**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÊ KHẮC TIẾN**

**KHÓA LUẬN TỐT NGHIỆP**

**PHÁT SINH DỮ LIỆU TẤN CÔNG CHỐNG LẠI IDS BẰNG MÔ HÌNH ĐỐI KHÁNG TẠO SINH**

**GENERATIVE ADVERSARIAL NETWORKS FOR ATTACK AGAINST INTRUSION DETECTION SYSTEM**

**KỸ SƯ NGÀNH AN TOÀN THÔNG TIN**

**TP. HỒ CHÍ MINH, 2020**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÊ KHẮC TIẾN – 16521221**

**KHÓA LUẬN TỐT NGHIỆP**

**PHÁT SINH DỮ LIỆU TẤN CÔNG CHỐNG LẠI IDS BẰNG MÔ HÌNH ĐỐI KHÁNG TẠO SINH**

**GENERATIVE ADVERSARIAL NETWORKS FOR ATTACK AGAINST INTRUSION DETECTION SYSTEM**

**KỸ SƯ NGÀNH AN TOÀN THÔNG TIN**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. PHẠM VĂN HẬU**

**TP. HỒ CHÍ MINH, 2020**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số 98/QĐ-ĐHCNTT ngày 02 tháng 03 năm 2020 của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

LỜI CẢM ƠN

Để hoàn thành đề tài này, em xin chân thành gửi lời cảm ơn và lòng biết ơn sâu sắc tới Thầy Chung Quang Khánh, người đã dành thời gian quý báu của mình để hướng dẫn trực tiếp cho em cách thức nghiên cứu đề tài, để em hoàn thành tốt đề tài của mình. Bên cạnh đó, em cũng gửi lời cám ơn đến các thầy cô trong trường Đại học Công nghệ thông tin – Tp Hồ Chí Minh nói chung và các thầy cô trong khoa Kỹ thuật máy tính, Bộ môn Vi mạch – Hệ thống nhúng nói riêng, đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt 4 năm vừa qua để em vững bước trên con đường học tập và công việc của mình sau này.

Xin cảm ơn các tất cả các bạn bè, anh chị, những người đã giúp đỡ em về mọi mặt trong quá trình em làm đề tài.

Đặc biệt con xin cảm ơn tới bố mẹ, anh chị em trong gia đình, những người có công sinh thành, nuôi dưỡng con thành người để con có ngày hôm nay. Cảm ơn những lời động viên của bố mẹ trong quá trình con học tập và lúc con gặp khó khăn.

Trong quá trình làm không thể không gặp những khó khăn và những sai sót, mong thầy cô và các bạn quan tâm đến đề tài của nhóm, góp ý để nhóm rút kinh nghiệm trong các công trình nghiên cứu sau này.

CHÂN THÀNH CẢM ƠN!

|  |  |
| --- | --- |
| Sinh viên thực hiện | |
|  |  |
|  |  |
| Huỳnh Tiến Đạt | Lâm Tiên Điền An |

Khoa Kỹ Thuật Máy Tính, Lớp KTMT2015

MỤC LỤC

[TÓM TẮT KHÓA LUẬN 1](#_Toc33705202)

[Chương 1. TỔNG QUAN 3](#_Toc33705203)

[1.1. Lý do chọn đề tài 3](#_Toc33705204)

[1.2. Tình hình nghiên cứu trong và ngoài nước 3](#_Toc33705205)

[1.2.1. Trong nước 3](#_Toc33705206)

[1.2.2. Ngoài nước 4](#_Toc33705207)

[1.3. Mục tiêu đề tài 6](#_Toc33705208)

[1.4. Đối tượng và phạm vi nghiên cứu. 7](#_Toc33705209)

[1.4.1. Đối tượng nghiên cứu 7](#_Toc33705210)

[1.4.2. Phạm vi nghiên cứu 7](#_Toc33705211)

[1.5. Ý nghĩa khoa học và thực tiễn của đề tài 7](#_Toc33705212)

[1.5.1. Ý nghĩa khoa học 7](#_Toc33705213)

[1.5.2. Ý nghĩa thực tiễn 7](#_Toc33705214)

[Chương 2. CƠ SỞ LÝ THUYẾT 8](#_Toc33705215)

[2.1. Chỉ số chất lượng không khí (Air Quality Index- AQI) 8](#_Toc33705216)

[2.1.1. Khái niệm 8](#_Toc33705217)

[2.1.2. Giới thiệu về hạt bụi lơ lửng 8](#_Toc33705218)

[2.1.3. Giới thiệu về PM2.5 8](#_Toc33705219)

[2.1.4. Giới thiệu về AQI 8](#_Toc33705220)

[2.1.5. Công thức tính toán AQI 9](#_Toc33705221)

[2.1.6. Công thức tính toán mật độ ô nhiễm 9](#_Toc33705222)

[2.1.7. Quy trình tính toán và sử dụng AQI trong đánh giá chất lượng môi trường không khí xung quanh 11](#_Toc33705223)

[2.2. Hệ điều hành FreeRTOS 12](#_Toc33705224)

[2.2.1. Hệ điều hành là gì ? 12](#_Toc33705225)

[2.2.2. FreeRTOS là gì ? 12](#_Toc33705226)

[2.3. Giao thức HTTP 13](#_Toc33705227)

[2.4. Giao thức TCP/IP 13](#_Toc33705228)

[2.5. Nền tảng (platform) NodeJS 15](#_Toc33705229)

[2.5.1. NodeJS là gì ? 15](#_Toc33705230)

[2.5.2. Ưu điểm của NodeJS 16](#_Toc33705231)

[2.6. Angular 17](#_Toc33705232)

[2.6.1. Angular là gì ? 17](#_Toc33705233)

[2.6.2. Các phiên bản của Angular 17](#_Toc33705234)

[2.6.3. Material design với Angular 18](#_Toc33705235)

[2.6.4. NgRx 18](#_Toc33705236)

[2.7. InfluxDB 20](#_Toc33705237)

[2.7.1. Time Series Database là gì ? 20](#_Toc33705238)

[2.7.2. InfluxDB 21](#_Toc33705239)

[2.8. MongoDB 22](#_Toc33705240)

[2.8.1. MongoDB là gì ? 22](#_Toc33705241)

[2.8.2. Ưu điểm của MongoDB 22](#_Toc33705242)

[2.8.3. Nhược điểm của MongoDB 23](#_Toc33705243)

[2.9. Smart Config 23](#_Toc33705244)

[2.10. Android 24](#_Toc33705245)

[2.10.1. Android là gì ? 24](#_Toc33705246)

[2.10.2. Ưu điểm của Android 25](#_Toc33705247)

[2.10.3. Nhược điểm của Android 25](#_Toc33705248)

[2.11. Retrofit 26](#_Toc33705249)

[2.11.1. Retrofit là gì ? 26](#_Toc33705250)

[2.11.2. Vậy tại sao nên sử dụng Retrofit? 27](#_Toc33705251)

[2.12. GSON 27](#_Toc33705252)

[Chương 3. HIỆN THỰC HỆ THỐNG 28](#_Toc33705253)

[3.1. Thiết kế hệ thống 28](#_Toc33705254)

[3.2. Giải pháp phần cứng 29](#_Toc33705255)

[3.2.1. Sơ đồ khối 29](#_Toc33705256)

[3.2.2. Cách thức hoạt động 30](#_Toc33705257)

[3.2.3. ESP32 WROVER KIT 30](#_Toc33705258)

[3.2.4. Module cảm biến bụi Shinyei PPD42NS 31](#_Toc33705259)

[3.2.5. Module cảm biến nhiệt độ và độ ẩm SHT31 33](#_Toc33705260)

[3.2.6. Module thời gian thực RTC DS1307 33](#_Toc33705261)

[3.3. Thiết kế và hiện thực phần mềm trên ESP32 WROVER KIT 34](#_Toc33705262)

[3.3.1. Lưu đồ giải thuật 34](#_Toc33705263)

[3.3.2. Chức năng và mô tả 39](#_Toc33705264)

[3.4. Thiết kế và hiện thực Server 40](#_Toc33705265)

[3.4.1. Thông tin của server 40](#_Toc33705266)

[3.4.2. Chức năng của server 40](#_Toc33705267)

[3.4.3. Thiết kế server với NodeJS 41](#_Toc33705268)

[3.4.4. Xây dựng các API 43](#_Toc33705269)

[3.5. Thiết kế và hiện thực Web 45](#_Toc33705270)

[3.5.1. Angular 8 45](#_Toc33705271)

[3.5.2. Chức năng của ứng dụng 45](#_Toc33705272)

[3.5.3. Thiết kế các màn hình ứng dụng 46](#_Toc33705273)

[3.5.4. Xây dựng các màn hình 52](#_Toc33705274)

[3.5.5. Xây dựng chức năng cho các màn hình 56](#_Toc33705275)

[3.6. Thiết kế và hiện thực ứng dụng di động trên Android 58](#_Toc33705276)

[3.6.1. Các chức năng của ứng dụng 58](#_Toc33705277)

[3.6.2. Thiết kế giao diện 58](#_Toc33705278)

[3.6.3. Xây dựng ứng dụng 61](#_Toc33705279)

[Chương 4. KẾT QUẢ 65](#_Toc33705280)

[4.1. Kết quả đạt được 65](#_Toc33705281)

[4.1.1. Phần cứng 65](#_Toc33705282)

[4.1.2. Phần mềm 65](#_Toc33705283)

[4.1.3. Thực nghiệm 66](#_Toc33705284)

[4.2. Thuận lợi 69](#_Toc33705285)

[4.3. Khó khăn 69](#_Toc33705286)

[Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 70](#_Toc33705287)

[5.1. Tổng kết 70](#_Toc33705288)

[5.2. Hạn chế và hướng phát triển 70](#_Toc33705289)

[TÀI LIỆU THAM KHẢO 72](#_Toc33705290)

DANH MỤC HÌNH VẼ

[Hình 1.1 Hình ảnh của thiết bị phân tích khí 3](#_Toc33705411)

[Hình 1.2 Thiết bị đo không khí của nhóm sinh viên Đại học Bách khoa Hà Nội 4](#_Toc33705412)

[Hình 1.3 Thiết bị AEROCET 531S của Met One Instruments - Mỹ 5](#_Toc33705413)

[Hình 1.4 PM-1063SD của LUTRON – Đài Loan 5](#_Toc33705414)

[Hình 1.5 Thiết bị đo lượng không khí Xiaomi PM2.5 – Trung Quốc 6](#_Toc33705415)

[Hình 2.1 Mô hình Client-Server 13](#_Toc33705416)

[Hình 2.2 Logo NodeJS 15](#_Toc33705417)

[Hình 2.3 Angular 17](#_Toc33705418)

[Hình 2.4 Biểu đồ luồng dữ liệu NgRx 19](#_Toc33705419)

[Hình 2.5 Logo Influxdb 21](#_Toc33705420)

[Hình 2.6 Logo MongoDB 22](#_Toc33705421)

[Hình 3.1 Sơ đồ hệ thống theo dõi chất lượng không khí AirNow 28](#_Toc33705422)

[Hình 3.2 Sơ đồ khối của thiết bị 29](#_Toc33705423)

[Hình 3.3 ESP32 WROVER KIT 31](#_Toc33705424)

[Hình 3.4 Cảm biến bụi Shinyei PPD42NS 32](#_Toc33705425)

[Hình 3.5 Cảm biến SHT31 33](#_Toc33705426)

[Hình 3.6 Module RTC DS1307 34](#_Toc33705427)

[Hình 3.7 Lưu đồ chung thể hiện các task 35](#_Toc33705428)

[Hình 3.8 Lưu đồ giải thuật của bước giao tiếp với server 36](#_Toc33705429)

[Hình 3.9 Mockup các thông tin cần hiển thị 47](#_Toc33705430)

[Hình 3.10 Các component cần để thể hiện các thông tin 47](#_Toc33705431)

[Hình 3.11 Mockup các biểu đồ cần hiển thị 48](#_Toc33705432)

[Hình 3.12 Mockup thể hiện số liệu thống kê 49](#_Toc33705433)

[Hình 3.13 Table hiển thị thông tin từng user 50](#_Toc33705434)

[Hình 3.14 Sidenav cùng với các công cụ để thay đổi thông tin user 50](#_Toc33705435)

[Hình 3.15 Table hiển thị thông tin từng thiết bị 51](#_Toc33705436)

[Hình 3.16 Sidenav cùng với các công cụ để thay đổi thông tin thiết bị 51](#_Toc33705437)

[Hình 3.17 Giao diện hiển thị thông tin mới nhất 52](#_Toc33705438)

[Hình 3.18 Giao diện hiển thị thông tin theo khoảng thời gian 53](#_Toc33705439)

[Hình 3.19 Giao diện quản lý user 54](#_Toc33705440)

[Hình 3.20 Sidenav tạo user mới 54](#_Toc33705441)

[Hình 3.21 Sidenav đổi mật khẩu user 55](#_Toc33705442)

[Hình 3.22 Giao diện quản lý thiết bị 55](#_Toc33705443)

[Hình 3.23 Sidenav thay đổi thông tin thiết bị 56](#_Toc33705444)

[Hình 3.24 Mockup màn hình xem dữ liệu mới nhất và dữ liệu theo tháng 60](#_Toc33705445)

[Hình 3.25 Mockup giao diện cài đặt của ứng dụng 61](#_Toc33705446)

[Hình 3.26 Giao diện xem dữ liệu mới nhất 62](#_Toc33705447)

[Hình 3.27 Giao diện xem dữ liệu theo tháng 63](#_Toc33705448)

[Hình 3.28 Giao diện cài đặt của ứng dụng 64](#_Toc33705449)

[Hình 4.1 Mặt trước của thiết bị 65](#_Toc33705450)

[Hình 4.2 Mặt bên của thiết bị 65](#_Toc33705451)

DANH MỤC BẢNG

[Bảng 2.1 Bảng thông số của bụi PM2.5(µg/m3) 9](#_Toc33705392)

[Bảng 2.2 Các giá trị của H và C trong điều kiện khô (không mưa) 11](#_Toc33705393)

[Bảng 2.3 Các giá trị của H và C trong khi có mưa 11](#_Toc33705394)

[Bảng 2.4 Bảng xác định mức cảnh báo ô nhiễm môi trường ở Việt Nam 11](#_Toc33705395)

[Bảng 3.1 Thông số kỹ thuật của ESP32 WORVER KIT 31](#_Toc33705396)

[Bảng 3.2 Thông số kỹ thuật của cảm biến bụi PPD42NS 32](#_Toc33705397)

[Bảng 3.3 Thông số kỹ thuật của cảm biến SHT31 33](#_Toc33705398)

[Bảng 3.4 Thông số kỹ thuật của module RTC DS1307 34](#_Toc33705399)

[Bảng 3.5 Danh sách chức năng của phần mềm trên ESP32 39](#_Toc33705400)

[Bảng 3.6 Thông tin về server 40](#_Toc33705401)

[Bảng 3.7 Bảng danh sách các chức năng của server 40](#_Toc33705402)

[Bảng 3.8 Danh sách module của server 42](#_Toc33705403)

[Bảng 3.9 Bảng danh sách các router của server và chức năng 42](#_Toc33705404)

[Bảng 3.10 Danh sách các API của server 43](#_Toc33705405)

[Bảng 3.11 Các chức năng của ứng dụng web 45](#_Toc33705406)

[Bảng 3.12 Màu sắc và ảnh mô tả của card thể hiện thông tin theo aqi 52](#_Toc33705407)

[Bảng 3.13 Bảng chức năng của ứng dụng Android 58](#_Toc33705408)

[Bảng 4.1 Kết quả đo thực tế tại đại sứ quán Hoa Kỳ 67](#_Toc33705409)

[Bảng 4.2 Kết quả đo sau khi thay đổi thời gian lấy mẫu 67](#_Toc33705410)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Chữ viết tắt** | **Chữ tường minh** |
| IoT | Internet of Thing |
| WIFI | Wireless Fidelity |
| API | Application Programming Interface |
| MCU | Microcontroller Unit |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| IP | Internet Protocol |
| ID | Identity |
| RTOS | Real-Time Operating System |
| HTTP | Hyper Text Transfer Protocol |
| SSID | Service Set Identifier |
| PM | Particulate Matter |

TÓM TẮT KHÓA LUẬN

Đề tài XÂY DỰNG HỆ THỐNG THEO DÕI CHẤT LƯỢNG KHÔNG KHÍ được thực hiện nhằm mục đích phục vụ cho người sử dụng có thể theo dõi chất lượng không khí theo thời gian thực bằng một thiết bị kích thước nhỏ, dễ dàng di chuyển và có màn hình hiển thị.

Thiết bị có thể xem được các thông tin cơ bản về chất lượng của không khí như: nhiệt độ, độ ẩm, mật độ hạt bụi có kích thước từ 1µm đến 2.5µm, chỉ số chất lượng không khí, xem thời gian.

Thiết bị có kết nối internet để gửi dữ liệu thu thập được lên server.

Dữ liệu từ thiết bị có thể theo dõi từ xa thông qua ứng dụng di động hoặc ứng dụng web.

MỞ ĐẦU

Môi trường không khí có vai trò rất quan trọng góp phần tạo nên sự sống trên trái đất – cung cấp O2 cho quá trình hô hấp của sự sống hay CO2 cho quá trình quang hợp của các loại sinh vật trên Trái Đất, đây là hai quá trình quan trọng cho sự tồn tại và phát triển của con người. Do đó chất lượng môi trường không khí là vấn đề quan trọng cần được quan tâm hàng đầu. Với sự phát triển kinh tế như hiện nay, bảo vệ môi trường không khí không chỉ là của riêng một quốc gia mà còn là vấn đề của tất cả các cá nhân, tập thể, mọi vùng, mọi khu vực ở khắp nơi trên Trái Đất.

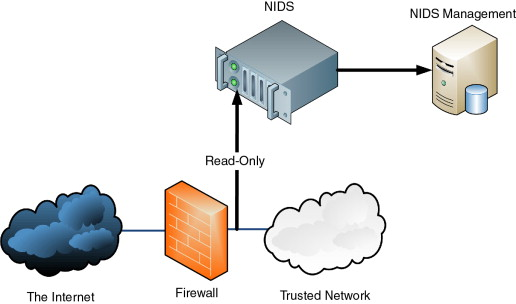
Với các hoạt động để phục vụ đời sống, loài người đang từng giờ từng phút thải vào môi trường không khí các chất độc hại, bụi, ... Thực trạng phát triển kinh tế - xã hội của các quốc gia trên thế giới trong thời gian qua đã có những tác động lớn đến môi trường, làm cho môi trường sống của con người bị thay đổi và ngày càng trở nên tồi tệ hơn. Những năm gần đây nhân loại đã phải quan tâm nhiều đến vấn đề ô nhiễm môi trường không khí đó là: sự biến đổi của khí hậu – nóng lên toàn cầu, sự suy giảm tầng ozone và mưa axít. Quá trình phát triển công nghiệp từ thế kỉ XVII đến nay, đặc biệt từ thế kỉ XX đã phá huỷ, gây tổn hại quá nặng nề đến các thành phần của môi trường. Vì thế, sang thế kỉ XXI này, việc bảo vệ các thành phần của môi trường đang là vấn đề cấp bách đối với toàn thể nhân loại.

Trong những năm gần đây quá trình công nghiệp hóa, hiện đại hóa đất nước ta diễn ra mạnh mẽ và đã thu được nhiều thành công đáng khích lệ. Cùng với quá trình phát triển của nền kinh tế, chất lượng không khí tại Việt Nam, đặc biệt là tại các đô thị lớn ngày càng suy giảm. Chất lượng không khí là lại một yếu tố ảnh hưởng trực tiếp đến sức khỏe con người, vì vậy việc tạo ra thiết bị theo dõi diễn biến chất lượng không khí, cảnh báo kịp thời cho cộng đồng là một yêu cầu rất cần thiết.

# TỔNG QUAN

## Lý do chọn đề tài

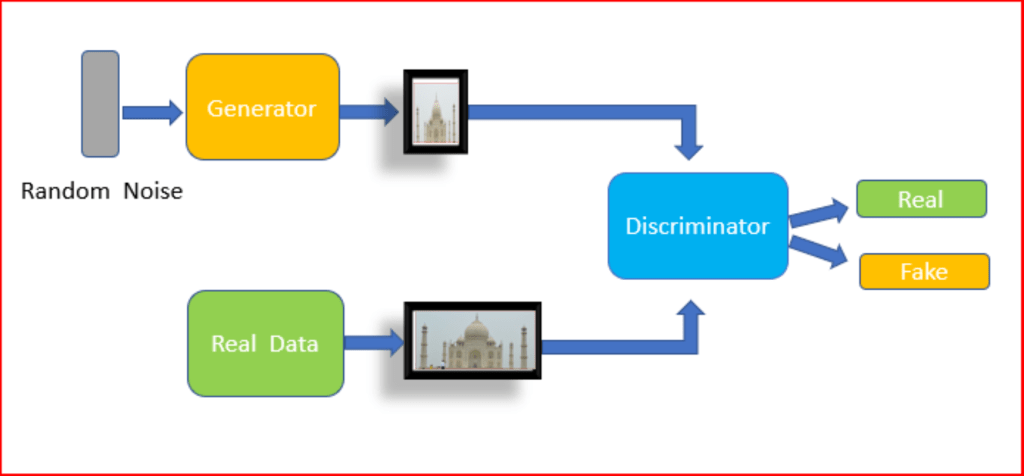
Với sự lan rộng của các mối đe dọa bảo mật trên internet, hệ thống phát hiện xâm nhập (IDS) trở thành công cụ thiết yếu để phát hiện và phòng tránh tấn công mạng được thể hiện dưới dạng lưu lượng độc hại. IDS giám sát lưu lượng mạng và đưa ra cảnh báo nếu lưu lượng không an toàn được xác định, phát hiện bởi bộ phân tích. Mục đích chính của IDS là phân loại giữa bản ghi mạng bình thường và bất thường thông qua các dữ liệu mà hệ thống có được trước đó, hoặc thông qua các phương pháp dự đoán.



Hình 1: IDS trong hệ thống mạng

Đối với các vấn đề phân loại, thuật toán học máy đã được áp dụng rộng rãi trong IDS và đạt được kết quả khả quan Các thuật toán phát hiện đã được sử dụng để giám sát và phân tích lưu lượng độc hại, bao gồm K-Nearest Neighbor, Support Vector Machine (SVM), Cây quyết định (Decision Tree), v.v [1]. Trong những năm gần đây, các thuật toán học sâu phát triển nhanh và thúc đẩy sự phát triển trong trong lĩnh vực phát hiện xâm nhập như Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Auto Encoder, v.v. [2]. Các thuật toán này giúp cải thiện độ chính xác và đơn giản hóa bài toán phát hiện xâm nhập [3] trong các hệ thống mạng.

Tuy nhiên, hệ thống phát hiện xâm nhập dần dần lộ ra lỗ hổng của nó khi gặp phải các mẫu đối kháng: input gần giống với input gốc nhưng được phân loại không chính xác [4]. Kẻ tấn công cố gắng đánh lừa để các mô hình phân loại sai mong muốn bằng cách sử dụng các mẫu lưu lượng độc đối kháng. Và các Mạng Đối kháng Tạo sinh GAN) là phương pháp được lựa chọn tiềm năng cho các cuộc tấn công đối kháng như vậy.



Hình 2 Nguyên tắc hoạt động chính của GAN

Goodfellow và cộng sự đã giới thiệu GAN, một khung để đào tạo các mô hình sinh dữ liệu (generative models) [5], với ý tưởng chính là hai mạng nơ-ron nhiều lớp Generator và Discriminator, chơi một trò chơi minimax để đưa đến một giải pháp tối ưu [6]. Bộ sinh (Generator) tìm hiểu cách thức dữ liệu được hình thành, sau đó đặt câu hỏi: Dựa vào giả định cách thức tạo ra dữ liệu, dữ liệu sẽ được phân vào nhóm nào. Bộ phân biệt (Discriminator) không quan tâm đến cách thức dữ liệu được tạo ra, chỉ phân loại dựa trên đầu vào dữ liệu. Với đặc điểm trên, mục đích của bộ sinh là đánh lừa bộ phân biệt, trong khi bộ phân biệt có vai trò phân loại dữ liệu đầu vào (đúng hoặc sai) để đưa ra phản hồi cho bộ sinh. Trò chơi kết thúc khi bộ phân biệt không thể xác định, phân biệt chính xác các dữ liệu đầu vào được tạo ra bởi bộ sinh.

GAN đã cho thấy sự tiên tiến của mình trong việc tạo ra hình ảnh, âm thanh và văn bản [7, 8, 9]. Chia sẻ đặc điểm tương tự của văn bản hoặc câu, bảo mật thông tin cũng là lĩnh vực tiềm năng ho GAN gần đây. Các nghiên cứu hiện tại đã sử dụng GAN để cải thiện phát hiện phần mềm độc hại hoặc tạo các ví dụ về phần mềm độc hại đối với các cuộc tấn công đe dọa hơn [10, 11]. Tuy vậy, ở thời điểm hiện tại, giới nghiên cứu bảo mật đang thiếu các nghiên cứu về GAN được sử dụng trong IDS. Chính vì vậy, đề tài này là hướng nghiên cứu tiềm năng giúp các IDS phát hiện và phòng ngừa các loại tấn công mới chưa được xác định.

## Tình hình nghiên cứu trong và ngoài nước

### Trong nước

Phân tích chất lượng không khí có khá nhiều nhóm nghiên cứu về vấn đề này, tuy nhiên phát triển thành các thiết bị phân tích không khí thì còn khá ít các nhóm thực hiện.



Hình 1.1 Hình ảnh của thiết bị phân tích khí

Theo những gì tìm hiểu được trước đây đã có một nhóm sinh viên của trường Đại học Bách khoa Hà Nội làm về đề tài này.



Hình 1.2 Thiết bị đo không khí của nhóm sinh viên Đại học Bách khoa Hà Nội

### Ngoài nước

Ô nhiễm không khí là tình trạng chung của Trái Đất, đặc biệt là các nước phát triển. Vì thế cho nên thiết bị phân tích không khí được phát triển từ khá lâu và không còn xa lạ đối với các nước trên thế giới.

Các thiết bị được sản xuất trên nhiều đất nước khác nhau, nhiều mẫu mã khác nhau.



Hình 1.3 Thiết bị AEROCET 531S của Met One Instruments - Mỹ



Hình 1.4 PM-1063SD của LUTRON – Đài Loan



Hình 1.5 Thiết bị đo lượng không khí Xiaomi PM2.5 – Trung Quốc

## Mục tiêu đề tài

Đề tài XÂY DỰNG HỆ THỐNG PHÂN TÍCH KHÔNG KHÍ, mục tiêu đặt ra là xây dựng hệ thống theo dõi từ xa, thiết kế và hiện thực một thiết bị đo chất lượng không khí có kích thước nhỏ, có thể dễ dàng di chuyển. Bao gồm cả phần cứng và phần mềm: chương trình xử lí các giá trị đầu ra của cảm biến.

Dựa vào công việc phải hoàn thành, nhóm đã đề ra mục tiêu:

* Tìm hiểu, thiết kế và hiện thực thiết bị:
  + Kết hợp cảm biến hạt bụi để đo được mật độ hạt lơ lửng trong một mét khối với sai số tối đa 10%.
  + Kết hợp cảm biến nhiệt độ và độ ẩm.
  + Kết hợp module thời gian thực để xem giờ.
  + Kết nối module wifi để gửi dữ liệu lên server.
* Xây dựng server:
  + Lưu trữ dữ liệu từ thiết bị đo.
  + Cung cấp các api cần thiết cho ứng dụng web và ứng dụng di động.
* Xây dựng ứng dụng trên điện thoại thông minh để theo dõi chất lượng không khí từ xa.
* Xây dựng ứng dụng web để quản lí các thiết bị và theo dõi chất lượng không khí một cách chi tiết hơn.

## Đối tượng và phạm vi nghiên cứu.

### Đối tượng nghiên cứu

Lập trình nhúng điều khiển kit esp32 wrover, cảm biến Shinyei PPD42NS, cảm biến nhiệt độ và độ ẩm SHT31, cảm biến thời gian thực RTC DS 1307.

Các giải thuật để tính toán các giá trị nhận được từ ngõ ra của các cảm biến.

Cách thực hiện giao tiếp truyền tải dữ liệu giữa kit và server.

Cách thực hiện giao tiếp truyền tải dữ liệu giữa điện thoại thông minh và server.

Xây dựng các chức năng của ứng dụng di động.

Xây dựng các chức năng của ứng dụng web.

### Phạm vi nghiên cứu

Thiết kế, thực hiện được thiết bị có thể đo được hạt lơ lửng. Sau đó gửi thông tin lên server, dựa vào những thông tin thu thập được ta có thể phân tích được mức độ ô nhiễm tại một khu vực xác định.

Thời gian nghiên cứu: 4 tháng (từ tháng 09/2019 đến tháng 12/2019).

Chi phí dự tính: 2 triệu đồng.

## Ý nghĩa khoa học và thực tiễn của đề tài

### Ý nghĩa khoa học

Mô phỏng thiết bị đo hạt bụi lơ lửng.

Mô phỏng được hệ thống theo dõi chất lượng không khí.

### Ý nghĩa thực tiễn

Chế tạo được thiết bị đo mật độ hạt lửng lơ dùng cảm biến PPD42NS, thiết bị chạy ổn định, có khả năng hiển thị kết quả trên máy tính.

Thiết bị có thể được ứng dụng để đo mật độ hạt lơ lửng tại nhà, các văn phòng cơ quan, khu công nghiệp,… để từ đó có những biện pháp xử lý khi mật độ hạt này vượt quá ngưỡng cho phép để đảm bảo an toàn về môi trường và sức khỏe cho con người.

Thiết bị là mô hình để khảo sát và lấy số liệu cho bài toán tìm kiếm phương pháp nâng cao độ chính xác cho thiết bị đo dùng PPD42NS, SHT31.

# CƠ SỞ LÝ THUYẾT

## Chỉ số chất lượng không khí (Air Quality Index- AQI)

### Khái niệm

Chỉ số chất lượng không khí AQI là chỉ số đại diện cho nồng độ của một nhóm các chất ô nhiễm gồm CO, NOx, SO2, O3 và bụi, nhằm cho biết tình trạng chất lượng không khí và mức độ ảnh hưởng đến sức khỏe con người.

### Giới thiệu về hạt bụi lơ lửng

Hạt bụi lơ lửng (Particulate Matter) là khái niệm dùng để chỉ những hạt dạng lỏng hoặc rắn rất nhỏ trôi nổi trong không khí. Các hạt này có thể do nhiều chất khác nhau tạo nên, như cacbon, sulfua, khí nitơ, và các hợp chất kim loại, v.v… Đối với các nhà khoa học môi trường, các hạt trôi nổi này nhằm chỉ những chất ô nhiễm rất nhỏ trong không khí, một trong những nguyên nhân chủ yếu gây ra ô nhiễm không khí. [4]

### Giới thiệu về PM2.5

Theo nghiên cứu của các nhà khoa học, những hạt có đường kính bé hơn 10 micromet là những hạt có thể bị con người hít vào khi thở, chúng sẽ tích tụ trên phổi, gây ra nguy hại cho sức khỏe con người.

Những hạt có đường kính bé hơn 2,5 micromet là những hạt đặc biệt nguy hiểm bởi vì chúng xâm nhập trực tiếp vào các túi phổi.

Vì vậy, các nhà khoa học dùng chỉ số PM 2.5 để biểu thị hàm lượng tiêu chuẩn của các hạt trôi nổi trong một mét khối không khí. Chỉ số này càng cao cũng có nghĩa là ô nhiễm không khí ở nơi đó càng nghiêm trọng. [4]

### Giới thiệu về AQI

AQI được tính toán cho từng thông số. Mỗi thông số sẽ xác định được một giá trị AQI cụ thể, giá trị AQI cuối cùng là giá trị lớn nhất trong các giá trị AQI của mỗi thông số. (ở đây không dùng phương pháp tính giá trị trung bình vì chỉ cần có một thông số vượt quá ngưỡng cho phép là có thể kết luận môi trường đã bị ô nhiễm và có ảnh hưởng đến sức khỏe của cộng đồng.) [16]

### Công thức tính toán AQI

Hiện nay trên thế giới rất nhiều quốc gia đã xây dựng phương pháp tính toán và công bố AQI cho cộng đồng.

Trong đề tài báo cáo này chúng em áp dụng phương pháp tính toán theo chuẩn của Hoa Kỳ.

Công thức tính AQI:

[16]

Với:

C: mật độ chất ô nhiễm đo được

Clow: mật độ chất ô nhiễm mức thấp ≤ C

Chigh: mật độ chất ô nhiễm mức cao ≥ C

Ilow: chỉ số tương ứng của Clow

Ihigh: chỉ số tương ứng của Chigh

Bảng 2.1 Bảng thông số của bụi PM2.5(µg/m3)

|  |  |  |  |
| --- | --- | --- | --- |
| Clow | Chigh | Ilow | Ihigh |
| 0 | 12 | 0 | 50 |
| 12.1 | 35.4 | 51 | 100 |
| 35.5 | 55.4 | 101 | 150 |
| 55.5 | 150.4 | 151 | 200 |
| 150.5 | 250.4 | 201 | 300 |
| 250.5 | 350.4 | 301 | 400 |
| 350.5 | 500.4 | 401 | 500 |

### Công thức tính toán mật độ ô nhiễm

Do cảm biến được sử dụng để đo chất lượng không khí là bộ đếm hạt laze, nên cần có một thuật toán để chuyển số hạt đếm được (đơn vị hạt/0.01cf) sang tổng khối lượng các hạt trên một mét khối (đơn vị ug/m3). Điều quan trọng cần lưu ý là chuyển đổi này chỉ là chuyển đổi xấp xỉ, bởi vì không thể định lượng được các tính chất chính xác của các hạt trong hàng ngàn hạt vi mô.

Trong đề tài này, thuật toán được giả định:

* Tất cả các hạt đều hình cầu, với khối lượng riêng .
* Bán kính của các hạt là .

Công thức để đếm số hạt (đơn vị pcs/0.01cf):

Trong đó ratio là tỉ lệ giữa số giây cảm biến đo được các hạt PM2.5 và tổng số giây cảm biến đo được, có công thức.

Công thức tính mật độ ô nhiễm (tổng khối lượng các hạt trên một mét khối):

[17]

Trong đó:

* Vol25 là thể tích của 1 hạt bụi, có công thức:

Các hệ số hiệu chỉnh đã được sử dụng để tăng độ chính xác của phương thức chuyển đổi này, dựa trên sự hiện diện của độ ẩm và mưa. Cụ thể, thuật toán cải tiến có thể được mô tả bằng mối quan hệ F = O x H x C [17], trong đó O là tổng khối lượng các hạt trên một mét khối (mô tả ở trên), H là tỉ lệ độ ẩm tương đối, C là hệ số hiệu chỉnh và F là nồng độ đầu ra cuối cùng. Bảng 2 cho thấy các giá trị của H và C trong điều kiện khô (không mưa), trong khi Bảng 3 cho thấy các giá trị của H và C khi có mưa.

Bảng 2.2 Các giá trị của H và C trong điều kiện khô (không mưa)

|  |  |
| --- | --- |
| H | C |
| 0-39% | 13 |
| 40-49% | 8 |
| 50-59% | 6 |
| 60-69% | 4 |
| 70-79% | 1.75 |
| 70-89% | 1.5 |
| 90-100% | 1 |

Bảng 2.3 Các giá trị của H và C trong khi có mưa

|  |  |
| --- | --- |
| H | C |
| 50-59% | 2.5 |
| 60-69% | 2.2 |
| 70-79% | 2 |
| 80-89% | 1.4 |
| 90-100% | 0.8 |

### Quy trình tính toán và sử dụng AQI trong đánh giá chất lượng môi trường không khí xung quanh

Quy trình tính toán và sử dụng AQI trong đánh giá chất lượng môi trường

không khí xung quanh bao gồm các bước sau:

1. Thu thập, tập hợp số liệu quan trắc từ thiết bị liên tục (số liệu đã qua xử lý).

2. Tính toán các chỉ số chất lượng không khí theo công thức.

3. Tính toán chỉ số chất lượng không khí theo giờ/theo ngày.

4. So sánh chỉ số chất lượng không khí với bảng xác định mức cảnh báo ô nhiễm môi trường không khí và mức độ ảnh hưởng tới sức khỏe con người. [13]

Bảng 2.4 Bảng xác định mức cảnh báo ô nhiễm môi trường ở Việt Nam

|  |  |  |
| --- | --- | --- |
| Khoảng giá trị AQI | Chất lượng không khí | Ảnh hưởng sức khỏe |
| 0-50 | Tốt | Không ảnh hưởng đến sức khỏe. |
| 51-100 | Trung bình | Nhóm nhạy cảm[[1]](#footnote-1) nên hạn chế thời gian ở bên ngoài. |
| 101-150 | Kém | Nhóm nhạy cảm cần hạn chế thời gian ở bên ngoài. |
| 151-200 | Xấu | Tác động ít ỏi tới các bệnh nhân mắc bệnh hô hấp. |
| 201-300 | Rất xấu | Nhóm nhạy cảm tránh ra ngoài. Những người khác hạn chế ở bên ngoài. |
| Trên 300 | Nguy hiểm | Tác động nghiêm trọng đối với những người mắc bệnh hô hấp. |

## Hệ điều hành FreeRTOS

### Hệ điều hành là gì ?

Hệ điều hành là một phần mềm chạy trên máy tính, dùng để điều hành, quản lý các thiết bị phần cứng và các tài nguyên phần mềm trên máy tính. Dựa trên nhu cầu và phần cứng khác nhau, nhà sản xuất tích hợp hệ điều hành tương ứng vào sản phẩm của mình, như:

* Hệ điều hành cho máy tính cá nhân (MacOS, Linux, Windows, …)
* General Purpose Operating System (GPOS).
* Hệ điều hành cho Server, Router, Switch (Cisco IOS, …)
* Hệ điều hành cho hệ thống nhúng (FreeRTOS, µC/OS, …)
* …

### FreeRTOS là gì ?

FreeRTOS là hệ điều hành thời gian thực dùng cho ứng dụng nhúng, nó có đầy đủ chức năng của một hệ điều hành: khởi tạo task, quản lý task, quản lý tranh chấp, quản lý vùng nhớ, giao tiếp giữa các task và hỗ trợ I/O. Vậy hệ điều hành thời gian thực là gì?

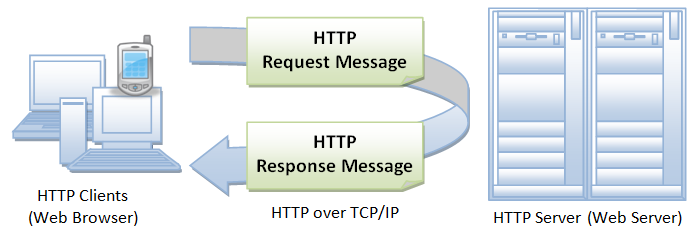
Hệ điều hành thời gian thực là hệ điều hành đa nhiệm – multitasking dùng cho các ứng dụng thời gian thực. [3]

## Giao thức HTTP

HTTP (Hyper Text Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web – WWW.

HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

Sơ đồ hoạt động của HTTP:



Hình 2.6 Mô hình Client-Server

HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

HTTP là một stateless protocol. Hay nói cách khác, request hiện tại không biết những gì đã hoàn thành trong request trước đó.

HTTP cho phép tạo các yêu cầu gửi và nhận các kiểu dữ liệu, do đó cho phép xây dựng hệ thống độc lập với dữ liệu được chuyển giao.

## Giao thức TCP/IP

Bộ giao thức TCP/IP là một bộ các giao thức truyền thông cài đặt chồng giao thức mà internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Transmission Control Protocol - Giao thức điều khiển truyền vận) và IP (Internet Protocol - Giao thức Internet). [2]

BỘ GIAO THỨC là định nghĩa của các giao thức còn CHỒNG GIAO THỨC là một hình thức xử lý bộ giao thức bằng phần mềm.

Các giao thức được phân chia thành các tầng, Trong đó TCP/IP có 4 tầng, mỗi tầng lại sử dụng các giao thức ở tầng dưới để đạt được mục đích của mình.

Layer 1. Network Access Layer

Network Access Layer xác định chi tiết về cách thức dữ liệu được gửi qua mạng, bởi các thiết bị phần cứng trực tiếp giao tiếp với môi trường mạng, chẳng hạn như cáp đồng trục, cáp quang hay dây đồng xoắn đôi. Các giao thức bao gồm trong Network Access Layer là Ethernet, Token Ring, FDDI, X.25, Frame Relay ...vv

Layer 2. Internet Layer

Internet Layer đóng gói dữ liệu vào các gói dữ liệu được biết đến dưới dạng các gói tin thông giao thức Internet Protocol, chứa địa chỉ nguồn và đích (địa chỉ logic hoặc địa chỉ IP) được sử dụng để chuyển tiếp các gói tin giữa các máy chủ và qua các mạng.

Layer 3. Transport Layer

Mục đích của Transport Layer là cho phép các thiết bị trên máy chủ nguồn và đích đến trao đổi dữ liệu. Transport Layer sẽ xác định mức độ service và trạng thái của kết nối được sử dụng khi vận chuyển dữ liệu.

Trong đó có giao thức chính trong lớp Transport là TCP (Transmission Control Protocol)

Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

Layer 4. Application Layer

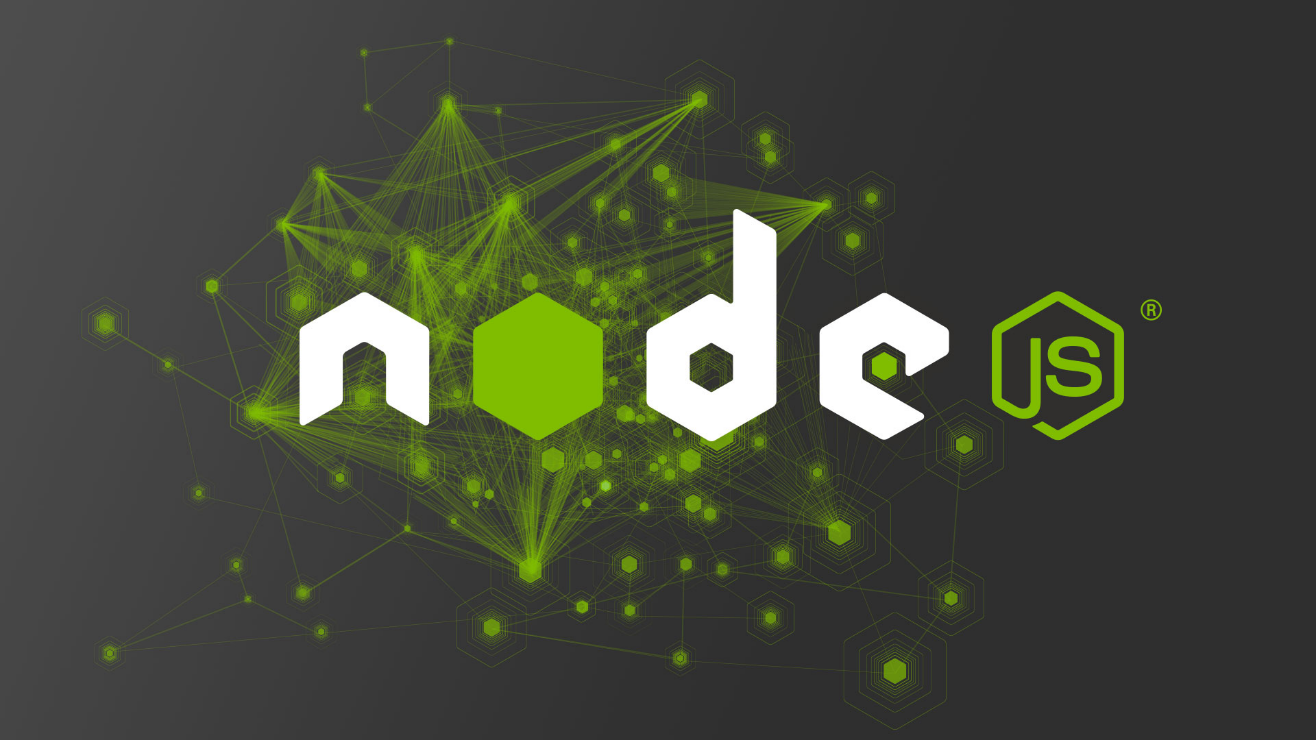
Các thực thể của lớp Application cung cấp các ứng dụng cho phép người dùng trao đổi dữ liệu ứng dụng qua mạng.

Một số ứng dụng thường gặp của chồng giao thức TCP/IP: FTP (File Transfer Protocol), DNS.

## Nền tảng (platform) NodeJS

### NodeJS là gì ?

Nodejs là một nền tảng (platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng. [10]



Hình 2.7 Logo NodeJS

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

NodeJS đang được các công ty lớn như Amazon, Ebay, Linkedin, Microsoft, Payal, Trello, Uber và còn nhiều cái tên nổi tiếng khác nữa sử dụng.

### Ưu điểm của NodeJS

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Các ứng dụng NodeJS được viết bằng JavaScript - một ngôn ngữ khá thông dụng. JavaScript là một ngôn ngữ dựa trên sự kiện, vì vậy bất cứ thứ gì xảy ra trên server đều tạo ra một sự kiện non-blocking. Mỗi kết nối mới sinh ra một sự kiện; dữ liệu nhận được từ một upload form sinh ra một sự kiện data-received; việc truy vấn dữ liệu từ database cũng sinh ra một sự kiện. Trong thực tế, điều này có nghĩa là một trang web Node.js sẽ chẳng bao giờ bị khóa (lock up) và có thể hỗ trợ cho hàng chục nghìn user truy cập cùng lúc. Node.js đóng vai trò của server - Apache - và thông dịch mã ứng dụng chạy trên nó. Giống như Apache, có rất nhiều module (thư viện) có thể được cài đặt để bổ sung thêm các đặc trưng và chức năng - như lưu trữ dữ liệu, hỗ trợ file Zip, đăng nhập bằng Facebook, hoặc các cổng thanh toán.

Nodejs chạy đa nền tảng phía server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

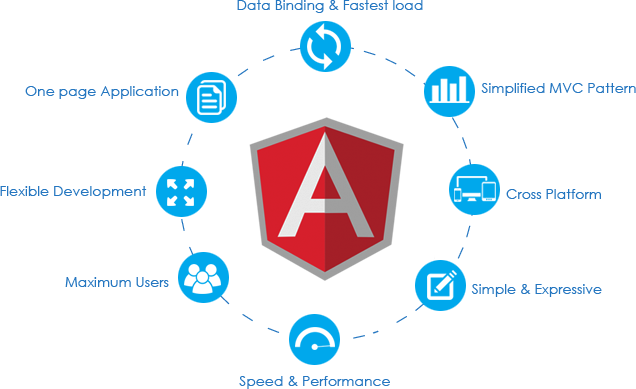
Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn các package đều hoàn toàn miễn phí.

Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

## Angular

### Angular là gì ?

AngularJs là một Javascript Framework được tạo ra để xây dựng các ứng dụng web động, nó thường được sử dụng để tạo ra các ứng dụng một trang (Single Page Application - SPA). Và hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML (các thuộc tính theo quy tắc của Angular). AngularJS là một nền tảng miễn phí và được hàng ngàn lập trình viên trên thế giới ưa chuộng. [8]



Hình 2.8 Angular

### Các phiên bản của Angular

Thực tế là Angular có 2 phiên bản hoàn toàn khác nhau, phiên bản 1 và phiên bản khác 1.

Phiên bản 1 có tên chính thức là AngularJS, bắt đầu từ phiên bản 2 nó có tên chính thức là Angular.

### Material design với Angular

Material Design là một ngôn ngữ thiết kế được Google tạo ra nhằm “phát triển một hệ thống nền duy nhất cho phép xây dựng các trải nghiệm đồng bộ giữa nhiều loại và kích thước thiết bị khác nhau. Việc thiết kế cho thiết bị di động được đặt lên hàng đầu, nó còn phải chú trọng đến việc nhập liệu bằng cảm ứng, giọng nói, chuột, bàn phím”.

Material cung cấp rất nhiều built-in module cho project angular, giúp việc phát triển ứng dụng với angular được đồng nhất, nhanh chóng và dễ dàng hơn rất nhiều.

Muốn sử dụng thành phần nào của material chúng ta chỉ cần import thành phần đó vào project. Các thành phần thông dụng của material như: button, sidenav, toolbar, card, menu, … hỗ trợ rất nhiều tính năng hữu ích giúp việc thiết kế và phát triển dễ dàng hơn.

Tài liệu hướng dẫn chi tiết về cách import thư viện và cách sử dụng các thành phần của material tại: [*https://material.angular.io/*](https://material.angular.io/)

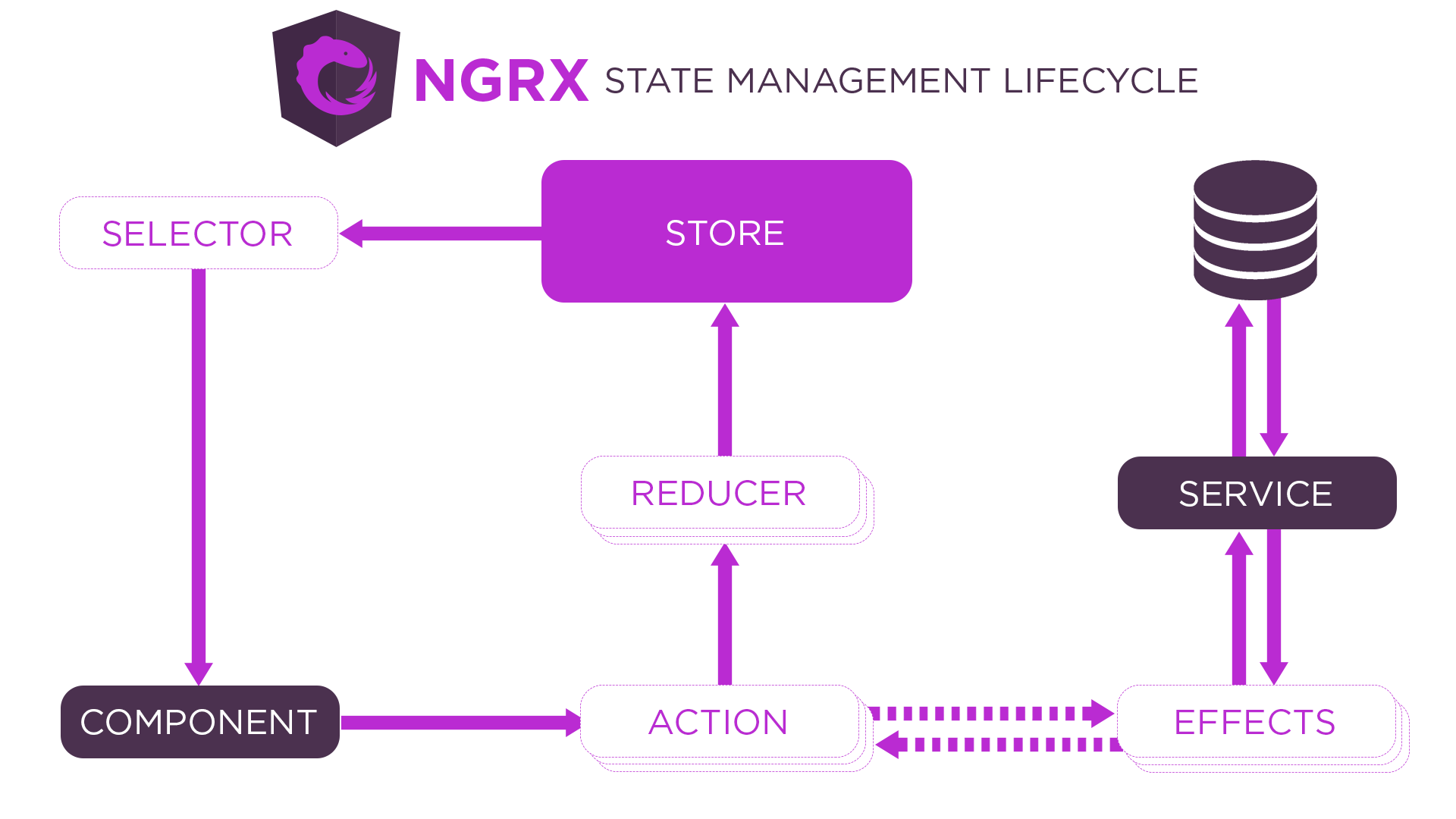
### NgRx

NgRx là một framework được xây dựng để phát triển “Reactive Applications” trong một ứng dụng Angular. NgRx cung cấp cho ta các tác vụ quản lý state để xây dựng một ứng dụng Angular với khả năng duy trì và mở rộng rất cao.

NgRx gồm 4 thành phần chính:

* Store: nơi chứa các State.
* Action: thể hiện các hành động, thao tác khi muốn thay đổi các State.
* Reducer: thực hiện việc chuyển đổi State từ trạng thái này sang trạng thái khác.
* Selector: thể hiện vùng chọn của các State được lưu trữ trong Store.

Ngoài ra NgRx còn có thêm nhiều khái niệm khác như Effect, Adapter...



Hình 2.9 Biểu đồ luồng dữ liệu NgRx

Giả sử điểm bắt đầu là Component và kết thúc là Store, từ Component ta muốn lấy ra State trong Store thì các bước hoạt động của nó như sau:

* Từ Component (thông qua giao diện như nhấn button hoặc điền vào form) ta tương tác với Reducer thông qua Action.
* Từ Action, Reducer sẽ làm một công việc tương ứng với Action đó.
* Từ Reducer, State sẽ được đưa vào Store.
* Từ Store, Component lấy ra State thông qua Selector.

Sử dụng NgRx ứng dụng của chúng ta sẽ có một Store chứa dữ liệu và Store này là duy nhất. Tất cả Component, Service trong ứng dụng sẽ dùng chung dữ liệu đó.

Tóm lại Angular cùng với các thư viện hỗ trợ của nó sẽ giúp phát triển ứng dụng web một cách hiệu quả và nhanh chóng. Material cùng với tài liệu hướng dẫn chi tiết của nó giúp hỗ trợ xây dựng giao diện. RxJS hỗ trợ xử lý bất đồng bộ, gọi đến các API để lấy dữ liệu. NgRx tạo ra một Store lưu trữ dữ liệu của ứng dụng. Tuy nhiên để hiểu rõ và sử dụng hiệu quả Angular cùng các thư viện của nó, đòi hỏi nhà phát triển cần nhiều thời gian để nghiên cứu.

## InfluxDB

### Time Series Database là gì ?

Time-series Data: là một chuỗi các điểm dữ liệu, thường là phép đo hoặc sự kiện được theo dõi, lấy mẫu và tổng hợp theo thời gian [9]. Dữ liệu có thể là số liệu máy chủ, giám sát hiệu suất ứng dụng, dữ liệu mạng, dữ liệu cảm biến, sự kiện, nhấp chuột, giao dịch trong một thị trường và nhiều loại dữ liệu phân tích khác... Phân tích chuỗi dữ liệu thời gian có mục đích nhận dạng và tập hợp lại các yếu tố, những biến đổi theo thời gian mà nó có ảnh hưởng đến giá trị của biến quan sát.

Trong Time-series Data có hai loại chính:

* Chuỗi thời gian thông thường (regular time series), loại thông thường được gọi là số liệu.
* Chuỗi thời gian bất thường (event) là những sự kiện.

Time-series Data được ứng dụng rộng rãi trong các lĩnh vực: IoT, DevOps, dự báo kinh tế, tính toán doanh số bán hàng, phân tích chứng khoán, phân tích thị trường, kiểm soát quy trình và chất lượng, phân tích điều tra, ...

Time-series Data rất quan trọng đối với cuộc sống hiện đại và là một phần quan trọng của big data hiện nay khi một doanh nghiệp làm dữ liệu thời gian thực có thể nhận tới hàng gigabyte mỗi giờ. Vì vậy, cần có cách thu thập và xử lý hợp lý với loại dữ liệu này.

Time-series Database: là cơ sở dữ liệu được xây dựng đặc biệt để xử lý các số liệu và sự kiện hoặc phép đo được đóng dấu thời gian. Time-series database được tối ưu hóa để đo lường sự thay đổi theo thời gian.

Tập dữ liệu chuỗi thời gian chủ yếu có ba đặc điểm sau:

* Dữ liệu mới luôn được lưu trữ dưới dạng một mục mới (INSERT chứ không phải UPDATE).
* Dữ liệu được lưu trữ theo thứ tự thời gian.
* Thời gian là một trục chính (khoảng thời gian có thể thường xuyên hoặc không đều).

Những time-series database hàng đầu thế giới: InfluxDB, Kdb+, Graphite, RRDTool, Prometheus, OpenTSDB, Druid, ...

### InfluxDB



Hình 2.10 Logo Influxdb

Influxdb thường được lựa chọn trong các trường hợp khi phải lưu trữ, tổ chức những dữ liệu lớn được đánh dầu bằng các nhãn thời gian (DevOps monitoring, IoT sensor data, ...)

Những tính năng chính mà Influxdb hỗ trợ:

* Có các API đọc ghi dễ hiểu, hiệu suất cao.
* Plugin hỗ trợ cho các giao thức nhập dữ liệu khác.
* Câu lệnh query tương đồng với SQL.
* Các truy vấn liên tục tự động tính toán dữ liệu tổng hợp để làm cho các truy vấn hiệu quả hơn.
* Influxdb có cả phiên bản mã nguồn mở và phiên bản cho enterprise.

Cũng bởi vì những đặc điểm và tính năng hữu ích của time-series database nên trong đề tài này, nhóm quyết định sử dụng InfluxDB một trong những time-series database nổi tiếng nhất để lưu trữ và phân tích dữ liệu từ các cảm biến.

## MongoDB

### MongoDB là gì ?



Hình 2.11 Logo MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, thuộc NoSql, được viết bởi ngôn ngữ C++. Vì được viết bằng C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị cơ sở dữ liệu quan hệ hiện nay. [11]

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như cơ sở dữ liệu quan hệ nên truy vấn sẽ rất nhanh.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

### Ưu điểm của MongoDB

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster.

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). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

### Nhược điểm của MongoDB

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điều này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

Với những ưu điểm của mongodb rất có thể trong tương lai nó sẽ thay thế các hệ quản trị cơ sở dữ liệu hiện nay. MongoDB thích hợp với các hệ thống realtime với yêu cầu truy vấn nhanh nên nhóm quyết định sử dụng mongodb để lưu trữ thông tin đăng nhập người dùng, thông tin các thiết bị, số liệu thống kê của hệ thống.

## Smart Config

Smart Config là công nghệ được TI tạo ra nhằm cấu hình cho phép thiết bị CC3000 kết nối với mạng WiFi một cách đơn giản nhất. Họ phát triển ứng dụng điện thoại, PC nhằm gửi thông tin như tên và mật khẩu mạng WiFi cho CC3000. [7]

ESPRESSIF cũng phát triển công nghệ tương tự như TI cho ESP32 của họ.

SmartConfig có thể truyền SSID và keyphrase từ một mạng wifi được bảo mật đến Esp32 đang “listen” mà chưa có khả năng giải mã các gói tin trên mạng đó.

Esp32 ban đầu chưa được kết nối với mạng, nhưng có khả năng giám sát lưu lượng trên mạng. Do đó, ứng dụng SmartConfig không thể gửi các gói tin trực tiếp đến esp32 được. Thay vào đó, nó sẽ gửi các gói tin UDP tới các máy hiện đang kết nối trong mạng - các Access Point AP. Các AP có thể không muốn nhận các gói tin, tuy nhiên điều đó không quan trọng, điều chúng ta cần là các gói tin UDP được truyền trên mạng.

Esp32 có thể theo dõi lưu lượng nhưng không thể giải mã các gói tin được bảo mật bằng chuẩn bảo mật Wi-Fi như WEB, WAP, WAP2...Nó thậm chí còn không chắc chắn các gói tin được mã hóa chứa dữ liệu UDP. Vậy làm thế nào để esp32 nhận ra các gói tin UDP?

Phép màu nằm ở độ dài của gói tin. Về cơ bản, SmartConfig không giải mã nội dung của gói tin đang được truyền mà nó giải mã dựa vào độ dài của gói tin. Wifi mã hóa gói tin sẽ thay đổi độ dài của gói tin, nhưng theo một cách nhất quán, nghĩa là nó thêm L bytes so với kích thước của mỗi gói tin, trong đó L là hằng số.

Ứng dụng SmartConfig cần mã hóa SSID và keypharase vào độ dài gói tin của chuỗi các gói UDP. Esp32 có thể giám sát các gói mã hóa và kích thước của chúng. Trong môi trường nhiều mạng Wi-Fi thì sao? esp32 có thể giám sát lưu lượng của nhiều mạng và nhiều gói tin như vậy, làm thế nào để nó tập trung giám sát các gói tin có liên quan? Về cơ bản, các ứng dụng SmartConfig sẽ gửi các gói UDP theo chu kì lặp lại, esp32 sẽ giám sát mạng và tìm thấy các gói được gửi lặp đi lặp lại và lấy thông tin từ đó.

## Android

### Android là gì ?

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android, Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.

Mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác.

Với rất nhiều đặc điểm đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu.

### Ưu điểm của Android

Trong khi các hệ điều hành như iOS, WindowPhone sử dụng mã nguồn đóng thì Android lại đi ngược lại bằng cách mở mã nguồn và cung cấp cho người dùng tùy ý chỉnh sửa, mặc sức “xào nấu” mà không có sự can thiệp hay cấm cản từ Google.

Thêm một điều mà các hệ điều hành khác như iOS, WindowPhone không có được đó là việc tự thiết kế giao diện màn hình HomeScreen. Ta có thể tự kéo thả những icon, Widget, đồng hồ, thời tiết, tin tức, bản đồ … và bất cứ gì cũng có thể để ra ngoài màn hình chính.

Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Nếu sử dụng một thiết bị Android, chắc hẳn ta sẽ không phải bận tâm vì “thiếu ứng dụng để cài” mà cái chúng ta sẽ bận tâm đó là “không đủ bộ nhớ để cài”.

### Nhược điểm của Android

Sau một thời gian sử dụng thiết bị Android, các dữ liệu của ứng dụng đã được lưu trong bộ nhớ đệm, mà máy không có quá trình tự động để dọn dẹp chúng đi. Những dữ liệu này nếu không được dọn dẹp sẽ vẫn nằm ở đó và chiếm bộ nhớ của máy gây ra tình trạng giật, lag.

Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.

Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.

Đến tháng 11/2017, 75% smartphone trên toàn thế giới chạy hệ điều hành Android, tỉ lệ được cung cấp bởi Newzoo. Thống kê về lượng smartphone được vận chuyển trên toàn cầu từ Tập đoàn Dữ liệu Quốc tế (IDC) cũng phản ánh điều tương tự. Hãng ước tính thiết bị Android chiếm 85.1 % tổng số smartphone xuất xưởng trong năm 2017, còn tỉ lệ dành cho iOS chỉ là 14.8%. Nên nhóm chọn android làm nền tảng để phát triển ứng dụng di động.

## Retrofit

### Retrofit là gì ?

[Retrofit](https://square.github.io/retrofit/) là một HTTP client type-safe cho Android và Java. Retrofit giúp dễ dàng kết nối đến một dịch vụ REST trên web bằng cách chuyển đổi API thành Java Interface.

Thư viện mạnh mẽ này giúp bạn dễ dàng xử lý dữ liệu JSON hoặc XML sau đó phân tích cú pháp thành Plain Old Java Objects (POJOs). Tất cả các yêu cầu GET, POST, PUT, PATCH, và DELETE đều có thể được thực thi.

Giống như hầu hết các phần mềm mã nguồn mở khác, Retrofit được xây dựng dựa trên một số thư viện mạnh mẽ và công cụ khác. Đằng sau nó, Retrofit làm cho việc sử dụng [OkHttp](http://square.github.io/okhttp/) (từ cùng một nhà phát triển) để xử lý các yêu cầu trên mạng. Ngoài ra, Retrofit không tích hợp bất kỳ một bộ chuyển đổi JSON nào để phân tích từ JSON thành các đối tượng Java. Thay vào đó nó đi kèm với các thư viện chuyển đổi JSON sau đây để xử lý điều đó:

• Gson: com.squareup.retrofit:converter-gson

• Jackson: com.squareup.retrofit:converter-jackson

Và đối với XML, Retrofit hỗ trợ:

• Simple Framework: com.squareup.retrofit2:converter-simpleframework

### Vậy tại sao nên sử dụng Retrofit?

Việc phát triển thư viện HTTP type-safe của chính bạn để giao tiếp với REST API có thể thật sự là một điều khó khăn: bạn phải xử lý nhiều chức năng chẳng hạn như tạo các kết nối, lưu trữ cache, thử lại các yêu cầu thất bại, phân tích luồng phản hồi, xử lý lỗi, và nhiều hơn nữa. Retrofit, mặt khác, được hoạch định rất tốt, có tài liệu hướng dẫn và kiểm đã được kiểm thử một thư viện đã được thử nghiệm rất nhiều trong thực tế sẽ giúp bạn tiết kiệm rất nhiều thời gian quý báu và giảm bớt căng thẳng.

## GSON

Gson là một thư viện Java cho phép người sử dụng có thể chuyển đổi từ một đối tượng Java sang JSON và cũng có thể chuyển đổi từ một đối tượng JSON sang Java. Gson có thể làm việc với đối tượng Java tùy ý bao gồm các đối tượng tồn tại sẵn mà bạn không có mã nguồn của chúng.

# HIỆN THỰC HỆ THỐNG

## Thiết kế hệ thống

Mục tiêu cuối cùng của hệ thống là theo dõi chất lượng không khí thông qua ứng dụng trên điện thoại thông minh được đo bằng thiết bị AirNow. Với chiếc điện thoại thông minh người sử dụng có thể theo dõi được chất lượng của không khí như xem nhiệt độ, độ ẩm, mật độ hạt bụi, chỉ số chất lượng không khí v.v… Để làm được điều này nhóm đã xây dựng một thiết bị theo dõi AirNow, hệ thống server và phần mềm sử dụng hệ điều hành Android để người sử dụng có thể tương tác như sơ đồ hệ thống bên dưới.

A screenshot of a cell phone

Description automatically generated

Hình 3.12 Sơ đồ hệ thống theo dõi chất lượng không khí AirNow

* Server: đóng vai trò là bộ phận lưu trữ, vận hành chính những dữ liệu của một hệ thống.
* Điện thoại thông minh: là nơi người dùng tương tác với hệ thống. Trên điện thoại thông minh có phần mềm đã được viết sẵn, phần mềm này tạo ra một giao diện cho người dùng tương tác, theo dõi chất lượng của không khí tại thời điểm hiện tại, có thể thiết lập kết nối wifi cho thiết bị. Tuy nhiên, người dùng phải kết nối điện thoại với mạng internet để có thể sử dụng phần mềm.
* Thiết bị AirNow: là thiết bị có thể giao tiếp với server thông qua wifi. Thiết bị có tích hợp các module để thực hiện thu thập và hiển thị thông tin nhiệt độ, độ ẩm, chỉ số chất lượng không khí, mật độ hạt bụi và xem thời gian thực.
* Ứng dụng Web: là nơi để quản lí các thiết bị, quản trị viên có thể xem được biểu đồ của dữ liệu một cách chi tiết hơn.

## Giải pháp phần cứng

### Sơ đồ khối

A screenshot of a cell phone

Description automatically generated

Hình 3.13 Sơ đồ khối của thiết bị

### Cách thức hoạt động

Khi hoạt động lần đầu tiên, thiết bị sẽ trống rỗng không có dữ liệu nào được lưu ở bộ nhớ điện tĩnh. Người dùng cần phải cài đặt wifi cho thiết bị thông qua một chiếc di động thông minh có kết nối wifi. Sau khi kết nối thành công thì SSID và mật khẩu của wifi sẽ được lưu lại vào bộ nhớ điện tĩnh của thiết bị, việc này làm cho thiết bị có thể tự động kết nối vào wifi sẵn có ở các lần hoạt động sau.

Thiết bị liên tục thu thập các thông tin như nhiệt độ, độ ẩm, chỉ số chất lượng không khí và mật độ hạt bụi. Nhiệt độ và độ ẩm được đọc từ cảm biển SHT31. Esp32 sẽ tính toán để có được chỉ số chất lượng không khí từ dữ liệu lấy được qua cảm biến bui PPD42NS. Sau một khoảng thời gian mặc định thiết bị sẽ tự động gửi dữ liệu lên server. Song song với việc thu thập thông tin thiết bị còn trực tiếp hiển thị kết quả đo được lên một màn hình LCD để người dùng có thể theo dõi trực tiếp mà không cần thông qua internet. Thiết bị còn có thể hiển thị được thời gian thực trên màn hình LCD.

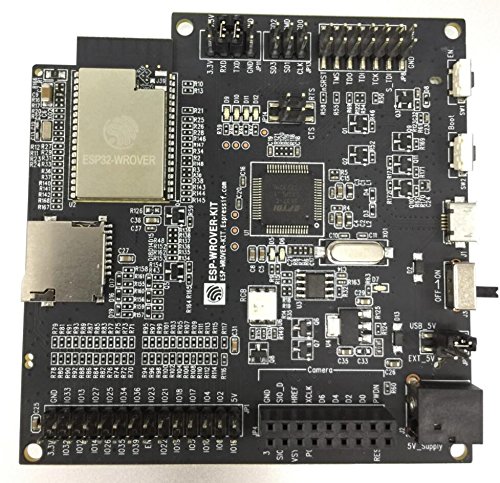
Thiết bị sử dụng nguồn điện 5V được cấp bởi pin sạc, với cổng sạc Micro USB rất phổ biến hiện nay nên dễ dàng trong việc sửa chữa hay cấp nguồn cho thiết bị.

### ESP32 WROVER KIT

ESP32-WROVER KIT là một kit tích hợp module ESP32-WROVER của Espressif Systems cho các ứng dụng wifi, bluetooth, low enegry. ESP32-WROVER thích hợp cho các ứng dụng cảm biến cần năng lượng thấp, như mã hóa tiếng nói, streaming video, mã hóa-giải mã MP3.

Kit được hỗ trợ SDK firmware từ hãng và rất thích hợp cho việc phát triển các dự án mã nguồn mở.

Kit có thể được sử dụng để điều khiển các cảm biến trong các dự án IoT, các thiết bị kết nối wifi, bluetooth, các dự án tự động hóa, nhà thông minh, streaming video với camera, mã hóa và xử lí giọng nói,… Việc phát triển các dự án mã nguồn mở với module thích hợp cho cả kĩ sư phần cứng và phần mềm.



Hình 3.14 ESP32 WROVER KIT

Bảng 3.5 Thông số kỹ thuật của ESP32 WORVER KIT

|  |  |
| --- | --- |
| Điện áp cung cấp | 3.3V |
| Tiêu chuẩn wifi | 802.11b/g/n, với tần số 2.4GHz |
| Hỗ trợ bảo mật | WPA/WPA2 |
| Tích hợp giao thức | TCP/IP |
| Tốc độ truyền Serial (Baud rate) | 115200 (tối đa) |
| Chế độ hoạt động | AP, STA và (AP + STA) |
| Bộ nhớ Flash | 1MB |
| Antenna PCB trace | có |
| Điện áp ra I/O | 3.6V (tối đa) |

### Module cảm biến bụi Shinyei PPD42NS

Module cảm biến bụi PPD42NS dùng để đo chất lượng không khí trong môi trường bằng cách đo nồng độ bụi. Mức độ hạt vật chất (mức độ PM) trong không khí được đo bằng cách đếm thời gian xung thấp (thời gian LPO) trong một đơn vị thời gian nhất định. Thời gian LPO tỉ lệ thuận với nồng độ PM.

Module cảm biến bụi PPD42NS rất dễ dàng sử dụng, có thể cung cấp dữ liệu chính xác cho các hệ thống lọc không khí, nó phát hiện được các hạt bụi cỡ 1μm. Được ứng dụng nhiều trong thực thế như: Máy lọc không khí, máy đo chất lượng không khí. Trên module còn có 1 lỗ via để giúp có thể cố định lại cảm biến.

A picture containing clock

Description automatically generated

Hình 3.15 Cảm biến bụi Shinyei PPD42NS

Bảng 3.6 Thông số kỹ thuật của cảm biến bụi PPD42NS

|  |  |
| --- | --- |
| Kích thước hạt | >= 1μm |
| Khoảng nồng độ | 0~28,000 pcs/lít (0~8,000pcs/0.01 CF=283ml) |
| Điện áp cung cấp | DC5V +/- 10% (CN1:Pin1=GND, Pin3=+5V)  Điện áp gợn sóng 30mV |
| Khoảng nhiệt độ | 0~45°C |
| Khoảng độ ẩm | 95%RH hoặc ít hơn (không ngưng tụ sương) |
| Mức tiêu thụ năng lượng | 90mA |
| Nhiệt độ bảo quản | -30~60°C |
| Thời gian ổn định | 3 phút sau khi bật nguồn |
| Kích thước | 59(W) × 45(H) × 22(D) [mm] |
| Trọng lượng | ~ 24g |
| Tính hiệu ngõ ra | Negative Logic, Digital output  Hi: over 4.0, Lo: under 0.7V |
| Tuổi thọ | Hoạt động ít nhất 7 năm |

### Module cảm biến nhiệt độ và độ ẩm SHT31

Cảm biến nhiệt độ và độ ẩm SHT31 được dùng cho các ứng dụng đo độ ẩm và nhiệt độ môi trường, cảm biến thuộc dòng trung cấp của hãng Sensirion. Cảm biến này được bán nhiều trên thị trường nên không quá khó khăn để có thể mua được cảm biến này.

A close up of a sign

Description automatically generated

Hình 3.16 Cảm biến SHT31

Bảng 3.7 Thông số kỹ thuật của cảm biến SHT31

|  |  |
| --- | --- |
| Dải đo độ ẩm | 0 - 100% RH |
| Độ chính xác độ ẩm | ± 2% RH |
| Dải đo nhiệt độ | -40°C ~ 125°C |
| Độ chính xác nhiệt độ | ± 0.3°C (tại -40°C ~ 90°C) |
| Điện áp cung cấp | 2.4V ~ 5.5V |
| Tính hiệu ngõ ra | I2C, Digital Output |
| Kích thước module | 11x13mm |

### Module thời gian thực RTC DS1307

Mạch thời gian thực RTC DS1307 được sử dụng để cung cấp thông tin thời gian: ngày, tháng, năm, giờ, phút, giây,...cho Vi điều khiển qua giao tiếp I2C, mạch tích hợp sẵn pin backup để duy trì thời gian trong trường hợp không cấp nguồn, ngoài ra mạch còn được tích hợp thêm IC EEPROM AT24C32 để lưu trữ thông tin khi cần, thích hợp cho các ứng dụng điều khiển hoặc đồng bộ dữ liệu thời gian thực RTC.



Hình 3.17 Module RTC DS1307

Bảng 3.8 Thông số kỹ thuật của module RTC DS1307

|  |  |
| --- | --- |
| Điện áp cung cấp | 5 V DC |
| Phạm vi nhiệt độ | 0-70oC |
| Chuẩn giao tiếp | I2C |
| Kích thước | 27 x 28 x 8.4mm. |

## Thiết kế và hiện thực phần mềm trên ESP32 WROVER KIT

### Lưu đồ giải thuật

Chương trình của thiết bị được thực hiện bằng cách thực hiện các bước như sau:

Bước 1: Khởi tạo, cấu hình phần cứng.

Bước 2: Dùng RTOS để chia chương trình chính thành 6 task chạy song song.

Task 1: Đo mật độ hạt bụi trong 1 cm3.

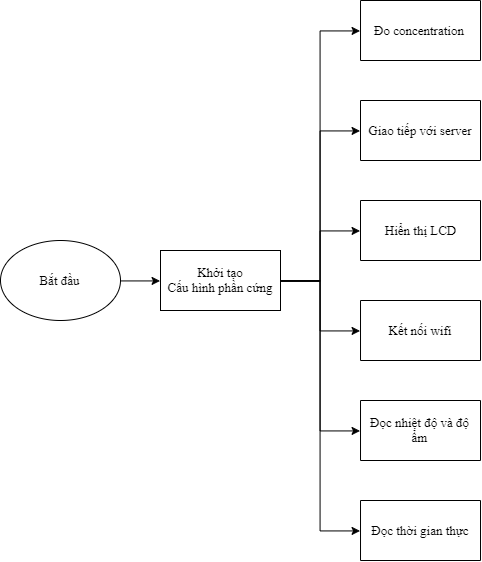
Task 2: Giao tiếp với server.

Task 3: Hiển thị các giá trị đo được lên màn hình LCD.

Task 4: Kết nối wifi.

Task 5: Đọc nhiệt độ và độ ẩm từ môi trường.

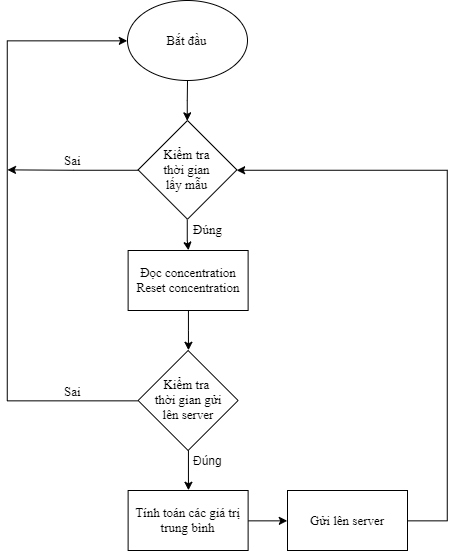
Task 6: Đọc thời gian thực.



Hình 3.18 Lưu đồ chung thể hiện các task

Trong đó task 2 và task 4 là 2 task trọng tâm cần được làm rõ trong hệ thống.

* Task 2 “giao tiếp với server”.



Hình 3.19 Lưu đồ giải thuật của bước giao tiếp với server

Cách bước thực hiện như sau:

Bước 1: Kiểm tra thời gian lấy mẫu của cảm biến bụi PPD42NS. Nếu thời gian lấy mẫu đúng thì thực hiện bước 2. Nếu thời gian lấy mẫu sai thì lặp lại bước 1.

Bước 2: Đọc giá trị concentration từ cảm biến PPD42NS, sau đó đặt lại giá trị đó bằng 0.

Bước 3: Kiểm tra thời gian gửi lên server. Nếu thời gian gửi lên server đúng thì thực hiện bước 4. Nếu thời gian gửi lên server sai thì quay lại bước 1.

Bước 4: Tính trung trình các giá trị đo nhiệt độ, độ ẩm và concentration đo được.

Bước 5. Khi tính toán xong, thiết bị sẽ thực hiện gửi các giá trị đo được lên server. Sau đó quay lại bước 1.

Dữ liệu gửi đi dạng JSON và có định dạng:

{

temp: <float>,

humi: <float>,

aqi: <int>,

pollutant: <float>,

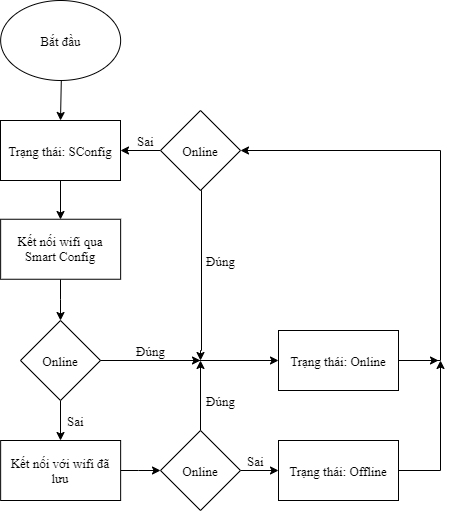
location: <String>,

device\_id: <String>

}

Trong đó:

* temp là nhiệt độ.
* humi là độ ẩm.
* aqi chỉ số ô nhiễm không khí.
* pollutant là mật độ hạt bụi.
* location là vị trí đang lấy dữ liệu.
* device\_id là mã xác nhận của thiết bị.
* Task 4 “kết nối wifi”.



Hình 3‑4 Lưu đồ giải thuật kết nối wifi

Cách bước thực hiện như sau:

Bước 1: Đưa thiết bị vào trạng thái SConfig.

Bước 2: Kết nối thiết bị thông qua Smart Config. Nếu kết nối thành công thì chuyển trạng thái sang “online”, nếu không thành công thì chuyển sang trạng thái “offline”.

Bước 3: Kiểm tra trạng thái kết nối. Nếu trạng thái là “online” thì thực hiện bước 6. Nếu trạng thái là “offline” thì thực hiện bước 4.

Bước 4: Kết nối wifi dựa vào ssid và password mà thiết bị đã lưu trong bộ nhớ điện tĩnh (non-volatile storage) khi kết nối thành công ở lần trước. Nếu kết nối thành công thì chuyển trạng thái sang “online”, nếu không thành công thì chuyển sang trạng thái “offline”.

Bước 5: Kiểm tra trạng thái kết nối. Nếu kết nối thành công thì chuyển trạng thái sang “online”, nếu không thành công thì chuyển sang trạng thái “offline”. Sau đó thực hiện bước 6.

Bước 6: Delay task 5 phút sau đó kiểm tra trạng thái kết nối. Nếu “online” thì lặp lại bước 6. Nếu “offline” thì quay lại bước 1.

### Chức năng và mô tả

Bảng 3.9 Danh sách chức năng của phần mềm trên ESP32

|  |  |
| --- | --- |
| Chức năng | Mô tả |
| Xem độ ẩm | Độ ẩm hiện tại, đơn vị là phần trăm (%) |
| Xem nhiệt độ | Nhiệt độ hiện tại, đơn vị là độ C (oC) |
| Xem chỉ số AQI | Chỉ số AQI hiện tại, kiểu số nguyên |
| Xem mật độ hạt bụi | Mật độ hạt bụi trong 1m3,kiểu số thực, đơn vị là ug/m3 |
| Xem thời gian thực | Xem thời gian theo múi giờ UTC+7 |
| Xem trạng thái kết nối mạng | Có 3 trạng thái: Online (màu xanh lá), Offline (màu xám), SConfig (màu đỏ) |

## Thiết kế và hiện thực Server

### Thông tin của server

Server sử dịch vụ điện toán đám mây của Amazon có tên Amazon Web Services. Trong luận văn này chúng em sử dụng gói “AWS Free Tier”, đây là gói có giới hạn được Amazon cho phép sử dụng miễn phí 1 năm gồm rất nhiều các dịch vụ khác nhau và được phép nâng cấp lên các gói cao cấp hơn bất kì lúc nào.

Bảng 3.10 Thông tin về server

|  |  |
| --- | --- |
| Thông tin | Mô tả |
| Vị trí đặt server | Tiểu bang Ohio, Đông Bắc Hoa Kỳ |
| Hệ điều hành | Ubuntu 18.04 LTS |
| Địa chỉ IPv4 | 13.59.35.198 |
| Loại ổ cứng | SSD |
| Dung lượng ổ cứng | 8GB |
| RAM | 1GB |
| Tốc độ đọc/ghi dữ liệu | 100 IOPS |

### Chức năng của server

Bảng 3.11 Bảng danh sách các chức năng của server

|  |  |
| --- | --- |
| Chức năng | Mô tả |
| Nhận dữ liệu từ thiết bị | Nhận các thông tin liên quan đến không khí: nhiệt độ, độ ẩm, chỉ số AQI, mật độ hạt bụi và mã xác nhận của thiết bị. Nếu mã thiết bị hợp lệ thì server sẽ lưu lại các thông tin nhận được vào cơ sở dữ liệu. |
| Kiểm tra đăng nhập | Kiểm tra thông tin đăng nhập. Nếu hợp lệ sẽ cho người dùng đến bảng điều khiển. Nếu không hợp lệ sẽ yêu cầu nhập lại |
| Xem dữ liệu mới nhất | Sau khi đăng nhập, người dùng có thể xem dữ liệu mới nhất |
| Xem dữ liệu trong một khoảng thời gian dưới dạng biểu đồ | Sau khi đăng nhập, người dùng có thể xem dữ liệu trong một khoảng thời gian dưới dạng biểu đồ |
| Chọn vị trí để xem dữ liệu | Sau khi đăng nhập, người dùng có thể chọn vị trí có thiết bị đo để xem dữ liệu |
| Quản lí các thiết bị (thêm, sửa, xóa) | Sau khi đăng nhập, người dùng có thể quản lí được các thiết bị. Có thể thêm, sửa, xóa thiết bị |
| Quản lí tài khoản (thêm, sửa, xóa) | Sau khi đăng nhập, người dùng có thể quản lí được các tài khoản trong hệ thống. Có thể thêm, sửa, xóa tài khoản |

### Thiết kế server với NodeJS

Sử dụng framework expressjs của NodeJs để xây dựng server. Express hỗ trợ giao thức HTTP và midleware tạo ra các api vô cùng mạnh mẽ và dễ sử dụng.

Ta sử dụng express để tạo ra một HTTP server chạy ở port 8000. Server này cho phép trả về trang html và cung cấp các api.

Để xây dựng được một server hoàn chỉnh ta cần một số module khác của NodeJS.

Bảng 3.12 Danh sách module của server

|  |  |
| --- | --- |
| Tên module | Mô tả |
| body-parser | Hỗ trợ lấy data form từ request body |
| cors | Cho phép các tài nguyên của một trang web có thể được truy vấn từ domain khác với domain của trang web đó |
| crypto | Mã hóa chuỗi |
| express-session | Quản lý session |
| influx | Hỗ trợ kết nối và truy vấn đến influxdb |
| mongodb | Hỗ trợ kết nối và truy vấn đến mongodb |
| passport | Xác thực người dùng |
| winston | Hỗ trợ ghi log hệ thống |

Bảng 3.13 Bảng danh sách các router của server và chức năng

|  |  |
| --- | --- |
| Tên route | Chức năng |
| /esp32 | Giao tiếp nhận dữ liệu từ thiết bị |
| /dashboard | Trả về trang html |
| /admin | API cung cấp chức năng quản lý (thêm, xóa, sửa) user |
| /device | API cung cấp chức năng quản lý (thêm, xóa, sửa) thiết bị |
| /api | Public API cung cấp các thông tin liên quan đến chất lượng không khí |

### Xây dựng các API

* esp32: Nhận dữ liệu từ thiết bị gửi lên. Dữ liệu gồm các số liệu liên quan đến chất lượng không khí, các số liệu này được server parse về kiểu float. Dựa vào chỉ số aqi ta sẽ gán mô tả mức độ ô nhiễm. Trước khi ghi dữ liệu vào database ta cần kiểm tra mã thiết bị được gửi lên theo request, nếu mã thiết bị tồn tại thì mới cho phép ghi, để hạn chế người phá hoại ghi dữ liệu rác vào hệ thống.
* dashboard: Trả về trang html được viết bởi Angular cho người dùng, trang này có chức năng theo dõi thông tin chất lượng không khí, quản lý user, quản lý thiết bị.
* admin: API quản lý user. Khi nhận được request, server thực hiện truy vấn đến mongodb và trả về kết quả tương ứng với các yêu cầu như thêm, xóa, sửa user.
* device: API quản lý thiết bị. Khi nhận được request, server thực hiện truy vấn đến mongodb và trả về kết quả ứng với các yêu cầu thêm, xóa, sửa thiết bị.
* api: API xem thông tin liên quan đến chất lượng không khí. Khi nhận được request, server thực hiện truy vấn đến influxdb và trả về các kết quả tương ứng

Bảng 3.14 Danh sách các API của server

|  |  |  |
| --- | --- | --- |
| Phương thức | URI | Chức năng |
| POST | /esp32 | Nhận dữ liệu từ thiết bị |
| GET | /login | Truy cập vào trang đăng nhập |
| POST | /login | Xác thực người dùng |
| GET | /dashboard/home | Truy cập vào trang chủ xem dữ liệu |
| GET | /dashboard/user | Truy cập vào trang quản lý người dùng |
| GET | /dashboard/device | Truy cập vào trang quản lý các thiết bị |
| GET | /dashboard/logout | Đăng xuất khỏi trang web |
| POST | /admin/userList | Lấy danh sách các user |
| POST | /admin/createUser | Tạo user mới |
| POST | /admin/updateUser | Cập nhật thông tin user đã có |
| POST | /admin/changePassword | Đổi mật khẩu user |
| POST | /admin/removeUser | Xóa uer |
| GET | /api/statistics | Lấy thống kê của hệ thống |
| GET | /api/forecast | Lấy thông tin chất lượng không khí mới nhất |
| GET | /api/locations | Lấy danh sách các địa điểm đặt thiết bị đo |
| GET | /api/airdata | Lấy thông tin chất lượng không khí theo khoảng thời gian |
| POST | /device/deviceList | Lấy thông tin các thiết bị đo |
| POST | /device/createDevice | Thêm thiết bị |
| POST | /device/updateDevice | Cập nhật thiết bị |
| POST | /device/removeDevice | Xóa thiết bị |

## **Thiết kế và hiện thực Web**

### Angular 8

**Angular 8** là phiên bản quan trọng được Google cập nhật vào năm 2019. Angular 8 tập trung vào các công cụ giúp Angular dễ dàng sử dụng hơn cho các nhà phát triển xây dụng các loại ứng dụng với sự tối ưu hóa về hiệu năng. Angular 8 hỗ trợ phiên bản TypeScript 3.4 giúp lập trình dễ dàng và nhanh hơn rất nhiều.

Để cài đặt Angular 8 thông qua npm, sử dụng câu lệnh sau:

npm install -g @angular/cli

### Chức năng của ứng dụng

Trước khi bắt đầu xây dựng một ứng dụng web ta cần xác định và liệt kê các chức năng của ứng dụng. Trong đề tài này ứng dụng web cần phải thể hiện được các số liệu mà thiết bị đo được, thống kê số lần thiết bị gửi dữ liệu lên server, quản lý (thêm, sửa, xóa) một số thông tin như các tài khoản đăng nhập vào ứng dụng, id của các thiết bị đo.

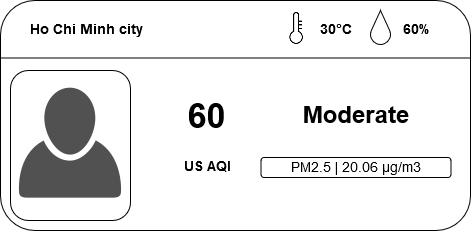
Bảng 3.15 Các chức năng của ứng dụng web

|  |  |
| --- | --- |
| Chức năng | Mô tả |
| Xem dữ liệu mới nhất | Xem các thông tin liên quan đến chất lượng không khí như: chỉ số aqi, nhiệt độ, độ ẩm. Các thông tin này được truy vấn từ hàng cuối cùng trong database. |
| Xem dữ liệu của tháng hiện tại, tuần hiện tại | Thể hiện dữ liệu các thông tin liên quan đến chất lượng không khí theo tuần hoặc tháng dưới dạng biểu đồ. Mỗi thông tin là một biểu đồ riêng. Dữ liệu được truy vấn theo thời gian từ database. |
| Quản lý thông tin các thiết bị | Quản lý thông tin các thiết bị đo dưới dạng bảng. Có thể thêm, sửa, xóa thông tin thiết bị như id, vị trí. Id dùng để xác thực thiết bị đo trước khi thêm dữ liệu vào database. |
| Quản lý thông tin người dùng | Quản lý thông tin đăng nhập và thông tin cá nhân của người dùng như tên đăng nhập, mật khẩu, email, họ tên. Các thông tin này được trình bày dưới dạng bảng. Có thể thêm, sửa, xóa các thông tin này. |
| Chọn vị trí để xem dữ liệu | Do có thể có nhiều thiết bị đo ở những vị trí khác nhau, nên người dùng có thể chọn vị trí muốn xem dữ liệu. Dữ liệu trả về sẽ là dữ liệu được truy vấn theo trường “location” trong database. |
| Chọn khoảng thời gian muốn xem dữ liệu | Khoảng thời gian này có thể là trong ngày, theo tuần, theo tháng hoặc một số lượng tùy chỉnh. Dữ liệu trả về được truy vấn theo trường “time” trong database. |

### Thiết kế các màn hình ứng dụng

* Thiết kế màn hình xem thông tin chất lượng không khí

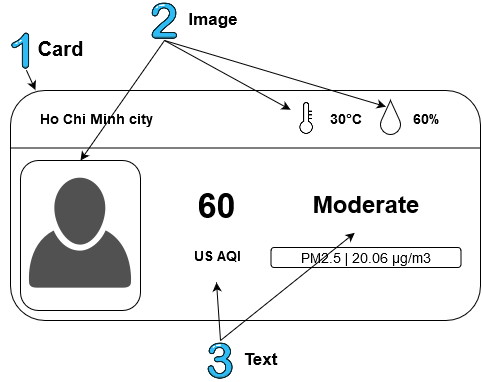
Các thông tin cần hiển thị: vị trí của thiết bị đo, nhiệt độ (°C), độ ẩm (%), chỉ số aqi, chỉ số pollutant (µg/m3), mức độ ô nhiễm tương ứng với chỉ số aqi.



Hình 3.20 Mockup các thông tin cần hiển thị

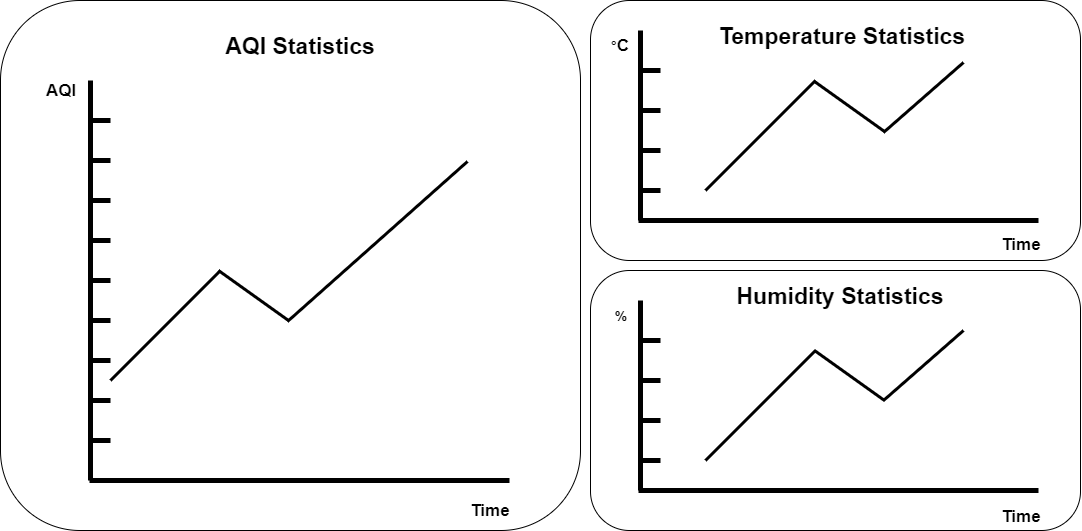
Các component cần có:

* + CardView: làm container để chứa các component khác
  + Text: Để hiển thị thông tin
  + ImageView: Để mô tả thông tin cho người dùng dễ nhận biết và phân biệt thông tin đó là nhiệt độ, độ ẩm hay chỉ số aqi



Hình 3.21 Các component cần để thể hiện các thông tin

Để hiển thị thông tin theo tháng, theo tuần hoặc theo một khoảng thời gian ta cần biểu đồ để thể hiện. Vì lượng dữ liệu có thể rất lớn nên dùng biểu đồ đường là thích hợp. Vậy trên màn hình cần phải thể hiện được 3 biểu đồ tương ứng với chỉ số chất lượng không khí, nhiệt độ và độ ẩm.



Hình 3.22 Mockup các biểu đồ cần hiển thị

Hiện nay có các thư viện hỗ trợ vẽ biểu đồ rất chuyên nghiệp tương thích với Angular như: d3js, swimlane ngx-charts, chartjs, canvasjs... Qua một thời gian tìm hiểu các thư viện này, nhóm nhận thấy các thư viện này đòi hỏi mất nhiều thời gian để nghiên cứu và điều chỉnh để có thể vẽ được biểu đồ như mong muốn. D3JS khá phức tạp, ít hướng dẫn chi tiết để sử dụng với Angular. Ngx-charts dễ dàng điều chỉnh các thuộc tính để vẽ được biểu đồ như mong muốn, hiệu ứng đẹp mắt, tuy nhiên dữ liệu vào theo một khuôn mẫu của thư viện. CanvasJS có bài viết chi tiết về cách điểu chỉnh các thuộc tính trên trang chủ, tương thích với Angular, có rất nhiều ví dụ được viết bằng TypeScript. Cuối cùng nhóm quyết định chọn CanvasJS để thể hiện thông tin, CanvasJS rất dễ làm quen, các ví dụ trên trang chủ của Canvas rất chi tiết, hỗ trợ chỉnh sửa code để xem các thay đổi ngay lập tức.

Chức năng lọc dữ liệu theo vị trí và khoảng thời gian cần tới 2 khung nhập liệu. Để tiện cho người dùng, ta sẽ sử dụng component autocomplete cho khung nhập vị trí và component select cho khung chọn khoảng thời gian. Autocomplete hỗ trợ gợi ý khi người dùng bắt đầu nhập một ký tự bất kì. Select hiện ra một danh sách các khoảng thời gian để chọn nhanh, người dùng không cần phải nhập gì cả. Khoảng thời gian có thể là 1 ngày, 1 tuần, 1 tháng hoặc 100 số liệu gần đây.

Bên cạnh thông tin về chỉ số chất lượng không khí mà thiết bị đo được, màn hình này cũng cần thể hiện được số liệu thống kê về số lần các api được gọi, số lần trang web được truy cập và một số thông tin khác. Các số liệu này được thể hiện trong các card như bên dưới:



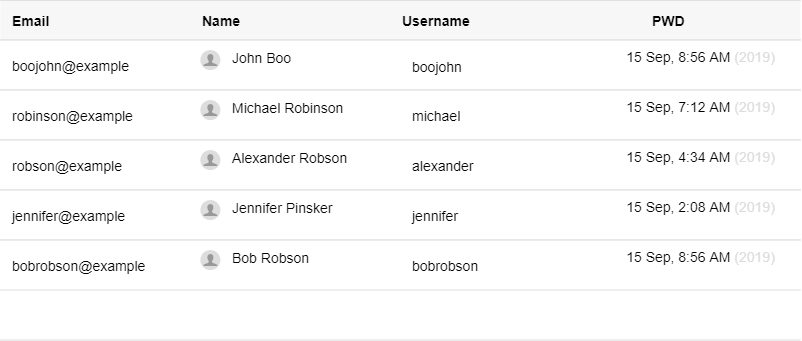
Hình 3.23 Mockup thể hiện số liệu thống kê

* Thiết kế màn hình quản lý thông tin người dùng

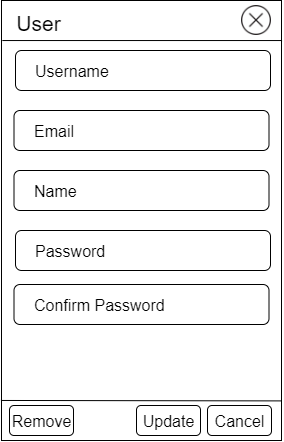
Các thông tin cần hiển thị: thông tin của từng user gồm email, họ tên, tên đăng nhập.

Các component cần có:

* Table: hiển thị thông tin của từng user, số lượng user có thể lớn nên sử dụng là table thích hợp. Table có kích thước vừa với màn hình, khi số lượng user lớn, màn hình không hiển thị hết thì thanh cuộn sẽ xuất hiện.
* Sidenav: khi chọn vào một hàng trong table, sidenav được mở ra. Sidenav dùng để hiển thị chi tiết về user đó cùng với các nút chức năng để có thể thêm, xóa, sửa thông tin của user.
* Button: tạo sự kiện thực hiện các chức năng tương ứng.



Hình 3.24 Table hiển thị thông tin từng user



Hình 3.25 Sidenav cùng với các công cụ để thay đổi thông tin user

* Thiết kế màn hình quản lý thông tin thiết bị

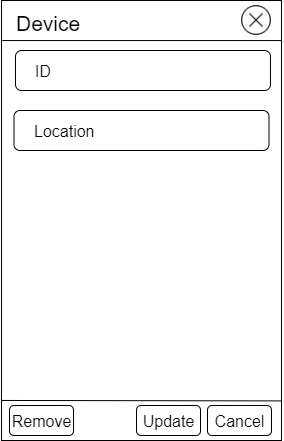
Các thông tin cần hiển thị: thông tin của từng thiết bị gồm id, vị trí.

Các component cần có:

* Table: hiển thị thông tin của từng thiết bị.
* Sidenav: khi chọn vào một hàng trong table, sidenav được mở ra. Sidenav dùng để hiển thị chi tiết về thiết bị đó cùng với các nút chức năng để có thể thêm, xóa, sửa thông tin của thiết bị.
* Button: tạo sự kiện thực hiện các chức năng tương ứng.



Hình 3.26 Table hiển thị thông tin từng thiết bị



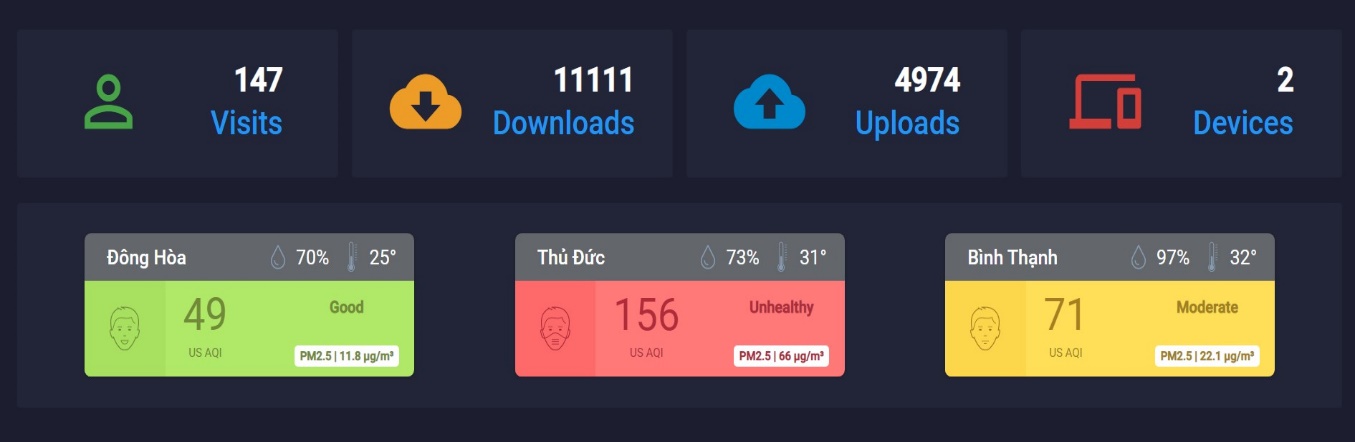
Hình 3.27 Sidenav cùng với các công cụ để thay đổi thông tin thiết bị

### Xây dựng các màn hình

Sử dụng FlexLayout để hỗ trợ bố trí các component trên giao diện. FlexLayout trong Angular thực chất là Flexbox – một kiểu dàn trang mà nó sẽ tự cân đối kích thước của các phần tử bên trong để hiển thị trong mọi thiết bị. Nói cách khác với FlexLayout ta không cần phải thiết lập kích thước của các phần tử, chỉ cần thiết lập nó hiển thị theo chiều ngang hay chiều dọc thì các phần tử bên trong sẽ hiển thị theo ý muốn.

* Xây dựng màn hình xem thông tin chất lượng không khí

Thiết lập FlexLayout là “row” với kiểu bố trí “space-between”. Bố trí các component theo mockup ta được kết quả:



Hình 3.28 Giao diện hiển thị thông tin mới nhất

Màu của mỗi card thông tin phụ thuộc vào chỉ số aqi và mức độ ô nhiễm:

Bảng 3.16 Màu sắc và ảnh mô tả của card thể hiện thông tin theo aqi

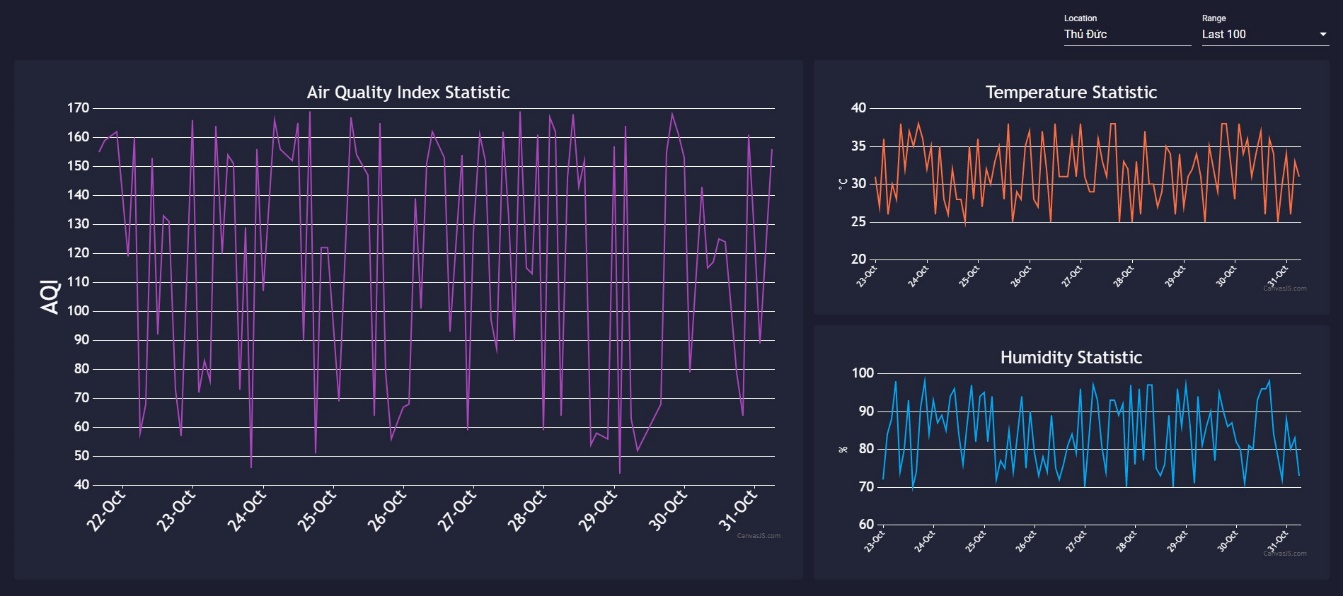
|  |  |  |
| --- | --- | --- |
| Khoảng giá trị AQI | Màu nền | Ảnh mô tả |
| 0-50 | Xanh lá |  |
| 51-100 | Vàng |  |
| 101-150 | Cam |  |
| 151-200 | Đỏ |  |
| 201-300 | Tím |  |

Mỗi card hiển thị các thông tin về chất lượng không khí mới nhất của một vị trí.

Để có thể thiết lập được màu nền của card và ảnh mô tả theo từng trạng thái ta sử dụng các thuộc tính \*ngIf và \*ngSwitchCase mà Angular hỗ trợ.

Ngoài ra Angular còn hỗ trợ \*ngFor để tạo ra các view giống nhau trên giao diện, giúp người phát triển đỡ mất thời gian viết lại các view tương tự nhau. Ở đây các card được sinh ra từ vòng lặp for của \*ngFor.

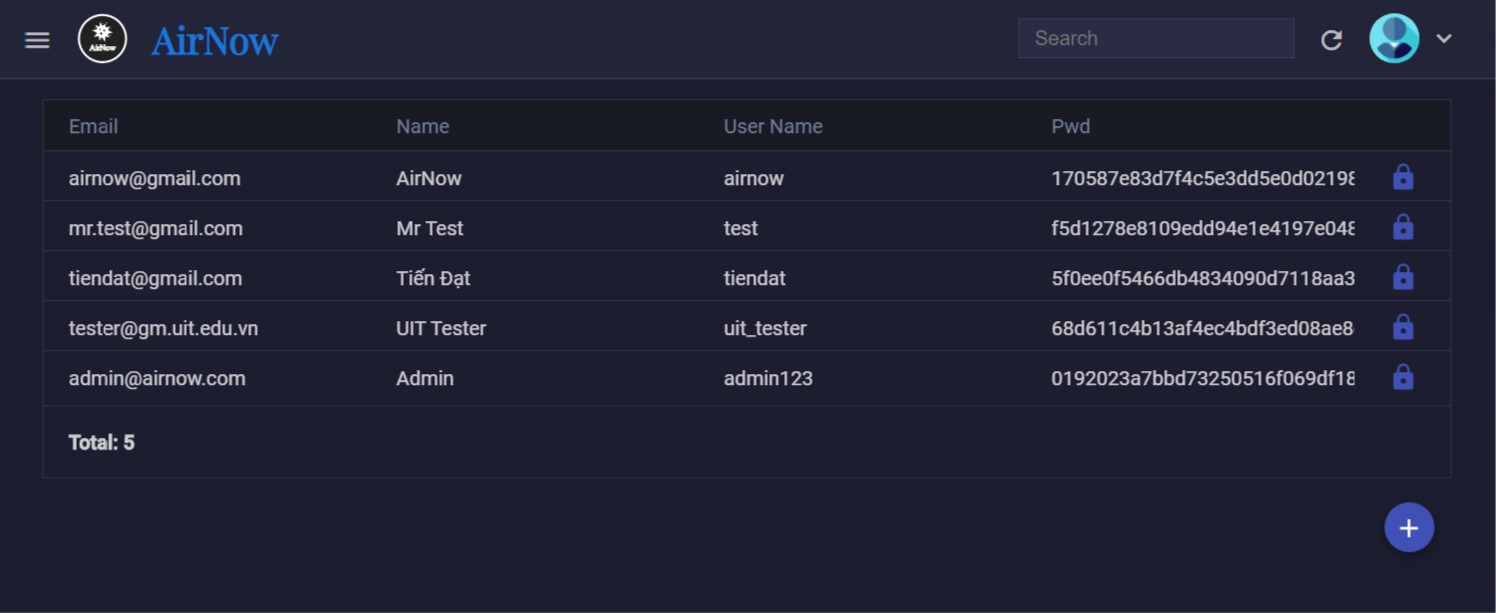
Tương tự với view các biểu đồ hiển thị thông tin trong một khoảng thời gian



Hình 3.29 Giao diện hiển thị thông tin theo khoảng thời gian

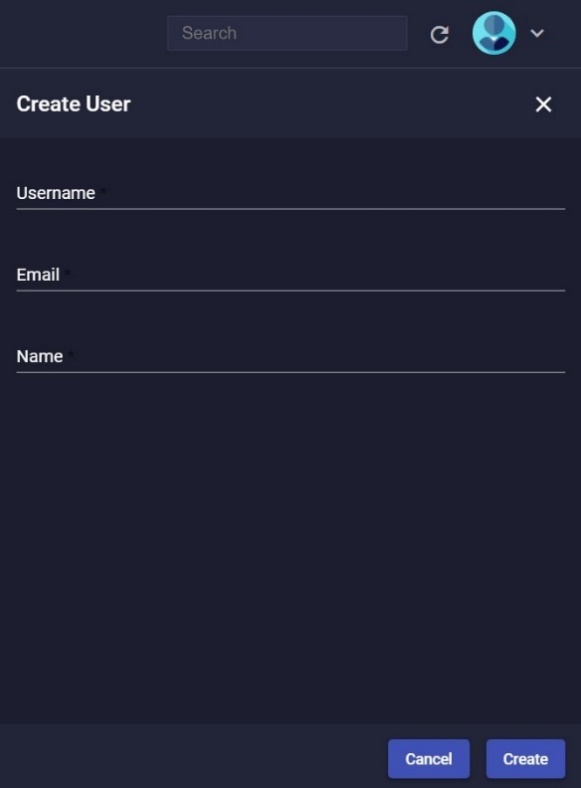
* Xây dựng màn hình quản lý user

Trong giao diện quản lý user ta cần một bảng hiển thị danh sách các user. Trong cộng đồng Angular có một thư viện rất nổi tiếng dùng để tạo bảng trên trang html là swimlane/ngx-datatable. Thư viện này hỗ trợ tạo bảng rất chuyên nghiệp, có thể tạo bảng động, căn chỉnh tự động và giúp người phát triển tiết kiệm thời gian viết mã nguồn.



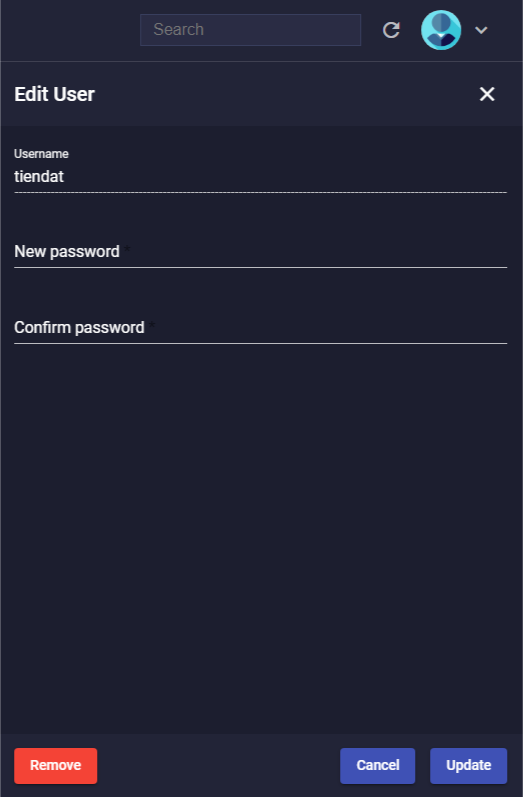
Hình 3.30 Giao diện quản lý user

Trên giao diện này có một button ở góc phải bên dưới. Khi bấm vào, sidenav gồm một form trống hiện ra để người quản lý đăng kí thông tin cho một user mới.



Hình 3.31 Sidenav tạo user mới

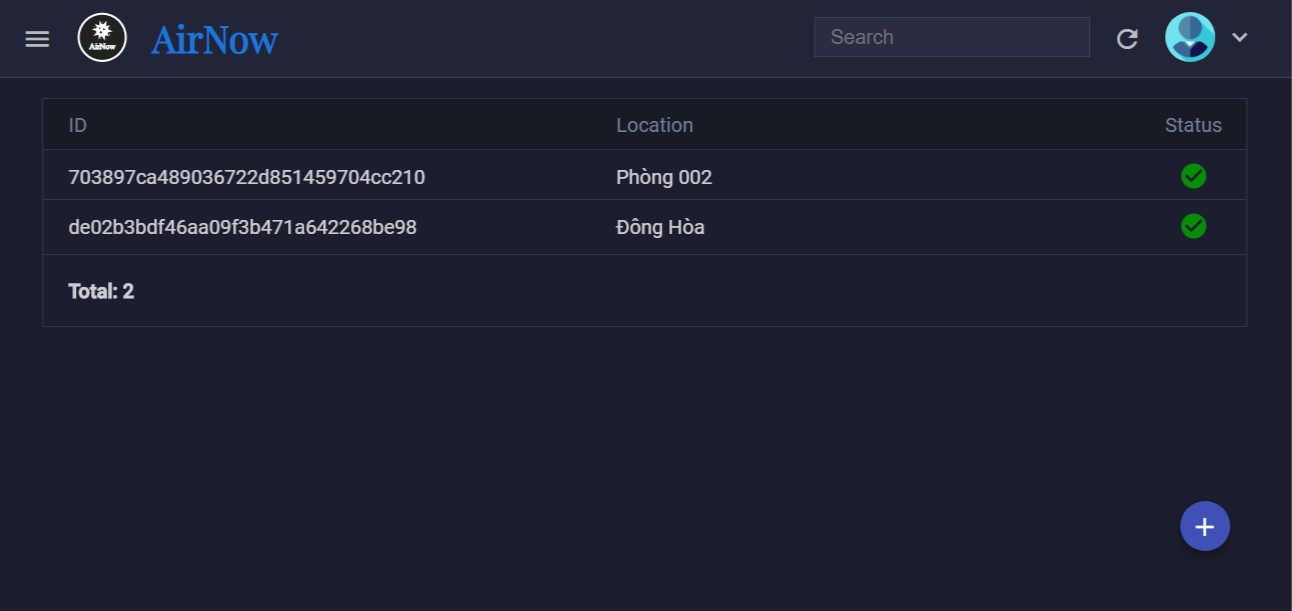
Mỗi hàng trong bảng có một button. Khi bấm chọn button này, một sidenav hiện ra giúp người quản lý thay đổi mật khẩu của user được chọn.



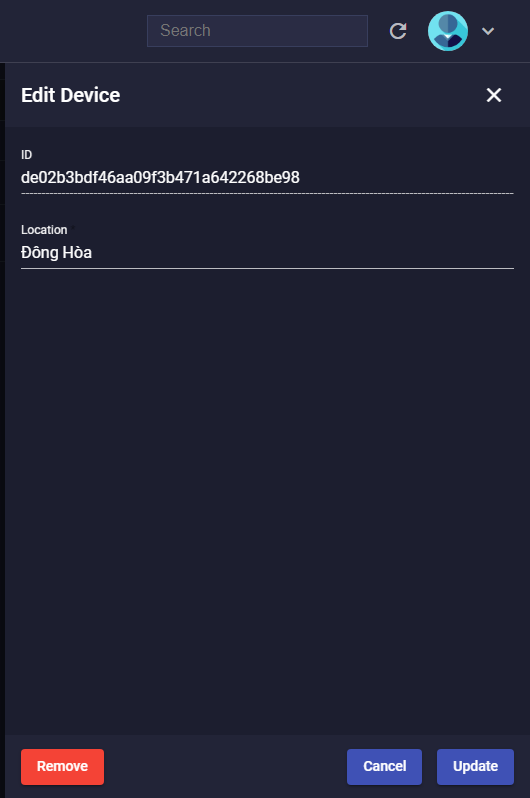
Hình 3.32 Sidenav đổi mật khẩu user

* Xây dựng màn hình quản lý thiết bị

Giao diện quản lý thiết bị tương tự như giao diện quản lý user.



Hình 3.33 Giao diện quản lý thiết bị



Hình 3.34 Sidenav thay đổi thông tin thiết bị

### Xây dựng chức năng cho các màn hình

Các màn hình khi được khởi tạo cần phải gọi đến api để lấy dữ liệu. Ta sẽ dùng NgRx để thực hiện gọi api lấy dữ liệu và lưu vào Store.

* Bước đầu tiên ta tạo các Action:

export const GET\_AIR = '[Dashboard] Get AirData';

export class GetAir implements Action {

readonly type = GET\_AIR;

constructor(public payload: any) { }

}

* Kế tiếp tạo Effect để gọi đến api:

@Effect()

GetAir = this.\_actions$.pipe(

ofType(DashboardActions.GET\_AIR),

switchMap((params) =>

this.http.get(this.SERVER\_URL + 'airdata', { params }).pipe(

map((data: any) => {

return new DashboardActions.FetchAir(data);

})));

Sau khi nhận được dữ liệu từ server ta sẽ gọi đến một Action tên FetchAir để truyền dữ liệu qua Reducer.

* Reducer nhận dữ liệu và lưu vào Store:

export function dashboardReducer(state = initialState, action: DashboardActions) {

switch (action.type) {

case DashboardActions.FETCH\_AIR: {

return state

.set('aqi', aqiList)

.set('temperature', temperatureList)

.set('humidity', humidityList);

}

}

}

Dữ liệu được lưu vào Store thông qua lệnh set.

* Khi dữ liệu được lưu vào Store, Selector bắt được việc thay đổi dữ liệu:

export function getAqi(): Selector<IAppState, DashboardState> {

return state$ => state$.pipe(

map(state => state.dashboard.get('aqi')),

distinctUntilChanged()

);

}

* Dữ liệu được cập nhật lên Component từ Selector và hiển thị ra cho người dùng.

Mỗi màn hình có một số State cụ thể và gọi đến api riêng để lấy dữ liệu.

Màn hình xem thông tin chất lượng không khí có các State sau:

* Statistics: thống kê lượng truy cập, số lần thiết bị gửi dữ liệu lên, ...
* Forecast: dữ liệu về không khí hiện tại ở một số vị trí.
* aqiList: dữ liệu aqi của một vị trí nào đó trong một khoảng thời gian.
* temperatureList: dữ liệu nhiệt độ của một vị trí trong một khoảng thời gian.
* humidityList: dữ liệu độ ẩm của một vị trí trong một khoảng thời gian.

## Thiết kế và hiện thực ứng dụng di động trên Android

### Các chức năng của ứng dụng

Bảng 3.17 Bảng chức năng của ứng dụng Android

|  |  |
| --- | --- |
| Chức năng | Mô tả |
| Xem dữ liệu mới nhất | Hiển thị các thông tin về chất lượng không khí như chỉ số aqi, nhiệt độ, độ ẩm. |
| Xem dữ liệu theo tháng | Hiển thị thông tin chất lượng không khí dưới theo tháng dạng biểu đồ. Mỗi tính chất là một biểu đồ. |
| Xem dữ liệu theo vị trí | Hiển thị một danh sách các vị trí để chọn xem thông tin tại vị trí đó |
| Cài đặt wifi cho thiết bị đo | Điều chỉnh tên wifi và mật khẩu kết nối cho thiết bị đo thông qua smartphone. |
| Cài đặt đơn vị hiển thị nhiệt độ | Độ C hoặc độ F |

### Thiết kế giao diện

* Thiết kế thanh chuyển hướng

Thanh chuyển hướng giúp chuyển đổi giữa các màn hình trong ứng dụng.

Android hỗ trợ cho chúng ta sẵn các ví dụ về thanh chuyển hướng. Trong đó có 2 dạng phổ biến là kiểu sidenav navigation và bottom navigation.

Ứng dụng này không cần quá nhiều màn hình nên sử dụng bottom navigation là hợp lý, giúp người dùng có thể chuyển đổi nhanh chóng giữa các màn hình.

* Thiết kế màn hình xem dữ liệu mới nhất

Sử dụng các component của material design kết hợp với LinearLayout của android ta có thể linh động bố trí các thành phần trên giao diện một cách nhanh chóng và hiệu quả.

Màn hình này cần một khung chọn vị trí và một danh sách hiển thị thông tin chi tiết chất lượng không khí.

Khung chọn vị trí sử dụng material input cùng với autocomplete. Khi focus vào khung một danh sách các vị trí sẽ hiện ra. Người dùng chọn một vị trí bất kì, dữ liệu của vị trí đó được cập nhật lên giao diện. Mặc định khi không có vị trí nào được chọn, ứng dụng sẽ load dữ liệu từ 3 vị trí mới nhất trong database.

Danh sách thông tin chất lượng không khí theo vị trí ta sử dụng ListView. Thiết lập chiều dài của ListView vừa với màn hình, nếu các thành phần trong List quá nhiều thì thanh cuộn trang sẽ xuất hiện. Các thành phần trong List thể hiện nhiều nội dung và phức tạp nên ta phải tự xây dựng một Adapter cho ListView này.

* Thiết kế màn hình xem dữ liệu theo tháng

Màn hình này cũng gồm một khung chọn vị trí và một danh sách.

Khung chọn vị trí có chức năng tương tự như trong màn hình xem dữ liệu mới nhất.

Danh sách ở màn hình này gồm 3 biểu đồ thể hiện aqi, nhiệt độ và độ ẩm trong tháng. Vì vậy cũng cần một Adapter riêng cho danh sách 3 biểu đồ này.

Để vẽ được biểu đồ trên nền tảng Android ta sẽ sử dụng thư viện MPChart. Thư viện này khá hữu ích, hỗ trợ vẽ rất nhiều dạng biểu đồ.

MPChart cũng như CanvasJS trong Angular, có nhiều ví dụ cụ thể. Tham khảo các ví dụ mẫu ta cũng có thể tự sáng tạo cho mình một biểu đồ theo sở thích và nhu cầu.

A screenshot of a cell phone

Description automatically generatedA close up of a map

Description automatically generated

Hình 3.35 Mockup màn hình xem dữ liệu mới nhất và dữ liệu theo tháng

* Thiết kế màn hình cài đặt

Màn hình này hiện tại cần 2 chức năng: thiết lập wifi cho thiết bị đo, thay đổi đơn vị nhiệt độ hiển thị.

Trong giao diện thiết lập wifi cho thiết bị đo cần phải hiển thị được tên wifi (SSID), địa chỉ mạng (BSSID), mật khẩu wifi và một nút bấm để xác nhận thao tác.

Do giao diện thiết lập wifi cần nhiều thành phần nhỏ nên ta sẽ tách giao diện này ra một màn hình khác.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Hình 3.36 Mockup giao diện cài đặt của ứng dụng

### Xây dựng ứng dụng

* Xây dựng màn màn hình xem dữ liệu mới nhất

Phần giao diện bằng việc kết hợp nhiều LinearLayout lại với nhau ta bố trí được các thành phần theo mô tả trong phần thiết kế.

Bao quanh ứng dụng là LinearLayout tổng được sắp xếp theo chiều dọc, gồm 2 thành phần là 1 TextField ở phía trên và một ListView ở phía dưới. Mỗi thành phần nhỏ trong ListView cũng được bố trí bằng LinearLayout. Màu sắc và ảnh mô tả mức độ ô nhiểm của các thành phần con trong ListView được quy định giống như trong phần xây dựng web.

Khi khởi tạo giao diện ta sẽ gọi đến api thông qua retrofit để lấy dữ liệu. Dữ liệu dạng JSON trả về được retrofit parse vào đối tượng. Thông qua Adapter của ListView ta lấy được dữ liệu cần hiển thị ra và cập nhật lên giao diện.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Hình 3.37 Giao diện xem dữ liệu mới nhất

* Xây dựng màn hình xem dữ liệu theo tháng

Màn hình này tương tự với màn hình xem dữ liệu mới nhất. Khác ở điểm là ListView của màn hình này là các biểu đồ.

Khi khởi tạo màn hình, retrofit gọi đến api để lấy danh sách địa điểm có thiết bị đo, sau đó ứng dụng mặc định chọn địa điểm đầu tiên trong danh sách để lấy dữ liệu.

Dữ liệu trả về là một mảng số liệu được đo trong tháng. Retrofit có nhiệm vụ parse mảng thành các đối tượng, ta sử dụng ArrayList để lưu các đối tượng này lại.

Cuối cùng mảng này được nạp vào thư viện MPChart để vẽ biểu đồ.

A close up of a map

Description automatically generatedA screenshot of a map

Description automatically generated

Hình 3.38 Giao diện xem dữ liệu theo tháng

* Xây dựng màn hình cài đặt

Màn hình này khá đơn giản, gồm một button gọi đến Activity cài đặt wifi cho thiết bị đó, một button group để chuyển đổi giữa độ C và độ F.

Khi người dùng chọn vào button “SMART CONFIG” ứng dụng sẽ tạo một Intent đến Activity cài đặt wifi. ESP cung cấp sẵn mã nguồn của ứng dụng thiết lập wifi cho thiết bị, ta chỉ cần thêm thư viện của ứng dụng này vào và viết lại Activity cho phù hợp.

Khi button group độ C, độ F được nhấn, ta bắt sự kiện nút nhấn lại và lưu cài đặt này vào sharePreference của Android. SharePreference của Android có dạng như hashmap, được lưu theo key-value.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Hình 3.39 Giao diện cài đặt của ứng dụng

# KẾT QUẢ

## Kết quả đạt được

### Phần cứng

Nhóm đã hoàn thành thiết kế thiết bị AirNow với một ngõ sạc 5V DC và một nút nguồn. Thiết bị có kích thước 10x10x8 (cm)

Vỏ hộp của thiết bị được thiết kế để bảo vệ mạch và linh kiện bên trong và đảm bảo an toàn cho người dùng.

A picture containing sitting, metal, open, black

Description automatically generated

Hình 4.40 Mặt trước của thiết bị

A picture containing black, sitting, small, suitcase

Description automatically generated

Hình 4.41 Mặt bên của thiết bị

### Phần mềm

Qua 3 tháng tìm hiểu và phát triển, nhóm đã xây dựng được hệ thống theo dõi chất lượng không khí:

* Thiết bị:
  + Áp dụng thành công KIT ESP32 vào việc tính toán, thu thập và gửi dữ liệu đo được lên server.
  + Áp dụng hệ điều hành FreeRTOS vào chương trình hoạt động của thiết bị AirNow.
  + Sử dụng smart config để thiết bị có thể kết nối vào mạng wifi dễ dàng hơn.
  + Xây dựng thành công chương trình firmware có chức năng: Lưu SSID và mật khẩu của wifi vào bộ nhớ điện tĩnh, giao tiếp với hệ thống server.
  + Với thời gian lấy mẫu 7 phút, chỉ số AQI rất ổn định, độ chính xác tương đối cao.
* Ứng dụng trên điện thoại thông minh:
* Xem dữ liệu mới nhất
* Xem dữ liệu tháng dạng biểu đồ
* Chọn vị trí để xem dữ liệu
* Cài wifi cho thiết bị đo
* Cài đặt hiển thị nhiệt độ theo độ C hoặc độ F
* Ứng dụng trên nền tảng web:
* Xem dữ liệu mới nhất
* Xem dữ liệu tháng dạng biểu đồ
* Chọn vị trí để xem dữ liệu
* Quản lý thông tin đăng nhập của người dùng
* Quản lý thông tin của thiết bị đo
* Địa chỉ trang web: <http://airnowuit.xyz/>
* Server:
* Kiểm tra đăng nhập
* Cung cấp các api cho thiết bị, ứng dụng web và ứng dụng di động
* Lưu trữ dữ liệu đo được từ thiết bị

### Thực nghiệm

Sau nhiều lần đo thực tế, nhóm nhận thấy nhiệt độ và độ ẩm thể hiện được gần chính xác giá trị thực tế. Chỉ số mật độ hạt bụi có chênh lệch rất nhiều.

Bảng 4.18 Kết quả đo thực tế tại đại sứ quán Hoa Kỳ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lần đo | Chỉ số trung bình | Thiết bị AirNow  (1) | Chỉ số tại đại sứ quán Hoa Kỳ Hồ Chí Minh  (2) | Độ chênh lệch  (1)-(2) |
| Lần 1 | Nhiệt độ | 31.07 | 30.00 | 1.07 oC |
| Độ ẩm | 56.00 | 51.00 | 4 % |
| Mật độ hạt bụi | 28.40 | 61.30 | -32.9 ug/m3 |
| Lần 2 | Nhiệt độ | 32.86 | 32.00 | 0.86 oC |
| Độ ẩm | 57.73 | 54.00 | 3.73 % |
| Mật độ hạt bụi | 23.91 | 41.90 | -17.99 ug/m3 |
| Lần 3 | Nhiệt độ | 33.73 | 31.00 | 2.73 oC |
| Độ ẩm | 54.8 | 53.00 | 1.8 % |
| Mật độ hạt bụi | 50.75 | 31.10 | 19.65 ug/m3 |
| Lần 4 | Nhiệt độ | 32.23 | 32.00 | 0.23 oC |
| Độ ẩm | 56.35 | 55.00 | 1.35 % |
| Mật độ hạt bụi | 49.57 | 54.10 | -4.53 ug/m3 |
| Lần 5 | Nhiệt độ | 32.89 | 32.00 | 0.89 oC |
| Độ ẩm | 53.77 | 55.00 | -1.23 % |
| Mật độ hạt bụi | 79.86 | 88.4 | -8.54 ug/m3 |

Sau khi nhóm thực hiện điều chỉnh thời gian lấy mẫu thành 7 phút thì chỉ số mật độ hạt bụi đã gần sát với giá trị thực tế.

Bảng 4.19 Kết quả đo sau khi thay đổi thời gian lấy mẫu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lần đo | Chỉ số trung bình | Thiết bị AirNow  (1) | Chỉ số tại đại sứ quán Hoa Kỳ Hồ Chí Minh  (2) | Độ chênh lệch  (1)-(2) |
| Lần 1 | Nhiệt độ | 31.42 | 31 | 0.42 oC |
| Độ ẩm | 45.66 | 43 | 2.66 % |
| Mật độ hạt bụi | 16.3 | 14 | 2.3 ug/m3 |
| Lần 2 | Nhiệt độ | 34.81 | 35 | -0.19 oC |
| Độ ẩm | 34.54 | 35 | 0.46 % |
| Mật độ hạt bụi | 40.1 | 35.9 | 4.2 ug/m3 |
| Lần 3 | Nhiệt độ | 29.25 | 28 | 1.25 oC |
| Độ ẩm | 49.06 | 51 | -1.94 % |
| Mật độ hạt bụi | 23.5 | 18.7 | 4.8 ug/m3 |
| Lần 4 | Nhiệt độ | 34.41 | 34 | 0.41 oC |
| Độ ẩm | 30.33 | 32 | -1.67 % |
| Mật độ hạt bụi | 35.4 | 37.9 | -2.5 ug/m3 |
| Lần 5 | Nhiệt độ | 34.75 | 34 | 0.75 oC |
| Độ ẩm | 48.52 | 44 | 4.52 % |
| Mật độ hạt bụi | 25.8 | 20.1 | 5.7 ug/m3 |
| Lần 6 | Nhiệt độ | 32.8 | 32 | 0.8 oC |
| Độ ẩm | 46.07 | 48 | -1.93 % |
| Mật độ hạt bụi | 35.4 | 30.1 | 5.3 ug/m3 |
| Lần 7 | Nhiệt độ | 32.2 | 32 | 0.2 oC |
| Độ ẩm | 52.98 | 48 | 4.98 % |
| Mật độ hạt bụi | 12 | 18.7 | -6.7 ug/m3 |
| Lần 8 | Nhiệt độ | 33.31 | 32 | 1.31 oC |
| Độ ẩm | 50.18 | 51 | -0.82 % |
| Mật độ hạt bụi | 14.9 | 15.9 | -1 ug/m3 |
| Lần 9 | Nhiệt độ | 32.57 | 32 | 0.57 oC |
| Độ ẩm | 53.49 | 51 | 2.49 % |
| Mật độ hạt bụi | 26.8 | 21.1 | 5.7 ug/m3 |
| Lần 10 | Nhiệt độ | 27.28 | 27 | 0.28 oC |
| Độ ẩm | 74.11 | 69 | 5.11 % |
| Mật độ hạt bụi | 18.7 | 14.9 | 3.8 ug/m3 |

## Thuận lợi

* Trong quá trình nhiêu cứu và phát triển hệ thống nhóm chúng em nhận được sự hỗ trợ rất hữu ích từ giảng viên hướng dẫn, sự giúp đỡ của người thân và bạn bè về mặt kinh tế và kỹ thuật.

- Được thầy phụ trách hỗ trợ về các thiết bị phần cứng.

- Lượng kiến thức cần thiết được truyền đạt từ các thầy cô khoa Kỹ Thuật Máy Tính và kinh nghiệm làm việc thực tế từ các Đồ án môn học của khoa đã giúp ích rất nhiều cho khóa luận này của chúng em.

## Khó khăn

* Phân chia công việc và thực hiện không phù hợp với thời gian đã đề ra.
* Kỹ năng thiết kế và đóng gói thiết bị còn yếu nên dẫn tới việc vỏ hộp chưa chắc chắn, việc bố trí vị trí của các thiết bị còn hạn chế.
* Kỹ năng lên kế hoạch thực hiện đề tài của nhóm còn kém.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Tổng kết

Sau quá trình nghiên cứu, tìm hiểu, học hỏi, nhóm đã thực hiện thành công đề tài: “**xây dựng hệ thống theo dõi chất lượng không khí**”. Về cơ bản, nhóm đã hoàn thành mục tiêu ban đầu đặt ra như:

* Xây dựng được thiết bị theo dõi chất lượng không khí AirNow có thể xem thông tin của môi trường vào thời điểm hiện tại với sai số 20% so với thiết bị đo ở đại sứ quán Hoa Kỳ. Tuy sai số lớn hơn so với mục tiêu đề ra ban đầu, nhưng kết quả vẫn nằm trong khoảng chấp nhận được, có thể dùng kết quả này để đánh giá chính xác tình trạng không khí.
* Xây dựng server có thể theo dõi các thông tin môi trường.
* Thiết kế ứng dụng di động với giao diện cơ bản giúp người dùng có thể theo dõi các thông tin môi trường, tương tác với hệ thống.

## Hạn chế và hướng phát triển

* Hạn chế:
* Mạch nóng nhanh khi thiết bị hoạt động.
* Vẫn còn độ chênh lệch giữa chỉ số hạt đo được của thiết bị và chỉ số hạt đo được thực tế.
* Thiết bị chưa có tính bảo mật cao.
* Đóng gói thiết bị chưa chắc chắn.
* Cần điện thoại thông minh để gửi mật khẩu wifi.
* Tốc độ server chỉ đủ đáp ứng các yêu cầu.
* Chức năng smart config của ứng dụng android chỉ sử dụng được khi điện thoại kết nối vào wifi, không thể sử dụng khi mở chế độ HostPost.
* Hướng phát triển:
* Thiết bị có thể tự định vị được vị trí thông qua GPS.
* Thiết bị có thể đo được hướng và lực gió
* Thiết bị có thể đo thêm các hạt như CO2, SO2, PM10, …
* Tăng thêm thời lượng pin.
* Cập nhật firmware của thiết bị qua wifi.
* Kiểm tra được các thiết bị đang online.
* Mở rộng quy mô sử dụng, áp dụng hệ thống vào khu công nghiệp.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | C. Mgbemena, "Làm quen với Retrofit 2 HTTP Client," 16 12 2016. [Online]. Available: https://code.tutsplus.com/vi/tutorials/getting-started-with-retrofit-2--cms-27792. |
| [2] | Đ. Đạt, "Tìm hiểu về HTTP (HyperText Transfer Protocol)," 17 9 2018. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N. |
| [3] | danhphan1307, "[Series RTOS] FreeRTOS trên Cortex-M3 (Phần 1)," 12 1 2017. [Online]. Available: https://kipalog.com/posts/Series-RTOS--FreeRTOS-tren-Cortex-M3--Phan-1. |
| [4] | Dienmaygiadinh, "Bụi pm2.5 là gì? Vì sao bụi này gây tác hại đến sức khoẻ con người ?," 9 10 2018. [Online]. Available: https://dienmaygiadinh.net/tin-tuc/bui-pm2-5-la-gi-vi-sao-bui-nay-gay-tac-hai-den-suc-khoe-con-nguoi.html. |
| [5] | E. Magazine, "Các thành phần cơ bản của một ứng dụng Android," 8 11 2012. [Online]. Available: http://expressmagazine.net/development/1311/cac-thanh-phan-co-ban-cua-mot-ung-dung-android. |
| [6] | H. Kỳ, "Đăng ký Server Linux free với Amazon Web Service và hướng dẫn sử dụng cơ bản," 24 7 2019. [Online]. Available: https://viblo.asia/p/dang-ky-server-linux-free-voi-amazon-web-service-va-huong-dan-su-dung-co-ban-ORNZqj1el0n. |
| [7] | N. T. Sang, "SMARTCONFIG ESP8266," 24 01 2016. [Online]. Available: https://xivila.com/talks/detail/204/smartconfig-esp8266/. |
| [8] | o7planning.org, "Giới thiệu về AngularJS và Angular," 20 8 2018. [Online]. Available: https://o7planning.org/vi/12077/gioi-thieu-ve-angularjs-va-angular. [Accessed 30 9 2019]. |
| [9] | t. chien, "Lưu trữ dữ liệu time series với InfluxDB cùng bài toán về dữ liệu chứng khoán," 16 6 2019. [Online]. Available: https://viblo.asia/p/luu-tru-du-lieu-time-series-voi-influxdb-cung-bai-toan-ve-du-lieu-chung-khoan-gGJ59xnGlX2. [Accessed 2019]. |
| [10] | T. Quân, "Một cái nhìn tổng quan nhất về Nodejs," 23 4 2018. [Online]. Available: https://viblo.asia/p/mot-cai-nhin-tong-quan-nhat-ve-nodejs-Ljy5VeJ3lra. [Accessed 26 10 2019]. |
| [11] | thanhhtv.96, "MongoDB là gì? Cơ sở dữ liệu phi quan hệ," 3 9 2018. [Online]. Available: https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N. |
| [12] | Thegioididong, "Android thống trị thị trường smartphone toàn cầu, nhưng ở Mỹ thì không," 1 2 2018. [Online]. Available: https://www.thegioididong.com/tin-tuc/android-thong-tri-thi-truong-toan-cau-nhung-o-my-thi-khong-1063266. |
| [13] | Trung tâm quan trắc môi trường, "Quy trình tính toán và sử dụng AQI trong đánh giá chất lượng môi trường không khí xung quanh," in *PHƯƠNG PHÁP TÍNH TOÁN CHỈ SỐ CHẤT LƯỢNG KHÔNG KHÍ (AQI)*, Hà Nội, 2011, p. 29. |
| [14] | V. Hung, "Sử dụng AWS Free Tier một cách thực sự miễn phí!," 21 2 2019. [Online]. Available: https://viblo.asia/p/su-dung-aws-free-tier-mot-cach-thuc-su-mien-phi-bWrZnavQKxw. |
| [15] | VinaSupport, "Hướng dẫn đăng ký dịch vụ Amazon Web Services (AWS) miễn phí 1 năm," 25 1 2018. [Online]. Available: https://vinasupport.com/huong-dan-dang-ky-dich-vu-amazon-web-services-aws/. |
| [16] | Wikipedia, "Chỉ số chất lượng không khí," 7 12 2019. [Online]. Available: https://vi.wikipedia.org/wiki/Ch%E1%BB%89\_s%E1%BB%91\_ch%E1%BA%A5t\_l%C6%B0%E1%BB%A3ng\_kh%C3%B4ng\_kh%C3%AD. |
| [17] | ECE-19, "Particle Count to Concentration – Conversion Algorithm," in *Preliminary Screening System for Ambient Air Quality in Southeast Philadelphia*, Southeast Philadelphia, 2009, p. 6. |
| [18] | Espressif System CO., LTD, "ESP-IDF Programming Guide," 2019. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/. |
| [19] | Espressif Systems, ESP32 Technical Reference Manual, 2018. |
| [20] | Maxim Integrated Products, DS1307, 2015. |
| [21] | Sensirion Company, SHT3x\_datasheet, 2016. |
| [22] | SHINYEI Technology Co.,LTD, PPD42NJ Datasheet, 2015. |

1. Nhóm nhạy cảm bao gồm: trẻ em, người già và những người mắc bệnh hô hấp. [↑](#footnote-ref-1)