**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công Nghệ Thông Tin và Truyền Thông**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**BÁO CÁO BÀI TẬP LỚN**

**Môn học: Công nghệ truyền thông IoT**

***Đề tài*: Smarthome**

GVHD: TS. Nguyễn Đức Toàn

SVTH: Vũ Duy Trường - 20173427

Nguyễn Minh Hoàng - 20173134

Nguyễn Trường Giang - 20173078

**Hà Nội, Tháng 02 năm 2022**

**Lời** **nói đầu**

Nhà thông minh là một ứng dụng trong số các ứng dụng của IoT. Nhà thông minh (tiếng Anh: home automation, smart home hoặc Intellihome) là kiểu nhà được lắp đặt các thiết bị điện, điện tử có tác dụng tự động hoá hoàn toàn hoặc bán tự động, thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử này giao tiếp với chủ nhân nhà thông qua bảng điện tử đặt trong nhà, phần mềm điện thoại di động, máy tính bảng hoặc một giao diện web.

Trong căn nhà thông minh, đồ dùng trong nhà từ phòng ngủ, phòng khách đến toilet đều gắn các bộ điều khiển điện tử có thể kết nối với Internet và điện thoại di động, cho phép chủ nhân điều khiển vật dụng từ xa hoặc lập trình cho thiết bị ở nhà hoạt động theo lịch. Nhà thông minh ngoài ra còn có một số ứng dụng sáng tạo hơn, gồm hệ thống điều khiển giải trí tại gia – loa công suất khác nhau, hệ thống điện thoại, liên lạc nội bộ, hệ thống tưới nước... Các thành phần của hệ thống nhà thông minh bao gồm các cảm biến (như cảm biến nhiệt độ, cảm biến ánh sáng hoặc do cử chỉ), các bộ điều khiển hoặc máy chủ và các thiết bị chấp hành khác. Nhờ hệ thống cảm biến, các bộ điều khiển và máy chủ có thể theo dõi các trạng thái bên trong ngồi nhà để đưa ra các quyết định điều khiển các thiết bị chấp hành một cách phù hợp nhằm đảm bảo môi trường sống tốt nhất cho con người.

Ngoài ra, cùng với sự phát triển của các thiết bị điện tử cá nhân như máy tính bảng và điện thoại thông minh, cac hệ thống nhà thông minh còn cung cấp khả năng tương tác với người sử dụng thông qua các thiết bị điện tử cá nhân cho phép con người có thể giám sát và điều khiển ngôi nhà từ bất cứ đâu. Hiện nay, trong lĩnh vực nhà thông minh, các kỹ sư vẫn đang tiếp tục sáng tạo để tạo ra nhiều tiện ích hơn nữa cho ngôi nhà cũng như tối ưu hoá về việc triển khai lắp đặt một ngôi nhà thông minh.

Trong tương lai, ngôi nhà thông minh có thể có khả năng "tư duy" để tự điều chỉnh các thiết bị một cách phù hợp và còn có khả năng giao tiếp với con người như trong các bộ phim viễn tưởng.

MỤC LỤC

[I. Mở đầu 5](#_Toc95941702)

[1.1 Giới thiệu đề tài 5](#_Toc95941703)

[1.2 Mục tiêu đề tài 5](#_Toc95941704)

[1.3 Yêu cầu của đề tài 5](#_Toc95941705)

[II. Thiết kế phần cứng 6](#_Toc95941706)

[2.1 Module wifi ESP8266 6](#_Toc95941707)

[2.2 Cảm biến độ ẩm nhiệt độ DHT11 7](#_Toc95941708)

[2.3 Module thu phát hồng ngoại V1 7](#_Toc95941709)

[III. Thiết kế hệ thống 9](#_Toc95941710)

[3.1 Sơ đồ hệ thống 9](#_Toc95941711)

[3.2 Giao thức MQTT 10](#_Toc95941712)

[a, Mô hình publish/subscribe 11](#_Toc95941713)

[b, Qualities of Service 11](#_Toc95941714)

[c, Bảo mật 12](#_Toc95941715)

[3.3 Node.js 12](#_Toc95941716)

[a, Ưu điểm: 12](#_Toc95941717)

[c, Nhược điêm: 13](#_Toc95941718)

[3.4 MongoDB 13](#_Toc95941719)

[a, NoSql 13](#_Toc95941720)

[b, Ưu điểm: 13](#_Toc95941721)

[c, Nhược điểm: 14](#_Toc95941722)

[VI. Phân tích thiết kế hệ thống 14](#_Toc95941723)

[4.1 Usecase tổng quan 14](#_Toc95941724)

[4.2 Phân giã các chức năng chính 15](#_Toc95941725)

[a, Quản lý tài khoản 15](#_Toc95941726)

[b, Quản lý điều khiển thiết bị 16](#_Toc95941727)

[c, Theo dõi thông số môi trường 17](#_Toc95941728)

[IV. Thiết kế giao diện 18](#_Toc95941729)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 21](#_Toc95941730)

[Kết luận 21](#_Toc95941731)

[Hướng phát triển 21](#_Toc95941732)

[Tài liệu tham khảo 23](#_Toc95941733)

[Hình 1. Sơ đồ ESP8266 5](#_Toc95941765)

[Hình 2. Cảm biến độ ẩm, nhiệt độ DHT11 6](#_Toc95941766)

[Hình 3 Module thu phát hồng ngoại V1 6](#_Toc95941767)

[Hình 4. Sơ đồ hệ thống 8](#_Toc95941768)

[Hình 5. Giao thức MQTT 9](#_Toc95941769)

[Hình 6. Qualities of service mqtt 10](#_Toc95941770)

[Hình 7 Màn hình đăng nhập 17](#_Toc95941771)

[Hình 8 Màn hình chọn phòng 17](#_Toc95941772)

[Hình 9 Màn hình chi tiết phòng 18](#_Toc95941773)

[Hình 10 Màn hình hẹn giờ 18](#_Toc95941774)

[Hình 11Màn hình điều khiển điều hoà 19](#_Toc95941775)

# I. Mở đầu

## 1.1 Giới thiệu đề tài

Nhà thông minh đang và sẽ là một xu hướng công nghệ trên thế giới, trở thành tiêu chuẩn của nhà ở hiện đại trong kỷ nguyên Internet of Things. Sự tiện nghi và tiết kiệm năng lượng cũng dẫn trở thành một tiêu chuẩn cần thiết nhằm nâng cao chất lượng sống của con người.

Nhà thông minh giúp cuộc sống con người trở nên tiện nghi và thoải mái hơn nhờ những cách sau:

* Các cánh cửa đều tự động đóng và mở khi chúng nhận diện được chủ nhà.
* Tự điều chỉnh độ sáng các bóng đèn khi có người đọc sách, xem tivi hay đi ngủ.
* Các thiết bị điện tử từ gia dụng đến giải trí trong ngôi nhà đều được điều khiển dễ dàng thông qua giọng nói hay điện thoại.
* Đưa ra các cảnh báo các vật dụng có thể gây nguy hiểm cho trẻ em khi chúng có ý định sử dụng.
* Tính năng an ninh như tự báo động khi có một người cố tình xâm nhạp trái phép, tự động khóa trái cửa, báo cảnh sát,… luôn trong tình trạng sẵn sàng.

## 1.2 Mục tiêu đề tài

Cuộc sống hiện đại, chúng ta thấy rằng mọi thứ xung quanh chúng ta điện thoại, tủ lạnh, máy hút bụi đều trở nên thông minh. Và gần đây chúng ta còn biết thêm với nhà thông minh. Hiện nay nhà thông minh gần được sử dụng phổ biến bởi những lợi ích nhà thông minh mang lại lợi ích cho cuộc sống con người.

Nhà thông minh giúp phục vụ mọi mục đích cho cuộc sống con người. Cuộc sống của con người trở nên tiện nghi với các thiết bị hiện đại, thông minh. Con người có thể tự do điểu chỉnh trong căn nhà mình theo ý thích của mình như: điều chỉnh nhiệt độ, điều chỉnh ánh sáng cũng như các hệ thống khác.

Đồng thời một căn nhà thông minh với hệ thống khóa cửa tự động cùng các hệ thống camera quan sát sẽ giúp các bạn an toàn hơn trong cuộc sống hàng ngày. Hay các bạn có thể yên tâm quan sát được ngôi nhà bạn khi bạn không có nhà

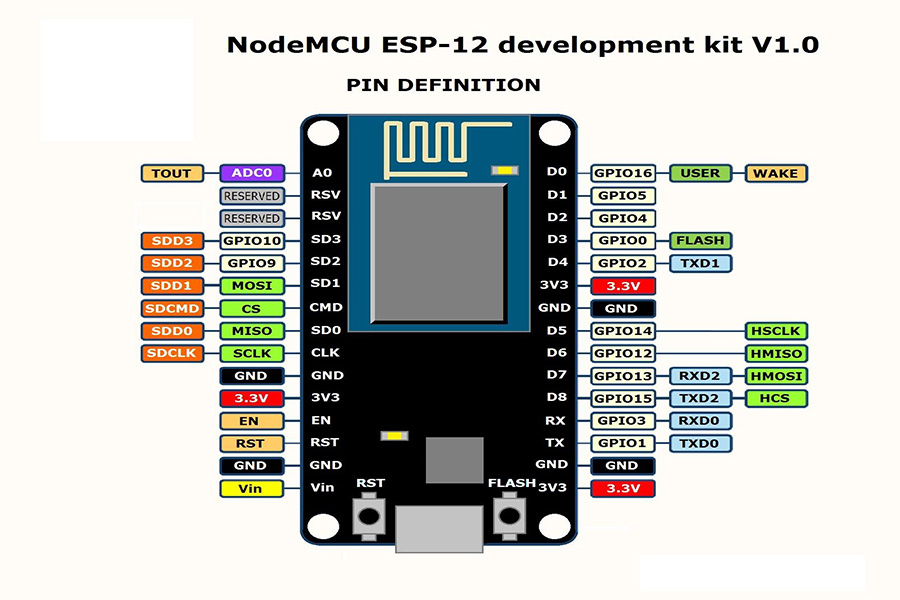
## 1.3 Yêu cầu của đề tài

* Hệ thống cảm biến hoạt động ổn định.
* Dữ liệu sẽ tự động lưu vào cơ sở dữ liệu.
* Phát triển một ứng dụng android để điểu khiển và theo dõi các thiết bị của hệ thống.

# II. Thiết kế phần cứng

## 2.1 Module wifi ESP8266

ESP8266 là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điệntử. ESP8266 được tích hợp sẵn Wifi, bộ nhớ flash 8Mb. ESP8266 có nhiều phiên bản và được đóng gói khác nhau nhưng khá giống nhau về chức năng cũng như khả năng lập trình. Hiện nay phổ biến nhất trên thị trường là ESP8266v7, ESP8266 12e, ESP8266 12f…



Hình 1. Sơ đồ ESP8266

Thông số kĩ thuật:

* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Điện áp hoạt động: 3.3V
* Điện áp vào: 5V thông qua cổng USB
* Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
* Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
* Bộ nhớ Flash: 4MB
* Giao tiếp: Cable Micro USB ( tương đương cáp sạc điện thoại )
* Hỗ trợ bảo mật: WPA/WPA2
* Tích hợp giao thức TCP/IP
* Lập trình trên các ngôn ngữ: C/C++, Micropython,

Mục đích sử dụng ESP8266 trong đề tài là để nhận dữ liệu từ các cảm biến, điểu khiển các thiết bị thông qua các chân, gửi và nhận dữ liệu lên cho server.

## 2.2 Cảm biến độ ẩm nhiệt độ DHT11

A circuit board

Description automatically generated

Hình 2. Cảm biến độ ẩm, nhiệt độ DHT11

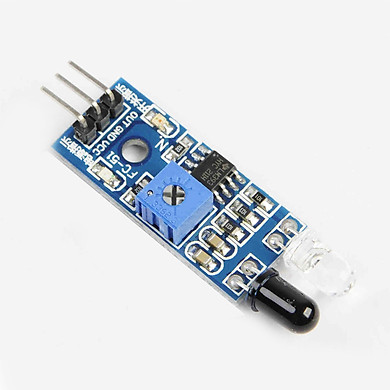
DHT11 Là cảm biến nhiệt độ, độ ẩm rất thông dụng hiện nay vì rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.

Thông số kỹ thuật:

* Điện áp hoạt động: 5VDC
* Chuẩn giao tiếp: TTL, 1 wire.
* Khoảng đo độ ẩm: 20%-80%RH sai số ± 5%RH
* Khoảng đo nhiệt độ: 0-50°C sai số ± 2°C
* Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
* Kích thước: 28mm x 12mm x10m

Mục đích sử dụng cảm biến độ ẩm trong đề tài là để đo độ ẩm trong phòng sau đó qua MQTT của ESP8266 gửi lên server.

## 2.3 Module thu phát hồng ngoại V1

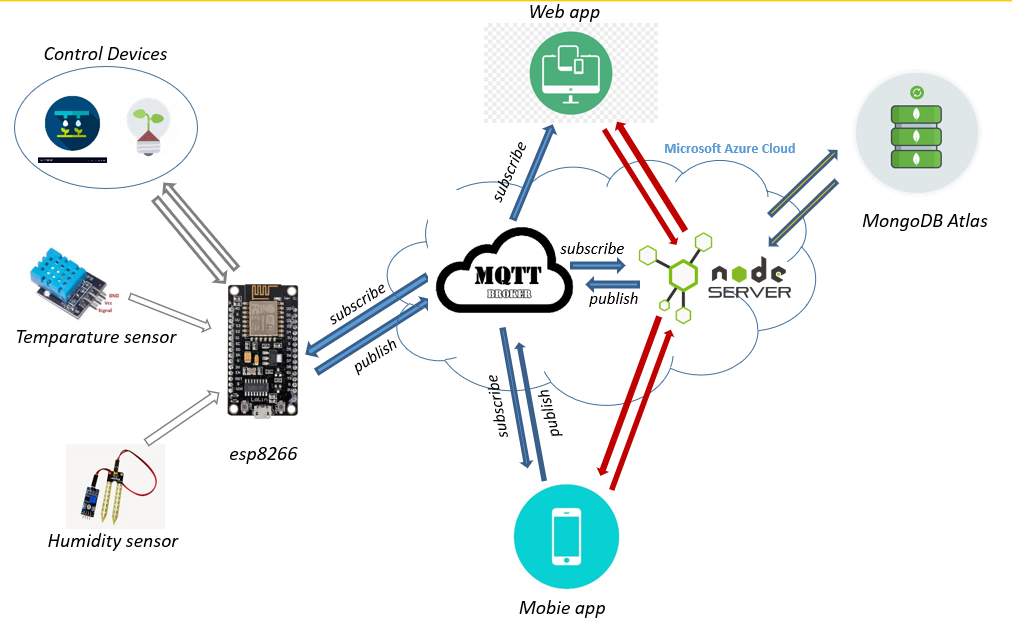


Hình 3 Module thu phát hồng ngoại V1

Module Thu Phát Hồng Ngoại V1 được tích hợp bộ phát hồng ngoại và bộ thu hồng ngoại. Bộ phát hồng ngoại là một diode phát sáng (LED) phát ra các tia hồng ngoại. Do đó, chúng được gọi là IR LED. Mặc dù LED IR trông giống như một đèn LED bình thường, bức xạ phát ra từ IR LED là sóng hồng ngoại nên con người không thể nhìn thấy bằng mắt thường được. Bộ thu hồng ngoại cũng được gọi là cảm biến hồng ngoại khi chúng phát hiện các tia từ bộ phát hồng ngoại. Bộ thu hồng ngoại có dạng photodiode và phototransistors. Photodiode hồng ngoại khác với điốt thông thường vì chúng chỉ phát hiện ra bức xạ hồng ngoại. Khi led phát hồng ngoại phát ra bức xạ, nó đến được vật thể và một số bức xạ phản xạ lại led thu hồng ngoại. Dựa trên cường độ thu của led thu hồng ngoại, đầu ra của cảm biến sẽ được xác định là mức cao hoặc thấp.

# III. Thiết kế hệ thống

## 3.1 Sơ đồ hệ thống



Hình 4. Sơ đồ hệ thống

Các tính năng của Mobile app:

* Đăng nhập, bảo mật xác thực bằng token
* Thống kê dữ liệu trong ngày bằng biểu đồ
* Xem dữ liệu thời gian thực như độ ẩm không khí, nhiệt độ môi trường
* Điều khiển thiết bị như khoá cửa, quạt, đèn, hẹn giờ tắt thiết bị

**Giải thích sơ đồ**

Tổng quan: esp8266 là thành phần có thể kết nối wifi và xử lý thu thập dữ liệu từ các sensor sau đó gửi dữ liệu lên MQTT broker thông qua giao thức mqtt.

Với tính năng thu thập dữ liệu, dữ liệu từ MQTT broker sẽ được chuyển đến NodeJS server và tiền xử lý rồi lưu trữ vào database

Với Web App có thể lấy dữ liệu từ database thông qua giao thức HTTP (Resful API) từ NodeJS server (ở web) và thống kê trên giao diện và web cũng có thể theo dõi dữ liệu thời gian thực thông qua việc subscribe MQTT broker, ngoài ra còn publish dữ liệu điều khiển tới MQTT broker để hẹn giờ điều khiển các thiết bị như khoá, đèn.

**Các thành phần sử dụng trên sơ đồ**

* Temperature Sensor: Cảm biến nhiệt độ
* Control Device: Thiết bị điều khiển gốm đèn, khoá, quạt điện
* Esp8266
* MQTT broker: Được triển khai trên máy ảo của Microsoft azure cloud thông qua Docker container
* Mobile App: phát triển bằng Java Android
* Nodejs server: phát triển bằng ExpressJS framework và deploy trên máy ảo của Microsoft azure cloud thông qua nginx
* MongoDB Atlas: Database cloud của mongodb

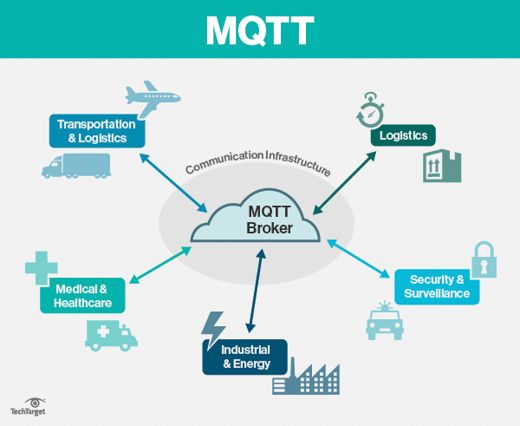
## 3.2 Giao thức MQTT

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

MQTT được thiết kế để liên lạc nhẹ giữa các thiết bị và hệ thống máy tính. MQTT được thiết kế ban đầu cho các mạng SCADA, các kịch bản sản xuất và băng thông thấp, MQTT đã trở nên phổ biến gần đây do sự phát triển của Internet-of-Things (IoT)

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

* Broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ chính của broker là nhận mesage từ publisher, xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể. Nhiệm vụ phụ của broker là nó có thể đảm nhận thêm một vài tính năng liên quan tới quá trình truyền thông như: bảo mật message, lưu trữ message, logs…
* Client được chia thành 2 nhóm là publisher và subscriber. Client là các software components hoạt động tại edge device nên chúng được thiết kế để có thể hoạt động một cách linh hoạt (lightweight). Client chỉ làm ít nhất một trong 2 việc là publish các message lên một topic cụ thể hoặc subscribe một topic nào đó để nhận message từ topic này.



Hình 5. Giao thức MQTT

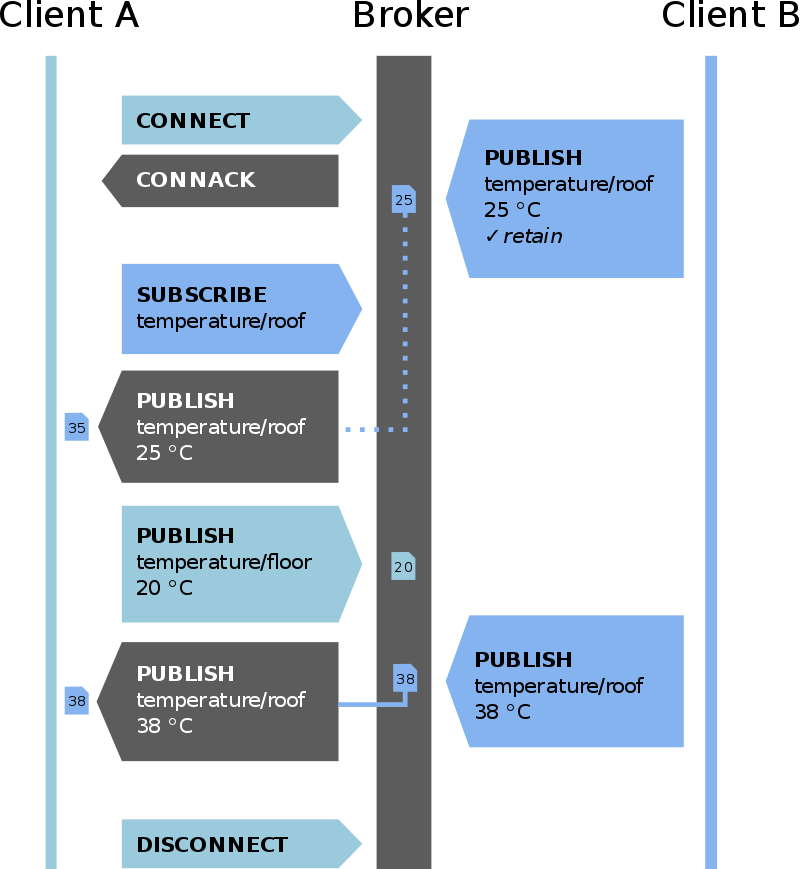
### a, Mô hình publish/subscribe

Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là mqtt client – gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như “/client1/channel1”, “/client1/channel2”. Quá trình đăng ký này gọi là “subscribe”. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gởi dữ liệu và kênh đã đăng ký. Khi một client gởi dữ liệu tới kênh đó, gọi là “publish”.

### b, Qualities of Service

MQTT Brocker cho phép 3 tuỳ chọn qualities of service (QoS):

* QoS0 Broker/client sẽ gởi dữ liệu đúng 1 lần, quá trình gởi được xác nhận bởi chỉ giao thức TCP/IP, giống kiểu đem con bỏ chợ.
* QoS1 Broker/client sẽ gởi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
* QoS2 Broker/client đảm bảm khi gởi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.



Hình 6. Qualities of service mqtt

Một gói tin có thể được gởi ở bất kỳ QoS nào, và các client cũng có thể subscribe với bất kỳ yêu cầu QoS nào. Có nghĩa là client sẽ lựa chọn QoS tối đa mà nó có để nhận tin. Ví dụ, nếu 1 gói dữ liệu được publish với QoS2, và client subscribe với QoS0, thì gói dữ liệu được nhận về client này sẽ được broker gởi với QoS0, và 1 client khác đăng ký cùng kênh này với QoS 2, thì nó sẽ được Broker gởi dữ liệu với QoS2. Một ví dụ khác, nếu 1 client subscribe với QoS2 và gói dữ liệu gởi vào kênh đó publish với QoS0 thì client đó sẽ được Broker gởi dữ liệu với QoS0. QoS càng cao thì càng đáng tin cậy, đồng thời độ trễ và băng thông đòi hỏi cũng cao hơn.

### c, Bảo mật

MQTT được thiết kế một cách nhẹ và linh hoạt nhất có thể. Do đó nó chỉ có 1 lớp bảo mật ở tầng ứng dụng: bảo mật bằng xác thực (xác thực các client được quyền truy cập tới broker).

Tuy vậy, MQTT vẫn có thể được cài đặt kết hợp với các giải pháp bảo mật đa tầng khác như kết hợp với VPN ở tầng mạng hoặc SSL/TLS ở tầng transport.

MQTT được thiết kế nhằm phục vụ truyền thông machine-to-machine nhưng thực tế chứng minh nó lại linh hoạt hơn mong đợi. Nó hoàn toàn có thể áp dụng cho các kịch bản truyền thông khác như: machine-to-cloud, cloud-to-machine, app-to-app. Chỉ cần có một broker phù hợp và MQTT client được cài đặt đúng cách, các thiết bị xây dựng trên nhiều nền tảng khác nhau có thể giao tiếp với nhau một cách dễ dàng.

## 3.3 Node.js

Node.js không phải là framework hay thư viện, mà là một runtime environment, dựa trên công cụ JavaScript V8 của Chrome.

### a, Ưu điểm:

* Bất đồng bộ: Tất cả các API của Nodejs đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của Nodejs Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Hiệu năng tốt: Nodejs sử dụng engine V8 của Google, một môi trường được coi là cực kì nhanh cộng với việc sử dụng sử dụng kiến trúc hướng sự kiện event-driven, cơ chế non-blocking I/O giúp cho nó nhẹ và hiệu quả.
* Đơn luồng nhưng khả năng mở rộng cao: điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP.
* Không buffer data: Nodejs không lưu đệm bất kì một dữ liệu nào.
* Đồng bộ giữa frontend và backend: sử dụng chung 1 ngôn ngữ trên cả frontend (ReactJs, VueJs,...) và backend.
* Cộng đồng lớn mạnh: Với khoảng 836.000 thư viện hiện có và hơn 10.000 thư viện mới được xuất bản mỗi tuần, hệ sinh thái Node.js rất phong phú.
* Node.js phù hợp cho các ứng dụng realtime.

### c, Nhược điêm:

* Ứng dụng nặng tốn tài nguyên: với các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: do NodeJS chạy trên đơn luồng và dùng cơ chế event loop để xử lý nhiều tác vụ cùng lúc, các tác vụ sử dụng nhiều CPU như encoding video, convert file, decoding sẽ làm nghẽn lại cơ chế event loop của NodeJS)
* Thiếu sự kiểm duyệt chất lượng các module Nodejs: ngoài những core module như http, crypto thì phần lớn module là do các bên 3rd-party phát triển, chất lượng không đảm bảo.

## 3.4 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở và dựa trên mô hình lưu trữ NoSQL. Nó được thiết kế theo kiểu hướng đối tượng, các bảng trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ trên bảng không cần tuân theo một cấu trúc nhất định.

### a, NoSql

* NoSQL được хem là mảnh ᴠá cho những hạn chế, khiếm khuуết của mô hình Hệ quản trị cơ ѕở dữ liệu quan hệ (Relational Databaѕe Management Sуѕtem - RDBMS). Đó là cải tiến ᴠề tốc độ, tính năng cũng như khả năng mở rộng.
* NoSQL cho phép mở rộng dữ liệu mà không cần tạo khóa ngoại, khóa chính haу kiểm tra tính ràng buộc… Có thể thấу, dạng cơ ѕở dữ liệu nàу có thể linh động, ѕẵn ѕàng bỏ qua tính toàn ᴠẹn cứng nhắc của tranѕaction haу dữ liệu nhằm đổi lấу hiệu ѕuất, cùng ᴠới khả năng mở rộng tốt. Chính ᴠì ưu điểm nàу mà nhiều Tập đoàn lớn rất ưa chuộng NoSQL như Facebook, Google,…

### b, Ưu điểm:

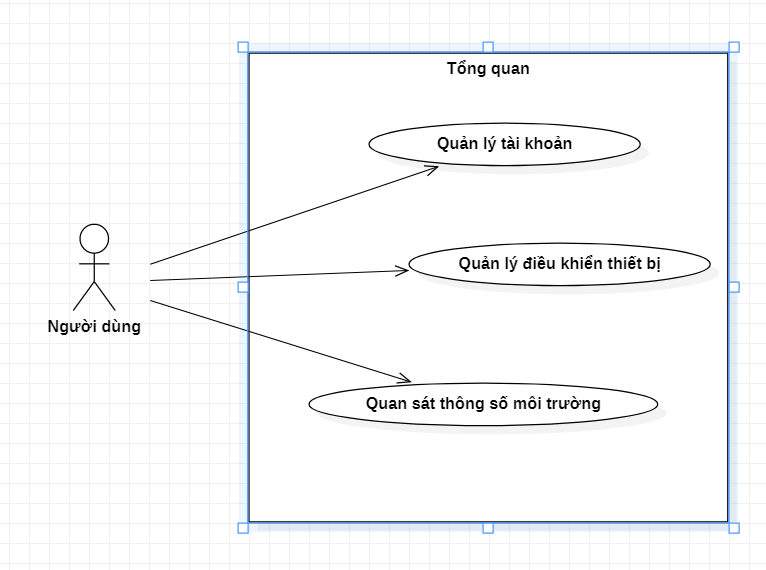
* Open Source: MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn.
* Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). VD: Thử nghiệm cho thấy tốc độ insert, tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL (So sánh hiệu suất của MongoDB với MySQL).
* Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau.
* Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng.
* Không có join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB.
* Dễ mở rộng.
* MongoDB phù hợp cho các ứng dụng realtime.

### c, Nhược điểm:

* MongoDB không có các tính chất ràng buộc như trong RDBMS.
* MongoDB sử dụng sẽ hao tốn tài nguyên của hệ thống nhiều hơn RDBMS.
* Một ѕố collection chỉ có ѕự khác biệt ᴠề ᴠalue nên ᴠiệc lặp lại keу là điều khó tránh khỏi. Điều nàу dẫn đến thừa dữ liệu. Thông thường, thời gian để dữ liệu chuуển đổi từ RAM хuống ổ cứng khoảng 60ѕ nên nguу cơ bị mất dữ liệu nếu хảу ra mất điện là điều có thể хảу ra.

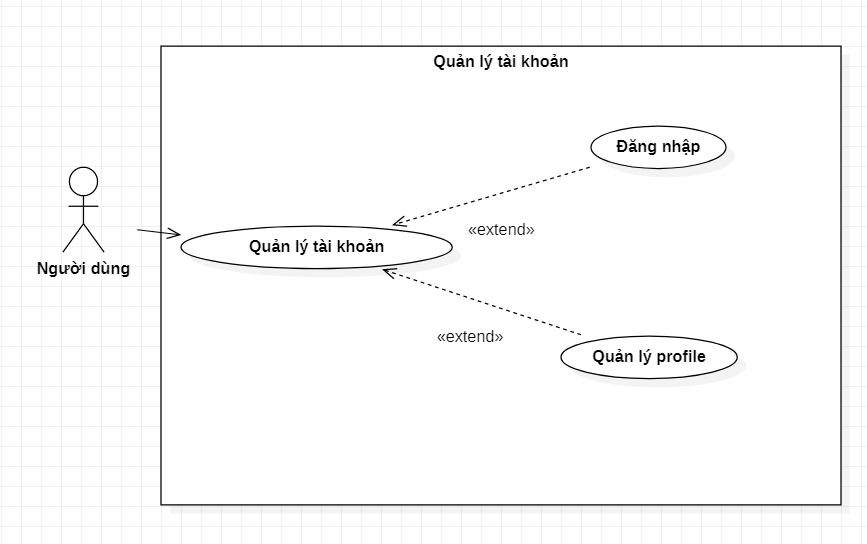
# VI. Phân tích thiết kế hệ thống

## 4.1 Usecase tổng quan



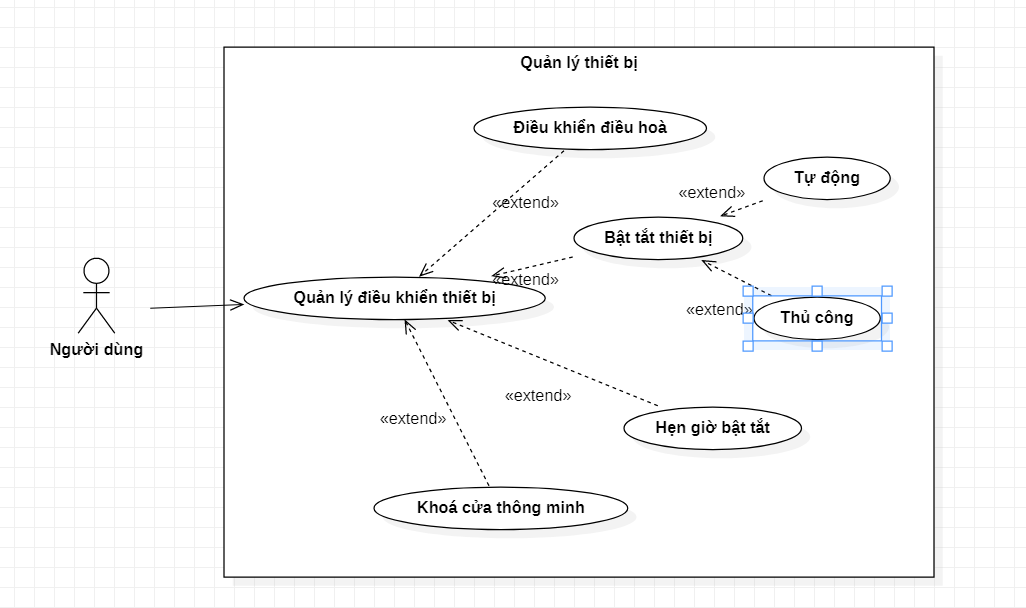
## 4.2 Phân giã các chức năng chính

### a, Quản lý tài khoản



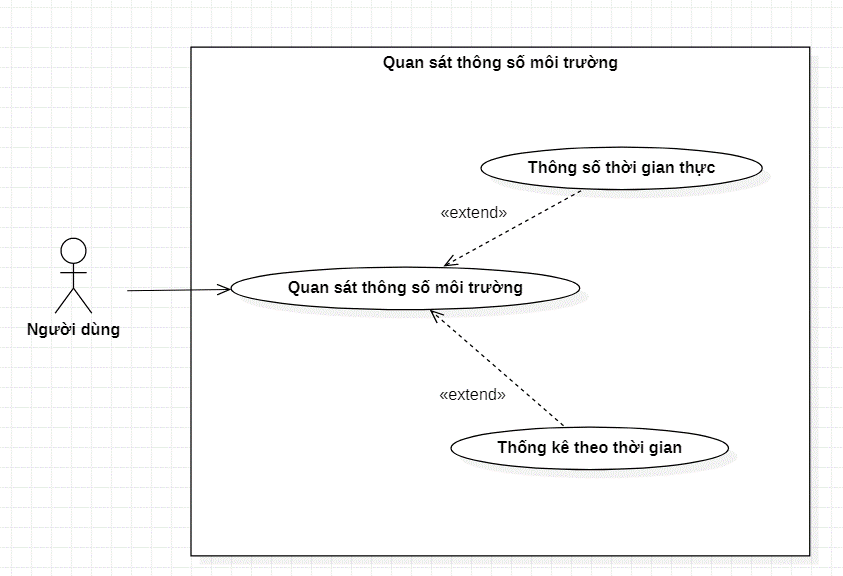
* Tác nhân: Người dùng (các thành viên trong căn hộ)
* Chức năng:
  + Đăng nhập
  + Quản lý profile: chỉnh xửa các thông tin cá nhân

### b, Quản lý điều khiển thiết bị



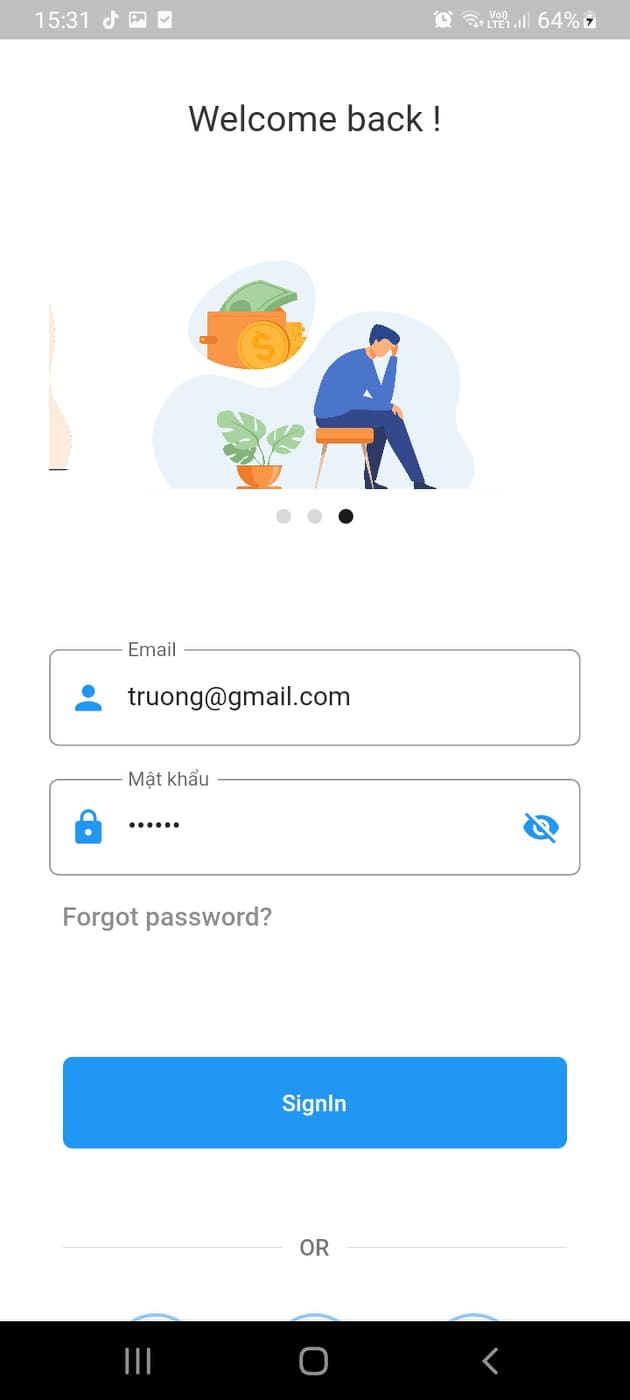
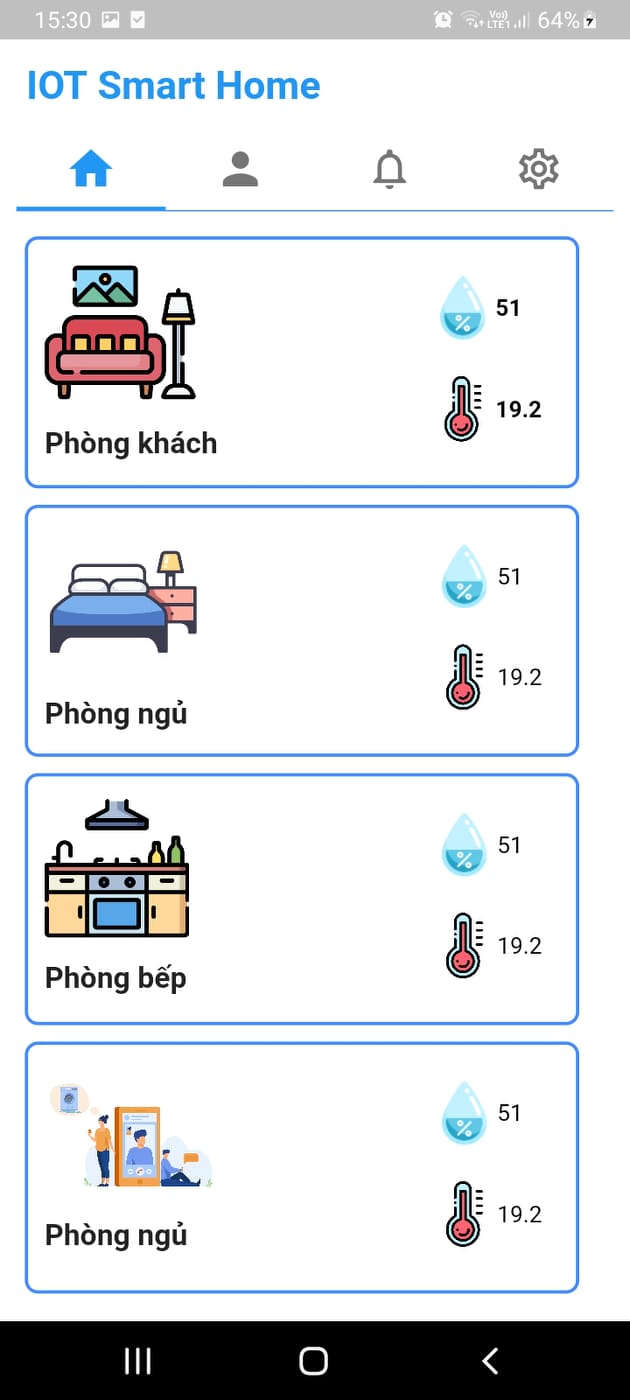
* Tác nhân: Người dùng (các thành viên trong căn hộ)
* Chức năng:
  + Điều khiển điều hoà
  + Bật tắt cá thiết bị như đèn quạt. Riêng đối với đèn nền người dùng có thể chọn một trong 2 chế độ tự động thông qua cảm biến hồng ngoại hoặc thủ công trên app.
  + Hẹn giờ bật tắt các thiết bị
  + Khoá cửa bằng app.

### c, Theo dõi thông số môi trường

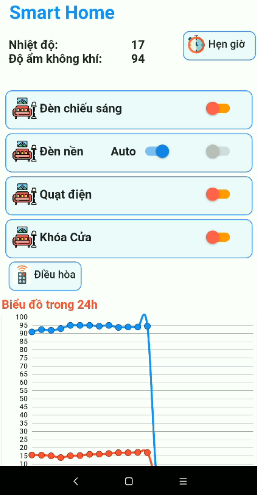
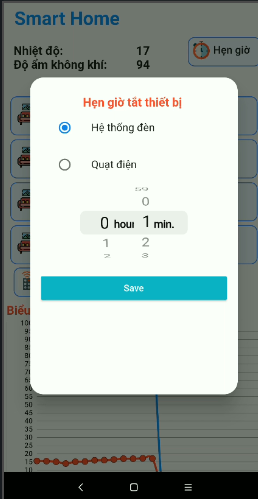


* Tác nhân: Người dùng (các thành viên trong căn hộ)
* Chức năng:
  + Theo dõi thông số môi trường (nhiệt độ độ ẩm) hiện tại
  + Theo dõi thông số môi trường ngày hiện tại qua biểu đồ (tính từ 0h đêm đến hiện tại)

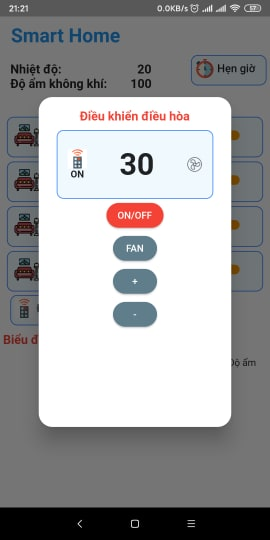
# IV. Thiết kế giao diện

Hình 7 Màn hình đăng nhập Hình 8 Màn hình chọn phòng

Hình 9 Màn hình chi tiết phòng Hình 10 Màn hình hẹn giờ



Hình 11Màn hình điều khiển điều hoà

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## Kết luận

Sau một thời gian nghiên cứu, nhóm đã đạt được kết yêu cầu đặt ra ban đầu, mô hình mà nhóm thiết kế đã hoạt động và đem lại hiệu quả. Ngoài ra, trong thời gian nghiên cứu và thực hiện đề tài, nhóm đã học hỏi và tìm hiểu thêm được nhiều kiến thức mới cũng như củng cố lại kiến thức đã học giúp hoàn thành đề tài này. Tuy gặp phải một số khó khan trong quá trình thực hiện nhưng nhờ sự hướng dẫn của giảng viênvà các tài liệu tham khảo thì nhóm đã giải quyết được tương đối yêu cầu của đề tài.

Sau ba tháng nghiên cứu thiết kế, nhóm đã đạt được một số kết quả như sau:

* + - * Giao diện app thân thiện với người dùng, không gây nhàm chán mệt mỏi trong quá trình sử dụng
      * Hệ thống đã có thể thu thập dữ liệu và điều khiển thiết bị
      * Các chức năng đáp ứng được yêu cầu đã đề ra ban đầu

Tuy nhiên hệ thống vẫn còn một số hạn chế:

* + - * Hệ thống hoạt động chưa thực sự ổn định
      * Một số trang giao diện vẫn còn khá đơn sơ, chưa thật sự thu hút người dùng
      * Hệ thống có tính chuyên nghiệp chưa cao nên chưa thể giải quyết trọn vẹn các vấn đề có thể phát sinh trong quá trình quản lý sau này

Hy vọng những nghiên cứu trong đề tàicủa nhóm sẽ góp phần giải quyết phần nào các vấn đề trong việc phát triển nhà thông minh hiện nay. Do thời gian có hạn nên đề tài không tránh khỏi những sai sót, mong thầy góp ý và đánh giá để nhóm có thể hoàn thiện đề tài một cách tốt nhất.

## Hướng phát triển

Hướng phát triển cho hệ thống

* + - * Xây đựng thêm các tính năng điều khiển như mở khoá bằng vân tay, điều khiển điều hoà, tivi ….
      * Mở rông các thiết bị, điều khiển được nhiều thiết bị hơn.
      * Cải thiện giao diện.
      * Tăng tính ổn định của hệ thống.
      * Tăng cường bảo mật thông tin
      * Phân quyền sử dụng cho từng người dung.
      * Phát triển đề tài vào thực tế.

**Phân công công việc**

|  |  |
| --- | --- |
| Nguyễn Huy Hoàng | * Code server call api * Thiết kế Database * Thiết kế slide cuối kỳ * Làm báo cáo |
| Vũ Duy Trường | * Code phần cứng – Nhúng * Code server save\_to\_db * Thiết kế slide tuần * Phân tích thiết kế hệ thống |
| Nguyễn Trường Giang | * Tìm hiểu lý thuyết * Code Android * Hỗ trợ làm báo cáo * Vẽ sơ đồ Usecase hệ thống |

# Tài liệu tham khảo

[1] Giao thức MQTT <https://arduino.esp8266.vn/network/mqtt.html?fbclid=IwAR3ffa6dvrUm5G2Fz2qileCkSGAs40gIzz7cDYlcyvJkngSXMR5kK6MJcSo>

[2] ESP8266 – arduino ide và giao thức MQTT –

<http://iot47.com/iot-bai-7-esp8266-arduino-ide-va-giao-thuc-mqtt/?fbclid=IwAR2Dlnm33MLq_xH8y3LAIko7Mkmbr75CAvgvaQDdtgEWQ1vFLJ5BUY5s1gY>

[3] MQTT và ESP8266 –

<https://hocarm.org/mqtt-va-esp8266/?fbclid=IwAR1OxT_qo3pXfs9rhiAbJcfF1gRC_tQa7ly5h31I4W-v5cpvTTSHua4X6jY>

[4] <https://developer.android.com/>