

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống quản lý giám sát trạm quan trắc thủy điện ứng dụng IoT

LƯƠNG ĐỨC HOÀ

hoa.ld151595@sis.hust.edu.vn

**Viện Công nghệ thông tin và Truyền thông
Chuyên ngành Kỹ thuật máy tính**

Giảng viên hướng dẫn: TS. Phạm Ngọc Hưng

Chữ ký của GVHD

Bộ môn: Kỹ thuật máy tính

Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 01/2021

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Lương Đức Hoà

Điện thoại liên lạc: 0366874653

Email: hoa.ld151595@sis.hust.edu.vn

Lớp: CNTT1.2 – K60

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại:

Thời gian làm ĐATN: Từ ngày .../.../2020 đến .../.../2021

2. Mục đích nội dung của ĐATN

Xây dựng hệ thống quản lý giám sát trạm quan trắc thủy điện ứng dụng công nghệ IoT.

3. Các nhiệm vụ cụ thể của ĐATN

- Thiết kế hệ thống thu thập dữ liệu, giám sát các trạm quan trắc cho nhà máy thủy điện.
- Thiết kế xây dựng kịch bản giao tiếp giữa thiết bị thu thập với máy chủ.
- Xây dựng phần mềm giám sát và quản lý trên nền website.

4. Lời cam đoan của sinh viên:

Tôi –*Lương Đức Hoà* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS Phạm Ngọc Hưng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

	<i>Hà Nội, ngày .. tháng .. năm 2021</i> Tác giả ĐATN <i>Lương Đức Hoà</i>
--	--

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

	<i>Hà Nội, ngày ... tháng ... năm 2021</i> Giáo viên hướng dẫn <i>TS Phạm Ngọc Hưng</i>
--	---

Lời cảm ơn

Xuyên suốt hơn 5 năm đại học cũng như trong quá trình hoàn thành đồ án tốt nghiệp em nhận được sự giúp đỡ và hỗ trợ về nhiều mặt của gia đình, thầy cô và bạn bè.

Em xin được gửi lời cảm ơn tới thầy Phạm Ngọc Hưng – giảng viên bộ môn Kỹ thuật máy tính, đã tận tình chỉ dẫn và giúp em có định hướng cũng như trong suốt quá trình hoàn thành đồ án tốt nghiệp của mình.

Em xin cảm ơn các bạn trong bộ môn Kỹ thuật máy tính Khoá 60, các bạn bè trên các diễn đàn – cộng đồng lập trình, cũng như các anh chị trên công ty đã hỗ trợ tư vấn về mặt kỹ thuật.

Em cũng xin gửi lời cảm ơn tới tất cả quý thầy cô trong trường Đại học Bách Khoa Hà Nội nói chung và thầy cô trong Viện Công nghệ thông tin và Truyền thông nói riêng đã đồng hành cùng em trong suốt quá trình học tập tại trường.

Cuối cùng, em xin phép dành lời kết để gửi lời cảm ơn tới gia đình và bạn bè đã luôn bên cạnh và ủng hộ em trong suốt những năm tháng đã qua để em có được ngày hôm nay.

Xin chân thành cảm ơn tất cả mọi người!

Tóm tắt nội dung đồ án

Ý tưởng của đồ án này nhằm nắm bắt xu thế “Internet of think - Vạn vật kết nối”; cùng với thực trạng tại các cơ sở nhà máy thủy điện, xử lý nước thải quy mô vừa và nhỏ mà bản thân em đã được may mắn tiếp xúc. Dưới những chia sẻ của những người đang trực tiếp làm việc tại các cơ sở này, do phạm vi thời gian và quy mô đồ án nên yêu cầu đặt ra của em đối với đề tài này là xây dựng một quy trình thu thập số liệu để hỗ trợ cho việc theo dõi số liệu thu thập tại các trạm đo, điểm đo của kỹ thuật viên thuận lợi và an toàn hơn.

Để thực hiện được yêu cầu này, em tận dụng các phần cứng đã và đang được lắp đặt ở hầu hết các công trình này làm nguồn vào dữ liệu. Bên cạnh đó, em xây dựng một hệ thống bao gồm một Broker Message Queue để thu thập dữ liệu, phần server tiếp nhận và lưu trữ dữ liệu Broker thu thập được, và website hiển thị phần dữ liệu này cùng một số chức năng quản lý cơ bản.

Đối với Broker Message Queue em chọn công nghệ RabbitMQ (công nghệ message queue được ưu tiên sử dụng phổ biến hiện nay), sử dụng RabbitMQ service để cài đặt trên hệ điều hành window 10 – máy cá nhân để cung cấp giao thức trao đổi tin nhắn trực tiếp. Đối với server và website em sử dụng công cụ IntelliJ lập trình framework Spring MVC (Model – View – Controller) ngôn ngữ Java chạy trực tiếp trên Apache Tomcat Server (được tích hợp trực tiếp trên IntelliJ) do phù hợp với điều kiện kỹ năng bản thân và mong muốn trau dồi kiến thức và kinh nghiệm để làm việc lâu dài với java web.

Sau một quá trình dày công nghiên cứu và xây dựng, hệ thống đã được em hoàn thành đáp ứng tương đối đầy đủ các chức năng theo yêu cầu của đề tài.

Qua quá trình tìm hiểu cũng như tham khảo ý kiến từ chính các đơn vị thi công và vận hành, mô hình em xây dựng có khả năng tiến đến thương mại hoá nếu được đầu tư bài bản và hoàn thiện đầy đủ các chức năng thiên về quản trị hệ thống và kinh doanh bán hàng.

Sau quá trình tìm hiểu và hoàn thành đồ án, em đã bổ sung cho mình thêm rất nhiều kỹ năng cứng về sử dụng ngôn ngữ Java, HTML, CSS, Javascripts... trên phần mềm IntelliJ, kỹ năng cài đặt, quản trị cũng như sử dụng RabbitMQ Broker, lý năng phân tích yêu cầu và giải quyết bài toán thực tế, bên cạnh việc hoàn thiện các kỹ năng mềm như kỹ năng giao tiếp, làm báo cáo, thuyết trình, lập kế hoạch...

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	ii
DANH MỤC HÌNH VẼ.....	vii
DANH MỤC CÁC TỪ VIẾT TẮT	ix
DANH MỤC BẢNG BIỂU	x
DANH MỤC THUẬT NGỮ.....	xi
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng và giải pháp.....	2
1.4 Bố cục đồ án.....	2
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG QUẢN LÝ GIÁM SÁT TRẠM QUAN TRẮC THỦY ĐIỆN	3
2.1 Phân tích - Khảo sát tình trạng thực tế.....	3
2.2 Thiết kế tổng quan hệ thống quản lý giám sát trạm điện	3
2.3 Thiết kế bộ thiết bị thu thập điều khiển cho trạm quan trắc	4
2.3.1 Bộ thiết bị thu thập	4
2.3.2 Lựa chọn thiết bị phần cứng và đề xuất giải pháp triển khai.....	4
2.3.3 Thiết kế giải pháp và giao thức giao tiếp.....	5
2.4 Broker MQTT	7
2.4.1 Giới thiệu chung	7
2.4.2 Lựa chọn công nghệ, giải pháp xây dựng và sử dụng Broker MQTT.....	7
2.5 Thiết kế phần mềm quản lý trên nền website.....	8
2.5.1 Tổng quan chức năng.....	8
2.5.2 Danh sách use case	12
2.5.3 Đặc tả chức năng	12
2.5.4 Thiết kế chi tiết các use case chính.....	15
2.5.5 Thiết kế cơ sở dữ liệu	16
2.5.6 Thiết kế giao diện	19
2.5.7 Lựa chọn công nghệ và ứng dụng.....	21
CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG	36
3.1 Mô hình triển khai hệ thống.....	36
3.2 Triển khai chi tiết từng thành phần	36

3.2.1	Bộ thiết bị thu thập	36
3.2.2	Broker RabbitMQ	36
3.2.3	Phần mềm website quản lý	42
3.3	Kết quả đạt được	45
CHƯƠNG 4. KẾT LUẬN.....		46
4.1	Kết luận	46
4.2	Ưu điểm.....	46
4.3	Nhược điểm.....	46
4.4	Hướng phát triển và bổ sung tính năng	46
TÀI LIỆU THAM KHẢO		47

DANH MỤC HÌNH VẼ

- Hình 2.1 Sơ đồ tổng quan hệ thống
- Hình 2.5 Biểu đồ use case tổng quan hệ thống
- Hình 2.6 Biểu đồ use case "Quản lý - phân quyền người dùng"
- Hình 2.7 Biểu đồ use case "Giám sát kết quả đo"
- Hình 2.8 Biểu đồ use case "Báo cáo thống kê"
- Hình 2.9 Biểu đồ use case "Quản lý trạm đo"
- Hình 2.10 Biểu đồ use case "Quản lý thiết bị đo"
- Hình 2.11 Biểu đồ use case "Thu thập và lưu trữ"
- Hình 2.12 Biểu đồ tuần tự use case "Giám sát kết quả đo"
- Hình 2.13 Biểu đồ tuần tự use case "Báo cáo thống kê"
- Hình 2.14 Biểu đồ tuần tự use case "Thu thập lưu trữ"
- Hình 2.13 Biểu đồ tuần tự use case "Báo cáo thống kê"
- Hình 2.14 Biểu đồ tuần tự use case "Thu thập lưu trữ"
- Hình 2.15 Database Diagram
- Hình 2.16 Thiết kế thô giao diện theo dõi
- Hình 2.17 Thiết kế thô giao diện thống kê báo cáo
- Hình 2.18 Thiết kế thô giao diện quản lý người dùng
- Hình 2.19 Mô hình MVC
- Hình 2.20 Ví dụ đồ thị sử dụng Highcharts
- Hình 2.21 Cấu trúc thư mục và tệp trong dự án Java Spring
- Hình 3.1 Sơ đồ triển khai hệ thống
- Hình 3.2 Lựa chọn phiên bản Erlang trên website
- Hình 3.3 Các bước cài đặt Erlang
- Hình 3.4 Vị trí download bộ cài RabbitMQ
- Hình 3.5 Thông báo lỗi khi chưa cài đặt Erlang
- Hình 3.6 Các bước cài đặt RabbitMQ
- Hình 3.7 Kết quả cài đặt RabbitMQ
- Hình 3.8 Enable plugin mqtt
- Hình 3.9 RabbitMQ service – start
- Hình 3.10 Giao diện đăng nhập RabbitMQ
- Hình 3.11 Giao diện quản lý Broker RabbitMQ
- Hình 3.12 Dịch vụ MQTT trên Broker RabbitMQ
- Hình 3.13 Nhập thông tin kết nối

Hình 3.14 Kết quả kết nối thành công MQTT Broker

Hình 3.15 Kiểm thử chức năng publish và subscribe

Hình 3.16 Giao diện đăng ký

Hình 3.17 Giao diện đăng nhập

Hình 3.18 Giao diện theo dõi giám sát

Hình 3.19 Giao diện quản lý trạm đo

Hình 3.20 Giao diện thêm mới trạm đo

Hình 3.21 Giao diện quản lý thiết bị đo

Hình 3.22 Giao diện thêm mới thiết bị đo

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
PLC S7 1200	Bộ điều khiển logic khả trình
SIMATIC-IOT2040	Một loại gateway dữ liệu thông minh
IoT	Internet Of Things – vạn vật kết nối
MQTT	Message Queuing Telemetry Transport - Giao thức truyền thông điệp
Json	JavaScript Object Notation - Kiểu dữ liệu mở trong JavaScript
http	Hypertext Transfer Protocol
MVC	Model – View – Controller
MSSQL Server	Microsoft SQL Server
CSDL	Cơ sở dữ liệu
HTML	Hypertext Markup Language - Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets - Các tập tin định kiểu theo tầng
AMQP	Advanced Message Queue Protocol – Giao thức nhận tin nhắn sử dụng hàng đợi
MySQL	Hệ quản trị cơ sở dữ liệu quan hệ
JRE	Java Runtime Environment – môi trường chạy java
API	Application Programming Interface – giao diện lập trình ứng dụng
ORM	Object Relationship mapping
IDE	Integrated Development Environment – môi trường viết code để phát triển ứng dụng

DANH MỤC BẢNG BIỂU

Bảng 2.1 Danh sách use case

Bảng 2.2 Đặc tả use case “Giám sát kết quả đo”

Bảng 2.3 Đặc tả use case “Báo cáo thống kê”

Bảng 2.4 Đặc tả use case "Thu thập và lưu trữ"

Bảng 2.5 Chi tiết bảng user_ - bảng người dùng

Bảng 2.6 Chi tiết bảng role – bảng quyền

Bảng 2.7 Chi tiết bảng station – bảng trạm đo

Bảng 2.8 Chi tiết bảng device – bảng thiết bị đo

Bảng 2.9 Chi tiết bảng value – bảng kết quả đo

DANH MỤC THUẬT NGỮ

Thuật ngữ	Ý nghĩa
Broker/MQTT Broker	Máy chủ môi giới
Java	Ngôn ngữ lập trình hướng đối tượng
Server	Máy chủ
Use case	Yêu cầu chức năng
Mqtt message	Gói tin/tin nhắn mqtt
Framework	Thư viện code viết sẵn
Hibernate	Thư viện kết nối và giao tiếp với database
Database	Trung tâm/kho lưu trữ dữ liệu
Just-In-Time	Đúng số lượng – tại đúng nơi – đúng thời điểm
Message	Tin nhắn
Logs	Vết hoạt động
Value	Giá trị - kết quả
Control	Điều khiển
Service	Dịch vụ
Controller	Bộ điều khiển – điều hướng
Status	Trạng thái
Enable	Kích hoạt
Public	Công khai
Private	Riêng tư

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Hiện nay, PLC S7 1200 hoặc mới hơn là bộ máy tính nhúng SIMATIC-IOT2040 (đều do Siemens cung cấp) là một trong các trang thiết bị phần cứng được sử dụng tại một số nhà máy để phục vụ đọc dữ liệu đo từ các thiết bị đo lường (nhiệt độ, dòng điện, điện áp, mực nước, chất lượng nước thải, ...) cùng với thực thi một số chức năng điều khiển chuyên biệt (điều khiển đóng - ngắt, ...). Tuy nhiên lại chưa ứng dụng nhiều hơn các công nghệ hiện đại hơn phục vụ các công việc này. Cùng với đó, thực trạng tại một số đơn vị, cán bộ công nhân viên thường xuyên phải ra tận điểm đo để thu thập dữ liệu đo bằng cách ghi chép và lưu trữ dưới dạng văn bản cứng (tài liệu giấy). Những điều này đem đến những khó khăn và bất tiện nhất định cũng như đôi chút lãng phí trong việc sử dụng công nghệ (các thông tin trên thu thập một trong các đơn vị thi công lắp đặt trang thiết bị cũng như qua một số cơ hội đi thực tế tại hiện trường của bản thân).

Trước thực trạng và nhu cầu của một số đơn vị cũng như tham khảo ý kiến trực tiếp từ phía doanh nghiệp cũng như qua tìm hiểu các công nghệ hiện hành, đánh giá việc xây dựng hệ thống này là cần thiết, hỗ trợ cho công việc theo dõi cũng như thu thập, lưu trữ kết quả đo đạc được tiện lợi và lâu bền, đồng thời hướng tới làn sóng IOT của thời đại 4.0 đang ngày một phát triển và lan rộng.

1.2 Mục tiêu và phạm vi đề tài

Trong đồ án tốt nghiệp này em tập trung vào các vấn đề liên quan đến xây dựng giao thức thu thập dữ liệu để phục vụ theo dõi giám sát và sinh báo cáo kết quả đo đạc trên nền tảng website. Bên cạnh đó bổ sung thêm một số chức năng quản lý đối với người dùng. Ứng dụng hoàn thành sẽ đáp ứng được nhu cầu của người sử dụng và tích hợp thêm các chức năng quản lý để phù hợp với thực tế. Cụ thể các tính năng của hệ thống được liệt kê như sau:

- Xây dựng giao thức giao tiếp giữa máy chủ webserver và máy tính nhúng thu thập đóng gói dữ liệu.
- Cung cấp dịch vụ kênh giao tiếp với giao thức MQTT bên cạnh các chức năng quản trị riêng biệt của Broker.
- Tiếp nhận dữ liệu dưới dạng message mqtt (hoặc gói tin http định dạng json) và lưu trữ.
- Truy vấn dữ liệu hiển thị dưới dạng đồ thị hỗ trợ việc theo dõi số đo theo khoảng thời gian.
- Quản lý người dùng.
- Quản lý trạm đo (bao gồm cả phân quyền người dùng...).
- Quản lý thiết bị đo.
- Sinh báo cáo kết quả đo đạc theo khoảng thời gian xác định.

Toàn bộ hệ thống hoàn thành sẽ cung cấp một phần mềm tương đối hoàn chỉnh với các chức năng cần thiết như được mô tả trên đây.

1.3 Định hướng và giải pháp

Trên hệ thống này, hệ thống xây dựng chức năng chính về giao thức thu thập dữ liệu phục vụ theo dõi giám sát và sinh báo cáo và các chức năng quản lý khác.

Phần bộ thiết bị thu thập và đóng gói kết quả đo, hệ thống xây dựng dựa trên các công nghệ có sẵn tại cơ sở bao gồm các đồng hồ đo đa năng, S7 1200 và IOT2040 sẽ được giới thiệu ở phần tiếp theo.

Phần tiếp nhận dữ liệu và website quản lý em cung cấp dịch vụ MQTT thông qua RabbitMQ broker – một công nghệ tiên tiến được sử dụng phổ biến hiện nay. Đối với website, em sử dụng ngôn ngữ Java cùng một số framework như Hibernate, Sverlet... với mô hình Spring MVC do định hướng phát triển của hệ thống lớn cũng như phù hợp với điều kiện của bản thân. CSDL sử dụng là MSSQL Server, hệ quản trị CSDL phổ biến và quen thuộc, tương thích tốt với java (kết nối thông qua Hibernate).

Với mục đích cung cấp một giao diện thân thiện, dễ sử dụng dành cho kỹ thuật viên của các nhà máy xí nghiệp nên em kết hợp sử dụng HTML, CSS, Javascript để xây dựng lên phần này.

1.4 Bố cục đồ án

Sau chương giới thiệu đề tài, báo cáo đồ án tốt nghiệp của em sẽ gồm các phần còn lại được mô tả dưới đây.

Chương 2 sẽ trình bày quá trình phân tích và tìm hiểu hiện trạng thực tế, từ đó xác định tác nhân cũng như phân tích thiết kế các phần chức năng, lựa chọn và giải thích công nghệ được ứng dụng cho hệ thống quản lý giám sát trạm quan trắc thủy điện

Chương 3 là phần mô tả việc triển khai chi tiết từng phần của hệ thống.

Chương cuối cùng là phần kết luận lại nội dung đồ án, đưa ra ưu nhược điểm trên hệ thống đã hoàn thành, những điều đã làm được và chưa làm được. Từ đó rút ra kinh nghiệm cũng như đưa ra phương án phát triển để cải tiến và hoàn thiện sản phẩm, hướng tới thương mại hoá trong tương lai.

Tiếp theo sẽ là chi tiết các chương của đồ án này.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG QUẢN LÝ GIÁM SÁT TRẠM QUAN TRẮC THỦY ĐIỆN

2.1 Phân tích - Khảo sát tình trạng thực tế

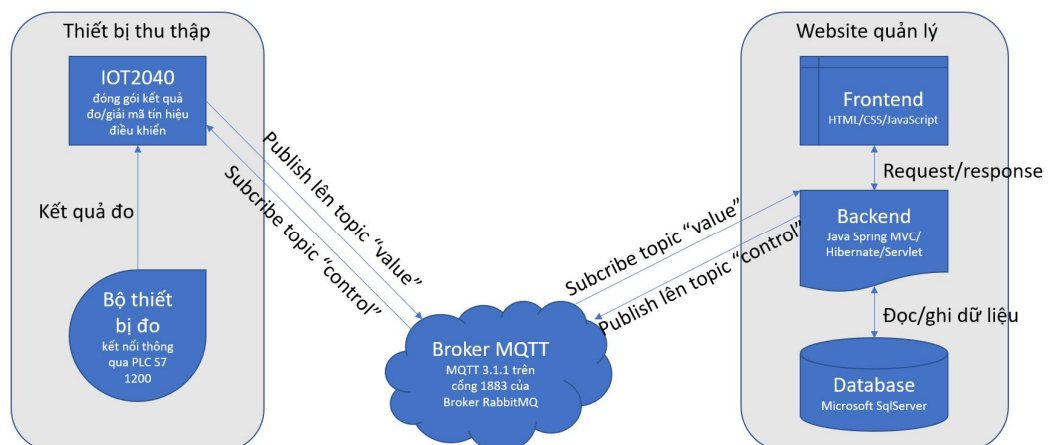
Trong thời gian ngắn có hội được tiếp xúc với môi trường làm việc tại một cơ sở nhà máy Thủy điện Suối Tân (thuộc địa bàn Mộc Châu, tỉnh Sơn La) cùng với thời gian làm việc với một đơn vị cung cấp và lắp đặt 2 thiết bị PLC S7 1200 và SIMATIC-IOT2040, em có một số thu hoạch sau: hiện tại PLC S7 1200 là một trong những thiết bị đang được sử dụng rộng rãi phục vụ theo dõi và một số chức năng điều khiển, kết quả đo thu được từ các thiết bị đo được hiển thị dưới dạng đồ thị, tuy nhiên lại được lưu trữ bằng nhập tay dữ liệu lên biểu mẫu báo cáo, rất mất thời gian và kém hiệu quả; tương tự như vậy đối với thiết bị SIMATIC-IOT2040.

Hình thức làm việc này đã hoạt động bình thường trong thời gian rất dài, không yêu cầu quá cao về chuyên môn và dễ dàng chuyển giao công việc cho người mới, tuy nhiên cũng có một số hạn chế sau:

- Vất vả trong quá trình lưu trữ dữ liệu.
- Dễ thất lạc bảng số liệu lưu trữ.
- Sai lệch của con người trong quá trình ghi chép.

Do những hạn chế này có thể thấy việc áp dụng công nghệ vào giải quyết chúng là điều cần thiết, nhằm khắc phục những nhược điểm trên đồng thời đảm bảo không hề ảnh hưởng đến hoạt động của hệ thống hiện hành.

2.2 Thiết kế tổng quan hệ thống quản lý giám sát trạm điện



Hình 2.1 Sơ đồ tổng quan hệ thống

Trên hình 2.1 bao gồm các thành phần của hệ thống này, bao gồm:

- Thiết bị đo: các loại sensor, cảm biến, đồng hồ ...trực tiếp đo giá trị từ ngoại cảnh.
- IOT2040: đóng vai trò làm thiết bị thu thập và đóng gói kết quả đo để gửi lên server hoặc publish lên broker MQTT.
- Broker: trạm chung chuyển tổng hợp các tin nhắn được gửi lên từ client vào một topic.
- Backend server: subscribe topic của broker để lắng nghe dữ liệu gửi lên và lưu lại xuống cơ sở dữ liệu; xử lý và trả lời các tương tác từ phía người dùng thực hiện trên giao diện và thực hiện yêu cầu các thao tác đối với cơ sở dữ liệu (nếu có).
- Database (cơ sở dữ liệu): lưu trữ và trả lời các truy vấn từ backend server.
- Frontend (giao diện): là nơi tương tác trực tiếp với người dùng, thực hiện các chức năng theo dõi, báo cáo, thống kê cũng như có thể thực hiện gửi gói tin điều khiển.

2.3 Thiết kế bộ thiết bị thu thập điều khiển cho trạm quan trắc

2.3.1 Bộ thiết bị thu thập

Bộ thiết bị thu thập phải bao gồm ít nhất 2 thành phần:

- Các cảm biến, đồng hồ đo, thiết bị đo có khả năng kết nối.
- Máy tính nhúng có cung cấp các giao thức kết nối tương thích, để đọc tín hiệu kết quả đo đạc từ các thiết bị trên. Đồng thời có khả năng đóng gói tùy biến và gửi các gói tin này với giao thức MQTT hoặc HTTP được cung cấp trong hệ thống.

Bộ thiết bị này nhằm mục đích đo đạc các thông số cần thiết, đóng gói lại và gửi lên hệ thống, cũng như có thể nhận và xử lý các gói tin điều khiển theo chiều ngược lại.

2.3.2 Lựa chọn thiết bị phần cứng và đề xuất giải pháp triển khai

Hệ thống này xây dựng sẽ lựa chọn các thiết bị dưới đây để đáp ứng yêu cầu đề ra và tương thích với một trong số các trang thiết bị sẵn có tại trạm thủy điện.



Hình 2.2 Bộ thiết bị thu thập đề xuất

*Chiều mũi tên là đường đi của dữ liệu.

“SIMATIC IOT2040” đóng vai trò làm máy tính nhúng tiếp nhận đóng gói dữ liệu, xử lý gói tin, “PLC S7 1200” là cầu nối, cung cấp các cổng kết nối với các thiết bị đo tương tự như “Đồng hồ đo đa năng” trên hình.

Hiện nay, tại một số trạm đã lắp đặt toàn bộ các thiết bị này hoặc các thiết bị với chức năng tương tự nhằm thu thập dữ liệu để theo dõi giám sát và một số chức năng điều khiển khác.

2.3.2.1. SIMATIC IOT2040

SIMATIC IOT2040 – cổng kết nối thông minh cho các giải pháp IoT trong công nghiệp – là một máy tính mạnh mẽ. SIMATIC IOT2040 có thể vận hành với hệ điều hành Yocto Linux, có thể tiếp nhận các ngôn ngữ lập trình bậc cao (như Java,...) thông qua các IDEs thông dụng (như Eclipse,...) cùng khả năng thu thập dữ liệu từ bộ điều khiển S7 1200. Hệ thống sử dụng thiết bị này với mục đích thu thập dữ liệu đo được từ các thiết bị kết nối trực tiếp với các cổng kết nối được cung cấp hoặc thông qua các thiết bị hỗ trợ khác, từ đó đóng gói và gửi gói tin theo giao thức HTTP lên trực tiếp server (gói tin json) hoặc theo giao thức MQTT lên MQTT broker.

SIMATIC IOT2040 cũng cung cấp rất nhiều các chức năng thú vị khác (có thể tham khảo tại tài liệu tham khảo [1]).

2.3.2.2. PLC S7 1200 và các thiết bị đo khác

Các trạm thủy điện thường lắp đặt rất nhiều các đồng hồ - thiết bị đo đa năng khác nhau để theo dõi giám sát từng vị trí và quy trình phức tạp của toàn nhà máy. Tuy nhiên các bộ theo dõi (tại các trạm thủy điện thường gọi là tủ điện) thường theo dõi một số loại thông số nhất định như: tần số dòng điện, điện áp, cường độ dòng điện, điện năng tiêu thụ, công suất máy; còn đối với vị trí trên đập thường là mực nước, lưu lượng dòng chảy. Các trạm sẽ lắp đặt các thiết bị đo tương ứng và kết nối với một hoặc nhiều thiết bị PLC S7 1200. Thiết bị này sẽ dẫn kết quả ra các đồng hồ điện tử hiện số hoặc các máy tính khác để phục vụ các mục đích khác nhau.

2.3.3 Thiết kế giải pháp và giao thức giao tiếp

2.3.3.3. Thiết kế giải pháp

Với việc SIMATIC IOT2040 cung cấp môi trường lập trình tương thích với các ngôn ngữ bậc cao, trên thiết bị này ta sẽ xây dựng một ứng dụng cung cấp chức năng của một MQTT Client cũng như của một HTTP Client. Ứng dụng này sẽ kết nối với các phần mềm sẵn có trên thiết bị để tiếp nhận dữ liệu và đóng gói lại dưới dạng gói tin json có định dạng như sau:

```
{ "id": "1", "value": "20", "time": "20210101003000" }
```

Trong đó “id” là mã serial của thiết bị đo, “value” là kết quả đo vào thời điểm “time”, ở đây biến thời gian sử dụng định dạng “YYYYmmDDhhMMss” (ví dụ

Trên mô hình sử dụng các thiết bị bao gồm “Module NodeMCU” (chứa ESP8266) vai trò tương tự IOT2040, “DHT11” là cảm biến đo nhiệt độ và độ ẩm, “Relay” chứa công tắc và cuối cùng là bóng đèn với các điều kiện nguồn điện vào và kết nối như hình (có thể sẽ bổ sung thêm một số thiết bị đo khác).

Mô hình này sẽ thay thế vai trò của “Bộ thiết bị thu thập” phía trên, nhằm cung cấp dữ liệu thử nghiệm cho hệ thống vì vậy nó sẽ tiến hành thực hiện các công việc được mô tả phía trên. Esp8266 trên NodeMCU sẽ sử dụng các thư viện “ESP8266WiFi.h” và “PubSubClient.h” để kết nối với MQTT Broker và thực hiện publish/subscribe, thư viện “DHT.h” để đọc giá trị từ “DHT11”, từ đó để thực hiện các công việc này.

2.4 Broker MQTT

2.4.1 Giới thiệu chung

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

Kiến trúc của MQTT bao gồm 2 thành phần chính là broker và clients.

Trong đó broker là trung tâm điểm giao của tất cả các client, đóng vai trò nhận message từ publisher (nhà xuất bản) và sắp xếp chúng theo một hàng đợi gửi tới một topic (chủ đề) cụ thể. Ngoài ra broker còn đảm nhiệm một số chức năng khác như bảo mật message, logs ...

Client có ít nhất một trong hai vai trò publisher, subscriber. Trong đó, publisher sẽ thực hiện gửi các message lên một topic. Sau đó, tất cả các subscriber nào đang subscribe topic này sẽ đều nhận được message đó.

MQTT tương thích với hầu hết các nền tảng hệ điều hành hiện nay.

Một số khái niệm thông dụng trong MQTT:

- Message: tin nhắn với định dạng mặc định là plain-text (chữ viết người có thể đọc được), tuy nhiên có thể dễ dàng tùy biến sang các định dạng khác để sử dụng cho mục đích của mình.
- Topic: có thể coi là một kênh giao tiếp chung giữa các publisher – subscriber với nhau. Topic có thể được khai báo dạng phân cấp (/cấp_1/cấp_2/...). Cùng với đó các subscriber có thể subscribe topic dạng “/topic_1/+” để theo dõi tất cả các kênh cùng cấp topic_1 hoặc “/topic_1/#” để theo dõi tất cả các kênh có cấp nhỏ hơn topic_1.

2.4.2 Lựa chọn công nghệ, giải pháp xây dựng và sử dụng Broker MQTT



Hình 2.4 Logo RabbitMQ

2.4.2.1. Giới thiệu RabbitMQ

RabbitMQ là một message broker (message-oriented middleware) sử dụng giao thức AMQP (Advanced Message Queue Protocol – Giao thức nhận tin nhắn sử dụng hàng đợi). Đây là chương trình đóng vai trò trung gian lưu trữ cũng như điều phối các yêu cầu (message) giữa người nhận và người gửi. Broker RabbitMQ dễ dàng cài đặt trên các hệ điều hành thông dụng.

Broker RabbitMQ là công nghệ hiện đại, mạnh mẽ và hỗ trợ rất nhiều tính năng thuận lợi cho các định hướng phát triển sau này. Tuy nhiên trong hệ thống này, ta chỉ sử dụng giao thức MQTT được hỗ trợ trên cổng 1883 (MQTT 3.1.1).

2.4.2.2. Giải pháp xây dựng

Tận dụng các tính năng trên cũng như các công nghệ được hỗ trợ được trên IOT2040, hệ thống này sẽ xây dựng client trên webserver và trên các thiết bị IOT2040 cùng kết nối tới một broker được cài đặt trên server cùng mạng.

Đối với phía webserver, client MQTT tại đây có vai trò làm subscriber, subscribe tới topic “value” để nhận toàn bộ các message trên topic này nhằm thu thập dữ liệu phục vụ cho chức năng theo dõi – giám sát. Bên cạnh đó, client này cũng đóng một vai trò khác là publisher tới topic “control” để thực hiện chức năng điều khiển đơn giản (nếu có).

Về phía IOT2040, client tại đây có vai trò publisher các dữ liệu đã được đóng gói gửi lên topic “value” và subscribe topic “control” để xử lý tín hiệu điều khiển (nếu có).

2.5 Thiết kế phần mềm quản lý trên nền website

2.5.1 Tổng quan chức năng

2.5.1.1. Biểu đồ use case tổng quan

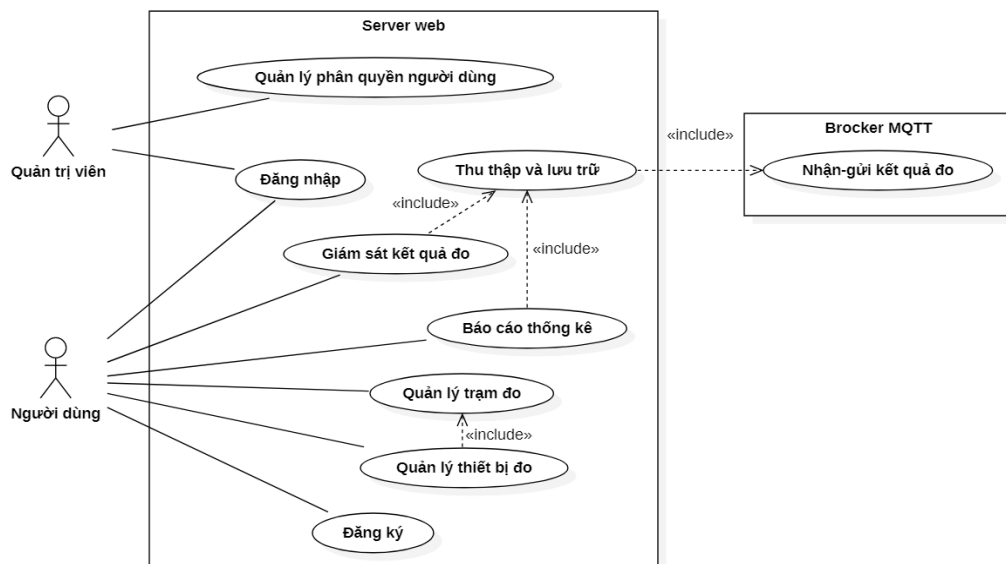
Các tác nhân tham gia hệ thống:

- Quản trị viên
- Người dùng

Vai trò của tác nhân trong hệ thống:

- a. Quản trị viên: Quản lý người dùng.
- b. Người dùng: Người sử dụng hầu hết các chức năng của hệ thống bao gồm: quản lý trạm đo, quản lý thiết bị, phân quyền trạm đo, theo dõi kết quả đo, sinh báo cáo thống kê kết quả đo...

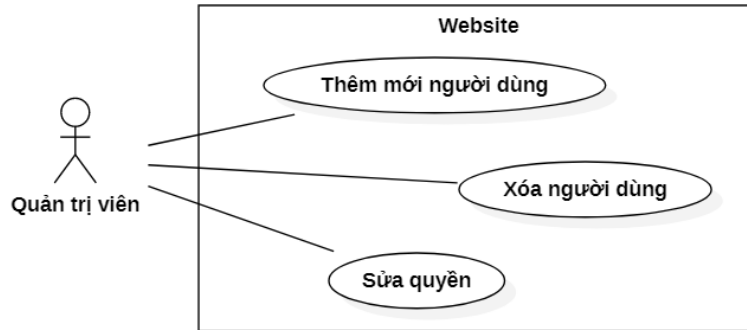
Bên cạnh đó là các use case hệ thống tương tác với Broker..



Hình 2.5 Biểu đồ use case tổng quan hệ thống

2.5.1.2. Biểu đồ use case phân rã “Quản lý - phân quyền người dùng”

Tác nhân của use case này là quản trị viên hệ thống. Trong hệ thống, người quản trị có vai trò thêm, xoá người dùng và phân quyền cho người dùng. Sau khi được tạo và phân quyền, người dùng mới có đầy đủ chức năng theo quyền hạn được cung cấp, có thể đăng nhập ngay lập tức vào hệ thống để thực thi các chức năng.



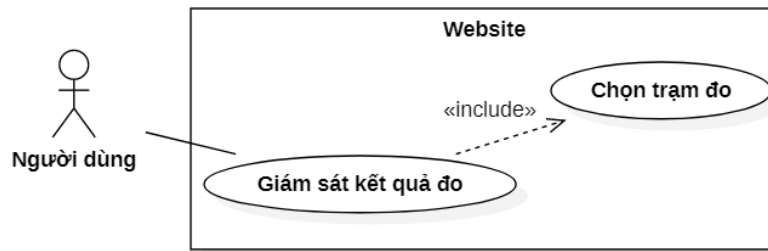
Hình 2.6 Biểu đồ use case "Quản lý - phân quyền người dùng"

Các use case con bao gồm “thêm mới người dùng”, “xoá người dùng”, “sửa quyền”:

- Thêm mới: Tài khoản được thêm mới lập tức có các chức năng theo quyền hạn được phân công.
- Xoá: Tài khoản sẽ bị xoá cùng với toàn bộ dữ liệu có liên quan đi kèm.
- Sửa quyền: Mỗi tài khoản sẽ chỉ có 1 quyền duy nhất, sau khi sửa quyền tài khoản sẽ được cập nhật quyền hạn mới với chức năng tương ứng, các dữ liệu thực thi khi còn ở quyền hạn cũ sẽ được giữ nguyên và chỉ có thể sử dụng khi trả lại quyền hạn ban đầu.

2.5.1.3. Biểu đồ use case phân rã “Giám sát kết quả đo”

Tác nhân của use case này sẽ là người dùng. Người dùng này sẽ được theo dõi và giám sát các thông số đo (cùng các chức năng đi kèm tương ứng) trên trạm đo của chính họ tạo ra, hoặc trạm đo được người khác cung cấp quyền theo dõi.

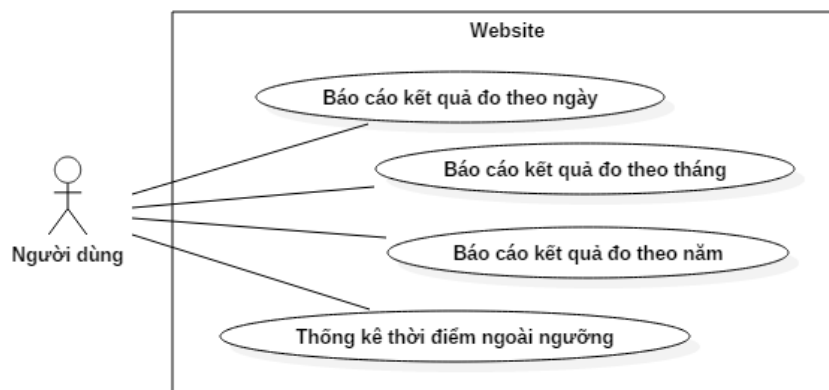


Hình 2.7 Biểu đồ use case “Giám sát kết quả đo”

Trong use case này, người dùng có thể lựa chọn trạm đo để theo dõi. Trên màn hình theo dõi sẽ là toàn bộ thông số mà trạm đo này đang thu thập, với các thông tin được cài đặt khi khởi tạo ban đầu hoặc như cập nhật mới nhất.

2.5.1.4. Biểu đồ use case phân rã “Báo cáo thống kê”

Trong use case này, người dùng lựa chọn loại báo cáo, chọn trạm đo, thiết bị đo, khoảng thời gian mà mình cần sinh báo cáo. Từ đó sinh báo cáo dưới dạng bảng biểu, đồ thị hoặc thống kê tương ứng.



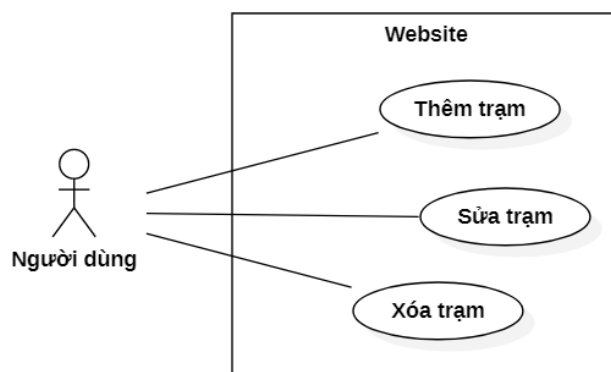
Hình 2.8 Biểu đồ use case “Báo cáo thống kê”

Các use case con bao gồm “báo cáo kết quả đo theo ngày”, “báo cáo kết quả đo theo tháng”, “báo cáo kết quả đo theo năm”, “thống kê thời điểm ngoài ngưỡng”.

- Báo cáo kết quả đo: Người dùng lựa chọn trạm đo, thiết bị đo, và chọn thời gian để sinh báo cáo dưới dạng bảng hoặc biểu đồ.
- Thống kê thời điểm ngoài ngưỡng: Người dùng lựa chọn khoảng thời gian cần thống kê, kết quả trả về sẽ là thống kê số lượng thời điểm ngoài ngưỡng của từng thiết bị trong trạm và chi tiết của các thống kê đó dưới dạng bảng.

*Các mẫu báo cáo thống kê tham khảo theo các màn hình xử lý dưới cơ hội được trực tiếp theo dõi thao tác của nhân viên nhà máy.

2.5.1.5. Biểu đồ use case phân rã “Quản lý trạm đo”

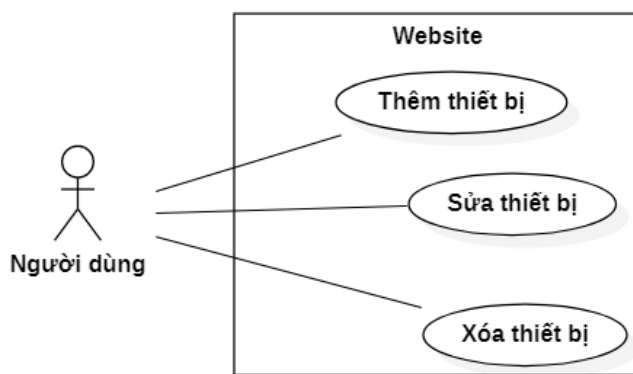


Hình 2.9 Biểu đồ use case “Quản lý trạm đo”

Người dùng là tác nhân của use case này. Các thao tác được thực thi trên trạm đo được tạo bởi chính người dùng này hoặc được cấp quyền từ trạm của người dùng khác.

- Thêm trạm: người dùng thêm trạm mới với các thông tin được yêu cầu.
- Sửa trạm: người dùng sửa các thông tin để cập nhật mới cho trạm.
- Xóa trạm: người dùng sẽ xóa trạm khỏi danh sách trạm do họ sở hữu, trạm sẽ bị xóa hoàn toàn khi không còn ai sở hữu hoặc được phân quyền trạm này nữa.

2.5.1.6. Biểu đồ use case phân rã “Quản lý thiết bị đo”



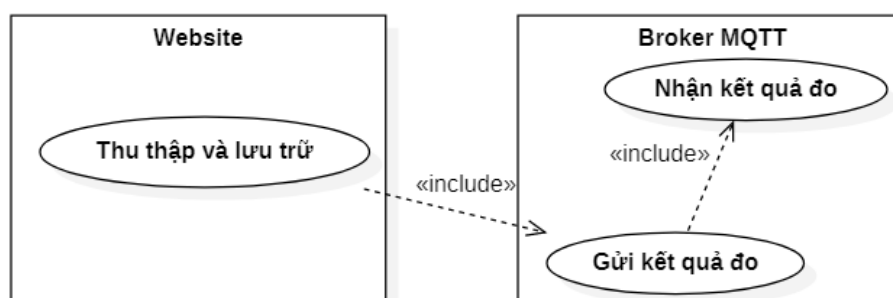
Hình 2.10 Biểu đồ use case “Quản lý thiết bị đo”

Người dùng có quyền quản lý trạm sẽ được quyền xử lý tương ứng với các thiết bị đo có trong trạm đo này.

- Thêm thiết bị: Tạo thêm thiết bị đo mới với các thông tin yêu cầu, cổng thu thập dữ liệu đo sẽ được kích hoạt khi thiết bị đo được tạo thành công.
- Sửa thiết bị: sửa các thông tin có trên bảng thiết bị đo.

- Xóa thiết bị: các dữ liệu và bản ghi thiết bị sẽ bị xóa khỏi các bảng tương ứng, cổng thu thập dữ liệu sẽ không còn hoạt động ghi lại các dữ liệu nếu vẫn tiếp tục gửi lên.

2.5.1.7. Biểu đồ use case phân rã “Thu thập và lưu trữ”



Hình 2.11 Biểu đồ use case “Thu thập và lưu trữ”

Use case này thể hiện tương tác giữa hai hệ thống web-server và Broker MQTT. Broker sẽ đóng vai trò trạm trung chuyển dữ liệu, nhận gói tin MQTT, web-server sẽ theo dõi kênh thông tin này để thu thập dữ liệu và lưu dữ liệu hợp lệ vào database.

2.5.2 Danh sách use case

Danh sách các use case của hệ thống được thống kê trong bảng 2.1 dưới đây.

Bảng 2.1 Danh sách use case

Mã use case	Tên use case
UC001	Đăng ký
UC002	Đăng nhập
UC003	Quản lý phân quyền người dùng
UC004	Giám sát kết quả đo
UC005	Báo cáo thống kê
UC006	Quản lý trạm đo
UC007	Quản lý thiết bị đo
UC008	Thu thập và lưu trữ

2.5.3 Đặc tả chức năng

Trong phần này sẽ trình bày chi tiết 3 use case chính có mã là UC004, UC005 và UC008.

2.5.3.8. Đặc tả use case “Giám sát kết quả đo”

Bảng dưới đây sẽ thể hiện phần đặc tả của use case “giám sát kết quả đo”. Chi tiết như bảng sau:

Bảng 2.2 Đặc tả use case “Giám sát kết quả đo”

Mã Use case	UC004	Tên Use case	Giám sát kết quả đo
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, được cấp quyền hoặc sở hữu ít nhất 1 trạm đo, trạm đo có ít nhất 1 thiết bị đang hoạt động tương thích với hệ thống này.		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Lựa chọn “theo dõi” trên menu.
	2.	Người dùng	Lựa chọn trạm đo trong danh sách trạm sở hữu và nhấn tìm kiếm.
	3.	Hệ thống	Trả lại giao diện giám sát bao gồm biểu đồ và kết quả đo mới nhất của các thiết bị đang hoạt động tương thích với hệ thống này.
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Hệ thống	Thông báo trạm chưa có thiết bị nào đang hoạt động.
	3b.	Người dùng	Tiến hành tạo thiết bị và liên hệ kích hoạt thiết bị tương thích với hệ thống.
Hậu điều kiện	Không có		

2.5.3.9. Đặc tả use case “Báo cáo thống kê”

Bảng 2.3 Đặc tả use case “Báo cáo thống kê”

Mã Use case	UC005	Tên Use case	Báo cáo thống kê
Tác nhân	Người dùng.		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, được cấp quyền hoặc sở hữu ít nhất 1 trạm đo, trạm đo có ít nhất 1 thiết bị đang hoạt động tương thích với hệ thống này.		
<u>Báo cáo theo ngày/tháng/năm.</u>			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Lựa chọn “báo cáo” trên menu.

	2.	Người dùng	Lựa chọn trạm đo trong danh sách trạm sở hữu, chọn thiết bị thuộc trạm đó (nếu cần thiết). Chọn báo cáo theo thời gian nào.
	3.	Hệ thống	Trả lại giao diện xem trước kết quả sinh báo cáo dưới định dạng tương ứng với chức năng “print” thông thường.
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Hệ thống	Trả về thông báo không có kết quả thích hợp với điều kiện để sinh báo cáo.
	3b	Người dùng	Tiến hành chọn lại điều kiện sinh báo cáo để có kết quả thích hợp.
<u>Thông kê thời điểm vượt ngưỡng</u>			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Lựa chọn “thống kê” trên menu.
	2.	Người dùng	Lựa chọn trạm đo trong danh sách trạm sở hữu, chọn thiết bị thuộc trạm đó (nếu cần thiết). Chọn thống kê theo thời gian nào.
	3.	Hệ thống	Trả lại giao diện xem trước kết quả sinh thống kê dưới định dạng tương ứng với chức năng “print” thông thường.
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Hệ thống	Trả về thông báo không có kết quả thích hợp với điều kiện để thống kê
	3b	Người dùng	Tiến hành chọn lại điều kiện sinh thống kê để có kết quả thích hợp.
Hậu điều kiện	Không có		

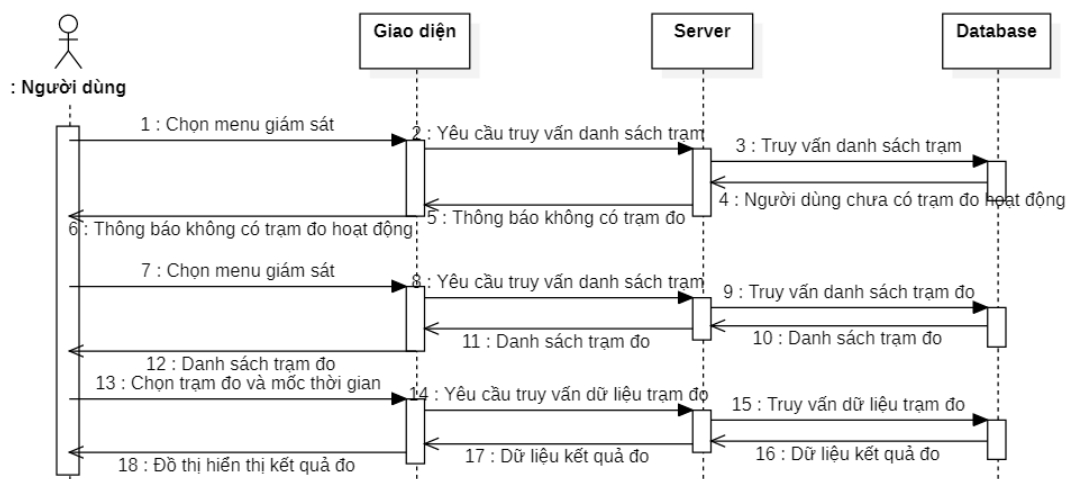
2.5.3.10. Đặc tả use case “Thu thập và lưu trữ”

Bảng 2.4 Đặc tả use case "Thu thập và lưu trữ"

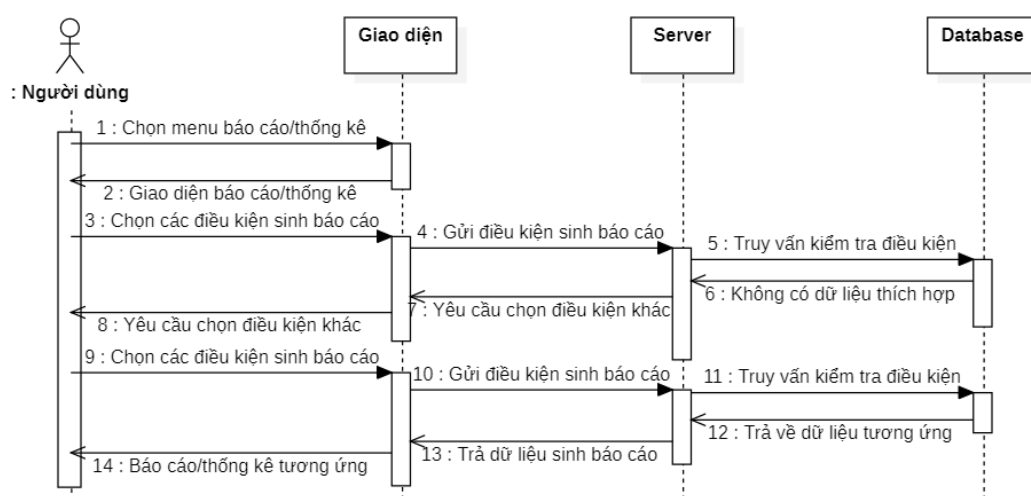
Mã Use case	UC008	Tên Use case	Thu thập và lưu trữ
Tác nhân	Hệ thống web-server và broker		
Tiền điều kiện	Trên hệ thống web đã khởi tạo trạm đo và thiết bị đo và đã được kết nối tới broker. Thiết bị đo được kết nối với bộ thu thập có tích hợp chức năng gửi dữ liệu dưới dạng tin nhắn mqtt và thiết bị này đã được kết nối với broker.		
<u>Thu thập</u>			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Broker	Lắng nghe tin nhắn từ các nguồn tương thích và trả lên một kênh giao tiếp được chỉ định.
	2.	Web-server	Theo dõi kênh giao tiếp trên để thu thập dữ liệu gửi lên
Luồng sự kiện thay thế	Không		
<u>Lưu trữ</u>			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Web-server	Tiếp nhận và kiểm tra các tin nhắn gửi về, lưu lại các dữ liệu hợp lệ và bỏ qua các dữ liệu khác.
Luồng sự kiện thay thế	Không		
Hậu điều kiện	Không		

2.5.4 Thiết kế chi tiết các use case chính

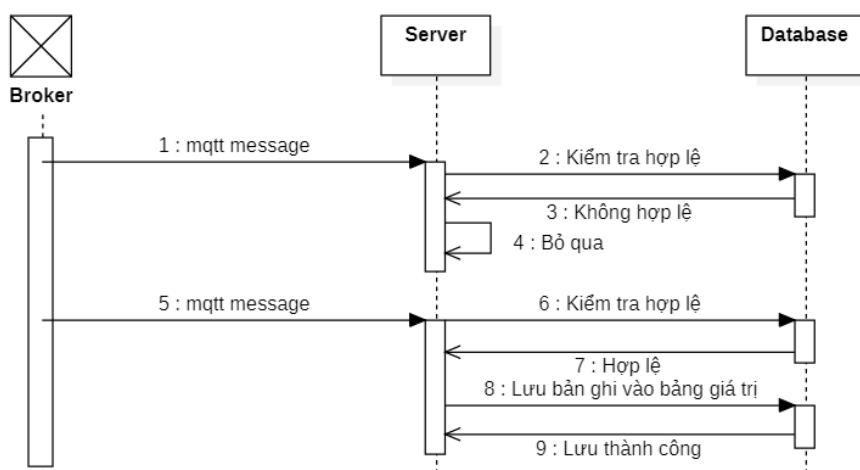
Trong phần này sẽ chỉ trình bày biểu đồ tuần tự của 3 use case chính là “Giám sát kết quả đo”, “Báo cáo thống kê” và “Thu thập lưu trữ”.



Hình 2.12 Biểu đồ tuần tự use case "Giám sát kết quả đo"



Hình 2.13 Biểu đồ tuần tự use case "Báo cáo thống kê"

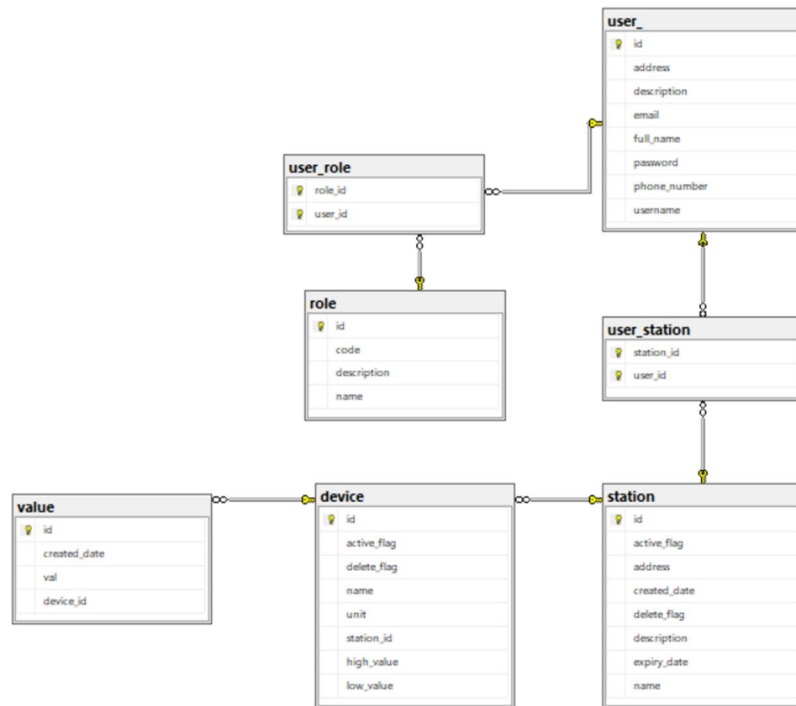


Hình 2.14 Biểu đồ tuần tự use case "Thu thập lưu trữ"

2.5.5 Thiết kế cơ sở dữ liệu

2.5.5.11. Phân tích mối quan hệ giữa các thực thể được lưu trữ

Hình 2.11 dưới đây mô tả mối quan hệ giữa các bảng được lưu trữ trong cơ sở dữ liệu.



Hình 2.15 Database Diagram

Bảng dưới đây sẽ mô tả chức năng các bảng có trên hình 2.12.

STT	Tên bảng	Mô tả chức năng
1	user_	Bảng lưu dữ liệu người dùng
2	role	Bảng lưu các quyền
3	user_role	Bảng lưu dữ liệu phân quyền người dùng
4	station	Bảng lưu thông tin trạm đo
5	device	Bảng lưu thông tin thiết bị đo
6	user_station	Bảng lưu mối quan hệ sở hữu giữa người dùng và trạm đo
7	value	Bảng lưu giá trị đo thu được

2.5.5.12. Thiết kế chi tiết một số bảng

Bảng 2.5 Chi tiết bảng user_ - bảng người dùng

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Giải thích
id	long	Có	Mã serial người dùng - tự sinh trong SQLServer
username	varchar	Có	Tên đăng nhập người dùng
password	varchar	Có	Mật khẩu người dùng
phone_number	varchar	Không	Số điện thoại người dùng

full_name	nvarchar(255)	Có	Họ tên người dùng
email	varchar	Không	Địa chỉ thư điện tử
address	nvarchar(255)	Không	Địa chỉ người dùng
description	nvarchar(255)	Không	Mô tả - ghi chú

Bảng 2.6 Chi tiết bảng role – bảng quyền

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Giải thích
id	long	Có	Mã serial quyền tự sinh
code	varchar	Có	Mã quản lý quyền
name	varchar	Có	Tên quyền
description	nvarchar(255)	Không	Mô tả - ghi chú

Bảng 2.7 Chi tiết bảng station – bảng trạm đo

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Giải thích
id	long	Có	Mã serial trạm tự sinh
name	nvarchar(255)	Có	Tên trạm
address	nvarchar(255)	Không	Địa chỉ trạm
description	nvarchar(255)	Không	Mô tả - ghi chú

Bảng 2.8 Chi tiết bảng device – bảng thiết bị đo

Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Giải thích
id	long	Có	Mã serial thiết bị đo tự sinh
station_id	long	Có	Mã serial trạm đo
name	nvarchar(255)	Có	Tên thiết bị đo
unit	varchar	Có	Đơn vị đo
low_value		Có	Ngưỡng thấp cảnh báo
high_value		Có	Ngưỡng cao cảnh báo

Bảng 2.9 Chi tiết bảng value – bảng kết quả đo

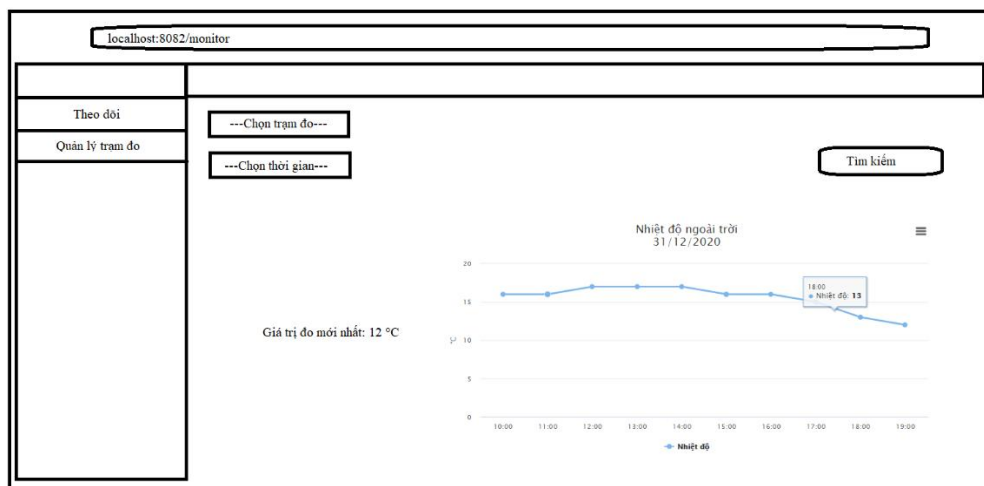
Tên thuộc tính	Kiểu dữ liệu	Bắt buộc	Giải thích
id	long	Có	Mã serial kết quả đo tự sinh
device_id	long	Có	Mã serial thiết bị đo

val		Có	Giá trị đo được
created_date	Datetime	Có	Thời gian nhận kết quả đo

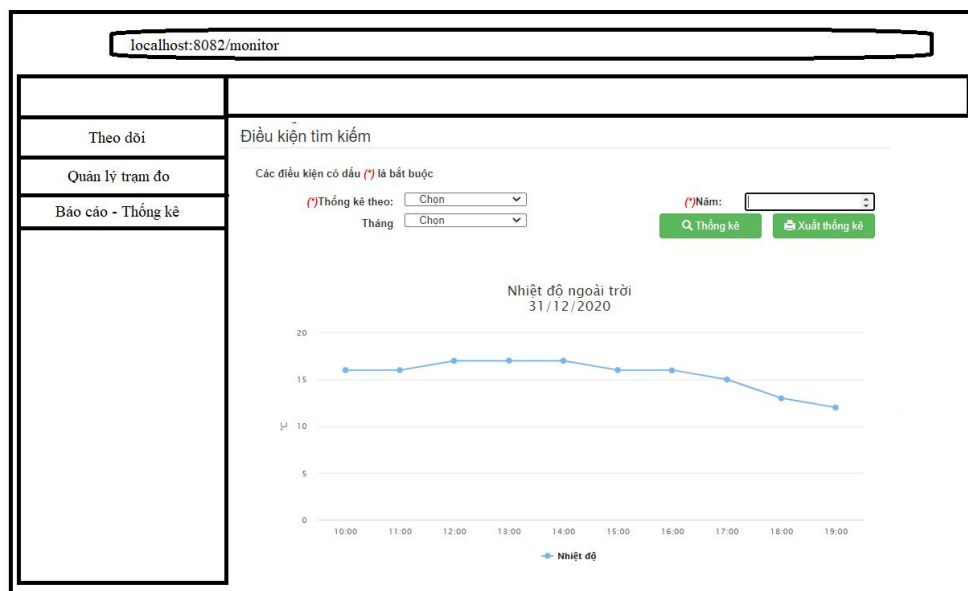
2.5.6 Thiết kế giao diện

Giao diện của website phải cơ bản dễ sử dụng, tập trung vào phần kỹ thuật để đảm bảo đầy đủ các chức năng hoạt động ổn định, không chú trọng quá nhiều vào châu chuốt giao diện. Các giao diện chính bao gồm:

- Giao diện đăng ký/đăng nhập.
- Giao diện quản lý/phân quyền người dùng (quyền hạn của nhà quản trị - admin).
- Giao diện quản lý trạm đo (quyền hạn của người dùng – người tạo trạm – user).
- Giao diện quản lý thiết bị đo (quyền hạn của người dùng sở hữu trạm của thiết bị đo này).
- Giao diện theo dõi giám sát (quyền hạn của người dùng sở hữu trạm).
- Giao diện báo cáo/thống kê (quyền hạn của người dùng sở hữu trạm).



Hình 2.16 Thiết kế thô giao diện theo dõi



Hình 2.17 Thiết kế thô giao diện thống kê báo cáo

localhost:8082/monitor

Theo dõi

Quản lý trạm đo

Báo cáo - Thống kê

Quản lý người dùng

Thêm mới

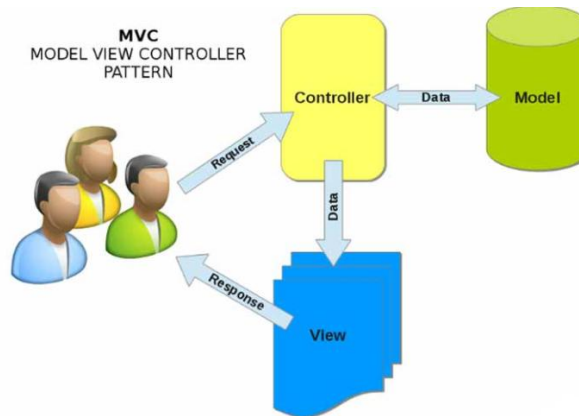
Danh sách người dùng

Hình 2.18 Thiết kế thô giao diện quản lý người dùng

- *Các giao diện quản lý trạm hay thiết bị đều có thiết kế tương tự như hình 2.17.
- *Kết quả của giao diện cuối cùng có thể được điều chỉnh cho thích hợp và thẩm mỹ hơn nhưng chức năng chính không thay đổi.

2.5.7 Lựa chọn công nghệ và ứng dụng

2.5.7.13. Giới thiệu mô hình MVC



Hình 2.19 Mô hình MVC

Mô hình MVC được định nghĩa đúng theo tên gọi của nó, gồm 3 thành phần M – Model, V – View và C – Controller. Trong đó:

- Controller đóng vai trò điều hướng các yêu cầu của người dùng và gọi đúng những phương thức xử lý chúng.
- Model chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất cơ sở dữ liệu,...
- View bao gồm toàn bộ phần xây dựng lên giao diện để người dùng quan sát vào thao tác trực tiếp.

Đây là mô hình mang tính chuyên nghiệp, nhờ được chia thành các thành phần độc lập nên dễ dàng phát triển và nâng cấp bảo trì. Tuy nhiên lại có hạn chế gây rườm rà phức tạp khi xây dựng một ứng dụng nhỏ.

Dựa trên mô hình này ứng dụng trong lập trình với ngôn ngữ Java, phần Controller chính là các lớp được gắn thẻ `@Controller`, đóng vai trò điều hướng; phần View là phần giao diện Frontend, được xây dựng với HTML/CSS và điều hướng bởi JavaScript; còn lại các thành phần khác như các lớp thực thể, các interface, service...đều thuộc phần Model. Tất cả sẽ được giới thiệu dưới đây.

2.5.7.14. Công nghệ sử dụng cho phần giao diện (Frontend).

a. HTML/CSS

Trong hệ thống này, sử dụng HTML – ngôn ngữ đánh dấu siêu văn bản - để dựng lên khung của giao diện và tận dụng CSS – các tập tin định kiểu theo tầng – để phục vụ mục đích làm đẹp giao diện.


```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:include="fragments/sidebar :: common-css-admin">
    <meta name="csrf-token" content="{{csrf_token()}}"/>
</head>
<body id="page-top">
<div id="wrapper">
    <div th:replace="fragments/sidebar :: side-bar"></div>
    <div id="content-wrapper" class="d-flex flex-column">
        <div id="content">
            <div th:replace="fragments/topbar :: top-bar"></div>
            <div class="container-fluid">
                <h1 class="h3 mb-2 text-gray-800">Báo cáo</h1>
            </div>
        </div>
        <div th:replace="fragments/footer :: footer"></div>
    </div>
</div>
<a class="scroll-to-top rounded" href="#page-top">
    <i class="fas fa-angle-up"></i>
</a>
<div th:replace="fragments/sidebar :: sidebar-js"></div>
<script th:src="@{~/js/common.js}"></script>
<script th:src="@{~/js/datatables/dataTables.bootstrap4.min.js}"></script>
<script type="text/javascript" src="../static/js/demo/datatables-demo.js"
    th:src="@{~/js/demo/datatables-demo.js}"></script>
</body>
</html>

```

Bên trên là ví dụ một đoạn HTML với các thẻ thông dụng như “<!DOCTYPE html>”, “<html>” – thẻ khai báo, “<head>” – thẻ header, “<body>” – thẻ nội dung cùng một số thẻ khác như “<div>”, “<a>”, “<i>”...

```
.container .btn {
    cursor: pointer;
    text-align: center;
    margin: 0 auto;
    width: 60px;
    color: #fff;
    background-color: #ff73b3;
    opacity: 1;
    transition: all 0.5s;
}
```

CSS có thể được khai báo như trên trong thẻ “<style>” của một file HTML, hoặc khai báo trực tiếp vào thuộc tính style của thẻ nào đó trong HTML nhằm mục đích làm đẹp cho giao diện. Như ví dụ trên đây, một đoạn CSS nhằm khai báo một số thuộc tính cho một nút (btn) như color – màu (giá trị max HEX), width – rộng,

b. JavaScript

JavaScript được khai báo trong thẻ “<script>” trong file HTML để phục vụ các mục đích điều hướng, thực thi hành động trên giao diện.

```
$('#content #btnDeleteStationRow').on('click', function (event) {
    event.preventDefault();
    let href = $(this).attr('href');
    $('#confirmDeleteStationModal #deleteStationRef').attr('href', href);
    $('#confirmDeleteStationModal').modal();
});
```

Phía trên là ví dụ một phương thức trong JavaScript, ở đây phương thức này thực thi hành động “click” của nút có id là “btnDeleteStationRow”, ở ví dụ này là để thực hiện hành động xóa station – trạm – khỏi danh sách.

Ngoài ra JavaScript còn phục vụ cho việc sử dụng thư viện đồ thị Highchart được dùng trong dự án này.

```
$(function () {
    $('#container2').highcharts({
        chart: {
            type: 'line'
        },
        title: {
```

```

        text: 'Nhiệt độ không khí 24h trước <br/>' + '11:00:00
01/01/2021'
    },
    xAxis: {
        categories: ['12:00', '13:00', '14:00', '15:00', '16:00', '17:00',
'18:00', '19:00', '20:00', '21:00', '22:00', '23:00', '00:00', '01:00', '02:00',
'03:00', '04:00', '05:00', '06:00', '07:00', '08:00', '09:00', '10:00', '11:00']
    },
    yAxis: {
        min: 0,
        title: {
            text: '°C'
        }
    },
    legend: {
        reversed: true
    },
    plotOptions: {
        series: {
            stacking: 'normal'
        }
    },
    series: [{
        name: 'Nhiệt độ không khí',
        data: [23, 23, 24, 23, 22, 20, 18, 17, 16, 15, 14, 13, 12, 11, 11,
11, 11, 13, 14, 16, 18, 20, 21, 21]
    }]
});

```

Thư viện này được cung cấp dưới dạng API json và được khai báo như trên, ta có thể tùy biến từng thành phần bằng việc sử dụng linh hoạt JavaScript. Với các thành phần đáng chú ý bao gồm:

- “chart: {type: ‘line’}”: đồ thị dạng đường, có nhiều loại đồ thị như column – cột, area – miền ...
- “title: {text: 'Nhiệt độ không khí...}”: là tiêu đề của đồ thị.
- “xAxis: {categories: ['12:00', '13:00', ...]}”: là các giá trị đóng trên trục hoành.

- “yAxis: {min: 0, title: {text: '°C'}}”: text - đơn vị đo hiển thị trên trục tung, min là giá trị thấp nhất hiển thị trên trục này.
- “series: [{name: 'Nhiệt độ không khí', data: [23...]}]”: series - sẽ bao gồm 1 danh sách đồ thị tương ứng, mỗi đối tượng trong series sẽ vẽ lên một đường đồ thị trên cùng một trục tọa độ này, trong đó name - tên của đường đồ thị và data – là dữ liệu đóng sang trục tung để xác định điểm trên trục tọa độ. Nếu có nhiều hơn một đường đồ thị, có thể chọn màu cho chúng hoặc thư viện sẽ tự sinh màu khác nhau theo các màu mặc định của nó.

Với mẫu khai báo như trên thu được kết quả như hình dưới đây.

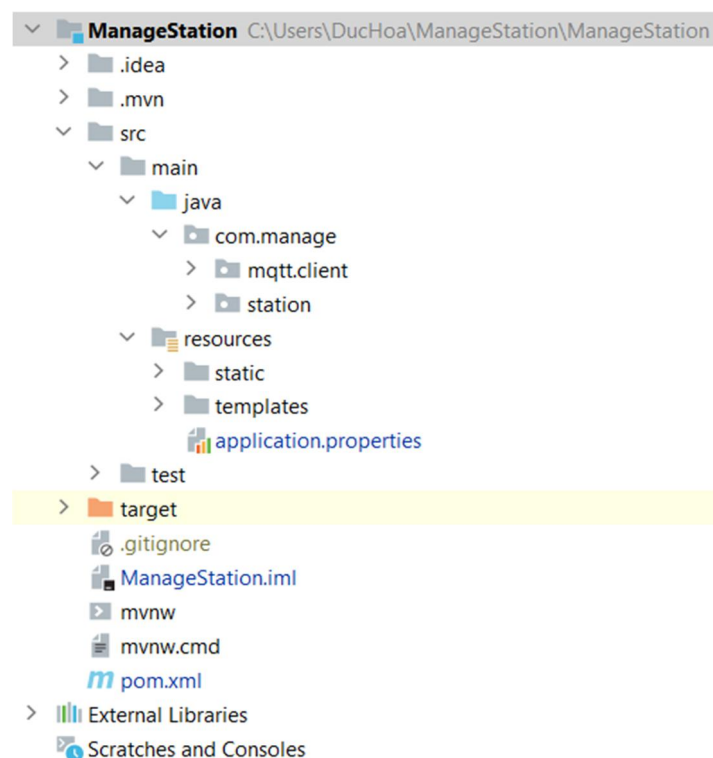


Hình 2.20 Ví dụ đồ thị sử dụng Highcharts

2.5.7.15. Công nghệ sử dụng cho phần hệ thống (Backend) – Java Spring

Java là một ngôn ngữ hướng đối tượng mạnh mẽ với rất nhiều những ưu điểm về tính bảo mật cũng như hiệu năng, trong hệ thống này sử dụng phiên bản Java 11 (JDK 11.0.9).

Phần này sử dụng cấu trúc của dự án Java Spring có cấu trúc thư mục và tệp như hình dưới đây.



Hình 2.21 Cấu trúc thư mục và tệp trong dự án Java Spring

Các thư viện được khai báo trong tệp “pom.xml”.

```
<?xml version="1.0" encoding="UTF-8"?>
<project                                xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.1</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.manage</groupId>
  <artifactId>ManageStation</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>ManageStation</name>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    ...
```

Trong thẻ “<dependency>” chính là thư viện sẽ được thêm vào tự động khi build dự án.

Dưới đây số khái niệm của Java và được ứng dụng trong hệ thống.

a. Lớp – class

```
@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class DeviceStationForm implements Serializable {
    private Long id;
    private Long stationId;
    private String name; //tên
    private String unit; //đơn vị đo
    private Double lowValue; //ngưỡng thấp
    private Double highValue; //ngưỡng cao
}
```

Trong ví dụ này các từ khóa “public”, “private” là phạm vi truy cập của lớp hoặc trường giá trị, “Long”, “String”...là kiểu dữ liệu, “DeviceStationForm” là tên lớp, “id”, “stationId”...là tên trường dữ liệu.

b. Phương thức – method

```
@Override
@Transactional(rollbackFor = Exception.class)
public void insertNewUserRole(UserRoleEntity userRoleEntity) {
    UserEntity userEntity = userRoleEntity.getUser();
    userEntity.setId(null);
    userEntity.setPassword(passwordEncoder.encode(userEntity.getPassword()));
    UserEntity userSaved = userRepository.save(userEntity);
    RoleEntity roleEntity = roleRepository.findById(
        userRoleEntity.getRole().getId()).get();
    ...
    userRoleEntity.setId(UserRoleId.builder().userId(userSaved.getId()).roleId(
        roleEntity.getId()).build());
    userRoleRepository.save(userRoleEntity);
}
```

Một phương thức được khai báo với phạm vi truy cập, kiểu dữ liệu trả về, tên phương thức, tham số truyền vào và nội dung các dòng lệnh thực thi.

Trong phương thức này, phạm vi truy cập là “public”, “void” là kiểu dữ liệu trả về, “insertNewUserRole” là tên phương thức, “UserRoleEntity userRoleEntity”

là tham số truyền vào, còn trong dấu ngoặc “{}” là phần nội dung của phương thức.

c. Interface

```
package com.manage.station.service;

...

import java.util.List;

public interface AdminService {

    List<RoleEntity> getAllRole();

    ...

}
```

Một interface sẽ sử dụng để khai báo các phương thức và không chứa nội dung thực thi của các phương thức này. Với “AdminService” là tên interface, phía trong dấu “{}” là các phương thức được khai báo gồm tên và kiểu trả về.

“package” có thể hiểu là đường dẫn thư mục quản lý code trong toàn “project”.

“import” là các lớp hoặc thư viện không cùng package với interface này.

d. Service

```
@Service

public class AdminServiceImpl implements AdminService {

    private final UserRepository userRepository;

    private final RoleRepository roleRepository;

    private final UserRoleRepository userRoleRepository;

    public AdminServiceImpl(UserRepository userRepository,

                           RoleRepository roleRepository,

                           UserRoleRepository userRoleRepository) {

        this.userRepository = userRepository;

        this.roleRepository = roleRepository;

        this.userRoleRepository = userRoleRepository;

    }

    @Override

    public List<RoleEntity> getAllRole() {

        return roleRepository.findAll();

    }...

}
```

Một lớp service sẽ kế thừa một interface (implements) tương ứng, và ghi đè (Override) các phương thức được khai báo trong interface bằng việc bổ sung

phần code thực thi của các phương thức này. Các hàm thực thi sẽ sử dụng các lớp có đuôi “Repository”, là các lớp kế thừa từ lớp JpaRepository, các lớp này sẽ gọi các hàm tương tác với cơ sở dữ liệu thông qua Hibernate (sẽ được mô tả ở phần sau).

e. Controller

```
@Controller
@RequestMapping("/admin")
public class AdminController {
    private final AdminService adminService;

    @Autowired
    public AdminController(AdminService adminService) {
        this.adminService = adminService;
    }

    @GetMapping("/manage-user")
    String manageUser(Model model) {
        return "admin/manage-user";
    }
    ...

    @RequestMapping(value = "/manage-users/new",
method = RequestMethod.POST)
    String addUser(UserRoleForm userRoleForm) {
        UserRoleEntity userRoleEntity = new UserRoleEntity();
        userRoleEntity.setUser(UserEntity.builder()
            .password(userRoleForm.getPassword())
            .email(userRoleForm.getEmail().trim())
            .address(userRoleForm.getAddress().trim())
            .fullName(userRoleForm.getFullName().trim())
            .phoneNumber(userRoleForm.getPhoneNumber().trim())
            .description(userRoleForm.getDescription().trim())
            .username(userRoleForm.getUsername().trim().toLowerCase())
            .build());
        userRoleEntity.setRole(RoleEntity.builder()
            .id(userRoleForm.getRoleId())
            .build());
        adminService.insertNewUserRole(userRoleEntity);
    }
}
```



```
        return "redirect:/admin/manage-users";
    }
    ...
```

Một controller sẽ sử dụng các phương thức của service và các file HTML cùng các thẻ đường dẫn để điều hướng trang web, phản hồi và thực thi hành động... Các đường dẫn này có thể được gọi bằng nhiều cách từ bên phía View, như submit, các phương thức onclick, hay sử dụng ajax để thực hiện.

Ví dụ trên hình, với đường dẫn “/admin/manage-user” sẽ trả về giao diện “manage-user” trong thư mục “admin”.

Các thẻ “@Controller” để khai báo controller, “@RequestMapping”, “@GetMapping” để khai báo đường dẫn có thể sử dụng tương ứng với phương thức bên dưới nó. Phần nội dung của các phương thức này sẽ thực thi các dòng lệnh trong đó nhằm trả về đúng với kết quả mong muốn.

f. Cấu hình kết nối MQTT

Để tạo một MQTT Client trên server java này, hệ thống sử dụng thư viện:

```
<dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.0</version>
</dependency>
```

Khởi tạo một MQTT Client và kết nối tới broker:

```
public static MqttClient client;
public static final String url = "tcp://127.0.0.1:1883";
public static final String user = "username";
public static final String pass = "pass";
public static final String topicSub = "value";
public static final String topicPub = "control";
...

String clientId = UUID.randomUUID().toString();
client = new MqttClient(url, clientId);
MqttConnectOptions connOpts = new MqttConnectOptions();
connOpts.setCleanSession(true);
connOpts.setKeepAliveInterval(180);
connOpts.setUsername(user);
connOpts.setPassword(pass.toCharArray());
```

```
client.connect(connOpts);
```

Khai báo các thông tin như đường dẫn (url) của MQTT Broker, tài khoản (user) và mật khẩu (pass), khởi tạo đối tượng MqttClient và khai báo như trên để kết nối tới MQTT Broker.

Xuất bản – publish – một message lên chủ đề - topic (ví dụ: tin nhắn gửi tín hiệu điều khiển):

```
MqttMessage message = new MqttMessage(content.getBytes());  
message.setQos(0);  
client.publish(topic, message);
```

Khởi tạo một đối tượng MqttMessage, biến “content” chứa nội dung của message cần gửi đi, cuối cùng là gọi hàm “publish” của MqttClient đã được kết nối (client) để xuất bản một tin nhắn lên “topic” được truyền vào phương thức này.

Theo dõi – subscribe – một chủ đề - topic (topic nhận kết quả đo):

```
static class MqttPostPropertyMessageListener  
implements IMqttMessageListener {  
    private final UserService userService;  
    MqttPostPropertyMessageListener(UserService userService) {  
        this.userService = userService;  
    }  
    @Override  
    public void messageArrived(String var1, MqttMessage var2) throws  
Exception {  
        if (var1.equals(topicSub)){  
            try {  
                JSONObject json = new JSONObject(var2.toString());  
                DeviceEntity deviceEntity = userService.  
getDeviceById(json.getLong("id"));  
                if (deviceEntity != null){  
                    ValueEntity valueEntity = new ValueEntity();  
                    valueEntity.setDevice(deviceEntity);  
                    Date createdAt = null;  
                    try {  
                        createdAt = new Date(json.getLong("createdAt"));  
                    } catch (Exception e){  
                        createdAt = new Date();  
                    }  
                }  
            }  
        }  
    }  
}
```



```
Select d FROM DeviceEntity d where d.station.id = :stationId order by  
d.name asc
```

Câu lệnh trên chứa các thành phần “*DeviceEntity d*” là tên bảng (ở đây đã được Hibernate kết nối trực tiếp với bảng “device”), “*d.station.id*” là điều kiện tìm kiếm, “*:stationId*” là giá trị truyền vào, còn “*select d*” là kết quả trả về mong muốn, cuối cùng là “*order by d.name*” là sắp xếp theo tên thiết bị “*asc*” là thuộc tính sắp xếp tăng dần.

Tóm lại, câu truy vấn trên hình mô tả việc tìm kiếm tất cả các thiết bị đo nào có mã trạm đo đúng bằng “*stationId*” truyền vào, kết quả trả về được sắp xếp theo tên thiết bị với thứ tự tăng dần.

Bên cạnh câu lệnh SELECT, SQLServer còn cung cấp các cú pháp câu lệnh khác như INSERT, UPDATE, DELETE, DROP, ALTER ... để phục vụ cho các chức năng tương đương thêm, sửa, xóa, ...

b. Giải pháp tương tác giữa Java với cơ sở dữ liệu – Hibernate

Để sử dụng được Hibernate, phải khai báo thư viện tương ứng với cơ sở dữ liệu sử dụng.

```
<dependency>  
    <groupId>com.microsoft.sqlserver</groupId>  
    <artifactId>mssql-jdbc</artifactId>  
    <scope>runtime</scope>  
</dependency>
```

Cấu hình khai báo cơ sở dữ liệu với các thông tin tương ứng dưới đây (tại đây phần này được khai báo trực tiếp trên máy local).

```
# khai báo cấu hình cơ sở dữ liệu  
spring.datasource.driver-class-  
name=com.microsoft.sqlserver.jdbc.SQLServerDriver  
spring.datasource.url=jdbc:sqlserver://127.0.0.1:1433;databaseName=Manage  
Station  
spring.datasource.username=sa  
spring.datasource.password=123456  
# =====  
# khai báo JPA / HIBERNATE  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.SQLServer2012Di  
alect
```

Tiếp theo là cấu hình kết nối các lớp đại diện cho bảng tương ứng dưới cơ sở dữ liệu như ví dụ dưới đây.

```
@Entity
@Table(name = "device")
//thiết bị đo
public class DeviceEntity implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;
    @ManyToOne(optional = false)
    @JoinColumn(name = "station_id", referencedColumnName = "id",
        nullable = false)
    private StationEntity station;
    @Column(name = "name", columnDefinition = "nvarchar(255)", nullable
        = false)
    ...
}
```

Một số thẻ thông dụng:

- @Entity: khai báo thực thể.
- @Table(name="user_"): khai báo liên kết với bảng user_ trong database.
- @Id: Khai báo mã định danh cho thực thể.
- @GenerateValue: khai báo trường tự sinh.
- @Colum(name="username", nullable = false, unique = true): khai báo liên kết với cột username trong bảng user_, với điều kiện không thể để null (nullable = false) và tính duy nhất (unique = true).
- @ ManyToOne: khai báo quan hệ một nhiều với một bảng khác.
- @JoinColumn: khai báo cột được dùng để khai báo quan hệ một nhiều.

Hibernate hỗ trợ một số cú pháp mặc định như: findById(ID) – tìm kiếm bằng mã serial của thực thể; save(E) – lưu một bản ghi mới; saveOrUpdate(E) – thêm mới hoặc cập nhật bản ghi; ... Trong đó E là viết tắt Entity chỉ thực thể cần tương tác, ID chỉ mã của thực thể đó được quy định bằng thẻ @Id trong phần khai báo liên kết. Ngoài ra còn sử dụng cách dưới đây để thực hiện các tùy biến truy vấn theo nhu cầu của nhà phát triển. Để gọi và thực thi được các phương thức này, cần khai báo interface kế thừa từ interface “JpaRepository<E,ID>” tương ứng với từng thực thể, trong đó “E” là thực thể tương ứng, “ID” là khóa chính định danh của thực thể này.

```
@Repository
```

```
public interface DeviceRepository extends JpaRepository<DeviceEntity,
Long> {
    @Query("Select d FROM DeviceEntity d where d.station.id = :stationId "
+ "order by d.name asc")
    List<DeviceEntity> findAllDeviceByStationId(@Param("stationId")
Long stationId);
}
```

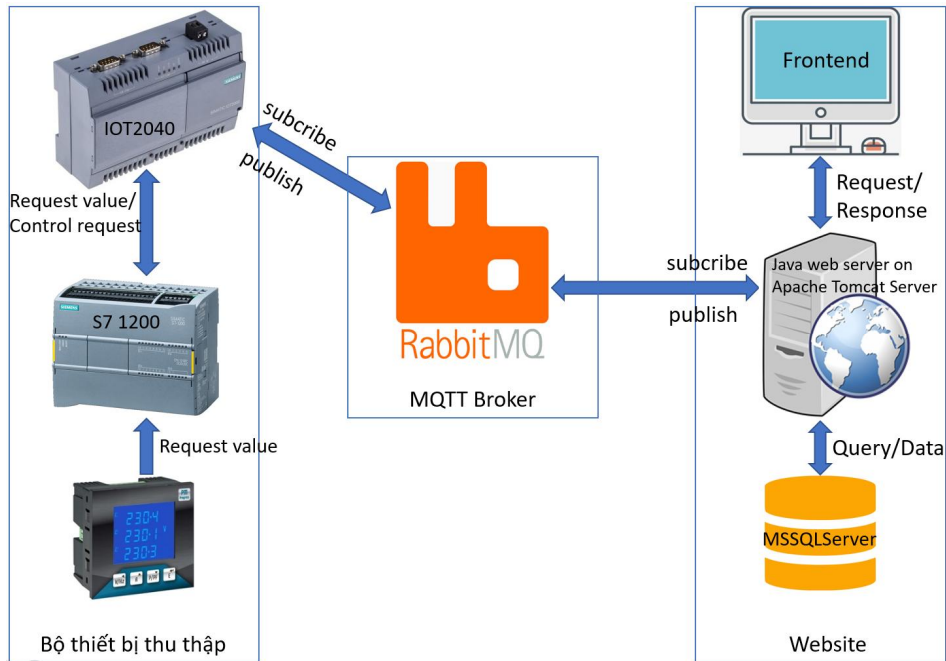
Trong interface này chứa hàm “*findAllDeviceByStationId*” (tên hàm truy vấn) với giá trị truyền vào là “*Long stationId*” (mã serial trạm đo kiểu dữ liệu long) và kiểu trả về là “*List<DeviceEntity>*” (một danh sách thiết bị đo trực thuộc trạm này). Thẻ “*@Query*” chứa câu truy vấn sẽ được gọi xuống database để truy xuất dữ liệu.

Từ đây, các lớp “Service” có thể gọi các hàm được khai báo tại đây cũng như các hàm mặc định của “JpaRepository” để thực hiện các chức năng cần thiết.

CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG

3.1 Mô hình triển khai hệ thống

Dựa trên thiết kế trên hình 2.1 cùng với công nghệ hiện có ứng dụng xây dựng hệ thống này theo sơ đồ trên hình 3.1 dưới đây.



Hình 3.1 Sơ đồ triển khai hệ thống

3.2 Triển khai chi tiết từng thành phần

3.2.1 Bộ thiết bị thu thập

Khi triển khai hệ thống, cần lắp đặt các thiết bị của bộ thiết bị thu thập để đảm bảo được chất lượng đường truyền theo đúng tiêu chuẩn công nghiệp. Tiếp theo, dựng một MQTT Client trên IOT2040 để tiếp nhận đóng gói dữ liệu và xử lý các gọi tin theo hướng dẫn của nhà cung cấp.

Do điều kiện không cho phép, phần này sẽ được demo với một MQTT Client sẵn có là MQTT.fx, hoặc sẽ sử dụng một MQTT Client khác viết bằng ngôn ngữ Java với chỉ màn hình console để thực hiện chức năng này (MQTT Client viết bằng ngôn ngữ Java hoàn toàn có khả năng tương thích và hoạt động bình thường trên máy tính IOT2040), phần dữ liệu đo sẽ được sinh ngẫu nhiên hoặc nhập trực tiếp trên ứng dụng này.

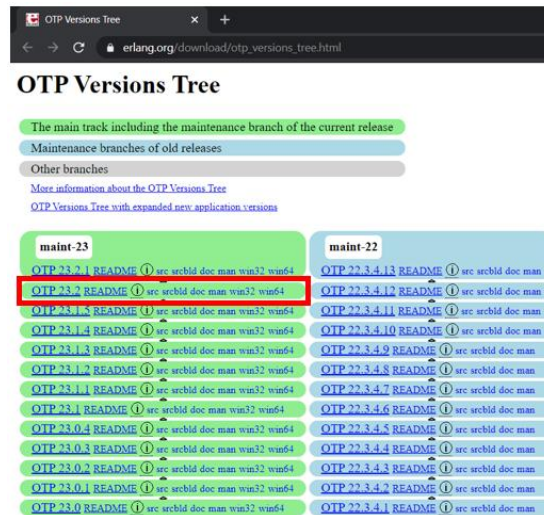
3.2.2 Broker RabbitMQ

3.2.2.1. Cài đặt và cấu hình

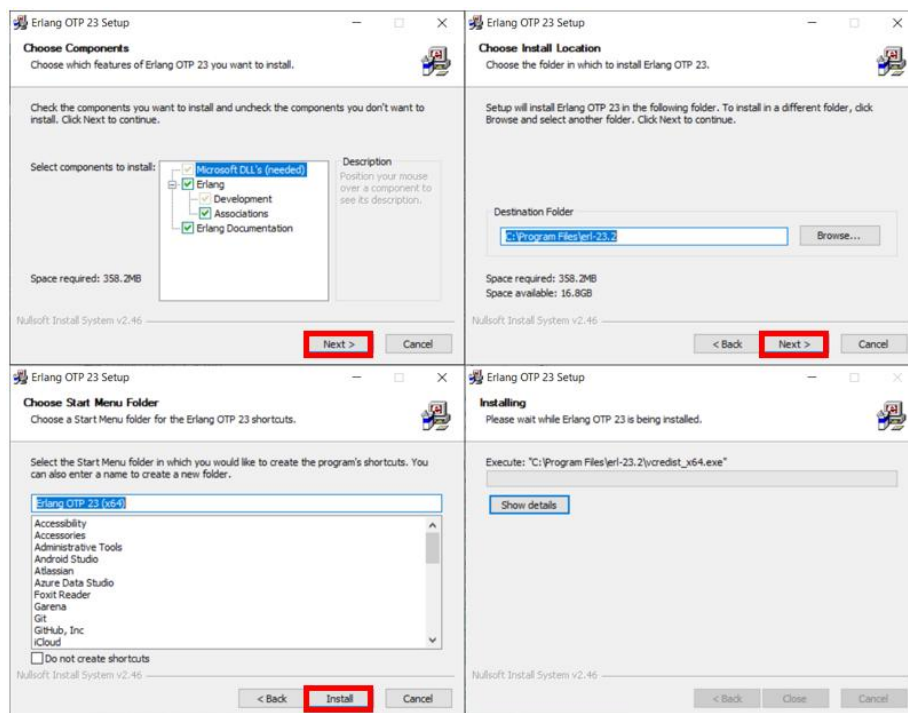
Chọn một máy tính cá nhân hoặc server bất kỳ nằm cùng mạng với server web và thiết bị IOT2040 để tiến hành cài đặt Broker RabbitMQ.

Các bước cài đặt và cấu hình Broker RabbitMQ được thực hiện theo hướng dẫn trên website của RabbitMQ (<https://www.rabbitmq.com/>). Dưới đây là hướng dẫn cài đặt cho hệ điều hành window.

Để cài đặt được RabbitMQ, trước hết phải cài đặt Erlang cho máy tính này. Truy cập https://erlang.org/download/otp_versions_tree.html chọn phiên bản tương ứng với hệ điều hành window 64bit và làm theo các bước dưới đây để cài đặt Erlang.



Hình 3.2 Lựa chọn phiên bản Erlang trên website



Hình 3.3 Các bước cài đặt Erlang

Sau khi Erlang được cài đặt xong, ta chuyển sang các bước để cài đặt RabbitMQ dưới đây.

- Truy cập <https://www.rabbitmq.com/install-windows.html>.

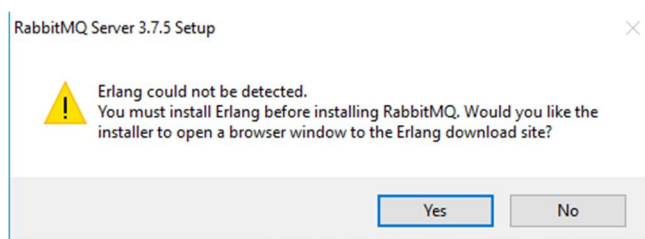
- Kéo xuống phần “Direct Downloads” và chọn như hình dưới đây để tải bộ cài về.

Direct Downloads

Description	Download	Signature
Installer for Windows systems (from GitHub , recommended)	rabbitmq-server-3.8.9.exe	Signature
Alternative download location (from Bintray)	rabbitmq-server-3.8.9.exe	

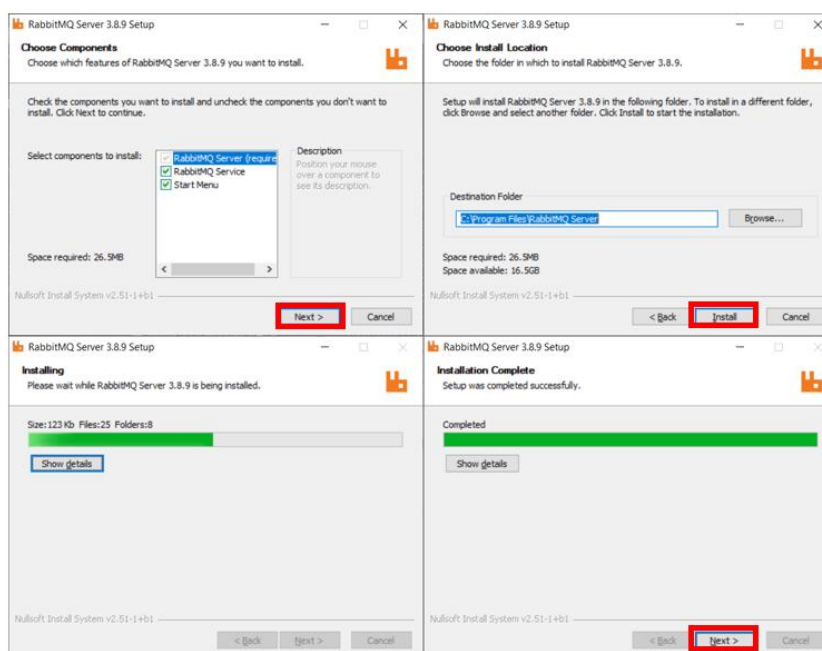
Hình 3.4 Vị trí download bộ cài RabbitMQ

- Sau khi tải về xong, chạy bộ cài dưới quyền administrator để cài đặt. (Nếu chưa cài đặt Erlang thành công, sẽ có một thông báo yêu cầu cài đặt Erlang khi chạy bộ cài này).



Hình 3.5 Thông báo lỗi khi chưa cài đặt Erlang

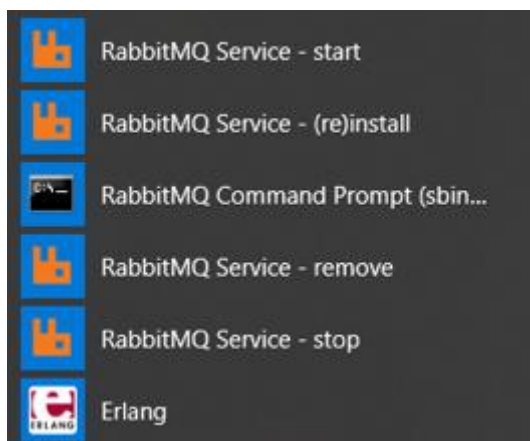
- Dưới đây là các bước cài đặt RabbitMQ.



Hình 3.6 Các bước cài đặt RabbitMQ

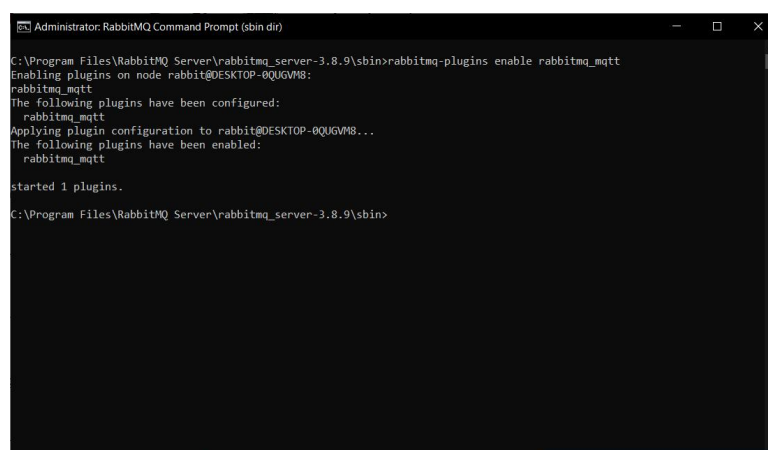
Trong quá trình cài đặt hệ thống sẽ yêu cầu các quyền cho cả Erlang và RabbitMQ, chọn “Allow Access” với quyền của Erlang và một số quyền khác để Broker có thể hoạt động được.

Sau khi cài đặt thành công, trong phần menu window có thêm các thành phần sau:



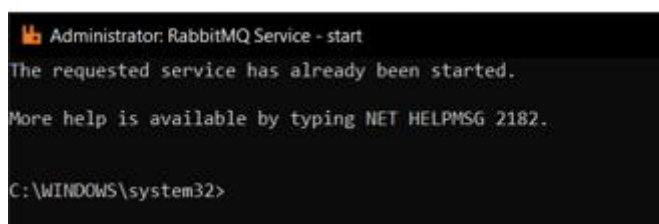
Hình 3.7 Kết quả cài đặt RabbitMQ

Tiếp theo ta mở “RabbitMQ Command Prompt (sbin...” (trên hình), chạy dòng lệnh “rabbitmq-plugins.bat enable rabbitmq_mqtt” để bật plugin Mqtt cho Broker.



Hình 3.8 Enable plugin mqtt

Sau đó ta mở “RabbitMQ service – start” (hình 3.8).



Hình 3.9 RabbitMQ service – start

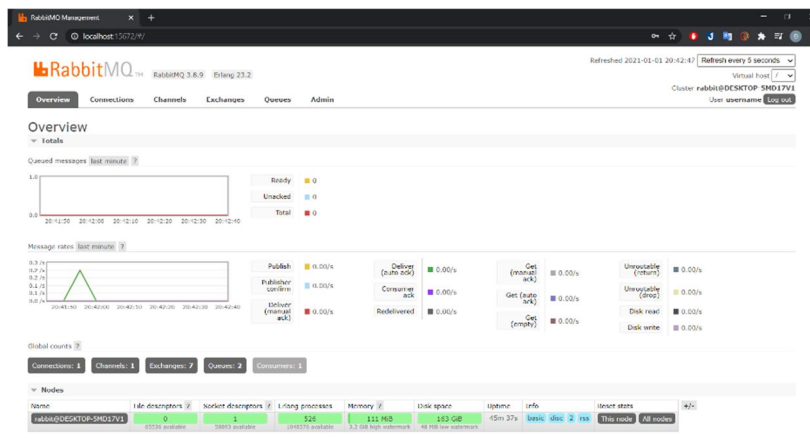
3.2.2.2. Một số thao tác trên phần mềm quản trị RabbitMQ Broker.

Sau khi hoàn thành bước trên, mở trình duyệt truy cập “localhost:15672” – cổng mặc định của RabbitMQ để truy cập vào trang quản lý của Broker này. Khởi đầu sẽ đăng nhập sử dụng tài khoản và mật khẩu đều là “guest”.



Hình 3.10 Giao diện đăng nhập RabbitMQ

Sau khi đăng nhập thành công ta thu được màn hình sau:



Hình 3.11 Giao diện quản lý Broker RabbitMQ

Tại đây ta có thể tiến hành thêm tài khoản và phân quyền ở tab Admin (như trên hình là kết quả đăng nhập tài khoản “username”) hoặc sử dụng các chức năng khác (tham khảo thêm tại <https://www.rabbitmq.com/>).

Để kiểm tra xem mqtt đã được mở trên Broker này hay chưa, kéo xuống phần “Ports and contexts” như hình 3.12 dưới đây.

▼ Ports and contexts

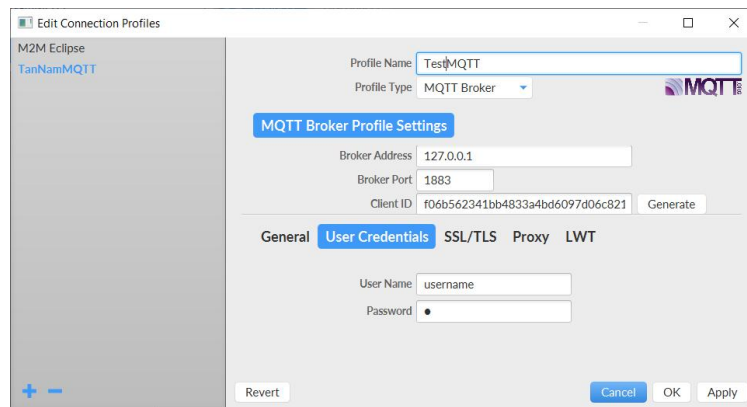
Listening ports

Protocol	Bound to	Port
amqp	::	5672
clustering	::	25672
http	::	15672
mqtt	::	1883
mqtt/ssl	::	8883

Hình 3.12 Dịch vụ MQTT trên Broker RabbitMQ

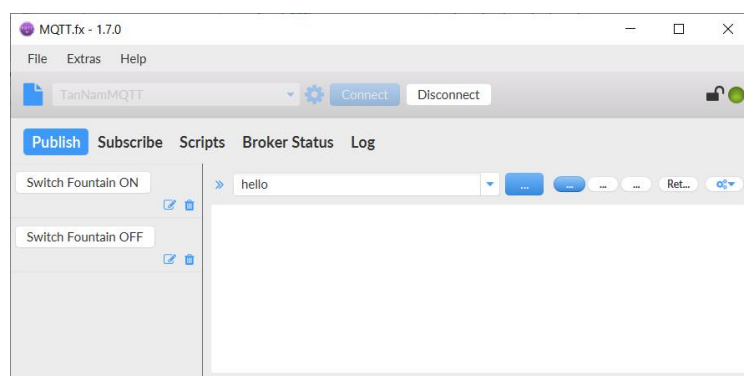
Tiến hành sử dụng một MQTT Client để kiểm tra hoạt động của cổng này (ở đây lấy ví dụ với phần mềm MQTT.fx).

Mở MQTT.fx và thiết lập như hình dưới.



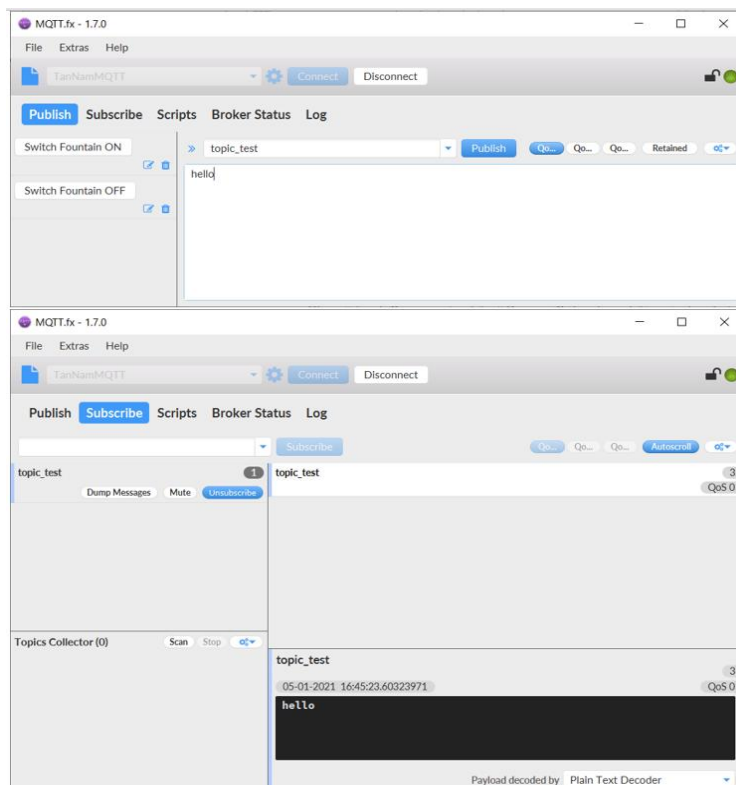
Hình 3.13 Nhập thông tin kết nối

Chọn “Apply”, ra ngoài nhấn “Connect” thu được kết quả như dưới đây là thành công.



Hình 3.14 Kết quả kết nối thành công MQTT Broker

Kiểm thử publish và subscribe như hình dưới đây là thành công.



Hình 3.15 Kiểm thử chức năng publish và subscribe

Như trên hình, Client này vừa publish lên topic “topic_test” vừa subscribe topic này nên đã thu được kết quả như trên.

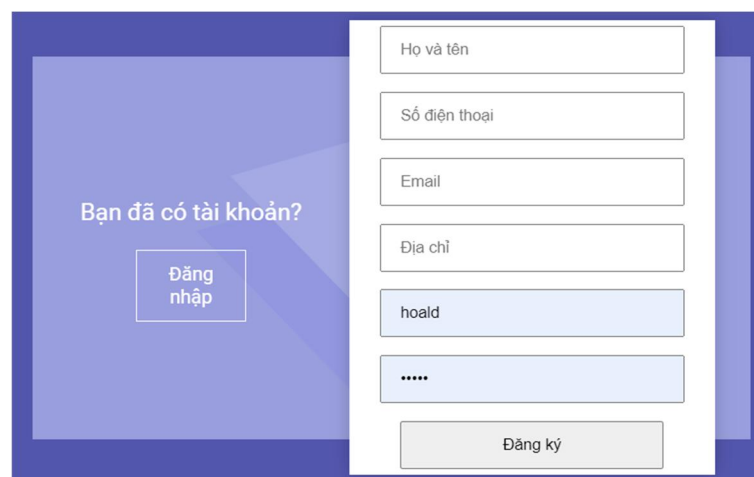
Vậy là ta đã hoàn thành cài đặt và kiểm thử chức năng MQTT trên Broker RabbitMQ.

3.2.3 Phần mềm website quản lý

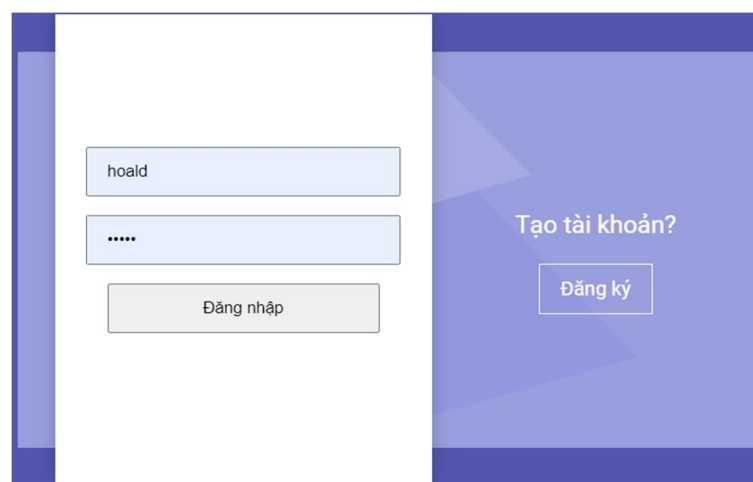
Để triển khai hệ thống website quản lý, cần có tối thiểu một máy chủ (tối ưu nhất là nên tách riêng từng thành phần Broker – máy chủ - cơ sở dữ liệu), bên cạnh đó là các máy khách chung mạng để truy cập theo dõi.

Sau khi cấu hình các thông tin liên quan tới cơ sở dữ liệu và MQTT Broker, phần mềm sẽ được đóng gói và chạy trên máy chủ dưới dạng dịch vụ. Sau đó tiến hành tạo ít nhất 1 tài khoản admin trực tiếp trên cơ sở dữ liệu bằng các thêm bản ghi vào bảng “user_” và bảng “role_user”. Cuối cùng các máy khách có thể tiết hành truy cập dưới đường dẫn “<địa chỉ máy chủ>:<cổng>” thực hiện các chức năng tương ứng.

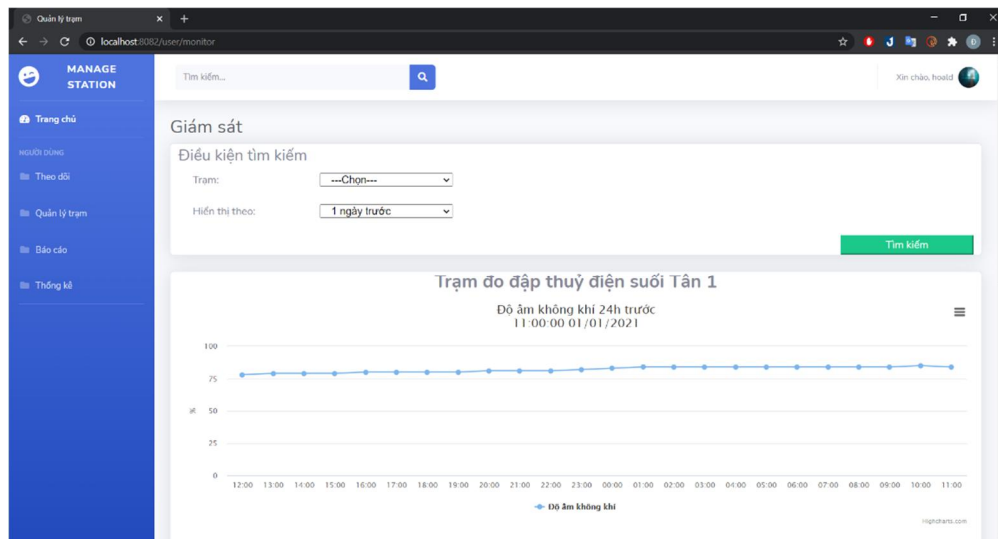
Một số giao diện và chức năng tương ứng.



Hình 3.16 Giao diện đăng ký



Hình 3.17 Giao diện đăng nhập



Hình 3.18 Giao diện theo dõi giám sát

The screenshot displays the 'Danh sách trạm quản lý' (Station Management List) page. It features a sidebar with 'MANAGE STATION' and navigation links. The main area has a search bar and a 'Thêm mới' (Add new) button. Below is a table with 5 entries. The table has columns: STT, Tên trạm, Địa chỉ, Mô tả, and Hành động. Each entry has buttons for 'Sửa' (Edit), 'Xóa' (Delete), and 'QL thiết bị' (Manage device). The table shows the following data:

STT	Tên trạm	Địa chỉ	Mô tả	Hành động
1	Điểm theo dõi đập 1	Mộc Châu	Theo dõi các thông số đập thủy điện	Sửa Xóa QL thiết bị
2	Trạm đo đập thủy điện suối Tân 2	Mộc Châu	theo dõi đập 2	Sửa Xóa QL thiết bị
3	Trạm theo dõi biến áp hoà lưới 1	Sơn La	theo dõi hoà lưới	Sửa Xóa QL thiết bị
4	Trạm theo dõi hoà lưới 3	Mộc Châu	theo dõi hoà lưới	Sửa Xóa QL thiết bị
5	Trạm theo dõi tuabin phát điện 1	Mộc Châu	theo dõi tuabin phát điện	Sửa Xóa QL thiết bị

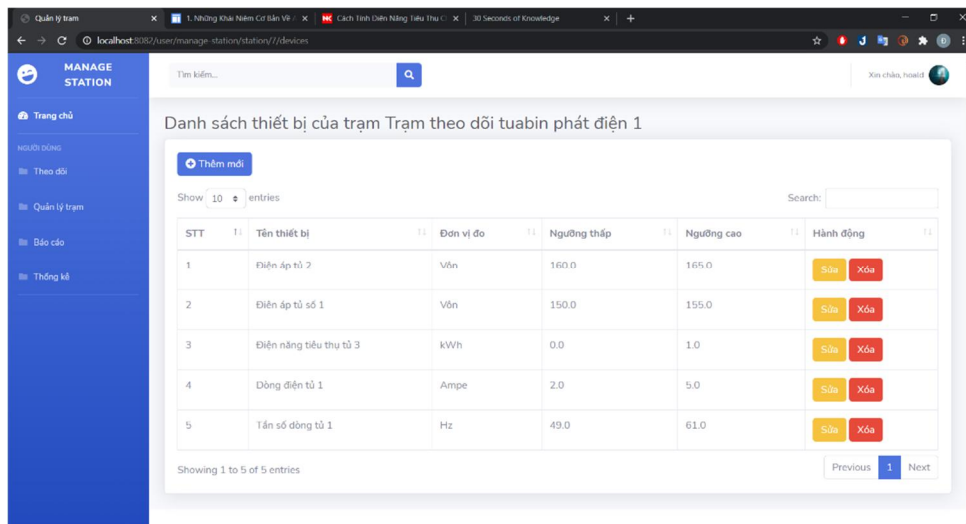
Showing 1 to 5 of 5 entries. Navigation: Previous 1 Next.

Hình 3.19 Giao diện quản lý trữ trạm đo

The screenshot displays the 'Thêm mới trạm quản lý' (Add new station management) form. It has a title bar with a close button. The form includes the following fields and buttons:

- Tên trạm:** Input field containing 'Trạm theo dõi hoà lưới 3'.
- Địa chỉ trạm:** Input field containing 'Mộc Châu'.
- Mô tả:** Input field containing 'theo dõi hoà lưới'.
- Hủy bỏ:** Button to cancel the operation.
- Thêm mới:** Button to add the new station.

Hình 3.20 Giao diện thêm mới trạm đo



Hình 3.21 Giao diện quản lý thiết bị đo

Thêm mới thiết bị cho trạm Trạm theo dõi tuabin
phát điện 1

Tên thiết bị:

Đơn vị đo:

Ngưỡng thấp:

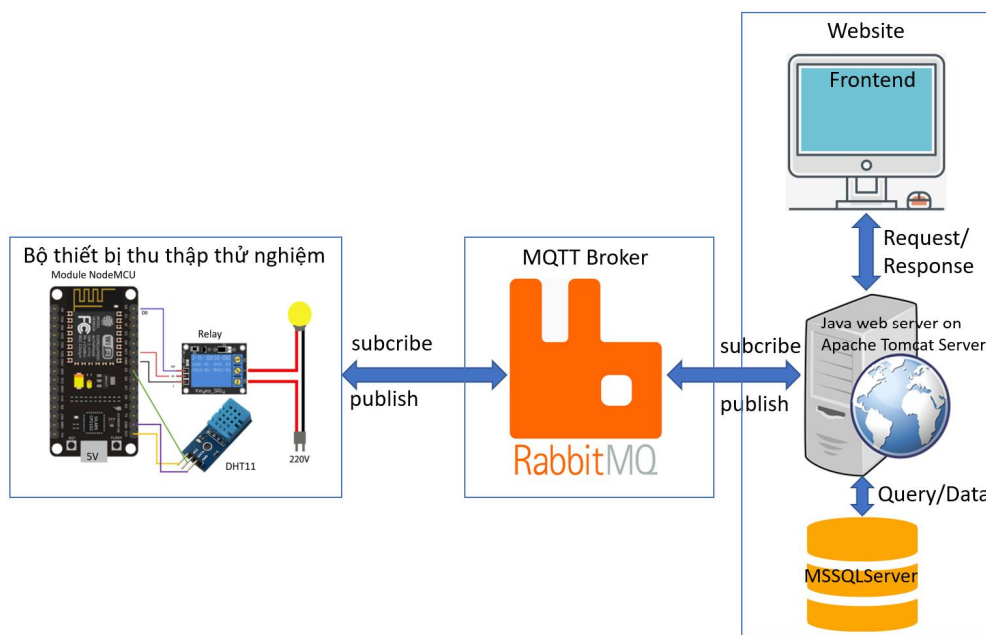
Ngưỡng cao:

Hủy bỏ

Thêm mới

Hình 3.22 Giao diện thêm mới thiết bị đo

Các tệp được xuất bản từ phần mềm này có dạng đồ thị như hình 3.18 hoặc bảng thống kê chi tiết, phục vụ bổ sung phần tương ứng vào các mẫu báo cáo thống kê. Thử nghiệm phần này, hệ thống sẽ được chạy hoàn toàn trên máy tính cá nhân được thực hiện theo mô hình dưới đây. Mô hình bộ thiết bị thử nghiệm sẽ đóng vai trò như một máy IOT2040 tại một trạm đo thực tế.



Hình 3.23 Sơ đồ thử nghiệm trên máy tính cá nhân

3.3 Kết quả đạt được

Nhờ quá trình tìm hiểu và phát triển đề tài này, em đã thu được rất nhiều các kỹ năng mới cho bản thân cũng như củng cố trau dồi những kỹ năng đã có từ trước như:

- Kỹ năng mềm: kỹ năng giao tiếp, kỹ năng làm các tài liệu (slide, báo cáo) một cách chuyên nghiệp.
- Kỹ năng về kỹ thuật: cài đặt, sử dụng và quản lý RabbitMQ Broker, các kỹ năng đối với HTML, CSS, JavaScript, Java Spring MVC, các framework như Servlet, Hibernate, sử dụng các phần mềm như MQTT.fx, IDE IntelliJ, quản lý source code với Git...

Ứng dụng các công nghệ trên đây, hệ thống đã thử chạy demo toàn bộ trên máy tính cá nhân của mình và thu được các kết quả như sau:

- Website đã nhận và hiển thị được các kết quả đo dưới dạng đồ thị, có thể sinh đồ thị này để hỗ trợ quá trình làm báo cáo dưới nhiều định dạng như pdf, png, jpeg, svg.
- Các Client phía “Thiết bị thu thập” đã nhận thành công các gói tin điều khiển.
- Toàn bộ hệ thống hoạt động tương đối ổn định trong môi trường thử nghiệm.

CHƯƠNG 4. KẾT LUẬN

4.1 Kết luận

Qua 3 chương trên đây em đã trình bày về hệ thống thu thập, giám sát và sinh báo cáo cho một hệ thống các trạm đo. Qua đó nêu ra những kỹ thuật và công nghệ cùng với kiến trúc đã áp dụng trong quá trình xây dựng hệ thống này. Về kết quả, em đã hoàn thành 100% các mục tiêu đề ra.

Trong quá trình xây dựng hệ thống này, em đã thu được rất nhiều kinh nghiệm và kiến thức từ những người đi trước. Những điều này là vô giá và sẽ đi cùng em trên chặng tiếp theo sau khi đường rời khỏi ghế nhà trường.

4.2 Ưu điểm

Hệ thống này gồm các ưu điểm:

- Giao diện trực quan, dễ sử dụng.
- Đáp ứng các tính năng thông thường cho công việc hằng ngày của cán bộ theo dõi – giám sát.
- Thuận tiện và linh hoạt về nơi làm việc cũng như trang thiết bị cho cán bộ nhân viên.

4.3 Nhược điểm

Tuy nhiên, hệ thống của em vẫn còn những vấn đề sau đây:

- Chưa cung cấp giao diện tương thích với mọi loại thiết bị.
- Chưa tối ưu hóa được hết truy vấn cơ sở dữ liệu.
- Còn phức tạp trong việc cấu hình thiết bị.

4.4 Hướng phát triển và bổ sung tính năng

Hệ thống em xây dựng ở đây chỉ là một modul nhỏ trong toàn bộ bài toán và còn chưa hoàn toàn thích hợp để có thể đưa công nghệ này vào thực tiễn.

Đối với định hướng phát triển nếu có điều kiện, em sẽ phát triển hệ thống này theo hướng cung cấp dịch vụ để duy trì lâu dài. Bên cạnh việc cải thiện giao diện, cần bổ sung thêm nhiều chức năng cho mảng giao dịch, bổ sung thêm các chức năng quản lý thời hạn, cấp quyền hoạt động, mở kết nối trạm, bổ sung kết nối thiết bị... Toàn bộ đều đi theo hướng dịch vụ và một đơn vị sẽ đứng ra cung cấp dịch vụ này, vừa triển khai lắp đặt phần cứng, vừa cung cấp thêm các lựa chọn cho các dịch vụ phần mềm. Hi vọng hướng phát triển này sẽ có cơ hội được bước ra ánh sáng trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] L. N. H. Nam, "SIMATIC IOT2040," Siemens Việt Nam, April 2020. [Online]. Available: https://assets.new.siemens.com/siemens/assets/api/uuid:5a68b416-9220-4bf8-b338-034d1e298d4f/SIMATIC-IOT2040-SiemensOnlineWorkshop-20200414_original.pdf.
- [2] PCLTECH, "PCLTECH," TỰ ĐỘNG HÓA CÔNG NGHIỆP PLCTECH, 2014. [Online]. Available: <https://drive.google.com/drive/folders/1LC-s-O030Cd-yTUATZRhNbl5w2aRHgI5>. [Accessed 31 12 2020].
- [3] "Spring," Spring, 2020. [Online]. Available: <https://spring.io/projects/spring-framework>.
- [4] "RabbitMQ," RabbitMQ, 2007. [Online]. Available: <https://www.rabbitmq.com/management.html>.
- [5] "Monamedia," Monamedia - Công ty thiết kế website cao cấp, [Online]. Available: <https://monamedia.co/mvc-la-gi-ung-dung-cua-mo-hinh-mvc-trong-lap-trinh/>.