công nhà cửa thành phố, v.v... Quá trình đô thị hóa, dân số ngày càng tăng, diện tích đất đai bị thu hẹp thì kỹ thuật thủy canh sẽ mang lại lợi ích to lớn cho đời sống, hơn thế nữa kỹ thuật này không gây ô nhiễm môi trường, tạo ra nông sản sạch, giúp chất lượng cuộc sống nâng cao.

2.2.1 Hệ thống dạng bắc (Wick System)

Hệ thống dạng bắc là dạng hệ thống đơn giản nhất, dung dịch dinh dưỡng và nước sẽ được tháp hút tự bê chứa dung dịch dinh dưỡng thông qua một sợi bắc đến giá thể để cung cấp cho rễ cây [4].

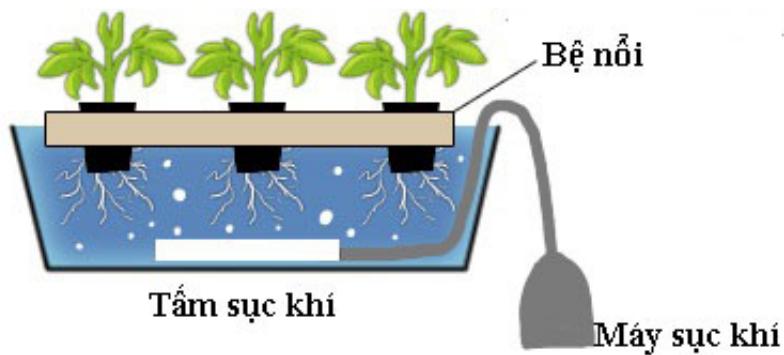


Hình 3: Mô hình hệ thống dạng bắc [4].

2.2.2 Hệ thống thủy canh tĩnh (Water Culture)

Hệ thống thủy canh tĩnh gồm thùng nước chứa dung dịch dinh dưỡng, rễ cây sẽ được giữ trong một bê dẻo nhẹ và chìm trong thùng chứa dinh dưỡng đó. Oxy sẽ được bơm thông qua một máy sục khí để cung cấp cho rễ cây. Hệ thống này thường được dùng phổ biến trong môi trường giảng dạy vì ít tốn kém, có thể tận dụng các vật chứa nước hoặc những bình chứa không rỉ khác [4].

WATER CULTURE

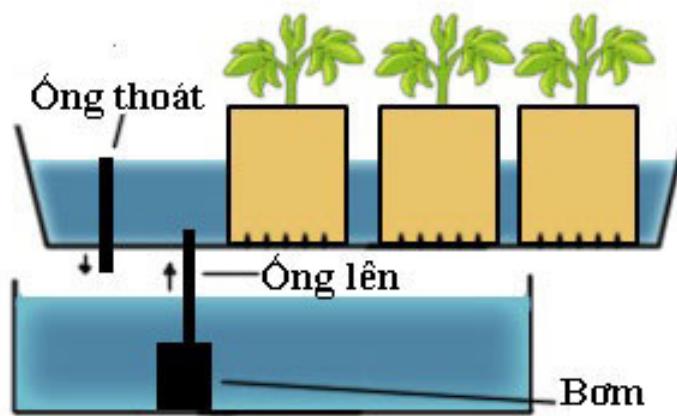


Hình 4: Mô hình hệ thống thủy canh tĩnh [4].

2.2.3 Hệ thống ngập và rút định kỳ (EBB and Flow System)

Hệ thống ngập và rút định kì hoạt động bằng cách làm trống khay trong dung dịch dinh dưỡng trong một khoảng thời gian, sau đó rút dung dịch này trở lại bồn chứa. Hoạt động được thực hiện bởi máy bơm có kết nối với bộ đếm thời gian bật tắt máy bơm để dung dịch vào khay và trở lại bồn chứa theo chu kì. Tùy thuộc vào từng loại cây mà sẽ có chu kì ngập rút dài ngắn khác nhau [4].

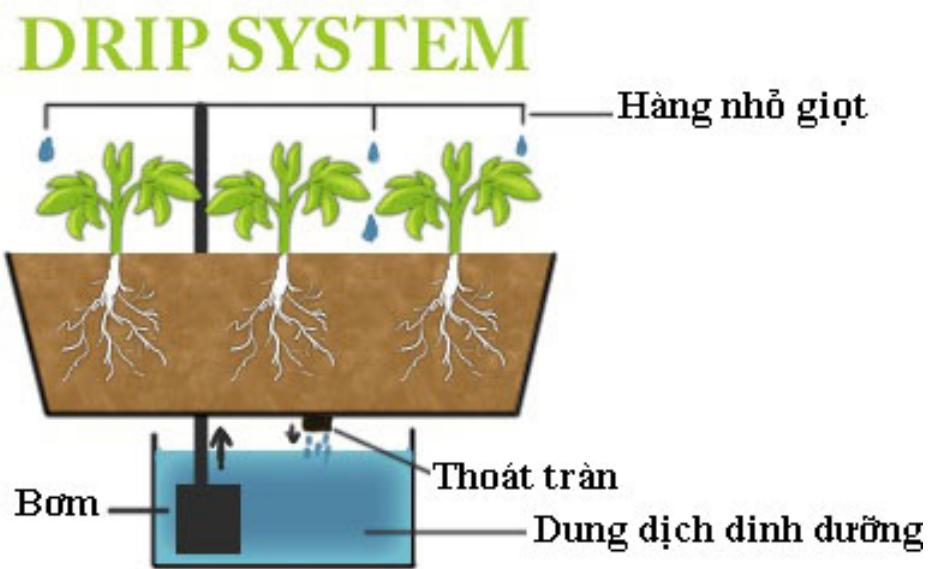
EBB AND FLOW



Hình 5: Mô hình hệ thống ngập và rút định kỳ [4].

2.2.4 Hệ thống nhỏ giọt (Drip System)

Hệ thống nhỏ giọt là hệ thống hoạt động bằng cách dùng máy bơm dung dịch lên nhỏ giọt trực tiếp vào các giá thể thông qua các đường ống dẫn, sau đó dung dịch dinh dưỡng dư sau khi thẩm qua các giá thể sẽ quay trở lại vào bể chứa để chuẩn bị được bơm lên lần kế tiếp. Quá trình này được tiếp diễn liên tục. Hệ thống này hoạt động khá hiệu quả dựa vào việc tái sử dụng dung dịch dinh dưỡng dư, không bị lãng phí [4].

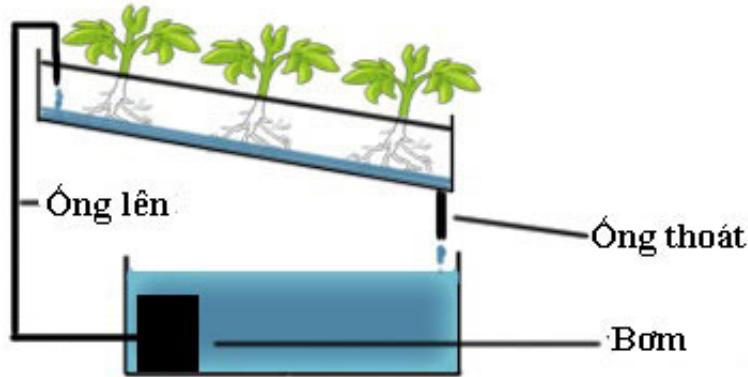


Hình 6: Mô hình hệ thống nhỏ giọt [4].

2.2.5 Hệ thống màng dinh dưỡng NFT (Nutrient Film Technique)

Hệ thống màng dinh dưỡng NFT là hệ thống hoạt động bằng việc dùng máy bơm liên tục dung dịch dinh dưỡng vào khay trồng để chảy qua rễ rồi trở lại bể chứa, quá trình này được diễn ra liên tục. Đối với kỹ thuật này thì thường không sử dụng thêm giá thể, giúp tiết kiệm về mặt chi phí giá thể [4].

NFT SYSTEM

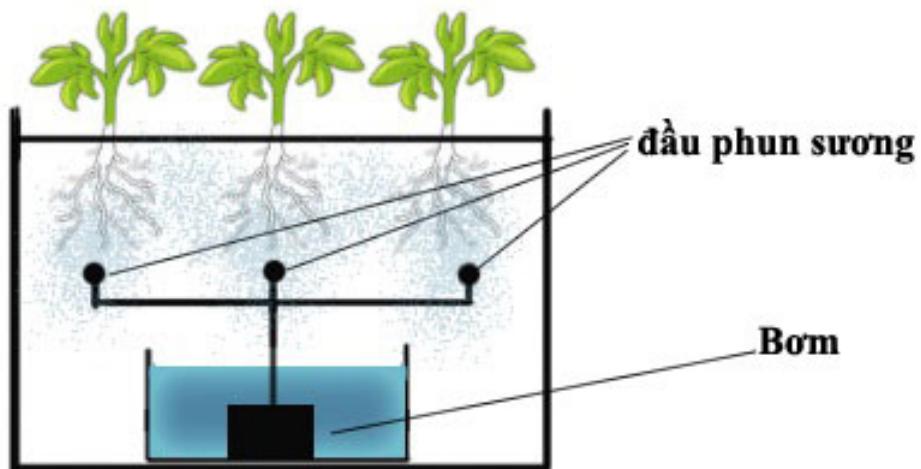


Hình 7: Mô hình hệ thống màng dinh dưỡng NFT [4].

2.2.6 Khí canh (Aeroponics)

Hệ thống khí canh hoạt động bằng cách phun sương dung dịch dinh dưỡng vào rễ cây được phơi trong không khí. Quá trình tạo sương được thực hiện định kì mỗi vài phút. Đối với kỹ thuật này cây được cung cấp đầy đủ cả chất dinh dưỡng, nước và không khí [4].

AEROPONICS



Hình 8: Mô hình hệ thống khí canh [4].

2.2.7 Các thông số trong dung dịch thủy canh, thiết bị đo ppm

- Độ dẫn điện EC (electro-conductivity) (tương đối) là chỉ số điện tản tổng nồng độ ion hòa tan trong dung dịch. Độ dẫn điện có thể được thể hiện bằng một số đơn vị khác nhau, nhưng đơn vị tiêu biểu được dùng để đo lường EC là millisiemens trên centimet (mS/cm) [6]. Chỉ số EC không diễn tả nồng độ của từng chất trong dung dịch đồng thời cũng không thể hiện mức độ cân bằng của các chất dinh dưỡng trong dung dịch. EC là thước đo độ dẫn điện từ hai đầu dò 1cm.

Trong suốt quá trình tăng trưởng, cây hấp thu khoáng chất mà chúng cần, do vậy duy trì EC ở một mức ổn định là rất quan trọng. Nếu dung dịch có chỉ số EC cao thì sự hấp thu nước của cây diễn ra nhanh hơn sự hấp thu khoáng chất. Điều này làm nồng độ dung dịch tăng cao và gây ngộ độc cho cây. Khi đó ta phải bổ sung thêm nước vào môi trường. Ngược lại, nếu EC thấp, cây sẽ hấp thu khoáng chất nhanh hơn hấp thu nước. Khi đó, nồng độ dung dịch giảm mạnh, cây sẽ không được cung cấp đầy đủ khoáng chất, chậm lớn và phát triển kém.

- Chỉ số TDS (Total Dissolved Solids) là chỉ số đo tổng lượng chất rắn hòa tan, tổng số các ion mang điện tích bao gồm khoáng chất, muối hoặc kim loại tồn tại trong một khối lượng nước nhất định. TDS thường được biểu thị bằng hàm số ml/L hoặc ppm (Parts Per Million). 1 ppm tương ứng với 1mg chất rắn hòa tan trong một lít nước. Hầu hết nước máy sẽ có chỉ số PPM rơi vào khoảng từ 200 – 400 ppm [6]. Chỉ số TDS cũng ảnh hưởng lớn đến sự phát triển của cây: Nếu TDS lên quá cao, nồng độ dung dịch vượt mức cho phép sẽ gây ra hiện tượng ngộ độc cho cây. Ngược lại, khi chỉ số TDS xuống thấp, dung dịch thủy canh sẽ không đảm bảo cung cấp đủ chất dinh dưỡng cho cây trồng.
- Tương quan giữa EC và TDS: mặc dù có một mối tương quan giữa EC và TDS nhưng chúng không giống nhau. TDS và EC là 2 tham số riêng biệt. TDS là tổng lượng chất rắn hòa tan trong nước. EC là khả năng của các chất co thể gây ra dòng điện. Lượng chất rắn như muối trong phân bón tỉ lệ trực tiếp với độ dẫn điện của nó, vì vậy lượng chất rắn cao gây độ dẫn cao. Vì khi phân bón hòa tan trong nước chúng trở thành các "ion", có mang điện tích âm hoặc dương, nên chúng sinh ra

dòng điện.

Mối quan hệ của TDS và độ dẫn đặc hiệu của nước ngầm có thể được ước lượng bằng phương trình sau:

$$\text{TDS} = k_e * \text{EC}$$

Trong đó: TDS có đơn vị mgL và EC là độ dẫn điện ở microsiemens trên mỗi centimet ở 25°C. Yếu tố tương quan k_e dao động từ 0,55 đến 0,8.

Giới hạn với cây trồng:

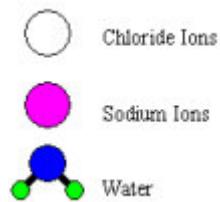
Cây trồng	EC (mS/cm)	TDS (ppm)
Cẩm chướng	2.4 - 5.0	1400 - 2450
Địa lan	0.6 - 1.5	420 - 560
Hoa hồng	1.5 - 2.4	1050 - 1750
Cà chua	2.4 - 5.0	1400 - 3500
Xà lách	0.6 - 1.5	280 - 1260
Cây chuối	2.4 - 5.0	1260 - 1540
Dâu tây	1.5 - 2.4	1260 - 1540
Ớt	1.5 - 2.4	1260 - 1540

Hình 9: Giới hạn với cây trồng [6].

- Tại sao lại dùng thiết bị này?

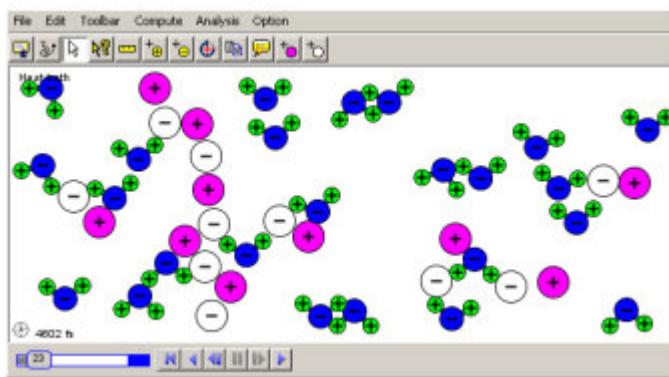
Nếu EC / PPM chỉ là đo điện dẫn (hoặc kháng) thì tại sao không sử dụng một đồng hồ volt / ohm trực tiếp? Bởi chúng đi qua dòng điện DC thông qua các đầu dò và bạn không thể đo độ dẫn của muối với dòng điện DC vì nó sẽ tách các phân tử ra ngoài, và vì các phân tử là điện dẫn điện, các phân tử điện sẽ thay đổi liên tục và sẽ không thu được kết quả gì. Bằng cách sử dụng một tín hiệu AC, với tần số đủ cao ($> 1\text{khz}$) các phân tử không có thời gian để di chuyển ra ngoài trước khi chúng được kéo theo hướng ngược lại.

Các thành phần ion:



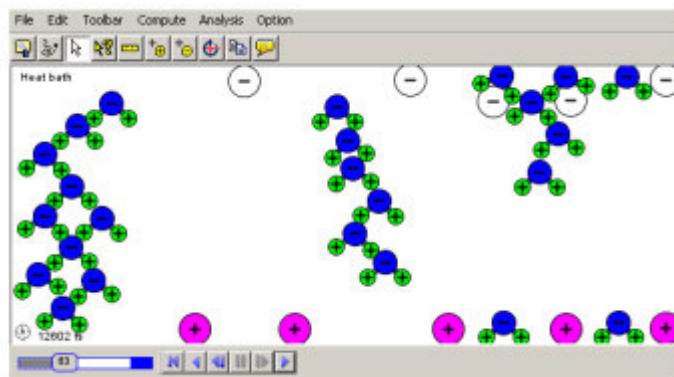
Hình 10: Các thành phần ion [33].

Trường hợp sử dụng dòng AC:



Hình 11: Trường hợp sử dụng dòng AC [33].

Trường hợp sử dụng dòng DC:



Hình 12: Trường hợp sử dụng dòng DC [33].

- Hoạt động: hai điện cực với một điện áp xoay chiều được đặt trong dung dịch. Điều này tạo ra một dòng điện phù thuộc vào bản chất dẫn điện của dung dịch. Thiết bị đọc dòng điện này và hiển thị theo đơn vị EC hoặc ppm.

2.3 Hệ thống thủy canh với IoT

Việc kết hợp thủy canh và IoT ở nước ngoài đã trở nên khá phổ biến trong khi ở Việt Nam vẫn còn mới mẻ. Việc kết hợp IoT vào thủy canh giúp người sử dụng mô hình có thể kiểm soát một cách dễ dàng tình trạng của vườn trồng thủy canh thông qua số liệu về nhiệt độ, độ ẩm, pH, v.v... mà các cảm biến gửi về qua mạng Internet. Đồng thời người dùng có thể chăm sóc vườn rau của mình mà không cần thiết phải tiếp xúc với nó nhờ các hoạt động như tưới nước, bơm oxy, cung cấp ánh sáng, quạt, v.v... đều được thiết lập theo lịch trình thông qua các ứng dụng. Người dùng chỉ việc thiết lập thông số lịch trình phù hợp cho từng cây trồng trong mỗi mùa vụ và chờ ngày thu hoạch. Kiểm soát thông qua mạng Internet cho phép người dùng quản lý vườn trồng của mình ở bất kỳ đâu, trong mọi thời điểm. Chính vì vậy mà thủy canh kết hợp với IoT là xu hướng mới và là tương lai của nông nghiệp đô thị nhờ tính tiện ích của nó.

2.4 Các hệ thống thủy canh tự động hiện có trên thị trường

Theo quá trình nghiên cứu của nhóm thì hiện nay trên thị trường chưa có nhiều hệ thống thủy canh tự động mang tính thương mại. Một số công ty start-up nổi tiếng trong lĩnh vực này như Hachi, Lisado nhưng những hệ thống này có các đặc điểm chung:

- Mới chỉ phát triển ở khu vực phía Bắc.
- Chỉ hỗ trợ nền tảng di động, chưa hỗ trợ môi trường web trên PC hay laptop.
- Chưa có một hệ thống để cộng đồng có thể chia sẻ kinh nghiệm về thủy canh với nhau.
- Chưa hỗ trợ tự động đo nồng độ dinh dưỡng ppm.
- Chưa hỗ trợ tính năng cài đặt lịch trình tự động cho các hoạt động chăm sóc vườn.

Sản phẩm của nhóm đang hướng tới sự khác biệt bằng cách tạo ra một hệ thống hỗ trợ các tính năng nói trên.

2.5 Giới thiệu về các công nghệ được sử dụng

2.5.1 RESTful web service [7]

- REST (Representational State Transfer) là một kiến trúc giao tiếp giữa client và server giúp hệ thống server có thể dễ dàng quản lý tài nguyên, được sử dụng nhiều trong việc phát triển các ứng dụng Web Services sử dụng giao thức HTTP.
- Một kiến trúc REST thường gồm một server cung cấp các API để truy xuất tài nguyên và các client truy xuất, chỉnh sửa các tài nguyên sử dụng các API đó qua giao thức HTTP.
- Các phương thức HTTP được sử dụng:
 - POST: sử dụng khi tạo một tài nguyên mới.
 - GET: dùng để truy xuất tài nguyên.
 - PUT: cập nhật tài nguyên.
 - DELETE: dùng để xóa bỏ tài nguyên.

JSON (Javascript Object Notation) [8]

- Trong hệ thống mà nhóm hiện thực sử dụng cấu trúc JSON để trao đổi dữ liệu.
- Cấu trúc dữ liệu của một JSON bao gồm nhiều cặp tên và giá trị đi kèm với nhau. JSON có 5 dạng dữ liệu chính:
 - Number: dạng số gồm số nguyên và số thực.
 - String: dạng chuỗi, nội dung được chứa trong cặp dấu nháy kép ("), những ký tự đặt biệt được escape bởi dấu \ Theo chuẩn JSON thì không sử dụng dấu nháy đơn như Javascript để bọc chuỗi.
 - Boolean: dạng luận lý gồm 2 giá trị là true và false.
 - Array: dạng mảng, gồm các phần tử phân cách nhau bởi dấu phẩy (,) và mảng được bao bởi cặp dấu ([]) và (]).

- Object: dạng đối tượng, gồm những cặp giá trị đi cùng nhau, mỗi cặp phân cách bởi dấu phẩy (,), đối tượng được bao bởi cặp dấu { và }, cặp giá trị bao gồm tên và giá trị được phân cách bởi dấu hai chấm (:).
- Null: giá trị null.



Hình 13: Cấu trúc một JSON [8].

- Ưu điểm của JSON:

- Dữ liệu trên nền cơ sở Javascript nên dễ dàng tiếp cận.
- Dữ liệu truyền tải ngắn gọn.
- Đễ chuyển đổi (parse) dữ liệu từ dạng chuỗi (nhận từ server) sang dữ liệu có thể sử dụng được (thành Object, Number, Array)...
- Đễ truy cập nội dung.

2.5.2 Nodejs [9]

- Nodejs là một nền tảng mã nguồn mở dựa trên ngôn ngữ JavaScript được dùng để xây dựng các ứng dụng chạy trên máy chủ, được phát triển trên nhiều nền tảng và hệ điều hành như Window, Linux, v.v...
- Nền tảng Nodejs hỗ trợ cung cấp nhiều module, giúp phát triển dễ dàng các ứng dụng trên máy chủ.



Hình 14: Cơ chế hoạt động của Nodejs [9].

- Nodejs có ưu điểm chạy rất nhanh vì nó thực hiện cơ chế non-blocking I/O và asynchronous (bất đồng bộ). Khi một request đến, nó không dừng lại chờ cho request đó xử lý xong mà vòng lặp event loop sẽ tiếp nhận các request liên tục và xử lý chúng liên tục. Khi một request xử lý xong, hàm callback sẽ được gọi để trả về kết quả cho client hoặc thực hiện một tác vụ nào đó. Như vậy các request sẽ được xử lý song song với nhau, tiết kiệm được rất nhiều thời gian. Nhưng đây cũng là một điểm khó đòi hỏi người lập trình phải quản lý chặt chẽ thứ tự thực hiện các hành động vì không phải lúc nào các tác vụ cũng chạy song song nhau, có những tác vụ phải thực hiện theo tuần tự.

2.5.3 Framework Expressjs [10]

- Expressjs là một framework của Nodejs.
- Hỗ trợ cung cấp cấu trúc đơn giản nhất của một dự án Nodejs (một khung sườn), giúp chúng ta có thể dựa trên đó phát triển một cách nhanh chóng một server RESTful API và có thể dễ dàng từ đó mở rộng nó.

2.5.4 Sequelizejs [11]

- Sequelize là một thư viện ORM dành cho Node.js. Nó hỗ trợ truy cập một cách dễ dàng đến PostgreSQL, MySQL, MariaDB, SQLite và MSSQL, v.v...

- ORM (Object Relational Mapping) là một kỹ thuật lập trình để ánh xạ dữ liệu giữa các CSDL quan hệ và đối tượng trong các ngôn ngữ lập trình. Trong đó, các đối tượng ánh xạ với các bảng, các quan hệ của đối tượng ánh xạ với các ràng buộc liên quan giữa các bảng.
- Sequelizejs sẽ giúp chúng ta ánh xạ dữ liệu từ cơ sở dữ liệu sang các đối tượng, hỗ trợ các hàm tương ứng để làm việc này thay vì phải viết những câu truy vấn bằng ngôn ngữ SQL.

2.5.5 Angularjs [12]

- Angular là framework dựa trên ngôn ngữ JavaScript, thường được sử dụng để xây dựng các ứng dụng Single Page Application (SPA).
- Ưu điểm của Angularjs:
 - Phát triển dựa trên Javascript.
 - Tạo các ứng dụng client-side theo mô hình MVC.
 - Khả năng tương thích cao, tự động xử lý mã javascript để phù hợp với mỗi trình duyệt.
 - Mã nguồn mở, miễn phí hoàn toàn và được sử dụng rộng rãi.

- Single Page Application là một kiểu ứng dụng web mà trong đó, công việc xử lý các thao tác của người dùng đều diễn ra trên một trang duy nhất. Người dùng sẽ có cảm giác trang web hoạt động nhanh hơn do hạn chế tối đa thao tác chuyển trang hoặc tải lại trang.

2.5.6 Angular 2 [13]

- Angular 2 là một framework để xây dựng các ứng dụng web và ứng dụng di động phía client.
- Angular 2 có những cải tiến so với Angular 1:
 - Dựa ra khái niệm Component, hầu hết các thành phần trong ứng dụng đều quy về Component làm cho ứng dụng trở nên dễ quản lý và dễ phát triển.

- Sử dụng ngôn ngữ Typescript thay vì Javascript như Angular 1.
- Tận dụng được những ưu điểm của ES6 như class, giúp cho những người đã quen với các ngôn ngữ kiểu như Java dễ dàng tiếp cận.

2.5.7 Framework Ionic 2 [14]

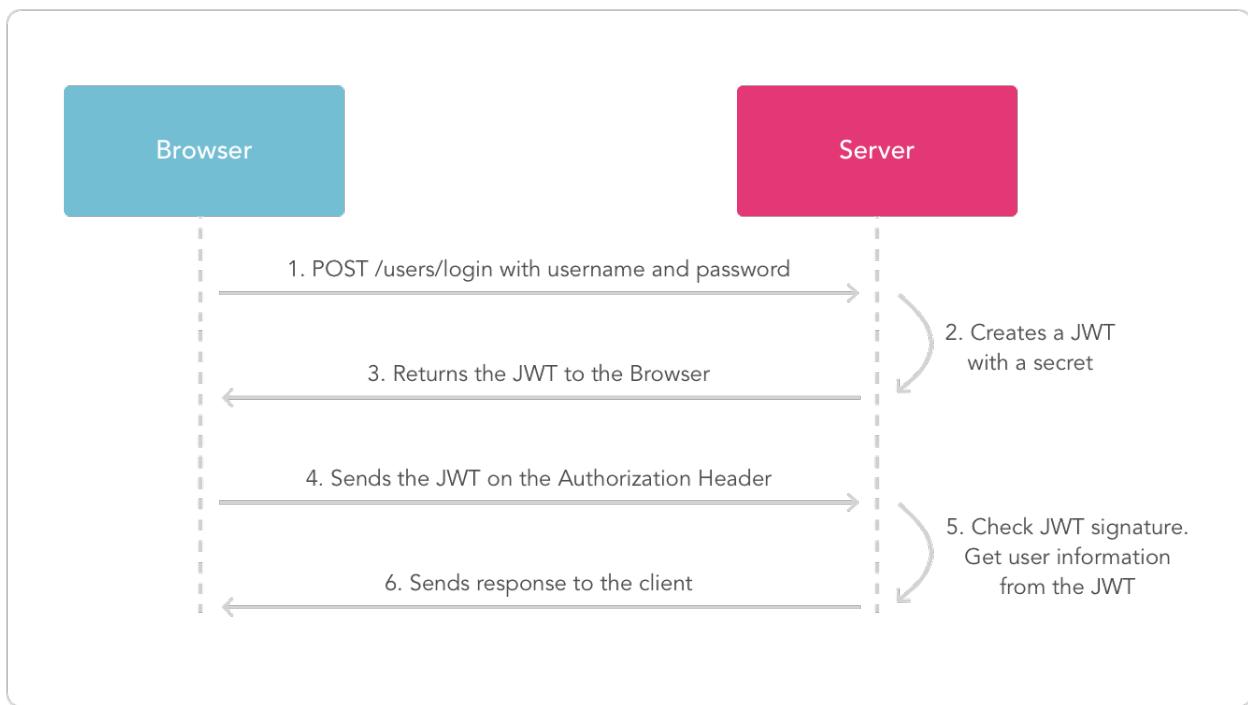
- Ionic là một framework dùng để xây dựng các ứng dụng di động trên ngôn ngữ HTML, CSS, và JavaScript, cho phép gọi các API để tạo ra các ứng dụng di động.
- Nó cần phải chạy với Angular để xử lý logic của ứng dụng và Cordova - một framework ứng dụng đa nền tảng cho phép biên dịch ứng dụng thành một tập tin có thể cài đặt và chạy nó bên trong web view của thiết bị di động.
- Ứng dụng được xây dựng với Cordova và Ionic có thể chạy trên cả thiết bị Android và iOS.
- Điểm cải tiến so với phiên bản 1 là Ionic 2 sử dụng Angular 2.

2.5.8 Typescript [15]

- TypeScript là một dự án mã nguồn mở hiện đang được phát triển và bảo trì bởi Microsoft. Nó là một ngôn ngữ cha của JavaScript, bổ sung kiểu tĩnh và lớp trên cơ sở lập trình hướng đối tượng.
- TypeScript có thể sử dụng để phát triển ứng dụng chạy cả phía client và server.
- Vì là tập cha của JavaScript nên bất kỳ chương trình JavaScript nào đã có cũng đều là chương trình TypeScript.

2.5.9 JSON Web Token [16]

- JSON Web Token (JWT) là một giao thức truyền dữ liệu an toàn giữa các thành phần trong hệ thống thông qua JSON. Các dữ liệu này có thể được xác thực độ tin cậy nhờ vào chữ ký bên trong nó, được mã hóa bằng HMAC hoặc RSA.



Hình 15: Quá trình đăng nhập bằng JWT [16].

- JWT gồm có 3 phần:

- Header

Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

```
{ "alg": "HS256", "typ": "JWT" }
```

- Payload

Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

- Signature

Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

```
HMACSHA256( base64UrlEncode(header) + "." +
base64UrlEncode(payload), secret)
```

Do bản thân Signature đã bao gồm cả header và payload nên Signature có thể dùng để kiểm tra tính toàn vẹn của dữ liệu khi truyền tải.

2.5.10 Postgresql [17]

- PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ, có thể chạy trên các hệ điều hành khác nhau như Linux, Unix, Window.
- Với đặc điểm của một hệ quản trị cơ sở dữ liệu, nó có các tính năng phức tạp như:
 - Kiểm soát truy cập đồng thời nhiều phiên bản.
 - Sao chép không đồng bộ.
 - Khôi phục dữ liệu tại từng thời điểm.

2.5.11 NodeBB [18]

- NodeBB là một framework mã nguồn mở để xây dựng diễn đàn.
- NodeBB có nhiều tính năng hiện đại của một diễn đàn và có một cộng đồng hỗ trợ phát triển, giúp hỗ trợ xây dựng lên diễn đàn dễ dàng.
- Nhóm đã sử dụng framework NodeBB để xây dựng diễn đàn kết hợp vào hệ thống ứng dụng web để tạo nơi trao đổi thông tin cho người dùng trong hệ thống.

2.5.12 Git [19]

- Git là một hệ thống quản lý những tập tin trên máy tính, các phiên bản phân tán, mã nguồn. Git sẽ lưu lại lịch sử quá trình thay đổi tại các thời điểm nên tiện lợi trong việc kiểm soát quản lý các tập tin.
- Có thể cho biết ai là tác giả của mỗi lần thay đổi tài nguyên, thông báo lỗi khi tập tin bị ghi đè lên nhau bởi nhiều người hay tạo ra một nhánh mới với mục đích sử dụng khác những vẫn giữ được các tập tin ban đầu.
- Ưu điểm:
 - Bảo mật, an toàn.

- Mọi người có thể chia sẻ tài nguyên lẫn nhau ở những nơi khác nhau.
- Dễ sử dụng, tăng hiệu quả trong công việc.
- Nhóm sử dụng Git và GitLab là những server git miễn phí để quản lý mã nguồn của hệ thống. Đường dẫn các mã nguồn của hệ thống:
 - Mã nguồn web server: <https://github.com/bathach95/hydroponic>
 - Mã nguồn mobile application: <https://github.com/bathach95/HydroponicMobileApp>
 - Mã nguồn board điều khiển: <https://gitlab.com/dungnd2/hydroponic>

2.5.13 XXTEA [20]

- XXTEA là một thuật toán mã hóa nâng cấp từ XTEA.
- XTEA (Extended Tiny Encryption Algorithm), được thiết kế bởi David Wheeler and Roger Needham của phòng thí nghiệm máy tính Cambridge, là dạng mã hóa mật mã theo khối, kích thước khối 64 bit, khóa bí mật 128 bit.
- Là một thuật toán mã hóa đơn giản dành cho các thiết bị phần cứng yếu, do đó nhóm đã sử dụng để mã hóa thông tin truyền tải giữa board điều khiển và server. Giải thuật này không được đăng ký bản quyền nên ai cũng có thể sử dụng nó tự do.

2.5.14 ARM [22]

- Cấu trúc ARM (viết tắt từ tên gốc là Advanced RISC Machine) là một loại cấu trúc vi xử lí kiểu RSIC được sử dụng rộng rãi trong các thiết kế nhúng.
- Do có đặc điểm tiết kiệm năng lượng, các bộ cpu ARM chiếm ưu thế trong các sản phẩm điện tử di động, mà với các sản phẩm này việc tiêu tán công suất thấp là một mục tiêu thiết kế quan trọng hàng đầu.
- Đi kèm với hệ thống vi xử lí là nền tảng lập trình đa dạng, các thư viện mã nguồn mở, hệ thống các board tham chiếu.

2.5.15 Mbed [23]

- Mbed là nền tảng và hệ điều hành được phát triển cho các thiết bị kết nối internet giữa trên các thiết bị 32-bit Arm Cortex-M.
- Mbed cung cấp nhiều phương pháp tiếp cận lập trình khác nhau:
 - Online: Chúng ta có thể lập trình online, biên dịch, tải file thực thi về rồi nạp vào thiết bị.
 - Offline: Lập trình trên nền tảng linux và make.

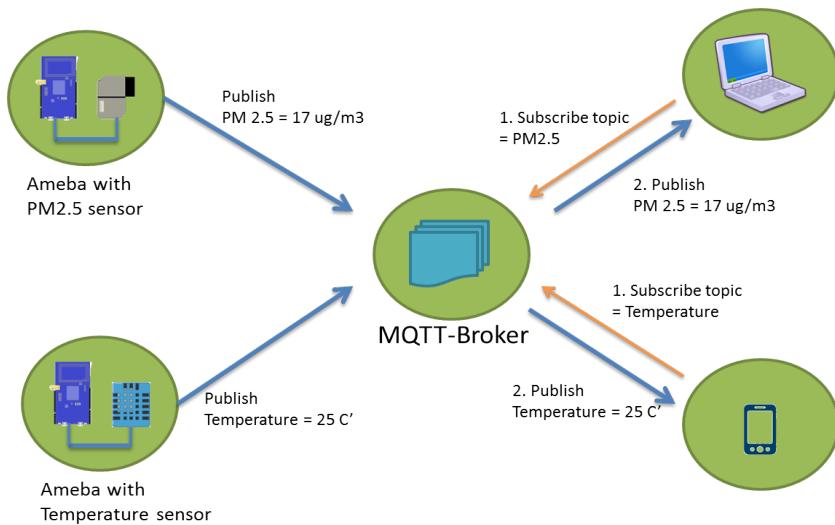
2.5.16 Arduino [24]

- Arduino là một board mạch vi xử lý, phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.
- Arduino là nền tảng mã nguồn mở được cộng đồng phát triển rộng rãi, nhiều tài liệu tham khảo nên dễ tiếp cận cho người mới bắt đầu.

2.5.17 MQTT [26]

- MQTT (Message Queuing Telemetry Transport) là một giao thức thường được sử dụng để kết nối cho các thiết bị Internet of Things.
- MQTT hoạt động theo mô hình gồm có các client kết nối và truyền tải dữ liệu với nhau thông qua một thành phần nằm ở giữa gọi là broker, thông qua giao thức TCP (Transmission Control Protocol).
- Dữ liệu từ một client sẽ được gửi (publish) vào một địa chỉ trong broker (một kênh trong broker). Các client khác nếu muốn nhận dữ liệu từ client này thì đăng ký (subscribe) vào kênh mà client đó gửi để nhận dữ liệu. Một client có thể subscribe vào nhiều kênh khác nhau. Mỗi client đã đăng ký vào một kênh sẽ nhận được dữ liệu khi bất kỳ client nào khác gửi dữ liệu vào kênh đó.

- Trong hệ thống được xây dựng, nhóm sử dụng giao thức MQTT để kết nối board và server. Với board và server đóng vai trò client và broker (mã nguồn mở) được dựng trên một máy cloud.

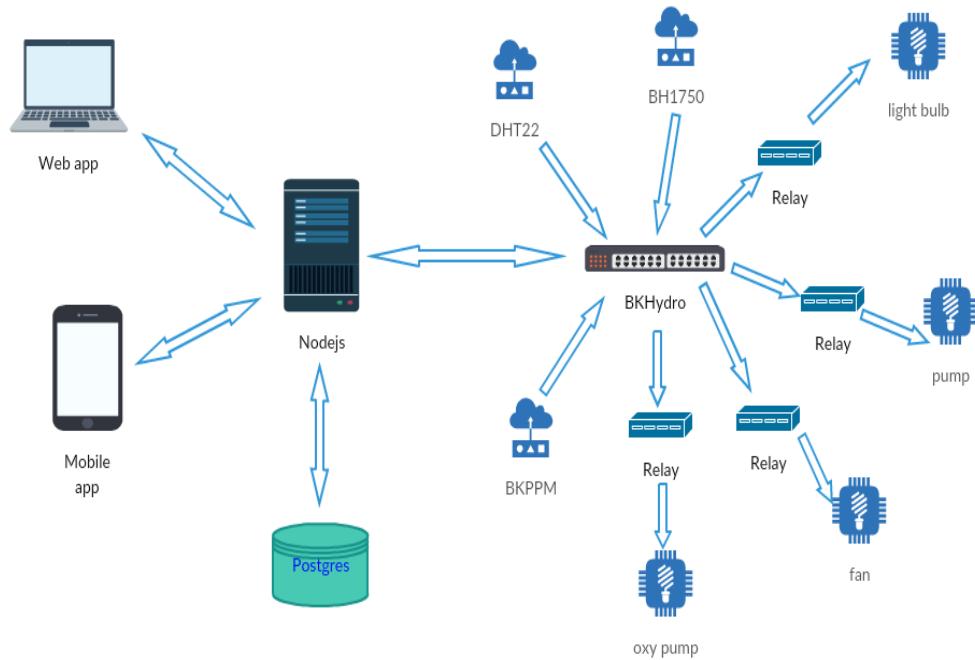


Hình 16: Một ví dụ mô tả các thiết bị liên lạc với nhau thông qua MQTT broker [27].

3 THIẾT KẾ HỆ THỐNG

3.1 Kiến trúc mô hình hệ thống

Kiến trúc hệ thống bao gồm các thành phần: web server, web app, database và hệ thống board điều khiển.



Hình 17: Kiến trúc hệ thống.

3.2 Hệ thống board điều khiển

3.2.1 Tổng quan mô hình board điều khiển

Hệ thống board điều khiển chia ra làm 3 thành phần chức năng chính:

- Thu thập dữ liệu các cảm biến.
- Định thời, điều khiển các actuator (máy bơm, sục oxi, đèn chiếu sáng...) theo thời gian thực.
- Truyền nhận dữ liệu với web-server thông qua môi trường wifi.

Các module, cảm biến của hệ thống:

- Cảm biến nhiệt độ, độ ẩm: nhiệt độ, độ ẩm được đo bởi module DHT22.



Hình 18: Cảm biến nhiệt độ, độ ẩm [28].

Thông tin kỹ thuật [28]:

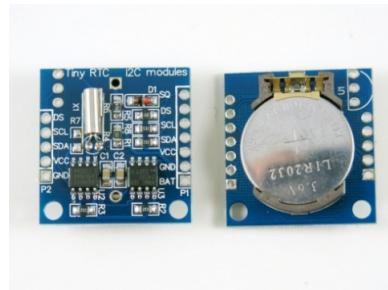
- Nguồn 3-5 VDC.
 - Dòng sử dụng: max là 2.5mA.
 - Đo tốt ở độ ẩm 0x100%RH, sai số 2-5%.
 - Đo tốt ở nhiệt độ 0-50°C, sai số 2°C.
 - Tần số lấy mẫu tối đa 0.5Hz (2 giây 1 lần).
 - Kích thước 27mm x 58mm x 13.5mm
 - 3 chân, khoảng cách 0.1 inch.
- Cảm biến đo cường độ ánh sáng: được đo bởi module BH1750.



Hình 19: Module BH1750 [29].

Thông tin kỹ thuật [29]:

- Nguồn: 3-5 VDC.
- Protocol giao tiếp: I2C
- Khoảng đo: 1->65535 lux.
- Đo tốt ở nhiệt độ 0-50°C, sai số 2°C.
- Kích cỡ: 21x16x3.3 mm.
- Thời gian thực: module Realtime clock có chức năng như một đồng hồ chạy độc lập với pin có khả năng cung cấp năng lượng tới 10 năm. Module đi kèm với bộ nhớ EEPROM AT14C32 có khả năng lưu trữ đến 32 KBit [30].



Hình 20: Module Realtime Clock DS1307 [30].

Thông tin kỹ thuật [30]:

- Nguồn cấp: 5VDC.
- Khả năng lưu trữ: 32K (EEPROM AT24C32).
- Protocol: I2C.
- Có pin chạy độc lập.
- Tần số ra: 1 HZ.
- Kích thước: 16x22x23mm.
- Relay: các actuators (Máy bơm nước, sục Oxi, led...) được kích thông qua relay.



Hình 21: Relay [32].

Thông số kỹ thuật [32]:

- Nguồn: 5 VDC.
- Điện áp kích: 5 VDC.
- Thanh ghi dịch: sử dụng chip 74HC595.



Hình 22: Thanh ghi dịch [32].

Thông số kỹ thuật [32]:

- Nguồn: 5 VDC.
- Các chân Output: 5 VDC.
- Protocol giao tiếp: SPI.

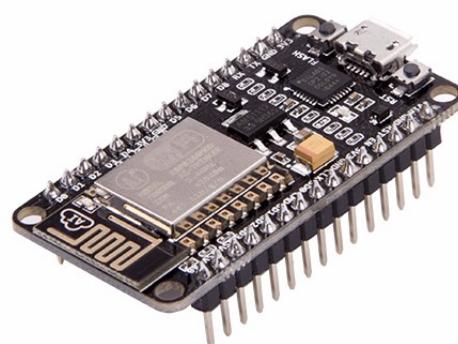
- Board ARM STM32F03C8T6: được sử dụng để làm board điều khiển các actuators.



Hình 23: Board ARM STM32F03C8T6 [32].

Thông số kỹ thuật [32]:

- RAM: 20KB.
- Flash: 64KB/128KB.
- Board NodeMCU ESP8266: được sử dụng để làm board truyền nhận dữ liệu qua wifi.



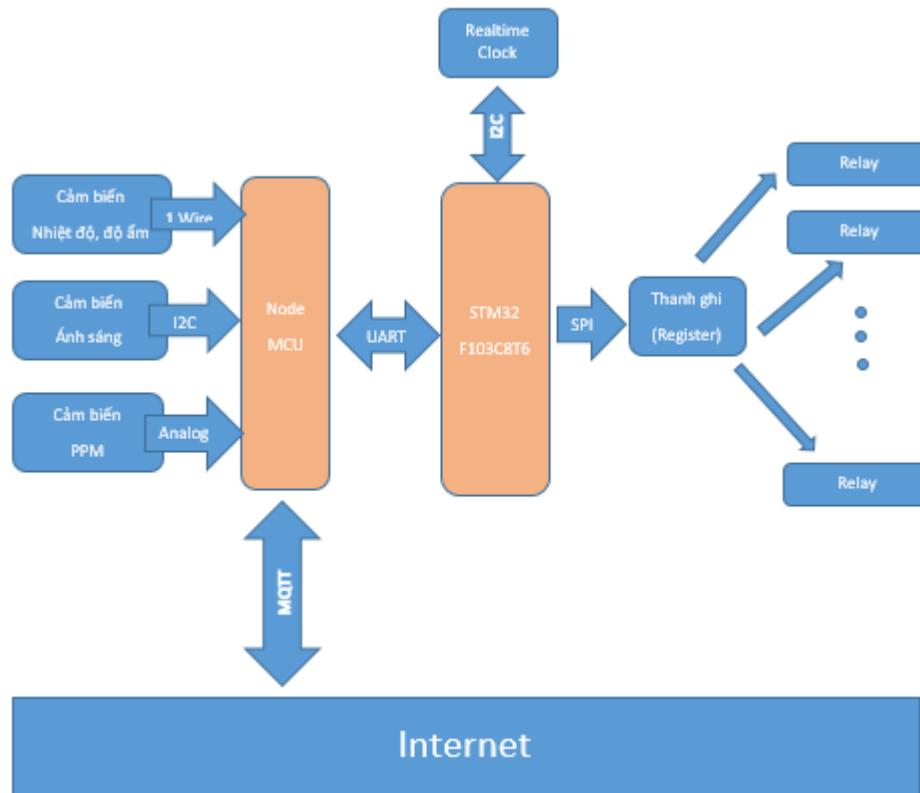
Hình 24: Board NodeMCU ESP8266 [32].

Thông số kỹ thuật [32]:

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua.
- Chip nạp và giao tiếp UART: CP2102.
- GPIO tương thích hoàn toàn với firmware NodeMCU.
- Cáp nguồn: 5VDC MicroUSB hoặc Vin.
- GPIO giao tiếp mức 3.3VDC.
- Tích hợp LED báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 25 x 50 mm.

3.2.2 Hoạt động, chức năng

Sơ đồ hoạt động:



Hình 25: Mô hình tổng quan.

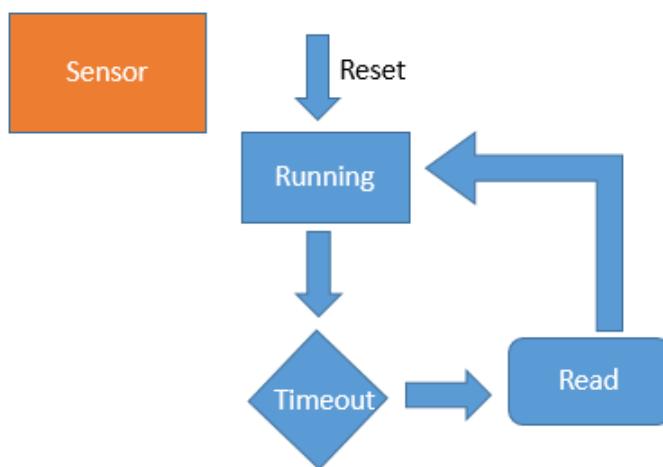
Thu thập dữ liệu cảm biến:

Các cảm biến được kết nối trực tiếp với board NodeMCU và lấy dữ liệu trực tiếp.

- Nhiệt độ, độ ẩm: module DHT22 giao tiếp 1 wire với board NodeMCU. Dữ liệu nhiệt độ (C), độ ẩm (%) được đọc sau mỗi (1s).
- Cường độ ánh sáng: module BH1750 giao tiếp I2C với board NodeMCU. Dữ liệu cường độ ánh sáng (lux) được đọc sau mỗi (1s).
- Module ppm hoạt động độc lập, đọc dữ liệu từ môi trường dung dịch, xuất đầu ra analog được đọc bởi board NodeMCU sau mỗi (1s).
- Định thời điểm đọc dữ liệu cảm biến sử dụng timer, timeout là 1s.

Hoạt động đọc, quản lý dữ liệu cảm biến sẽ được thực hiện bởi 1 đối tượng:

- Quản lý các thông tin: nhiệt độ, độ ẩm, PPM, ánh sáng.
- Hoạt động độc lập, định thời điểm đọc dữ liệu theo sơ đồ Hình 26.



Hình 26: Hoạt động độc lập, định thời điểm đọc dữ liệu.

Điều khiển các actuators

Thông số các actuator:

- Máy bơm nước:



Hình 27: Máy bơm nước.

- Nguồn: 12 VDC.
 - Công suất: 500L/h
 - Khả năng phun nước độ cao 2m phương thẳng đứng.
- Máy sục oxy:



Hình 28: Máy sục oxy.

- Nguồn: 12 VDC.
- Đèn chiếu sáng:

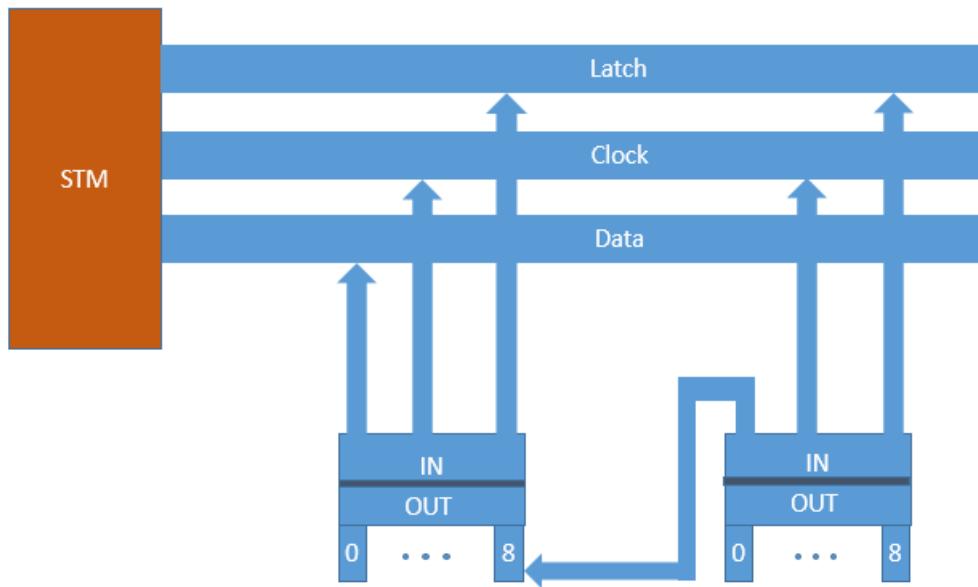


Hình 29: Đèn led.

- Nguồn: 12 VDC.
- Công suất: 15W/m

Cách điều khiển các actuator:

- Hoạt động của thanh ghi dịch:



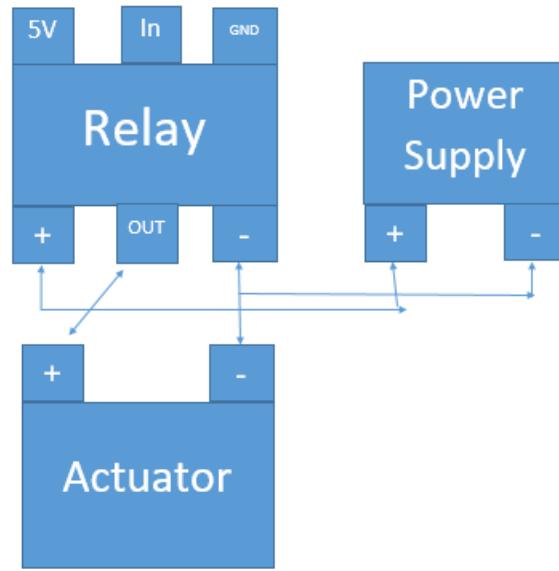
Hình 30: Hoạt động của thanh ghi dịch.

Giao tiếp của thanh ghi dịch với board điều khiển là SPI, các chân điều khiển:

- DATA: chân dữ liệu.
- LATCH: dây dữ liệu từ thanh ghi bên trong chip xuống các pinout, cụ thể là 8 chân, điện áp ra là 5V.
- CLOCK: tín hiệu clock lấy từ board điều khiển.

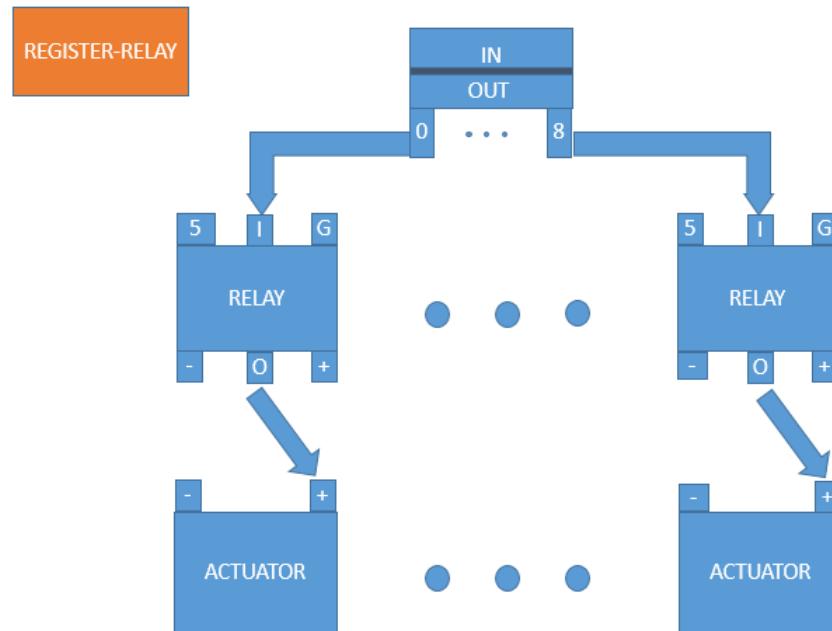
Chân DATA của thanh ghi đầu tiên sẽ nối với chân tín hiệu đầu ra của board điều khiển, DATA của các thanh ghi tiếp theo sẽ lấy từ pinout thứ 9 của thanh ghi liền trước nó.

- Các actuators sẽ được kích hoạt trực tiếp thông qua relay:



Hình 31: Hoạt động của các actuators.

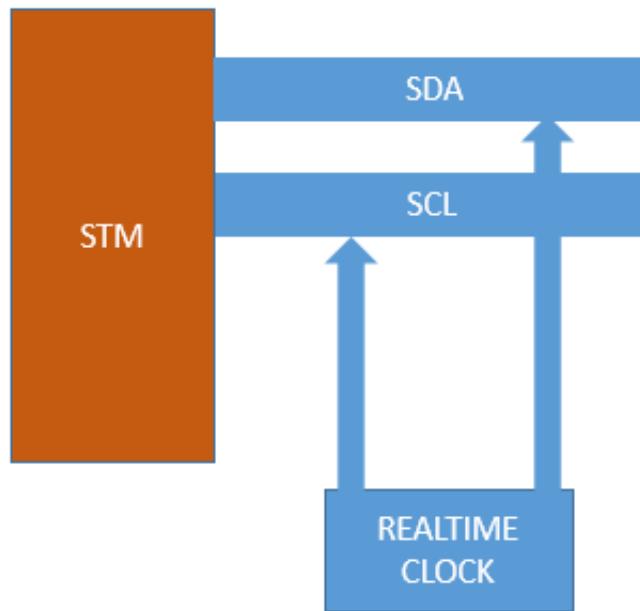
- Tín hiệu điều khiển relay sẽ lấy từ thanh ghi:



Hình 32: Hoạt động của tín hiệu điều khiển relay.

Thời gian thực:

- Module Realtime Clock DS1307 sẽ hoạt động độc lập với pin, cù thê, các thông số thời gian thực trong module không bị ảnh hưởng bởi nguồn nuôi của toàn mạch.
- Thời gian thực (year, month, day, hour, min, sec) có thể được đọc và ghi bởi board điều khiển, cù thê là NodeMCU.



Hình 33: Hoạt động Thời gian thực.

- Dịnh thời đọc dữ liệu thời gian thực sử dụng timer, timeout là 1/3s.

Tổ chức, quản lí hoạt động của các actuators.

- Chân điều khiển: Các actuator được điều khiển (tắt, mở) thông qua tín hiệu các chân của thanh ghi dịch (74HC595) và relay nên không phụ thuộc vào các pinout của board điều khiển (STM32F103C8T6). Có thể tăng số chân điều khiển (cũng như tăng số actuators) bằng cách thêm các thanh ghi dịch, số pinout sử dụng luôn là 3, cù thê là các chân SPI (DATA, LATCH, CLOCK).

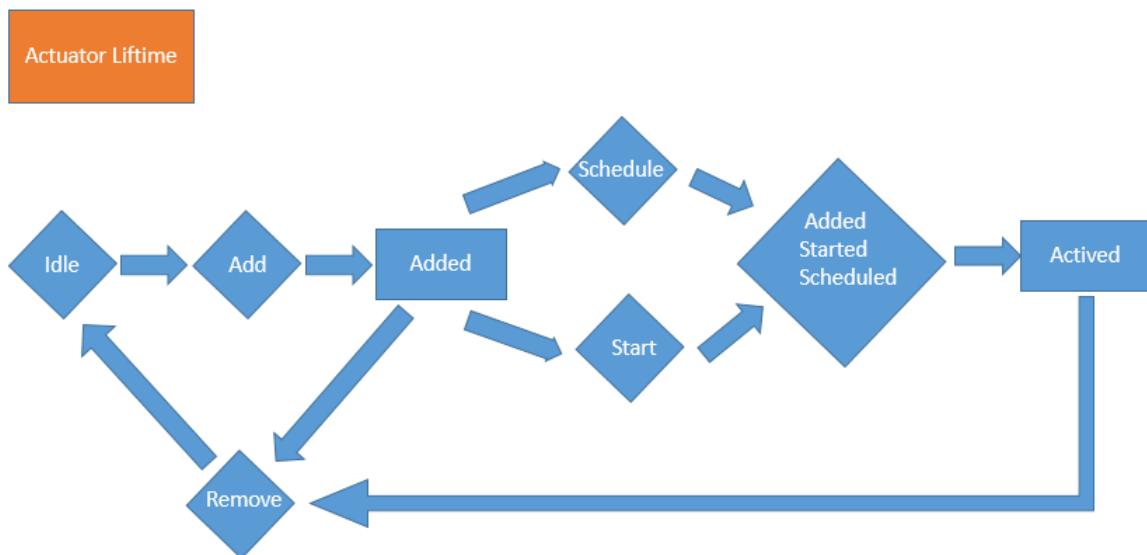
- Thời gian biểu:

Mỗi actuator sẽ hoạt động theo một thời gian biểu, cù thê trong thủy canh:

Start	Duration	Repeat Interval	Stop
Thời điểm bắt đầu hoạt động.	Thời lượng actuator được mở.	Thời lượng tắt actuator giữa hai lần mở.	Thời điểm tắt actuator cho đến thời điểm Start kế tiếp.

Định dạng thời gian cho mỗi kiểu như trên là hhmmss. Mỗi thời gian biểu cho mỗi ngày sẽ bao gồm nhiều mẫu như trên, ví dụ với máy bơm: Máy bơm sẽ bắt đầu bơm nước từ 08:00:00 (start), mỗi lần tưới sẽ kéo dài trong 15s (duration), sau đó sẽ tắt trong 10s (repeat interval), lại bật tiếp 15s, cứ thế cho đến 09:00:00 (stop).

- Lifetime của mỗi actuator (phần mềm).

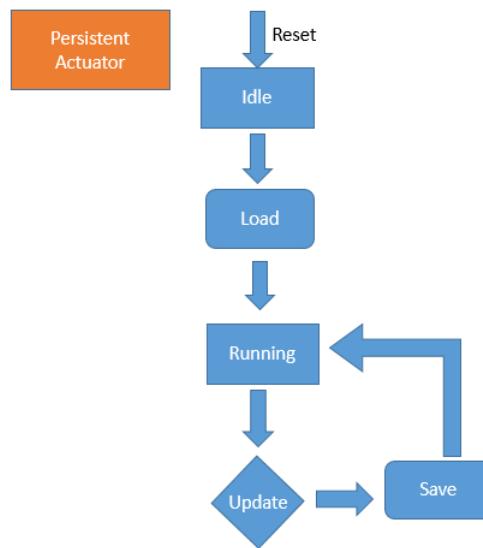


Hình 34: Actuator lifetime.

Mỗi actuator sẽ là một đối tượng (phần mềm), giải thích lifetime:

- Ban đầu, sẽ không có actuator nào, các actuator không được sét tĩnh.
- Khi nhận gói tin (Add) từ web-server, một đối tượng actuator sẽ được tạo ra, là một node trong danh sách liên kết (linked-list) các actuators. Mỗi đối tượng chứa các thông tin:

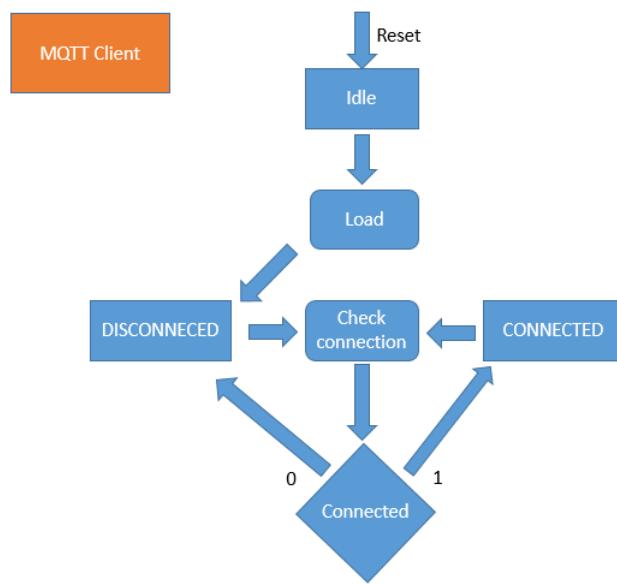
- * Actuator ID: định danh của actuator.
- * Primary: là actuator chính hay phụ.
- * Schedule: thời gian biểu hoạt động.
- * Started: điều kiện để hoạt động.
- Sau khi được add (Added), đối tượng sẽ chờ nhận thêm hai gói tin (thứ tự bắt kì) là (Start) và (Schedule) để chuyển sang trạng thái (Actived).
 - * Nhận gói (Schedule): đối tượng sẽ lưu lại thông tin schedule.
 - * Nhận gói (Start): đối tượng sẽ cập nhập thông tin (Started).
- Ở trạng thái (Added) và (Actived) khi nhận được gói (Remove) , đối tượng sẽ bị xóa.
- Ở trạng thái (Actived), đối tượng actuator sẽ hoạt động với thông tin (Schedule) theo thời gian thực đọc được từ module Realtime Clock DS1307.
- Lưu trữ thông tin actuator.
 - Để đáp ứng tính độc lập của board điều khiển, các thông tin của mỗi actuator sẽ được lưu lại trong bộ nhớ ROM, tránh trường hợp reset của board (có thể do nhiều nguyên nhân khác nhau: mất điện, crash..).
 - Thông tin actuator được thêm trong lúc board đang hoạt động ở thời điểm ngẫu nhiên (runtime), không sét tĩnh, đây là ưu điểm của giải pháp này, cho phép mở rộng, thay đổi các actuator.
 - Bộ nhớ lưu trữ là EEPROM, phân vùng lưu trữ sẽ nêu cụ thể trong phần (Lưu trữ dữ liệu) nằm trên board NodeMCU.
 - Cấu trúc dữ liệu lưu trữ được nêu cụ thể trong phần hiện thực.
 - Dưới đây là cách hoạt động của việc lưu trữ (Save) và tải lại (Load). Qui ước: Các sự kiện (Add), (Start), (Schedule), (Update), (Remove) sẽ được gọi chung là (Update). Các trạng thái (Added), (Actived) được gọi chung là (Running).



Hình 35: Hoạt động lưu trữ thông tin actuator.

Truyền nhận dữ liệu với giao thức MQTT

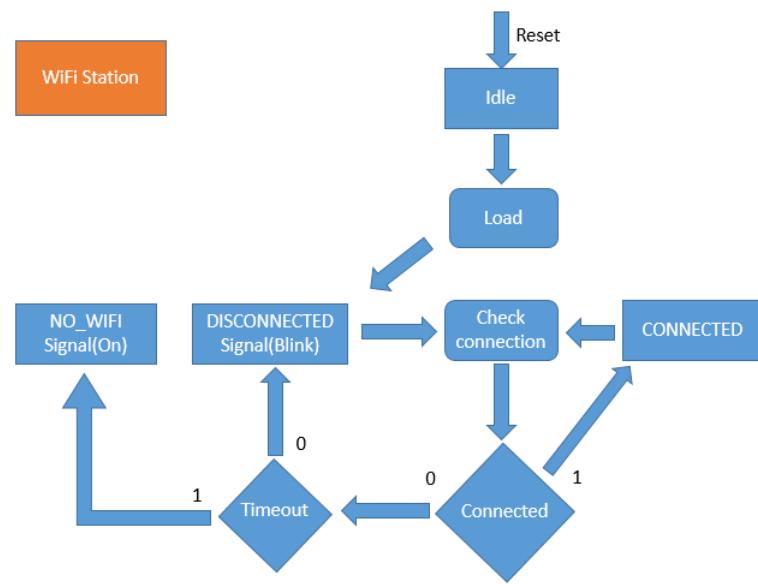
- Đối tượng thực hiện công việc này hoạt động trên board NodeMCU với vai trò là một MQTT Client, bao gồm:
 - Lưu trữ các thông tin:
 - * MQTT broker.
 - * MQTT port.
 - * MQTT subscribe channel.
 - * MQTT publish channel.
 - Các chức năng:
 - * Publish: truyền dữ liệu tới kênh web-server đăng ký (cũng đóng vai trò là một MQTT Client).
 - * Subscribe: đăng ký kênh nhận dữ liệu từ web-server.
- Các thông tin của đối tượng MQTT Client cũng được lưu trữ trong ROM, nêu trong phần (lưu trữ dữ liệu), và cho phép cấu hình (cấu hình WiFi).
- Đối tượng hoạt động độc lập theo sơ đồ dưới:



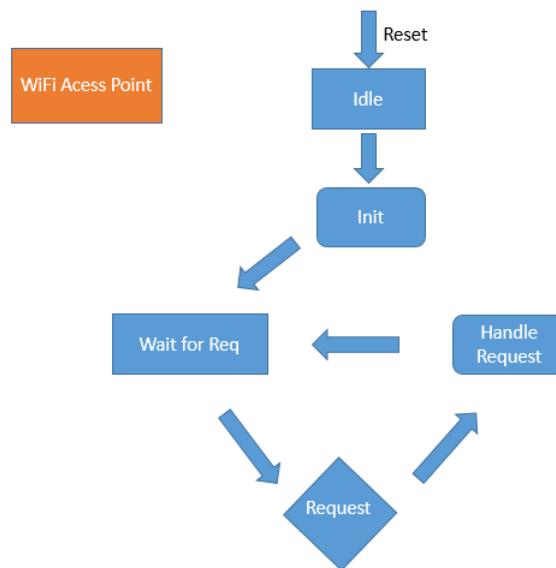
Hình 36: MQTT client.

Cấu hình, hoạt động của các chế độ WiFi

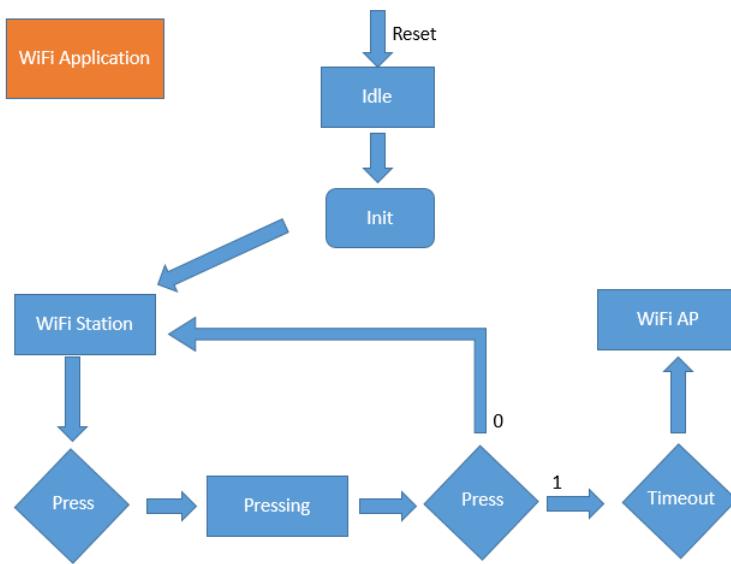
- Board NodeMCU cho phép hoạt động ở 3 chế độ:
 - Station: đóng vai trò là một thiết bị kết nối vào mạng wifi tương tự như điện thoại.
 - Access Point: Phát wifi, cho phép thiết bị khác kết nối vào.
 - Both: cả hai chức năng trên.
- Ở dự án này, 2 chế độ được sử dụng là (Station) và (Access Point), cụ thể:
 - Station: Ở chế độ này, board sẽ thực hiện chức năng của MQTT Client.
 - Access Point: cung cấp chức năng của một Web-server nhỏ, cho phép cấu hình:
 - * Các thông số wifi (tên wifi, mật khẩu) khi board hoạt động ở (Station).
 - * Thông tin của đối tượng MQTT Client (broker, port...).
 - * Các thông tin sau khi cấu hình sẽ được lưu trữ trong EEPROM.
- Chức năng chuyển chế độ sẽ được thực hiện bởi một nút nhấn.
- Các chế độ hoạt động như sau:



Hình 37: WiFi station.



Hình 38: WiFi AP.



Hình 39: WiFi application.

Đệm dữ liệu (FIFO)

- Chức năng:
 - Để đảm bảo tính ổn định, đồng bộ giữa quá trình nhận - xử lý các gói tin.
 - Giảm tải thời gian xử lý interrupt (có khả năng gây xung đột các task).
- Các vị trí cần sử dụng bộ đệm:
 - Khi nhận các gói tin serial (uart).
 - Khi gửi các gói tin MQTT: việc gửi các gói tin MQTT là thông qua môi trường wifi, nên không phải lúc nào cũng đảm bảo tính kết nối giữa client với broker, vì vậy cần bộ đệm để lưu trữ các gói tin trước khi gửi.
- Cấu trúc bộ đệm: các gói tin xếp theo thứ tự FIFO.

Lưu trữ dữ liệu

- Các thông tin cần được lưu trữ trong EEPROM:
 - Thông tin wifi station: ssid, password.
 - Thông tin MQTT: broker, port, subscribe channel, publish channel.

- Actuators.
- Việc lưu trữ và quản lý sẽ do một đối tượng thực hiện.
- EEPROM đặt trên board NodeMCU, đối tượng Actuators lại hoạt động trên board STM nên sẽ cần một giao thức để tải dữ liệu từ NodeMCU sang STM.

Giao tiếp giữa NodeMCU và STM32F103C8T6

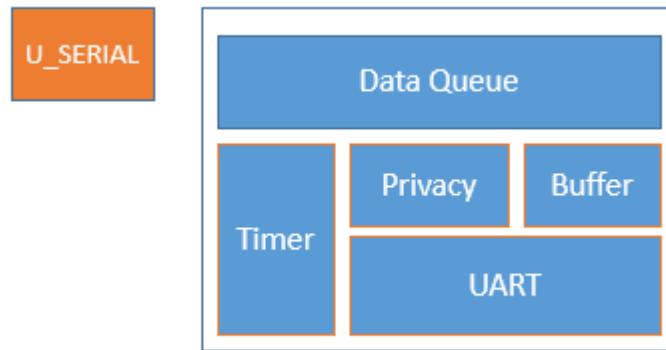
- Các gói tin trao đổi giữa STM và Node MCU:

Packet	Send
Web-server packets (1)	NodeMCU
Actuators Info (2)	STM, NodeMCU
Load actuators request (3)	STM
Load done (4)	Node MCU
Setting (5)	STM

Bảng 1: Các gói tin trao đổi giữa STM và Node MCU.

- (1) Các gói tin NodeMCU nhận được từ web - server sẽ được chuyển cho STM.
- (2) Dữ liệu trạng thái các actuators:
 - * STM gửi NodeMCU: Actuator nào thay đổi trạng thái (update) sẽ được gửi từ STM để cập nhật trạng thái tại bộ nhớ EEPROM trên NodeMCU.
 - * NodeMCU gửi STM: khi nhận được yêu cầu tải lại (load) từ STM, NodeMCU sẽ tải lại dữ liệu trên EEPROM và gửi cho STM.
- (3) Mỗi khi reset, board STM sẽ tải lại dữ liệu trạng thái các actuator bằng cách gửi gói tin này.
- (4) NodeMCU thông báo đã tải hết dữ liệu các actuators.
- (5) STM thông báo nếu đang trong thời gian hoạt động mùa vụ cho NodeMCU, khi nằm ngoài thời gian hoạt động, NodeMCU sẽ không gửi dữ liệu các cảm biến cho web - server.
- Giao thức lớp dưới được sử dụng là UART.
- Một giao thức đơn giản được xây dựng trên lớp UART nhằm:

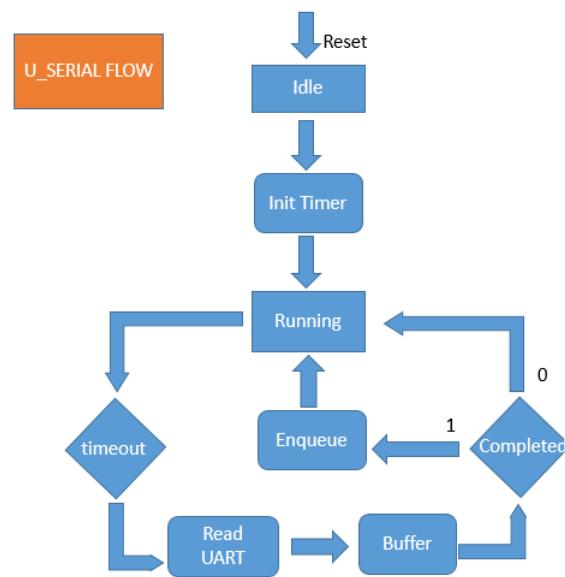
- Truyền, nhận các gói tin lớn.
- Đảm bảo tính toàn vẹn của gói tin.
- Cấu trúc giao thức:



Hình 40: Cấu trúc giao thức UART.

* Các thành phần chính:

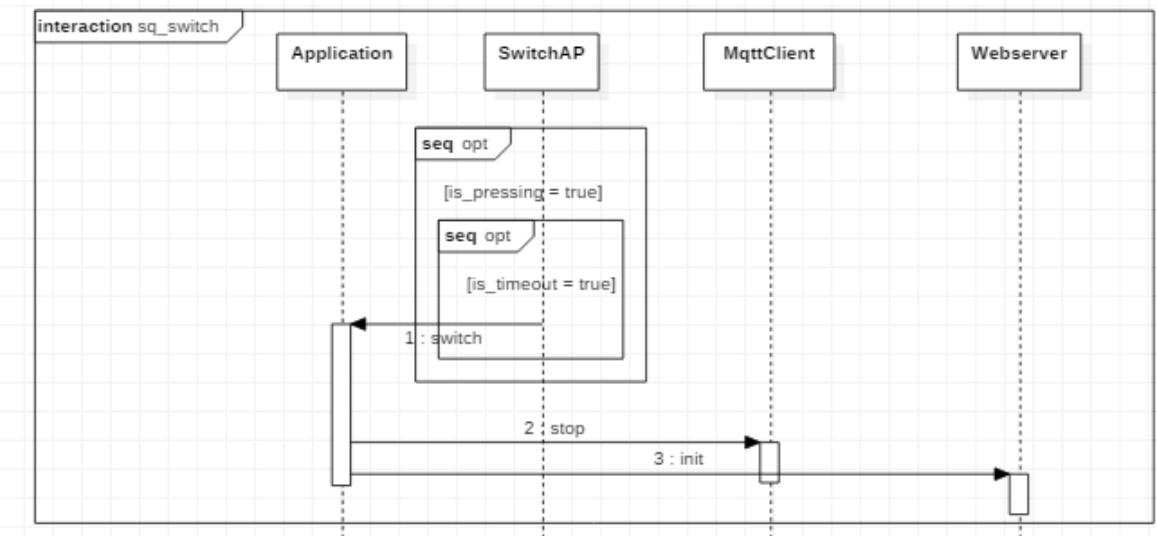
- Timer: định thời đọc dữ liệu có sẵn trong UART ở lớp dưới.
- Buffer: bộ đệm tạm thời cho dữ liệu rời rạc.
- Privacy: chính sách truyền nhận.
- Data Queue: gói tin toàn vẹn được đưa vào queue (FIFO), nếu cụ thể trong phần (Đệm dữ liệu).
- UART: giao thức truyền nhận lớp dưới.



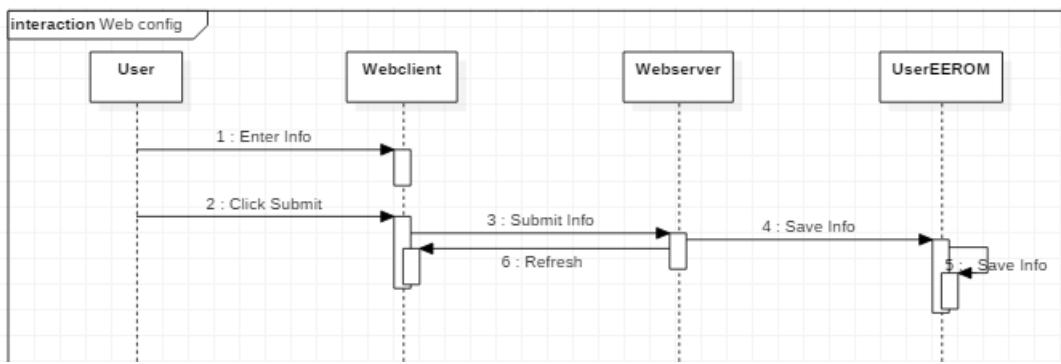
Hình 41: Cách hoạt động của giao thức UART.

3.2.3 Kiến trúc phần mềm

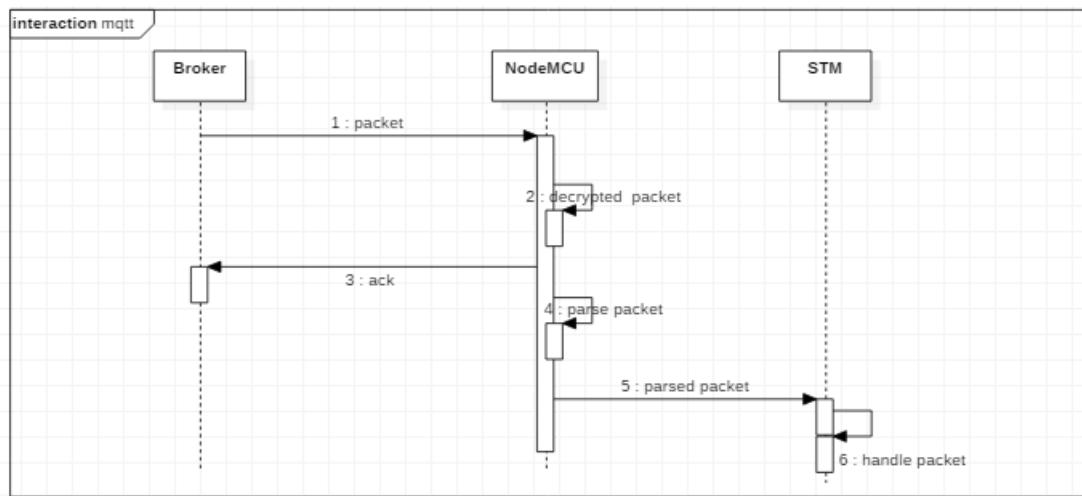
- Sequence diagram



Hình 42: Cấu hình wifi.



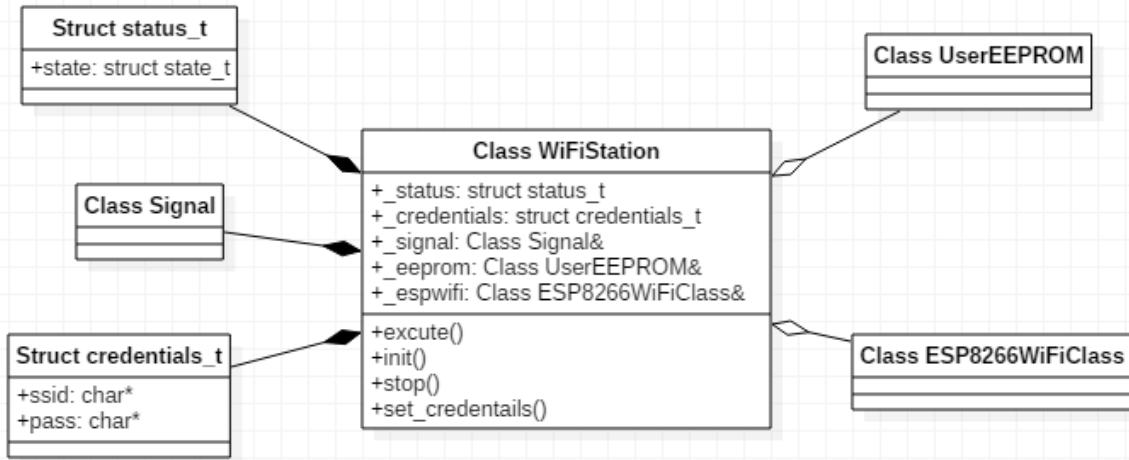
Hình 43: Hoạt động của web-server hoạt động trên board Node MCU.



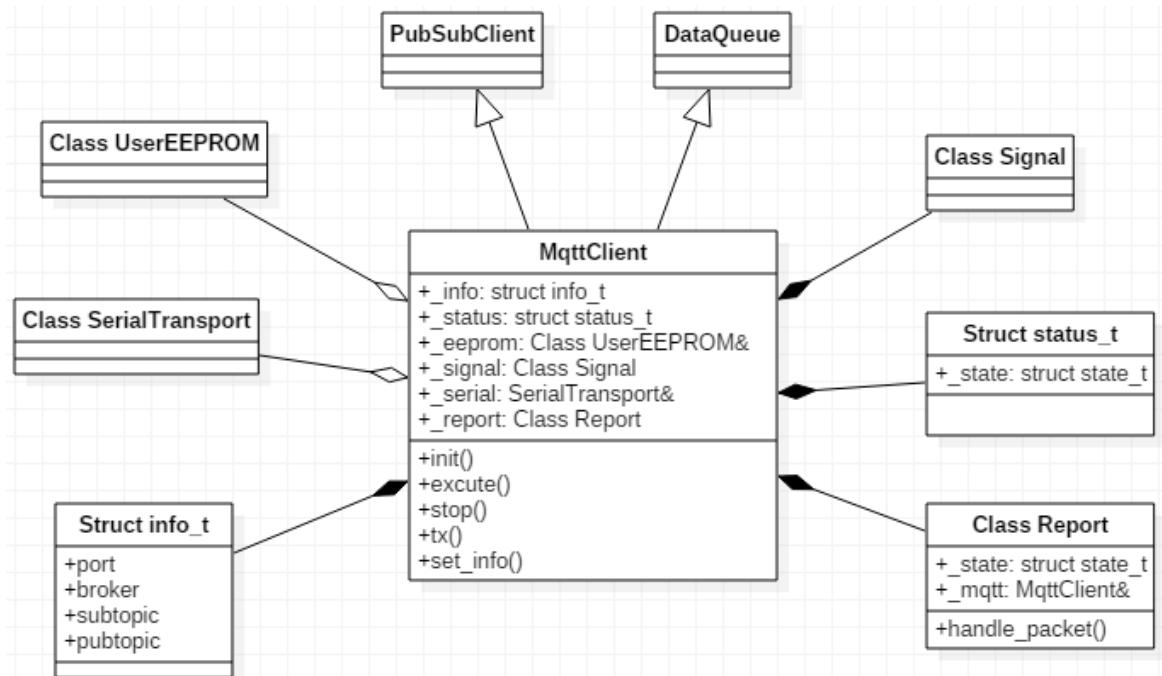
Hình 44: Truyền nhận dữ liệu.

- Class diagram

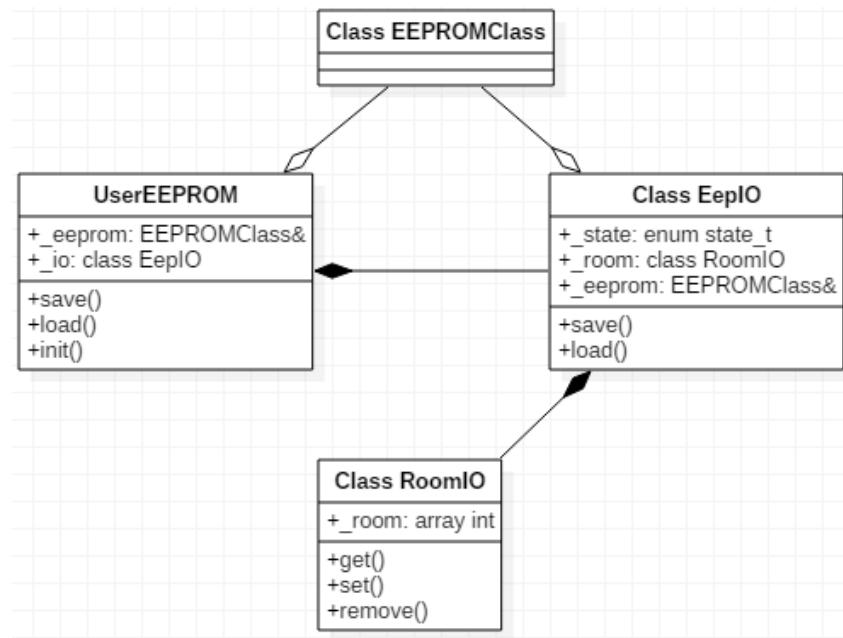
- Các class hoạt động trên board NodeMCU:



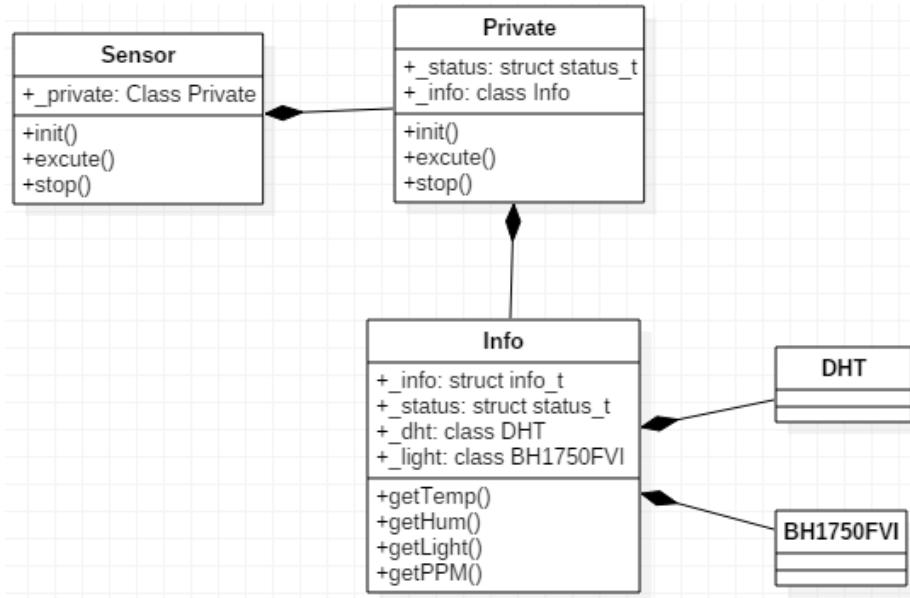
Hình 45: WiFiStation.



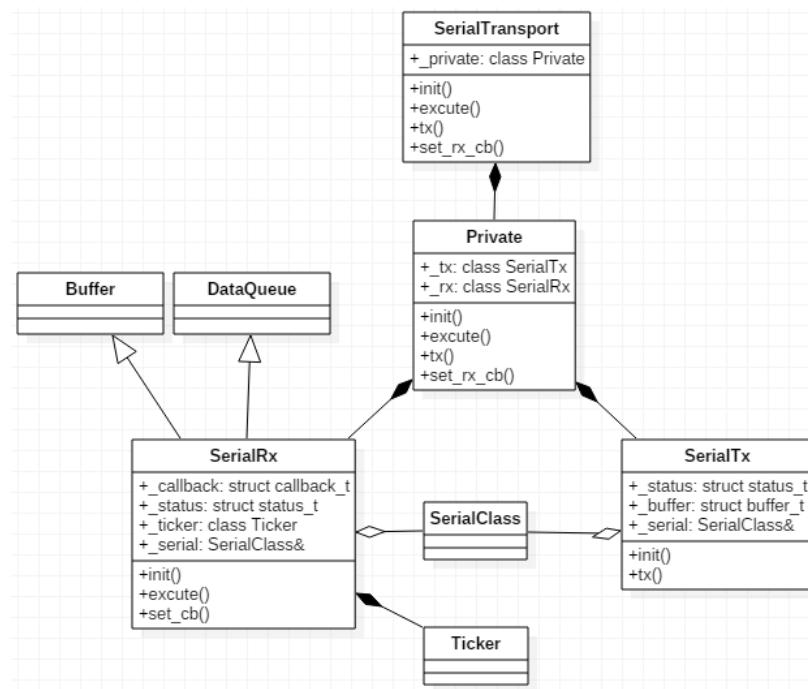
Hình 46: MqttClient.



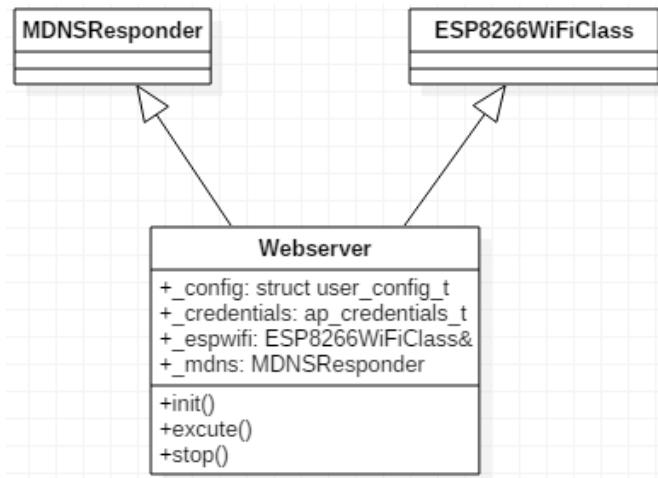
Hình 47: UserEEPROM.



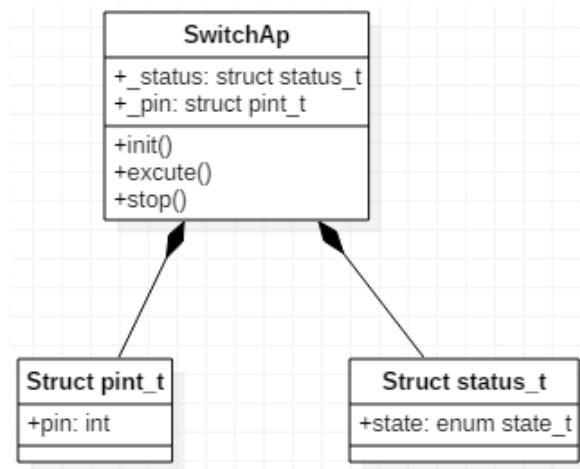
Hình 48: Sensor.



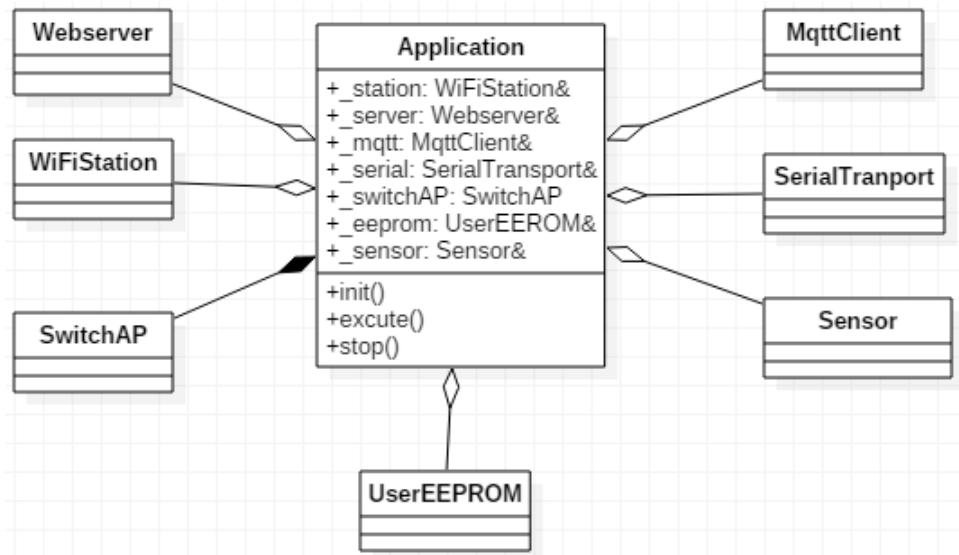
Hình 49: SerialTransport.



Hình 50: Webserver.

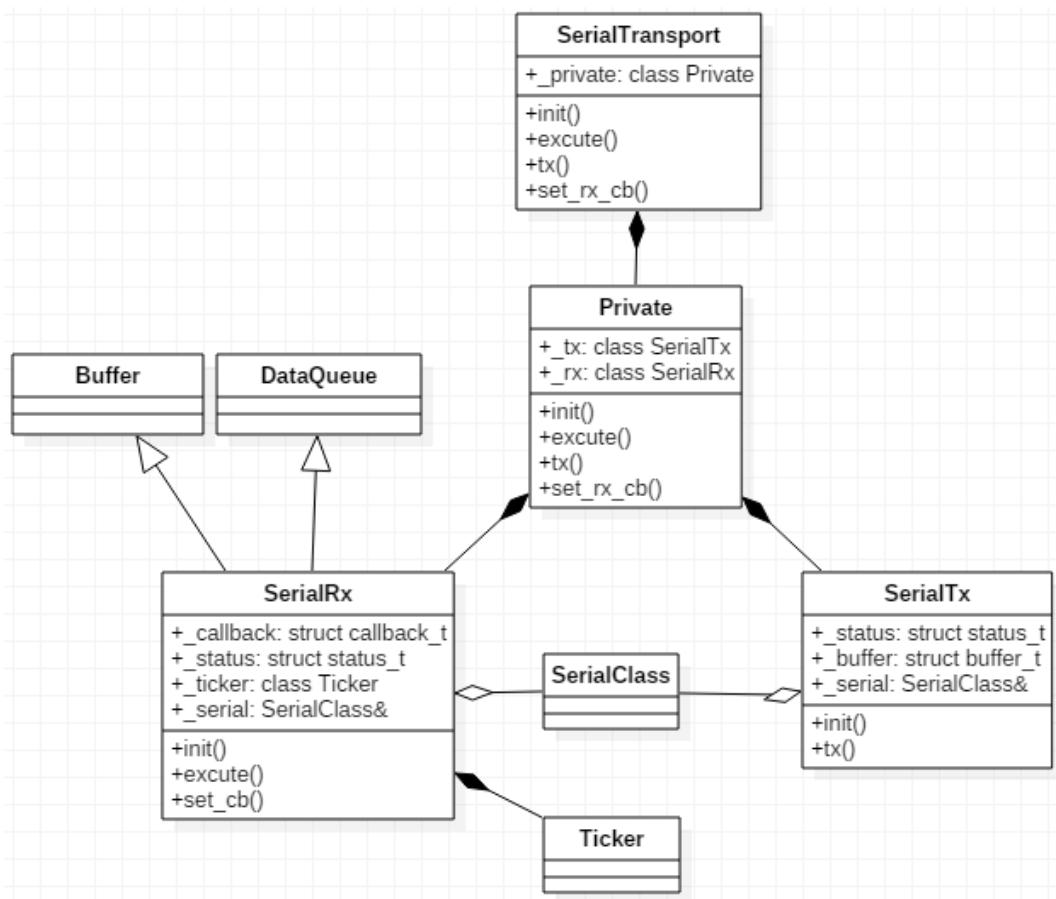


Hình 51: SwitchAP.

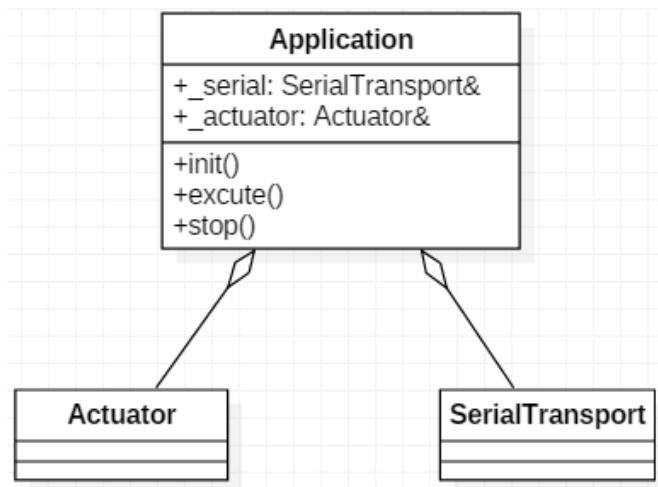


Hình 52: Application.

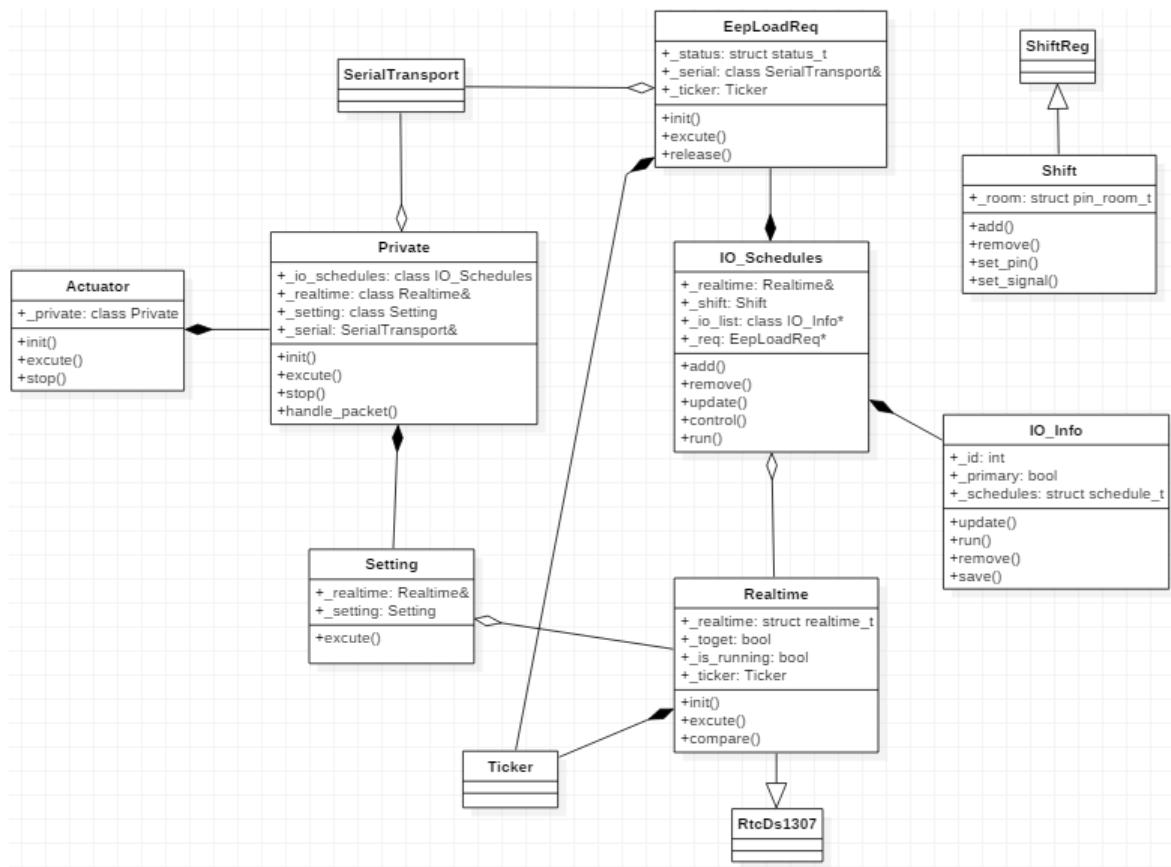
- Các class hoạt động trên board STM:



Hình 53: SerialTransport.



Hình 54: Application.

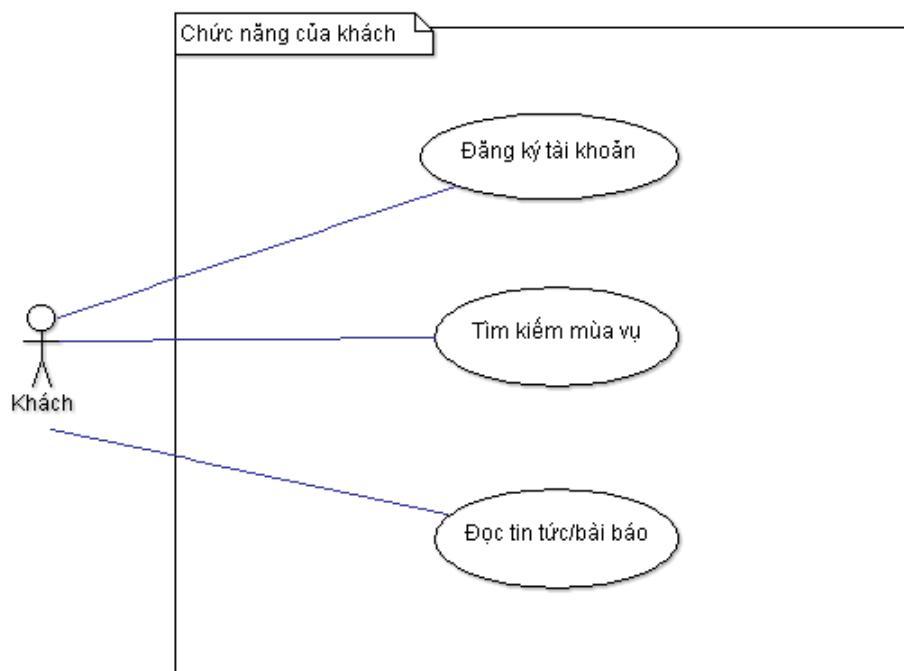


Hình 55: Actuator.

3.3 Ứng dụng web

3.3.1 Sơ đồ use case

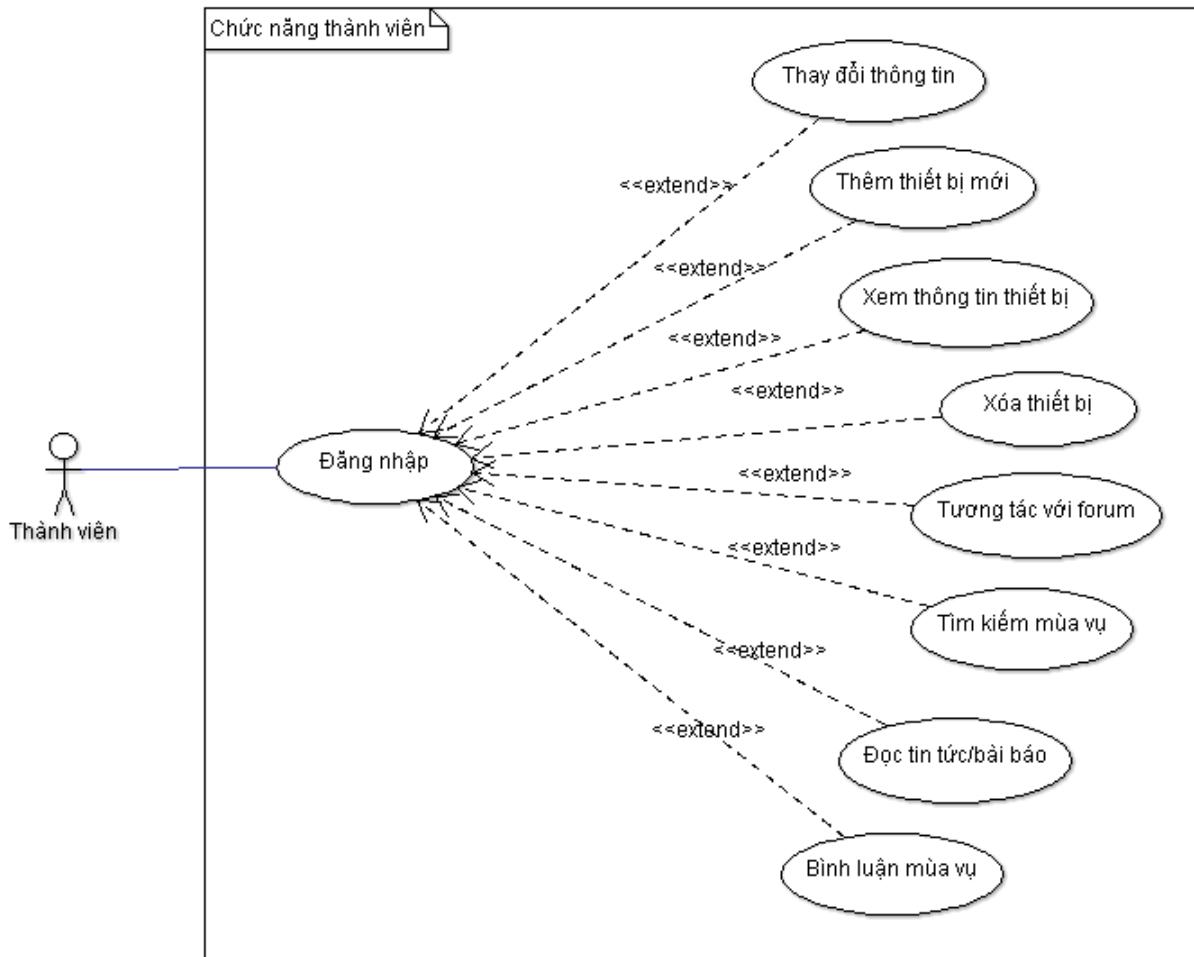
Các chức năng của tài khoản khi chưa thực hiện đăng nhập tài khoản.



Hình 56: Các chức năng của khách.

Chi tiết các use case và các mô tả use case các chức năng của khách được trình bày trong phần phụ lục A cuối luận văn.

Các chức năng của tài khoản thành viên sau khi đăng nhập.

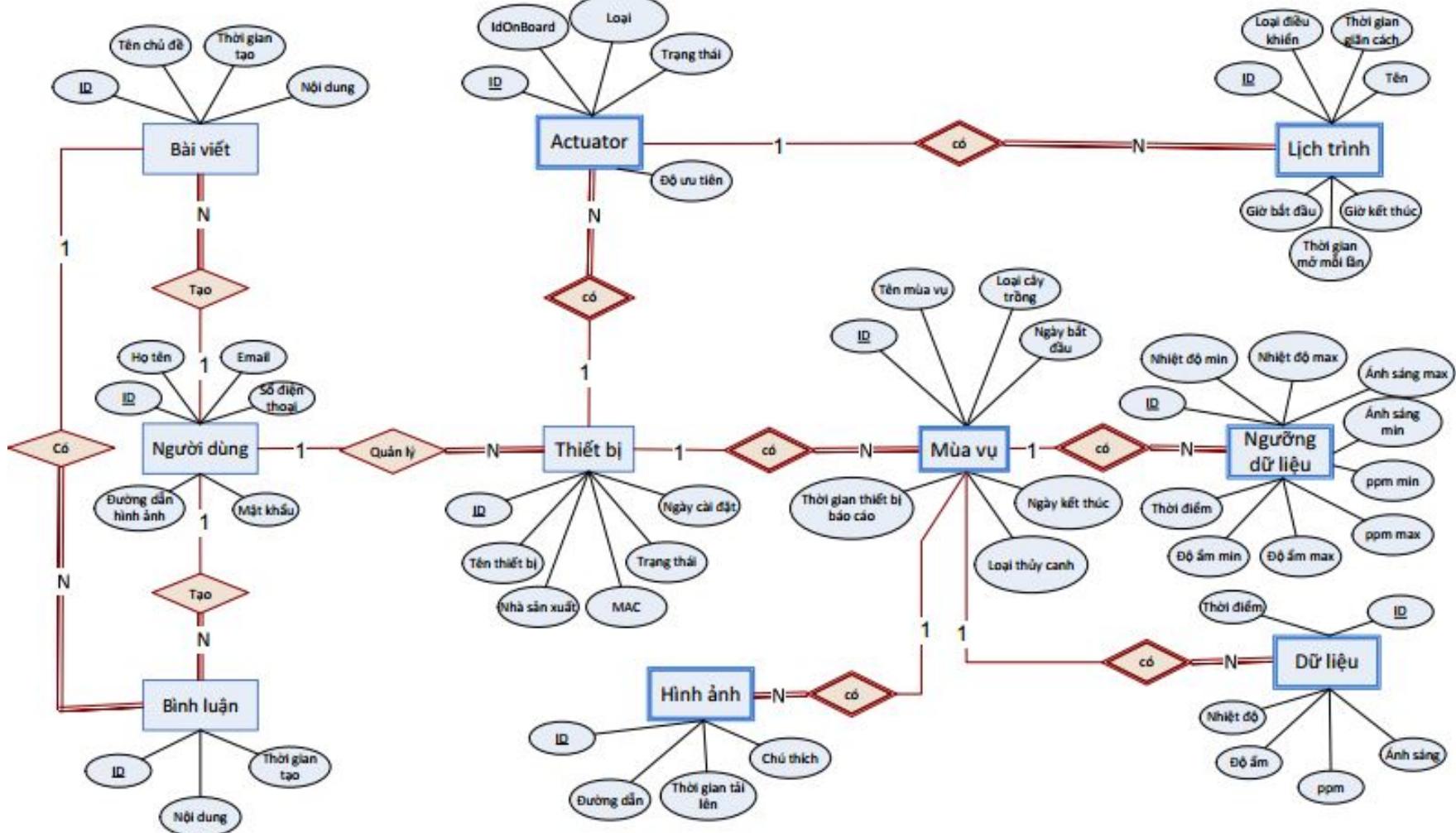


Hình 57: Các chức năng của thành viên.

Chi tiết các use case và các mô tả use case các chức năng của thành viên được trình bày trong phần phụ lục A cuối luận văn.

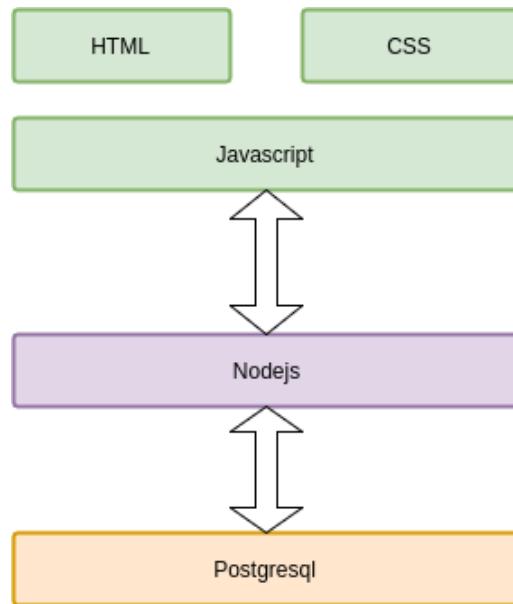
3.3.2 Thiết kế database

55



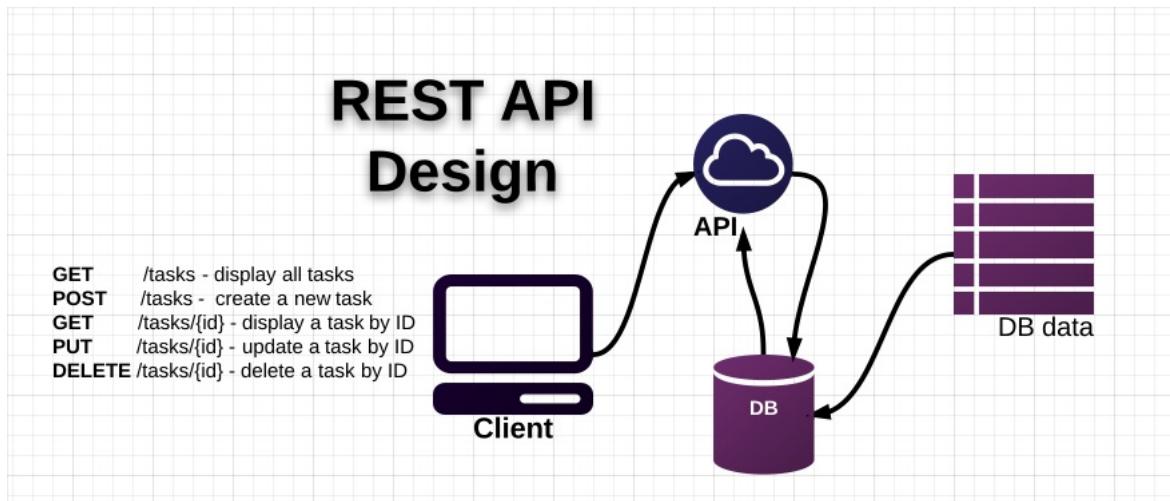
Hình 58: Thiết kế database.

3.3.3 Thiết kế API



Hình 59: Mô hình cụm server.

- Server được nhóm xây dựng trên nền tảng Nodejs sử dụng framework Expressjs.
- Cơ sở dữ liệu bên dưới để lưu trữ toàn bộ dữ liệu của hệ thống là Postgresql.
- Server tương tác với cơ sở dữ liệu Postgresql nhờ thư viện Sequelize hỗ trợ, giúp cho việc truy xuất dữ liệu một cách dễ dàng, không cần phải viết ra câu lệnh SQL tinh minh.
- Phía client giao diện được xây dựng trên ngôn ngữ HTML, CSS và JavaScript, sử dụng framework Angularjs.
- Các API được thiết kế theo chuẩn RESTful với các phương thức được sử dụng bao gồm: GET, POST, PUT và DELETE.



Hình 60: Mô hình RESTful API [37].

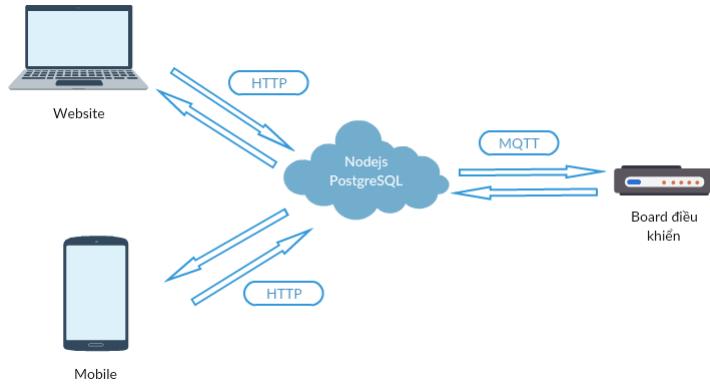
3.4 Ứng dụng di động

Ứng dụng được viết trên framework Ionic 2, sử dụng Angular 2 và có thể chạy trên Android hoặc iOS (hiện tại nhóm mới thử nghiệm trên Android).

- Ứng dụng nhằm mục đích giúp người dùng có thể theo dõi trạng thái của hệ thống một cách nhanh chóng.
- Chức năng chính:
 - Thêm một board mạch mới bằng cách quét mã QR.
 - Xem thông tin tình trạng hệ thống hiện tại (nhiệt độ, độ ẩm, cường độ ánh sáng, nồng độ dinh dưỡng).
 - Xem số lượng và trạng thái các relay trên board.
 - Có thể tắt, mở các relay trên board.
- Cách thức hoạt động: truy vấn API của web service để lấy và gửi thông tin.

3.5 Giao tiếp giữa Board điều khiển và Web

3.5.1 Giao thức



Hình 61: Mô hình các giao thức trong hệ thống.

Sử dụng giao thức MQTT cho cả hai chiều tương tác là gửi dữ liệu từ thiết bị lên và từ server gửi xuống. Các gói tin đều được mã hóa XXTEA để đảm bảo tính bảo mật.

3.5.2 Cấu trúc các gói tin

Các bảng sau mô tả cụ thể cấu trúc các gói tin:

	Số thứ tự	Cấu trúc gói tin			
Trường	SN	MAC	CMD_ID	DATA_LENGTH	DATA
Kích thước (byte)	4	12	2	4	>0
Mô tả	số thứ tự gói tin gửi và nhận	địa chỉ MAC	định danh gói tin	độ dài gói tin	dữ liệu

Bảng 2: Cấu trúc chung của các gói tin.

Chi tiết các gói tin:

Gửi từ	Server		
CMD_ID	01		
DATA_LENGTH	0034		
Trường dữ liệu	thời gian bắt đầu	thời gian kết thúc	thời gian gửi dữ liệu
Kích thước (byte)	14	14	6
Cấu trúc	yyyymmddhhmmss	yyyymmddhhmmss	hhmmss
Mô tả, ví dụ	yyyy: năm mm: tháng dd: ngày hh: giờ mm: phút ss: giây		

Bảng 3: Cấu hình.

Gửi từ	Server			
CMD_ID	02			
DATA_LENGTH	(2+2+6x4xSCH_NUM)xIO_NUM			
Trường dữ liệu	IO_NUM	IO_ID	SCH_NUM	lịch trình
Kích thước (byte)	2	2	2	6x4xSCH_NUM
Cấu trúc	XX	XX	XX	4x[hhmmss]xSCH_NUM
Mô tả, ví dụ	số lượng actuator trên board	ID actuator trên board	số lịch trình	hhmmss: 050000

Bảng 4: Lịch trình cho actuator.

Gửi từ	Server	
CMD_ID	03	
DATA_LENGTH	0003	
Trường dữ liệu	IO_ID	lệnh điều khiển
Kích thước (byte)	2	1
Cấu trúc	XX	X
Mô tả, ví dụ		giá trị: 0/1 0: tắt 1: bật

Bảng 5: Điều khiển actuator.

Gửi từ	Board điều khiển			
CMD_ID	04			
DATA_LENGTH	0012			
Trường dữ liệu	nhiệt độ	độ ẩm	ánh sáng	ppm
Kích thước (byte)	2	2	4	4
Cấu trúc	XX	XX	XXXX	XXXX

Bảng 6: Dữ liệu gửi từ các cảm biến.

Gửi từ	Board điều khiển
CMD_ID	05
DATA_LENGTH	0014
Trường dữ liệu	REAL_TIME
Kích thước (byte)	14
Cấu trúc	yyyymmddhhmmss

Bảng 7: Đồng bộ thời gian thực.

Gửi từ	Server		
CMD_ID	06		
DATA_LENGTH	0004		
Trường dữ liệu	IO_ID	lệnh điều khiển	mức ưu tiên
Kích thước (byte)	2	1	1
Cấu trúc	XX	X	X
Mô tả, ví dụ	11 - 19: bơm nước 21 - 29: đèn 31 - 39: quạt 41 - 49: bơm oxy	0: thêm actuator 1: xóa actuator	0: primary 1: secondary

Bảng 8: Thêm bớt actuator.

Gửi từ	Board điều khiển
CMD_ID	07
DATA_LENGTH	0001
Trường dữ liệu	HANDLED
Kích thước (byte)	1
Cấu trúc	X
Mô tả, ví dụ	0: gói tin sai, không nhận được 1: xử lý thành công gói tin

Bảng 9: Gói tin phản hồi.

Gửi từ	Server
CMD_ID	08
DATA_LENGTH	0001
Trường dữ liệu	ACTIVE
Kích thước (byte)	1
Cấu trúc	X
Mô tả, ví dụ	0: thêm board mới 1: xóa board

Bảng 10: Thêm, bớt một board điều khiển.

4 HIỆN THỰC HỆ THỐNG

4.1 Hệ thống Board điều khiển

4.1.1 Môi trường phát triển

Các công cụ và môi trường phát triển được sử dụng để thiết kế và hiện thực board điều khiển:

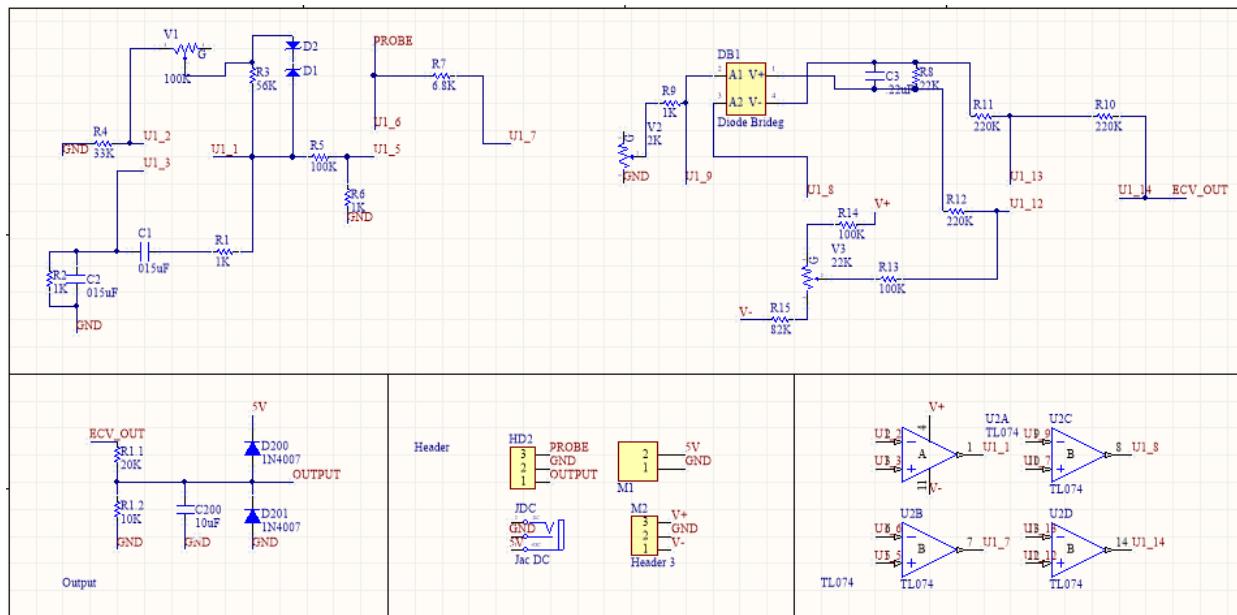
- Windows 10, Linux (Ubuntu 16)
- Công cụ vẽ và thiết kế mạch: Altium
- IDE: Qt5, Visual studio code, Sublime Text

4.1.2 Mạch chức năng đo ppm

1. Cơ sở nghiên cứu

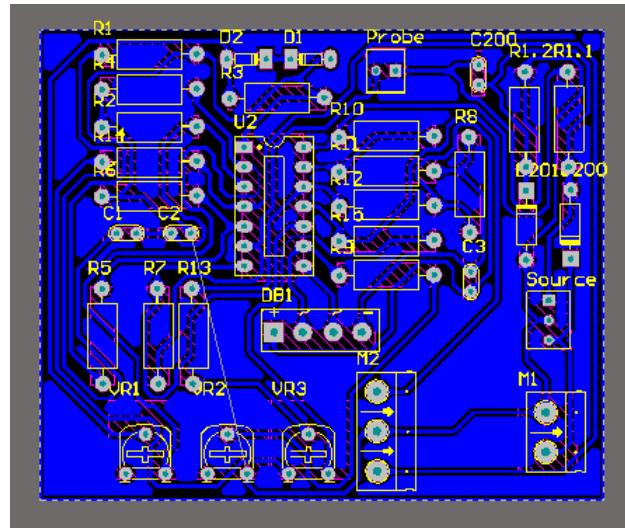
- Việc hiện thực module này tham khảo các bài viết tại [33] [34] [35]
- Bài viết khá chi tiết về lý thuyết và các bước hiện thực toàn bộ module.
- Các phần lý thuyết và giải thích bên dưới được tổng hợp và dịch thuật từ các bài viết này.

2. Schematic

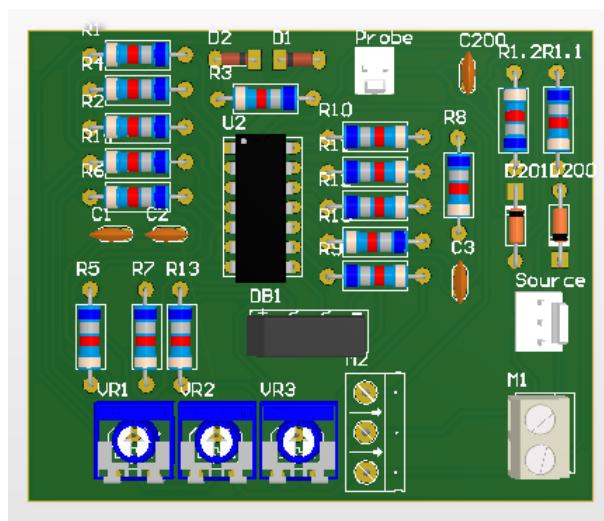


Hình 62: PPM schematic.

3. PCB



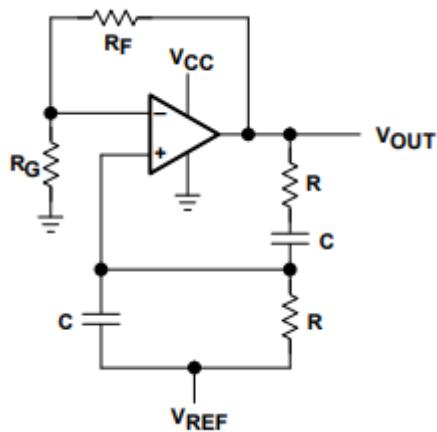
Hình 63: PPM PCB mô hình 2D.



Hình 64: PPM PCB mô hình 3D.

Thành phần chính trong mạch:

- Oscillator cầu Wein: là mạch điện tử tạo ra tín hiệu điện tử dao động, thường là một sóng sin hay sóng vuông. Bộ tạo dao động chuyển đổi dòng điện một chiều (DC) từ nguồn điện sang tín hiệu dòng xoay chiều (AC). Ưu điểm của mạch là chỉ có một vài thành phần và sự ổn định tần số tốt. Phản nhược điểm của mạch là biên độ đầu ra biến dạng cao gây khó khăn trong việc thu thập. Có một vài cách để giảm thiểu tác động này.



Hình 65: Oscillator cầu Wein [34].

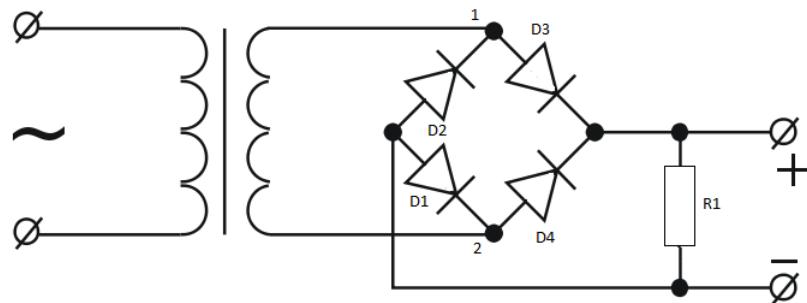
Các giá trị được cung cấp để tạo ra mạch dao động có tần số là 10 kHz.

Ta có:

$$R = 1 \text{ k}\Omega, C = 0.015 \mu\text{F}, \text{suy ra } F = \frac{1}{2\pi RC}$$

Chúng được sử dụng rộng rãi trong nhiều thiết bị điện tử. Ví dụ phổ biến của tín hiệu được tạo ra bởi dao động bao gồm các tín hiệu phát sóng của đài phát thanh và truyền hình, tín hiệu đồng hồ mà điều chỉnh máy tính và đồng hồ thạch anh, âm thanh được tạo ra bởi beepers điện tử và trò chơi điện tử... Ta sử dụng một tín hiệu AC để đo độ dãy của muối, có thể thay đổi giá trị R, C trong oscillator để đảm bảo rằng tần số đầu ra là trên 1 kHz, tần số thấp hơn sẽ cung cấp các giá trị không ổn định.

- Gain loop: mạch khuếch đại nguồn tín hiệu nhận được từ đầu dò 1cm đặt trong dung dịch
- Bộ chuyển đổi AC thành DC: Một mạch chỉnh lưu, một mạch điện bao gồm các linh kiện điện - điện tử dùng để biến đổi dòng điện xoay chiều thành dòng điện một chiều. Mạch chỉnh lưu cả chu kỳ thường dùng 4 Diode mắc theo hình cầu (còn gọi là mạch chỉnh lưu cầu). Mạch chỉnh lưu có thể được sử dụng trong các bộ nguồn cung cấp dòng điện một chiều, hoặc trong các mạch tách sóng tín hiệu vô tuyến điện trong các thiết bị vô tuyến, sử dụng mạch chỉnh lưu nhiều diốt (4 diốt) để có thể biến đổi từ xoay chiều thành một chiều.



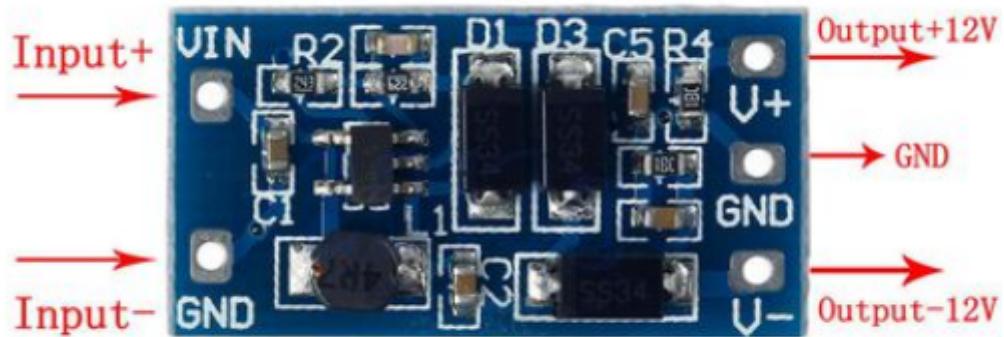
Hình 66: Cầu diode [35].

Nửa chu kì đầu, khi điểm 1 dương so với điểm 2, dòng chạy từ 1 qua D3 qua tải R1 qua D1 về đầu âm. Nửa chu kì sau, khi điểm 2 dương so với điểm 1,

dòng chạy từ 2 qua D4 qua tải R1 qua D2 về đầu âm. Cả chu kì đều có dòng điện chạy qua tải.

4. Các linh kiện điện tử

- Nguồn đôi 12V: Mạch tạo nguồn đôi 12V có chức năng đổi từ nguồn đơn DC sang nguồn đôi +12V. Với điện áp ngõ vào từ 2.8V - 5VDC.



Hình 67: Nguồn đôi 12V - Trước [32].



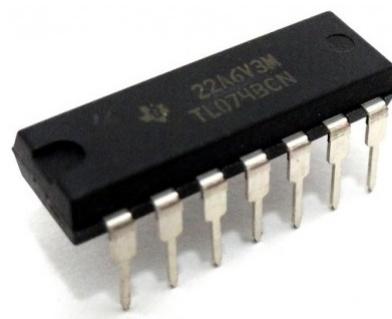
Hình 68: Nguồn đôi 12V - Sau [32].

- Jack DC: kết nối với nguồn DC để cấp nguồn cho mạch

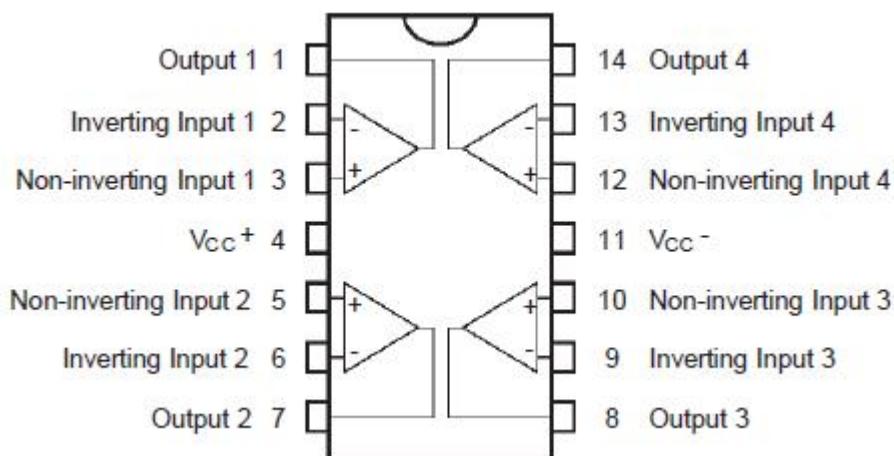


Hình 69: Jack DC [32].

- IC TL074: gồm 4 opamp trong 1 IC, dùng chung 1 nguồn đôi.



Hình 70: IC TL074 [32].



Hình 71: TL074 PIN [36].

- Diode Zener: diode ổn áp, nó được chế tạo sao cho khi phân cực ngược thì diode Zener sẽ ghim một mức điện áp gần cố định bằng giá trị ghi trên diode, làm ổn áp cho mạch điện.



Hình 72: Diode Zener [32].

- Tụ điện:



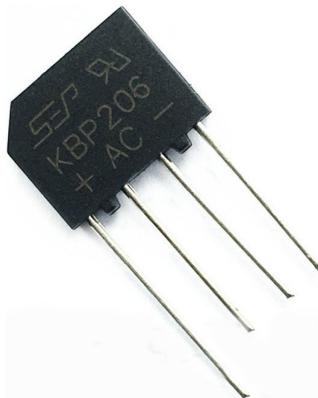
Hình 73: Tụ $0.015 \mu\text{F}$ [32].



Hình 74: Tụ $0.22 \mu\text{F}$ [32].

- Cầu Diode: gồm 4 diode đơn mắc thành cầu chỉnh lưu được đóng gói trong một vỏ duy nhất, dùng để chỉnh lưu dòng điện xoay chiều thành một chiều.

- Điện áp tối đa: 600V.
- Dòng điện định mức: 4A.
- Nhiệt độ hoạt động: -55°C đến 150°C.



Hình 75: Cầu Diode [32].

4.1.3 Hệ thống board điều khiển

1. Nhiệt độ, độ ẩm

- Nối dây

DHT	Node MCU
VCC	3.3V
OUT	D3
GND	GND

Bảng 11: Nối chân DHT-NodeMCU

- Thư viện

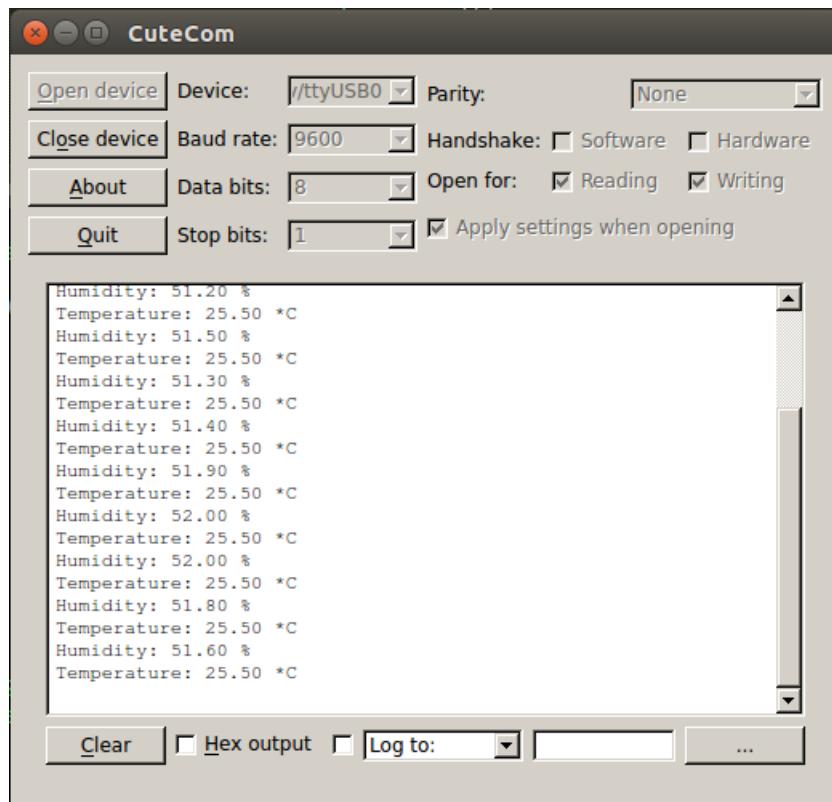
DHT.cpp	Add Class Sensor into Esp8266 Folder
DHT.h	Add Class Sensor into Esp8266 Folder
DHT_U.cpp	Add Class Sensor into Esp8266 Folder
DHT_U.h	Add Class Sensor into Esp8266 Folder

Hình 76: Thư viện DHT.

hydroponic / esp8266 / libraries / DHT /

Hình 77: Đường dẫn thư viện DHT.

- Kết quả đo



Hình 78: Kết quả đo nhiệt độ, độ ẩm DHT.

2. Ánh sáng

- Nối dây

BH1750	Node MCU
VCC	3.3V
GND	GND
SCL	D1
SDA	D2
ADDR	D0

Bảng 12: Nối chân BH1750 – Node MCU

- Thư viện

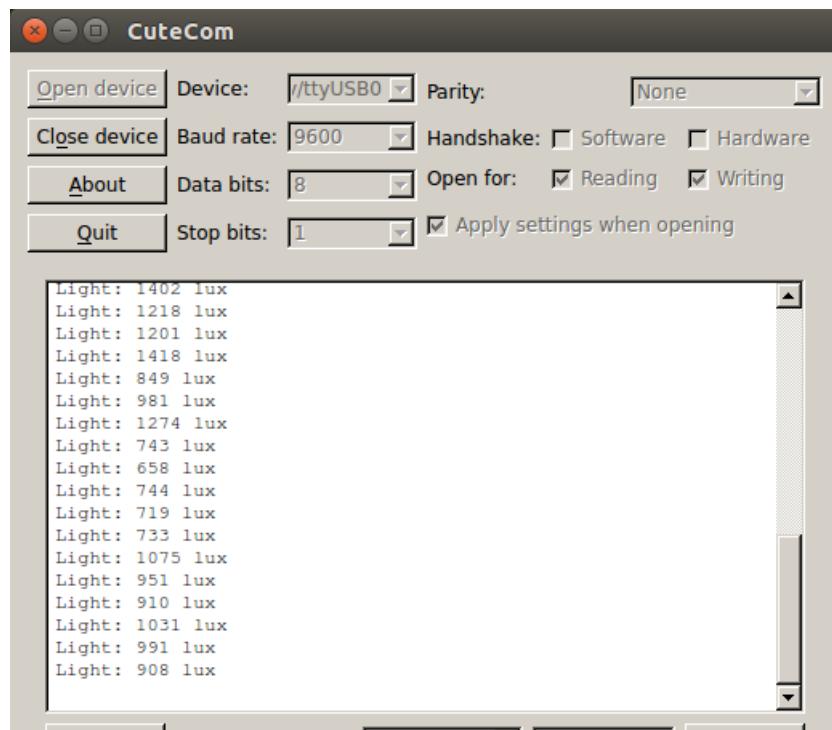


Hình 79: Thư viện BH1750 .

hydroponic / esp8266 / libraries / BH1750FVI /

Hình 80: Đường dẫn thư viện BH1750.

- Kết quả đo



Hình 81: Kết quả đo cường độ ánh sáng BH1750.

3. PPM

- Nối dây

4. Thời gian thực

- Nối dây

PPM	Node MCU
Analog Out	A0

Bảng 13: Nối chân PPM – Node MCU

DS1307	SMT32F103C8T6
GND	GND
VCC	5V
SDA	PB_9
SCL	PB_9

Bảng 14: Nối chân DS1307 – STM32F103C8T6

- Thư viện

DS1307.cpp	Update Actuator
DS1307.h	Update Actuator
DateTime.cpp	Update Actuator
DateTime.h	Update Actuator

Hình 82: Thư viện thời gian thực.

hydroponic / STM32F103C8T6_Qt / Libraries / DS1307 /

Hình 83: Đường dẫn thư viện thời gian thực.

5. Điều khiển các actuator

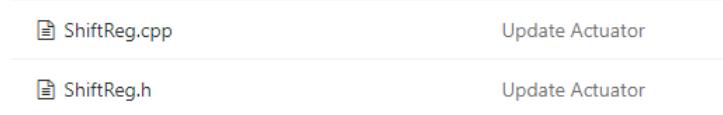
(a) Thanh ghi dịch

- Nối dây

74HC595	STM32F103C8T6
DATA	PA_7
LATCH	PA_6
CLOCK	PA_5

Bảng 15: Nối chân 74HC595 – STM32F103C8T6

- Thư viện



Hình 84: Thư viện thanh ghi dịch.

[hydroponic / STM32F103C8T6_Qt / Libraries / ShiftReg /](#)

Hình 85: Đường dẫn thư viện thanh ghi dịch.

(b) Nguồn cho các actuator

Các actuators sử dụng nguồn 12 VDC lấy từ adapter chuyển đổi AC 220V sang DC 12V.



Hình 86: Nguồn cho các actuator.

6. Giao tiếp với web-server

(a) Mã hóa dữ liệu

- Các gói tin trao đổi (truyền-nhận) gián tiếp với web-server (qua MQTT) sẽ được mã hóa sử dụng thư viện:



Hình 87: Thư viện mã hóa xxtea.

- Đường dẫn:

hydroponic / esp8266 / ScanWebserver / src / xxtea /

Hình 88: Đường dẫn thư viện xxtea.

(b) Giao thức tầng dưới – thư viện MQTT

- Như đã đề cập giao thức truyền nhận giữa web-server là MQTT, cả thiết bị lẫn web-server đều đóng vai trò là một MQTT Client
 - Thiết bị gửi tới broker (publish) các gói tin vào các kênh (channel) đã được cấu hình sẵn và thông nhất giữa thiết bị - webserver.
 - Thiết bị nhận các gói tin từ broker qua các kênh đã đăng ký (subscribe), khi webserver gửi gói tin vào các kênh đó.
- Thư viện

PubSubClient.cpp	Nothing
PubSubClient.h	Nothing

Hình 89: Thư viện MQTT.

hydroponic / esp8266 / ScanWebserver / src / PubSubClient /

Hình 90: Đường dẫn thư viện MQTT.

(c) Giao thức truyền nhận lớp trên

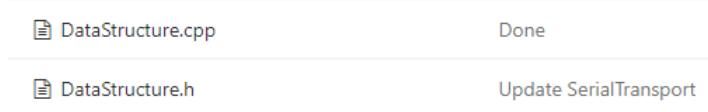
- Gói tin trên đường truyền có định dạng là (string), ví dụ:

STT	Gói tin thật	Gói tin khi gửi (nhận)
1	[1][2][3]	"123"
2	[0xA][0xB]	"0A0B"
3	"MXYZ"	không hợp lệ

Bảng 16: Các trường hợp giá trị dữ liệu.

Trong dự án này toàn bộ dữ liệu có giá trị (value) nằm trong 3 trường hợp

- i. $0 \leq \text{value} \leq 9$
 - ii. ' A ' $\leq \text{value} \leq 'F'$, dùng để gửi địa chỉ MAC.
 - iii. các trường hợp còn lại không hợp lệ
- Các cấu trúc dữ liệu và phương thức cho giao thức được định nghĩa trong:



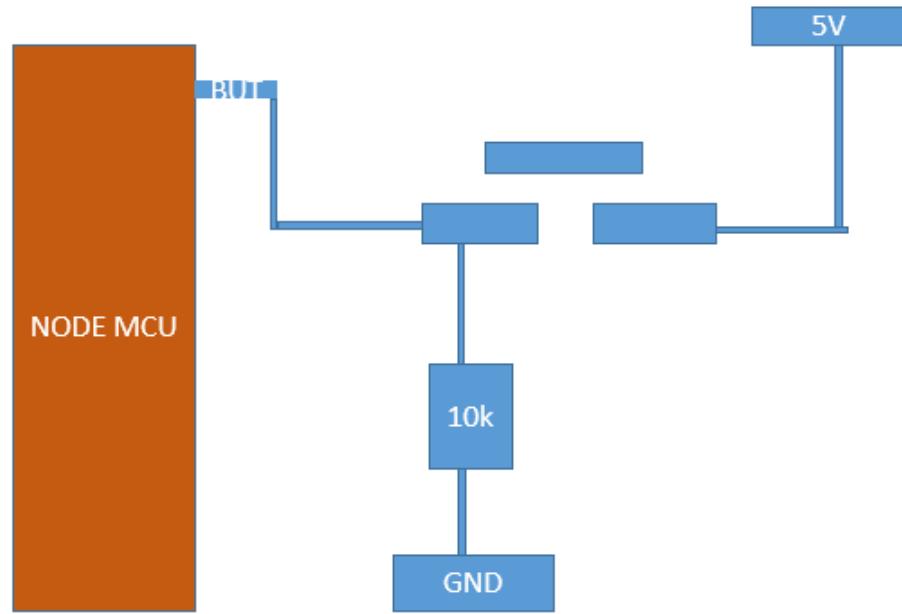
Hình 91: Thư viện cấu trúc dữ liệu.

hydroponic / esp8266 / ScanWebserver / src / DataStructure /

Hình 92: Đường dẫn thư viện cấu trúc dữ liệu.

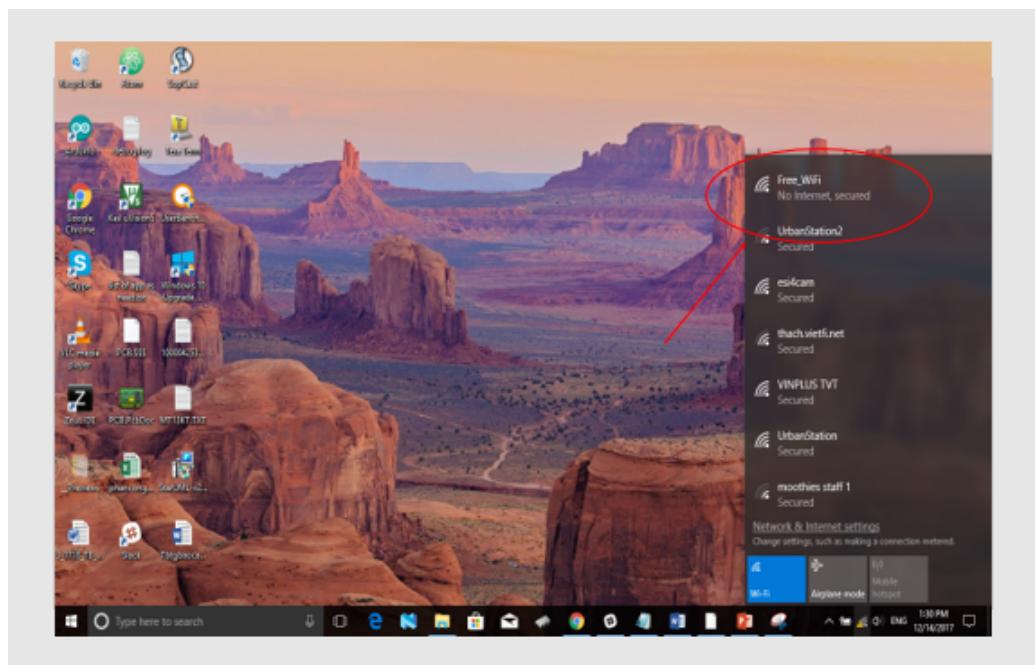
7. Web server

- Webserver hoạt động trên board NodeMCU, ở chế độ Access Point.
- Board chuyển từ chế độ Station sang Access Point khi nhận tín hiệu nhấn liên tục nút nhấn trong vòng 3s.



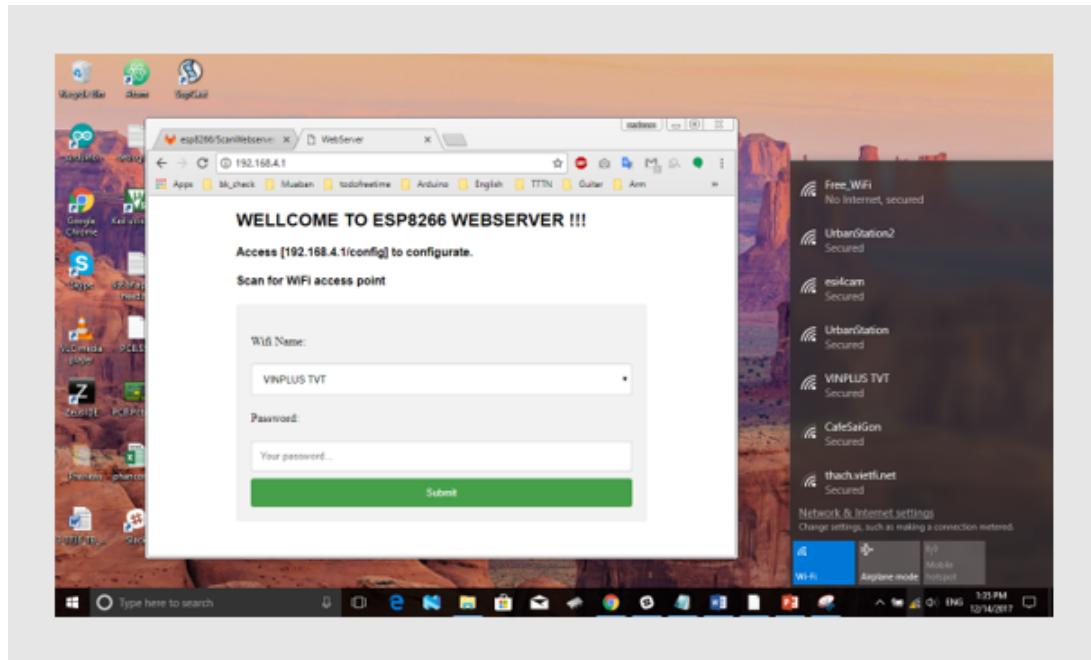
Hình 93: Nút nhấn.

- Giao diện và chức năng:
 - Khi chuyển sang chế độ Access Point và hoạt động chức năng webserver, Node MCU sẽ phát ra wifi.



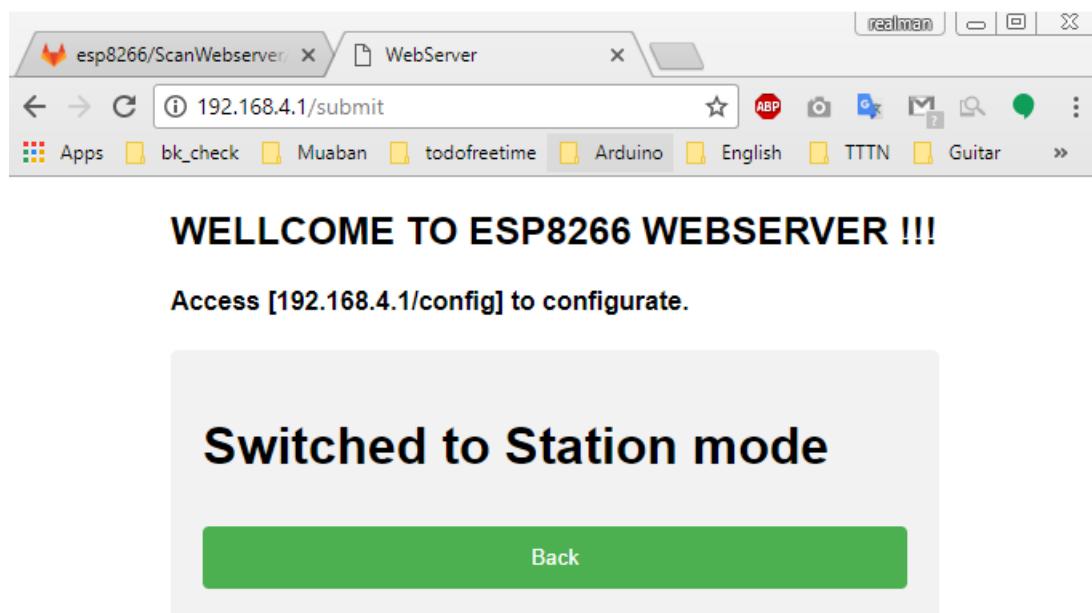
Hình 94: Node MCU phát ra wifi.

- Sau khi kết nối vào Access Point của Node MCU, mở trình duyệt, truy cập vào địa chỉ (192.168.4.1) ta sẽ kết nối với webserver.

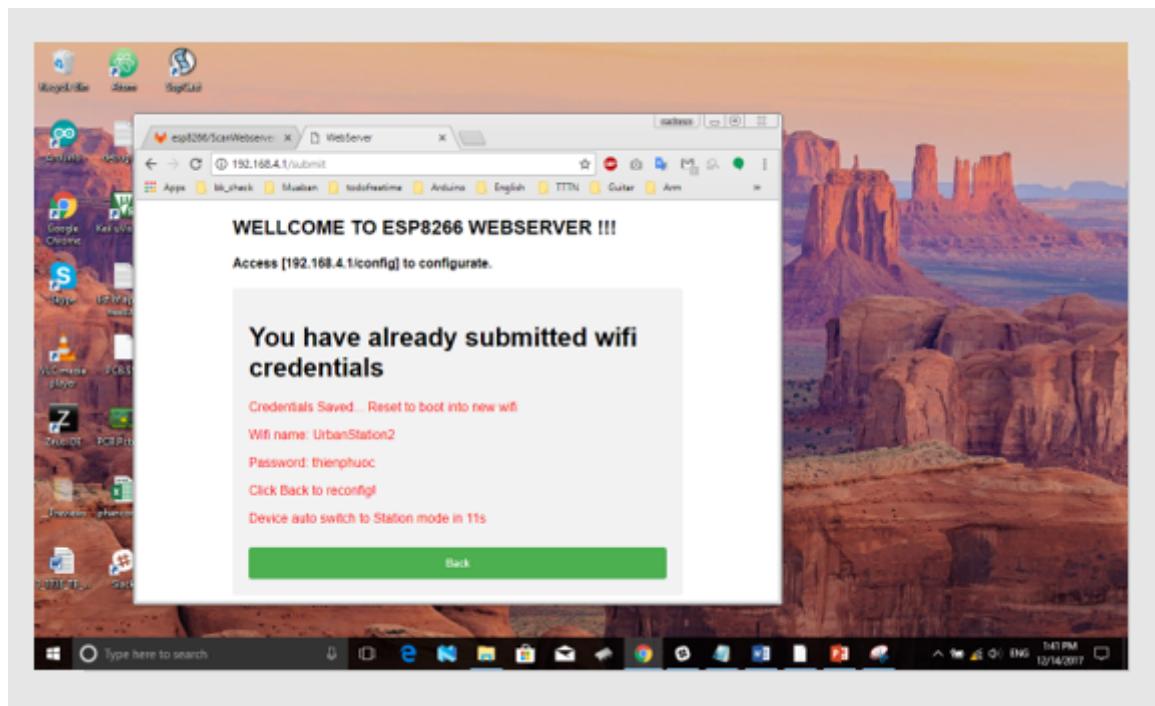


Hình 95: Truy cập vào địa chỉ để kết nối với webserver.

- Sau khi chọn (wifi name) và nhập (password), trình duyệt sẽ gửi 2 giá trị này về board và được lưu lại trong bộ nhớ eeprom.

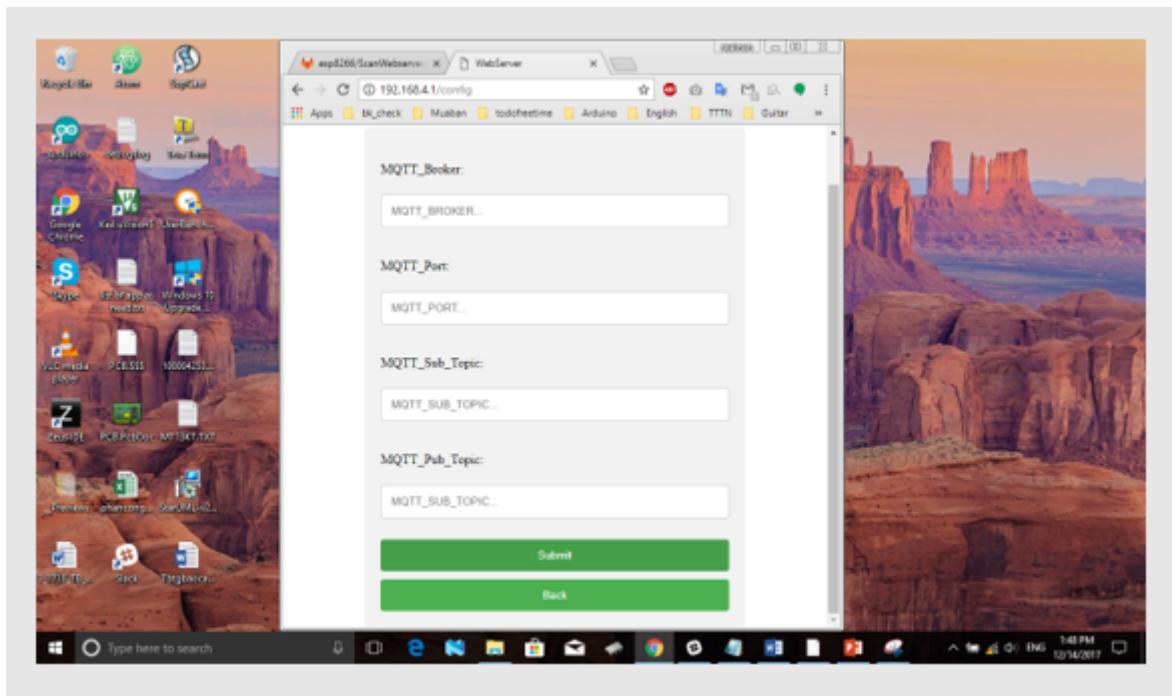


Hình 96: Thiết lập kết nối.



Hình 97: Thiết lập kết nối.

- Ngoài ra webserver còn cung cấp chức năng cấu hình các thông tin của giao thức MQTT, truy cập vào (192.168.4.1/config)



Hình 98: Cấu hình giao thức MQTT.

8. Đèn tín hiệu

- Đèn tín hiệu sử dụng led đơn, tiêu thụ dòng nhỏ.
- Hoạt động của đèn tín hiệu có ba trạng thái chính:
 - Tắt: 0V
 - Sáng: 3.3V.
 - Nháy: kết hợp timer.
- Các chức năng bao gồm:
 - Đèn báo tín hiệu wifi:
 - * Kết nối: đèn tắt.
 - * Mất kết nối: đèn nháy.
 - * Mất kết nối trong hơn 30s đèn sáng.
 - Đèn báo tín hiệu MQTT:
 - * Kết nối với broker : đèn tắt.
 - * Mất kết nối: đèn sáng.

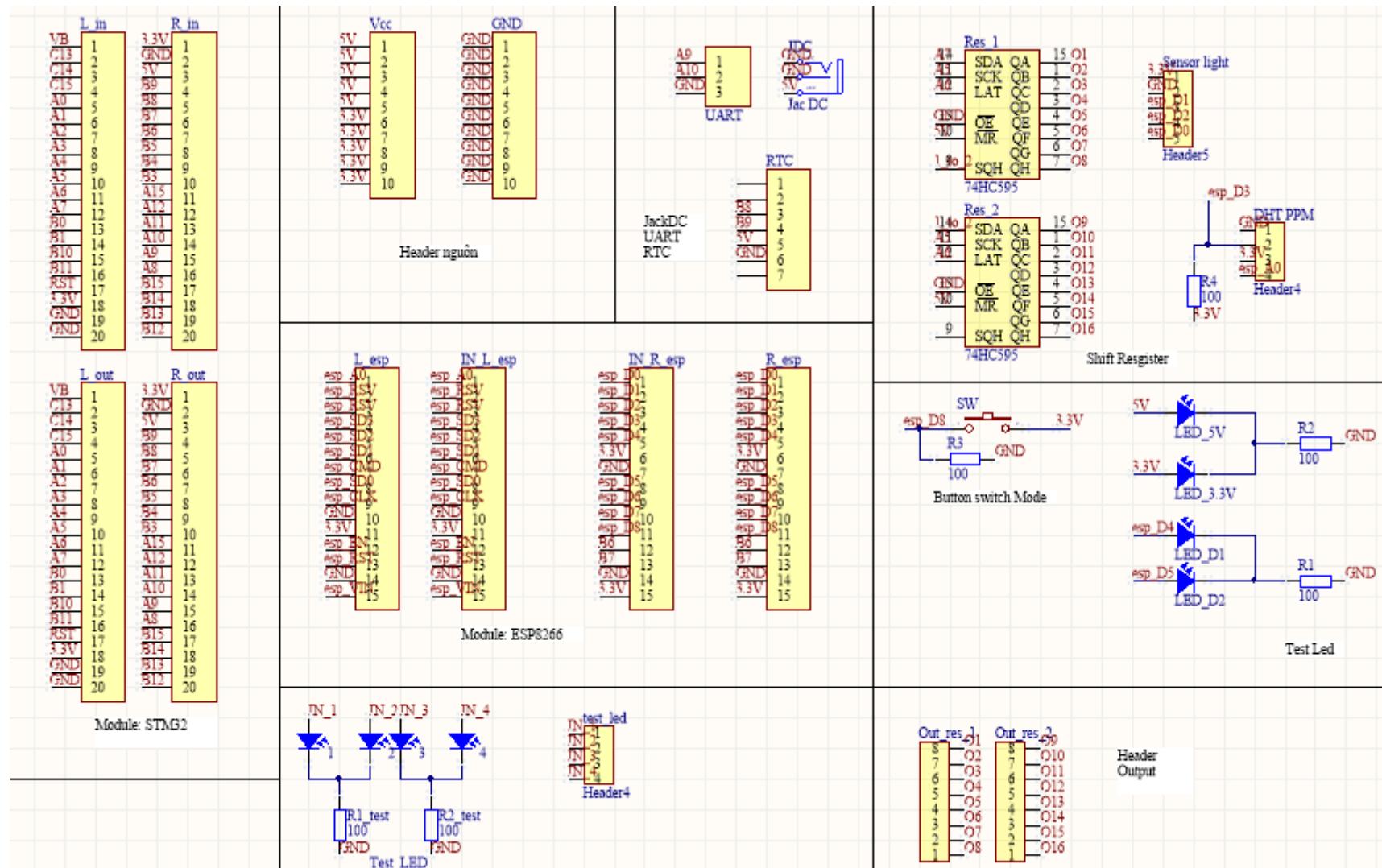
- Đèn báo tín hiệu của các actuator:
 - * Actuator chưa được kích hoạt: đèn tắt.
 - * Actuator được kích hoạt: đèn sáng.
 - * Actuator đang ở trong thời gian chạy thời gian biểu: đèn nháy.

9. Thiết kế mạch

- Schematic

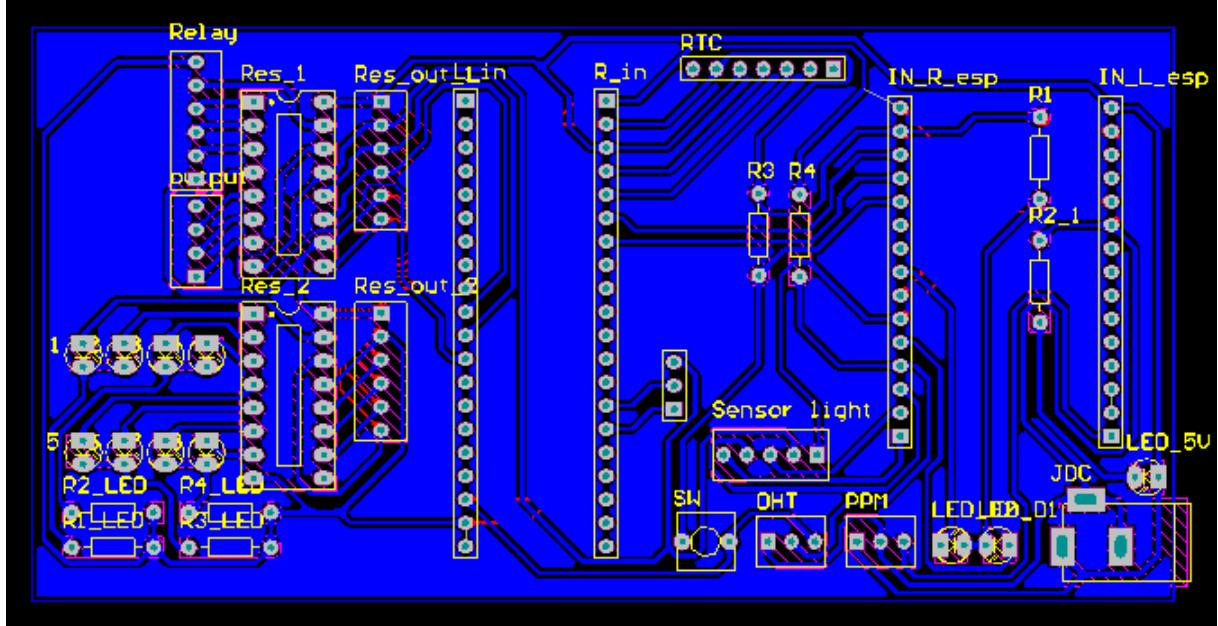
4.1 Hệ thống Board điều khiển

4 HIỆN THỰC HỆ THỐNG

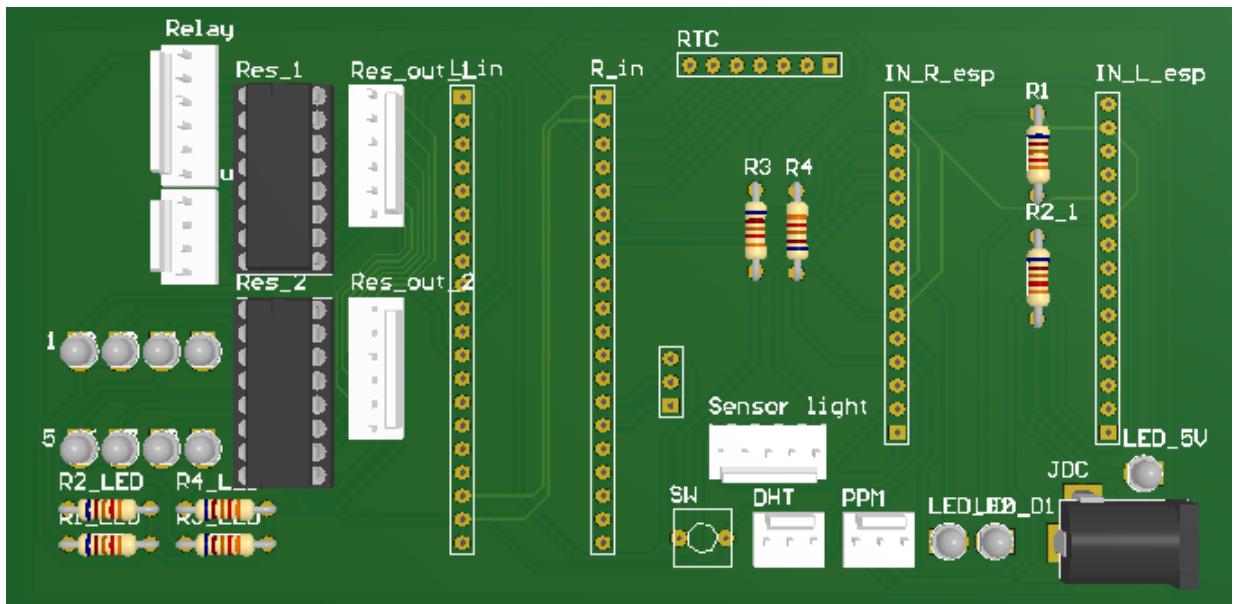


Hình 99: Schematic.

- PCB



Hình 100: PCB mô hình 2D



Hình 101: PCB mô hình 3D

10. Mô hình thử nghiệm



Hình 102: Mô hình thử nghiệm - 1.



Hình 103: Mô hình thử nghiệm - 2.



Hình 104: Mô hình thử nghiệm - 3.

4.2 Ứng dụng web

4.2.1 Môi trường phát triển

Các công cụ và môi trường phát triển được sử dụng để phát triển trang web:

- Hệ điều hành: Windows 10, Linux (Manjaro)
- IDE: Visual studio code
- Nodejs 6.x
- Angularjs 1.6.4
- Git 2.15.1
- Google chrome 63.0.3239.84 để kiểm thử

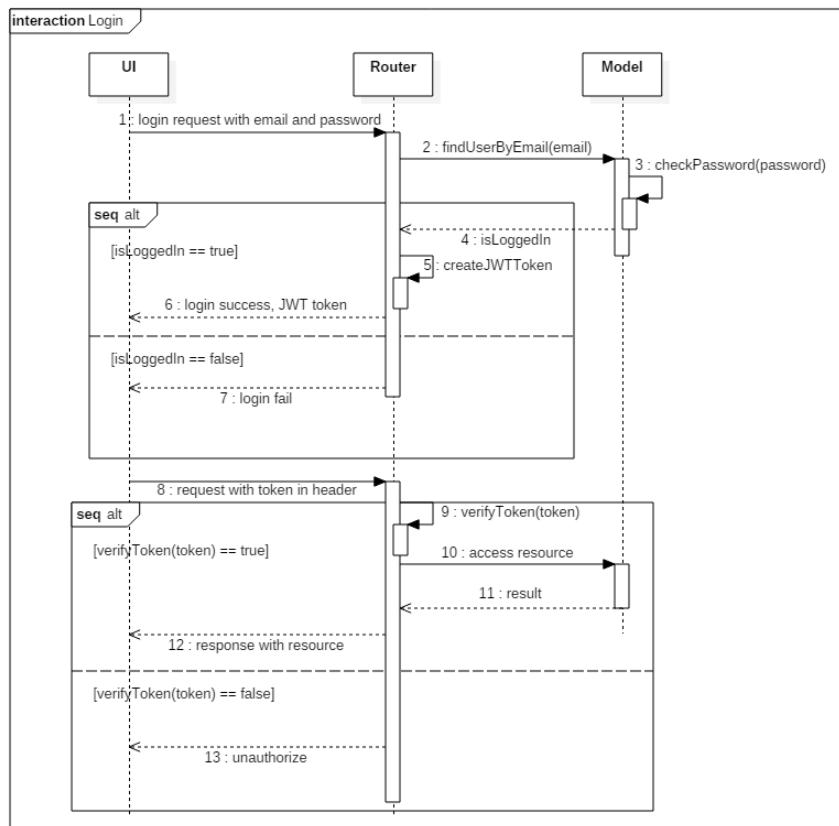
4.2.2 Các chức năng chính

- **Đăng ký**

Người dùng có thể đăng ký tài khoản thông qua trang Đăng ký (Sign up) để có thể bắt đầu sử dụng các chức năng trong hệ thống. Sau khi đăng ký thành công, hệ thống sẽ gửi một email cùng đường link xác nhận email có thật đến email người dùng vừa đăng ký. Người dùng phải nhấn vào đường link để kích hoạt tài khoản của mình. Tài khoản chưa kích hoạt sẽ không thể viết các bài viết chia sẻ trên trang Article.

- **Đăng nhập**

Quá trình đăng nhập sử dụng JWT được thể hiện ở sơ đồ tuần tự sau:



Hình 105: Thao tác đăng nhập vào hệ thống sử dụng JWT.

Sau khi người dùng đăng nhập thành công và hệ thống gửi trả về JWT, JWT sẽ được lưu trong cookie và được thêm vào header của mỗi request được gửi đi trong các lần sau. Khi người dùng đăng xuất khỏi hệ thống, JWT trong cookie sẽ được xóa đi.

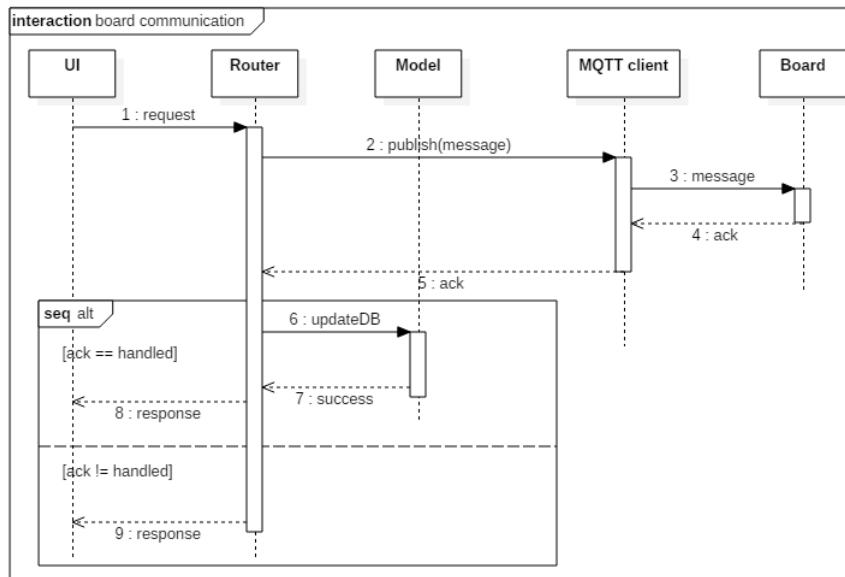
• Phân quyền người dùng

Hệ thống sử dụng module acl của Nodejs để phân ra 3 quyền người dùng: admin, mod và member.

- Member: người dùng bình thường, với các chức năng cơ bản như: quản lý các board điều khiển của mình, quản lý các mùa vụ, ...
- Mod: có tất cả các quyền của member, đồng thời có khả năng quản lý các bài viết (xóa, kiểm định) và các bình luận trong bài viết.
- Admin: có tất cả các quyền của mod, đồng thời có khả năng quản lý các người dùng trong hệ thống (xóa, nâng cấp quyền hạn).

- **Tương tác với board điều khiển**

Đây là chức năng quan trọng và được nhóm đặt làm trọng tâm. Quá trình tương tác với board điều khiển thông qua các gói tin (message) đã được nhóm thiết kế trong phần thiết kế hệ thống được thể hiện trong sơ đồ tuần tự sau:



Hình 106: Nhóm các thao tác tương tác với board điều khiển.

Các thao tác tương tác với board điều khiển bao gồm:

- Thêm, xóa board điều khiển.
- Thêm, xóa, sửa mùa vụ.
- Thêm, xóa, sửa actuator.
- Gửi lịch trình hoạt động.
- Nhận dữ liệu cảm biến từ board điều khiển.

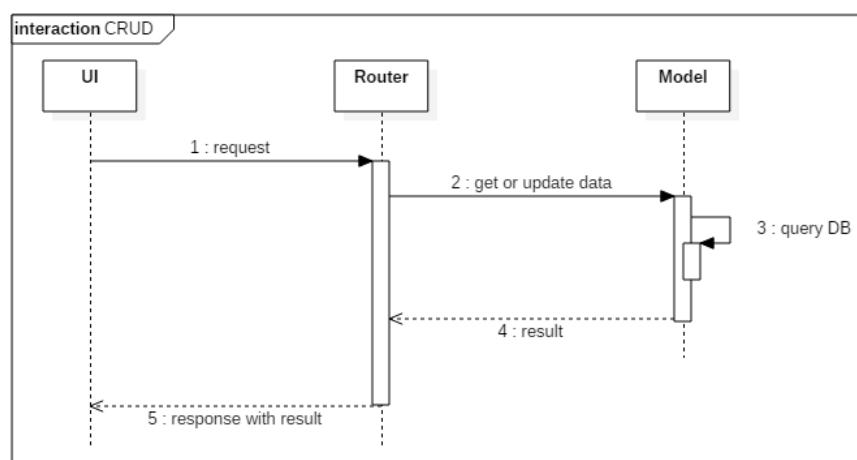
Thao tác của người sử dụng phía client được thực hiện ở thành phần UI (web/mobile), thao tác sẽ gửi yêu cầu đến server. Phía server nhận yêu cầu, nếu yêu cầu có tương tác gửi dữ liệu xuống phía board điều khiển thì server sẽ thực hiện gửi dữ liệu xuống board theo giao thức MQTT (server gửi dữ liệu xuống broker trung gian, board nhận dữ liệu từ broker). Board điều khiển nhận được dữ liệu, kiểm tra tính hợp lệ của dữ liệu và thực hiện thay đổi chức năng của board theo dữ liệu rồi

gửi gói tin ACK phản hồi về cho phía server qua giao thức MQTT. Phía server sau khi nhận gói ACK phản hồi thì mới thực hiện thao tác cập nhật thông tin trên cơ sở dữ liệu để đồng bộ, rồi gửi phản hồi về cho phía người dùng, kết thúc một yêu cầu. Mỗi thao tác tương tác với board điều khiển có một khoảng thời gian timeout nhất định (hiện tại thiết lập là 5 giây) để chờ gói tin ACK. Nếu sau khoảng thời gian timeout mà server không nhận được gói tin ACK từ board, server sẽ trả về cho web app là thao tác tương tác không thực hiện được và người dùng phải thực hiện lại.

Gói tin dữ liệu cảm biến gửi từ board điều khiển cũng đóng vai trò là tín hiệu xác định board điều khiển có còn hoạt động hay không. Ngay khi thêm board điều khiển mới vào hệ thống, board sẽ lập tức gửi gói tin dữ liệu cảm biến để thông báo board đã hoạt động. Nếu sau một khoảng thời gian timeout nhất định (hiện tại được thiết lập là 10 giây) mà server không nhận được gói tin dữ liệu cảm biến, server sẽ chuyển trạng thái của board từ “running” sang “no connection”, đồng thời gửi thông báo đến thiết bị di động của người dùng. Khi dữ liệu cảm biến vượt ngưỡng cho phép do người dùng thiết lập, server cũng gửi thông báo xuống thiết bị di động của người dùng.

- **Các chức năng CRUD khác**

Quá trình thực hiện các thao tác CRUD khác được thể hiện chung qua sơ đồ tuần tự sau:



Hình 107: Nhóm các thao tác CRUD.

Trên đây trình bày tóm tắt các chức năng đã hiện thực trong hệ thống, cụ thể và chi tiết hơn sẽ được mô tả cho từng API trong phần hiện thực server API tiếp theo.

4.2.3 Hiện thực server API

Các API được thiết kế theo chuẩn RESTful với dữ liệu được trao đổi theo kiểu JSON.

Mẫu dữ liệu trả về cho tất cả các API:

- Không có data:

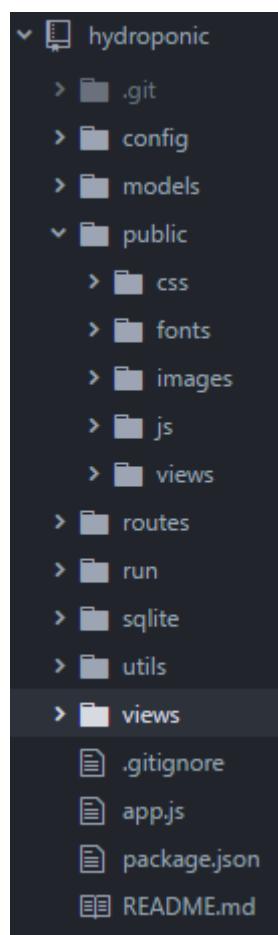
```
{  
    success: success,  
    message: message  
}
```

- Có data:

```
{  
    success: success,  
    data: data (*),  
    message: message  
}
```

Trong đó success nhận giá trị true nếu request được xử lý thành công và fail nếu thất bại.

Chi tiếp các API được thiết kế cho các resource được trình bày trong phần phụ lục B cuối luận văn.



Hình 108: Cây thư mục mã nguồn.

4.2.4 Cấu trúc thư mục

Server được tổ chức theo mô hình MVC với các thư mục chính:

- models: chứa các model cho các đối tượng tương ứng trong database.
 - routes: controller tiếp nhận các request và xử lý.
 - views : các file html hiển thị.

Ngoài ra, các thư mục khác:

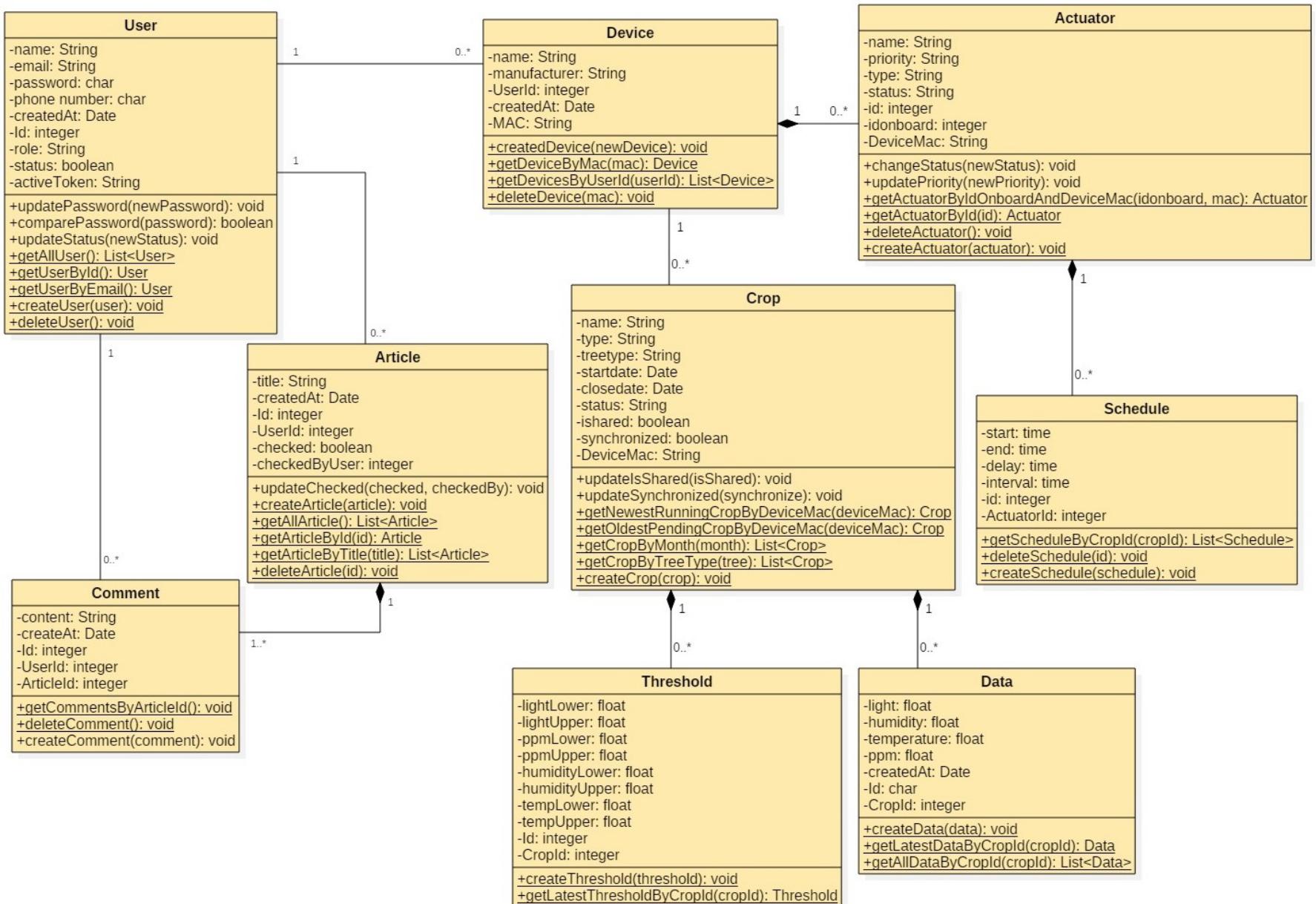
- config: thiết lập thông tin database khi kết nối.
 - public: chứa mã css, javascript, angularjs, images, các thư viện front-end cần dùng.
 - run: cấu hình cho server như port,...

- utils: các file chứa hàm hỗ trợ.
- file app.js: file server chính.
- file package.json: các gói hỗ trợ trong ứng dụng dành cho nodejs.

4.2.5 Xây dựng database

Sử dụng Sequelize để hiện thực database giống như sơ đồ sau. Chú thích các đối tượng trong sơ đồ lớp:

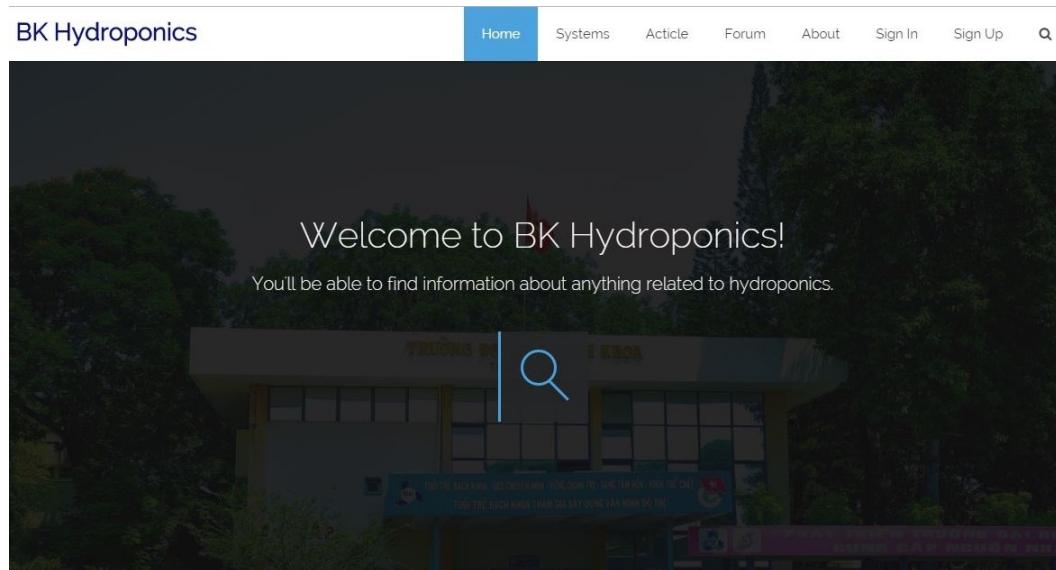
- User: người sử dụng.
- Device: board điều khiển.
- Article: bài viết do người dùng viết.
- Comment: bình luận trong bài viết.
- Crop: một mảng vụ.
- Threshold: ngưỡng dữ liệu.
- Data: dữ liệu cảm biến từ board điều khiển gửi lên.
- Actuator: các thiết bị máy bơm, quạt, đèn.
- Schedule: lịch trình hoạt động của mỗi actuator.



Hình 109: Thiết kế sơ đồ lớp.

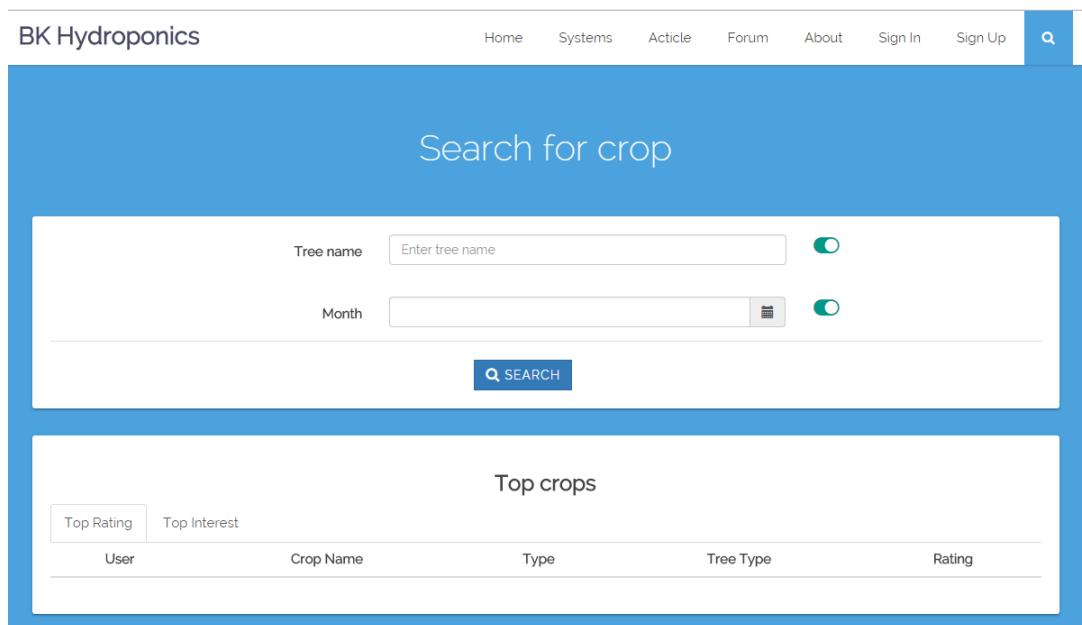
4.2.6 Một số hình ảnh

- Giao diện trang chủ sau khi truy cập vào địa chỉ hệ thống.



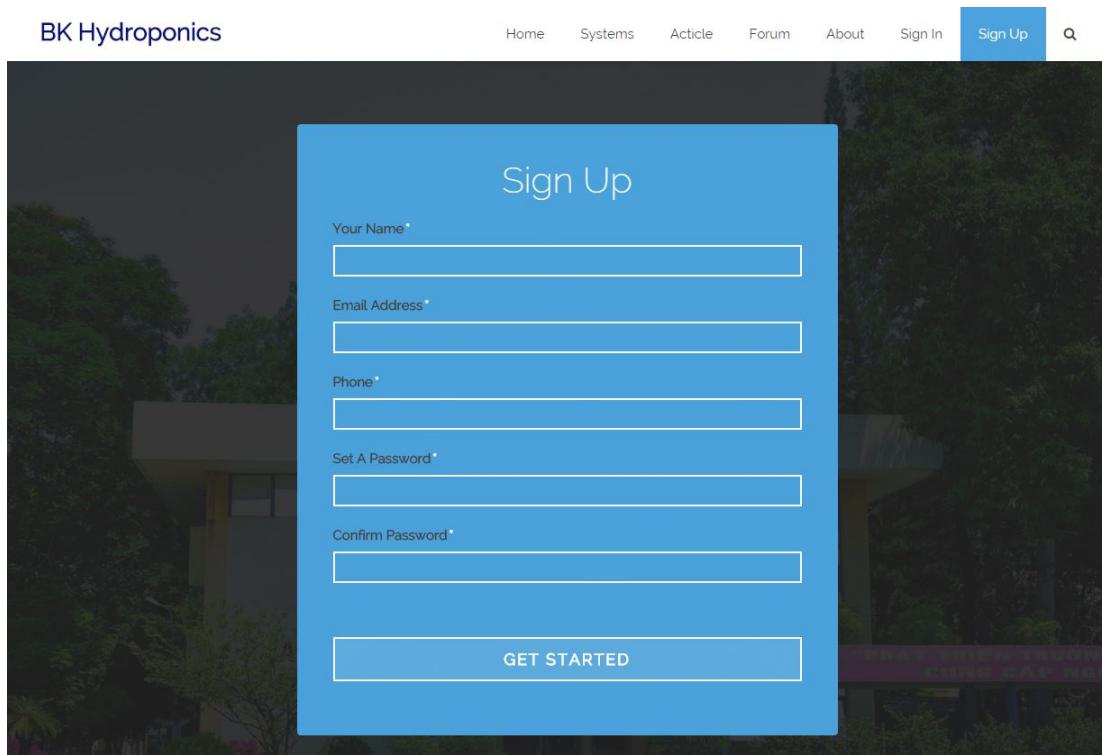
Hình 110: Trang chủ.

- Giao diện chức năng tìm kiếm mùa vụ: Người dùng có thể tìm kiếm mùa vụ do người khác chia sẻ để tham khảo các thông tin mùa vụ đó. Có thể tìm kiếm theo loại cây trồng, theo tháng trồng cây trong năm.



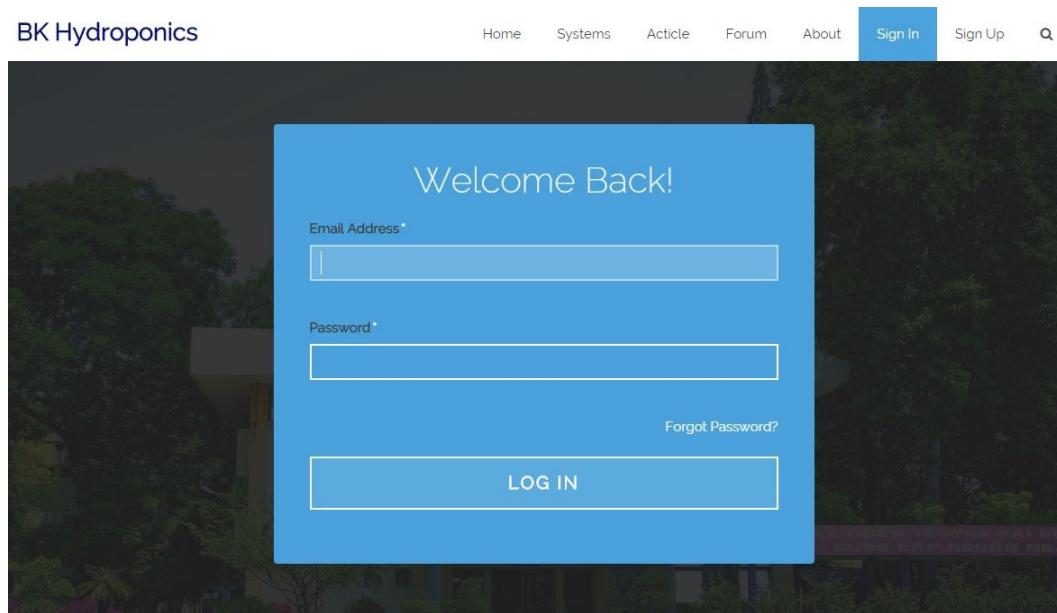
Hình 111: Trang tìm kiếm mùa vụ.

- Giao diện trang đăng ký trở thành thành viên của hệ thống.



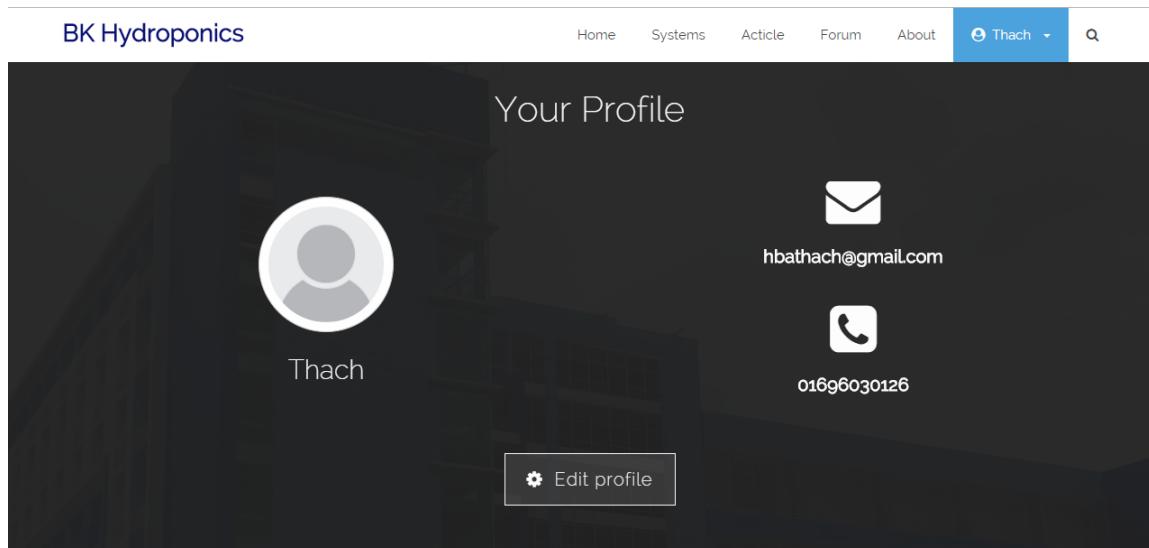
Hình 112: Trang đăng ký.

- Giao diện trang đăng nhập vào hệ thống.



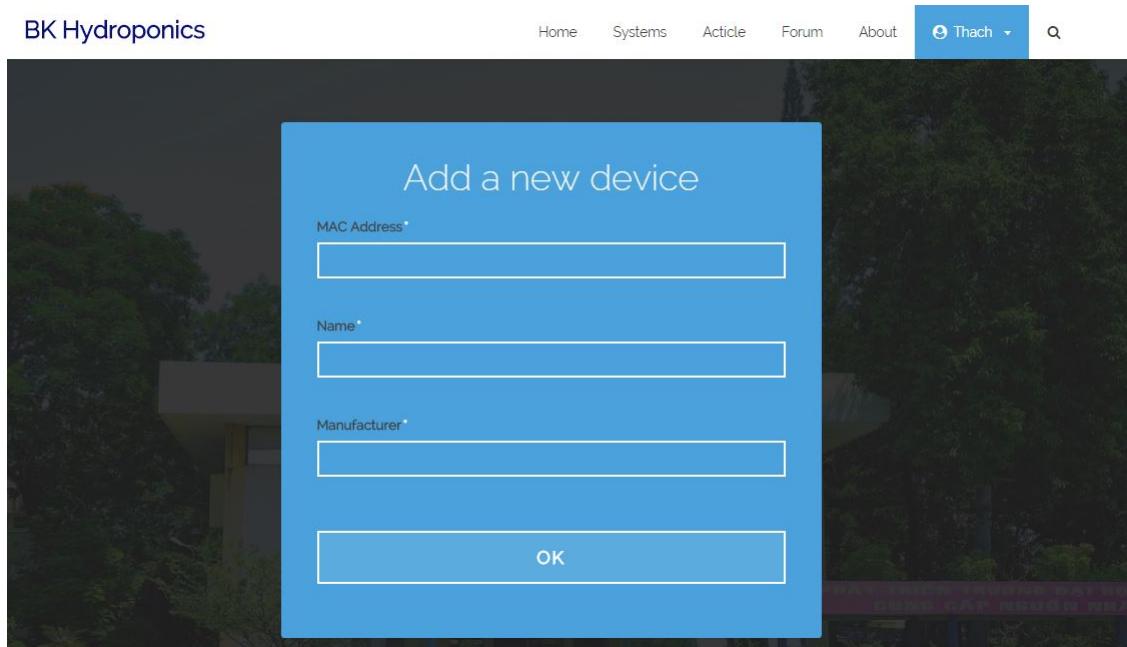
Hình 113: Trang đăng nhập.

- Giao diện trang cá nhân sau khi đăng nhập hệ thống, hiển thị thông tin cá nhân của tài khoản và danh sách các thiết bị (board) liên kết với tài khoản này.



Hình 114: Trang cá nhân.

- Giao diện sau khi nhấn chọn thêm vào một board mới cho tài khoản. Người dùng nhập vào các thông tin về: địa chỉ MAC của thiết bị, tên thiết bị, tên nhà sản xuất (BK Hydroponic). Một tài khoản có thể quản lý nhiều thiết bị (board) cùng lúc.



Hình 115: Thêm một board mới.

- Giao diện danh sách các board điều khiển đã thêm vào tài khoản (nằm cùng trang thông tin tài khoản).

The screenshot shows a table with the following data:

No.	MAC Address	Device Name	Installation Date	Status	Remarks	Operation
1	223344556677	smart device 22	13-12-2017 10:35:22	stop	sony	

Below the table, there is a message: "Showing 1 to 1 of 1 entries". At the bottom right, there are "Previous" and "Next" buttons, and a "1" button. A large blue button at the bottom center says "+ Add new device".

Hình 116: Danh sách các board điều khiển.

- Giao diện danh sách các actuator điều khiển trên thiết bị (board), sau khi nhấn vào xem thông tin của một thiết bị. Một thiết bị có thể có nhiều actuator điều khiển quạt, bơm nước, đèn và sục oxy.

The screenshot shows a table with the following data:

No.	ID	Type	Priority	Status	Created Date	Operation
1	1	Water	Primary	off	26-12-2017 09:11:08 +0700	
2	2	Water	Primary	off	26-12-2017 10:06:45 +0700	
3	1	Lighting	Primary	off	26-12-2017 10:07:16 +0700	

Below the table, there is a message: "Showing 1 to 3 of 3 entries". At the bottom right, there are "Previous" and "Next" buttons, and a "1" button. A large blue button at the bottom center says "New Actuator".

Hình 117: Danh sách các actuator của một board.

- Giao diện để thêm một actuator cho thiết bị. Người dùng nhập vào các thông tin: ID của actuator trên thiết bị, loại actuator (primary/secondary), tên actuator và loại actuator (bơm nước/quạt/sục oxy/dèn).

Add new actuator for this device

ID:

Priority:

Name:

Type:

Hình 118: Thêm một actuator.

- Giao diện danh sách các mùa vụ bao gồm mùa vụ đã kết thúc/đang chạy/cài đặt cho thời gian sắp tới, sau khi nhấn vào xem thông tin của một thiết bị. Một thiết bị có thể được cài đặt để chạy cho nhiều mùa vụ, tại các thời điểm khác nhau. Tại cùng một thời điểm chỉ có một mùa vụ đang chạy.

List of crops

List of crops of your device: 112233445566

Show	10	entries	Search:					
No.	Crop Name	Type of Hydroponics	Tree Type	Start Date	End Date	Status	Operation	Share
1	crop 1	aeroponics	salad	10-12-2017 15:51:00	31-12-2017 15:51:00	running		

Showing 1 to 1 of 1 entries

Previous Next

Hình 119: Danh sách các mùa vụ.

- Giao diện để thêm vào một mùa vụ mới, người dùng nhập vào các thông tin: Tên mùa vụ, loại cây trồng cho mùa vụ này, ngày bắt đầu mùa vụ, ngày kết thúc mùa vụ, thời gian giãn cách để thiết bị gửi dữ liệu giám sát lên và loại thủy canh cho mùa vụ này (một trong 6 loại thủy canh).

Add new crop for this device

Crop Name:

Tree Type:

Starting Date:

Closing Date:

Report Time: seconds

Type:

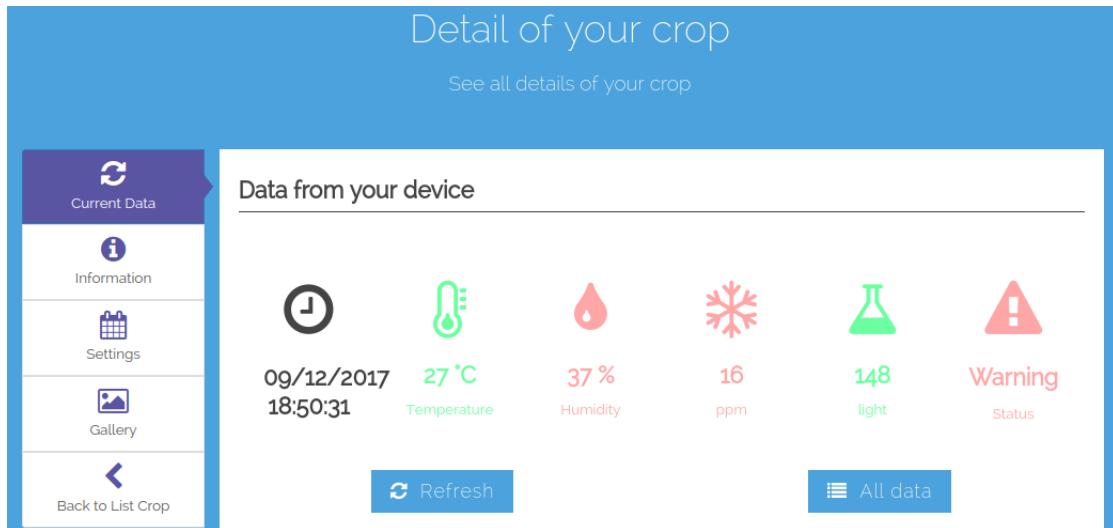
Hình 120: Thêm một mùa vụ.

- Giao diện thông tin chi tiết của mùa vụ, sau khi người dùng nhấn vào một mùa vụ trong danh sách các mùa vụ và chọn tab thông tin mùa vụ trong giao diện. Trang sẽ hiển thị thông tin về mùa vụ mà người dùng đã nhập lúc tạo.

MAC	112233445566
Crop name	crop 1
Hydroponics type	aeroponics
Tree type	salad
Start date	10-12-2017 15:51:00
End date	31-12-2017 15:51:00
Report delay time (s)	20 seconds

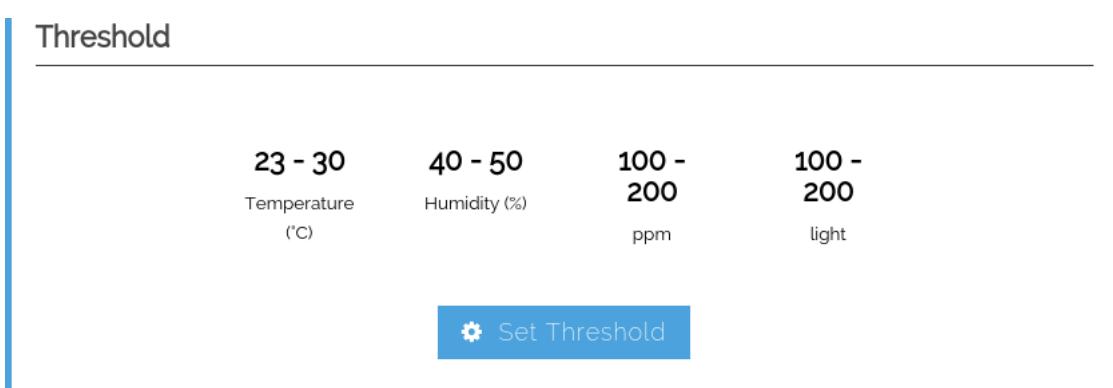
Hình 121: Thông tin mùa vụ.

- Giao diện giám sát các thông số chi tiết của mùa vụ, sau khi người dùng nhấn vào một mùa vụ trong danh sách các mùa vụ và chọn tab dữ liệu hiện tại trong giao diện. Hiển thị các thông tin mới nhất về nhiệt độ, độ ẩm, nồng độ dinh dưỡng và ánh sáng của hệ thống thủy canh.



Hình 122: Chi tiết mùa vụ.

- Giao diện ngưỡng dữ liệu, nằm cùng trong tab dữ liệu hiện tại. Giao diện hiển thị ngưỡng dữ liệu cảnh báo của hệ thống do người dùng cài đặt, bao gồm ngưỡng cao và thấp, vượt ngoài khõi ngưỡng này hệ thống sẽ hiển thị cảnh báo cho người dùng.



Hình 123: Ngưỡng dữ liệu.

- Giao diện lịch trình hoạt động của các actuator của thiết bị, hiển thị thời gian hoạt động của các actuator do người dùng cài đặt cho thiết bị. Danh sách lịch trình này được cài đặt cho các actuator để thực hiện bơm nước/quạt/sục oxy/dèn. Mỗi

actuator có thể được cài đặt nhiều lịch trình. Mỗi lịch trình bao gồm các thông tin: Thời gian bắt đầu, thời gian kết thúc, thời gian thực hiện và thời gian giãn cách giữa các lần thực hiện. Danh sách lịch trình hiển thị ở giao diện có thể đang đồng bộ hoặc không đồng bộ với thiết bị được thể hiện thông qua nút trạng thái trong giao diện. Nếu không đồng bộ, người dùng nhấn nút để đồng bộ lịch trình với thiết bị.

The screenshot shows a user interface titled 'Hardware Settings'. On the left is a vertical sidebar with icons for 'Current Data', 'Information', 'Settings' (which is selected and highlighted in purple), and 'Gallery'. Below the sidebar is a link 'Back to List Crop'. The main area is titled 'Hardware Settings' and contains a table of scheduled events. The table has columns: Type, Name, Start Time, End Time, Interval Time, Delay Time, and Operation. One row is visible: 'Water' under Type, 'water 1' under Name, '17:33:00' under Start Time, '17:40:00' under End Time, '5' under Interval Time, '5' under Delay Time, and a trash bin and gear icon under Operation. At the top right of the main area are 'Status' (checked) and a 'Sync' button. Below the table are buttons for 'Previous' (with page number 1) and 'Next'.

Hình 124: Lịch trình hoạt động của các actuator.

- Giao diện thêm một lịch trình. Người dùng chọn loại actuator, rồi chọn actuator trong danh sách các actuaor đang có, nhập các thông tin như: thời gian bắt đầu, thời gian kết thúc, thời gian thực hiện và thời gian giãn cách. Hệ thống sẽ kiểm tra thời gian bắt đầu và kết thúc của lịch trình này có bị trùng với lịch trình nào khác đã được cài đặt.

The screenshot shows a modal dialog box titled 'Add new settings for this device'. It contains fields for 'Relay Type' (dropdown), 'Relay Name' (dropdown), 'Start Time' (text input with a clock icon), 'End Time' (text input with a clock icon), 'Interval Time' (text input with 'seconds' unit), and 'Delay Time' (text input with 'seconds' unit). At the bottom are 'OK' and 'CLOSE' buttons.

Hình 125: Thêm lịch trình cho actuator.

- Giao diện trang diễn đàn, được xây dựng trên nền tảng NodeBB. Được cấu hình Single-Sign-On (SSO) - chia sẻ trạng thái đăng nhập của hệ thống chính với hệ thống đăng nhập trang diễn đàn trên nền tảng NodeBB. Tài khoản đăng nhập diễn đàn trên nền tảng NodeBB cũng chính là tài khoản đăng nhập hệ thống BK Hydroponics.

The screenshot shows the homepage of a NodeBB forum. At the top, there's a navigation bar with icons for search, notifications, and user profile. Below it is a 'CATEGORIES' section with four items:

- Thông báo**: Contains 0 topics and 0 posts. Description: Các thông báo của ban quản trị.
- Hỏi đáp**: Contains 1 topic and 1 post. Description: Đặt câu hỏi và giải đáp các thắc mắc. A recent post is shown: "Welcome to your brand new NodeBB forum! This is what a topic and post looks like. As".
- Chia sẻ**: Contains 0 topics and 0 posts. Description: Các bài viết chia sẻ kinh nghiệm.
- Góc kỹ thuật**: Contains 0 topics and 0 posts. Description: Trao đổi các kỹ thuật trồng rau.

At the bottom, it says "Powered by NodeBB | Contributors".

Hình 126: Trang forum.

- Giao diện trang quản lý dành cho quyền Administrator/Moderator. Hiện tại, tài khoản quyền Administrator sẽ có quyền quản lý bài viết, quản lý các tài khoản; tài khoản có quyền Moderator sẽ chỉ có quyền quản lý bài viết.

The screenshot shows the administration dashboard of the BK Hydroponics system. On the left is a sidebar with a user profile for 'Thach' and links to 'Back to Home Page', 'Administration', 'Manage Users', and 'Manage Articles'. The main area has three buttons:

- Back to Home Page** (with a house icon)
- Manage Users** (with a person icon)
- Manage Articles** (with a document icon)

Hình 127: Trang quản lý dành cho administrator/moderator.

4.3 Ứng dụng di động

4.3.1 Môi trường phát triển

Các môi trường và công cụ để phát triển ứng dụng di động:

- Hệ điều hành Manjaro.
- Angular 4.4.3.
- Ionic 2.
- Visual studio code.
- Android sdk 16 (hỗ trợ từ android 4.1 trở lên).
- Genymotion 2.11.0 để kiểm thử ứng dụng.

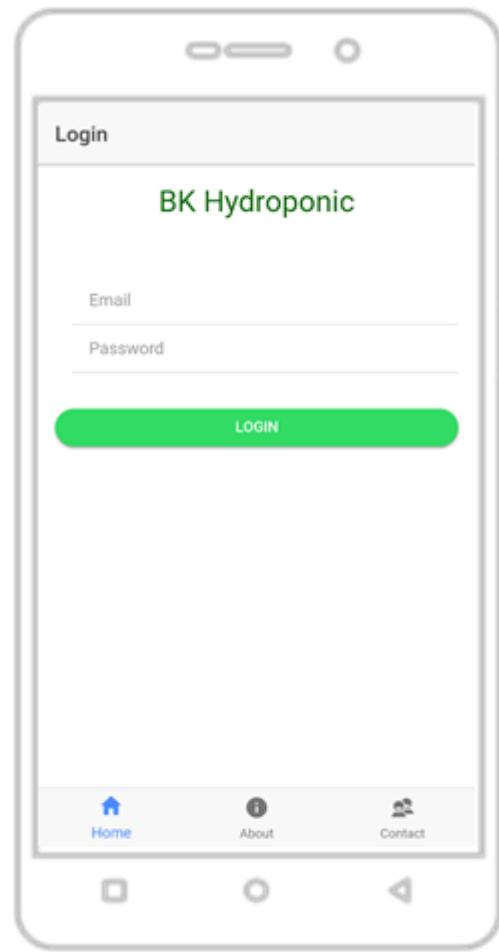
4.3.2 Các chức năng chính

Ứng dụng mobile với các chức năng tương tự như ứng dụng web, tuy nhiên với thời gian hạn chế nên hiện tại nhóm chỉ mới phát triển một số chức năng cơ bản sau:

- Đăng nhập.
- Xem danh sách các board điều khiển.
- Quét mã QR để thêm nhanh board mới. Mã QR sẽ đi kèm trên từng sản phẩm thiết bị (board).
- Xem thông tin môi trường từ cảm biến.
- Theo dõi các actuator và điều khiển bật tắt chúng.
- Hiển thị thông báo khi thông số môi trường vượt ngưỡng cho phép.

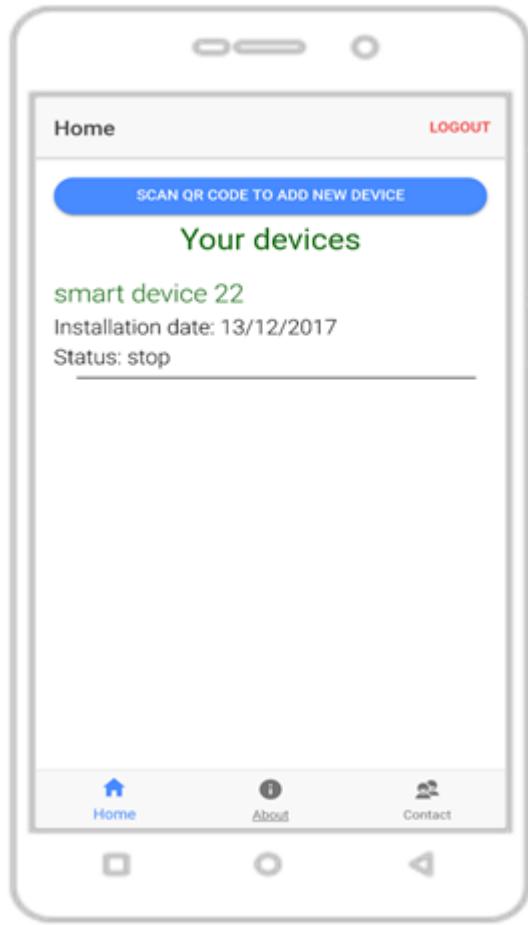
4.3.3 Một số hình ảnh

- Chức năng đăng nhập tài khoản.



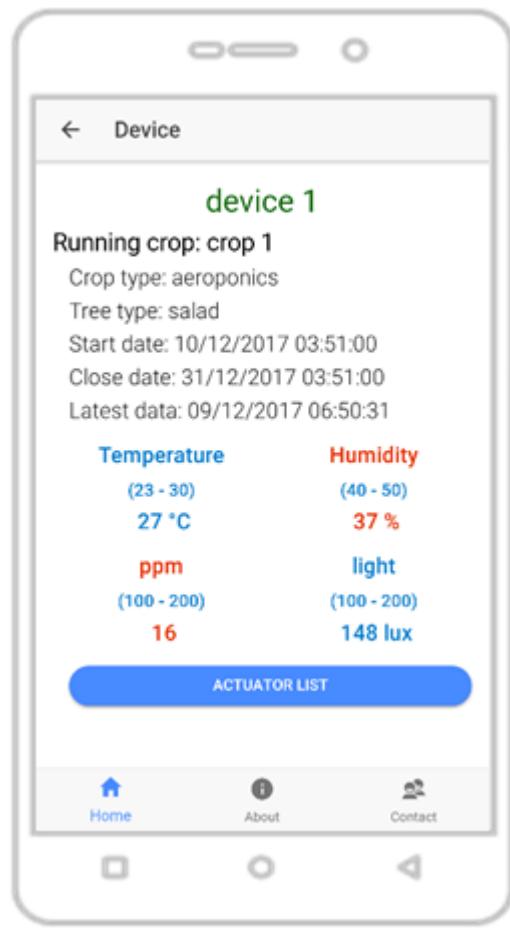
Hình 128: Trang đăng nhập.

- Xem thông tin các thiết bị của tài khoản sau khi đăng nhập.



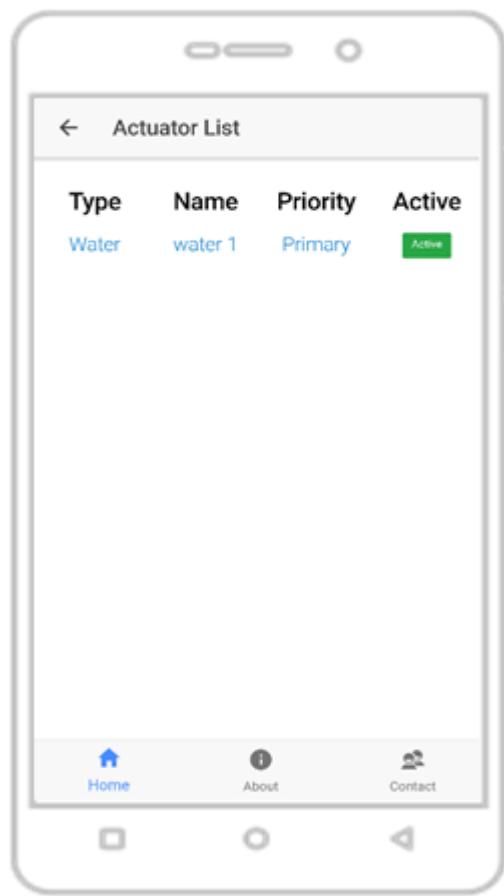
Hình 129: Trang chủ sau khi đăng nhập.

- Xem thông tin chi tiết của mùa vụ và thiết bị.



Hình 130: Thông tin mùa vụ và dữ liệu.

- Xem danh sách các actuator trên thiết bị và điều khiển bật tắt.



Hình 131: Danh sách các actuator.

5 KIỂM THỬ HỆ THỐNG

5.1 Web server

Web server được triển khai trên cloud Google engine ở địa chỉ <http://35.198.199.4:3210/>. Các tính năng CRUD trên các đối tượng User, Article, Comment, Device, Crop, Actuator, Schedule, Threshold hoạt động ổn định.

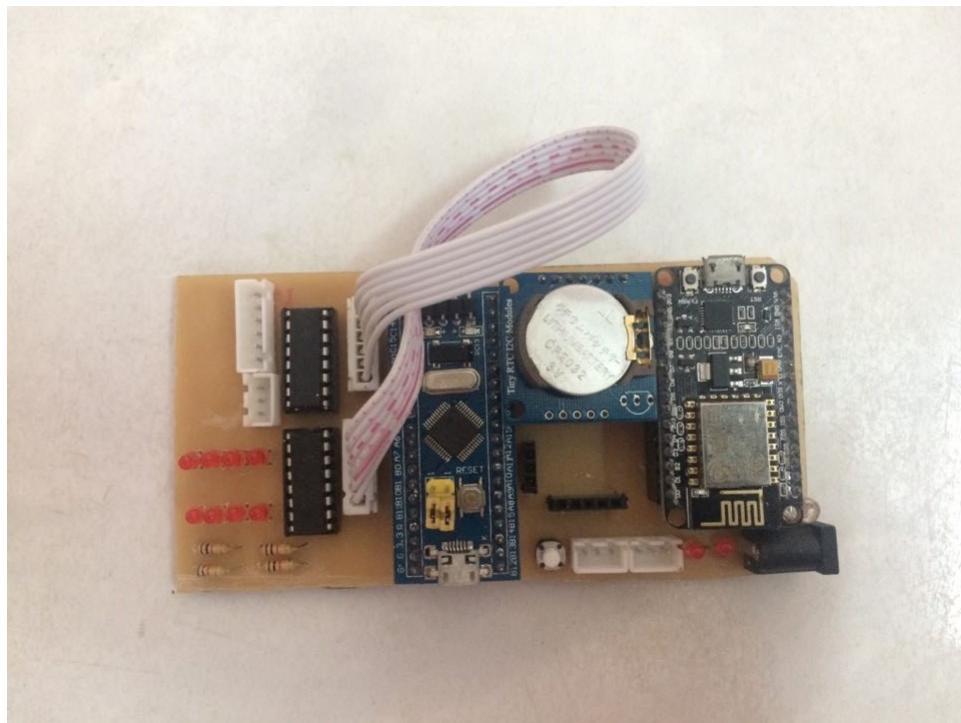
5.2 Ứng dụng di động

Ứng dụng được kiểm thử trên máy ảo google nexus 5 chạy android 6.0 của Genymotion và máy thật samsung galaxy note 5 chạy android 7.0. Các chức năng đăng nhập, hiển thị thông tin và điều khiển các actuator hoạt động ổn định.

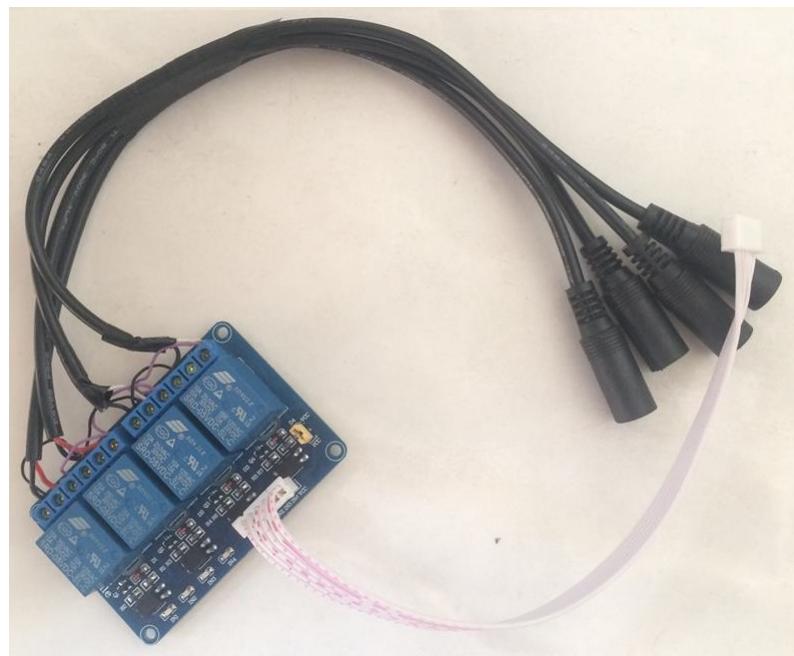
5.3 Board điều khiển

5.3.1 Thiết bị

- Prototype:



Hình 132: Board prototype.



Hình 133: Relay prototype.

- Dóng hộp:

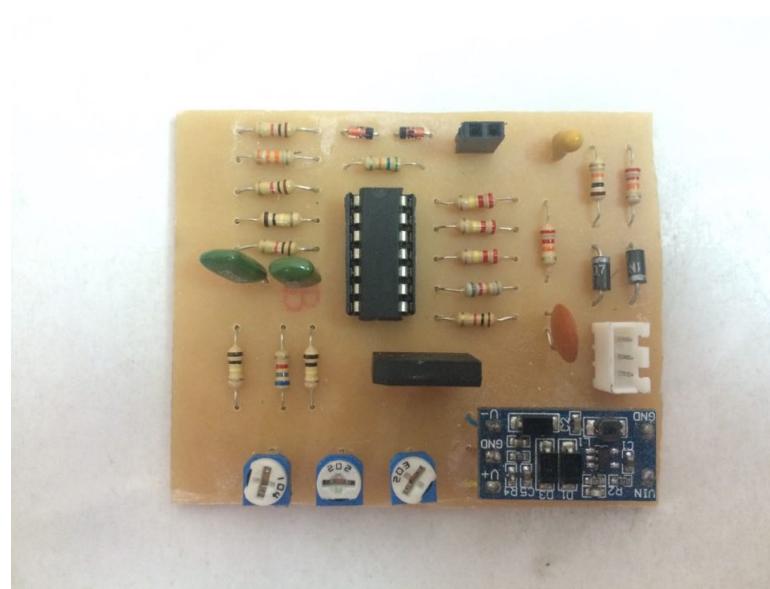


Hình 134: Dóng hộp.

5.3.2 Các chức năng

- Mạch đo chức năng PPM.

– Prototype:

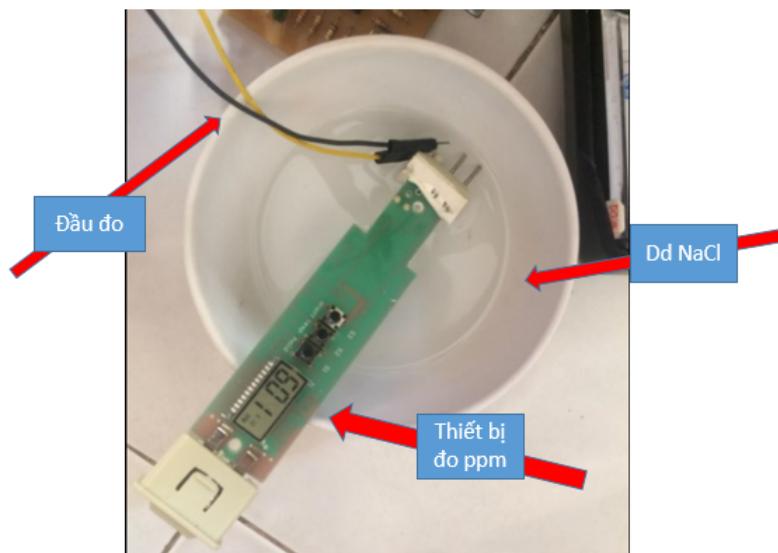


Hình 135: PPM prototype.

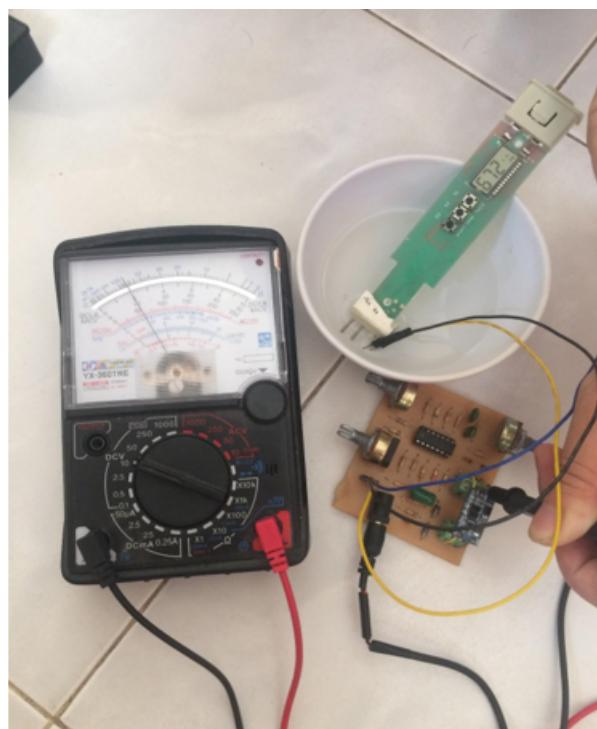


Hình 136: PPM đóng vỏ.

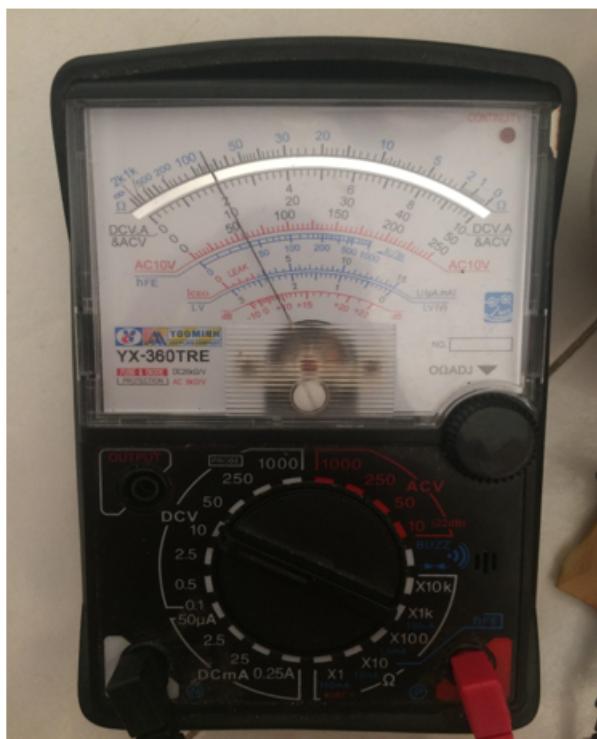
– Tiến hành đo:



Hình 137: Đo ppm dung dịch muối ăn 1.

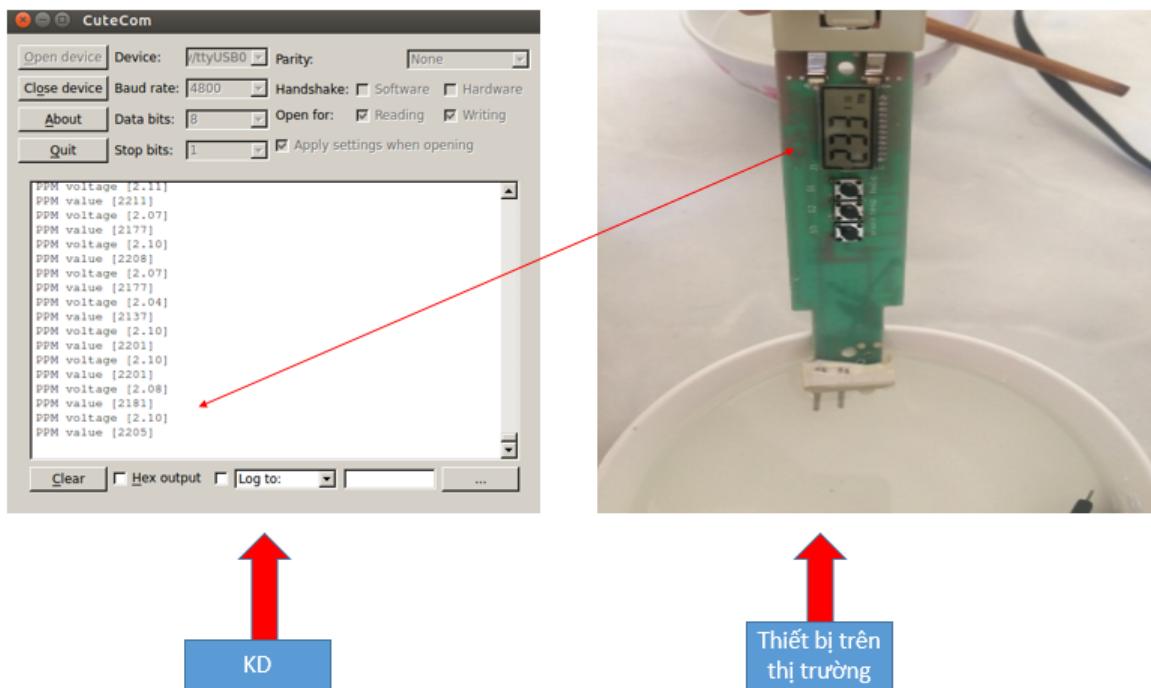


Hình 138: Đo ppm dung dịch muối ăn 2.

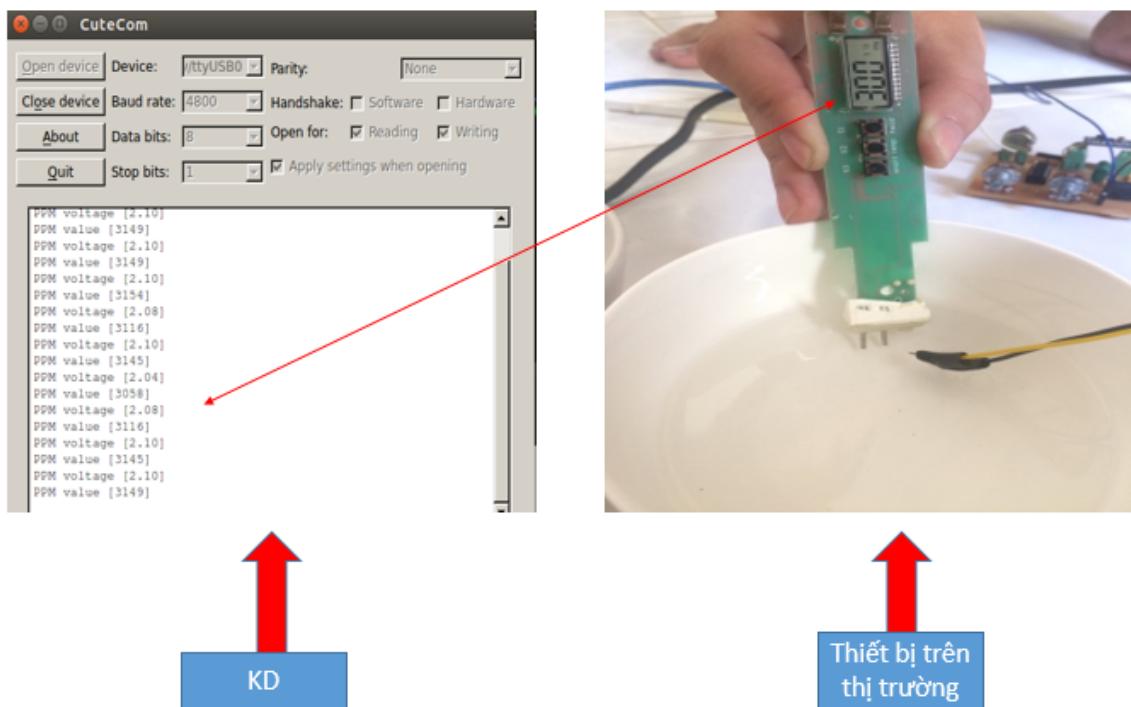


Hình 139: Đo ppm dung dịch muối ăn 3.

– Kết quả:



Hình 140: Kết quả đo ppm 1.



Hình 141: Kết quả đo ppm 2.

Các chức năng đã hoạt động ổn định:

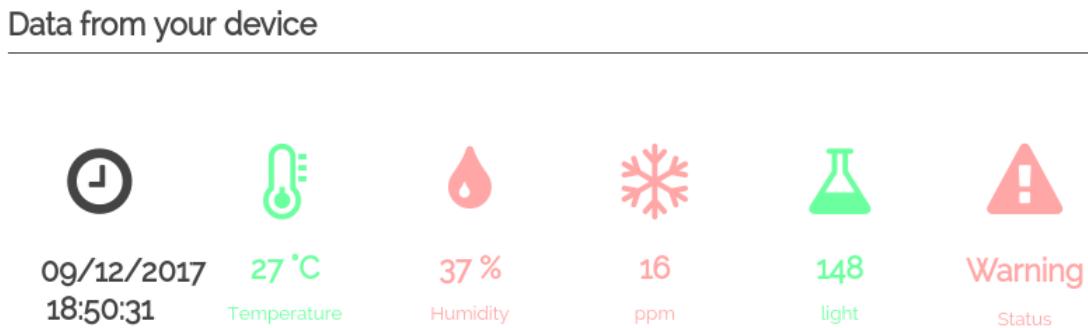
- Thu thập dữ liệu cảm biến:
 - Đọc dữ liệu nhiệt độ, độ ẩm.
 - Đọc dữ liệu ánh sáng.
 - Đọc dữ liệu ppm.
 - Định thời.
 - Điều khiển các actuators:
 - Đọc thời gian thực.
 - Hoạt động theo schedule.
 - Lưu trữ trạng thái các actuator vào EEPROM.
 - Tải lại trạng thái các actuator từ EEPROM.
 - Đèn báo tín hiệu các actuator.

- Giao tiếp với webserver:
 - Protocol hoạt động đúng.
 - Protocol Mqtt truyền nhận đúng.
- Cấu hình wifi, mqtt:
 - Chuyển chế độ từ Station sang Access Point.
 - Webserver hoạt động ổn định.
 - Lưu trữ các thông số sau khi cấu hình vào EEPROM.
 - Tải lại các thông số sau khi reset.

5.4 Kết nối giữa ứng dụng web và board điều khiển

5.4.1 Dữ liệu nhận từ cảm biến

Hệ thống tiếp nhận, xử lý và hiển thị các thông số môi trường, đồng thời hiển thị trạng thái dữ liệu nếu dữ liệu vượt ngưỡng định trước



Hình 142: Dữ liệu gửi lên từ cảm biến.

5.4.2 Điều khiển các actuator

Đã tiến hành thử nghiệm chức năng hẹn giờ và hoạt động theo giờ gian biểu của actuators:

- Các actuator được thêm và kích hoạt từ webserver
- Người sử dụng lên lịch và gửi cho board điều khiển.

- Sau khi nhận gói tin lịch trình từ webserver board điều khiển có thể hoạt động độc lập theo thời gian chính xác.

Chức năng được tiến hành thử nghiệm và liên tục cải thiện tính năng và độ chính xác trong quá trình làm. Chi tiết demo tính năng đã được quay lại, xem tại các đường dẫn online sau:

- Quá trình thêm board điều khiển, actuator và lịch trình:

<https://www.youtube.com/watch?v=1S7umkosvZ0&feature=youtu.be>

- Quá trình chạy actuator (đèn, máy sục oxy) theo lịch trình:

<https://www.youtube.com/watch?v=uav8kdqJhKw&feature=youtu.be>

- Quá trình nhận dữ liệu cảm biến từ board điều khiển:

https://www.youtube.com/watch?v=RCdho_9JA1c&feature=youtu.be

6 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

6.1 Tổng kết

6.1.1 Kết quả đạt được

- Web server API và web app:

Sử dụng Nodejs và Angularjs hiện thực thành công các chức năng:

- Cung cấp hệ thống API cho web app và mobile app.
- Phân quyền người dùng: member, mod và admin.
- Vẽ đồ thị theo dõi các thông số môi trường.
- Giao tiếp với board điều khiển thông qua hệ thống giao thức đã thiết kế.

- Mobile app:

Sử dụng framework Ionic 2 hiện thực thành công các chức năng:

- Đăng nhập
- Xem danh sách các board điều khiển.
- Quét mã QR để nhanh chóng thêm board điều khiển mới.
- Giám sát thông tin môi trường.
- Theo dõi các actuator và điều khiển bật, tắt chúng.

- Board điều khiển

- Mạch đo chức năng PPM.
- Thu thập dữ liệu cảm biến:

- * Đọc dữ liệu nhiệt độ, độ ẩm.
- * Đọc dữ liệu ánh sáng.
- * Đọc dữ liệu ppm.
- * Định thời.

- Điều khiển các actuator:

- * Đọc thời gian thực.
- * Hoạt động theo schedule
- * Lưu trữ trạng thái các actuator vào EEPROM.
- * Tải lại trạng thái các actuator từ EEPROM.
- * Đèn báo tín hiệu các actuator.
- Giao tiếp với webserver:
 - * Protocol hoạt động đúng.
 - * Protocol Mqtt truyền nhận đúng.
- Cấu hình wifi, mqtt:
 - * Chuyển chế độ từ Station sang Access Point.
 - * Webserver hoạt động ổn định.
 - * Lưu trữ các thông số sau khi cấu hình vào EEPROM.
 - * Tải lại các thông số sau khi reset.

6.1.2 Những hạn chế của hệ thống

- Thiết bị đo ppm còn sai số cao so với những dung dịch có giá trị ppm lớn.
- Giao diện web còn nhiều điểm chưa thuận tiện cho người dùng.
- Mới dừng ở mức cảnh báo khi các thông số vượt ngưỡng.
- Chưa hiện thực hoàn chỉnh giao thức giao tiếp giữa board điều khiển và web server đã thiết kế.

6.1.3 Thuận lợi, khó khăn trong quá trình thực hiện đề tài

Thuận lợi

- Được sự hỗ trợ về kinh phí và kiến thức từ giáo viên hướng dẫn.

Khó khăn

- Dành nhiều thời gian để tìm hiểu các công nghệ mới.
- Dành nhiều thời gian cho việc hiện thực, chưa có nhiều thời gian để thử nghiệm hệ thống.
- Thiếu các thiết bị di động để thử nghiệm ứng dụng di động trên nhiều nền tảng khác nhau.
- Khó khăn trong việc tìm thiết kế và hiện thực giao thức giao tiếp, gửi nhận các gói tin giữa board điều khiển và web api.

6.1.4 Kiến thức thu thập được

- Có được những hiểu biết nhất định về thủy canh.
- Học hỏi được các công nghệ mới như Nodejs, Angularjs, Angular 2, Ionic.
- Biết thêm về các giao thức thường dùng trong IoT.
- Rèn luyện các kỹ năng thiết kế phần mềm, thiết kế CSDL đã học.

6.2 Hướng mở rộng hệ thống

6.2.1 Ứng dụng web

- Nghiên cứu kỹ hơn về ảnh hưởng của các nhân tố môi trường (nhiệt độ, độ ẩm, pH, ppm, ánh sáng, oxy) đến sự phát triển của các loại cây phổ biến trong từng thời kỳ, từ đó đưa ra các giải pháp tự động hóa việc chăm sóc cho từng giai đoạn phát triển của cây.
- Mở rộng chức năng chia sẻ cộng đồng: Người dùng có thể upload ảnh đại diện, upload ảnh về các mùa vụ mà mình đang trồng và chia sẻ nó. Cho phép người dùng xuất ra các file setting của một mùa vụ nào đó và chia sẻ nó, người dùng khác có thể nạp file đó vào profile của mình và chạy trên thiết bị.

- Cải thiện kết nối giữa server và board điều khiển: Khắc phục thời gian chờ gói tin ACK từ board điều khiển bằng cách thiết lập một khoảng thời gian nhất định cho việc chờ gói tin ACK. Quá khoảng thời gian đó thì phản hồi về cho website là gói tin không đến được board điều khiển.
- Xây dựng giao diện đa ngôn ngữ, có thể chuyển đổi qua lại.
- Cải thiện hơn về giao diện và trải nghiệm.

6.2.2 Ứng dụng di động

- Cải thiện hơn về giao diện và trải nghiệm người dùng.
- Xây dựng đầy đủ các chức năng cho ứng dụng di động như các chức năng đã hiện thực cho ứng dụng web.

6.2.3 Board điều khiển

- Hoàn thiện mạch chức năng ppm.
- Hoàn thiện board điều khiển: nghiên cứu xây dựng mạch đo pH và các thông số khác.
- Phát triển mô hình hệ thống theo hướng thương mại hóa, tiến hành thử nghiệm thực tế.

6.3 Quá trình làm việc của nhóm

Công việc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tìm hiểu về thủy canh, các loại thủy canh và cách hoạt động của mỗi loại	x														
Tìm hiểu Nodejs		x	x												
Tìm hiểu Angularjs			x	x											
Thiết kế database (thiết kế, bổ sung)			x	x	x	x	x	x	x						
Thiết kế protocol (thiết kế, bổ sung)			x	x	x	x	x	x	x	x	x	x	x		
Xây dựng server API (xây dựng, cập nhật, sửa lỗi)			x	x	x	x	x	x	x	x	x	x	x	x	x
Xây dựng front-end (xây dựng, cập nhật, sửa lỗi)			x	x	x	x	x	x							
Tìm hiểu angular 2, framework ionic 2									x	x					
Xây dựng ứng dụng di động (xây dựng, cập nhật, sửa lỗi)									x	x	x	x	x		
Viết báo cáo											x	x	x		

Bảng 17: Bảng công việc của Huỳnh Bá Thạch.

Công việc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tìm hiểu về thủy canh, các loại thủy canh và cách hoạt động của mỗi loại	x	x													
Tìm hiểu về các ứng dụng IoT và các giao thức dùng cho IoT	x	x	x												
Tìm hiểu Nodejs và Angularjs		x	x	x	x										
Thiết kế database (thiết kế, bổ sung)		x	x	x	x	x	x								
Tìm hiểu và thiết kế xây dựng giao diện ứng dụng web (thiết kế, xây dựng, cập nhật)			x	x	x	x	x	x	x	x	x				
Xây dựng server API (xây dựng, cập nhật, sửa lỗi)			x	x	x	x	x	x	x	x	x	x	x	x	
Xây dựng front-end (xây dựng, cập nhật, sửa lỗi)			x	x	x	x	x	x	x	x	x	x	x	x	x
Viết báo cáo												x	x	x	x

Bảng 18: Bảng công việc của Tạ Chí Tây.

Công việc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Định hình công việc	x														
Tìm hiểu kiến thức về máy đo ppm	x	x													
Thử nghiệm mô phỏng, schematic mạch đo ppm		x	x	x											
Thiết kế prototype, làm mạch, thử nghiệm máy đo ppm			x	x	x		x	x							
Tìm hiểu gitlab, xây dựng cấu trúc project					x	x	x								
Thiết kế, hiện thực chức năng web-server trên board Node MCU					x	x	x	x	x						
Thiết kế prototype, schematic và PCB cho board điều khiển					x	x	x	x							
Thiết kế, thực hiện mô hình thử nghiệm							x	x	x	x					
Viết báo cáo									x	x	x	x			

Bảng 19: Bảng công việc của Nguyễn Lê Minh Khôi.

Công việc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Định hình công việc	x														
Tìm hiểu, thử nghiệm Arduino	x	x	x												
Tìm hiểu, thử nghiệm Arm, Mbed		x	x	x											
Tìm hiểu gitlab, cấu trúc project			x	x	x										
Thiết kế phần mềm			x	x											
Hiện thực thiết kế tái thiết kế				x	x	x	x	x	x	x					
Hiện thực mô hình thử nghiệm									x	x	x	x			
Chạy thử mô hình										x	x	x	x		
Viết báo cáo								x	x	x	x	x	x	x	x

Bảng 20: Bảng công việc của Nguyễn Dinh Dung.

Tài liệu

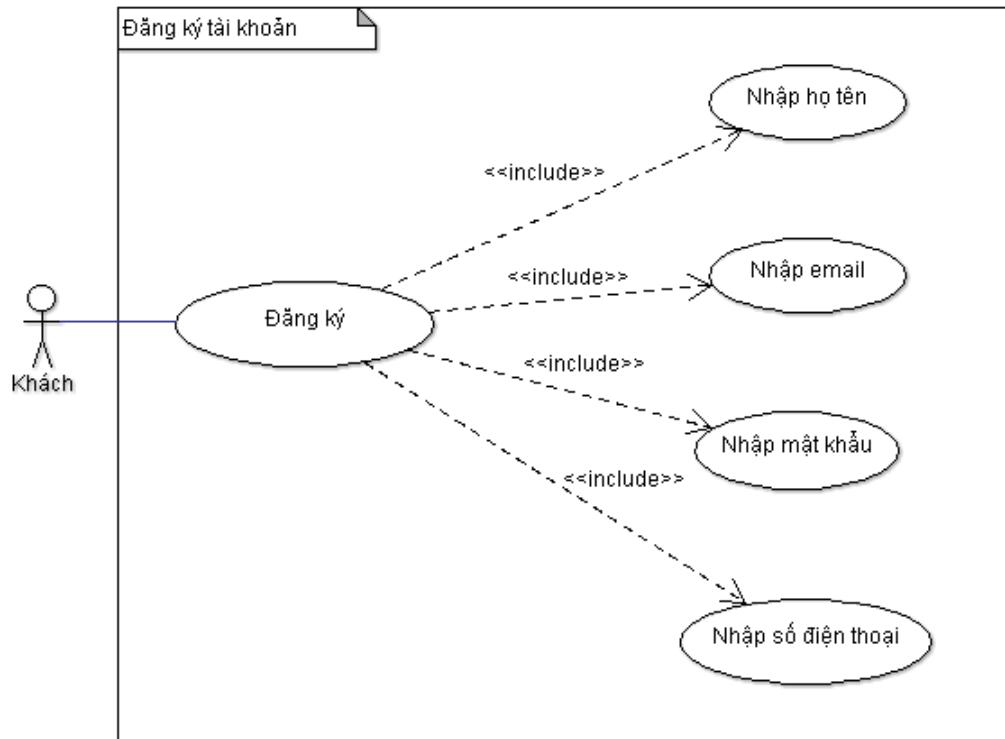
- [1] Wikipedia, *Internet Vạn Vật*, <https://vi.wikipedia.org/>, [Online; Cập nhật lần cuối vào 13/12/2017]
- [2] Gartner, *Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016*, <https://www.gartner.com/>, [Online; Cập nhật lần cuối vào 07/02/2017]
- [3] devlog.vn, *Giới thiệu tổng quát về IoT*, <https://devlog.vn/>, [Online; Cập nhật lần cuối vào 28/10/2017]
- [4] Wikipedia, *Thủy canh*, <https://vi.wikipedia.org/>, [Online; Cập nhật lần cuối vào 17/08/2017]
- [5] Home Hydro Systems, *The basic Hydroponic System types*, <http://www.homehydrosystems.com/>, [Online; Cập nhật lần cuối vào 2017]
- [6] hachi.com.vn, *Chỉ số EC và TDS – Hai chỉ số quan trọng trong thủy canh*, <http://hachi.com.vn/>, [Online; Cập nhật lần cuối vào 06/09/2017]
- [7] codehub.vn, *RESTful Web Services*, <https://www.codehub.vn/>, [Online; Cập nhật lần cuối vào 19/03/2016]
- [8] Freetuts.net, *Cấu trúc chuỗi JSON*, <https://freetuts.net/>, [Online; Cập nhật lần cuối vào 06/12/2014]
- [9] Vietjack (2015), *Nodejs*, <http://vietjack.com/>, [Online; Cập nhật lần cuối vào 2017]
- [10] Expressjs.com, *ExpressJS Framework*, <https://expressjs.com/>, [Online; Cập nhật lần cuối vào 2017]
- [11] Sequelizejs.com, *Sequelizejs documentation*, <http://docs.sequelizejs.com/>, [Online; Cập nhật lần cuối vào 2017]
- [12] Freetuts.net, *Tổng quan về AngularJS*, <https://freetuts.net/>, [Online; Cập nhật lần cuối vào 14/11/2014]

- [13] Viblo.asia, *Angular 2*, <https://viblo.asia/> [Online; Cập nhật lần cuối vào 25/02/2017]
- [14] EnvatoTuts+, *Giới thiệu Ionic 2*, <https://code.tutsplus.com/>, [Online; Cập nhật lần cuối vào 22/02/2017]
- [15] Wikipedia, *TypeScript*, <https://vi.wikipedia.org/>, [Online; Cập nhật lần cuối vào 20/04/2016]
- [16] techmaster.vn, *Khái niệm về JSON Web Token*, <https://techmaster.vn/>, [Online; Cập nhật lần cuối vào 10/07/ 2016]
- [17] Wikipedia, *PostgreSQL* , <https://vi.wikipedia.org/>, [Online; Cập nhật lần cuối vào 22/08/2017]
- [18] NodeBB.org, *NodeBB framework*, <https://docs.archive.nodebb.org/>, [Online; Cập nhật lần cuối vào 2014]
- [19] atmarkcafe.org, *Git và Github*, <http://atmarkcafe.org/>, [Online; Cập nhật lần cuối vào 25/04/2015]
- [20] picvietnam.com, *Giải thuật mật mã hóa XTEA với ASM30* , <http://www.picvietnam.com/>, [Online; Cập nhật lần cuối vào 06/08/2007]
- [21] Wikipedia, *XXTEA*, <https://en.wikipedia.org/>, [Online; Cập nhật lần cuối vào 13/03/2017]
- [22] Wikipedia, *Cấu trúc ARM*, <https://vi.wikipedia.org/>, [Online; Cập nhật lần cuối vào 16/08/2017]
- [23] Wikipedia, *mbed*, <https://en.wikipedia.org/>, [Online; Cập nhật lần cuối vào 18/12/2017]
- [24] Wikipedia, *Arduino*, <https://en.wikipedia.org/>, [Online; Cập nhật lần cuối vào 31/05/2017]
- [25] Freetuts.net, *Javascript*, <https://freetuts.net/>, [Online; Cập nhật lần cuối vào 06/01/2015]

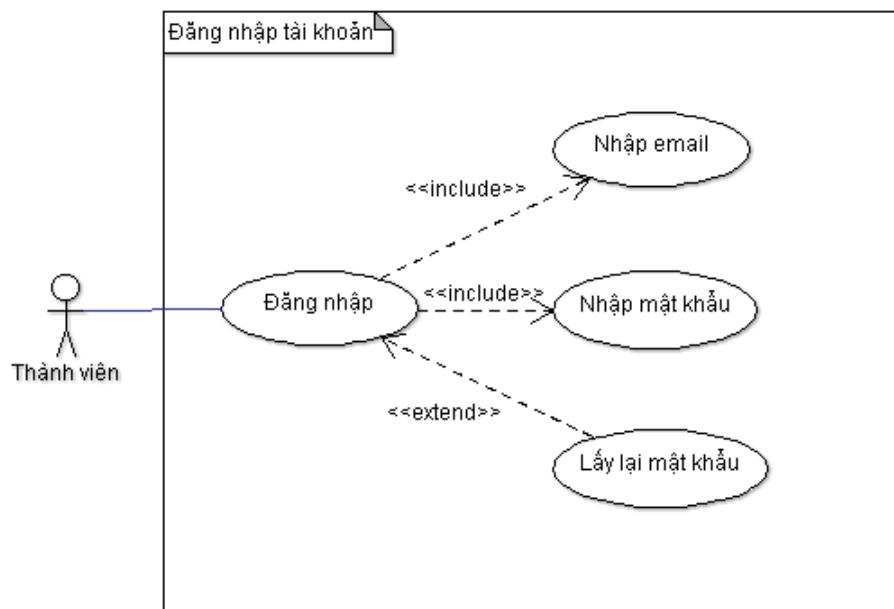
- [26] Arduino.vn, *Giao thức MQTT*, <http://arduino.vn/>, [Online; Cập nhật lần cuối vào 12/09/2016]
- [27] Amebaiot.com, *MQTT*, <https://www.amebaiot.com/>, [Online; Cập nhật lần cuối vào 2018]
- [28] hshop.vn, *dht22*, <http://hshop.vn/>, [Online; Cập nhật lần cuối vào 2017]
- [29] hshop.vn, *bh1750*, <http://hshop.vn/>, [Online; Cập nhật lần cuối vào 2017]
- [30] hshop.vn, *ds1307*, <http://hshop.vn/>, [Online; Cập nhật lần cuối vào 2017]
- [31] hshop.vn, *hshop*, <http://hshop.vn/>, [Online; Cập nhật lần cuối vào 2017]
- [32] dientuachau.com, *dientuachau*, <http://www.dientuachau.com/>, [Online; Cập nhật lần cuối vào 2017]
- [33] PPM, *PPM*, <http://www.octiva.net/projects/ppm/>
- [34] vlabs.iitb.ac.in, *Wien Bridge Oscillator*, <http://vlabs.iitb.ac.in/>
- [35] instructables.com, *AC to DC Converter*, <http://www.instructables.com/>
- [36] pinterest.se, *TL074*, <https://www.pinterest.se/>
- [37] techtalk.vn, *Thiết kế RESTful APIs*, <https://techtalk.vn/>, [Online; Cập nhật lần cuối vào 07/02/2017]

A PHỤ LỤC A: CHI TIẾT CÁC USE CASE

Chi tiết các use case và các mô tả use case cho các chức năng của khách và thành viên:



Hình 143: Đăng ký.



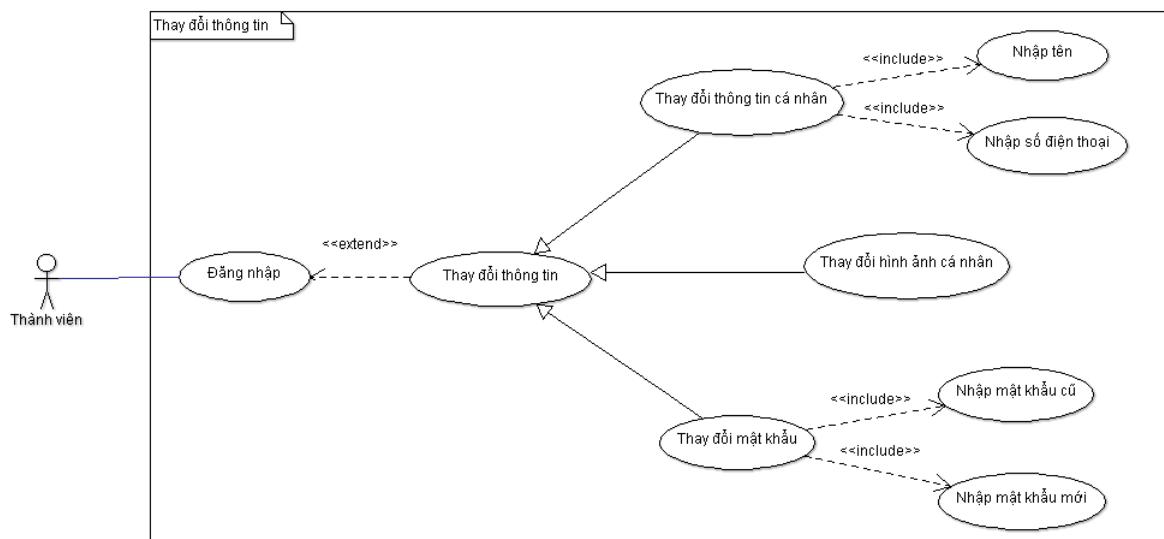
Hình 144: Đăng nhập.

ID	Name	Created by	Created at
UC_REG_001	Dăng ký tài khoản	Tạ Chí Tây	20/3/2017
Actor: Khách			
Mô tả ngắn gọn: khách có thẻ đăng ký thành viên			
Kịch bản chính:			
1. Người dùng khởi động khung đăng ký bằng cách nhấn nút Register 2. Người dùng nhập các thông tin cần thiết trong khung đăng ký và nhấn Register. 3. Hệ thống tạo người dùng mới 4. Hệ thống thông báo người dùng đã đăng ký thành công.			
Kịch bản phụ:			
2.1. Các ô thông tin bị đe dọa trống hoặc nhập sai định dạng, mật khẩu không khớp, email đã đăng ký thì hệ thống sẽ thông báo.			
Các quan hệ khác:			
<ul style="list-style-type: none"> - Include: nhập họ tên - Include: nhập email - Include: nhập mật khẩu - Include: nhập số điện thoại 			

Bảng 21: Mô tả use case đăng ký.

ID	Name	Created by	Created at
UC_LOGIN_001	Dăng nhập	Tạ Chí Tây	20/3/2017
Actor: Khách			
Mô tả ngắn gọn: khách có thẻ đăng nhập vào hệ thống			
Tiền điều kiện: người dùng đã đăng ký thành công			
Kịch bản chính:			
1. Người dùng khởi động khung đăng nhập bằng cách nhấn vào nút Login. 2. Người dùng nhập email và mật khẩu vào các ô tương ứng và nhấn nút Sign In. 3. Hệ thống xác thực người dùng thông qua email và mật khẩu. 4. Hệ thống thông báo người dùng đã đăng nhập thành công.			
Kịch bản phụ:			
2.1 Tên truy cập không tồn tại trong cơ sở dữ liệu hoặc mật khẩu không chính xác. Hệ thống hiển thị lại biểu mẫu Đăng nhập và hiển thị thông báo lỗi trên màn hình và yêu cầu nhập lại. 5. Nhấn vào nút "Quên mật khẩu". Hệ thống sẽ gửi đường link reset mật khẩu đến email của người dùng và thông báo cho người dùng biết.			
Các quan hệ khác:			
<ul style="list-style-type: none"> - Include: nhập email - Include: nhập mật khẩu 			

Bảng 22: Mô tả use case đăng nhập.



Hình 145: Use case thay đổi thông tin.

ID	Name	Created by	Created at
UC_CHANGEINFO_001	Thay đổi thông tin cá nhân	Tạ Chí Tâm	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng nhập tên và số điện thoại mình muốn thay đổi để cập nhật thông tin			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi thông tin bằng cách nhấn vào nút Edit. 2. Người dùng nhập tên và số điện thoại mới vào các ô tương ứng và nhấn nút OK. 3. Hệ thống xác thực người dùng và cập nhật thông tin. 4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng để các ô nhập rỗng, hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên - Include: nhập số điện thoại			

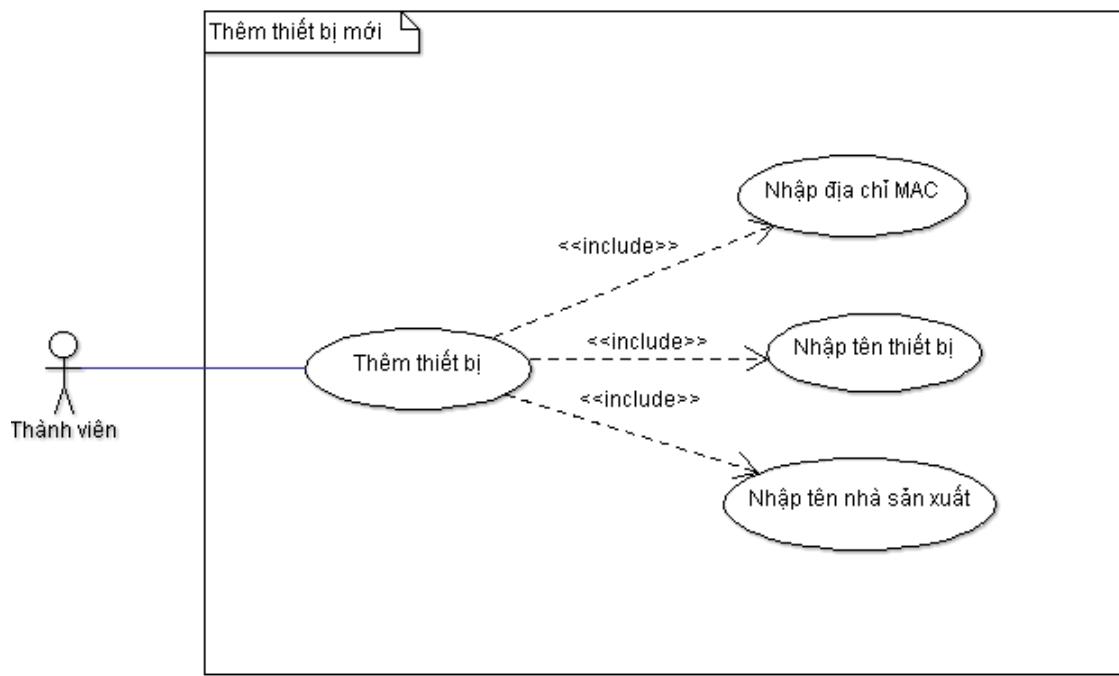
Bảng 23: Mô tả use case thay đổi thông tin cá nhân.

ID	Name	Created by	Created at
UC_CHANGEINFO_002	Thay đổi ảnh cá nhân	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thay đổi ảnh đại diện của mình			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi ảnh cá nhân bằng cách nhấn chọn "Change avatar".			
2. Người dùng chọn đường dẫn đến tập tin ảnh mới trên máy và nhấn OK.			
3. Hệ thống xác thực người dùng và cập nhật thông tin.			
4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng không chọn đường dẫn thì hệ thống sẽ thông báo lỗi.			

Bảng 24: Use case thay đổi hình ảnh cá nhân.

ID	Name	Created by	Created at
UC_CHANGEINFO_003	Thay đổi mật khẩu	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thay đổi mật khẩu của mình			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi mật khẩu bằng cách nhấn nút "Change password".			
2. Người dùng nhập thông tin vào các ô tương ứng và nhấn OK.			
3. Hệ thống xác thực người dùng và cập nhật thông tin.			
4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập sai mật khẩu cũ, nhập mật khẩu mới không trùng nhau hoặc để trống các ô thì hệ thống sẽ thông báo.			
Các ràng buộc: thông tin hiển thị dưới dạng ký tự *.			
Các quan hệ khác:			
- Include: nhập mật khẩu cũ			
- Include: nhập mật khẩu mới			

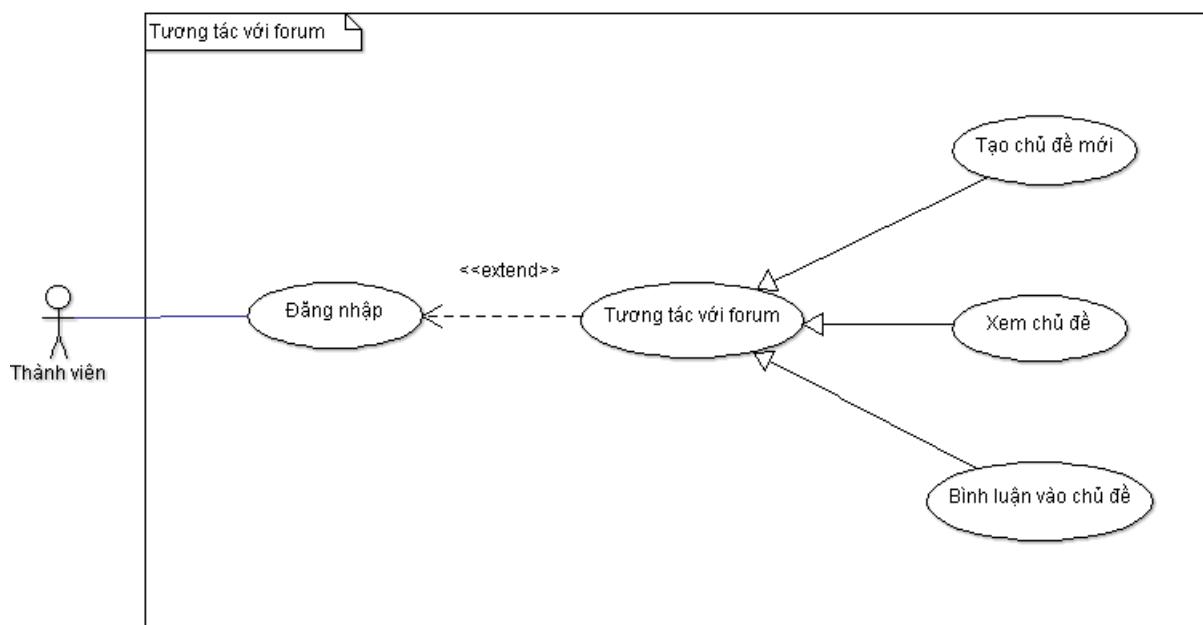
Bảng 25: Mô tả use case thay đổi mật khẩu.



Hình 146: Use case thêm thiết bị.

ID	Name	Created by	Created at
UC_ADDDEVICE_001	Thêm board điều khiển	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thêm board điều khiển mới.			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng nhấn vào nút thêm thiết bị mới trong trang quản lý board điều khiển. 2. Người dùng nhập thông tin vào các ô tương ứng và nhấn OK. 3. Hệ thống xác thực người dùng và thêm board điều khiển mới. 4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Địa chỉ MAC nhập không đúng định dạng, các ô nhập để trống hệ thống sẽ báo lỗi.			
Các ràng buộc: địa chỉ MAC phải đúng định dạng.			
Các quan hệ khác:			
- Include: nhập MAC - Include: nhập tên board điều khiển - Include: nhập tên nhà sản xuất			

Bảng 26: Mô tả use case thêm thiết bị.



Hình 147: Use case tương tác với diễn đàn.

ID	Name	Created by	Created at
UC_FORUM_001	Tạo bài viết mới	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: thành viên tạo một bài viết mới.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở mục Bài viết			
Kịch bản chính:			
1. Người dùng khởi động khung tạo bài viết mới bằng cách nhấn nút New article. 2. Người dùng nhập tiêu đề của bài viết, nội dung cụ thể và nhấn OK. 3. Hệ thống xác thực người dùng và tạo bài viết mới. 4. Hệ thống thông báo người dùng đã tạo thành công.			
Kịch bản phụ:			
2.1 Các ô thông tin bị để trống thì hệ thống sẽ thông báo lỗi.			

Bảng 27: Mô tả use case tạo bài viết mới.

ID UC_FORUM_002	Name Xóa bài viết	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Mod			
Mô tả ngắn gọn: mod thực hiện xóa bài viết.			
Tiền đề kiện: người dùng đã đăng nhập thành công và có quyền là mod			
Kịch bản chính:			
1. Người dùng nhấn vào nút xóa bài viết trên danh sách các bài viết. 2. Hệ thống hỏi lại người dùng có chắc chắn hay không. 3. Hệ thống xóa bài đăng nếu người dùng nhấn OK. 4. Hệ thống thông báo người dùng đã xóa thành công.			

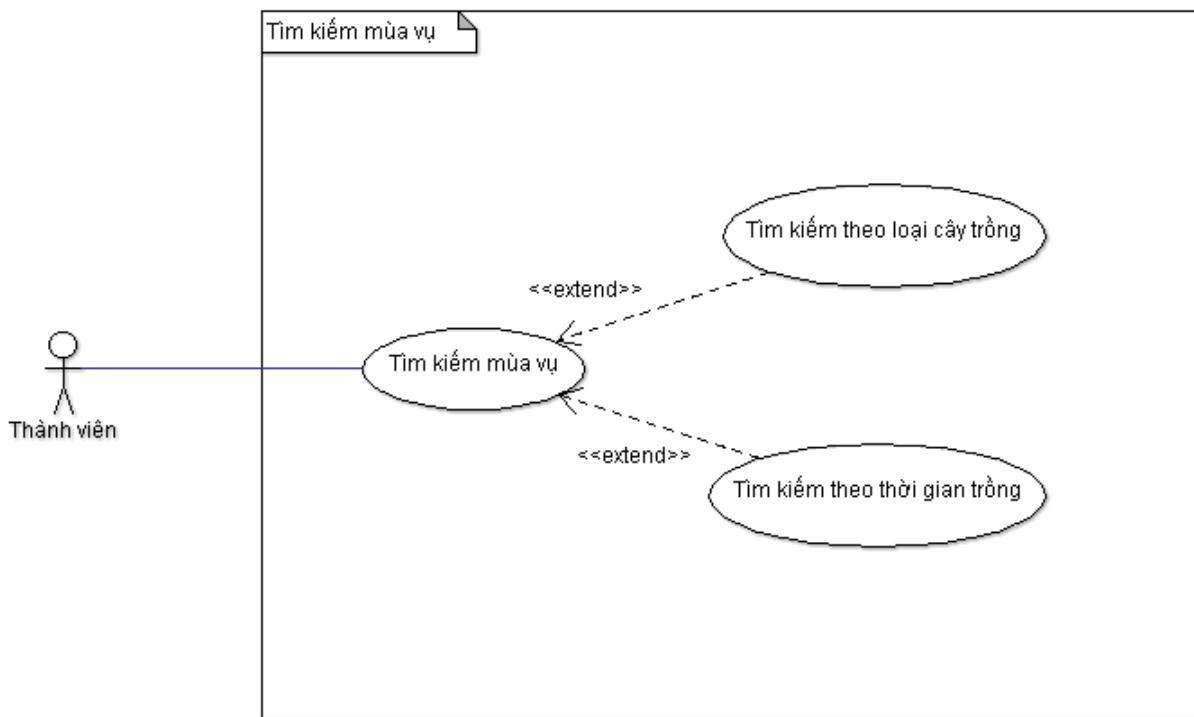
Bảng 28: Mô tả use case xóa bài viết.

ID UC_FORUM_003	Name Bình luận bài viết	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: thành viên tạo một bình luận mới trong bài viết.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang xem bài viết.			
Kịch bản chính:			
1. Người dùng nhấn chọn vào nút bên luận trên bài đăng đang xem. 2. Người dùng nhập nội dung bình luận của mình vào bài đăng và nhấn OK. 3. Hệ thống xác thực người dùng và tạo bình luận mới.			
Kịch bản phụ:			
2.1 Các ô thông tin bị để trống thì hệ thống sẽ thông báo lỗi.			

Bảng 29: Mô tả use case thêm bình luận bài viết.

ID UC_REMOVEDEVICE_001	Name Xóa board điều khiển	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa board điều khiển của mình.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị các board điều khiển.			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một board điều khiển trong danh sách. 2. Hệ thống hiển thị thông báo xác nhận xóa board điều khiển của người dùng. 3. Người dùng chọn OK để xóa đi board điều khiển muốn xóa.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy xóa board điều khiển.			

Bảng 30: Mô tả use case xóa thiết bị.



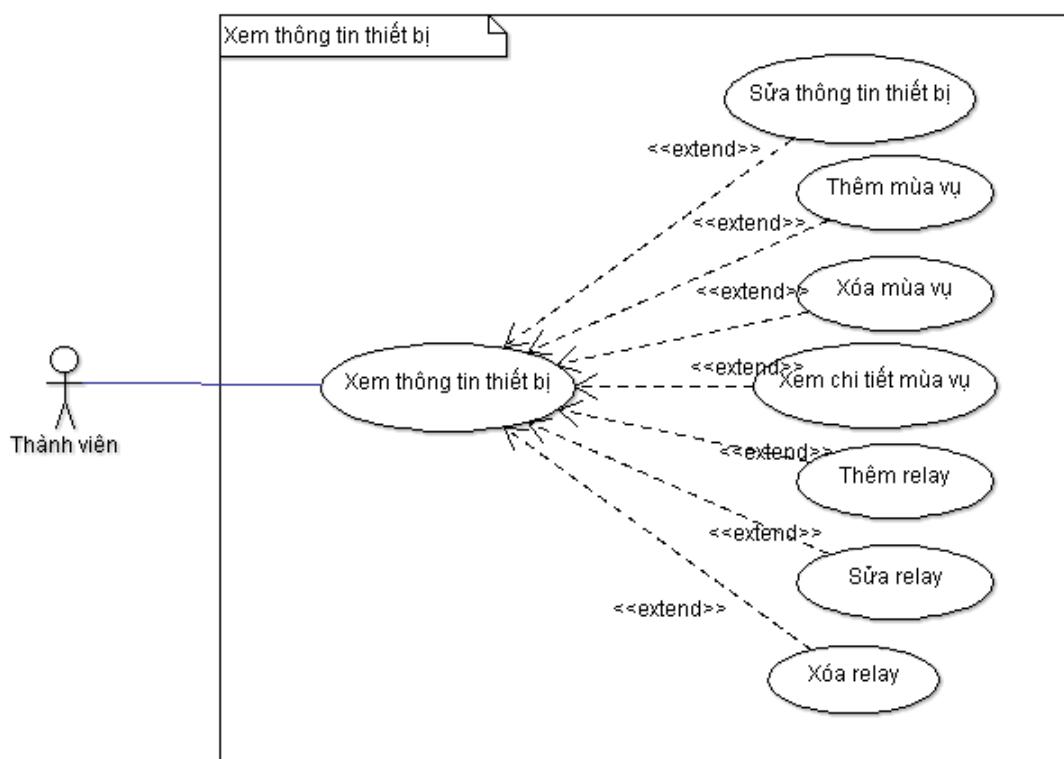
Hình 148: Use case tìm kiếm mùa vụ.

ID	Name	Created by	Created at
UC_FIND_001	Tìm kiếm mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thực hiện Tìm kiếm mùa vụ để xem thông tin mùa được chia sẻ.			
Tiền điều kiện: người dùng đã đăng nhập thành công.			
Kịch bản chính:			
1. Người dùng nhấn vào nút tìm kiếm trong trang của mình sau khi đã đăng nhập.			
2. Người dùng nhập vào thông tin của mùa vụ cần tìm kiếm: tìm kiếm theo tên loại cây trồng, tìm kiếm theo tháng...			
3. Hệ thống thực hiện tìm kiếm các mùa vụ thỏa điều kiện tìm kiếm của người dùng và hiển thị trong trang kết quả tìm kiếm.			
4. Người sử dụng nhấn chọn vào mục mong muốn để xem thông tin của mùa vụ.			
Kịch bản phụ:			
3.1 Hệ thống không hiển thị kết quả tìm kiếm nào phù hợp với yêu cầu của người dùng.			

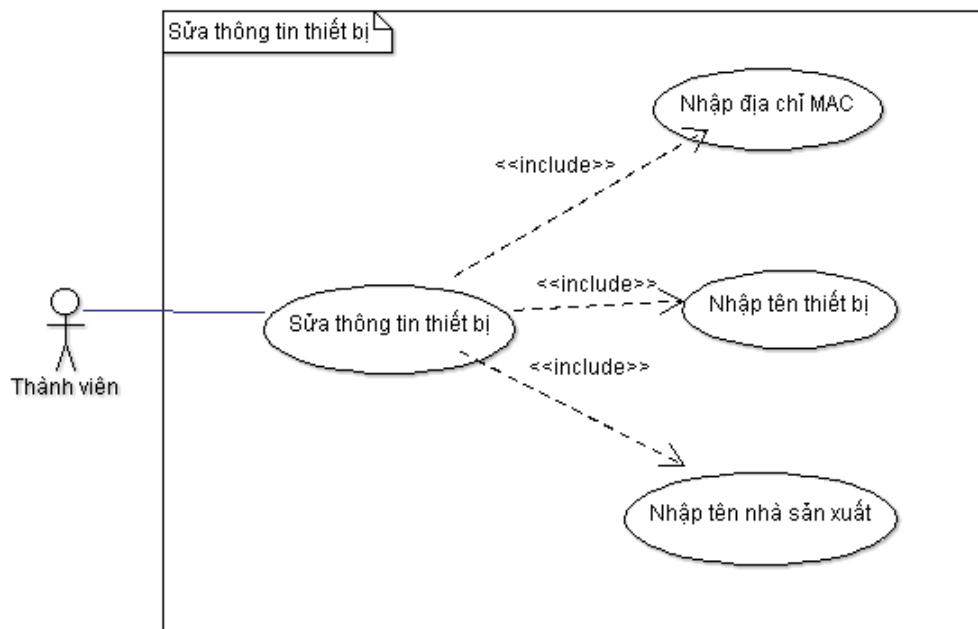
Bảng 31: Mô tả use case tìm kiếm mùa vụ.

ID	Name	Created by	Created at
UC_FIND_002	Gửi bình luận mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thực hiện bình luận mùa vụ của người khác.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang xem thông tin mùa vụ của người khác chia sẻ.			
Kịch bản chính:			
1. Người dùng nhập bình luận cho mùa vụ. 2. Người dùng chọn điểm đánh giá cho mùa vụ. 3. Người dùng nhấn nút để gửi bình luận lên server. 4. Hệ thống hiển thị thông báo người dùng gửi bình luận thành công.			
Kịch bản phụ:			
4.1 Hệ thống hiển thị thông báo người dùng gửi không thành công do có lỗi.			

Bảng 32: Mô tả use case gửi bình luận mùa vụ.



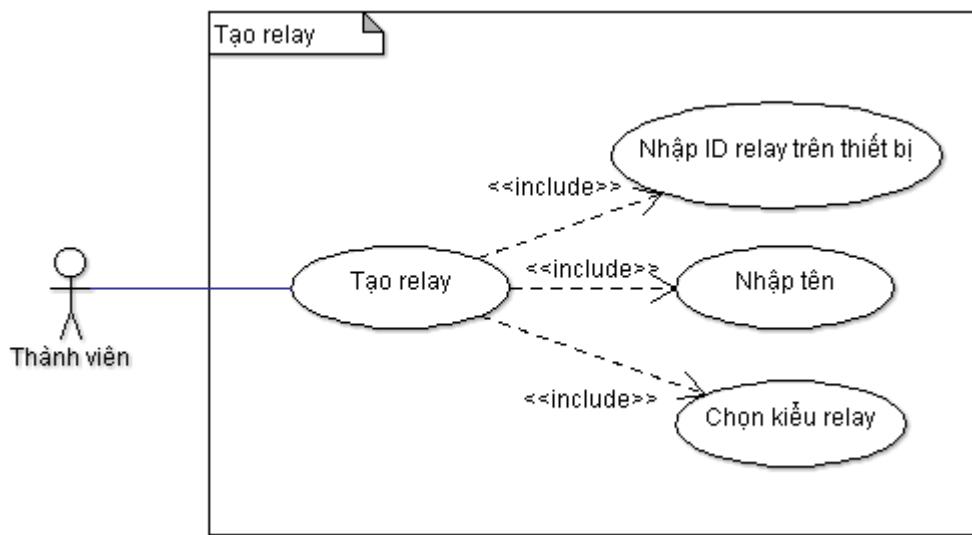
Hình 149: Use case xem thông tin thiết bị.



Hình 150: Use case sửa thông tin thiết bị.

ID	Name	Created by	Created at
UC_DEVICE_001	Sửa thông tin board điều khiển	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thực hiện chỉnh sửa thông tin của một board điều khiển của mình.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở trang board điều khiển.			
Kịch bản chính:			
1. Người dùng nhấn vào nút chỉnh sửa thông tin board điều khiển. 2. Người dùng nhập vào các thông tin cập nhật mới về board điều khiển của mình và nhấn OK. 3. Hệ thống thực hiện cập nhật thông tin của board điều khiển mới. 4. Hệ thống thông báo người dùng đã cập nhật thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
<ul style="list-style-type: none"> - Include: nhập MAC - Include: nhập tên thiết bị - Include: nhập tên nhà sản xuất 			

Bảng 33: Mô tả use case sửa thông tin thiết bị.



Hình 151: Use case tạo relay.

ID UC_RELAY_001	Name Tạo relay	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thêm relay mới vào hệ thống.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin thiết bị.			
Kịch bản chính:			
1. Người dùng nhấn vào nút tạo mới relay trên trang thông tin của thiết bị. 2. Người dùng nhập các dữ liệu cần thiết: ID của relay trên thiết bị, tên, loại relay. 3. Hệ thống tạo một relay mới. 4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

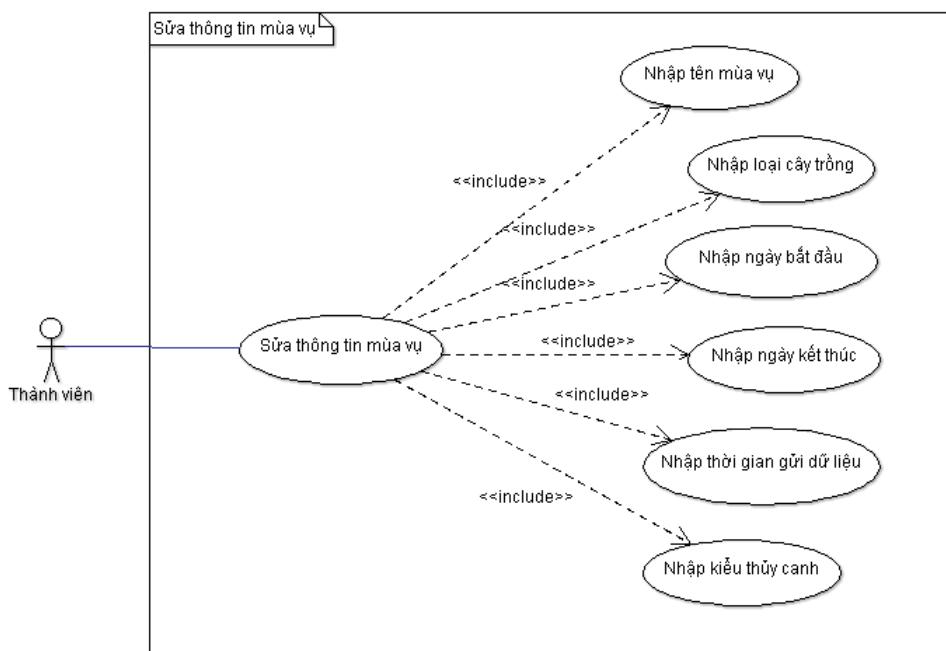
Bảng 34: Mô tả use case tạo relay.

ID	Name	Created by	Created at
UC_RELAY_002	Xóa relay	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa relay trên thiết bị.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị thông tin thiết bị (có danh sách các relay).			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một relay trong danh sách. 2. Hệ thống hiển thị thông báo xác nhận xóa relay của người dùng. 3. Người dùng chọn OK để xóa đi relay muốn xóa.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy xóa relay.			

Bảng 35: Mô tả use case xóa relay.

ID	Name	Created by	Created at
UC_RELAY_003	Sửa relay	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng sửa trạng thái của relay trên thiết bị.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị thông tin thiết bị (có danh sách các relay).			
Kịch bản chính:			
1. Người dùng nhấn nút chuyển trạng thái của relay hoặc nhấn nút chuyển ưu tiên relay thành Primary/Secondary. 2. Hệ thống hiển thị thông báo xác nhận thay đổi trên relay của người dùng. 3. Người dùng chọn OK để xác nhận sự thay đổi muốn thực hiện.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy thao tác thay đổi relay.			

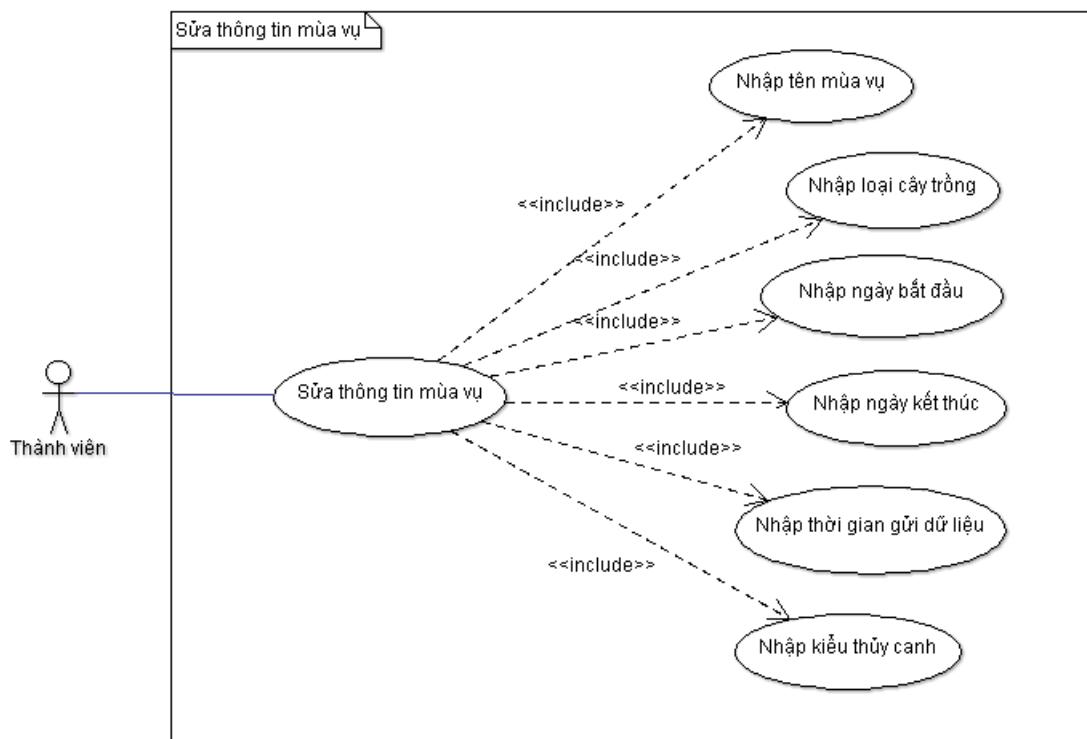
Bảng 36: Mô tả use case sửa relay.



Hình 152: Use case thêm mùa vụ mới.

ID UC_CROP_001	Name Tạo mùa vụ mới	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng tạo thêm mùa vụ mới cho thiết bị.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin của thiết bị.			
Kịch bản chính:			
1. Người dùng khởi động khung thêm thiết bị mới bằng cách nhấn vào nút thêm thiết bị mới trong trang thông tin thiết bị của mình. 2. Người dùng nhập thông tin của mùa vụ mới vào và nhấn OK. 3. Hệ thống thực hiện lưu thông tin mùa vụ mới vào cơ sở dữ liệu. 4. Hệ thống thông báo người dùng đã thực hiện thêm mùa vụ mới vào thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên mùa vụ. / - Include: nhập loại cây trồng. / - Include: nhập ngày bắt đầu. / - Include: nhập ngày kết thúc. / - Include: nhập thời gian gửi dữ liệu. / - Include: nhập kiểu thủy canh.			

Bảng 37: Mô tả use case tạo mùa vụ mới.



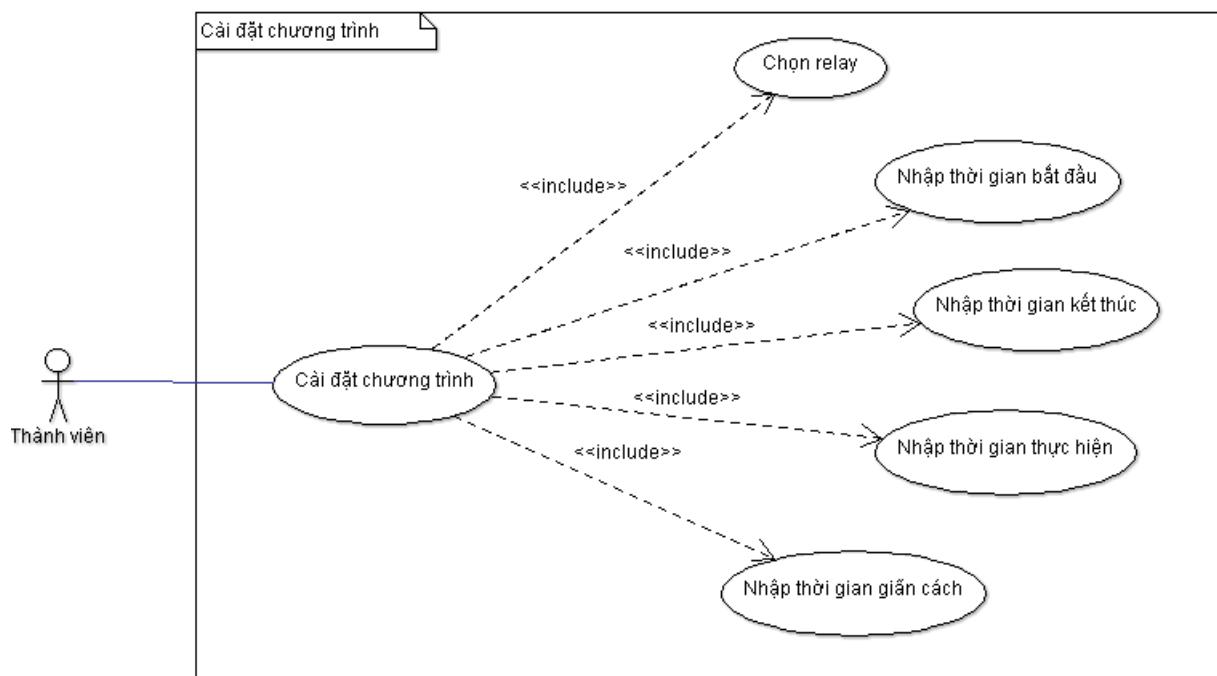
Hình 153: Use case sửa thông tin mùa vụ.

ID	Name	Created by	Created at
UC_CROP_003	Sửa thông tin mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng sửa thông tin mùa vụ hiện có.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng khởi động khung chỉnh sửa thông tin mùa vụ bằng cách nhấp vào nút chỉnh sửa.			
2. Người dùng nhập thông tin cập nhật mới của mùa vụ và nhấp nút cập nhập.			
3. Hệ thống cập nhật thông tin mùa vụ mới đã được người dùng nhập.			
4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên mùa vụ. / - Include: nhập loại cây trồng. / - Include: nhập ngày bắt đầu. / - Include: nhập ngày kết thúc. / - Include: nhập thời gian gửi dữ liệu. / - Include: nhập kiểu thủy canh			

Bảng 38: Mô tả use case sửa thông tin mùa vụ.

ID	Name	Created by	Created at
UC_CROP_002	Xóa mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa mùa vụ của mình.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin của thiết bị.			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một mùa vụ trong danh sách các mùa vụ đã có. 2. Hệ thống hiển thị thông báo xác nhận xóa mùa vụ của người dùng. 3. Người dùng chọn OK để xóa mùa vụ. 4. Hệ thống xóa mùa vụ được chọn. 5. Hệ thống thông báo xóa thành công.			
Kịch bản phụ:			
3.1 Người dùng chọn hủy để hủy xóa mùa vụ.			

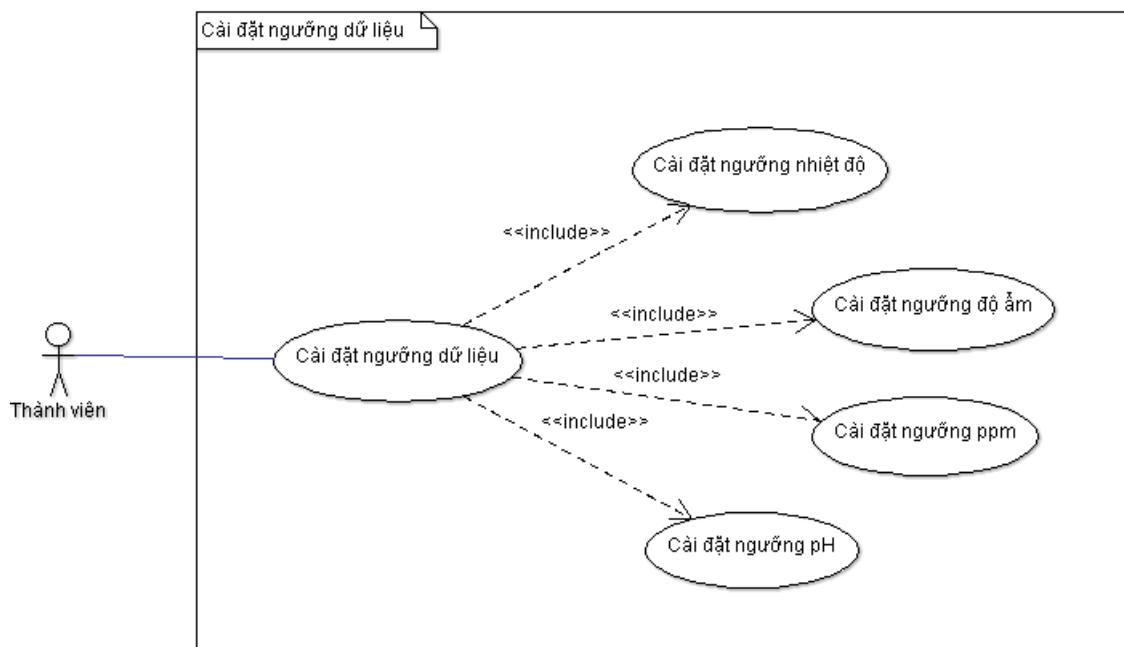
Bảng 39: Mô tả use case xóa mùa vụ.



Hình 154: Use case cài đặt chương trình mùa vụ.

ID	Name	Created by	Created at
UC_CROP_004	Cài đặt chương trình	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng cài đặt lịch trình cho mùa vụ.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng mở khung cài đặt bằng cách nhấn vào nút Edit bên lịch trình. 2. Người dùng nhập các dữ liệu cần thiết. 3. Hệ thống tạo một lịch trình mới, lưu vào database và gửi xuống thiết bị. 4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

Bảng 40: Mô tả use case cài đặt chương trình cho mùa vụ.



Hình 155: Use case cài đặt ngưỡng dữ liệu.

A PHỤ LỤC A: CHI TIẾT CÁC USE CASE

ID	Name	Created by	Created at
UC_CROP_005	Cài đặt ngưỡng dữ liệu	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng cài đặt ngưỡng cảnh báo dữ liệu.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng nhấn vào nút Edit bên ô hiển thị ngưỡng để mở ô cài đặt.			
2. Người dùng nhập các dữ liệu cần thiết.			
3. Hệ thống tạo một ngưỡng mới.			
4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

Bảng 41: Mô tả use case cài đặt ngưỡng dữ liệu.

B PHỤ LỤC B: CHI TIẾT CÁC API

- Người dùng (User)

- Lấy thông tin người dùng:

- * Phương thức: GET
 - * URL: /user/info
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên: { user.id: 12 }
 - * Dữ liệu trả về (*): thông tin của người dùng (không chứa mật khẩu và token kích hoạt).

- Kiểm tra email mới có hợp lệ không:

- * Phương thức: POST
 - * URL: /user/checkemail
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên: { email: "email" }

- Đăng ký người dùng:

- * Phương thức: POST
 - * URL: /user/register
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
{ name: "name",
password: "123",
email: "email",
phone: "12345" }

- Đăng nhập:

- * Phương thức: POST
- * URL: /user/login
- * Yêu cầu đăng nhập: không
- * Quyền truy cập: không yêu cầu
- * Dữ liệu gửi lên:
{ password: "123",
email: "email", }
- Cập nhật thông tin người dùng:
 - * Phương thức: PUT
 - * URL: /user/update
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ name: "name",
email: "email",
phone: "1234567" }
- Thay đổi mật khẩu người dùng:
 - * Phương thức: PUT
 - * URL: /user/changepass
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ currPass: "123",
newPass: "567" }
- Reset mật khẩu người dùng:
 - * Phương thức: POST
 - * URL: /user/resetpass
 - * Yêu cầu đăng nhập: có

- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ email: "email" }
- Kích hoạt tài khoản người dùng:
 - * Phương thức: PUT
 - * URL: /user/active
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ email: "email",
activeToken: "abc" }
- Chứng thực JWT token của người dùng:
 - * Phương thức: GET
 - * URL: /user/verifytoken
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
{ token: "abc" }
- Lấy danh sách tất cả người dùng:
 - * Phương thức: GET
 - * URL: /user/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: admin
 - * Dữ liệu trả về: mảng các đối tượng người dùng
- Xóa người dùng:
 - * Phương thức: DELETE
 - * URL: /user/delete
 - * Yêu cầu đăng nhập: có

- * Quyền truy cập: admin
- * Dữ liệu gửi lên:
{ userId: 12 }
- Lấy thông tin người dùng:
 - * Phương thức: GET
 - * URL: /user/detail
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ userId: 12 }
 - * Dữ liệu trả về: tất cả thông tin của người dùng (bao gồm mật khẩu và token kích hoạt).
- Cập nhật role của người dùng:
 - * Phương thức: PUT
 - * URL: /user/detail
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: admin
 - * Dữ liệu gửi lên:
{ userId: 12,
newRole: "mod" }
- Board điều khiển (Device):
 - Lấy tất cả board điều khiển:
 - * Phương thức: GET
 - * URL: /device/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ user.id: 12 }

- * Dữ liệu trả về: mảng các đối tượng board điều khiển.
- Lấy thông tin của các board đang chạy:
 - * Phương thức: GET
 - * URL: /device/running
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { user.id: 12 }
 - * Dữ liệu trả về: mảng các đối tượng board điều khiển đang chạy..
- Cập nhật trạng thái board điều khiển:
 - * Phương thức: PUT
 - * URL: /device/status
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { mac: "112233445566",
 - newStatus: "running" }
- Lấy thông tin của một board điều khiển:
 - * Phương thức: GET
 - * URL: /device/one
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { mac: "112233445566", }
 - * Dữ liệu trả về: một đối tượng board điều khiển
- Thêm board điều khiển mới:
 - * Phương thức: POST
 - * URL: /device/add

- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:

```
{ mac: "112233445566",
  name: "device 1",
  manufacture: "sony",
  status: "no connection", }
```
- Thêm board điều khiển mới:
 - * Phương thức: POST
 - * URL: /device/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ mac: "112233445566",
  name: "device 1",
  manufacture: "sony",
  status: "no connection", }
```
- Xóa board điều khiển:
 - * Phương thức: DELETE
 - * URL: /device/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ mac: "112233445566" }
```
- Lịch trình (schedule)
 - Lấy thông tin tất cả lịch trình hiện tại của một mùa vụ:
 - * Phương thức: GET
 - * URL: /schedule/all

- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ cropId: 12 }
- * Dữ liệu trả về: mảng các đối tượng lịch trình (schedule)
- Xóa một lịch trình:
 - * Phương thức: DELETE
 - * URL: /schedule/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,
scheduleId: 12 }
- Đồng bộ lịch trình hiện tại xuống thiết bị:
 - * Phương thức: GET
 - * URL: /schedule/sync
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,
mac: "112233445566" }
- Thêm vào một lịch trình cài đặt:
 - * Phương thức: POST
 - * URL: /schedule/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,

```
actuatorId: 12,  
starttime: 030303,  
endtime: 050505,  
intervaltime: 45,  
delaytime: 5 }
```

- Nguồn dữ liệu (threshold):

- Lấy nguồn dữ liệu mới nhất:
 - * Phương thức: GET
 - * URL: /threshold/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12 }
 - * Dữ liệu trả về: Nguồn dữ liệu cao và thấp mới nhất của mùa vụ hiện tại.
 - Thêm vào một nguồn dữ liệu mới:
 - * Phương thức: POST
 - * URL: /threshold/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ temperatureLower: 23,
temperatureUpper: 25,
humidityLower: 45,
humidityUpper: 50,
ppmLower: 100,
ppmUpper: 200,
lightLower: 1000,
lightUpper: 3000 }

- Actuator (máy bơm, quạt, đèn, ...)
 - Thêm một actuator mới cho board
 - * Phương thức: POST
 - * URL: /actuator/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ actuator: { status: "off",  
idOnBoard: 12,  
name: "water 1",  
type: "water",  
priority: "Primary" },  
deviceMac: "112233445566" }
```
 - Lấy tất cả các actuator trên board:
 - * Phương thức: GET
 - * URL: /actuator/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ deviceMac: "112233445566" }
```
 - * Dữ liệu trả về: mảng các đối tượng actuator
 - Thay đổi trạng thái của actuator:
 - * Phương thức: PUT
 - * URL: /actuator/status
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ Id: 12,  
Status: "on" }
```

- Thay đổi độ ưu tiên của một actuator:
 - * Phương thức: PUT
 - * URL: /actuator/priority
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { Id: 12,
 - priority: "Secondary" }
- Xóa một actuator:
 - * Phương thức: DELETE
 - * URL: /actuator/priority
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { Id: 12 }
- Bài viết (article)
 - Lấy tất cả các bài viết:
 - * Phương thức: GET
 - * URL: /article/all
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu trả về: mảng các đối tượng article
 - Lấy một bài viết theo ID:
 - * Phương thức: GET
 - * URL: /article/one
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu

- * Dữ liệu trả về: đối tượng article
- Thêm một bài viết mới:
 - * Phương thức: POST
 - * URL: /article/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ content: "content",
  title: "title" }
```
- Xóa một bài viết:
 - * Phương thức: DELETE
 - * URL: /article/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:

```
{ id: 12 }
```
- Kiểm duyệt bài viết:
 - * Phương thức: PUT
 - * URL: /article/check
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:

```
{ id: 12,
  checked: true }
```
- Bình luận (comment)
 - Thêm một bình luận cho bài viết:
 - * Phương thức: POST

- * URL: /comment/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { content: "content", ArticleId: 12 }
- Lấy tất cả các comment trong bài viết:
 - * Phương thức: GET
 - * URL: /comment/all
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
 - { ArticleId: 12 }
 - * Dữ liệu trả về: mảng các đối tượng comment
- Xóa một comment:
 - * Phương thức: DELETE
 - * URL: /comment/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:
 - { commentId : 12 }
- Mùa vụ (crop)
 - Lấy tất cả các mùa vụ của một board:
 - * Phương thức: GET
 - * URL: /crop/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member

- * Dữ liệu gửi lên:
{ mac: "112233445566" }
- Lấy một mùa vụ bằng ID:
 - * Phương thức: GET
 - * URL: /crop/one
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12 }
 - * Dữ liệu trả về: một đối tượng crop
- Lấy mùa vụ mới nhất đang chạy trên board:
 - * Phương thức: GET
 - * URL: /crop/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ mac: "112233445566" }
 - * Dữ liệu trả về: một đối tượng crop
- Thêm một mùa vụ mới:
 - * Phương thức: POST
 - * URL: /crop/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ DeviceMac: "112233445566",
status: "pending",
name: "crop 1",
startdate: "12-20-2017 05:00:00",

```
closedate: "01-20-2017 05:00:00",
treetype: "salad",
reporttime: 60 }
```

– Xóa một mùa vụ:

- * Phương thức: DELETE
- * URL: /crop/delete
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ cropId: 12 }

– Thay đổi thông tin mùa vụ:

- * Phương thức: PUT
- * URL: /crop/edit
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ name: "crop 1",
startdate: "12-20-2017 05:00:00",
closedate: "01-20-2017 05:00:00",
treetype: "salad",
reporttime: 60 }

– Thay đổi chế độ chia sẻ:

- * Phương thức: PUT
- * URL: /crop/share
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ Id: 12,
share: true }

- Tìm mùa vụ:
 - * Phương thức: GET
 - * URL: /crop/search
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:

```
{ tree: "salad",  
month: 12 }
```
- Dữ liệu cảm biến (data):
 - Lấy tất cả dữ liệu từ board gửi lên :
 - * Phương thức: GET
 - * URL: /data/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ cropId: 12 }
```
 - * Dữ liệu trả về: mảng các đối tượng data.
 - Lấy dữ liệu mới nhất:
 - * Phương thức: GET
 - * URL: /data/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ cropId: 12 }
```
 - * Dữ liệu trả về: đối tượng data.