TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----

Icon

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**IoT và Ứng dụng**

***Đề tài:* NHÀ THÔNG MINH ĐO ĐỘ ẨM, NHIỆT ĐỘ VÀ ĐIỀU KHIỂN BẬT TẮT ĐÈN**

Giảng viên: T**S. Nguyễn Đình Thuận**

Sinh viên thực hiện:

| STT | Họ và tên | MSSV |
| --- | --- | --- |
| 1 | Trần Đức Sơn | 20163571 |
|  |  |  |
|  |  |  |

**Hà Nội, 3/2022**

MỤC LỤC

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN 4](#_Toc97065433)

[1.1. Đặt vấn đề 4](#_Toc97065434)

[1.2. Thiết kế mô hình 4](#_Toc97065435)

[1.3. Xây dựng hệ thống 4](#_Toc97065436)

[CHƯƠNG 2. CHUẨN BỊ 7](#_Toc97065437)

[2.1. Các chức năng chính cần làm 7](#_Toc97065438)

[2.2. Thiết bị sử dụng 7](#_Toc97065439)

[CHƯƠNG 3. CÁC BƯỚC THỰC HIỆN 8](#_Toc97065440)

[3.1. Triển khai hệ thống IoT 8](#_Toc97065441)

[3.1.1. Sơ đồ triển khai tổng quát 8](#_Toc97065442)

[3.2. Chi tiết các bước thực hiện 8](#_Toc97065443)

[3.2.1. Publish dữ liệu lên broker 8](#_Toc97065444)

[3.2.2. Lưu vào cơ sở dữ liệu 9](#_Toc97065445)

[3.2.3. Hiển thị dữ liệu trên ứng dụng/trang web 11](#_Toc97065446)

[3.2.4. Điều khiển đèn LED 13](#_Toc97065447)

[3.2.5. Tool test chức năng và khả năng đáp ứng của Server 13](#_Toc97065448)

[3.3. Triển khai thực tế 13](#_Toc97065449)

[CHƯƠNG 4. KẾT LUẬN 14](#_Toc97065450)

[4.1. Nhận xét và kết luận 14](#_Toc97065451)

[4.2. Hướng phát triển 14](#_Toc97065452)

**DANH SÁCH HÌNH ẢNH**

Hình 1: Sơ đồ tổng quát ............................................................................................. 5

Hình 2: Sơ đồ Usecase ............................................................................................... 5

Hình 3: Cấu trúc phần cứng cơ bản của Thiết bị IoT ................................................ 6

Hình 4: Mô phỏng RabbitMQ trên Docker ……….................................................... 9

Hình 5: Dữ liệu nhiệt độ, độ ẩm được lưu trữ được lưu trữ vào database ................ 11

Hình 6: Cấu trúc bảng lưu thông tin người dùng ...................................................... 11

Hình 7: Các bảng chính lưu trên NoSQL Database …………………….................. 11

Hình 8: Giao diện quản lý trên Website .................................................................... 13

# GIỚI THIỆU TỔNG QUAN

## Đặt vấn đề

Thời đại 4.0 đòi hỏi yêu cầu tự động hóa cao. Nhu cầu mọi người muốn tìm một thiết bị, hệ thống có thể thực hiện được các hoạt động cơ bản trong nhà như điều khiển bật tắt đèn hay theo dõi được thông tin nhiệt độ, độ ẩm trong nhà. Chính vì vậy, em đã thiết kế ra hệ thống thực hiện các chức năng cơ bản, đồng thời nghiên cứu bổ sung thêm một số chức năng nâng cao cho phù hợp

Một số ưu điểm của hệ thống:

* Kiểm soát các thông số cơ bản của thời tiết: nhiệt độ, độ ẩm.
* Điều khiển bật tắt đèn thông qua giao diện Website
* Phân quyền, kiểm soát đăng nhập, chỉ những người có tài khoản mới có thể đăng nhập hệ thống và điều khiển

Tuy vậy phương pháp này còn có một số nhược điểm:

* Hệ thống còn tương đối đơn giản. Nếu muốn mở rộng yêu cầu cấu hình phần cứng và phần mềm đáp ứng.

## Thiết kế mô hình

Mô hình được thiết kế gồm 3 phần: Phần thiết bị IoT, phần Broker chịu trách nhiệm truyền tải MQTT Message và phần server/website chịu trách nhiệm điều khiển.

## Xây dựng hệ thống

**Đối tượng sử dụng:** Gia đình nhỏ smarthome. Những đối tượng muốn điều khiển, theo dõi các thiết bị IoT trong nhà thông qua một chiếc smartphone nhỏ gọn hoặc một website đơn giản

**Các chức năng chính:**

* Điều khiển bật tắt đèn.
* Thu thập thông tin về độ ẩm, nhiệt độ ngôi nhà

**Thiết bị, công nghệ sử dụng:**

* Bộ kit NodeMCU ESP8266, Sensor DHT11, .
* Giao thức giao tiếp: HTTP/HTTPS, MQTT, SocketIO

**Mô hình triển khai hệ thống:**

* Sơ đồ chung hệ thống:

Diagram

Description automatically generated

Hình 1: Sơ đồ tổng quát

* Sơ đồ Usecase cơ bản

Diagram

Description automatically generated

Hình 2: Sơ đồ Usecase

* Database cơ bản: Mô hình sử dụng MongoDB để lưu trữ thông tin nhiệt độ, độ ẩm được gửi về từ phía thiết bị. Thông tin đính kèm bao gồm “temperature” (Nhiệt độ), “humidity”(Độ ẩm), “time”(Đại diện cho thời gian)
* Mô tả tóm tắt sự kết nối giữa các phần cứng:

Diagram

Description automatically generated

Hình 3: Cấu trúc phần cứng cơ bản của Thiết bị IoT

# CHUẨN BỊ

## Các chức năng chính cần làm

* Trang web cho người dùng quản lý:
* Quan sát, quản lý tình hình độ ẩm, nhiệt độ
* Điều khiển bật/tắt đèn
* Đăng nhập, Đăng ký tài khoản
* Server:
* Kết nối Database.
* Kết nối MQTT, Socket.
* Viết các API phục vụ cho quản lý phía người dùng.
* Phần cứng:
* Tìm hiểu thông số, cách lắp đặt và sử dụng.
* Thu thập dữ liệu gửi về Server.
* Đảm bảo hoạt động và hoạt động tốt.

## Thiết bị sử dụng

Phần cứng sử dụng bao gồm: 1x kit NodeMCU ESP8266, 1x Sensor DHT11, 1 Đèn led.

# CÁC BƯỚC THỰC HIỆN

## Triển khai hệ thống IoT

### Sơ đồ triển khai tổng quát

Để triển khai được hệ thống thì cần chia ra làm các bộ phận riêng:

* Phần thiết bị IoT: Chức năng điều khiển đèn. Nhận lệnh từ phần điều khiển để thu nhận thông tin nhiệt độ, độ ẩm và gửi trả về cho server thông qua chip ESP8266 được tích hợp trong kit NodeMCU.
* Phần Broker: Chịu trách nhiệm trung gian trong việc nhận thông điệp MQTT Message từ phía hệ thống máy hút bụi
* Phần Backend/Frontend/Database: Lưu trữ thông tin nhận được, điều khiển, biểu diễn thông tin, hỗ trợ tương tác với người dùng

## Chi tiết các bước thực hiện

### Publish dữ liệu lên broker

Để trao đổi dữ liệu giữa máy hút bụi và server backend cần thiết kế giao tiếp thông qua một Broker cung cấp Queue hỗ trợ mô hình Pub/Sub. Ở đây hệ thống sử dụng RabbitMQ đóng vai trò như Broker để làm trung gian vận chuyển các thông điệp MQTT Message giúp gửi và nhận dữ liệu với thiết bị IoT .

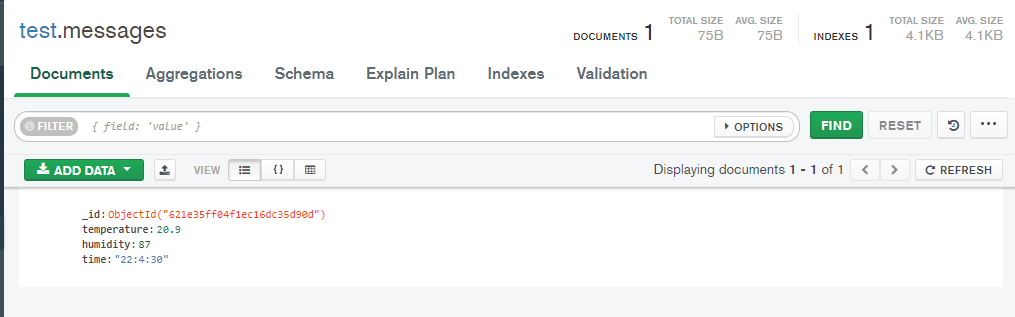
Graphical user interface, application

Description automatically generated

Hình 4: Mô phỏng RabbitMQ trên Docker

### Lưu vào cơ sở dữ liệu

Thông tin về nhiệt độ, độ ẩm sẽ được lưu lại vào MongoDB để người dùng có thể theo dõi được thông qua website. Hệ thống đăng ký một topic ‘homeinfo’ để cập nhật thông tin nhiệt độ độ ẩm, trao đổi message giữa thiết bị và Server.



Hình 5: Dữ liệu nhiệt độ, độ ẩm được lưu trữ vào database

Graphical user interface, text, application, email

Description automatically generated

Hình 6: Cấu trúc bảng lưu thông tin người dung

***Cấu trúc các bảng chính:***

Table

Description automatically generated

Hình 7: Các bảng chính trên NoSQL Database

***Giải thích các trường thông tin chính:***

Bảng Messages:

* temperature: lưu thông tin nhiệt độ phòng
* humidity: lưu thông tin độ ẩm phòng
* time: thời gian tương ứng với các thông tin trên

Bảng User:

* name: tên người dung
* email: email đăng ký
* phone: điện thoại người dung
* logined: kiểm tra tài khoản đã đăng nhập hay chưa
* lastOnline: Thời gian đăng nhập lần cuối
* password: mật khẩu đăng nhập
* date: Thời gian đăng ký
* tokens: các token socketio đã được cấp

### Hiển thị dữ liệu trên ứng dụng/trang web

Server cung cấp API hỗ trợ update dữ liệu, cập nhật lại nhiệt độ và độ ẩm trung bình trong ngày.

Graphical user interface, text, application, website

Description automatically generated

Hình 8: Giao diện quản lý trên Website

***Danh sách các API chính:***

Tài khoản:

* Auth/login: đăng nhập
* Auth/logout: đăng xuất
* Auth/register: đăng ký

Điều khiển đèn:

* Messages/turn-on-led: bật đèn
* Messages/turn-off-led: tắt đèn

***SocketIO***

* Cập nhật nhiệt độ, độ ẩm theo thời gian định kỳ. Cứ N giây sẽ emit thông điệp lên server để lấy dữ liệu nhiệt độ độ ẩm

### Điều khiển đèn LED

Để phục vụ chức năng điều khiển đèn LED. Cần thiết kế API có chức năng gửi thông điệp MQTT đển cho thiết bị IoT. Ta tạo một topic mới có tên ‘dkled’ ở Broker để lắng nghe message trao đổi giữa server và thiết bị.

### Tool test chức năng và khả năng đáp ứng của Server

* Xây dựng bằng Python
* Giả lập N thiết bị cùng lúc gửi thông tin trực tiếp đến Server để kiểm tra khả năng đáp ứng (N là số thiết bị người dùng nhập vào)
* Giả lập 1 thiết bị gửi định kỳ mỗi N giây (N giây là số người dùng nhập vào)
* Kết quả: Server vẫn hoạt động bình thường

## Triển khai thực tế

Chi tiết phần triển khai có đính kèm trong phần demo

# KẾT LUẬN

## Nhận xét và kết luận

Nhìn chung, mô hình thiết kế đảm bảo yêu cầu bài toán đặt ra là thiết kế một hệ thống kết hợp hệ thống IoT điều khiển đèn và theo dõi nhiệt độ, độ ẩm phòng.

Hệ thống nhìn chung triển khai đơn giản. Do hạn chế về thời gian và thiết bị nên chỉ triển khai 1 thiết bị trên mỗi chức năng đã thiết kế.

Bài toán phù hợp để làm quen với một hệ thống IoT và phân tích, thiết kế hệ thống. Thông qua Đề tài, em học được cách tự tìm hiểu, nghiên cứu. Làm quen sâu hơn về phân tích thiết kế và các công nghệ liên quan đến IoT

## Hướng phát triển

Hiện tại đặt ra nhu cầu mở rộng trên hộ gia định với quy mô lớn hơn, yêu cầu nhiều thiết bị và phân cấp, quản lý các thiết bị. Những yêu cầu này cần kỹ thuật xử lý cao hơn nhưng trong khuôn khổ của một project nhỏ em chưa có thời gian đáp ứng được.

**TÀI LIỆU THAM KHẢO**

1. Arduino IDE: https://www.arduino.cc/en/software
2. RabbitMQ: <https://hub.docker.com/_/rabbitmq>
3. Cơ bản về NodeMCU ESP8266: <https://lophocvui.com/iot-internet-of-things/smart-home/tong-quan-ve-esp8266/>