

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

HOÀNG MINH TRÍ

ĐƯƠNG QUỐC GIA

LUẬN VĂN TỐT NGHIỆP

**ỨNG DỤNG CÔNG NGHỆ WEB-BASED VÀO GIÁM SÁT VÀ
THU THẬP DỮ LIỆU CHO NHÀ MÁY VỪA VÀ NHỎ**

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

TP. HỒ CHÍ MINH, 2022

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

HOÀNG MINH TRÍ - 1811297

ĐƯƠNG QUỐC GIA - 1811992

LUẬN VĂN TỐT NGHIỆP

ỨNG DỤNG CÔNG NGHỆ WEB-BASED VÀO GIÁM SÁT VÀ
THU THẬP DỮ LIỆU CHO NHÀ MÁY VÙA VÀ NHỎ
SUPERVISION AND DATA ACQUISITION FOR SMALL AND MEDIUM
FACTORY: A WEB-BASED APPROACH

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN

TS. TRẦN NGỌC HUY

TP. HỒ CHÍ MINH, 2022

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN**

Tên luận văn:

ỨNG DỤNG CÔNG NGHỆ WEB-BASED VÀO GIÁM SÁT VÀ THU THẬP DỮ LIỆU CHO NHÀ MÁY VỪA VÀ NHỎ / SUPERVISION AND DATA ACQUISITION FOR SMALL AND MEDIUM FACTORY: A WEB-BASED APPROACH

Nhóm Sinh viên thực hiện:

Hoàng Minh Trí

1811297 TS. Trần Ngọc Huy

Dương Quốc Gia

1811992 TS. Trần Ngọc Huy

Cán bộ hướng dẫn:

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang

Số chương

Số bảng số liệu

Số hình vẽ

Số tài liệu tham khảo

Sản phẩm

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Hoàng Minh Trí:...../**10**

Dương Quốc Gia:...../**10**

Cán bộ hướng dẫn

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ PHẢN BIỆN**

Tên luận văn:

ỨNG DỤNG CÔNG NGHỆ WEB-BASED VÀO GIÁM SÁT VÀ THU THẬP DỮ LIỆU CHO NHÀ MÁY VỪA VÀ NHỎ / SUPERVISION AND DATA ACQUISITION FOR SMALL AND MEDIUM FACTORY: A WEB-BASED APPROACH

Nhóm Sinh viên thực hiện:

Hoàng Minh Trí 1811297

Dương Quốc Gia 1811992

Cán bộ phản biện:

Đánh giá Luận văn

5. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

6. Về nội dung luận văn:

7. Về tính ứng dụng:

8. Về thái độ làm việc của sinh viên:

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Hoàng Minh Trí:...../**10**

Dương Quốc Gia:...../**10**

Người nhận xét

ĐỀ CƯƠNG CHI TIẾT

TÊN LUẬN VĂN: ỨNG DỤNG CÔNG NGHỆ WEB-BASED VÀO GIÁM SÁT VÀ THU THẬP DỮ LIỆU CHO NHÀ MÁY VỪA VÀ NHỎ

Cán bộ hướng dẫn: TS. Trần Ngọc Huy

Thời gian thực hiện: Từ ngày 01/02/2022 đến ngày 31/05/2022

Sinh viên thực hiện:

Hoàng Minh Trí - 1811297

Dương Quốc Gia - 1811992

Nội dung đề tài:

Mục tiêu: Thiết kế được cơ sở dữ liệu và công nghệ đọc dữ liệu liên tục sử dụng MQTT và web socket SignalR. Ứng dụng không những trực quan hóa các dữ liệu từ cơ sở dữ liệu mà còn có thể cảnh báo khi có hệ thống quá tải hoặc bất thường tại nhà máy.

Phạm vi: Ứng dụng tại các nhà máy sản xuất của những công ty quy mô nhỏ, cụ thể là ứng dụng thực tế vào nhà máy của một công ty sản xuất nhựa.

Đối tượng: Phần mềm chạy trên bất cứ nền tảng duyệt web nào có kết nối Internet và một máy chủ có bộ nhớ đủ lớn.

Phương pháp thực hiện: Khảo sát môi trường thực tế, đề ra giải pháp phần mềm hỗ trợ các trưởng ca, nhân viên giám sát ở nhà máy của công ty. Lựa chọn mô hình, công nghệ phần mềm để xây dựng hệ thống giám sát nhà máy.

Kết quả mong đợi: Hệ thống có khả năng vận hành ổn định, đọc được liên tục các

thông số vận hành của các máy với sai số cho phép là 500ms, phần mềm hỗ trợ tính toán các chỉ số sản xuất, báo cáo cần thiết cũng như đưa ra cảnh báo kịp cho từng khu vực sản xuất của nhà máy.

Kế hoạch thực hiện:

Thời gian	Nội dung thực hiện	
	Hoàng Minh Trí	Dương Quốc Gia
01/02 – 15/02/2022	Khảo sát các cụm máy đang triển khai hoặc sẽ xây dựng trong tháng 05, tìm hiểu giao thức truyền nhận dữ liệu hiện tại.	
16/02 – 28/02/2022	Lựa chọn công nghệ nền tảng cho ứng dụng web.	Lựa chọn công nghệ nền tảng để xây dựng hệ thống thu thập dữ liệu và các node thu thập dữ liệu.
01/03 – 15/03/2022	Lên phương án cho phần cứng: IPC, màn hình, chuột và phần mềm điều khiển chuột từ xa. Thiết kế giao diện bằng Figma: Thiết kế các ký tự máy đặc thù cho từng khu vực và thiết	Tìm hiểu về giao thức MQTT truyền thống và SocketTCP. Tìm hiểu về tiêu chuẩn truyền thông MQTT Sparkplug. Tìm hiểu về cơ sở dữ liệu chuỗi thời gian InfluxDb. Tìm kiếm và học cách sử dụng các thư viện liên quan để có thể áp dụng các kiến thức tìm hiểu

	kết các trang cần có của web.	được vào hệ thống.
16/03 – 26/03/2022	Chỉnh sửa giao diện của ứng dụng web theo yêu cầu của các anh chị trưởng bộ phận tại công ty.	Lên kiến trúc thiết kế cho hệ thống thu thập và lưu trữ dữ liệu. Cấu trúc dữ liệu quản lý tag. Và thiết kế node thu thập dữ liệu.
27/03 – 05/04/2022	Phân tích, thiết kế cấu trúc dữ liệu dựa trên quy trình và giao diện đã thống nhất với công ty.	Lập trình và xử lý đầu vào, đầu ra cho hệ thống thu thập và lưu trữ dữ liệu. Tích hợp InfluxDb và MQTT Sparkplug vào hệ thống.
05/04 – 30/04/2022	Lập trình và xử lý giao diện, tính toán các chỉ số sản xuất, xây dựng file báo cáo theo ngày và tháng từ các dữ liệu thô, đưa ra cảnh báo nếu có bất thường xảy ra tại nhà máy	Lập trình và xây dựng các node thu thập dữ liệu sử dụng MQTT, MQTT Sparkplug, SocketTCP để thu thập dữ liệu từ các module hiện tại của nhà máy.
01/05 – 10/05/2022	Cải thiện trải nghiệm người dùng và hiệu năng hoạt động của hệ thống.	
11/05 – 20/05/2022	Tinh chỉnh các tính năng theo yêu cầu của các trưởng bộ phận trong công ty. Kiểm tra hiệu suất sản phẩm hiện tại. Quay và chỉnh sửa video demo về quy trình vận hành hệ thống và video sử dụng phần mềm. Viết báo cáo luận văn.	

Xác nhận của Cán bộ hướng dẫn

TS. Trần Ngọc Huy

TP. HCM, ngàythángnăm.....

Sinh viên

Hoàng Minh Trí

Dương Quốc Gia

DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.
5. – Ủy viên.

MỤC LỤC

LỜI CẢM ƠN	1
LỜI MỞ ĐẦU	2
TÓM TẮT LUẬN VĂN	3
Chương 1. GIỚI THIỆU ĐỀ TÀI	4
1.1. Lý do chọn đề tài.....	4
1.2. Nhiệm vụ đề tài	4
1.3. Phương pháp thực hiện.....	5
Chương 2. TỔNG QUAN CÁC KHU VỰC HIỆN TẠI TẠI NHÀ MÁY	6
2.1. Thực trạng nhà máy	6
2.1.1. Khu vực máy ép	7
2.1.2. Khu vực đóng gói	8
2.1.3. Khu vực lắp ráp.....	8
2.1.4. Kho vận.....	10
2.1.5. Phòng kiểm tra chất lượng sản phẩm.....	11
2.1.6. Kế hoạch sản xuất và lịch trình sản xuất.....	12
2.2. Phương án đề xuất.....	13
2.2.1. Xây dựng node cho các khu vực	14
2.2.2. Xây dựng chức năng chính của phần mềm web.....	15
Chương 3. GIỚI THIỆU CÁC THIẾT BỊ TRONG HỆ THỐNG	17
3.1. Cấu trúc hệ thống	17
3.2. Xây dựng phương án.....	17

3.2.1.	Các window service chạy trên máy tính quản lý của mỗi khu vực	17
3.2.2.	Server – Máy chủ.....	18
Chương 4. TỔNG QUAN HỆ THỐNG THU THẬP VÀ CẤU TRÚC DỮ LIỆU.....		19
4.1.	Các công nghệ, ngôn ngữ và phần mềm sử dụng	19
4.1.1.	Ngôn ngữ lập trình	19
4.1.2.	Cơ sở dữ liệu chuỗi thời gian InfluxDB	19
4.1.3.	Cơ sở dữ liệu SQL Server 2019 và SSMS	21
4.1.4.	Giao thức MQTT	21
4.1.5.	Giao thức OPC UA	24
4.1.6.	Giao thức SocketTCP	24
4.1.7.	Tiêu chuẩn truyền thông MQTT Sparkplug	26
4.1.8.	Truyền dữ liệu bằng SignalR	31
4.2.	Tổng quan hệ thống	32
4.3.	Xây dựng các node thu thập dữ liệu	33
4.3.1.	Thu thập dữ liệu máy ép.....	34
4.3.2.	Thu thập cụm dữ liệu lắp ráp	39
4.3.3.	Thu thập dữ liệu cụm đóng gói.....	42
4.4.	Thiết kế hệ thống lưu trữ dữ liệu.....	45
4.4.1.	Thiết kế cấu trúc dữ liệu quản lý tag.....	45
4.4.2.	Xây dựng chương trình thu thập và lưu trữ dữ liệu.....	47
Chương 5. PHẦN MỀM ỨNG DỤNG WEB – WEB APPLICATION		50
5.1.	Các công nghệ, ngôn ngữ và phần mềm sử dụng	50
5.1.1.	Công nghệ.....	50

5.1.2.	Thư viện ReactJS	50
5.1.3.	Ngôn ngữ lập trình	51
5.1.4.	Phần mềm sử dụng	52
5.2.	Giao diện thiết kế và giải thuật lập trình	54
5.2.1.	Trang đăng nhập	56
5.2.2.	Trang Dashboard	59
5.2.3.	Trang Tiến độ kế hoạch	61
5.2.4.	Trang Kho vận và Chi tiết kho vận	65
5.2.5.	Trang Phòng QA/QC thiết bị và các trang Chi tiết.....	66
5.2.6.	Trang Khu máy ép tổng quan và trang các máy ép chi tiết.....	67
5.2.7.	Trang Khu đóng gói và trang Chi tiết từng cụm máy.....	69
5.2.8.	Trang Khu vực lắp ráp.....	70
5.2.9.	Trang Báo cáo các khu vực	71
5.2.10.	Trang Báo cáo chỉ số Overall Equipment Effectiveness (OEE index)	73
5.2.11.	Trang cảnh báo	77
5.3.	Lập trình Web App	78
5.3.1.	Phương pháp lập trình	78
5.3.2.	Giao thức truyền nhận dữ liệu liên tục	81
5.3.3.	Các bước lập trình	83
Chương 6.	KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ	86
6.1.	Kết quả thực hiện	86
6.1.1.	Khu vực máy ép	86
6.1.2.	Khu vực đóng gói	88

6.1.3.	Khu vực lắp ráp.....	91
6.1.4.	Phòng KTCLSP.....	92
6.1.5.	Truy xuất dữ liệu báo cáo và tính toán chỉ số OEE	94
6.2.	Đánh giá.....	100
6.3.	Hướng phát triển.....	101
	TÀI LIỆU THAM KHẢO.....	102

DANH MỤC HÌNH

Hình 2.1 Sơ đồ hiện tại của nhà máy và ghi chú các khu vực	6
Hình 2.2 Sơ đồ khu vực máy ép của nhà máy.....	7
Hình 2.3 Bộ phân loại sản phẩm đóng gói	8
Hình 2.4 Bộ đếm các chi tiết lắp ráp.....	9
Hình 2.5 Khu vực kho vận nhìn từ trên.....	10
Hình 2.6 Phòng QA/QC thiết bị (phòng KTCLSP)	11
Hình 2.7 File kế hoạch sản xuất mỗi tháng của công ty	12
Hình 2.8 Sơ đồ thể hiện ý tưởng xây dựng hệ thống thu thập dữ liệu tại công ty	14
Hình 2.9 Mô hình kiến trúc tổng quan của dự án	15
Hình 3.1 Sơ đồ cấu trúc hệ thống.....	17
Hình 4.1 Hình ảnh mô tả kiến trúc của InfluxDB	20
Hình 4.2 Hình ảnh mô tả giao thức MQTT	22
Hình 4.3 Socket trong mô hình OSI so với Internet.....	25
Hình 4.4 Hình ảnh mô tả giao tiếp SocketTCP	26
Hình 4.5. Hình ảnh mô tả cấu trúc topic namespace.....	28
Hình 4.6 Ví dụ về payload của Sparkplug.....	31
Hình 4.7 Sơ đồ tổng quan hệ thống.....	32
Hình 4.8 Mô hình hệ thống thu thập dữ liệu khu vực máy ép.....	34
Hình 4.9 Lưu đồ giải thuật node thu thập dữ liệu khu vực máy ép.....	35
Hình 4.10 Mô hình hệ thống thu thập dữ liệu khu vực lắp ráp	39
Hình 4.11 Lưu đồ giải thuật node thu thập cụm lắp ráp	40
Hình 4.12 Mô hình hệ thống thu thập dữ liệu khu vực đóng gói	42
Hình 4.13 Lưu đồ giải thuật node thu thập cụm đóng gói	43
Hình 4.14 Cấu trúc dữ liệu quản lý tag.....	46
Hình 4.15 Cấu trúc chương trình của máy chủ	47

Hình 4.16 Cấu trúc dữ liệu cho chương trình đọc dữ liệu InfluxDB.....	48
Hình 5.1 Giao diện phần mềm Postman	54
Hình 5.2 Sơ đồ các trang của ứng dụng web giám sát và thu thập dữ liệu	55
Hình 5.3 Sidebar (Ngăn kéo) của web giúp chuyển hướng trang	56
Hình 5.4 Thiết kế giao diện đăng nhập.....	56
Hình 5.5 Giao diện đăng nhập của ứng dụng.....	57
Hình 5.6 Chuyển hướng trang đăng nhập.....	57
Hình 5.7 Giải thuật xử lý đăng nhập	58
Hình 5.8 Sơ đồ chuyển hướng của trang Dashboard	59
Hình 5.9 Thiết kế giao diện trang Dashboard	60
Hình 5.10 Giao diện trang Dashboard của ứng dụng.....	60
Hình 5.11 Cấu trúc dữ liệu báo cáo kiểm tra tiến độ khu vực máy ép	61
Hình 5.12 Cấu trúc dữ liệu báo cáo kiểm tra tiến độ khu vực đóng gói	62
Hình 5.13 Giải thuật sàng lọc dữ liệu với cấu trúc dữ liệu được định sẵn.....	63
Hình 5.14 Lưu đồ giải thuật xử lý dữ liệu theo trang kiểm tra tiến độ kế hoạch sản xuất	64
Hình 5.15 Giao diện báo cáo kiểm tra tiến độ sản xuất theo ngày.....	65
Hình 5.16 Giao diện báo cáo kiểm tra tiến độ sản xuất theo tháng.....	65
Hình 5.17 Thiết kế giao diện trang Kho vận tổng quát (trái) và giao diện phần mềm (phải)	66
Hình 5.18 Thiết kế giao diện trang kho chi tiết (trái) và giao diện phần mềm (phải)	66
Hình 5.19 Thiết kế giao diện trang PKTCLSP tổng quát (trái) và giao diện ứng dụng (phải)	67
Hình 5.20 Thiết kế giao diện trang giám sát thông số các máy của PKTCLSP chi tiết (trái) và giao diện ứng dụng (phải)	67
Hình 5.21 Giám sát thông số máy ép dưới dạng bản đồ nhà máy.....	68
Hình 5.22 Giám sát thông số máy ép dưới dạng bảng (danh sách).....	68
Hình 5.23 Giao diện trang giám sát thông số chi tiết của máy ép	69

Hình 5.24 Thiết kế giao diện trang giám sát quy trình đóng gói (trái) và giao diện ứng dụng.....	69
Hình 5.25 Giao diện giám sát thông số hoạt động chi tiết của cụm máy đóng gói	70
Hình 5.26 Giao diện demo giám sát khu vực lắp ráp	70
Hình 5.27 Cấu trúc dữ liệu của trang báo cáo phòng KTCLSP	71
Hình 5.28 Mô tả trang báo cáo khu vực Phòng QA/QC thiết bị	71
Hình 5.29 Mô tả trang báo cáo khu vực máy ép.....	72
Hình 5.30 Mô tả đồ thị báo cáo của khu vực máy ép.....	72
Hình 5.31 Mô tả trang báo cáo khu vực đóng gói sản phẩm.....	73
Hình 5.32 Thiết kế giao diện trang báo cáo chỉ số OEE	73
Hình 5.33 Giao diện trang báo cáo và tính toán chỉ số OEE	74
Hình 5.34 Cấu trúc dữ liệu của chỉ số OEE	76
Hình 5.35 Thiết kế giao diện trang cảnh báo (trái) và giao diện ứng dụng.....	77
Hình 5.36 Mô tả quy chức năng excel cảnh báo của ứng dụng.....	77
Hình 5.37 Mô hình hoạt động của kiến trúc Redux	78
Hình 5.41 Mô tả quy trình đăng nhập bằng kỹ thuật OpenID Connect.....	80
Hình 5.42 Sơ đồ mô tả cách luồng dữ liệu hoạt động trên SignalR WebSocket.	81
Hình 6.1 Windows Service chạy ổn định trên máy tính quản lý khu vực máy ép	86
Hình 6.2 Dữ liệu 2 máy ép ghi nhận được trên database thể hiện dưới dạng biểu đồ.....	87
Hình 6.3 Kết quả hiển thị dữ liệu tổng quát cập nhật từ khu vực máy ép	87
Hình 6.4 Kết quả hiển thị dữ liệu tổng quát cập nhật từ khu vực máy ép dưới dạng sơ đồ công ty.....	88
Hình 6.5 Trang giám sát chi tiết máy ép công suất L6.....	88
Hình 6.6 Windows Service chạy ổn định khi đọc dữ liệu cụm đóng gói DG2	89
Hình 6.7 Windows Service chạy ổn định khi đọc dữ liệu cụm đóng gói DG2	89

Hình 6.8 Dữ liệu cụm đóng gói ghi nhận được trên database thể hiện dưới dạng biểu đồ	90
Hình 6.9 Kết quả hiển thị dữ liệu cập nhật từ khu vực đóng gói	90
Hình 6.10 Trang giám sát chi tiết cụm máy đóng gói 3.....	91
Hình 6.11 Windows Service chạy ổn định khi đọc dữ liệu cụm lắp ráp LR1	91
Hình 6.12 Dữ liệu cụm lắp ráp ghi nhận được trên database thể hiện dưới dạng biểu đồ	92
Hình 6.13 Kết quả hiển thị dữ liệu cập nhật từ khu vực lắp ráp.....	92
Hình 6.14 Kết quả hiển thị dữ liệu cập nhật từ phòng QA/QC thiết bị	93
Hình 6.15 Trang giám sát chi tiết máy kiểm tra độ biến dạng hệ 1.....	93
Hình 6.16 Trang giám sát chi tiết máy kiểm tra độ biến dạng hệ 2.....	94
Hình 6.17 Kết quả truy xuất của trang tổng quát kho vận có chú thích	95
Hình 6.18 Kết quả truy xuất của trang chi tiết mã sản phẩm kho vận có chú thích	95
Hình 6.19 Trang báo cáo máy ép kèm chú thích.....	96
Hình 6.20 Truy xuất và thể dữ liệu trong trang báo cáo khu vực đóng gói kèm chú thích.....	97
Hình 6.21 Xuất file báo cáo thành công.....	97
Hình 6.22 Hình ảnh thể hiện kết quả truy xuất dữ liệu báo cáo tháng.....	98
Hình 6.23 Kết quả báo cáo kiểm tra độ biến dạng với bài kiểm tra chịu tải tĩnh kèm ghi chú.....	99
Hình 6.24 Kết quả truy xuất trang báo cáo chỉ số OEE kèm chú thích.....	100

DANH MỤC BẢNG

Bảng 4.1 Bảng quy đổi kiểu dữ liệu	31
Bảng 4.2 Bảng dữ liệu máy ép	38
Bảng 5.1 Bảng tham chiếu tương đương cấu trúc dữ liệu và công thức tính chỉ số OEE	77
Bảng 5.2 Các thuộc tính trong sơ đồ hiển thị của <i>PackingPlanTracking</i>	85

DANH MỤC TỪ VIẾT TẮT

WA:	Web Application, chỉ các phần mềm ứng dụng chạy trên web
CSDL:	Cơ sở dữ liệu
IPC:	Industrial PC - máy tính công nghiệp
SSMS:	SQL Server Management Studio
HTTP:	Hypertext Transfer Protocol
UI:	User Interface
UX:	User Experience
BL:	Business Logic
PKTCLSP:	Phòng kiểm tra chất lượng sản phẩm hay còn gọi là Phòng QA/QC thiết bị
DA:	Desktop App
OEE:	Overall Equipment Effectiveness
SPA:	Single Page Application
CSS:	Cascading Style Sheets
MQTT:	MQ Telemetry Transport
JWT:	JSON Web Token

LỜI CẢM ƠN

Quá trình thực hiện luận văn tốt nghiệp là giai đoạn quan trọng nhất của sinh viên. Để hoàn thành được luận văn này, nhóm em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hỗ trợ đến từ nhiều thầy cô, tổ chức cũng như bạn bè và các em khóa dưới. Nhóm em xin được gửi lời cảm ơn chân thành đến tất cả mọi người đã tạo điều kiện và giúp đỡ nhóm em trong khoảng thời gian học tập, nghiên cứu luận văn vừa qua.

Trước hết, nhóm xin cảm ơn đến Ban giám hiệu Trường Đại học Bách khoa – Đại học Quốc gia TP.HCM, Khoa Điện – Điện tử và Bộ môn Điều khiển Tự động của trường đã tạo điều kiện, cơ sở vật chất và môi trường học tập cực kỳ tốt để nhóm rèn luyện và học hỏi thêm nhiều kiến thức trong suốt những năm tháng đại học.

Nhóm em cũng xin gửi lời cảm ơn đến Công ty Cổ phần nhựa đã tạo điều kiện, cơ hội cho nhóm em ở công ty được tiếp cận với các quy trình sản xuất tại nhà máy để nhóm em có thể thực hiện đề tài luận văn một cách chi tiết và mang nhiều tính ứng dụng hơn.

Đặc biệt, nhóm em xin cảm ơn sâu sắc đến TS. Trần Ngọc Huy – giảng viên hướng dẫn của nhóm. Thầy là người đã tận tình hướng dẫn và định hướng cho nhóm trong suốt quá trình thực hiện luận văn.

Và nhóm cũng xin gửi lời cảm ơn đến các bạn sinh viên khóa K18 và các anh khóa K17 và các em khóa K19 – những người luôn sẵn sàng hỗ trợ và giúp đỡ, chia sẻ kinh nghiệm quý báu cho nhóm em trong học tập lẫn cuộc sống.

Cuối cùng, vì kinh nghiệm và kiến thức còn hạn chế, luận văn của nhóm sẽ không thể tránh khỏi những sai sót. Nhóm em rất mong nhận được những lời nhận xét và góp ý từ quý thầy cô cũng như bạn bè để giúp nhóm hoàn thiện luận văn và nâng cao kiến thức chuyên ngành hơn.

Nhóm em xin chân thành cảm ơn.

LỜI MỞ ĐẦU

Hiện nay, cuộc cách mạng công nghiệp lần thứ 4 đã và đang tác động mạnh mẽ đến xu hướng nghiên cứu và cải tiến quy trình sản xuất của các công ty. Xu hướng giám sát dữ liệu từ xa, Internet vạn vật công nghiệp (IIoT) đang rất được quan tâm bởi các doanh nghiệp ở các quy mô khác nhau. Những lợi ích của việc lưu trữ dữ liệu dưới dạng điện tử là vô cùng lớn, giúp cải thiện hiệu suất làm việc cả về tốc độ và độ chính xác của nhà máy. Lợi ích mang lại khi có thể kiểm tra được trạng thái sản xuất tại nhà máy mà không cần phải xuống tận từng khu vực sản xuất để kiểm tra là vô cùng lớn, giúp cải thiện tối đa những lịch bảo trì định kỳ không đáng có. Đặc biệt là với doanh nghiệp vừa và nhỏ, ở các doanh nghiệp này, các khu vực sản xuất đang rời rạc với nhau, không có tính liên kết và các báo cáo của họ chỉ dùng ở viết bằng giấy tờ và sử dụng excel. Không những thế, các kỹ sư có tay nghề thường sẽ không chú tâm vào các doanh nghiệp sản xuất vừa và nhỏ vì họ hiểu được các doanh nghiệp này đang thiếu một hệ thống kiểm tra, giám sát từ xa giúp các kỹ sư nâng cao trình độ tay nghề và tiết kiệm thời gian vận hành sản xuất vì hạn chế được việc giám sát quy trình tại chỗ của nhà máy. Chính vì mục đích muốn cải thiện công tác quản lý trạng thái sản xuất nhà máy, nhóm đã xây dựng một hệ thống giám sát và thu thập dữ liệu để hỗ trợ các trưởng ca, trưởng bộ phận.

TÓM TẮT LUẬN VĂN

Từ các module sản xuất rời rạc hiện tại của một công ty nhựa, nhóm em nhận thấy đây là một cơ hội tiềm tàng để phát triển một hệ thống giám sát và thu thập dữ liệu, nhất là ở các khu vực: Khu vực máy ép, Khu vực đóng gói, Phòng kiểm tra chất lượng sản phẩm, Kho vận, Khu vực lắp ráp. Nhóm sẽ ưu tiên xây dựng mô hình tại các khu vực này trước vì thường xuyên xảy ra các trường hợp: không đạt chỉ số đo lường hiệu quả công việc, nhân viên đứng máy gian lận chỉ số cài đặt trong quá trình làm việc, các máy kiểm tra chất lượng sản phẩm xảy ra lỗi đột ngột và không có nhân viên trực ca có mặt tại thời điểm đó. Qua đó, nhóm đề ra phương án sử dụng ngôn ngữ C# để xây dựng phần mềm Windows Service để đọc dữ liệu một cách mạch lạc thông qua MQTT Sparkplug, kết hợp với ngôn ngữ JavaScript và thư viện ReactJS để xây dựng phần mềm web. Phần mềm này có chức năng trực quan hóa các dữ liệu nhận được liên tục từ máy chủ, truy xuất báo cáo và tính toán các chỉ số sản xuất cho 2 khu vực máy ép và khu vực đóng gói như chỉ số OEE, tiến độ thực hiện theo kế hoạch sản xuất ngày và tháng đã định ra trước đó. Với dự án này, mô hình cần thực hiện đọc thành công và truy xuất dữ liệu 40 máy ép, 6 cụm đóng gói, 4 thiết bị kiểm tra chất lượng sản phẩm, các dữ liệu đã được xây dựng sẵn trên CSDL máy chủ của công ty để phục vụ mục đích theo dõi báo cáo và tính toán nhiều chỉ số sản xuất khác nhau. Tuy nhiên trong đề tài với phạm vi sinh viên, nhóm đặt ra mục tiêu đọc dữ liệu 2 máy ép L6, L10, 1 máy lắp ráp LR1, 2 máy đóng gói DG2, DG3, truy xuất dữ liệu báo cáo và tính toán hàm mục tiêu OEE.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Như đã đề cập trước đó, sau khi quan sát hoạt động của công ty nhựa, nhóm nghiên cứu nhận thấy đây là cơ hội tiềm năng để phát triển hệ thống thu thập dữ liệu cho khu vực bên trong nhà máy. Nhà máy bỏ qua giai đoạn giám sát và trách nhiệm giám sát thuộc về các bộ phận ở từng khu vực. Điều này bị ảnh hưởng bởi yếu tố con người, mất đi tính minh bạch của sản xuất, không đảm bảo được khối lượng sản xuất và chất lượng của sản phẩm. Về lâu dài có thể làm hỏng hình ảnh của công ty và ảnh hưởng đến công ty.

Để có thể giải quyết được vấn đề này, cần rất nhiều thời gian và chất xám. Vì vậy, sau khi nhóm trao đổi với ban lãnh đạo của công ty, nhóm sẽ thực hiện giai đoạn đầu tiên trong cuộc đổi mới quy trình này - bằng việc xây dựng các node dữ liệu tại các khu vực, cụm đã có sẵn các máy có thể giao tiếp với cơ sở dữ liệu thông qua các máy tính quản lý của mỗi khu vực, cụ thể là các khu vực: kho vận, máy ép, đóng gói, lắp ráp, phòng kiểm tra chất lượng sản phẩm.

1.2. Nhiệm vụ đề tài

- Khi thu thập dữ liệu trên máy tính quản lý của mỗi khu vực, dữ liệu được thu thập bằng phần mềm chạy nền được gọi là node của mỗi khu vực. Dữ liệu được lưu trữ trong cơ sở dữ liệu dành riêng cho việc truyền dữ liệu. Phần mềm nền được viết bằng ngôn ngữ C# và SQL.
- Xây dựng phần mềm web chạy trên đa trình duyệt dựa trên ngôn ngữ JavaScript. Phần mềm cần chạy ổn định, cập nhật dữ liệu liên tục, cảnh báo lỗi kịp thời. Giao diện rõ ràng, tách biệt tốt và nhân viên dễ tiếp cận. Ngoài ra, phần mềm cần hạn chế các lỗi liên quan đến logic và quy trình tẻ nhạt để tối ưu hóa giao diện người dùng và trải nghiệm người dùng.

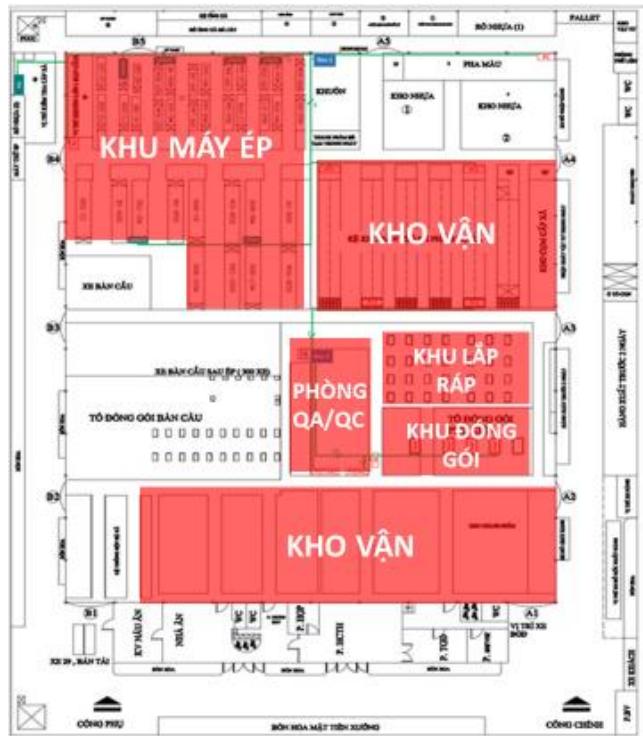
1.3. Phương pháp thực hiện

Để hoàn thành đề tài, nhóm đề ra phương pháp thực hiện như sau:

- Phân công công việc rõ ràng, mỗi người phụ trách một mảng công nghệ.
- Nghiên cứu và lựa chọn ra những công nghệ đang là xu hướng của thế giới, với các tiêu chí như sau: mức độ phù hợp, sự phổ biến, sự ổn định của nền tảng, được hỗ trợ dài hạn, cộng đồng người dùng đông đảo...
- Lên phương án thiết kế giao diện bằng những công cụ chuyên dụng.
- Thường xuyên trao đổi với ban lãnh đạo của công ty và các trưởng ca, trưởng bộ phận để nắm bắt được các nhu cầu sử dụng của công ty.
- Lập trình và phát triển các ứng dụng, trong quá trình đó phải thường xuyên thực hiện các bài kiểm tra để đánh giá xem đã phù hợp với người dùng hay chưa.
- Xây dựng hệ thống để có thể chạy thử nghiệm, mục tiêu hướng đến là chạy thử nghiệm tại môi trường nhà máy để có thể đánh giá chính xác chất lượng vận hành của các ứng dụng.

Chương 2. TỔNG QUAN CÁC KHU VỰC HIỆN TẠI TẠI NHÀ MÁY

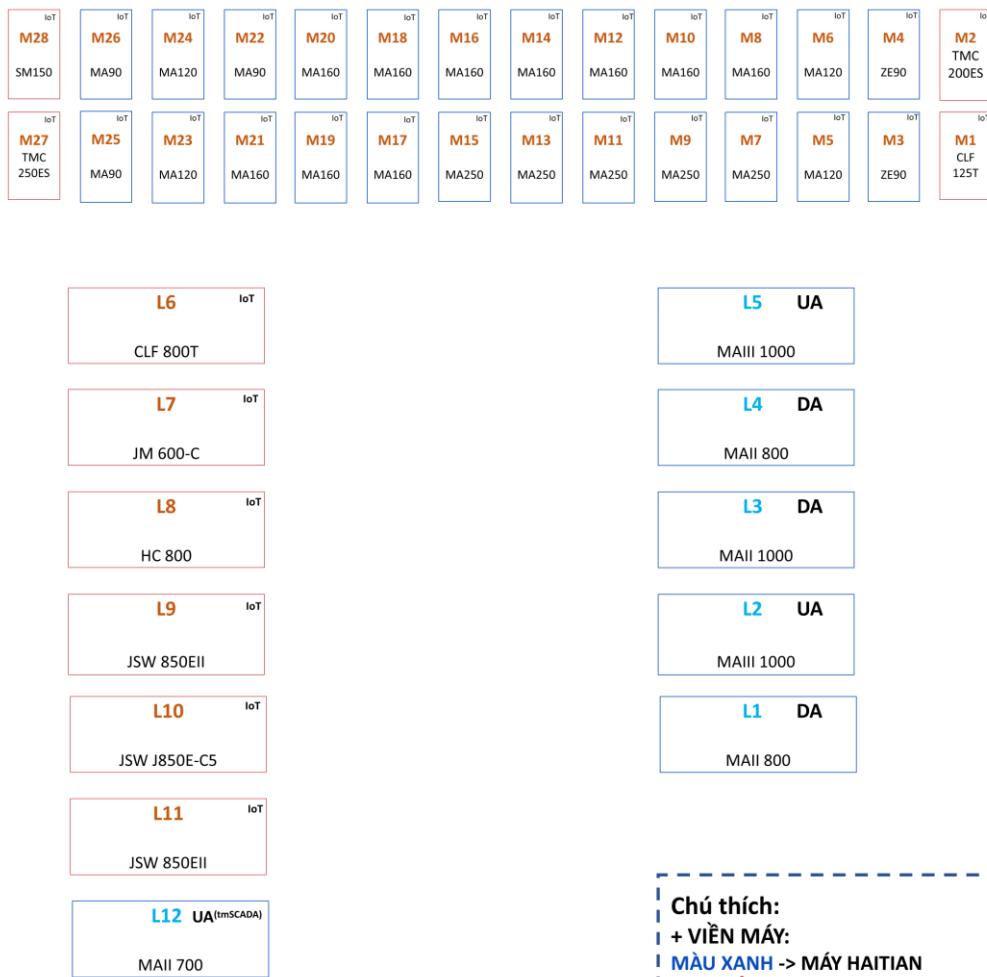
2.1. Thực trạng nhà máy



Hình 2.1 Sơ đồ hiện tại của nhà máy và ghi chú các khu vực

Hiện tại, các khu vực sản xuất tại nhà máy đang có các thiết bị vận hành như sau:

2.1.1. Khu vực máy ép



Chú thích:
 + VIỀN MÁY:
MÀU XANH -> MÁY HAITIAN
MÀU ĐỎ -> MÁY HÃNG KHÁC

+ TÊN MÁY
MÀU XANH -> [Giải pháp OPC UA/DA](#) (click vào link để xem chi tiết) (6 máy)
MÀU CAM -> [Giải pháp IoT](#) (34 máy)

Hình 2.2 Sơ đồ khu vực máy ép của nhà máy

Bao gồm 28 máy ép có công suất nhỏ và 12 máy ép có công suất lớn. Định hướng của công ty trong 2 năm sẽ thực hiện mở rộng mô hình đọc thông số chu kỳ ép (bao gồm chu kỳ ép kỹ thuật và thời gian thực hiện thủ công của công nhân, hay còn có tên gọi khác là thời gian cửa mở) của tất cả các máy hiện có. Một ứng dụng máy tính đặt tại khu vực để quản lý hoạt động các máy và trích xuất báo cáo cần thiết.

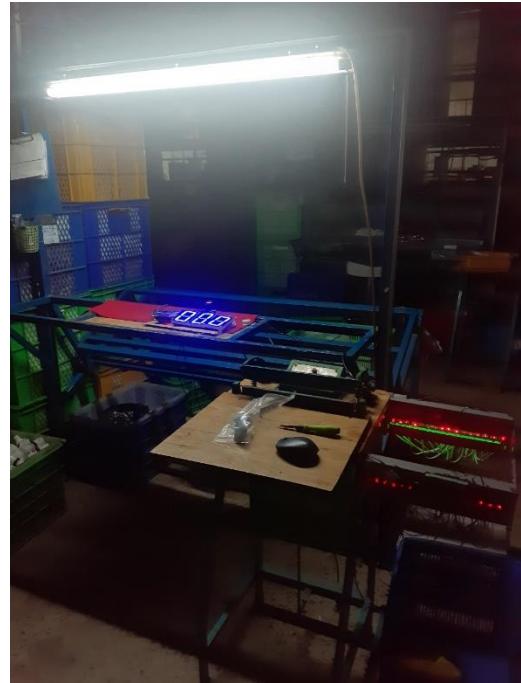
2.1.2. Khu vực đóng gói



Hình 2.3 Bộ phân loại sản phẩm đóng gói

Đã triển khai một hế thống phân loại sản phẩm lõi hoặc thiếu, trong năm nay sẽ triển khai nhân rộng hệ thống này cho năm cụm máy còn lại. Ngoài ra, có một phần mềm quản lý các công nhân đóng gói, tiến độ thực hiện của họ và xuất ra các báo cáo ca cần thiết cho khu vực.

2.1.3. Khu vực lắp ráp



Hình 2.4 Bộ đếm các chi tiết lắp ráp

Hiện tại tuy đã có một hệ thống kiểm tra số lượng thực hiện của các công nhân lắp ráp nhưng chưa ứng dụng thực tế. Tuy nhiên khả năng đưa hệ thống vào vận hành cao nên nhóm quyết định sẽ xây dựng một node cho khu vực lắp ráp.

2.1.4. Kho vận



Hình 2.5 Khu vực kho vận nhìn từ trên

Ở khu vực kho vận là trường hợp đặc biệt khi trưởng bộ phận không đặt ra yêu cầu cần dữ liệu liên tục. Tuy nhiên, hệ thống cần phải biết được số lượng của từng mã sản phẩm trong kho, vị trí tương ứng của mỗi rõ sản phẩm và thẻ kho của mã sản phẩm đó theo ngày cụ thể.

2.1.5. Phòng kiểm tra chất lượng sản phẩm



Hình 2.6 Phòng QA/QC thiết bị (phòng KTCLSP)

Phòng kiểm tra chất lượng phẩm là phòng có các máy kiểm tra tự động với thời gian vận hành cao. Cụ thể như hình trên, tại PKTCLSP, các máy có thể giao tiếp với ứng dụng máy tính để lưu trữ dữ liệu bao gồm:

- Máy kiểm tra độ bền đóng êm của sản phẩm nắp bồn cầu
- Máy kiểm tra độ bền đóng cưỡng bức của sản phẩm nắp bồn cầu
- Máy kiểm tra độ biến dạng của sản phẩm nắp bồn cầu
- Máy kiểm tra độ bền chống thấm

Tại phòng đã có một ứng dụng máy tính quản lý, vận hành các máy và truy xuất báo cáo liên quan đến khâu kiểm tra chất lượng sản phẩm của công ty.

2.1.6. Kế hoạch sản xuất và lịch trình sản xuất

A	B	C	D	E	F	G	H	I	J	K	L	M	N
STT	Seri nhóm	Mã số	Tên sản phẩm	Đvt	Dung sản phẩm	Dòng sản phẩm	Nắp bát cầu	Bộ sả	Bộ hơi & phụ kiện bát cầu	Cấp xả & gạt nhún	Linh kiện bát cầu	Linh xả	Tổng kế hoạch
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12	12	12	12	12	12	12
13	13	13	13	13	13	13	13	13	13	13	13	13	13
14	14	14	14	14	14	14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15	15	15	15	15	15	15
16	16	16	16	16	16	16	16	16	16	16	16	16	16
17	17	17	17	17	17	17	17	17	17	17	17	17	17
18	18	18	18	18	18	18	18	18	18	18	18	18	18
19	19	19	19	19	19	19	19	19	19	19	19	19	19
20	20	20	20	20	20	20	20	20	20	20	20	20	20
21	21	21	21	21	21	21	21	21	21	21	21	21	21
22	22	22	22	22	22	22	22	22	22	22	22	22	22
23	23	23	23	23	23	23	23	23	23	23	23	23	23
24	24	24	24	24	24	24	24	24	24	24	24	24	24
25	25	25	25	25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26	26	26	26	26
27	27	27	27	27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28	28	28	28	28
29	29	29	29	29	29	29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30	30	30	30	30	30	30
31	31	31	31	31	31	31	31	31	31	31	31	31	31
32	32	32	32	32	32	32	32	32	32	32	32	32	32
33	33	33	33	33	33	33	33	33	33	33	33	33	33
34	34	34	34	34	34	34	34	34	34	34	34	34	34
35	35	35	35	35	35	35	35	35	35	35	35	35	35
36	36	36	36	36	36	36	36	36	36	36	36	36	36
37	37	37	37	37	37	37	37	37	37	37	37	37	37
38	38	38	38	38	38	38	38	38	38	38	38	38	38
39	39	39	39	39	39	39	39	39	39	39	39	39	39
40	40	40	40	40	40	40	40	40	40	40	40	40	40
41	41	41	41	41	41	41	41	41	41	41	41	41	41
42	42	42	42	42	42	42	42	42	42	42	42	42	42
43	43	43	43	43	43	43	43	43	43	43	43	43	43

Hình 2.7 File kế hoạch sản xuất mỗi tháng của công ty

Cụ thể, mỗi tháng công ty sẽ có 1 file kế hoạch sản xuất định sẵn theo tháng và 1 file lịch trình sản xuất theo ngày. Các trưởng bộ phận sẽ điều tiết công việc xuống cho trưởng ca và trưởng ca dựa vào đó sẽ phân phôi việc cho nhân viên. Nhân viên thực hiện xong sẽ viết báo cáo cuối ca cho trưởng ca, và các trưởng bộ phận nếu muốn theo dõi tiến độ sẽ lấy các báo cáo cuối ca từ các trưởng ca và đối chiếu với kế hoạch sản xuất hoặc lịch trình tùy theo mục đích của người trưởng bộ phận đó. Dựa theo khái niệm kế hoạch sản xuất, khu vực máy ép, khu vực đóng gói, khu vực lắp ráp là được công ty soạn sẵn lịch trình và kế hoạch sản xuất, phòng kiểm tra chất lượng sản phẩm không nằm trong kế hoạch sản xuất của công ty.

Nhận xét:

- Các khu vực trên đã đạt được yêu cầu cơ bản để có thể xây dựng một node thu thập dữ liệu đó là một ứng dụng máy tính quản lý từng khu vực.

- Cấu trúc dữ liệu cũng như cơ sở dữ liệu đã được xây dựng từ trước đó, cũng như là các đầu cuối của API, từ các dữ liệu thô đó có thể tính toán báo cáo và các chỉ số sản xuất phù hợp.
- Các module đang vận hành rời rạc, các ứng dụng quản lý không liên kết trao đổi với nhau nên dễ xảy ra tình trạng thiếu hàng, xót hàng khi bàn giao từ khu vực này sang khu vực khác.
- Ở khu vực kho vận dù có phần mềm quản lý kho nhưng chỉ có chức năng xem tồn kho chứ không có chức năng kiểm tra vị trí rõ và số lượng tồn kho của mỗi rõ đó.
- Có thể tối ưu hóa quy trình đổi chiếu, so sánh kết quả thực hiện theo ngày hoặc tháng của khu vực máy ép, đóng gói với kế hoạch và lịch trình sản xuất cho các trưởng bộ phận, không cần phải thông qua báo cáo của từng ca.

2.2. Phương án đề xuất

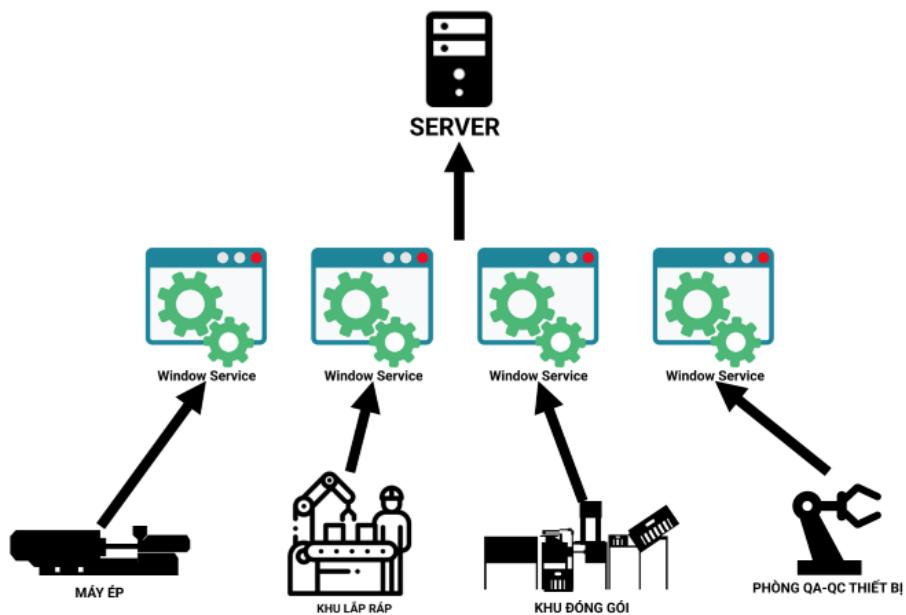
Dựa vào yêu cầu của công ty và điều kiện hiện tại, nhóm quyết định tập trung xây dựng các node lên ứng dụng máy tính đã có sẵn tại 4 khu vực nói trên (không bao gồm kho vận). Đối với kho vận, phần mềm sẽ lấy dữ liệu từ các đầu cuối của API có sẵn và trực quan hóa dữ liệu vào và ra của mã sản phẩm nhất định, điều mà phần mềm quản lý kho hiện tại ở công ty chưa đáp ứng được.



Hình 2.8 Sơ đồ thể hiện ý tưởng xây dựng hệ thống thu thập dữ liệu tại công ty Nhóm sẽ sử dụng ngôn ngữ C# để tạo ra các node thực hiện truyền tải luồng dữ liệu. Các luồng dữ liệu này được lưu vào cơ sở dữ liệu InfluxDB. Để con người có thể giao tiếp với cơ sở dữ liệu thì phải cần dùng đến phần mềm web sử dụng ngôn ngữ JavaScript hỗ trợ.

2.2.1. Xây dựng node cho các khu vực

Với định hướng của nhóm là xây dựng một hệ thống quản lý các thông số vận hành nhà máy dưới dạng tag, ta có thể bắt gặp trong nhiều phần mềm giám sát chuyên dụng hiện nay, nhóm quyết định xây dựng các node module thu thập dữ liệu liên tục phục vụ cho việc giám sát trạng thái sản xuất của nhà máy.



Hình 2.9 Mô hình kiến trúc tổng quan của dự án

2.2.2. Xây dựng chức năng chính của phần mềm web

Truy cập luồng dữ liệu để hiển thị thông số máy của các khu vực một cách trực quan nhất. Cụ thể:

- Khu vực máy ép: Cập nhật chu kỳ ép (bao gồm chu kỳ kỹ thuật và thời gian cửa mở), trạng thái, tiến độ hoạt động và mã sản phẩm đang thực hiện của từng máy hoạt động,
- Khu vực lắp ráp: Vì các máy đếm sản phẩm trong khu vực lắp ráp còn đang trong tiến trình phát triển nên nhóm chỉ có thể dừng lại ở việc xây dựng node cho trạng thái máy và giá trị đếm hiện tại, các tính năng như cảnh báo, báo cáo sẽ không phát triển ở khu vực này.
- Khu vực đóng gói: Đọc trạng thái hoạt động, thông tin đơn hàng đóng gói, số lượng sản phẩm lỗi, và nhân viên đứng ca của 6 cụm máy phân loại đóng gói.

- Phòng kiểm tra chất lượng sản phẩm: Giám sát thông số hoạt động của 4 máy bao gồm: thời điểm xảy ra các sự kiện, trạng thái máy hoạt động, tiến trình thực hiện.

Tính toán chỉ số OEE: Dựa trên yêu cầu quan sát chỉ số OEE của ban lãnh đạo công ty, phần mềm web có thuật toán tính chỉ số OEE cho khu vực máy dựa trên các dữ liệu thô có sẵn trong cơ sở dữ liệu theo công thức:

$$OEE = \frac{Availability (\%) * Performance (\%) * Quality (\%)}{10000}$$

Kiểm tra tiến độ theo kế hoạch sản xuất: Phần mềm có thể dựa vào dữ liệu có sẵn trong CSDL để tính toán, đưa ra tiến độ sản xuất theo ngày hoặc theo tháng của 2 khu vực máy ép và khu vực đóng gói sản phẩm. Nhóm nhận thấy vì khu vực lắp ráp chưa được triển khai tin học hóa dữ liệu nên dữ liệu có sẵn trên cơ sở dữ liệu là không đủ để có thể kiểm tra

Xây dựng báo cáo cho:

- Khu vực máy ép: Đồ thị thể hiện độ chênh lệch giữa chu kỳ ép cài đặt và chu kỳ ép thực tế của nhân viên, tránh tình trạng gian lận trong quá trình vận hành.
- Khu vực đóng gói: Hỗ trợ trưởng bộ phận xem và xuất file báo cáo tháng theo biểu mẫu được quy định sẵn trong công ty.
- Phòng kiểm tra chất lượng sản phẩm: Hỗ trợ trưởng bộ phận xem và xuất file báo cáo theo biểu mẫu được quy định sẵn trong công ty sau khi có kết quả kiểm tra của từng máy.

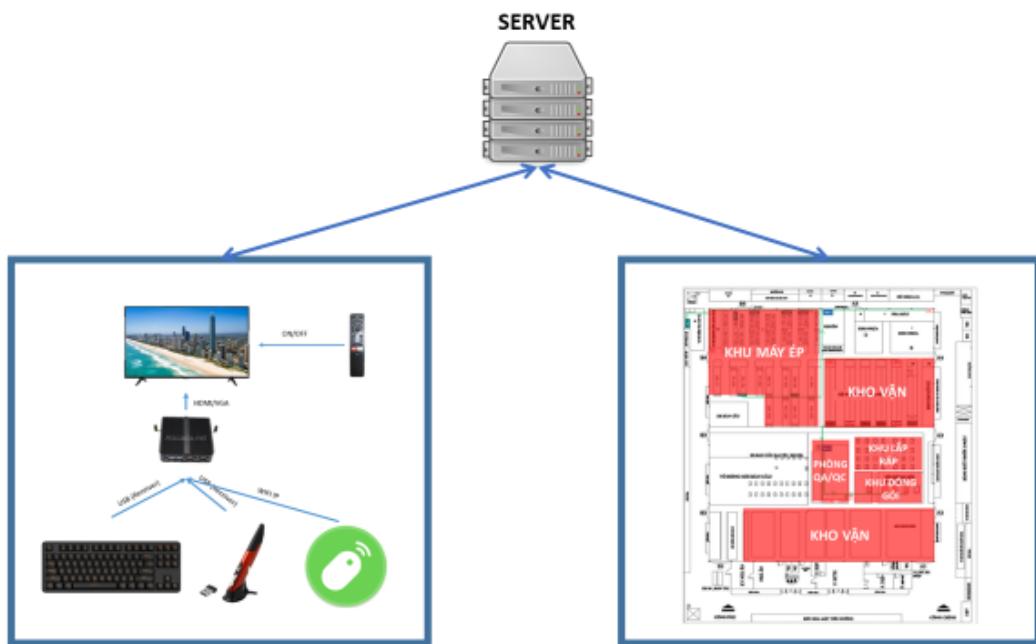
Theo dõi mã sản phẩm trong kho hàng: Hỗ trợ thủ kho biết được số lượng tồn kho cũng như vị trí sản phẩm kèm số lượng của mã chi tiết đó.

Chương 3. GIỚI THIỆU CÁC THIẾT BỊ TRONG HỆ THỐNG

3.1. Cấu trúc hệ thống

Hệ thống gồm:

- Phần mềm web được xem như một client để truy xuất dữ liệu máy chủ.
- Các phần mềm chạy ngầm tại các máy tính quản lý khu vực.
- Server: Server hay còn gọi là máy chủ là thiết bị làm trung tâm của hệ, là nơi lưu trữ các dữ liệu quan trọng của hệ thống. Dựa trên giao tiếp HTTP thông qua dây Ethernet hoặc Wifi để có thể phản hồi những yêu cầu gửi đến từ Client (máy tính để bàn và thiết bị di động).



Hình 3.1 Sơ đồ cấu trúc hệ thống

3.2. Xây dựng phương án

3.2.1. Các window service chạy trên máy tính quản lý của mỗi khu vực

Windows Service là 1 ứng dụng chạy nền trong một khoảng thời gian dài từ khi mở máy tính cho đến khi tắt nó đi. Nó có thể tự động chạy khi máy tính

được khởi động, có thể restart hay pause mà không cần một sự tác động nào của người dùng tới các công cụ liên quan tới UI.

Với tính chất như vậy, các window service sẽ có nhiệm vụ thu thập dữ liệu thông qua máy tính quản lý của mỗi khu vực.

3.2.2. Server – Máy chủ

Nếu đề tài này được ứng dụng tại nhà máy thì sẽ được sử dụng máy chủ của công ty. Vì thế trong quá trình thiết kế và xây dựng ứng dụng, nhóm sử dụng song song máy chủ được chạy trên đám mây (máy chủ ảo) và máy chủ được chạy tại local, vừa linh hoạt được việc ứng dụng tại nhà máy, vừa tiện trong việc truyền nhận từ xa trong thời gian giãn cách.

Với máy chủ chạy trên đám mây, chỉ cần chú ý đường dẫn URL đúng của cloud chứa cơ sở dữ liệu của máy chủ là được.

Chương 4. TỔNG QUAN HỆ THỐNG THU THẬP VÀ CẤU TRÚC DỮ LIỆU

4.1. Các công nghệ, ngôn ngữ và phần mềm sử dụng

4.1.1. Ngôn ngữ lập trình

Về ngôn ngữ lập trình, C# là ngôn ngữ lập trình hướng đối tượng có khả năng tạo ra nhiều ứng dụng mạnh mẽ cho nền tảng Windows như các thành phần máy chủ, ứng dụng máy tính, ứng dụng di động, dịch vụ web và nhiều khả năng khác nữa. C# cũng có rất nhiều thư viện hỗ trợ khác nhau với nhiều tính năng thuận tiện cho việc phát triển ứng dụng.

4.1.2. Cơ sở dữ liệu chuỗi thời gian InfluxDB

4.1.2.1. Cơ sở dữ liệu InfluxDB

InfluxDB là một hệ thống quản lý cơ sở dữ liệu được phát triển bởi InfluxData Inc. InfluxDB là mã nguồn mở và có thể được sử dụng miễn phí. Phiên bản InfluxDB Enterprise cung cấp các thỏa thuận bảo trì và kiểm soát truy cập đặc biệt cho khách hàng doanh nghiệp và được cài đặt trên máy chủ trong mạng công ty.

Flux được tối ưu hóa cho các quy trình ETL (trích xuất: extract, chuyển đổi: transform, tải: load) trong cơ sở dữ liệu và không tương thích với ngôn ngữ truy vấn InfluxQL đã sử dụng trước đó. Tuy nhiên, InfluxData đang lên kế hoạch cho các khách hàng hiện tại dịch code InfluxQL sang Flux.

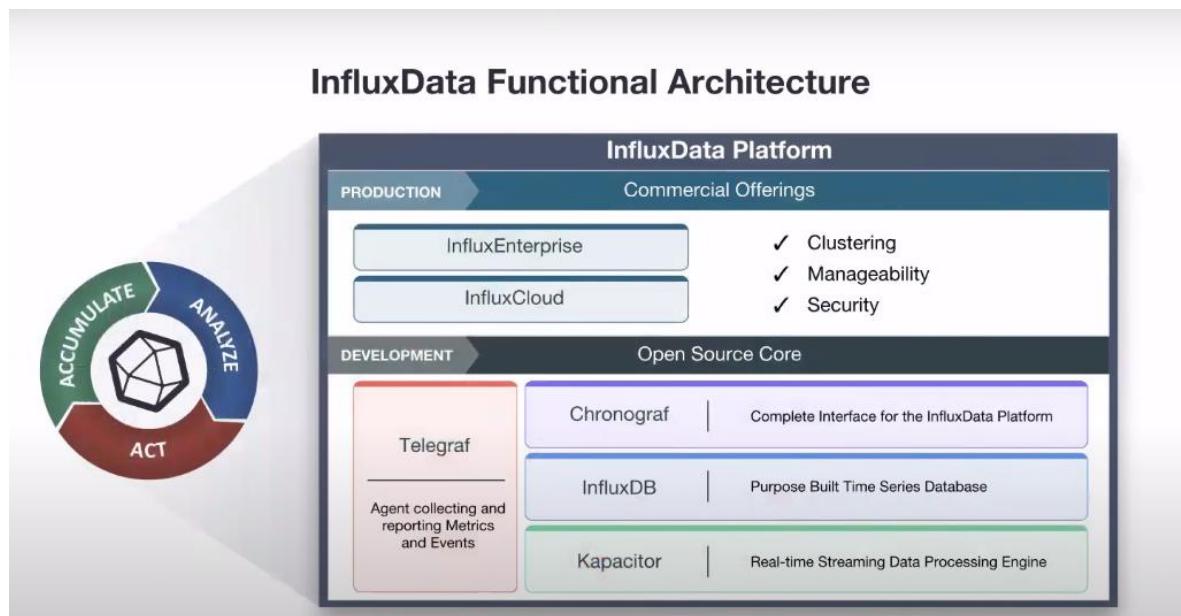
4.1.2.2. Lý do nên sử dụng InfluxDB

InfluxDB lý tưởng cho cơ sở dữ liệu chuỗi thời gian (TSDB), lưu trữ chuỗi thời gian – time series. Những cơ sở dữ liệu này được sử dụng, cùng với những thứ khác, để lưu trữ và phân tích dữ liệu cảm biến hoặc giao thức với dấu mốc thời gian trong một khoảng thời gian nhất định. Ví dụ: các thiết bị Internet of Things hoặc các công cụ đo lường khoa học cung cấp hàng triệu tập dữ liệu đến trong một luồng dữ liệu liên tục.

Dữ liệu này phải được xử lý nhanh chóng khi nó đến cơ sở dữ liệu. Vì lý do này, InfluxDB bao gồm một dịch vụ thời gian tích hợp sử dụng Giao thức thời gian mạng (NTP) để đảm bảo rằng thời gian được đồng bộ hóa giữa tất cả các hệ thống. Với InfluxDB, một cơ sở dữ liệu có thể rất nhỏ gọn và chỉ được chứa hai hoặc ba cột. Trong ví dụ này, nguồn dữ liệu, giá trị thực và dấu thời gian tương ứng được lưu trữ trong cơ sở dữ liệu.

4.1.2.3. Lợi thế của InfluxDB

So với các cơ sở dữ liệu quan hệ thông thường, Time Series DB như InfluxDB mang lại lợi thế về tốc độ rõ ràng khi lưu trữ và xử lý dữ liệu đo lường được đánh dấu mốc thời gian. DBMS truyền thống chậm lại khi tổ chức các chỉ mục phức tạp, các chỉ mục này hoàn toàn không được sử dụng trong lĩnh vực ứng dụng này. InfluxDB có thể duy trì tốc độ ghi cao trong một thời gian dài vì nó sử dụng một chỉ mục rất đơn giản.



Hình 4.1 Hình ảnh mô tả kiến trúc của InfluxDB

4.1.3. Cơ sở dữ liệu SQL Server 2019 và SSMS

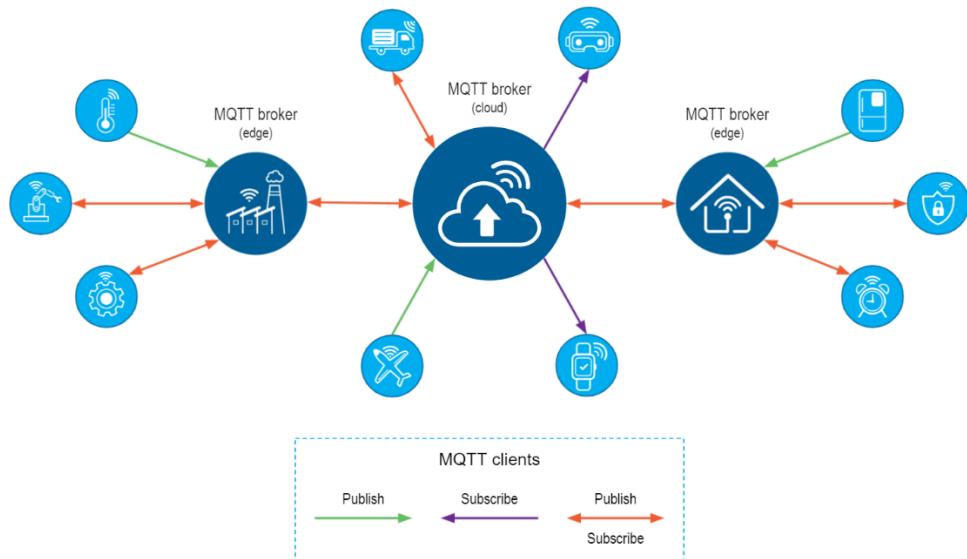
Cơ sở dữ liệu, hay còn được gọi tắt là database, là một hệ thống tập hợp các thông tin, dữ liệu được xây dựng theo một cấu trúc nhất định nhằm lưu trữ thông tin của tổ chức, doanh nghiệp. Để quản lý được các cơ sở dữ liệu ta cần có hệ quản trị cơ sở dữ liệu (Database Management System) - là một hệ thống được thiết kế cho phép quản lý và tương tác với cơ sở dữ liệu.

Hiện nay có rất nhiều hệ quản trị cơ sở dữ liệu khác nhau với các ưu nhược điểm riêng như MySQL, Oracle, Microsoft SQL Server, MongoDB... Trong đó, nhóm đã chọn sử dụng SQL Server. Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ giúp trao đổi thông tin với cơ sở dữ liệu do Microsoft phát triển.

Để làm việc với SQL Server cần có SQL Server Management Studio (SSMS). SSMS là một phần mềm được thiết kế bởi Microsoft, giúp lập trình viên thực hiện truy vấn, thiết kế và quản lý cơ sở dữ liệu SQL Server trên máy tính cục bộ hoặc trên cloud. Ứng dụng này cung cấp một bộ công cụ mạnh mẽ, giao diện tương tác trực quan để lập trình viên có thể cấu hình, giám sát và tương tác với các thể hiện của SQL Server.

4.1.4. Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức gởi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng machine to machine.



Hình 4.2 Hình ảnh mô tả giao thức MQTT

Publish, subscribe

Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là MQTT client - gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một hoặc nhiều kênh (topic), có định dạng "/IMM/M18/CycleMess". Quá trình đăng ký này gọi là "subscribe", bất kỳ khi nào topic nhận được dữ liệu mới thì tất cả các client đăng ký tới đều nhận được dữ liệu này. Khi một client gửi dữ liệu tới một topic nào đó thì gọi là "publish".

QoS (Qualities of service)

Mỗi một gói tin khi được gửi đi hoặc hành động đăng ký tới một topic đều đi kèm QoS. Có 3 tùy chọn QoS:

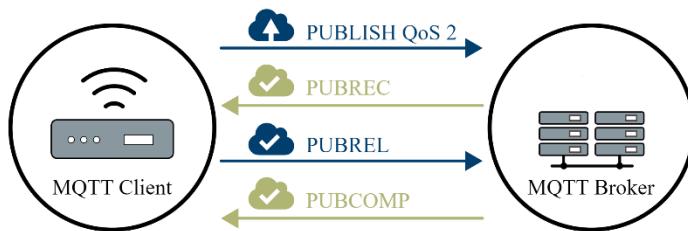
- QoS0 Broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP



- QoS1 Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu



- QoS2 Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay



Retain

Còn Retain được gắn với mỗi một tin nhắn, khi broker nhận được tin nhắn với cờ Retain = 1 thì broker phải lưu trữ lại tin nhắn đó và gửi đến bất kỳ client nào đăng ký đến topic đó trong tương lai. Tại một thời điểm broker chỉ lưu một tin nhắn Retain trên một topic, vì vậy tin nhắn Retain hiện tại sẽ bị xoá khi nhận một tin nhắn mới có cờ Retain = 1.

LWT (Last Will and Testament)

Tin nhắn LWT là một tính năng đặc biệt của MQTT. Khi một client kết nối đến broker, cấu hình có thể bao gồm topic LWT như "IMM/L6/LWT" với tin nhắn "Lost connect". Broker sẽ lưu trữ tin nhắn này, khi client có cấu hình LWT trước đó mất kết nối đến broker thì broker sẽ gửi tin nhắn LWT đến tất cả các client đăng ký đến topic LWT. Bằng tính năng này ta có thể xác định trạng thái kết nối đến một client đang publish dữ liệu.

4.1.5. Giao thức OPC UA

OPC UA là một trong những giao thức quan trọng nhất cho cuộc cách mạng công nghiệp 4.0 và các nền tảng IoT. Với OPC UA, việc tiếp cận với máy móc, thiết bị và các hệ thống trong môi trường công nghiệp đều được chuẩn hóa và cho các thiết bị không cùng nhà sản xuất có thể trao đổi dữ liệu.

OPC UA ra đời đã khắc phục được những hạn chế của OPC, đồng thời còn được cung cấp những tính năng để đáp ứng cho những yêu cầu của cách mạng công nghiệp 4.0. Đến năm 2017, OPC UA được cập nhật thêm tính năng Publish – Subscribe để đáp ứng những yêu cầu của các hệ thống TSN (hệ thống đòi hỏi dữ liệu được truyền với độ trễ rất thấp), tăng tốc độ truyền nhận dữ liệu bằng cơ chế Publish và Subscribe.

Những ưu điểm của OPC UA:

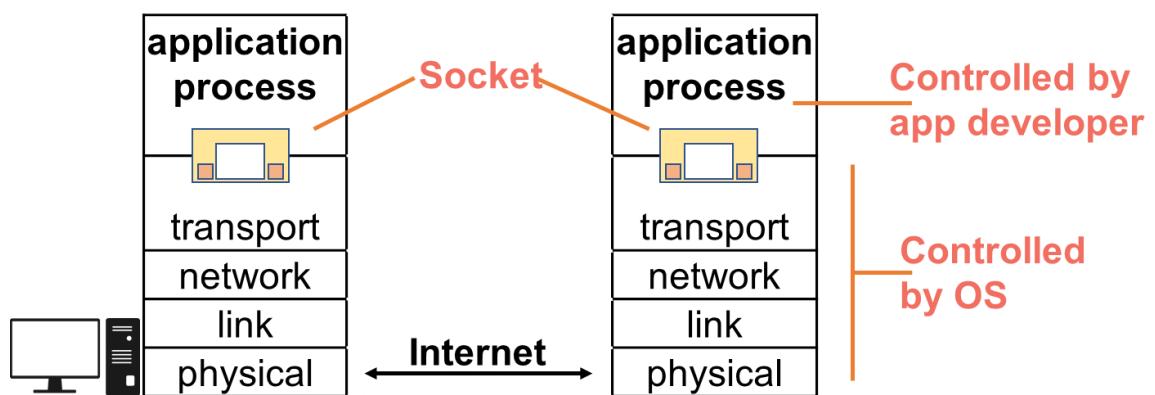
- OPC UA là chuẩn quốc tế IEC 62541.
- OPC UA Server và Client có thể được lập trình và chạy trên nhiều hệ điều hành khác nhau, cũng như các nền tảng phần cứng khác nhau.
- OPC UA Server và Client có thể được chạy trên các thiết bị trường như cảm biến và cơ cấu chấp hành. Như vậy dữ liệu từ cảm biến, thiết bị chấp hành có thể được đưa thẳng lên Cloud mà không cần phải thông qua phần mềm trung gian.
- Có tính bảo mật cao, sử dụng nhiều lớp bảo mật.
- Chính những ưu điểm trên, OPC UA được xem là nền tảng của công nghiệp 4.0.

4.1.6. Giao thức SocketTCP

Socket là một dạng kết nối để lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet, LAN. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để

kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

Một chức năng khác của socket là giúp các tầng TCP hoặc TCP Layer định danh ứng dụng mà dữ liệu sẽ được gửi tới thông qua sự ràng buộc với một cổng port (thể hiện là một con số cụ thể), từ đó tiến hành kết nối giữa client và server.

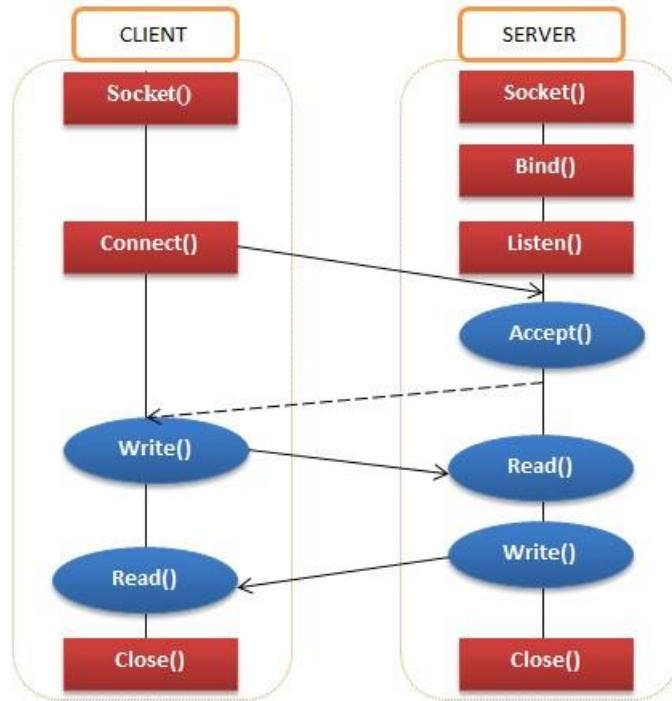


Hình 4.3 Socket trong mô hình OSI so với Internet

Chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

Socket có nhiều loại, tuy nhiên để có thể truyền thông qua tcp/ip giữa 2 thiết bị là Raspberry Pi và PC thông qua cổng LAN, ta chọn stream socket.

Stream Socket hay còn gọi là socket hướng kết nối, là socket hoạt động thông qua giao thức TCP (Transmission Control Protocol). Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau.



Hình 4.4 Hình ảnh mô tả giao tiếp SocketTCP

4.1.7. Tiêu chuẩn truyền thông MQTT Sparkplug

4.1.7.1. Giới thiệu

Sparkplug cung cấp tiêu chuẩn mở và miễn phí hỗ trợ cho các gateway Edge of Network (EoN) hoặc thiết bị đầu cuối hỗ trợ tương thích MQTT và ứng dụng MQTT, giao tiếp hai chiều trong hạ tầng mạng MQTT.

Sparkplug thừa nhận rằng MQTT được sử dụng trong nhiều trường hợp dùng làm giải pháp, ứng dụng và sự biến đổi gần như không thể xác định của các cấu trúc liên kết mạng. Để đạt được mục tiêu đó, đặc tả Sparkplug cỗ găng hoàn thành ba mục tiêu sau.

MQTT topic namespace:

Một trong nhiều tính năng hấp dẫn của MQTT là không chỉ định bất kỳ topic namespace bắt buộc nào trong quá trình triển khai. MQTT đã chiếm vị trí thống trị trên nhiều giải pháp IoT. Mục đích của tiêu chuẩn Sparkplug là xác

định và ghi lại Topic namespace đã được chọn lọc kỹ và tối ưu hóa cho lĩnh vực giải pháp SCADA/IIoT.

MQTT state management:

Một trong những khía cạnh đặc đáo của MQTT là: nó ban đầu được thiết kế cho các hệ thống SCADA thời gian thực, để giúp người dùng giảm độ trễ dữ liệu trên cơ sở hạ tầng mạng bị giới hạn băng thông và thường không đáng tin cậy. Trong nhiều cách triển khai, mặc dù lợi ích đầy đủ của “Continuous Session Awareness” này vẫn chưa được hiểu rõ, hoặc thậm chí không được triển khai. Mục đích của tiêu chuẩn Sparkplug là tận dụng tối đa khả năng Continuous Session Awareness của MQTT vì nó áp dụng cho các giải pháp SCADA/IIoT thời gian thực.

MQTT Payload:

Cũng giống như đặc điểm đầu MQTT không chỉ định bất kỳ topic namespace cụ thể nào và cũng không chỉ định bất kỳ payload mã hóa dữ liệu cụ thể nào. Mục đích của tiêu chuẩn Sparkplug là cố gắng xác định cấu trúc payload mã hóa và vẫn giữ được các tính chất ban đầu, nhẹ, hiệu quả băng thông, độ trễ thấp của MQTT.

Sparkplug đã xác định một cách tiếp cận mà topic namespace có thể hỗ trợ trong việc xác định sơ đồ mã hóa của bất kỳ payload cụ thể nào. Hiện tại, có hai lược đồ mã hóa do Sparkplug định nghĩa mà thông số kỹ thuật này hỗ trợ. Đầu tiên là lược đồ mã hóa Sparkplug A dựa trên định nghĩa Bộ giao thức “Kura open source Google Protocol Buffer”. Phương pháp thứ hai là lược đồ mã hóa Sparkplug B cung cấp mô hình dữ liệu phong phú hơn được phát triển với phản hồi của nhiều hệ thống tích hợp và các người dùng sử dụng MQTT.

4.1.7.2. Sparkplug topic namespace

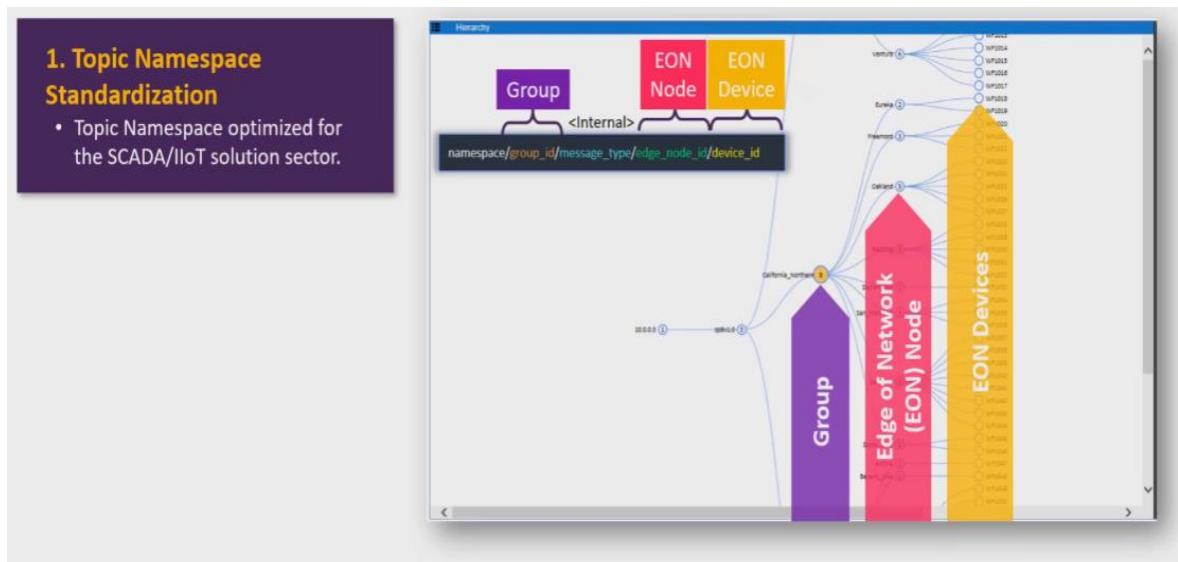
Cái hay của MQTT là ta có thể nghĩ ra một topic tùy ý như “Portland / Nhiệt độ”, kết nối với Máy chủ MQTT và bắt đầu publish giá trị nhiệt độ. Để dữ

liệu này hữu ích cho các ứng dụng MQTT Client khác muốn sử dụng các giá trị nhiệt độ, topic namespace cần được mọi người tham gia trao đổi dữ liệu hiểu kĩ.

Mỗi message MQTT được tạo ra bao gồm một topic và một payload. Các thành phần này là chung nhất của một tin nhắn MQTT được đo bằng byte trên dây. Đặc tả Sparkplug được thiết kế để giữ cho các thành phần này có ý nghĩa và dễ hiểu, nhưng không dài dòng đến mức ảnh hưởng tiêu cực đến việc trao đổi dữ liệu nhạy cảm với b/s.

Tất cả các máy khách MQTT sử dụng đặc tả Sparkplug™ sẽ sử dụng cấu trúc topic namespace sau:

namespace/group_id/message_type/edge_node_id/[device_id]



Hình 4.5. Hình ảnh mô tả cấu trúc topic namespace

Thành phần namespace: Đặc tả Sparkplug™ hiện tại xác định hai (2) namespace. Một dành cho định nghĩa payload Sparkplug™ A, và một định nghĩa khác dành cho định nghĩa payload Sparkplug™ B.

Đối với phiên bản Sparkplug™ A của định nghĩa trọng tải, hằng số chuỗi UTF-8 cho phần tử không gian tên sẽ là: "SpAv1.0"

Đối với phiên bản Sparkplug™ B của thông số kỹ thuật, hằng số chuỗi UTF-8 cho phần tử không gian tên sẽ là: "SpBv1.0"

Thành phần group_id: Phần tử group_id của topic namespace cung cấp một nhóm hợp lý các node MQTT EoN.

Thành phần message_type: Phần tử message_type của topic namespace cung cấp một chỉ dẫn về cách xử lý payload MQTT.

Các phần tử message_type sau được xác định như sau:

- NBIRTH - Giấy khai sinh cho các node MQTT EoN.
- NDEATH - Giấy khai tử cho các node MQTT EoN.
- DBIRTH - Giấy khai sinh cho thiết bị.
- DDEATH - Giấy khai tử cho thiết bị.
- NDATA - Thông báo dữ liệu node.
- DDATA - Thông báo dữ liệu thiết bị.
- NCMD - Thông báo lệnh node.
- DCMD - Thông báo lệnh thiết bị.
- STATE - Thông báo trạng thái ứng dụng.

Thành phần edge_node_id: Phần tử edge_node_id của topic namespace Sparkplug xác định duy nhất node MQTT EoN trong hạ tầng.

Thành phần device_id: Phần tử device_id của topic namespace Sparkplug™ xác định một thiết bị được gắn (vật lý hoặc logic) vào node MQTT EoN. Lưu ý rằng device_id là một phần tử tùy chọn trong topic namespace vì một số message sẽ có nguồn gốc hoặc được chuyển đến edge_node_id và device_id sẽ không bắt buộc.

4.1.7.3. Sparkplug payload

Một Sparkplug Payload có dạng là chuỗi JSON gồm các thành phần:

- Timestamp: nghĩa là mốc thời gian mà message được published, timestamp có dạng là số nguyên 64bit không dấu tính bằng ms được tính từ mốc 1/1/1970.
- Metrics: là một mảng các chỉ số các giá trị key/value/datatype, v.v
- Seq: là số thứ tự cho mỗi tin nhắn, tối đa là 256 giá trị từ 0 đến 255.

Trong metrics có các thành phần:

- Name: là tên mà ta cần truy cập có dạng chuỗi UTF-8 và phân cách bằng dấu gạch chép, được hiểu như là một đường dẫn thư mục
- Alias: là số nguyên 64bit không dấu, là đại diện cho một group EON node cung cấp cho NBIRTH hoặc DBIRTH. Lưu ý là không có 2 alias đại diện cho 1 EON node. Khi được định nghĩa trong NBIRTH hoặc DBIRTH, các message chỉ gửi cùng với alias thay vì một cái tên để giảm thiểu kích thước cho message.
- Timestamp: nghĩa là mốc thời gian mà message được published, timestamp có dạng là số nguyên 64bit không dấu tính bằng ms được tính từ mốc 1/1/1970.
- Data type: kiểu dữ liệu 32bit không dấu, theo bảng qui đổi như sau

Unknown	0	Double	10
Int 8	1	Boolean	11
Int 16	2	String	12
Int 32	3	DateTime	13

Int 64	4	Text	14
UInt 8	5	UUID	15
UInt 16	6	DataSet	16
UInt 32	7	Bytes	17
UInt 64	8	File	18
Float	9	Template	19

Bảng 4.1 Bảng quy đổi kiểu dữ liệu

- Value: là các giá trị mong muốn của người dùng

```
{
    "timestamp": 1486144502122,
    "metrics": [
        {
            "name": "My Metric",
            "alias": 1,
            "timestamp": 1479123452194,
            "dataType": "String",
            "value": "Test"
        }
    ],
    "seq": 2
}
```

Hình 4.6 Ví dụ về payload của Sparkplug

4.1.8. Truyền dữ liệu bằng SignalR

ASP.NET SignalR là một thư viện cho các lập trình viên Asp.Net đơn giản quá trình thêm chức năng gửi dữ liệu tốc độ cao trong phát triển ứng dụng.

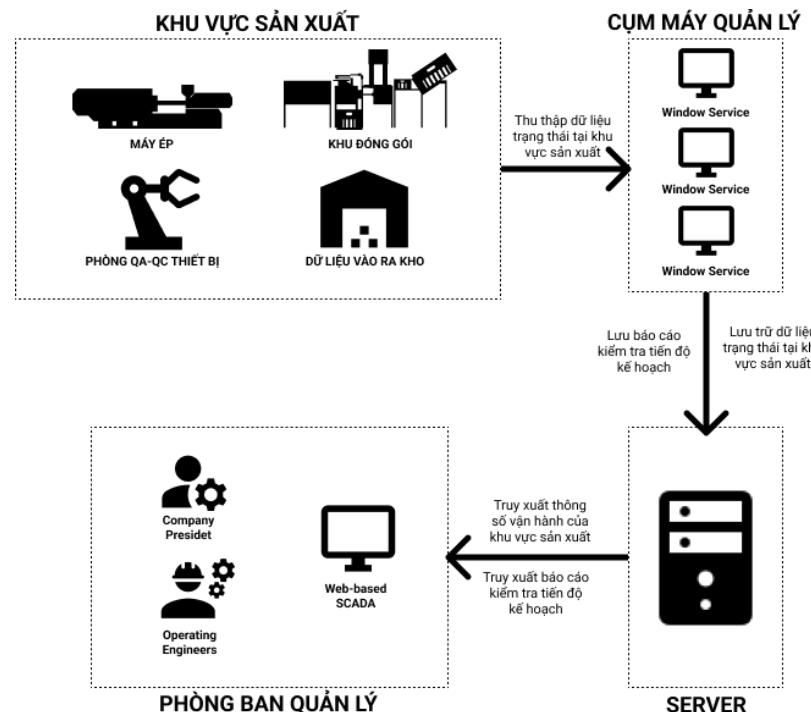
Đơn giản có thể kiểu nó sẽ có một Hub trên Server làm trung chuyển dữ liệu và client có thể publish dữ liệu và subscribe topic để có thể trao đổi dữ liệu với nhau - gần tương tự như với giao thức MQTT.

SignalR này được sử dụng để chờ bất kì ứng dụng web nào trên hệ thống mạng gọi tới để truy cập dữ liệu.

4.2. Tổng quan hệ thống

Dựa vào khó khăn hiện tại của công ty, nhóm em đề xuất xây dựng một hệ thống thu thập dữ liệu kết nối giữa phòng ban quản lý và khu vực sản xuất thông qua máy chủ. Hệ thống gồm các thành phần chính sau:

- Phần mềm ứng dụng web chạy trên máy tính: dùng để thu thập các dữ liệu. Ngoài ra, phần mềm còn có thể truy xuất các dữ liệu lịch sử cũ thông qua giao tiếp với máy chủ nhờ giao thức HTTP.
- Máy chủ (Server): dùng để lưu trữ và xử lý dữ liệu bao gồm database và một hoặc nhiều các ứng dụng web (bao gồm các API).
- Window Service: đóng vai trò như 1 node tại một khu vực có nhiệm vụ thu thập dữ liệu và lưu trữ tại CSDL của máy chủ công ty



Hình 4.7 Sơ đồ tổng quan hệ thống

Trong các phần trên, nhóm em đã thực hiện lập trình phần mềm web và ứng dụng chạy ngầm window service tại các máy tính quản lý khu vực. Trong đó, nhóm em chú trọng thiết kế giao diện đơn giản cho nhân viên, giao tiếp với

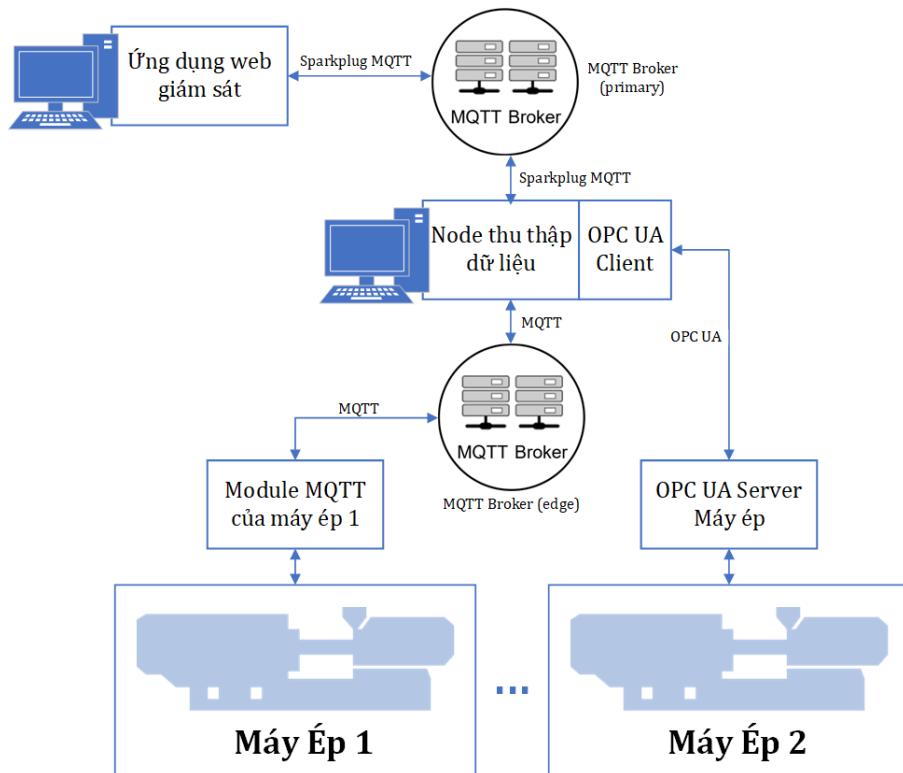
database tại máy tính, quản lý và truyền nhận dữ liệu với máy chủ, truy xuất dữ liệu thông qua giao tiếp với máy chủ nhờ giao thức HTTP và web socket SignalR.

4.3. Xây dựng các node thu thập dữ liệu

Trong mô hình trên các node sẽ đóng vai trò thu thập dữ liệu của từng cụm máy trong khu vực thông qua các giao thức, chuẩn truyền thông khác nhau như: MQTT, Socket TCP, OPC UA, v.v và chuyển tất cả thành một chuẩn chung duy nhất Sparkplug MQTT để gửi về máy chủ thu thập dữ liệu. Dự vào từng khu vực nhất định có đầu ra như thế nào mà node thu thập dữ liệu sẽ được thiết kế khác nhau.

Có thể xem các node thu thập dữ liệu là gateway để chuyển đổi chuẩn truyền thành Sparkplug MQTT, ngoài ra node cũng sẽ tạo ra một ngõ ra để ứng dụng máy tính tại từng cụm có thể truy cập đến và lấy dữ liệu mà node đọc được. Dữ liệu được gửi Sparkplug MQTT lên broker chính sẽ được máy chủ thu thập dữ liệu nhận lại và thực hiện lưu trữ, cho phép truy xuất từ ứng dụng web.

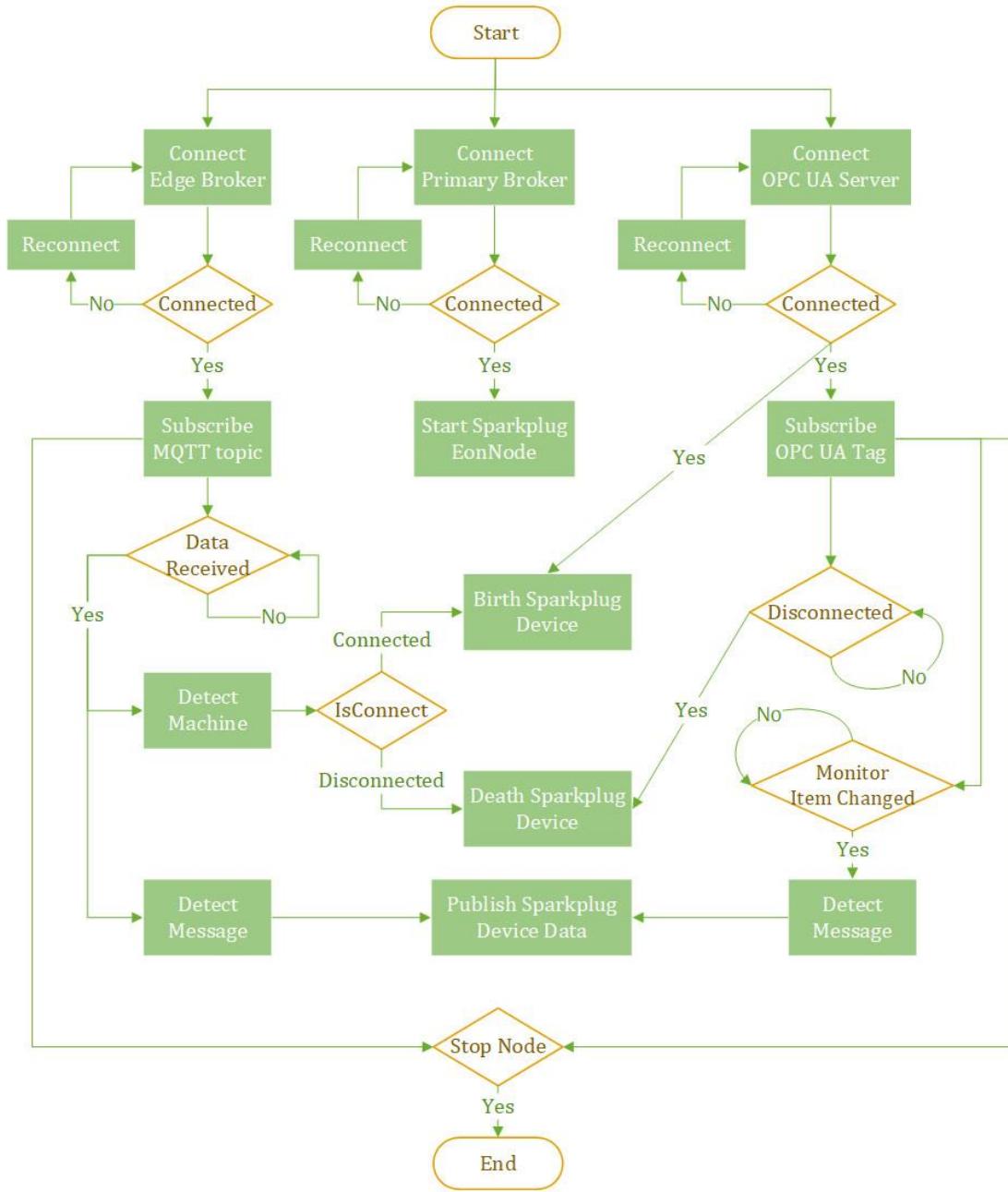
4.3.1. Thu thập dữ liệu máy ép



Hình 4.8 Mô hình hệ thống thu thập dữ liệu khu vực máy ép

Trong sơ đồ trên các module MQTT của máy ép là các thiết bị đã có sẵn trong máy ép. Có chức năng đọc các giá trị như: chu kỳ ép, thời gian mở cửa, số sản phẩm đã ép trong ca, ... Sau đó chuyển tất cả thành tin nhắn MQTT và gửi lên broker phụ tại khu vực máy ép.

Việc cần làm của node thu thập dữ liệu là đăng ký vào các topic mà module máy ép gửi lên để nhận được dữ liệu. Sau đó chuyển thành Sparkplug MQTT và gửi dữ liệu về máy chủ thu thập dữ liệu để lưu trữ.



Hình 4.9 Lưu đồ giải thuật node thu thập máy ép

Ở trong sơ đồ trên node thu thập dữ liệu hoạt động như sau:

- Trước khi khởi chạy chương trình node thu thập dữ liệu, phải đảm bảo rằng các thông số của MQTT Broker và OPC UA Server phải được khai báo đầy đủ và chính xác.

- Các thông số ở đây bao gồm: cấu hình kết nối đến MQTT Edge Broker để nhận dữ liệu từ module MQTT của máy ép, cấu hình OPC UA Server của controller máy ép để đọc dữ liệu, cấu hình kết nối đến MQTT Primary Broker để gửi dữ liệu về máy chủ thu thập dữ liệu.
- Khi chương trình bắt đầu khởi chạy → Thực hiện kết nối đến MQTT Primary Broker và MQTT Edge Broker. Nếu trong quá trình kết nối các lỗi về cấu hình thông số MQTT Broker sai sẽ được log ra màn hình. Nếu kết nối đến MQTT Broker bị gián đoạn trong quá trình hoạt động thì chương trình sẽ tự động thực hiện kết nối lại theo thông số “ReconnectInterval” đã được khai báo trong cấu hình. Thường thông số này sẽ là 30s để kết nối mạng được ổn định trước khi thực hiện kết nối lại. Thực hiện tương tự để kết nối đến OPC UA Server máy ép để đọc dữ liệu trong máy ép. Khi có mất kết nối sẽ thực hiện kết nối lại theo thời gian reconnect đã khai báo trước và khi kết nối mạng trở lại ổn định.
- Sau khi kết nối đến MQTT Edge Broker thành công thì chương trình sẽ thực hiện đăng ký vào các topic của máy ép để nhận dữ liệu. Và chờ tin nhắn đến trong sự kiện “MqttClient_ApplicationMessageReceived”. Trong sự kiện này sẽ xử lý kiểm tra tin nhắn đến từ máy ép nào. Nếu là tin nhắn báo máy ép được khởi động thì sẽ thực hiện khai sinh một device bằng cách gửi tin nhắn Sparkplug MQTT BirthDevice đến MQTT Broker Primary. Nếu là tin nhắn báo máy ép tắt thì sẽ thực hiện khai tử một device bằng cách gửi tin nhắn Sparkplug MQTT DeathDevice đến MQTT Broker Primary. Nếu là tin nhắn dữ liệu máy ép thì sẽ thực hiện gửi tin nhắn Sparkplug MQTT DataDevice đến MQTT Broker Primary.
- Sau khi kết nối thành công đến OPC UA Server chương trình sẽ thực hiện gửi tin nhắn Sparkplug MQTT BirthDevice đến MQTT Broker Primary. Đồng thời đăng ký đến các item trong OPC UA Server để nhận sự kiện “MonitorItem DataChanged”. Trong sự kiện này sẽ xử lý thông báo thay

đổi của các Item → Và thực hiện xử lý các thông tin trong thông báo “Item Notification” và chuyển thành các tin nhắn Sparkplug MQTT và gửi đến MQTT Broker Primary. Nếu có sự kiện mất kết nối thì sẽ gửi tin nhắn Sparkplug MQTT DataDevice đến MQTT Broker Primary.

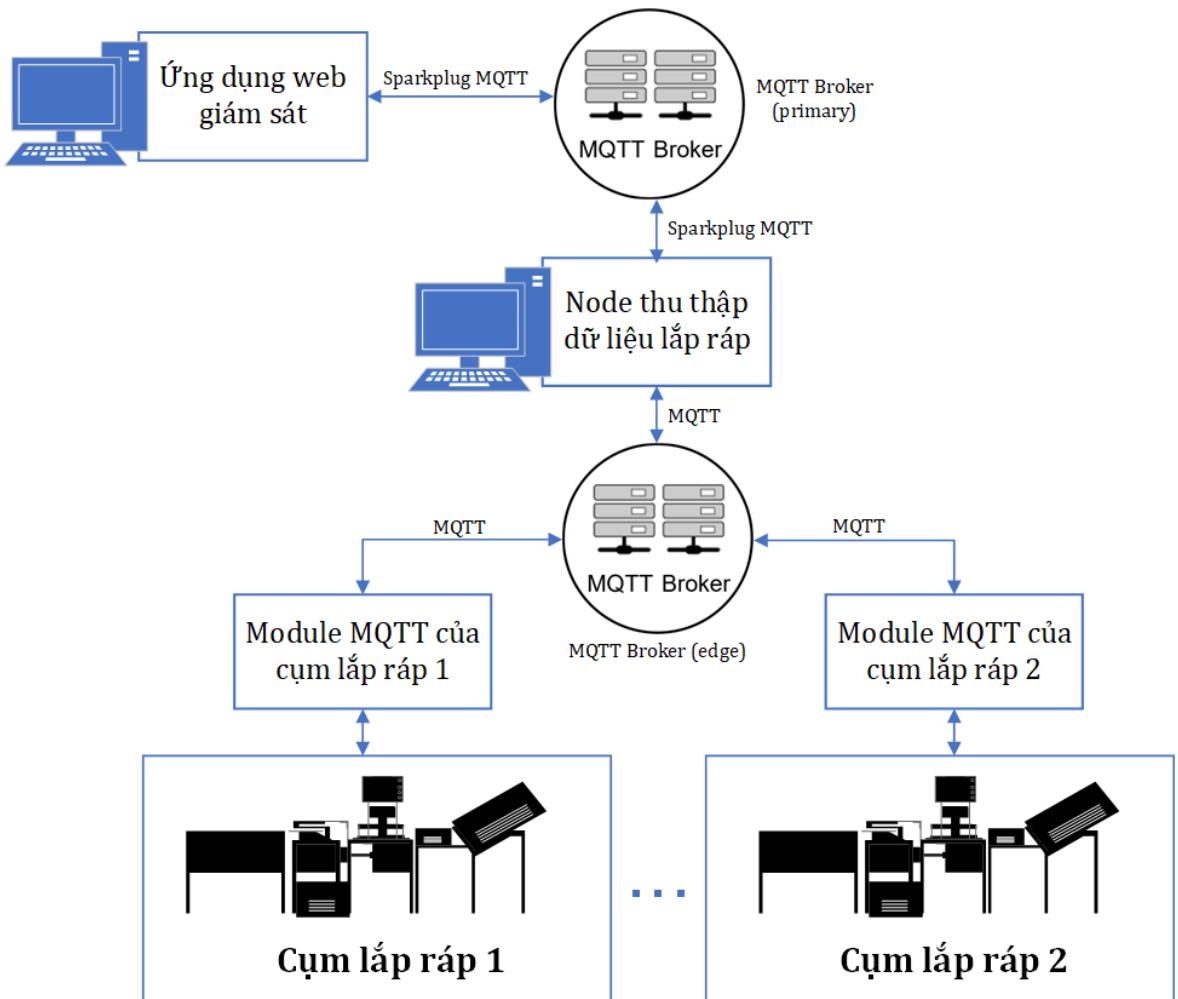
- Trong thiết kế này node thu thập dữ liệu đóng vai trò như một Sparkplug MQTT EonNode để chuyển các thiết bị không có Sparkplug thành các Sparkplug Device.

Tên topic	Mục đích	Định dạng
IMM/L6/CycleMessage	ESP32 Publish dữ liệu về chu kỳ ép	{"Timestamp":"05-09-2018 00:10:00", "CycleTime":35.56, "OpenTime":4.52, "Mode":1, "CounterShot":34, "MoldID": "NX35", "ProductId":"HA40T", "SetCycle":35}
IMM/L6/MachineStatus	ESP32 Publish dữ liệu về trạng thái của máy	{"Timestamp":"05/09/2018 00:10:00", "MachineStatus":4}
IMM/L6/LWT	Topic LWT khi ESP32 kết nối đến broker, khi ESP32 mất kết nối, broker sẽ tự gửi tin nhắn mất kết nối vào topic này	"Lost connect to device"
IMM/L6/DAMess	Desktop app gửi	{"Timestamp":"05/09/2018

	tin nhắn như thay khuôn, reboot cho ESP32	00:10:00", "Command":2}
IMM/L6/ConfigMess	Desktop app gửi tin nhắn config về sản phẩm ép hiện tại, chu kỳ tiêu chuẩn	{"Timestamp":"05/09/2018 00:10:00", "MoldId":"HA40", "ProductId":"HA40T", "CycleTime":35}
IMM/L6/SyncTime	Topic để Desktop app gửi thời gian cho ESP32 để đồng bộ về mặt thời gian thực	{"Year":2022, "Month":4, "Day":12, "Hour":14, "Min":21, "Sec":12}
IMM/L6/Feedback	Topic để ESP32 gửi tin nhắn Feedback khi có lỗi thẻ nhớ, lỗi mô đun thời gian thực	{"Mess":4}

Bảng 4.2 Bảng dữ liệu máy ép

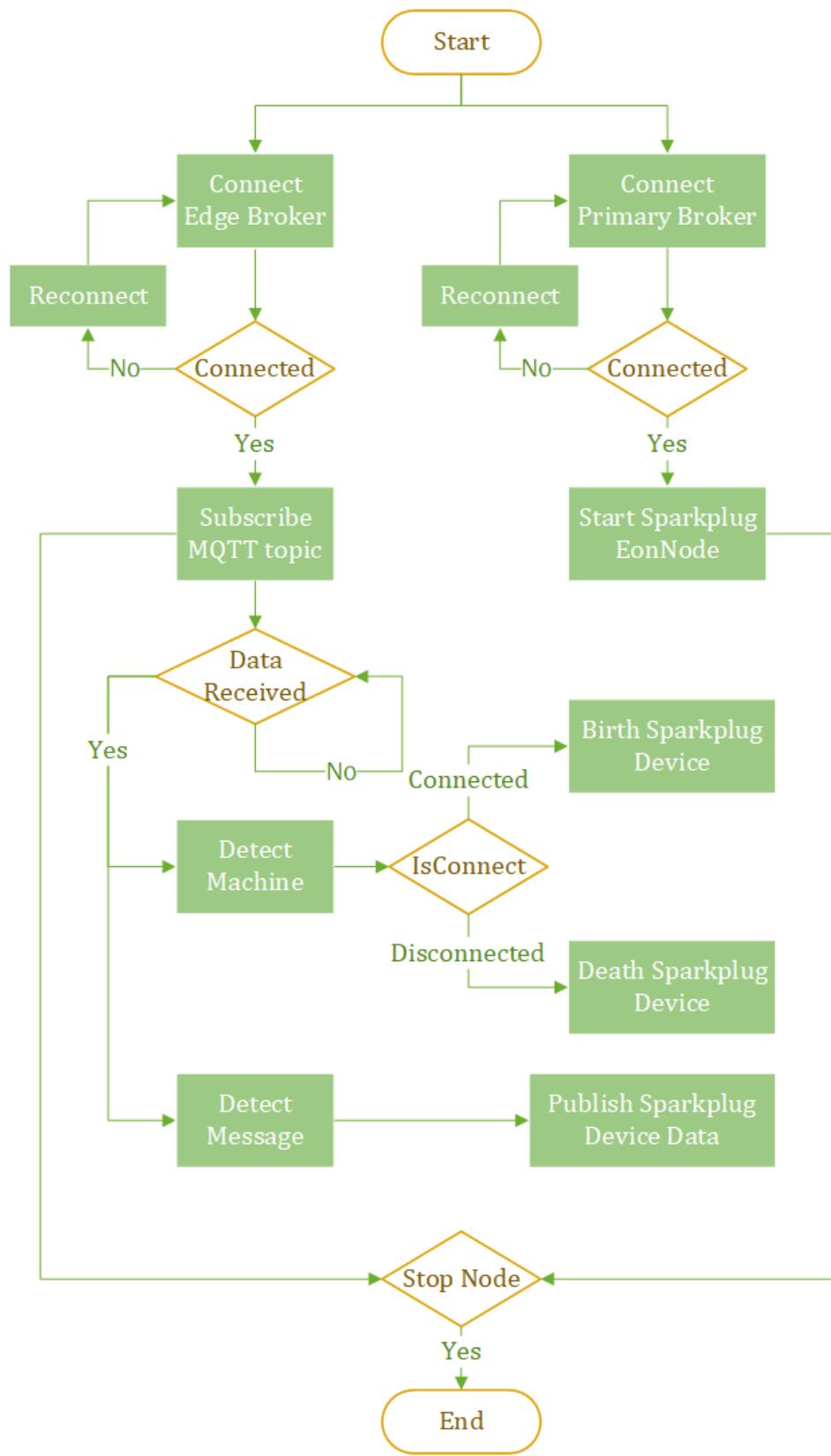
4.3.2. Thu thập cụm dữ liệu lắp ráp



Hình 4.10 Mô hình hệ thống thu thập dữ liệu khu vực lắp ráp

Trong mô hình trên dữ liệu sản phẩm lắp ráp sẽ được đếm và gửi lên MQTT Edge Broker tại khu vực máy ép, thông qua các module ESP32 có sẵn và được vận hành tại nhà máy.

Việc cần làm của node thu thập dữ liệu là đăng ký vào các topic mà module ESP32 tại khu lắp ráp gửi lên để nhận được dữ liệu đếm. Sau đó chuyển thành Sparkplug MQTT và gửi dữ liệu về máy chủ thu thập dữ liệu để lưu trữ.



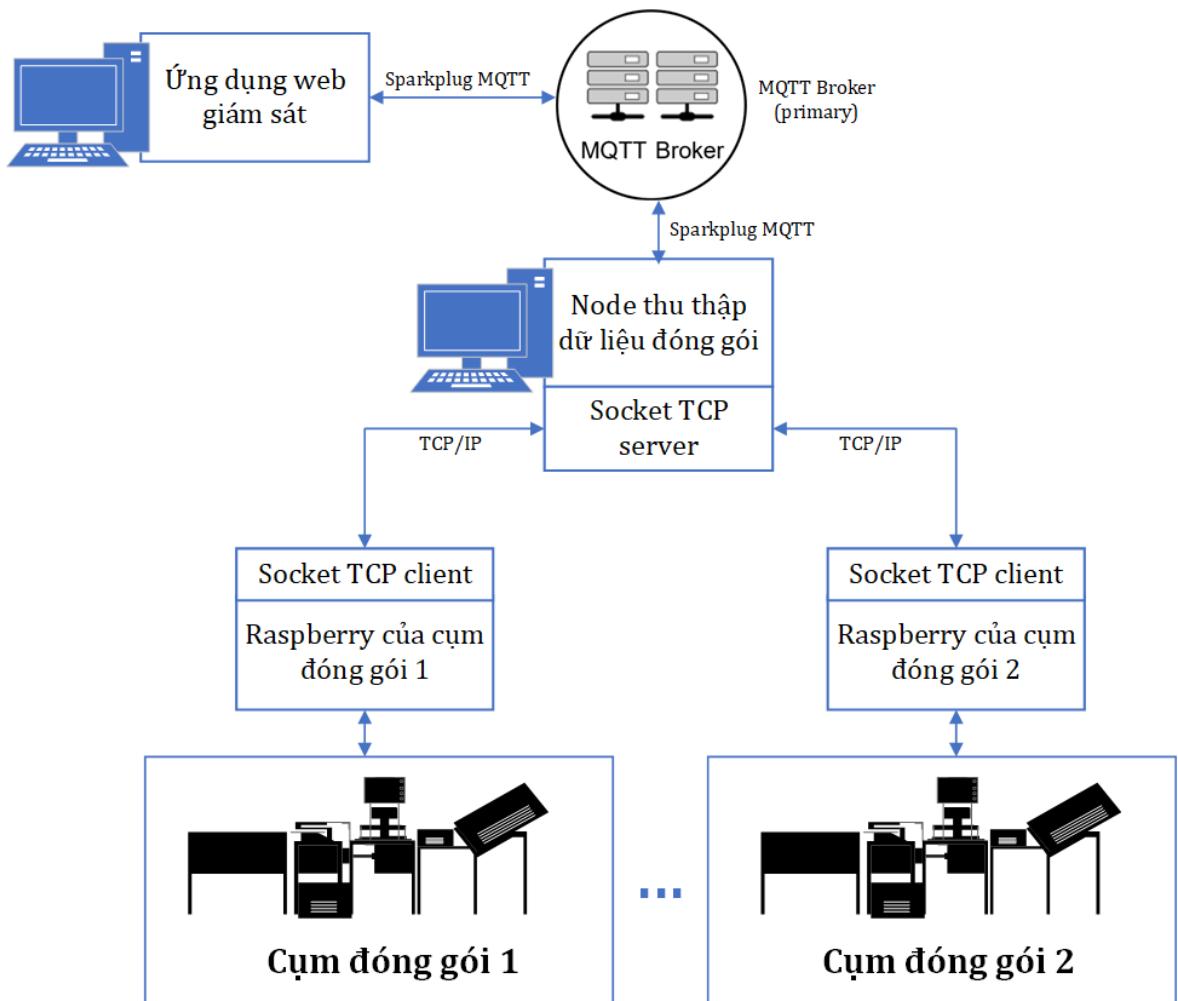
Hình 4.11 Lưu đồ giải thuật node thu thập cụm lắp ráp

Ở trong sơ đồ trên node thu thập dữ liệu hoạt động như sau:

- Trước khi khởi chạy chương trình node thu thập dữ liệu, phải đảm bảo rằng các thông số của MQTT Broker phải được khai báo đầy đủ và chính xác.
- Các thông số ở đây bao gồm: cấu hình kết nối đến MQTT Edge Broker để nhận dữ liệu từ module ESP32 của cụm lắp ráp, cấu hình kết nối đến MQTT Primary Broker để gửi dữ liệu về máy chủ thu thập dữ liệu.
- Khi chương trình bắt đầu khởi chạy → Thực hiện kết nối đến MQTT Primary Broker và MQTT Edge Broker. Nếu trong quá trình kết nối các lỗi về cấu hình thông số MQTT Broker sai sẽ được log ra màn hình. Nếu kết nối đến MQTT Broker bị gián đoạn trong quá trình hoạt động thì chương trình sẽ tự động thực hiện kết nối lại theo thông số “ReconnectInterval” đã được khai báo trong cấu hình. Thường thông số này sẽ là 30s để kết nối mạng được ổn định trước khi thực hiện kết nối lại.
- Sau khi kết nối đến MQTT Edge Broker thành công thì chương trình sẽ thực hiện đăng ký vào các topic của cụm lắp ráp để nhận dữ liệu. Và chờ tin nhắn đến trong sự kiện “MqttClient_ApplicationMessageReceived”. Trong sự kiện này sẽ xử lý kiểm tra tin nhắn đến từ cụm lắp ráp nào. Nếu là tin nhắn báo cụm lắp ráp đang hoạt động thì sẽ thực hiện khai sinh một device bằng cách gửi tin nhắn Sparkplug MQTT BirthDevice đến MQTT Broker Primary. Nếu là tin nhắn báo cụm lắp ráp đã dừng hoặc nghỉ thì sẽ thực hiện khai tử một device bằng cách gửi tin nhắn Sparkplug MQTT DeathDevice đến MQTT Broker Primary. Nếu là tin nhắn dữ liệu đếm của cụm lắp ráp thì sẽ thực hiện gửi tin nhắn Sparkplug MQTT DataDevice đến MQTT Broker Primary.

Trong thiết kế này node thu thập dữ liệu đóng vai trò như một Sparkplug MQTT EonNode để chuyển các thiết bị ESP32 chưa được lập trình để hỗ trợ Sparkplug thành các Sparkplug Device.

4.3.3. Thu thập dữ liệu cụm đóng gói

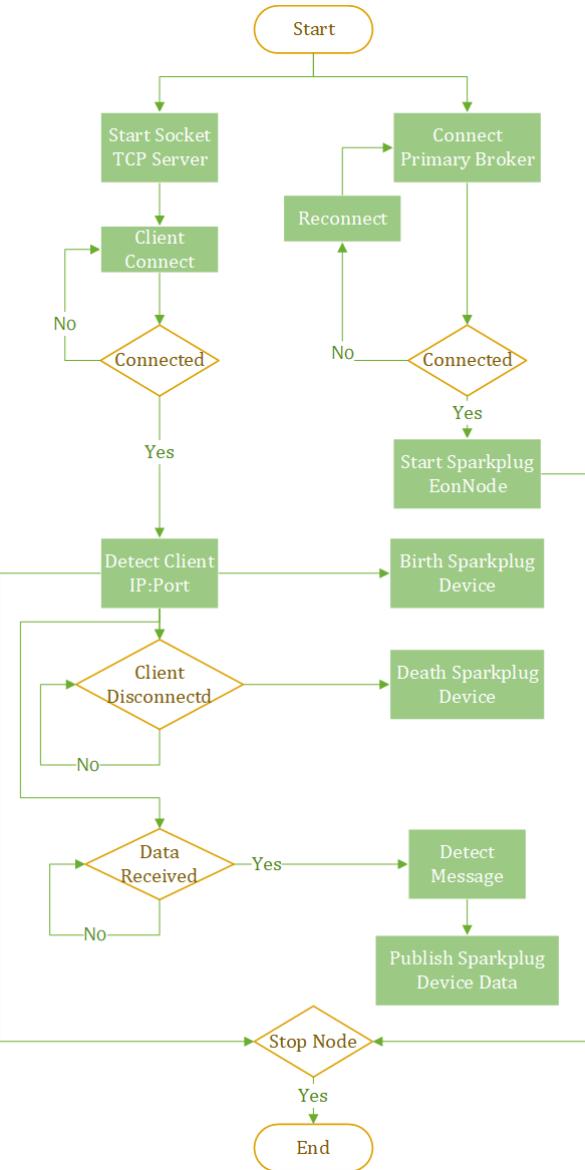


Hình 4.12 Mô hình hệ thống thu thập dữ liệu khu vực đóng gói

Trong mô hình trên dữ liệu thu thập của cụm đóng gói sẽ được đếm và xử lý thông qua các module raspberry. Các dữ liệu này sẽ bao gồm: trạng thái máy, số sản phẩm đóng gói thành công, số sản phẩm đóng gói sai.

Các module raspberry này đã có sẵn và được lập trình trước tại nhà máy. Việc giao tiếp với raspberry ta sử dụng socket để truyền nhận dữ liệu, đảm bảo tốc độ truyền dữ liệu nhanh chóng.

Node thu thập dữ liệu tại cụm đóng gói sẽ có nhiệm vụ. Làm một socket TCP server để nhận các thông tin từ các raspberry đóng vai trò là các socket TCP client. Sau đó chuyển các dữ liệu này thành tin nhắn Sparkplug MQTT và gửi về máy chủ để lưu trữ.



Hình 4.13 Lưu đồ giải thuật node thu thập cụm đóng gói

Ở trong sơ đồ trên node thu thập dữ liệu cụm đóng gói hoạt động như sau:

- Trước khi khởi chạy chương trình node thu thập dữ liệu. Phải đảm bảo tất cả các cấu hình cài đặt đã đúng. Như cấu hình kết nối đến MQTT

Primary Broker để gửi tin nhắn về máy chủ thu thập dữ liệu. Và cấu hình Socket TCP Server để nhận dữ liệu đếm từ raspberry của cụm đóng gói, đảm bảo rằng đã mở port cho Socket TCP Server tránh trường hợp lỗi do không kết nối đến được server.

- Khi chương trình bắt đầu khởi chạy → Thực hiện kết nối đến MQTT Primary Broker và khởi tạo một Socket TCP Server.
- Nếu trong quá trình kết nối với MQTT Primary Broker xảy ra lỗi về cấu hình thông số MQTT Broker thì chương trình sẽ thực hiện log ra console lỗi để thông báo. Nếu kết nối giữa MQTT Client và MQTT Broker bị gián đoạn thì chương trình sẽ thực hiện kết nối lại dựa trên thông số “ReconnectInterval” đã được khai báo trong cấu hình. Giá trị này thường được để là 30s để kết nối mạng ổn định lại trước thi thực hiện kết nối lại.
- Nếu trong quá trình khởi chạy Socket Server mà chương trình xảy ra lỗi thì cũng sẽ được log ra console để thông báo. Nếu Socket client bị mất kết nối với Socket server thì Socket client sẽ tự phát hiện và cố gắng kết nối lại.
- Sau khi Socket client kết nối đến Socket server → Chương trình sẽ nhận được sự kiện “ClientConnected” → Trong sự kiện này thì thực hiện quản lý các Socket client bằng một “Dictionary<Tkey, TValue>” để quản lý ID và IP:Port chung với nhau trên một biến duy nhất → Nếu có client kết nối tới thì thêm một biến <ID, IP:Port> vào trong Dictionary → Và gửi một tin nhắn Sparkplug MQTT BirthDevice để khai sinh một device.
- Khi Socket Client ngắt kết nối đến Socket client → Chương trình sẽ nhận được sự kiện “ClientDisconnected” → Xoá biến có IP:Port bằng với giá trị IP:Port của client vừa ngắt kết nối khỏi Dictionary → Và gửi một tin nhắn Sparkplug MQTT DeathDevice để khai tử một device.

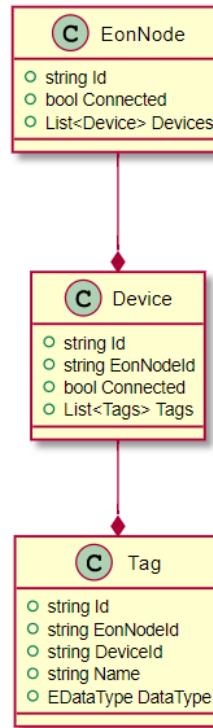
Khi có dữ liệu gửi lên từ Socket client → Chương trình sẽ nhận được sự kiện “DataReceived” trong sự kiện này sẽ chứa dữ liệu gửi lên từ Socket Client → Trong sự kiện này chương trình sẽ thực hiện đọc và phân tách dữ liệu được gửi lên từ Socket client → Sau đó chuyển các dữ liệu này thành các tin nhắn Sparkplug MQTT DeviceData và gửi về MQTT Primary Broker để máy chủ thu thập dữ liệu lưu trữ lại.

4.4. Thiết kế hệ thống lưu trữ dữ liệu

4.4.1. Thiết kế cấu trúc dữ liệu quản lý tag

Trước khi xây dựng chương trình thu thập và lưu trữ dữ liệu, thiết kế cấu trúc dữ liệu quản lý tag là điều cần làm trước tiên. Cấu trúc dữ liệu là cách lưu trữ, truyền nhận và tổ chức dữ liệu theo một hệ thống để đạt được hiệu quả sử dụng nhất định. Sử dụng ngôn ngữ UML để mô hình hóa các cấu trúc dữ liệu sao cho trực quan và dễ nhìn đối với người lập trình.

Dựa trên ý tưởng và tham khảo các hệ dữ liệu quản lý khác, nhóm em lên ý tưởng cấu trúc dữ liệu như sau:



Hình 4.14 Cấu trúc dữ liệu quản lý tag

Như cấu trúc trên, cơ sở dữ liệu quản lý tag sẽ được chia làm 3 phần. Đầu tiên là bảng quản lý các EonNode, mục tiêu là lưu trữ các thông tin chi tiết liên quan đến EonNode như: id node, trạng thái kết nối, và danh sách các Device thuộc quản lý của EonNode đó.

Tầng tiếp theo là bảng lưu trữ thông tin của các Device, mục tiêu là lưu trữ các thông tin chi tiết liên quan đến Device như: id device, id eonnode mà device thuộc về, trạng thái kết nối, và danh sách các Tag thuộc vào quản lý của Device đó.

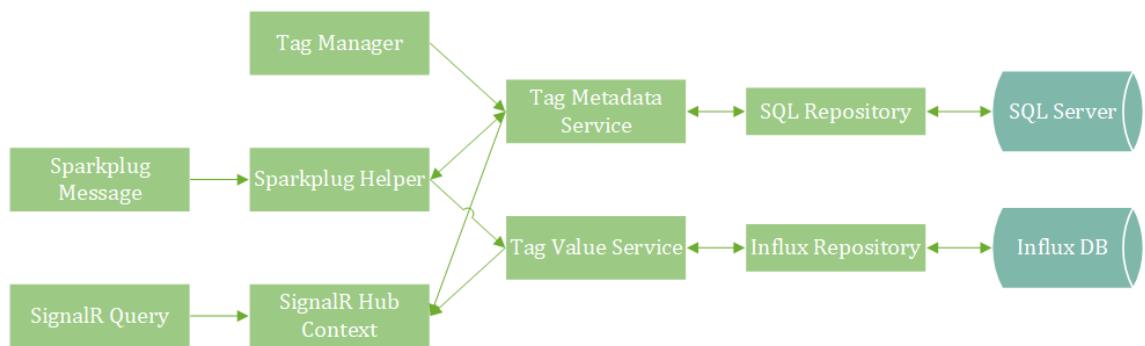
Tầng cuối cùng trong cấu trúc là bảng lưu trữ thông tin của các Tag, mục tiêu là lưu trữ các thông tin chi tiết liên quan đến Tag như: id tag, tên tag, id device và id eonnode mà tag thuộc về, kiểu dữ liệu.

Việc chia cấu trúc ra nhiều tần như vẽ giúp dễ dàng trong việc quản lý các tag và phát triển hệ thống rộng lớn trong tương lai. Tóm lại, cấu trúc dữ liệu quản lý tag tổng sẽ gồm các dữ liệu theo từng mảng. Các mảng sẽ được viết

trong các class. Khai báo thuộc tính và kiểu dữ liệu của các thuộc tính đó trong từng class.

4.4.2. Xây dựng chương trình thu thập và lưu trữ dữ liệu

Mục tiêu cơ bản của chương trình thu thập và lưu trữ dữ liệu là nhận các tin nhắn MQTT Sparkplug từ các Node. Sau đó thực hiện truy xuất đến cơ sở dữ liệu quản lý tag để lấy thông tin EonNode, Device, Tag sẽ dùng các thông tin đó để lưu trữ dữ liệu thu thập được vào cơ sở dữ liệu chuỗi thời gian InfluxDB.



Hình 4.15 Cấu trúc chương trình của máy chủ

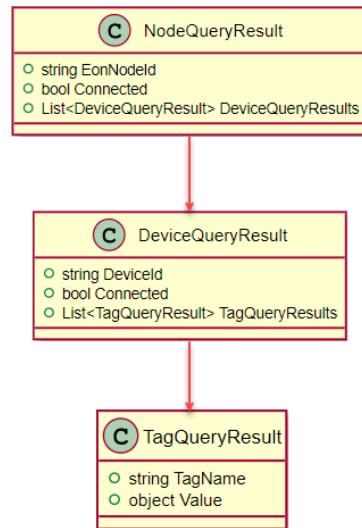
Theo sơ đồ trên, các thông tin về EonNode, Device, Tag như id, danh sách các thành phần con, tên sẽ được quản lý ở mục Tag Manager và lưu trữ vào cơ sở dữ liệu lưu trữ lâu dài là SQL Server. Các tác vụ của mục Manager này bao gồm thêm, xoá, sửa, lấy ra. Tầng này chỉ đảm nhận nhiệm vụ tiếp nhận và xử lý thông tin. Việc thực hiện lưu trữ vào cơ sở dữ liệu sẽ được thực hiện trong tầng kế tiếp của sơ đồ hệ thống.

Nhận dữ liệu:

- Việc quản lý các thông tin được nhận về từ MQTT Primary Broker sẽ được nhận và xử lý tại Sparkplug Helper. Tại đây sẽ thực hiện cập nhật các dữ liệu liên quan đến EonNode và Device. Thực hiện nhận các giá trị dữ liệu được gửi về từ các node thu thập dữ liệu → Sau đó sẽ dùng tên của Tag “TagName” để gọi đến tầng tiếp theo và truy xuất đến cơ sở dữ

liệu SQL Server để lấy thông tin của Tag đó ra → Tiếp đến sẽ thực hiện tạo các điểm lưu theo thời gian là “PointData” trong InfluxDB → Gọi đến tầng tiếp theo để thực hiện lưu trữ dữ liệu vào InfluxDB.

Đọc dữ liệu:



Hình 4.16 Cấu trúc dữ liệu cho chương trình đọc dữ liệu InfluxDB

- SignalR Hub sẽ dùng để nhận các yêu cầu truy xuất dữ liệu từ ứng dụng web. SignalR hỗ trợ cho việc truyền dữ liệu đến ứng dụng web một cách nhanh nhất độ trễ thấp. Trong nhiều dự án web-based của các công ty khác trên thế giới cũng sử dụng SignalR để gửi dữ liệu giám sát đến ứng dụng web một cách nhanh nhất.
- Trong “Hub Context” sẽ để ra các đầu hàm để ứng dụng phía client có thể gọi đến và thực hiện hàm đó → Sau khi hàm được gọi đến chương trình sẽ phân tách thành các “TagName” cần lấy → Sau đó sẽ dùng các “TagName” đó để truy xuất đến tầng tiếp theo để truy xuất SQL Server và lấy thông tin đầy đủ của Tag → Và thực hiện gọi đến tầng tiếp theo bằng các thông tin đầy đủ của Tag có được và truy xuất vào InfluxDB để lấy dữ liệu ra và trả về cho ứng dụng web.

SQL Repository:

- Trong SQL Repository sẽ thực hiện các thao tác lên cơ sở dữ liệu SQL Server. Có nhiều cách để thao tác lên cơ sở dữ liệu SQL Server như SQL, sử dụng thông qua các thư viện như Entity Framework, Dapper. v.v Nhóm em chọn sử dụng Entity Framework để thực hiện thao tác lên cơ sở dữ liệu do tính thông dụng của Entity Framework và tốc độ truy xuất dữ liệu của Entity Framework Core đã là khá cao.
- Thao tác đến với cơ sở dữ liệu sẽ được thực hiện thông qua việc gọi đến một đối tượng gián tiếp trong chương trình đó là “DbContext”. Các bảng, các đối tượng trong cơ sở dữ liệu sẽ được biểu diễn bằng các class, các DbSet trong chương trình.

Influx Repository:

- Influx Repository sẽ thực hiện thao tác với cơ sở dữ liệu InfluxDB. Cơ sở dữ liệu chuỗi thời gian InfluxDB sẽ được dùng để lưu trữ các giá trị của Tag theo dòng thời gian. Ưu điểm lớn nhất của InfluxDB là tốc độ ghi rất nhanh và độ chính xác thời gian khi ghi cao. Bên cạnh đó InfluxDB cũng có ưu điểm là không chiếm nhiều không gian bộ nhớ do có khả năng lưu trữ một lượng dữ liệu lớn mà chỉ tốn một lượng nhỏ dung lượng ổ cứng. Các giá trị này là các giá trị dữ liệu và sẽ có thể xoá đi theo chu kỳ cố định thường là 3 tháng.
- Thao tác đến với InfluxDB sẽ được thực hiện thông qua class “InfluxDbCore”. Trong class này sẽ thực hiện lấy các API được cung cấp sẵn bởi Influx và tạo “DataPoint” để ghi dữ liệu vào cơ sở dữ liệu. Hoặc tạo ra các “Flux Query” bằng ngôn ngữ Flux để thực hiện đọc dữ liệu từ cơ sở dữ liệu.

Sau khi hoàn thành hết các phần của chương trình thì sẽ thực hiện ghép lại và chạy thử chương trình. Nhằm tìm ra các lỗi, không tương thích của chương trình và khắc phục.

Chương 5. PHẦN MỀM ỨNG DỤNG WEB – WEB APPLICATION

5.1. Các công nghệ, ngôn ngữ và phần mềm sử dụng

5.1.1. Công nghệ

Với sự phát triển của ứng dụng web, các ứng dụng Desktop đang dần trở nên lạc hậu. Nay là thời đại của những trang web thân thiện, dễ dùng, dễ dàng bảo trì, không bị phụ thuộc vào bất cứ phần cứng nào. Tuy vậy thì phần ứng dụng web đang dần được tối ưu hóa hiệu năng sử dụng. Trong đề tài này sử dụng thư viện ReactJS với công nghệ SPA.

Với SPA, chúng ta sẽ được tiếp cận một cụm từ khá mới là Client Side Rendering khi Client mới là người render ra từ những dòng HTML, CSS để hiển thị. Nhờ đó mà chúng ta cũng tách biệt được giữ những công việc mà Server phải làm khi lượng truy cập Internet ngày càng đông, khiến Server phải chịu tải rất nặng khi vừa phải xử lý dữ liệu lại phải render ra để trả lại cho người dùng.

5.1.2. Thư viện ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được tạo ra bởi Facebook giúp lập trình viên tạo ra được các ứng dụng đơn trang hoạt động với tốc độ nhanh chóng, mượt mà, dễ mở rộng và nâng cấp.

Những ưu điểm nổi bật của ReactJS:

Web React hoạt động với tốc độ và hiệu suất vượt trội hơn hẳn so với các loại khác, bởi vì React có thể tạo ra cho chính bản thân nó một DOM ảo, đây là nơi mà các component được tồn tại trên đó. Việc tạo ra DOM như vậy giúp cải thiện hiệu suất làm việc rất nhiều, khi có tính toán cần thay đổi hoặc cần cập nhật những gì lên DOM thì ReactJS đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên DOM, làm như vậy sẽ giúp cho ReactJS tránh được những thao tác không cần thiết gây giảm hiệu suất ứng dụng.

Lợi ích thứ hai mà ReactJS đem lại đó chính là việc viết các đoạn code JS sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX nghĩa là cú pháp này cho phép ta trộn được giữa code HTML và JavaScript nên ta có thể xây dựng phần khung HTML và logic hoạt động cho một component chung trong một file JavaScript, giúp dễ dàng hơn trong việc phát triển và sửa lỗi.

Cộng đồng phát triển rộng lớn nên mọi thắc mắc đều dễ dàng có được sự giải đáp và giúp đỡ. Hơn nữa, ngày càng có thêm nhiều công cụ và giải pháp mới hiệu quả hơn cho việc lập trình ReactJS bởi sự đóng góp của các nhà phát triển trong cộng đồng.

Tóm lại, với sự ưu việt về hiệu năng, sự dễ dàng trong việc xây dựng và phát triển của mình, cộng đồng rộng lớn cùng với sự “chống lưng” của ông trùm công nghệ Facebook, ReactJS đã, đang và sẽ là một trong những thư viện mạnh mẽ và phổ biến nhất để lập trình web. Trong đề tài này, nhóm sử dụng thư viện ReactJS phiên bản 17.0.2, phiên bản ổn định cuối cùng của thư viện trước khi Facebook triển khai phiên bản 18.0.1 gần đây.

5.1.3. Ngôn ngữ lập trình

Về ngôn ngữ lập trình, JavaScript là ngôn ngữ lập trình thông dịch được phát triển từ các khái niệm dữ liệu động. Ngôn ngữ này được dùng rộng rãi cho việc xây dựng các trang web người dùng cũng như phía máy chủ. JavaScript có cú pháp tương tự C, .js là phần mở rộng thường được dùng cho JavaScript. JavaScript cũng có rất nhiều thư viện hỗ trợ khác nhau với nhiều tính năng thuận tiện cho việc phát triển ứng dụng, có thể kể đến như thư viện chính.

Điểm nổi bật khi sử dụng JavaScript:

- Dễ tiếp cận hơn so với các ngôn ngữ lập trình khác.
- Dễ dàng phát hiện và khắc phục, sửa chữa lỗi.

- Hoạt động được trên nhiều nền tảng trình duyệt.
- Hỗ trợ web tương tác hiệu quả với người sử dụng.
- Tiết kiệm lưu lượng tải dữ liệu cho máy chủ (server) do người dùng sẽ xác nhận Input (đầu vào) rồi mới gửi web đến máy chủ.
- Ngôn ngữ lập trình JavaScript cho phép tạo nhiều giao diện khác nhau.
- Phản hồi tức thời đến người truy cập.

Tuy nhiên cũng phải kể đến một số nhược điểm:

- Dễ bị truy cập vào các storage trong web.
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng.
- Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

5.1.4. Phần mềm sử dụng

5.1.4.1. Visual Studio Code

Visual Studio Code (VSC) đã quá nổi tiếng về sự linh hoạt và mức độ tiện lợi khi phát triển web, ứng dụng. Phần mềm này không quá nặng nhưng đạt hiệu suất cực kì cao, extension cực kì nhiều giúp chạy được nhiều framework, gọi được các thiết bị ảo ví dụ như Android Virtual Device (AVD) của Android Studio. Tính năng debug của VSC cũng cực kì dễ sử dụng và hiển thị được các giá trị biến rõ ràng trong quá trình debug.

Visual Studio Code chỉ cần cài thêm 1 vài Extensions hỗ trợ là có thể thoải mái xây dựng ứng dụng. ReactJS cũng tích hợp tính năng gọi các web hot reload để phục vụ mục đích phát triển. Hoạt động build app cho đề tài này chủ yếu sử dụng trên VSC.

5.1.4.2. Figma

Phần mềm chuyên dụng để thiết kế UX/UI cho các ứng dụng (kể cả ứng dụng di động, ứng dụng máy tính để bàn và web). Phần mềm hỗ trợ cộng đồng thư viện lớn (community) giúp giảm thời gian tìm kiếm các tài nguyên thiết kế. Bản thiết kế của Figma có thể xem được trên phần mềm, web, thậm chí là xuất ra dạng PDF để có thể trình bày với cấp trên để thống nhất ý tưởng về mặt giao diện. Ngoài ra Figma là một hệ thống phần mềm đa nền tảng, có thể sử dụng trên Android, iOS, Windows và MacOS.

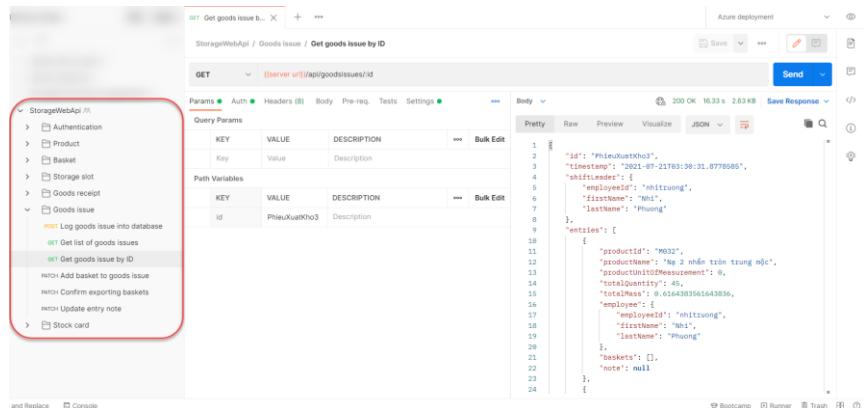
Figma trong đề tài này sử dụng với mục đích thiết kế giao diện, chủ đề cho Web App. Sau đó trình bày với giảng viên hướng dẫn và các anh chị bên phía công ty để thống nhất về cách hiển thị, các tính năng có trên ứng dụng sẽ được thể hiện như thế nào, tương tác của nhân viên đối với các sự kiện...

5.1.4.3. Postman

Là một phần mềm để tương tác với RESTful API, giúp việc thử nghiệm các giao tiếp với máy chủ mà không cần code. Postman hỗ trợ tất cả các phương thức HTTP như GET, POST, PATCH, PUT... Các thông số có thể truyền dễ dàng vào các endpoint thông qua bảng cài đặt.

Do sử dụng các máy chủ free để test, nên việc hết hạn máy chủ xảy ra thường xuyên, hoặc việc test trên máy chủ local thì phải đổi đường dẫn thường xuyên. Postman dễ dàng chuyển đổi đường link này thông qua các local environment.

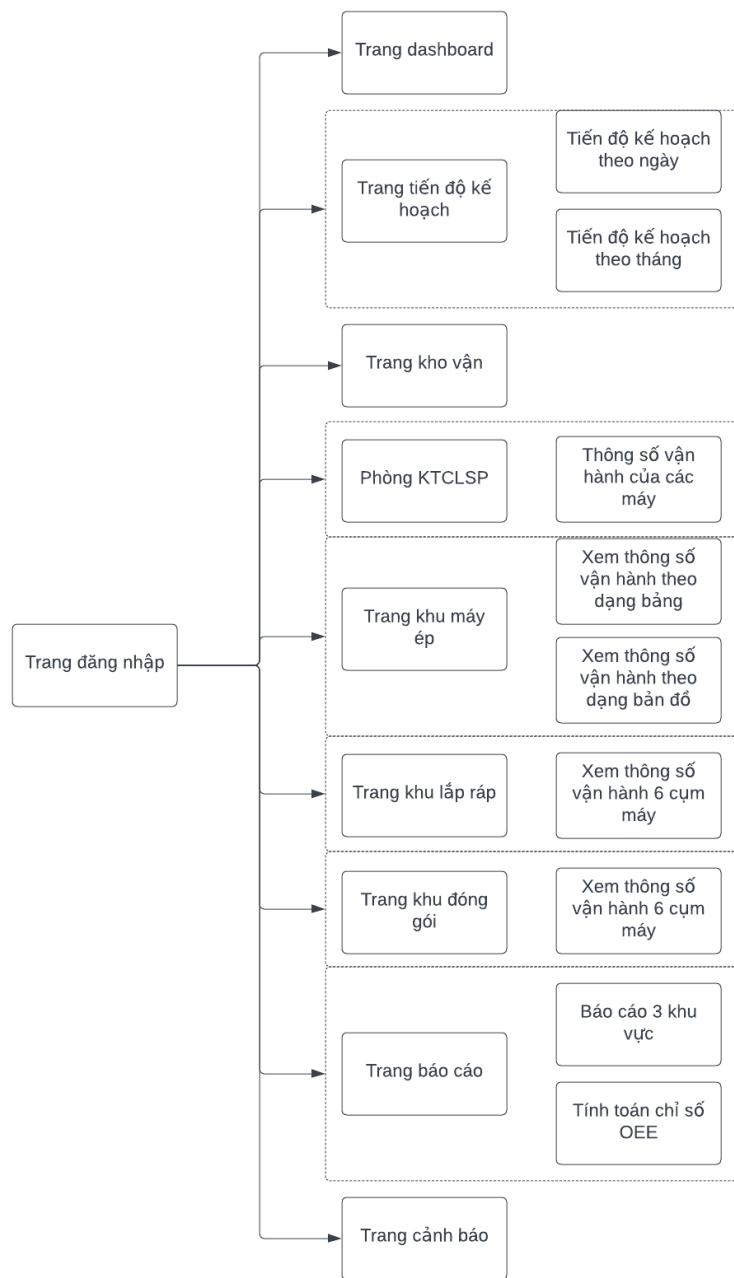
Postman có hỗ trợ các workspace nhóm người, có thể thêm tối đa 3 người (đối với bản free) và không giới hạn với bản trả phí. Vì thế được sử dụng trong đề tài này để tạo nên một hệ ứng dụng: backend, desktop app, mobile app. Quá trình phát triển thêm một tính năng sẽ trải qua giai đoạn test trên Postman trước khi bắt tay vào viết service trên các app client.



Hình 5.1 Giao diện phần mềm Postman

5.2. Giao diện thiết kế và giải thuật lập trình

Bước đầu tiên của xây dựng một ứng dụng không phải là lập trình, mà phải là nghiên cứu cấu trúc dữ liệu, đưa ra thiết kế giao diện và giải thuật cho việc hiển thị, xử lý dữ liệu trên giao diện đó.



Hình 5.2 Sơ đồ các trang của ứng dụng web giám sát và thu thập dữ liệu

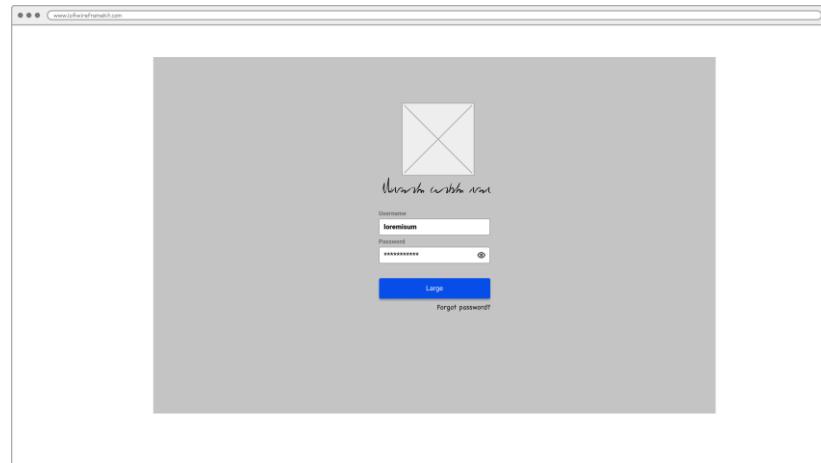
Để có thể chuyển đổi giữa các nhóm trang một cách nhanh chóng, cần có một phần tử chuyển trang, trong đề tài này nhóm sử dụng Sidebar như hình bên dưới.



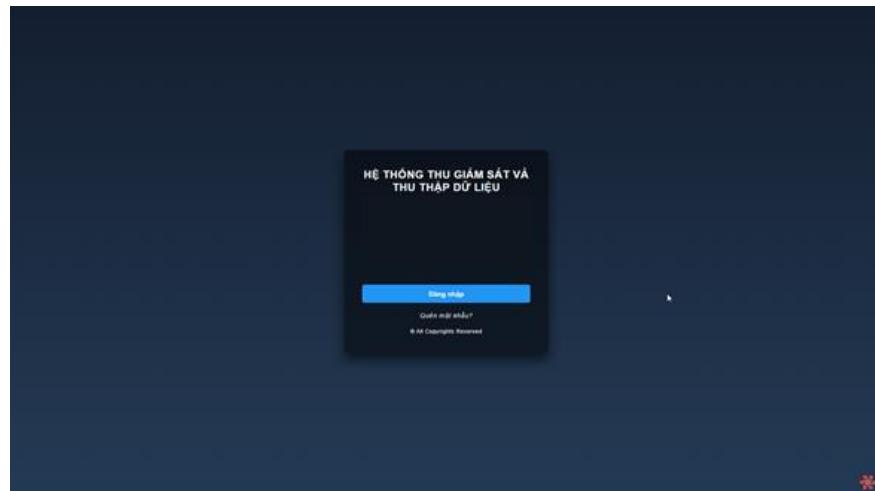
Hình 5.3 Sidebar (Ngăn kéo) của web giúp chuyển hướng trang

5.2.1. Trang đăng nhập

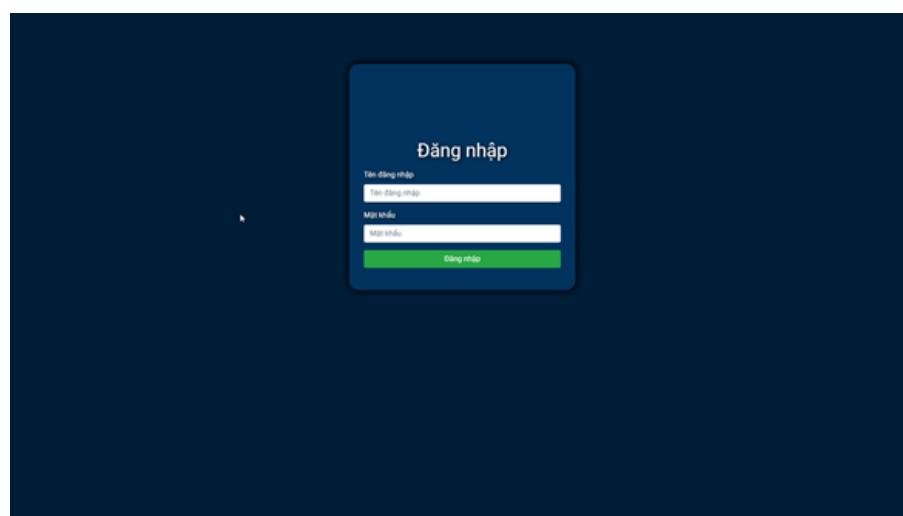
Trang đăng nhập là trang đầu tiên hiển thị cho người dùng khi người dùng đăng nhập lần đầu tiên. Trong trang này chứa logo của các đơn vị, tên ứng dụng và các nút đăng nhập. Ngoài ra còn nút “Quên mật khẩu” nhưng chưa thực hiện.



Hình 5.4 Thiết kế giao diện đăng nhập



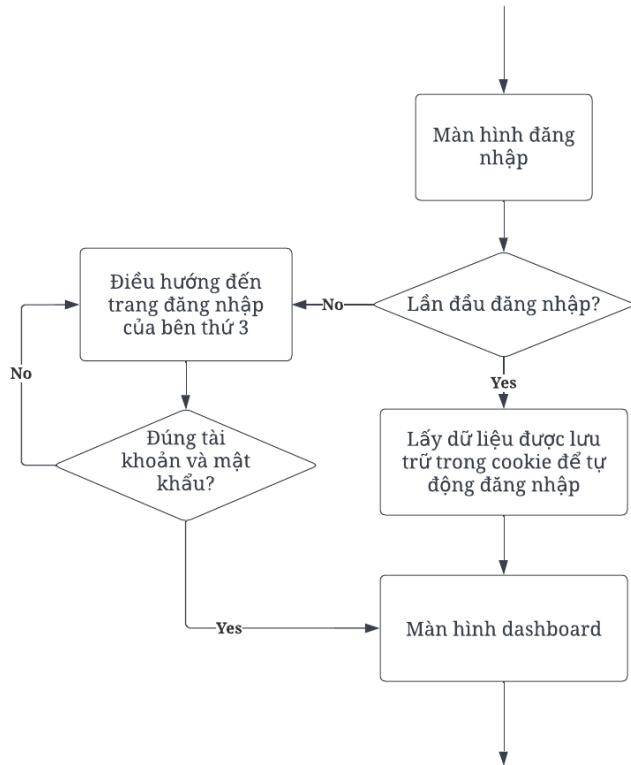
Hình 5.5 Giao diện đăng nhập của ứng dụng



Hình 5.6 Chuyển hướng trang đăng nhập

Sau khi nhấn nút đăng nhập, web sẽ điều hướng tới giao diện đăng nhập của máy chủ nhờ kỹ thuật OpenID Connect sẽ được đề cập ở phần sau.

Giải thuật cơ bản cho trang đăng nhập:



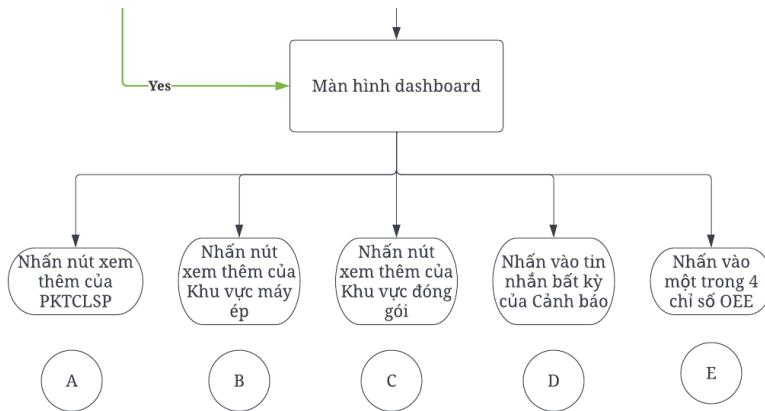
Hình 5.7 Giải thuật xử lý đăng nhập

Đối với người dùng đăng nhập lần đầu, sẽ được điều hướng đến trang đăng nhập. Sau khi nhập tài khoản và mật khẩu được cấp, nhấn nút “Đăng nhập”, ứng dụng sẽ gửi dữ liệu lên máy chủ thông qua giao thức HTTP để yêu cầu đăng nhập. Máy chủ sẽ chuyển hướng về trang chủ kèm theo chuỗi token (một chuỗi mã dùng để xác nhận đăng nhập, dùng cho những hành động sửa đổi dữ liệu trên máy chủ như POST, PUT, PATCH) trong chuỗi truy vấn, token này sẽ được lưu vào cookie, duy trì đăng nhập trong 1 ngày.

Đối với người dùng đăng nhập không phải lần đầu, phần mềm sẽ lấy thông tin user trong cookie để tự động đăng nhập, người dùng không cần phải nhập tài khoản và mật khẩu lần thứ 2.

5.2.2. Trang Dashboard

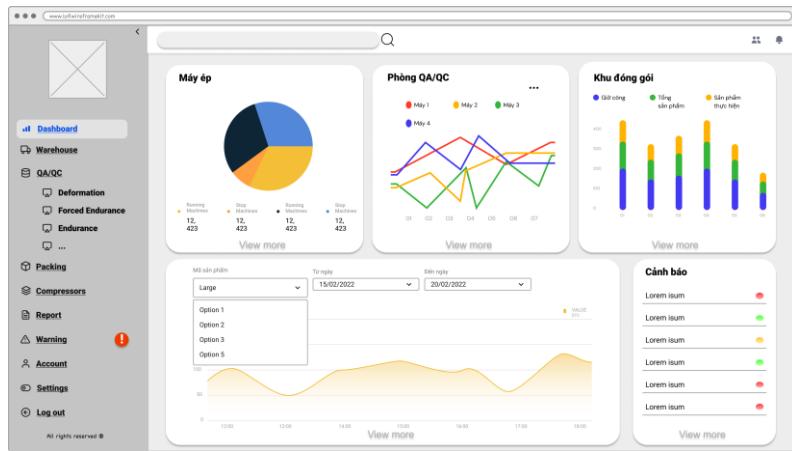
Sau khi đăng nhập thành công, web sẽ chuyển hướng về trang Dashboard có chức năng xem các thông số vận hành tổng quát của máy, chỉ số OEE trong 30 ngày gần nhất và các sự kiện alarm gần nhất.



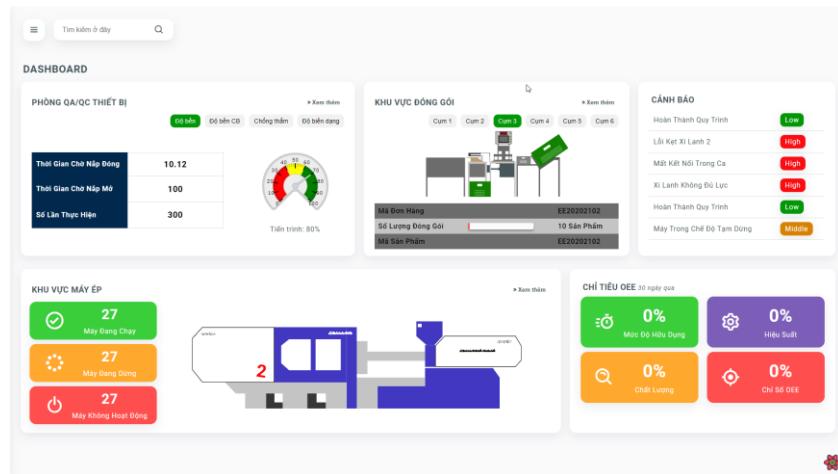
Hình 5.8 Sơ đồ chuyển hướng của trang Dashboard

Trong trang có 5 cách để chuyển hướng đến nội dung mình cần xem:

- Nút xem thêm của PKTCLSP: chuyển đến trang giám sát thông số 4 máy vận hành của phòng.
- Nút xem thêm của khu vực máy ép: chuyển đến trang giám sát thông số các máy ép dưới dạng bảng.
- Nút xem thêm của khu vực đóng gói: chuyển đến trang giám sát quá trình làm việc của công nhân của 6 cụm đóng gói.
- Nhấn vào tin nhắn bất kỳ của Cảnh báo: ở mục cảnh báo không thể hiện rõ alarm đang ở khu vực nào, người dùng cần nhấn vào từng cảnh báo để chuyển hướng đến trang Cảnh báo và xem tin nhắn alarm cụ thể.
- Nhấn vào 1 trong 4 chỉ số OEE, trang sẽ chuyển hướng đến trang báo cáo OEE để xem các chỉ số OEE khu vực máy ép.



Hình 5.9 Thiết kế giao diện trang Dashboard



Hình 5.10 Giao diện trang Dashboard của ứng dụng

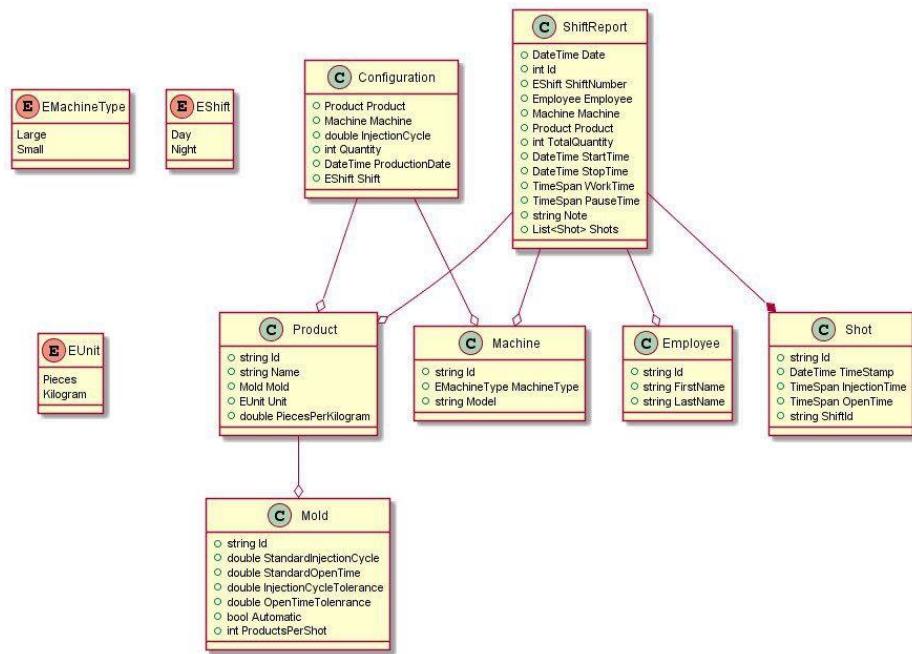
Cụ thể các thông số tổng quát ở trang này:

- PKTCLSP: Thông số cài đặt của 4 máy, trong trường hợp 4 máy không vận hành, các thông số sẽ có giá trị 0.
- Khu vực đóng gói: Giám sát mã đơn hàng, số lượng thực hiện được và lỗi do công nhân đóng gói thiếu xảy ra trong ca làm.
- Khu vực máy ép: Xem được số lượng máy đang hoạt động, tạm dừng hay ngắt kết nối.
- Chỉ số OEE: Tính toán 4 chỉ số OEE trong 30 ngày gần nhất.

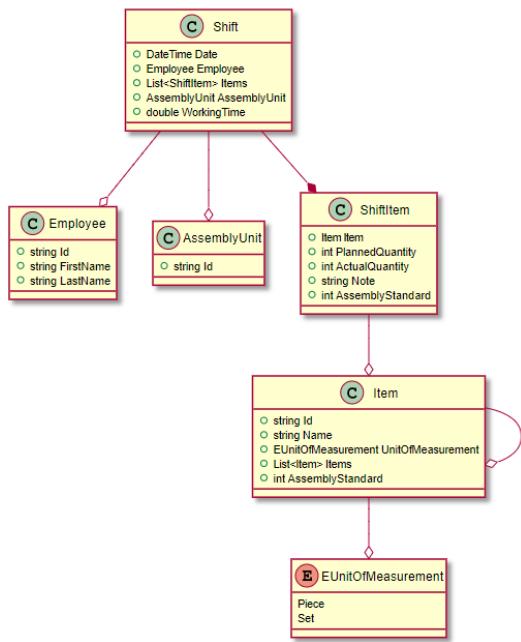
5.2.3. Trang Tiến độ kế hoạch

Ở trang này, người dùng có thể truy xuất được tiến độ sản xuất theo kế hoạch sản xuất (theo tháng) hay lịch trình (theo ngày).

Cấu trúc dữ liệu trả về của endpoint tiến độ kế hoạch máy ép, và báo cáo:



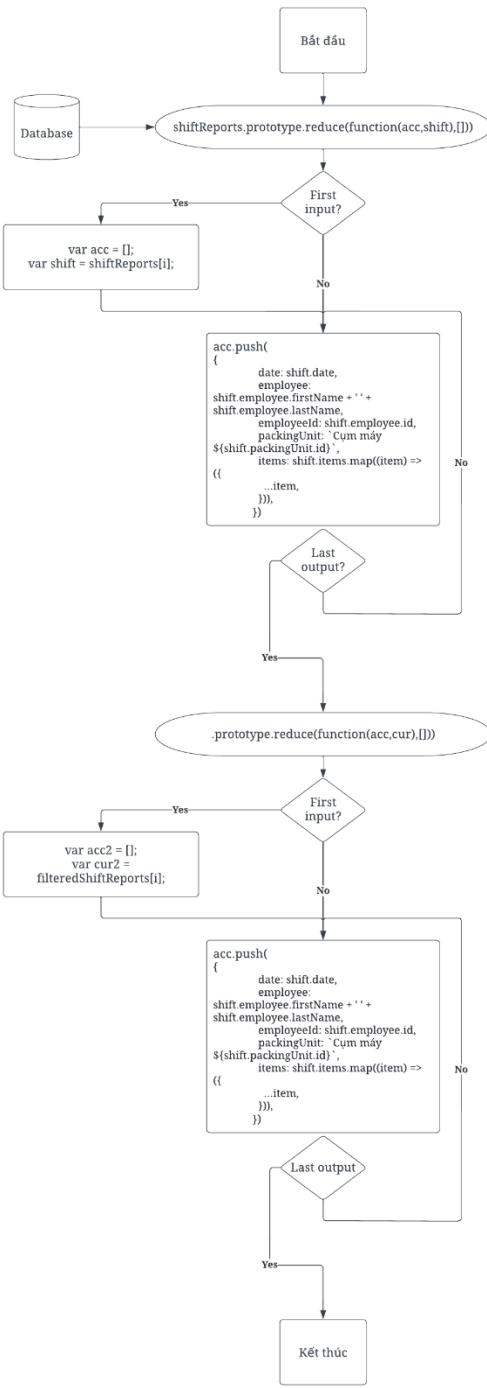
Hình 5.11 Cấu trúc dữ liệu báo cáo kiểm tra tiến độ khu vực máy ép



Hình 5.12 Cấu trúc dữ liệu báo cáo kiểm tra tiến độ khu vực đóng gói

Ở dữ liệu trả về cho phần mềm của khu vực đóng gói, ta chỉ cần lấy thông số “actualQuantity” chia cho “plannedQuantity” sẽ ra được tiến độ lịch trình theo ngày và tháng mong muốn. Tuy nhiên ở dữ liệu trả về của khu vực máy ép, ta chỉ có thông số “totalQuantity” có thể hiểu như là một “actualQuantity” và không có thông số “plannedQuantity” vì thông số này được công ty tính bằng cách lấy tổng thời gian thực hiện (trường “workTime”) chia cho chu kỳ ép tiêu chuẩn của khuôn đúc.

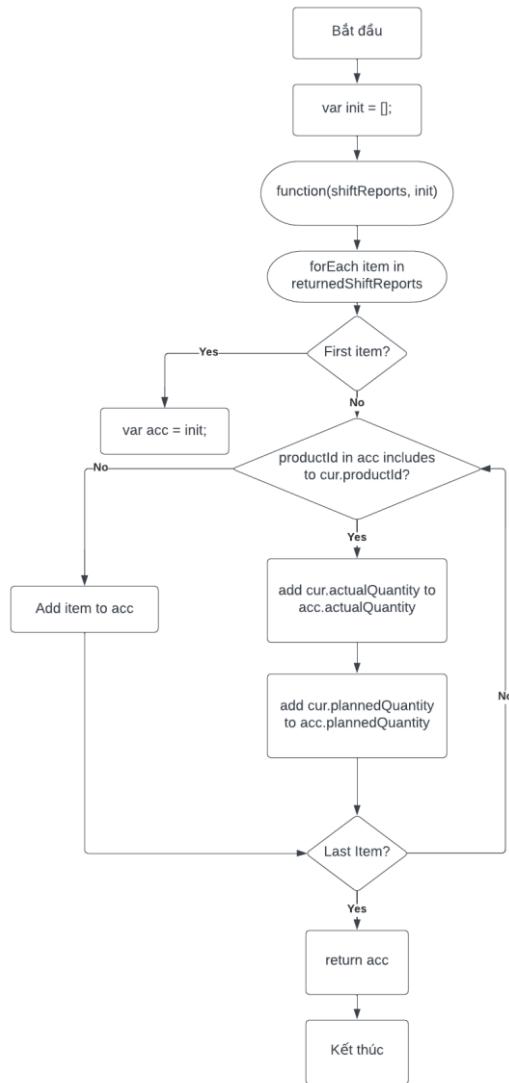
Với cấu trúc dữ liệu máy ép như trên, nhóm thiết kế giải thuật xử lý dữ liệu cho trang kiểm tra lịch trình sản xuất theo ngày như hình.



Hình 5.13 Giải thuật sàng lọc dữ liệu với cấu trúc dữ liệu được định sẵn

Với dữ liệu kiểm tra kế hoạch sản xuất theo tháng, theo cấu trúc trúc dữ liệu của database, các gói JSON trả về là báo cáo kết quả theo từng ca. Chính vì thế, ta cần phải có một lưu đồ giải thuật lập trình phục vụ cho mục đích tổng hợp

các trường “plannedQuantity” và “actualQuantity” theo trường “productId”.
 Lưu đồ giải thuật được thực hiện theo hình.



Hình 5.14 Lưu đồ giải thuật xử lý dữ liệu theo trang kiểm tra tiến độ kế hoạch sản xuất

Vì kiểm soát tiến độ sản xuất là yêu cầu thêm của phía công ty sau lần khảo sát thứ 2 nên nhóm thực hiện thiết kế và lập trình thẳng trên trình duyệt web.

TRANG CHỦ / BÁO CÁO TIẾN ĐỘ THEO NGÀY														
LỊCH TRÌNH ĐÓNG GÓI					LỊCH TRÌNH MÁY ÉP									
Bộ Lọc														
Ngày truy xuất 01/02/2022														
Truy xuất														
Kết Quả Tìm Kiếm														
Ngày	Nhân Viên Đóng Máy	Mã Nhân Viên	Cụm Máy	Mã Sản Phẩm	Tên Sản Phẩm	Lịch Trình Đóng Gói	Thực Hiện	Giải Trình	Tiến Độ					
22/08/2021	Tâm Dinh Khanh	F4089436-44e0-463c-87f5-4699b0205f60	Cụm Máy 6	HAD201T-HA3	Nắp Bán Câu HA-03 Trắng HA4	1000	580	Không	<div style="width: 58%;"></div>					
22/08/2021	Tâm Dinh Khanh	F4089436-44e0-463c-87f5-4699b0205f60	Cụm Máy 6	HAD201T-HA2	Nắp Bán Câu HA-03 Nguyên HA7	1000	750	Không	<div style="width: 75%;"></div>					
22/08/2021	Tâm Dinh Khanh	F4089436-44e0-463c-87f5-4699b0205f60	Cụm Máy 6	HAD201T-HA2	Nắp Bán Câu HA-03 Trắng HA2	1000	657	Không	<div style="width: 65.7%;"></div>					
02/02/2022	Hoài Dinh Khanh	D59090fb-9ab5-46de-89ff-4699b0205f60	Cụm Máy 4	HAD201T-C05	Nắp Bán Câu HA-03 Trắng HA4	1000	996	Không	<div style="width: 99.6%;"></div>					
02/02/2022	Hoài Dinh Khanh	D59090fb-9ab5-46de-89ff-4699b0205f60	Cụm Máy 4	HAD201T-HA3	Nắp Bán Câu HA-02 Trắng HA4	1000	566	Không	<div style="width: 56.6%;"></div>					
02/02/2022	Hoài Dinh Khanh	D59090fb-9ab5-46de-89ff-4699b0205f60	Cụm Máy 4	HAD201T-C02	Nắp Bán Câu HA-02 Trắng HA4	1000	940	Không	<div style="width: 94%;"></div>					

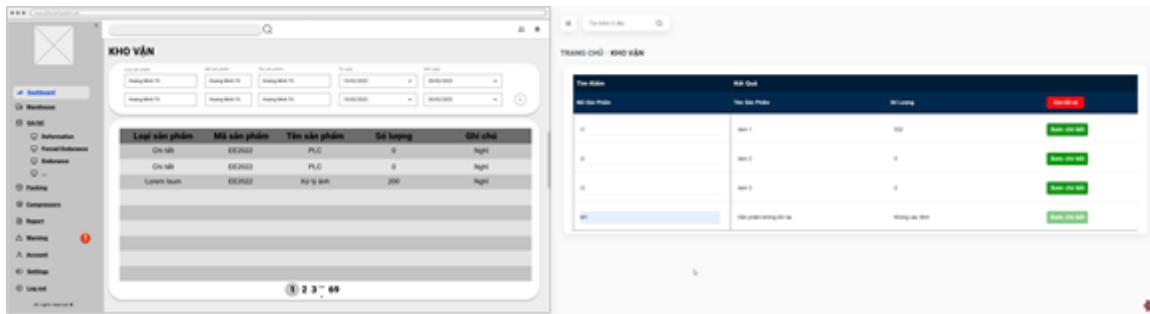
Hình 5.15 Giao diện báo cáo kiểm tra tiến độ sản xuất theo ngày

TRANG CHỦ / BÁO CÁO TIẾN ĐỘ THEO THÁNG				
LỊCH TRÌNH ĐÓNG GÓI				
LỊCH TRÌNH MÁY ÉP				
Bộ Lọc				
Tháng truy xuất Hè 2022				
Truy xuất				
Kết Quả Tìm Kiếm				
Mã Sản Phẩm	Tên Sản Phẩm	Số Lượng Cái Đặt	Số Lượng Đóng Gói Thực	Tiến Độ
HAD201T-HA2	Nắp Bán Câu HA-03 Ngực TT1	2000	1342	<div style="width: 67.1%;"></div>
HAD101T-HA1	Nắp Bán Câu HA-03 Biển HA6	2000	1616	<div style="width: 80.8%;"></div>
HAD101T-HA2	Nắp Bán Câu HA-03 Trắng HA2	1000	723	<div style="width: 72.3%;"></div>
HAD101T-HA3	Nắp Bán Câu HA-03 Trắng HC1	2000	1219	<div style="width: 60.9%;"></div>
HAD201T-C02	Nắp Bán Câu HA-03 Ngực TT1	1000	794	<div style="width: 79.4%;"></div>
HAD201T-HA5	Nắp Bán Câu HA-03 Trắng HC1	1000	740	<div style="width: 74%;"></div>
HAD101T-HA4	Nắp Bán Câu HA-03 Trắng HA2	1000	679	<div style="width: 67.9%;"></div>
HAD201T-C05	Nắp Bán Câu HA-03 Biển HA6	1000	521	<div style="width: 52.1%;"></div>
HAD201T-HA3	Nắp Bán Câu HA-02 Trắng HA4	1000	519	<div style="width: 51.9%;"></div>

Hình 5.16 Giao diện báo cáo kiểm tra tiến độ sản xuất theo tháng

5.2.4. Trang Kho vận và Chi tiết kho vận

Ở trang Kho vận tổng quát, người dùng dùng điền mã sản phẩm hợp lệ vào ô truy xuất mã sản phẩm, nếu hợp lệ, sẽ hiển thị tên sản phẩm và số lượng tồn của mã sản phẩm đó. Người dùng có thể nhấn vào nút xem chi tiết để chuyển hướng đến trang Chi tiết của mã sản phẩm đó.

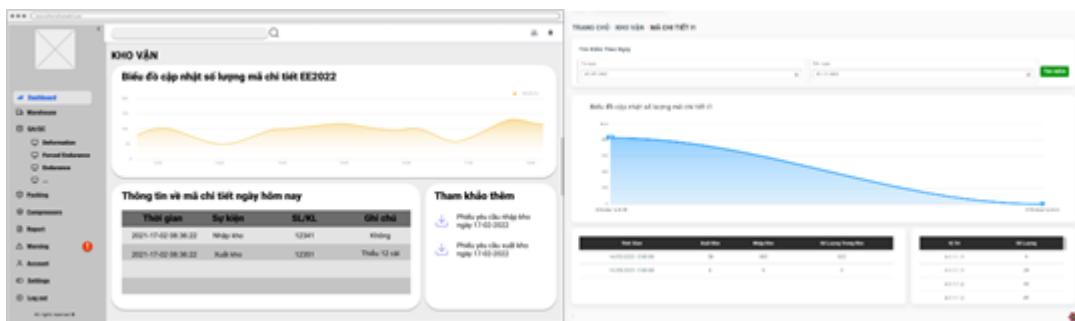


Hình 5.17 Thiết kế giao diện trang Kho vận tổng quát (trái) và giao diện phần mềm (phải)

Trang chi tiết kho vận có mục đích kiểm tra tồn kho của mã sản phẩm bất kỳ.

Cụ thể:

- Số lượng tồn của các mã chi tiết ép máy nhập kho bất kỳ.
- Đồ thị thể hiện số lượng tồn theo các ngày.
- Thông tin nhập kho và xuất kho của các ngày đó.
- Vị trí từng rõ chứa mã sản phẩm và số lượng còn lại.



Hình 5.18 Thiết kế giao diện trang kho chi tiết (trái) và giao diện phần mềm (phải)

5.2.5. Trang Phòng QA/QC thiết bị và các trang Chi tiết

Ở trang này có chức năng giám sát những thông số cài đặt và trạng thái hoạt động cho 4 máy kiểm tra chất lượng bao gồm:

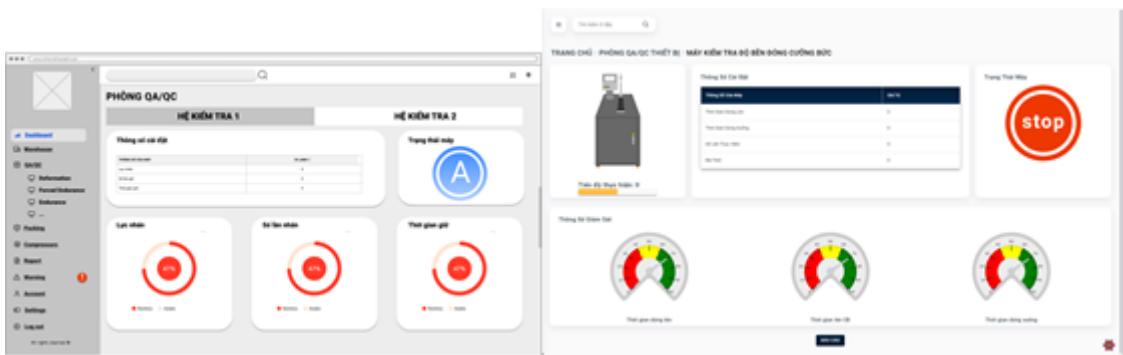
- Máy kiểm tra độ bền rơi êm
- Máy kiểm tra độ bền cưỡng bức

- Máy kiểm tra độ biến dạng
- Máy kiểm tra chống thấm



Hình 5.19 Thiết kế giao diện trang PKTCLSP tổng quát (trái) và giao diện ứng dụng (phải)

Người dùng có thể nhấn vào từng thẻ (card) để xem chi tiết các thông số chi tiết của máy. Các trang Chi tiết này đều có chung một kiểu thiết kế.

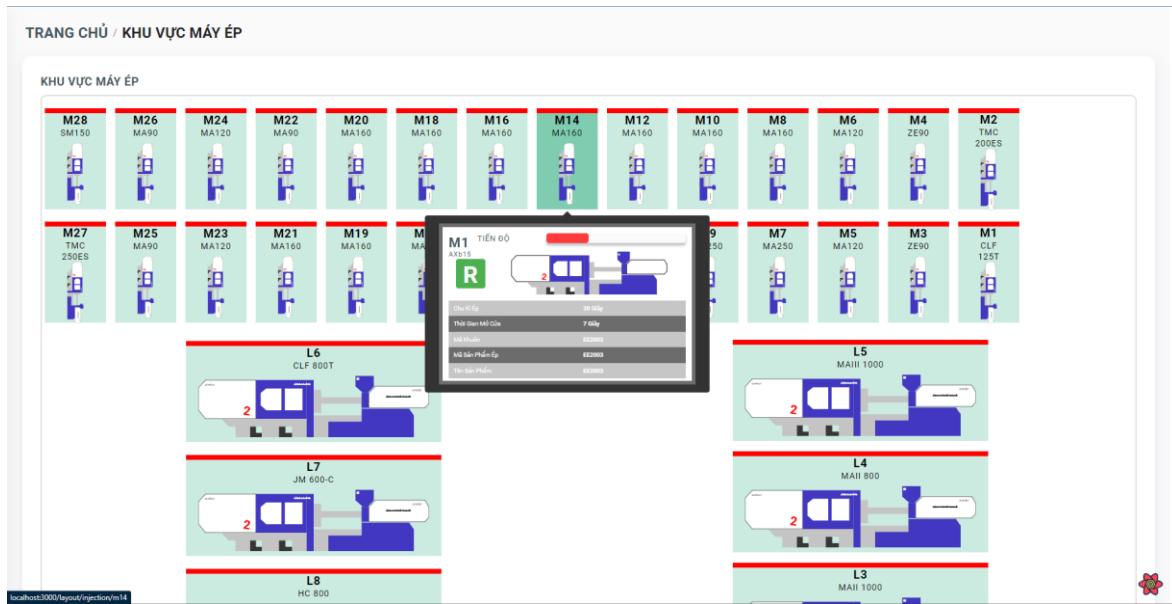


Hình 5.20 Thiết kế giao diện trang giám sát thông số các máy của PKTCLSP chi tiết (trái) và giao diện ứng dụng (phải)

5.2.6. Trang Khu máy ép tổng quan và trang các máy ép chi tiết

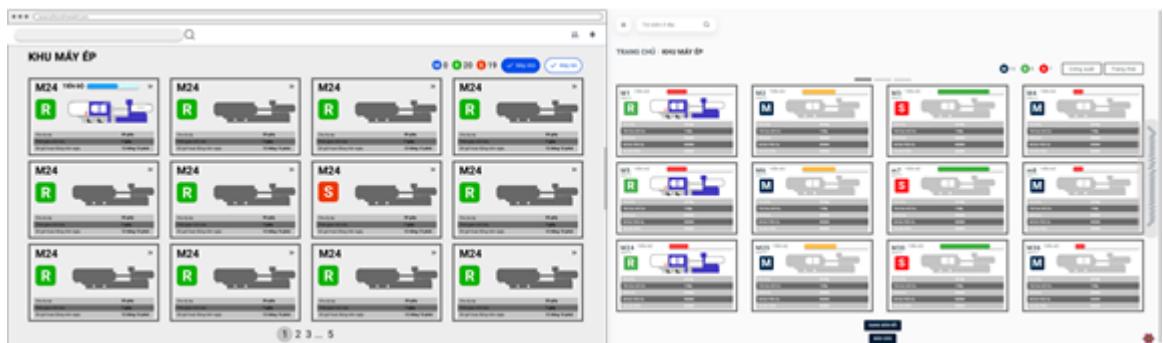
Ở trang này, người dùng có thể quan sát tổng quan trạng thái sản xuất của các máy ép phun dưới 2 dạng:

- Bản đồ: cho phép người dùng xem thông số các máy theo layout của nhà máy.



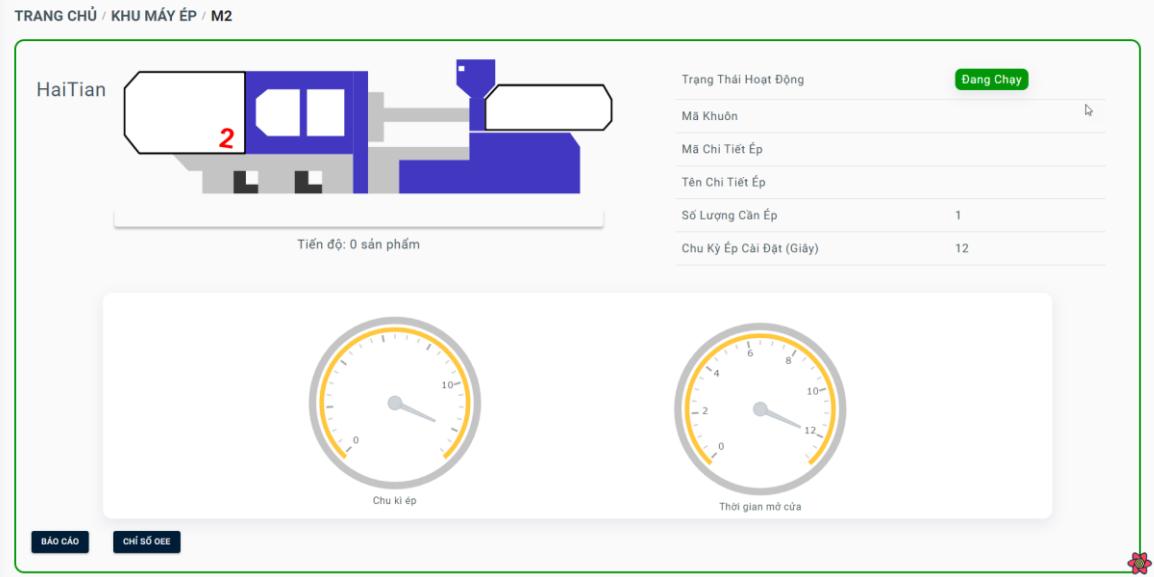
Hình 5.21 Giám sát thông số máy ép dưới dạng bản đồ nhà máy

- Bảng: giám sát thông số dưới dạng bảng, có bộ lọc để người dùng có thể quan sát trực quan hơn.



Hình 5.22 Giám sát thông số máy ép dưới dạng bảng (danh sách)

Ở 2 trang tổng quan này, người dùng có thể nhấn vào để đi đến trang chi tiết mỗi máy ép phun để quan sát thông số chi tiết của mỗi máy.



Hình 5.23 Giao diện trang giám sát thông số chi tiết của máy ép

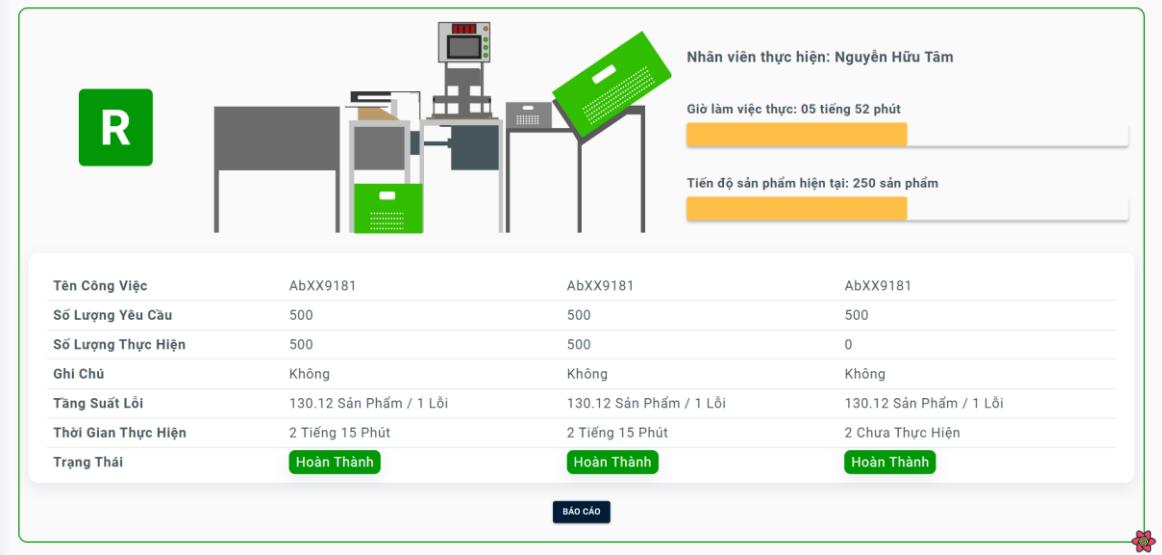
5.2.7. Trang Khu đóng gói và trang Chi tiết từng cụm máy

Trang Khu đóng gói tổng quát cho phép người dùng giám sát quy trình đóng gói của nhân viên tại 6 cụm máy bao gồm: tiến độ thực hiện, mã đơn hàng và lỗi đóng gói xảy ra.



Hình 5.24 Thiết kế giao diện trang giám sát quy trình đóng gói (trái) và giao diện ứng dụng

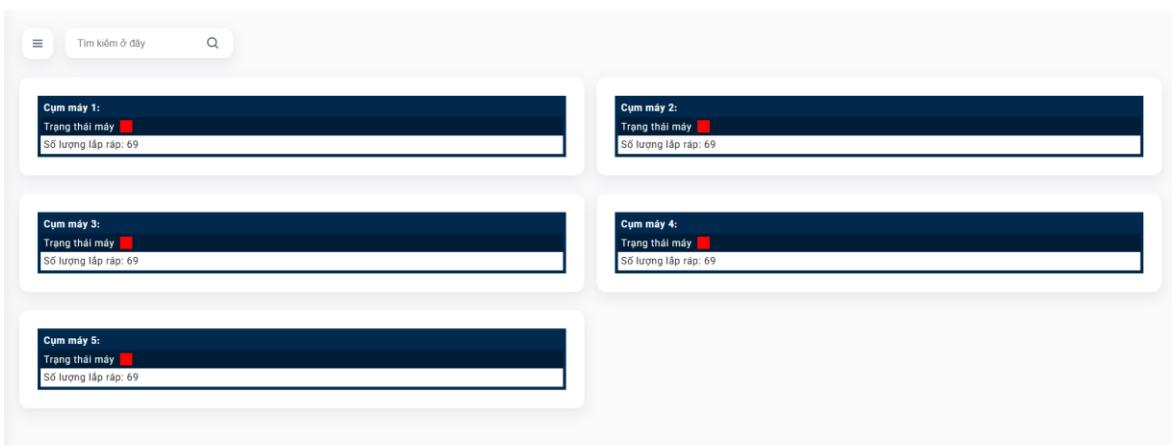
Người dùng có thể nhấp vào từng thẻ (card) để điều hướng đến trang giám sát cụm đóng gói cụ thể.



Hình 5.25 Giao diện giám sát thông số hoạt động chi tiết của cụm máy đóng gói

5.2.8. Trang Khu vực lắp ráp

Như đã nói trước đó, ở khu vực lắp ráp đang thử nghiệm tích hợp bộ đếm quản lý số lượng lắp ráp của nhân viên và lắp đặt một máy tính quản lý các bộ đếm đó nên nhóm chỉ thực hiện thu thập dữ liệu thông qua Window Service cho 5 cụm máy lắp ráp. Các dữ liệu thu thập bao gồm: trạng thái cụm máy, giá trị hiện tại tương ứng với số lượng nhân viên lắp ráp

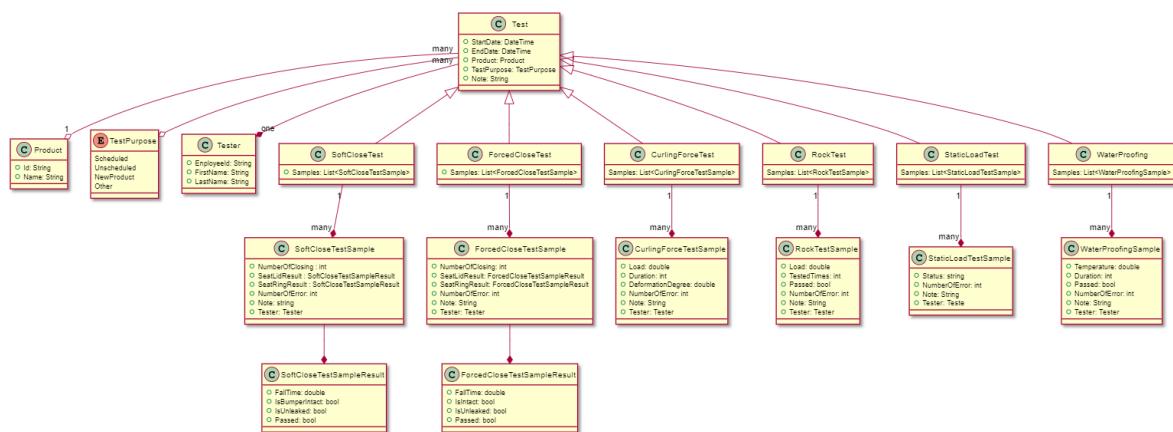


Hình 5.26 Giao diện demo giám sát khu vực lắp ráp

5.2.9. Trang Báo cáo các khu vực

Ở trang này, người dùng có thể truy xuất các báo cáo cần thiết ở 3 khu vực cụ thể: PKTCLSP, Khu vực máy ép, Khu vực đóng gói. Cụ thể:

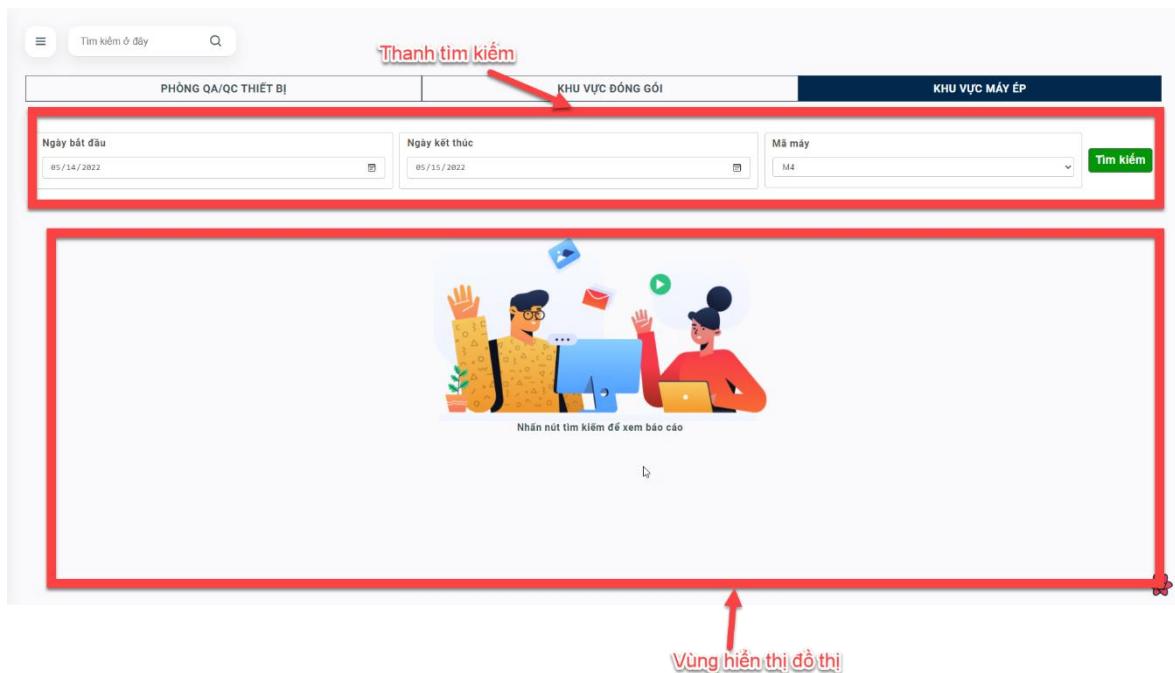
- Ở trang báo cáo “Phòng QA/QC thiết bị”, người dùng có thể truy xuất lại kết quả kiểm tra của 4 máy và xuất ra file excel với từng form báo cáo được quy định khác nhau. Cấu trúc dữ liệu và mô tả trang được thể hiện như hình.



Hình 5.27 Cấu trúc dữ liệu của trang báo cáo phòng KTCLSP

Hình 5.28 Mô tả trang báo cáo khu vực Phòng QA/QC thiết bị

- Ở trang báo cáo của “Khu vực máy ép”: người dùng có thể truy xuất đồ thị thể hiện độ chênh lệch giữa chu kỳ ép cài đặt và chu kỳ ép thực tế của nhân viên, tránh tình trạng gian lận trong quá trình vận hành. Cấu trúc dữ liệu của trang này tương đương với cấu trúc dữ liệu của trang báo cáo tiến độ kế hoạch sản xuất.



Hình 5.29 Mô tả trang báo cáo khu vực máy ép



Hình 5.30 Mô tả đồ thị báo cáo của khu vực máy ép

- Ở trang báo cáo của “Khu vực đóng gói”: Hỗ trợ trưởng bộ phận xem và xuất file báo cáo tháng theo biểu mẫu được quy định sẵn trong công ty.

Thanh tìm kiếm

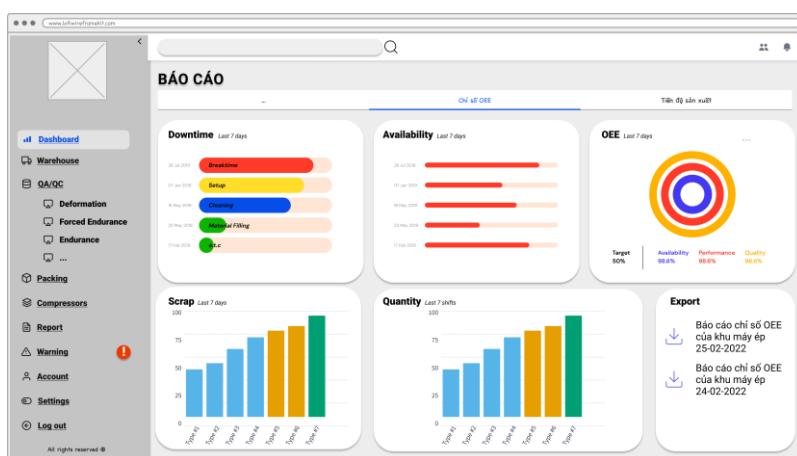
PHÒNG QA/QC THIẾT BỊ	KHU VỰC ĐÓNG GÓI	KHU VỰC MÁY ÉP																																																																																	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text" value="Ngày bắt đầu"/> 05/01/2022 Tim kiếm </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="text" value="Tên báo cáo"/> Báo cáo đóng gói 05/2022 Xuất excel </div>																																																																																			
<p>KẾ QUẢ TÌM KIẾM</p> <table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th>STT</th> <th>Ngày</th> <th>Mã Công Việc Tim kiếm ở đây</th> <th>Tên Công Việc Tim kiếm ở đây</th> <th>Đvt</th> <th>Số Lượng Tim kiếm ở đây</th> <th>Người Thực Hiện Tim kiếm ở đây</th> <th>Thời Gian Thực Hiện</th> <th>Ghi Chú Tim kiếm ở đây</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>06/05/2021</td> <td>HA0101T-HA4</td> <td>Nắp Bàn Cầu HA-03 Ngọc HA7</td> <td>Bộ</td> <td>664</td> <td>Đinh Khánh Dũng</td> <td>22.26</td> <td>Không</td> </tr> <tr> <td>2</td> <td>06/05/2021</td> <td>HA0201T-HA3</td> <td>Nắp Bàn Cầu HA-03 Biển TT1</td> <td>Cái</td> <td>765</td> <td>Đinh Khánh Dũng</td> <td>22.26</td> <td>Không</td> </tr> <tr> <td>3</td> <td>06/05/2021</td> <td>HA0201T-HA2</td> <td>Nắp Bàn Cầu HA-03 Ngọc HA7</td> <td>Cái</td> <td>606</td> <td>Đinh Khánh Dũng</td> <td>22.26</td> <td>Không</td> </tr> <tr> <td>1</td> <td>14/09/2021</td> <td>HA0101T-HA2</td> <td>Nắp Bàn Cầu HA-03 Trắng TT1</td> <td>Cái</td> <td>886</td> <td>Trịnh Dũng</td> <td>22.55</td> <td>Không</td> </tr> <tr> <td>2</td> <td>14/09/2021</td> <td>HA0101T-HA2</td> <td>Nắp Bàn Cầu HA-03 Ngọc HA7</td> <td>Bộ</td> <td>968</td> <td>Trịnh Dũng</td> <td>22.55</td> <td>Không</td> </tr> <tr> <td>3</td> <td>14/09/2021</td> <td>HA0101T-HA4</td> <td>Nắp Bàn Cầu HA-02 Trắng HA4</td> <td>Bộ</td> <td>594</td> <td>Trịnh Dũng</td> <td>22.55</td> <td>Không</td> </tr> <tr> <td>4</td> <td>14/09/2021</td> <td>HA0101T-HA2</td> <td>Nắp Bàn Cầu HA-03 Ngọc TT1</td> <td>Bộ</td> <td>766</td> <td>Trịnh Dũng</td> <td>22.55</td> <td>Không</td> </tr> <tr> <td>1</td> <td>15/01/2022</td> <td>HA0201T-HA2</td> <td>Nắp Bàn Cầu HA-03 Biển TT1</td> <td>Bộ</td> <td>849</td> <td>Trịnh Lê</td> <td>22.74</td> <td>Không</td> </tr> </tbody> </table>			STT	Ngày	Mã Công Việc Tim kiếm ở đây	Tên Công Việc Tim kiếm ở đây	Đvt	Số Lượng Tim kiếm ở đây	Người Thực Hiện Tim kiếm ở đây	Thời Gian Thực Hiện	Ghi Chú Tim kiếm ở đây	1	06/05/2021	HA0101T-HA4	Nắp Bàn Cầu HA-03 Ngọc HA7	Bộ	664	Đinh Khánh Dũng	22.26	Không	2	06/05/2021	HA0201T-HA3	Nắp Bàn Cầu HA-03 Biển TT1	Cái	765	Đinh Khánh Dũng	22.26	Không	3	06/05/2021	HA0201T-HA2	Nắp Bàn Cầu HA-03 Ngọc HA7	Cái	606	Đinh Khánh Dũng	22.26	Không	1	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Trắng TT1	Cái	886	Trịnh Dũng	22.55	Không	2	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Ngọc HA7	Bộ	968	Trịnh Dũng	22.55	Không	3	14/09/2021	HA0101T-HA4	Nắp Bàn Cầu HA-02 Trắng HA4	Bộ	594	Trịnh Dũng	22.55	Không	4	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Ngọc TT1	Bộ	766	Trịnh Dũng	22.55	Không	1	15/01/2022	HA0201T-HA2	Nắp Bàn Cầu HA-03 Biển TT1	Bộ	849	Trịnh Lê	22.74	Không
STT	Ngày	Mã Công Việc Tim kiếm ở đây	Tên Công Việc Tim kiếm ở đây	Đvt	Số Lượng Tim kiếm ở đây	Người Thực Hiện Tim kiếm ở đây	Thời Gian Thực Hiện	Ghi Chú Tim kiếm ở đây																																																																											
1	06/05/2021	HA0101T-HA4	Nắp Bàn Cầu HA-03 Ngọc HA7	Bộ	664	Đinh Khánh Dũng	22.26	Không																																																																											
2	06/05/2021	HA0201T-HA3	Nắp Bàn Cầu HA-03 Biển TT1	Cái	765	Đinh Khánh Dũng	22.26	Không																																																																											
3	06/05/2021	HA0201T-HA2	Nắp Bàn Cầu HA-03 Ngọc HA7	Cái	606	Đinh Khánh Dũng	22.26	Không																																																																											
1	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Trắng TT1	Cái	886	Trịnh Dũng	22.55	Không																																																																											
2	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Ngọc HA7	Bộ	968	Trịnh Dũng	22.55	Không																																																																											
3	14/09/2021	HA0101T-HA4	Nắp Bàn Cầu HA-02 Trắng HA4	Bộ	594	Trịnh Dũng	22.55	Không																																																																											
4	14/09/2021	HA0101T-HA2	Nắp Bàn Cầu HA-03 Ngọc TT1	Bộ	766	Trịnh Dũng	22.55	Không																																																																											
1	15/01/2022	HA0201T-HA2	Nắp Bàn Cầu HA-03 Biển TT1	Bộ	849	Trịnh Lê	22.74	Không																																																																											

Dữ liệu trả về dưới dạng bảng được lấy từ CSDL

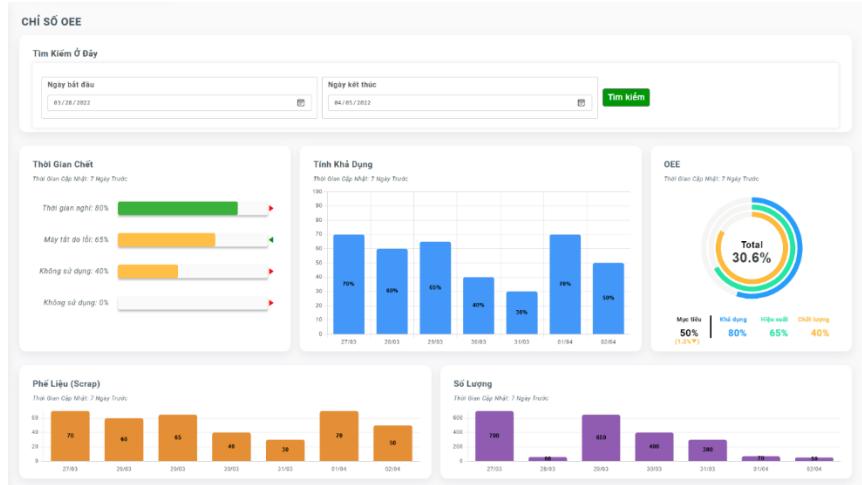
Hình 5.31 Mô tả trang báo cáo khu vực đóng gói sản phẩm

5.2.10. Trang Báo cáo chỉ số Overall Equipment Effectiveness (OEE index)

Trang này được xây dựng để phục vụ yêu cầu của ban điều hành của công ty. Ở trang này, người dùng có thể xem được các chỉ số sản xuất của máy ép như hình dưới đây:



Hình 5.32 Thiết kế giao diện trang báo cáo chỉ số OEE



Hình 5.33 Giao diện trang báo cáo và tính toán chỉ số OEE

Cụ thể:

- Đối với mục “Thời gian chết”, ta có công thức

$$\frac{\text{Planned Production Time} - \text{Actual Production Time}}{\text{Standard Working Time}}$$

$$= \frac{\text{Pause Time}}{\text{Standard Working Time}}$$

Với *Pause Time* là thời gian ghi nhận máy đang tạm dừng, *Standard Working Time* là thời gian tiêu chuẩn của một ca làm dưới công ty (12 tiếng). Tùy vào lý do tạm dừng máy là gì, ta sẽ có 4 hạn mục chuẩn: Thời gian nghỉ, VĐK mất kết nối đến máy ép, Tạm dừng thay khuôn.

- Đối với mục “Tính khả dụng”, ta có công thức:

$$A = \frac{\text{Actual Working Time}}{\text{Standard Working Time}} (\%)$$

Với *Actual Working Time* là thời gian máy sử dụng thực tế trong 1 ca làm việc và *Standard Working Time* là thời gian tiêu chuẩn của một ca làm dưới công ty (12 tiếng). Tiêu chuẩn một ngày sẽ có 2 ca làm việc và mỗi ca kéo dài 12 tiếng, để có thể hiển thị được dữ liệu trên đồ thị, ta cần sử dụng phép cộng 2 trường dữ liệu “totalQuantity” của 2 ca đêm và ngày đó.

- Đối với mục “Phế liệu (Scrap)”, Phế liệu được định nghĩa là các sản phẩm mà máy ép ép lỗi trong 1 ngày, được tính dựa trên công thức:

$$(Shot Count * Parts Per Shot) - Actual Quantity$$

Ở Window Service của khu vực máy ép, ta có thể thu thập được dữ liệu quan trọng để tính toán số lượng Phế liệu đó là *Shot Count* (tổng số lần ép trong 1 ca làm), trường *Parts Per Shot* được đặc trưng bởi mỗi khuôn mà máy ép sử dụng và có sẵn trên CSDL. Trong khi đó, *Actual Quantity* được các nhân viên báo cáo cuối ca sau khi đã hoàn thành ca làm của mình.

- Đối với mục Số lượng, dữ liệu được vẽ bởi trường *Actual Quantity*, như đã nói ở trên, được các nhân viên báo cáo cuối ca sau khi đã hoàn thành ca làm của mình.
- Đối với mục chỉ số OEE, ta có 4 chỉ số tổng quát: Quality (Chất lượng), Performance (Hiệu suất), Availability, Overall Equipment Effectiveness. Chỉ số Availability đã được nêu ra ở trên, các chỉ số còn lại lần lượt được tính toán như sau:

$$\begin{aligned} P &= \frac{\text{Total Produced Parts}}{\text{Actual Working Time} * \text{Design Capacity}} \\ &= \frac{\text{Total Produced Parts} * \text{Standard Cycle Time}}{\text{Actual Working Time}} (\%) \end{aligned}$$

Với *Total Produced Parts* chính là tổng sản phẩm ép ra của máy và được tính bằng công thức *Shot Count * Parts Per Shot*, *Standard Cycle Time* là tiêu chuẩn chu kỳ ép cài đặt và sẽ tương ứng với mỗi khuôn mà máy ép sử dụng trong ca làm. Cuối cùng, *Actual Working Time* chính là thời gian máy sử dụng thực tế trong 1 ca làm việc.

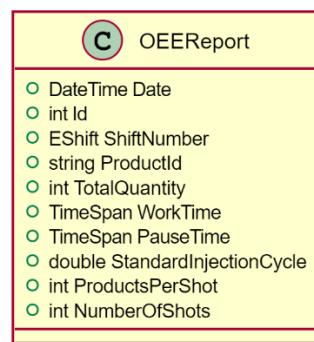
$$Q = \frac{\text{Total Qualified Parts}}{\text{Total Produced Parts}} = \frac{\text{Actual Quantity}}{\text{Shot Count} * \text{Parts Per Shot}}$$

Dựa theo quy trình công ty, biến *Total Qualified Parts* tương đương với *Actual Quantity* vì trước khi báo cáo cuối ca, nhân viên kiểm soát chất lượng sẽ kiểm tra chất lượng sản phẩm của từng máy ép. *Total Produced Parts*, như đã đề cập, được tính bằng công thức *Shot Count * Parts Per Shot*.

- Với 3 chỉ số A, P, Q, ta sẽ cho ra được chỉ số OEE của khu vực máy ép:

$$OEE = \frac{A * P * Q}{10000} (\%)$$

Cấu trúc dữ liệu của chỉ số OEE.



Hình 5.34 Cấu trúc dữ liệu của chỉ số OEE

Theo cấu trúc dữ liệu trên, đối chiếu với công thức vừa nêu, nhóm đưa ra bảng tham chiếu tương đương.

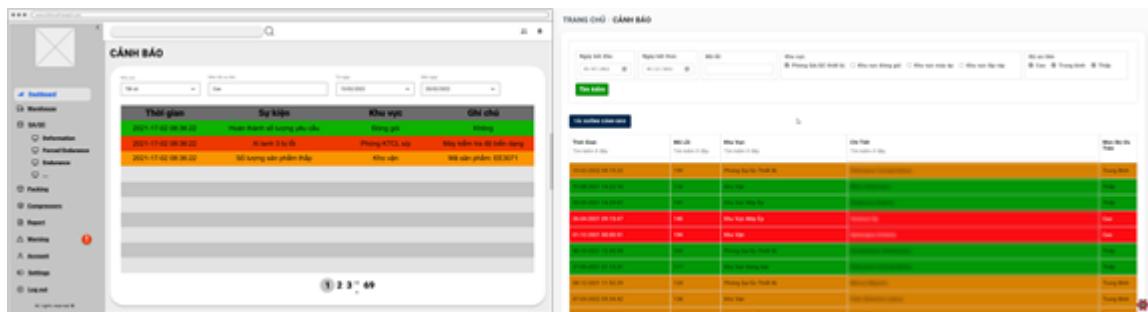
Trường trong cấu trúc dữ liệu	Biến trong công thức chỉ số OEE
TotalQuantity	<i>Actual Quantity</i>
WorkTime	<i>Actual Working Time</i>
PauseTime	<i>Pause Time</i>
StandardInjectionCycle	<i>Standard Cycle Time</i>
ProductsPerShot	<i>Parts Per Shot</i>

NumberOfShots	Shot Count
---------------	------------

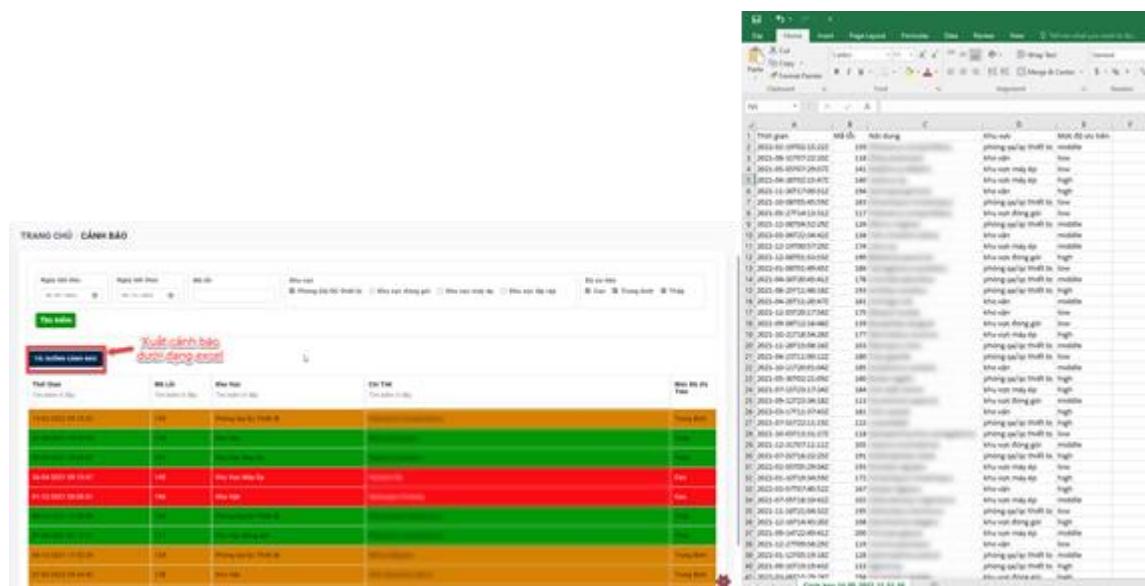
Bảng 5.1 Bảng tham chiếu tương đương cấu trúc dữ liệu và công thức tính chỉ số OEE

5.2.11. Trang cảnh báo

Ở trang này, người dùng có thể truy xuất lại các thông báo quan trọng ở các khu vực PKTCLSP, Khu vực máy ép, Khu vực lắp ráp, Khu vực đóng gói.



Hình 5.35 Thiết kế giao diện trang cảnh báo (trái) và giao diện ứng dụng
Phầm mềm có hỗ trợ người dùng in các sự kiện dưới dạng file excel.

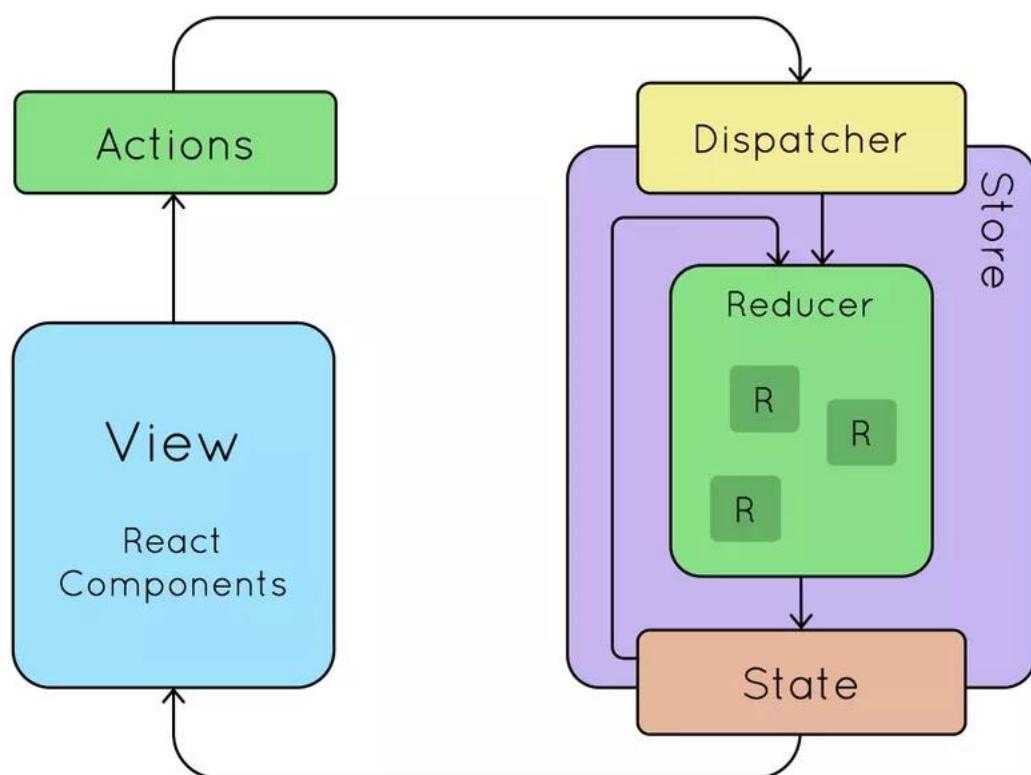


Hình 5.36 Mô tả quy chức năng excel cảnh báo của ứng dụng

5.3. Lập trình Web App

5.3.1. Phương pháp lập trình

Ở dự án sẽ sử dụng *Redux – predictable state management tool* hay gọi ngắn gọn là Redux. Đây là một phương pháp quản lý các trạng thái trong 1 trang web (*state-management*), được phát triển theo dạng thư viện và là một trong những phương thức quản lý state mà cả thế giới khuyên dùng khi xây dựng một ứng dụng cỡ vừa và lớn.



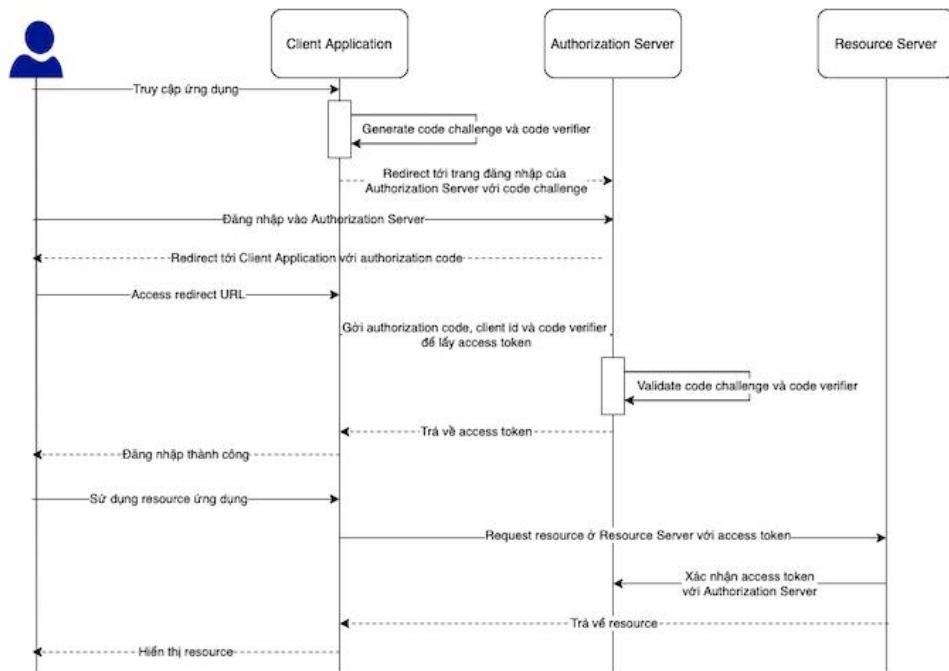
Hình 5.37 Mô hình hoạt động của kiến trúc Redux

Như trên hình thì ta có 5 thành phần chính cấu tạo nên Redux: *Actions, Store, Reducers, Dispatcher, State*. Trong đó, actions là các sự kiện tương tác từ người dùng với app như là nhấn nút, điền vào form, chuyển hướng trang web. Reducers là các hàm xử lý dữ liệu, chúng lấy trạng thái (state) hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như trong Store. Store lưu trạng thái ứng dụng và cả một ứng dụng chỉ có duy nhất

1 store. Ta có thể truy cập các state được lưu, cập nhật state, và đăng ký hoặc hủy đăng ký các listeners thông qua các phương thức Dispatcher.

Ngoài ra để có thể phân quyền cho người dùng, tránh trường hợp bất kỳ ai có tài khoản cũng có thể tác động đến phần mềm, ứng dụng sử dụng kỹ thuật OpenID Connect. Phân quyền là một cơ chế bảo mật nhằm xác định mức quyền truy cập (access level) hoặc những đặc quyền (privilege) liên quan đến tài nguyên hệ thống bao gồm file, các dịch vụ, chương trình máy tính, dữ liệu hoặc tính năng của ứng dụng. Đây là quá trình trao hoặc từ chối quyền truy cập vào một tài nguyên mạng cho phép người dùng truy cập vào nhiều loại tài nguyên khác nhau dựa trên danh tính của họ.

Hầu hết các hệ thống bảo mật đều dựa trên một quy trình hai bước. Bước đầu tiên là xác thực để xác định danh tính người dùng và bước hai là phân quyền, bước này cho phép người dùng truy cập vào các loại tài nguyên dựa trên danh tính đã được xác thực. Các hệ điều hành hiện đại phụ thuộc vào các quy trình phân quyền được thiết kế hiệu quả để tạo điều kiện thuận lợi cho việc triển khai và quản lý ứng dụng. Các yếu tố chính bao gồm loại người dùng, số và đặc điểm nhận dạng, yêu cầu xác minh và các hành động và vai trò liên quan.



Hình 5.38 Mô tả quy trình đăng nhập bằng kỹ thuật OpenID Connect

Theo như trong hình, người dùng khi truy cập ứng dụng, phần mềm sẽ tự truy cập, kiểm thử người dùng đã đăng nhập trước đó chưa, nếu chưa sẽ chuyển hướng đến trang đăng nhập kèm code challenge. Sau đó, người dùng sẽ nhập tài khoản kèm mật khẩu, nếu đăng nhập thành công sẽ chuyển từ trang xác thực về lại ứng dụng web kèm đoạn JWT, để có thể lấy được dữ liệu trong cơ sở dữ liệu của máy chủ, các phương thức CRUD bắt buộc ứng dụng phải kèm theo đoạn mã token, ở phía máy chủ sẽ giải mã đoạn token để phân tích quyền truy cập của người dùng để dựa vào đó mà đồng ý hoặc từ chối cho phép truy cập resource của máy chủ. Phương thức đăng nhập phân quyền có một vài ưu điểm sau:

- Cho phép truy cập hạn chế vào dữ liệu của người dùng và cho phép truy cập khi authorization JWT hết hạn.
- Có khả năng chia sẻ dữ liệu cho người dùng mà không phải tiết lộ thông tin cá nhân.

- Dễ dàng hơn trong việc xác thực và phân quyền nhân viên.

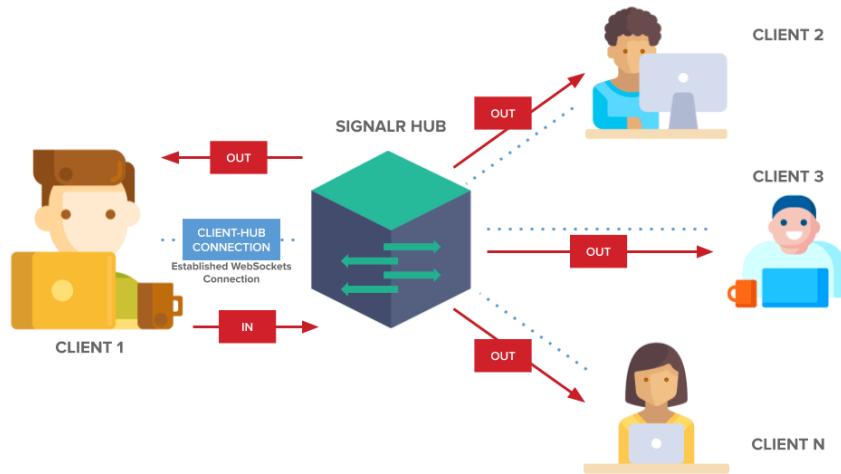
5.3.2. Giao thức truyền nhận dữ liệu liên tục

Để có thể nhận dữ liệu liên tục và hạn chế ảnh hưởng đến hiệu suất của phần mềm, nhóm sử dụng công nghệ websocket, cụ thể là SignalR websocket, được phát triển bởi Microsoft.

ASP.NET SignalR là một thư viện giúp đơn giản hóa quá trình thêm chức năng web real-time trong phát triển ứng dụng.

SignalR có thể sử dụng trong bất kì chức năng web real-time nào. Trong đó ứng dụng chat trên web là một ví dụ điển hình. Ngoài ra, các ứng dụng dashboards, monitoring là những đối tượng hướng đến việc sử dụng SignalR.

SignalR cung cấp một API đơn giản cho việc tạo server-to-client remote procedure call (RPC) để gọi những hàm JavaScript trong trình duyệt (và những nền tảng khác) từ code .Net của server-side. SignalR cũng bao gồm API cho việc quản lý kết nối (connect và disconnect events).



Hình 5.39 Sơ đồ mô tả cách luồng dữ liệu hoạt động trên SignalR WebSocket

Như trên hình, người dùng sẽ “bắt tay” 1 hub tại máy chủ, để có thể lấy dữ liệu từ người dùng khác, ta cần phải sử dụng các phương thức có sẵn như “invoke”, “on”, “build” ...

Trong dự án này, ứng dụng web sẽ là phía quyết định thời gian lấy mẫu của ứng dụng: sau một thời gian nhất định, ứng dụng sẽ “invoke” đến 1 hàm của hub với một chuỗi string định sẵn, kèm theo một object đóng vai trò như một cách gọi biến (tag) và sử dụng. Ví dụ:

```
id = setInterval(async () => {
    const rawData = await getTagsData(
        connection,
        'qaqclab',
        ['plc'],
        [
            'Sp Force Cylinder 12',
            'Sp No Press 12',
            'Sp Time Hold 12',
            'Pv Force Cylinder 1',
            'Pv Force Cylinder 2',
            'Pv No Press 1',
            'Pv No Press 2',
            'Pv Time Hold 1',
            'Pv Time Hold 2',
            'Mode App',
            'Green App',
            'Red App',
            'Error App',
            'Error Code',
        ]
    );
}, 300);
```

Trong đó, hàm getTagsData:

```
async function getTagsData(connection, eonNodeId,
deviceQueries, tagNames) {
    const nodeQuery = {
        EonNodeId: eonNodeId,
        DeviceQueries: deviceQueries.map((deviceQuery) => {
```

```

    return {
      DeviceId: deviceQuery,
      TagNames: tagNames,
    };
  }),
};

var result = await connection.invoke('GetListTags',
nodeQuery);
return result;
}

```

Ở đoạn code trên, sau khoảng thời gian 300ms, chúng ta sẽ gọi đến hàm “GetListTags” với node là khu vực Phòng QA/QC thiết bị “qaqc”, thiết bị ta cần giám sát trong phòng là “PLC”, và tương tự, các biến ta cần giám sát trong thiết bị đó là “Sp Force Cylinder 12”, “Sp Force Cylinder 3” ...

Ưu điểm lớn nhất khi hệ thống cho phép phía ứng dụng quy định thời gian cập nhật dữ liệu chính là ứng dụng có thể tùy biến, tùy máy, tùy khu vực mà có thể chỉnh thông số sao cho phù hợp, tránh tình trạng render nhiều lần làm giảm hiệu suất web. Ngoài ra, nếu như về định hướng tương lai, công ty có mở rộng tin học hóa, số hóa các khu vực khác để lưu trữ dữ liệu vào CSDL, phần mềm có thể nhanh chóng thích ứng, xây dựng và tạo ra một websocket hoàn toàn mới cho các khu vực đó.

5.3.3. Các bước lập trình

Khi lập trình một trang cho ứng dụng Web, cần thực hiện các quy trình như sau:

- Phân tích rõ cấu trúc dữ liệu của phần sắp lập trình. Ví dụ cấu trúc dữ liệu của một Chuỗi trả về cho kiểm tra tiến độ sản xuất theo ngày (*PlanTracking*) của khu vực đóng gói sẽ gồm một danh sách các *Shift* bên trong, mỗi *Shift* lại gồm danh sách các *ShiftItem*, trong *Item* lại có danh

sách các *Item* khác. Tuy nhiên, ứng dụng không thể hiện toàn bộ dữ liệu, mà sẽ tách từng element, để tạo thành một bảng dữ liệu cô đọng nhất.

Thuộc tính	Kiểu dữ liệu	Hiển thị	Chức năng
Date	DateTime	x	Ngày thực hiện đóng gói
Employee.Id	String	x	Mã nhân viên
Employee.FirstName	String	X	Tên nhân viên
Employee.LastName	String	X	Họ và chữ lót của nhân viên
PackingByUnit.Id	String	X	Đơn vị đóng gói
WorkingTime	double	X	Thời gian hoàn thành sản phẩm
ShiftItem.Item	List<Item>	X	Danh sách các sản phẩm công nhân thực hiện được vào ngày hôm đó
ShiftItem.PlannedQuantity	Int	X	Số lượng đóng gói cần thực hiện
ShiftItem.ActualQuantity	Int	X	Số lượng đóng gói thực hiện thực tế của nhân viên
ShiftItem.Note	String	X	Ghi chú hoặc giải

			trình
ShiftItem.PackingStandard	Int		Tiêu chuẩn đóng gói của sản phẩm
ShiftItem.UnitOfMeasurement	Enum		Đơn vị đo lường: kí lô gam hoặc cái
ShiftItem.Item.Item	List<Item>		Danh sách các chi tiết tạo nên thành phẩm đó

Bảng 5.2 Các thuộc tính trong sê hiển thị của *PackingPlanTracking*

- Thiết kế giao diện bằng Figma trước khi tiến hành lập trình.
- Tiến hành lập trình giao diện dựa trên bản thiết kế.
- Lập trình phần xử lý trạng thái:
 - Xác định các *action* có thể có.
 - Xác định các *state* (mỗi state sẽ re-render lại web một lần).
 - Khởi tạo *Store* (một lần cho cả ứng dụng) và lập trình cho *Slice* nhận các *Actions* và trả về các *state* (gọi là *Reducers*).
- Kết nối UI components với các *State* và *Dispatcher* để mỗi khi các *Reducer* xử lý các *Actions* thì UI có thể cập nhật *State* và re-render, sử dụng hooks *useSelector* và *useDispatcher*.

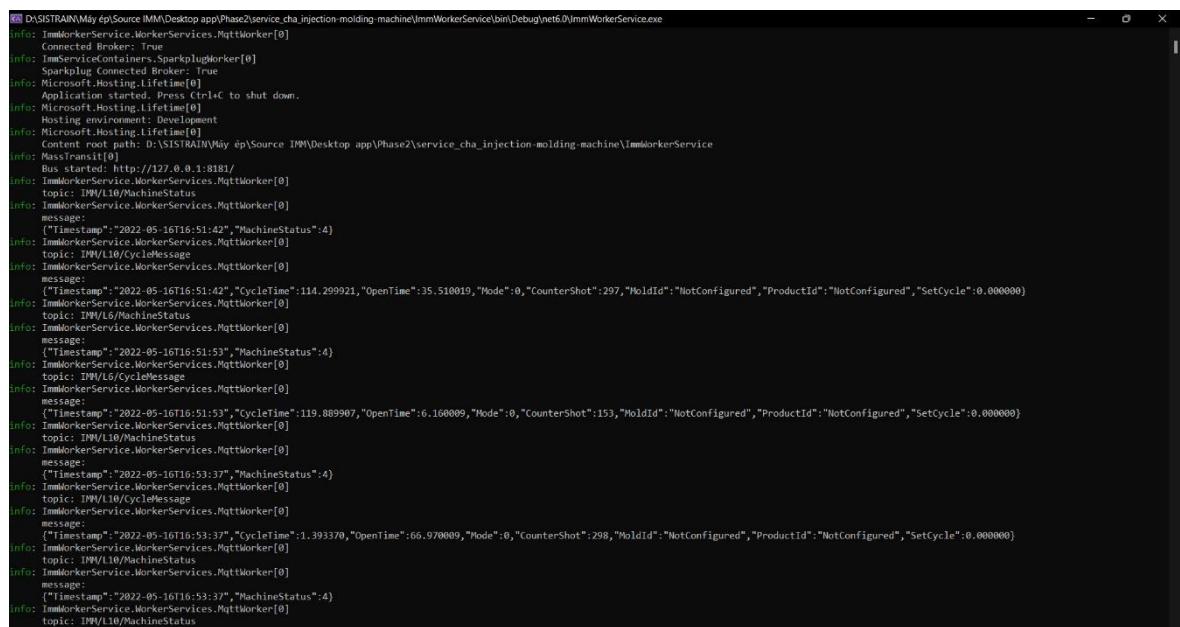
Đó là sơ bộ các bước cơ bản khi tư duy và xây dựng một trang web dữ liệu động, phần lập trình chi tiết sẽ rất dài nếu trình bày toàn bộ, nên trong nội dung quyển này, nhóm em chỉ trình bày các bước và một số ví dụ cơ bản nhất để minh họa công việc cho từng bước lập trình.

Chương 6. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

6.1. Kết quả thực hiện

6.1.1. Khu vực máy ép

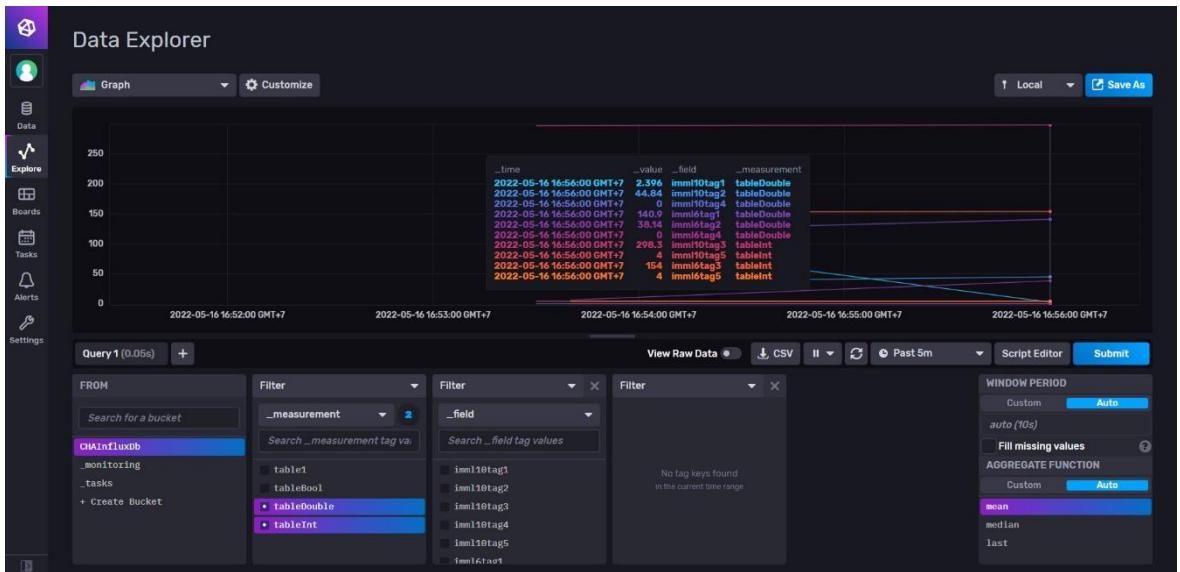
Ở khu vực máy ép, nhóm đã thực hiện chạy chương trình Windows Service để thu thập dữ liệu sau đó gửi lên Broker, cụ thể là 2 máy L6 và L10. Kết quả thu được như hình.



```
DAISTRAIN\May\Source\IMM\Desktop app\Phase2\service_cha_injection-molding-machine\ImmWorkerService\bin\Debug\net6.0\ImmWorkerService.exe
info: ImmWorkerService.WorkerServices.MqttWorker[0]
Connected Broker: True
info: ImmServiceContainers.SparkplugWorker[0]
Sparkplug Connected Broker: True
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: D:\DAISTRAIN\Máy ép\Source\IMM\Desktop app\Phase2\service_cha_injection-molding-machine\ImmWorkerService
info: Microsoft.Hosting.Lifetime[0]
Bus started at http://127.0.0.1:8183/
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/MachineStatus
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:42", "MachineStatus": 4}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/CycleMessage
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:42", "CycleTime": 114.299921, "OpenTime": 35.510019, "Mode": 0, "CounterShot": 297, "MoldId": "NotConfigured", "ProductId": "NotConfigured", "SetCycle": 0.000000}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L6/MachineStatus
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:42", "MachineStatus": 4}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L6/CycleMessage
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:42", "CycleTime": 119.889907, "OpenTime": 6.100009, "Mode": 0, "CounterShot": 153, "MoldId": "NotConfigured", "ProductId": "NotConfigured", "SetCycle": 0.000000}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/MachineStatus
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:43", "MachineStatus": 4}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/CycleMessage
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:51:43", "CycleTime": 1.393370, "OpenTime": 66.970009, "Mode": 0, "CounterShot": 298, "MoldId": "NotConfigured", "ProductId": "NotConfigured", "SetCycle": 0.000000}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/MachineStatus
info: ImmWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-16T16:53:37", "MachineStatus": 4}
info: ImmWorkerService.WorkerServices.MqttWorker[0]
topic: IWM/L10/MachineStatus
```

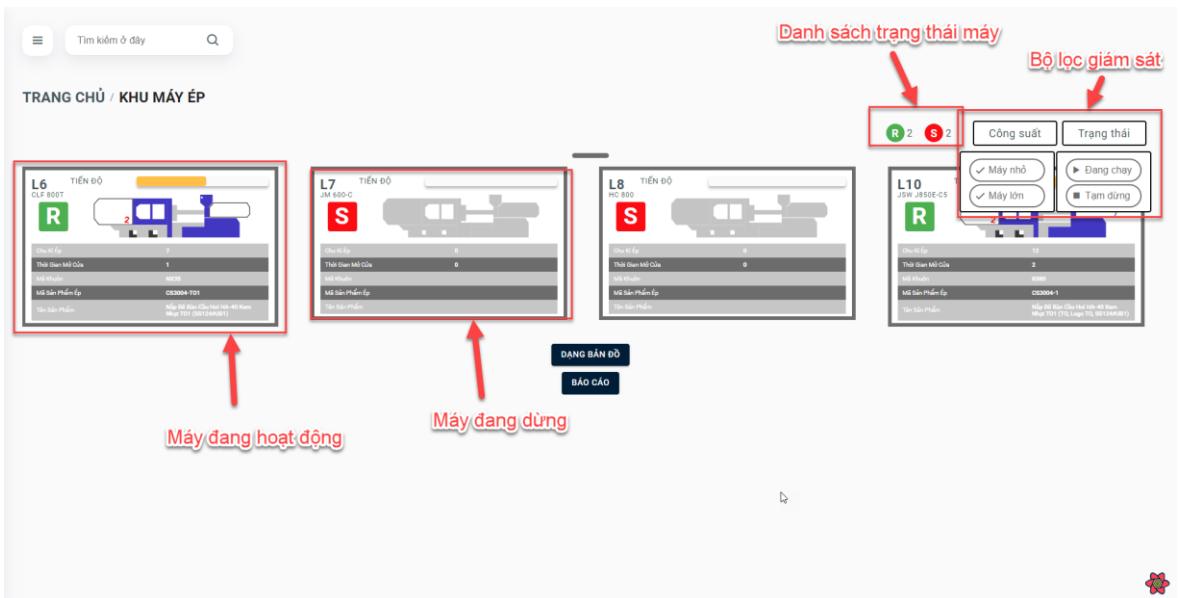
Hình 6.1 Windows Service chạy ổn định trên máy tính quản lý khu vực máy ép

Dữ liệu được Windows Service đọc và được Broker gửi lên máy chủ cơ sở dữ liệu và được thể hiện thông qua đồ thị như hình dưới.

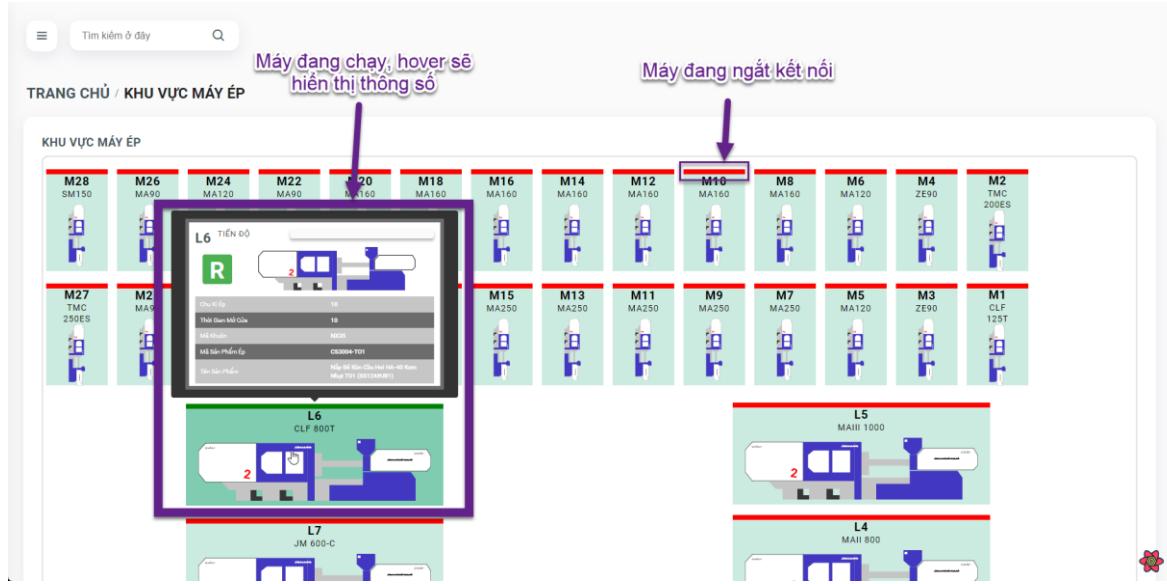


Hình 6.2 Dữ liệu 2 máy ép ghi nhận được trên database thể hiện dưới dạng biểu đồ

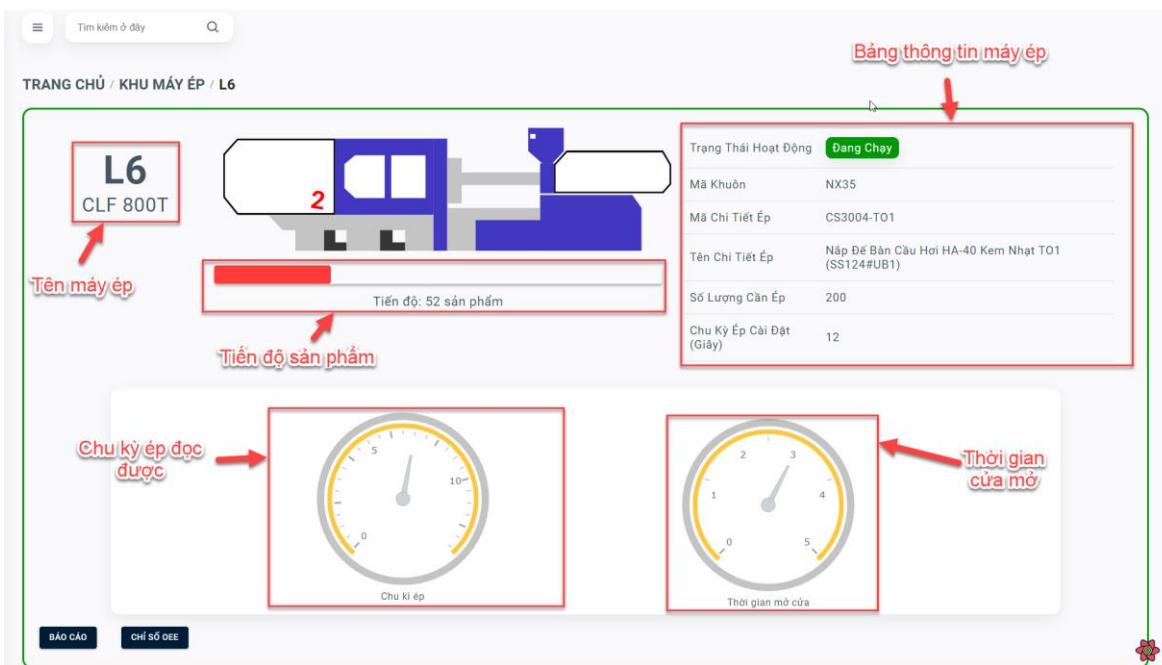
Kết quả phần mềm web nhận dữ liệu và hiển thị ra trang giám sát khu vực máy ép.



Hình 6.3 Kết quả hiển thị dữ liệu tổng quát cập nhật từ khu vực máy ép



Hình 6.4 Kết quả hiển thị dữ liệu tổng quát cập nhật từ khu vực máy ép dưới dạng sơ đồ công ty



Hình 6.5 Trang giám sát chi tiết máy ép công suất L6

6.1.2. Khu vực đóng gói

Ở khu vực đóng gói, nhóm thực hiện đọc dữ liệu đối với 2 cụm đóng gói DG2 và DG3 bằng Windows Service, kết quả thực hiện được thể hiện như các hình dưới.

```

C:\Users\la\Desktop\WPF\Project\Service\service_cha_packing-system\PackingSystemWorkerService\bin\Debug\net6.0\PackingSystemWorkerService.exe
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      Tcp Server Started: True
Info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\la\Desktop\WPF\Project\Service\service_cha_packing-system\PackingSystemWorkerService
Info: HasTransit[0]
      Bus started: http://127.0.0.1:8181/
Info: PackingSystemWorkerService.WorkerServices.SparkplugWorker[0]
      Sparkplug Connected Broker: True
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57604] Connect|DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57604] client connected with machineId: DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57604] client disconnected: Normal
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57606] Connect|DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57606] client connected with machineId: DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57606] client connected with machineId: DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57606] client disconnected: Normal
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] Connect|DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] client connected with machineId: DG3
RasConfigurationMessageReceived
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] Message|DG3|2022-05-05T07:07:07|0|1|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] Message|DG3|2022-05-05T07:07:07|0|2|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] Message|DG3|2022-05-05T07:07:07|0|3|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] client disconnected: Normal
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] Connect|DG3
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57608] client connected with machineId: DG3

```

Hình 6.6 Windows Service chạy ổn định khi đọc dữ liệu cụm đóng gói DG2

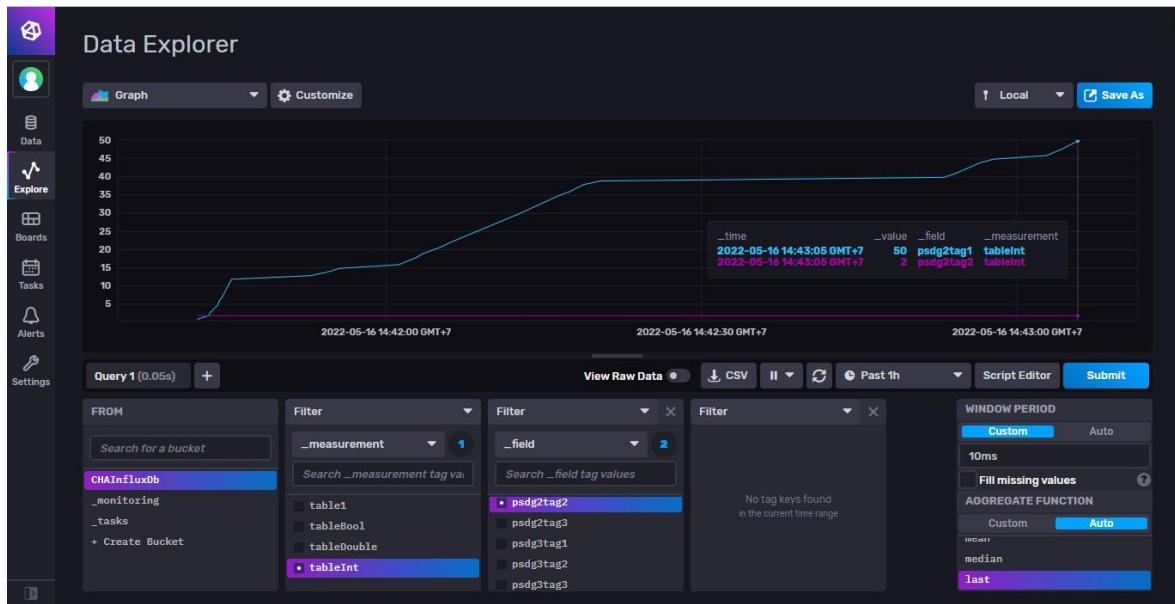
```

Select C:\Users\la\Desktop\WPF\Project\Service\service_cha_packing-system\PackingSystemWorkerService\bin\Debug\net6.0\PackingSystemWorkerService.exe
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|31|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|32|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|33|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|34|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|35|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|36|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|37|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|38|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|39|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|40|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|41|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|42|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|43|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|44|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|45|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|46|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|47|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|48|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|49|2
Info: PackingSystemWorkerService.WorkerServices.TcpWorker[0]
      [10.84.70.220:57612] Message|DG2|2022-05-05T07:07:07|0|50|2

```

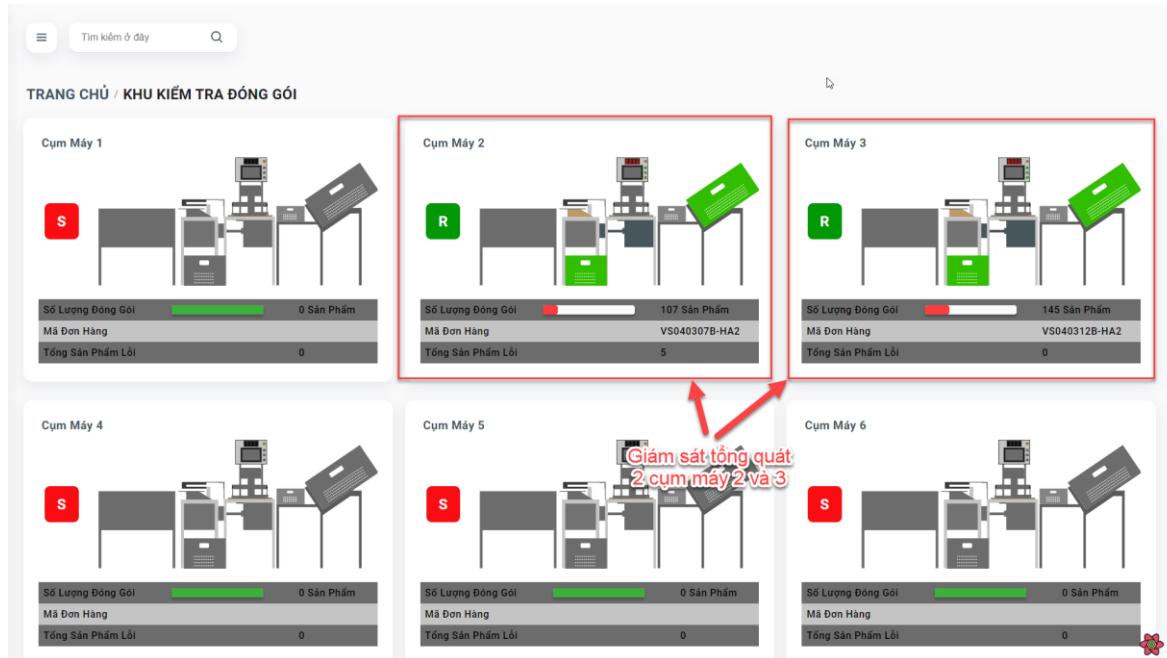
Hình 6.7 Windows Service chạy ổn định khi đọc dữ liệu cụm đóng gói DG2

Vì Windows Service chạy ổn định nên hệ thống có thể ghi vào CSDL InfluxDB và thu được kết quả như 2 hình dưới.

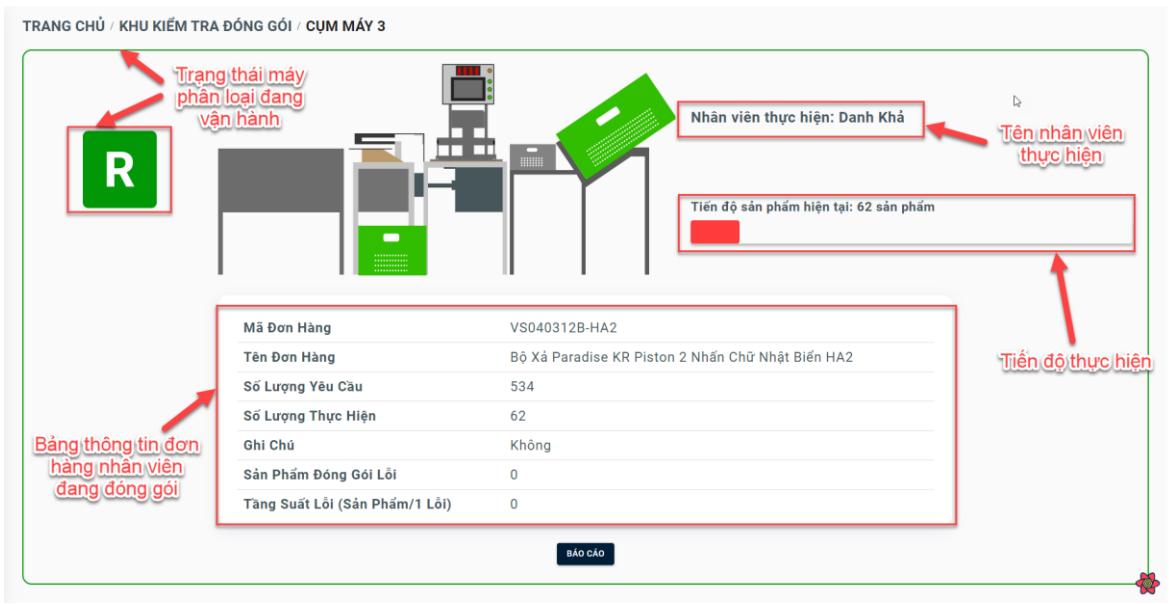


Hình 6.8 Dữ liệu cụm đóng gói ghi nhận được trên database thể hiện dưới dạng biểu đồ

Kết quả phần mềm web nhận dữ liệu và hiển thị ra trang giám sát khu vực đóng gói.



Hình 6.9 Kết quả hiển thị dữ liệu cập nhật từ khu vực đóng gói



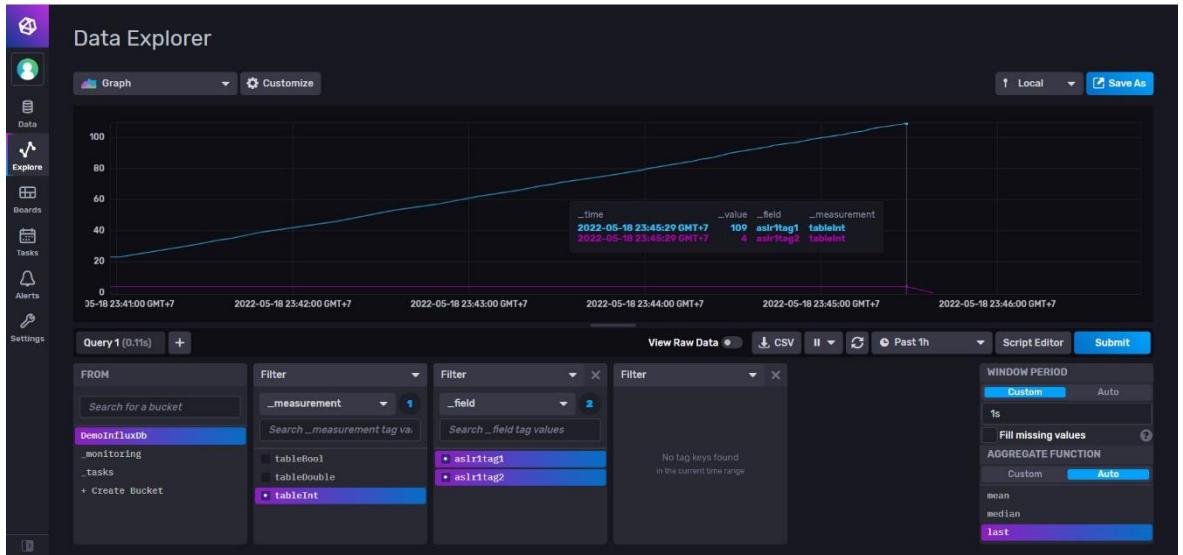
Hình 6.10 Trang giám sát chi tiết cụm máy đóng gói 3

6.1.3. Khu vực lắp ráp

Nhóm thực hiện đọc dữ liệu cụm máy đếm của khu vực lắp ráp, cụ thể là cụm lắp ráp LR1 và thu được kết quả như hình.

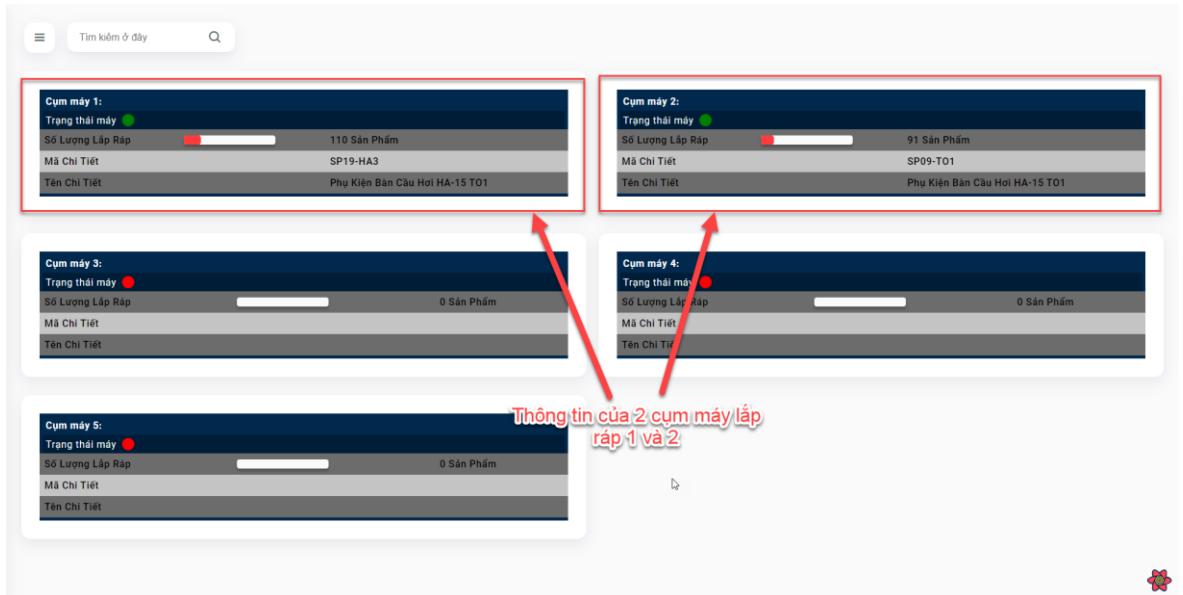
```
xs: E:\STRAIN\27_NEBSACADA_CHA\000_LVTN\WorkerService\Assembly\SystemWorkerService\Assembly\SystemWorkerService\bim\Debug\net5.0\Assembly\SystemWorkerService.exe
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
Connected Broker: True
Info: Assembly\SystemWorkerService.WorkerServices.SparkplugWorker[0]
Sparkplug Connected Broker: True
Info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
Content root path: E:\STRAIN\27_NEBSACADA_CHA\000_LVTN\WorkerService\Assembly\SystemWorkerService\Assembly\SystemWorkerService
Info: Host.Translation[0]
Bus started: https://127.0.0.1:8181/
ConfigurationMessageReceived
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/MachineStatus
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "MachineId": 1, "MachineStatus": 4}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/ValueMessage
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "ItemId": 1, "CurrentValue": 23}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/MachineStatus
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "MachineId": 1, "MachineStatus": 4}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/ValueMessage
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "ItemId": 1, "CurrentValue": 23}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/MachineStatus
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "MachineId": 1, "MachineStatus": 4}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/ValueMessage
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "ItemId": 1, "CurrentValue": 23}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/MachineStatus
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "MachineId": 1, "MachineStatus": 4}
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
topic: AN/LR1/ValueMessage
Info: Assembly\SystemWorkerService.WorkerServices.MqttWorker[0]
message:
{"Timestamp": "2022-05-18T11:23:40", "ItemId": 1, "CurrentValue": 23}
```

Hình 6.11 Windows Service chạy ổn định khi đọc dữ liệu cụm lắp ráp LR1



Hình 6.12 Dữ liệu cụm lắp ráp ghi nhận được trên database thể hiện dưới dạng biểu đồ

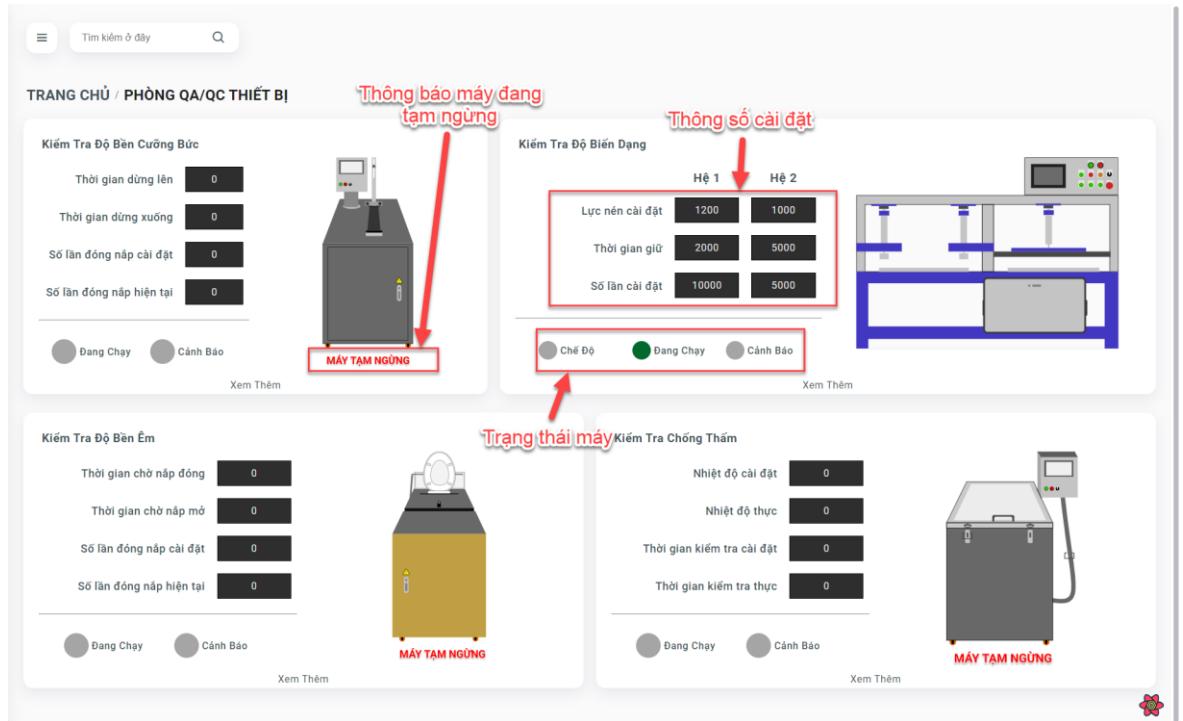
Do Windows Service chạy ổn định, nên nhóm thu được dữ liệu trên database influxDB như đồ thị dưới đây.



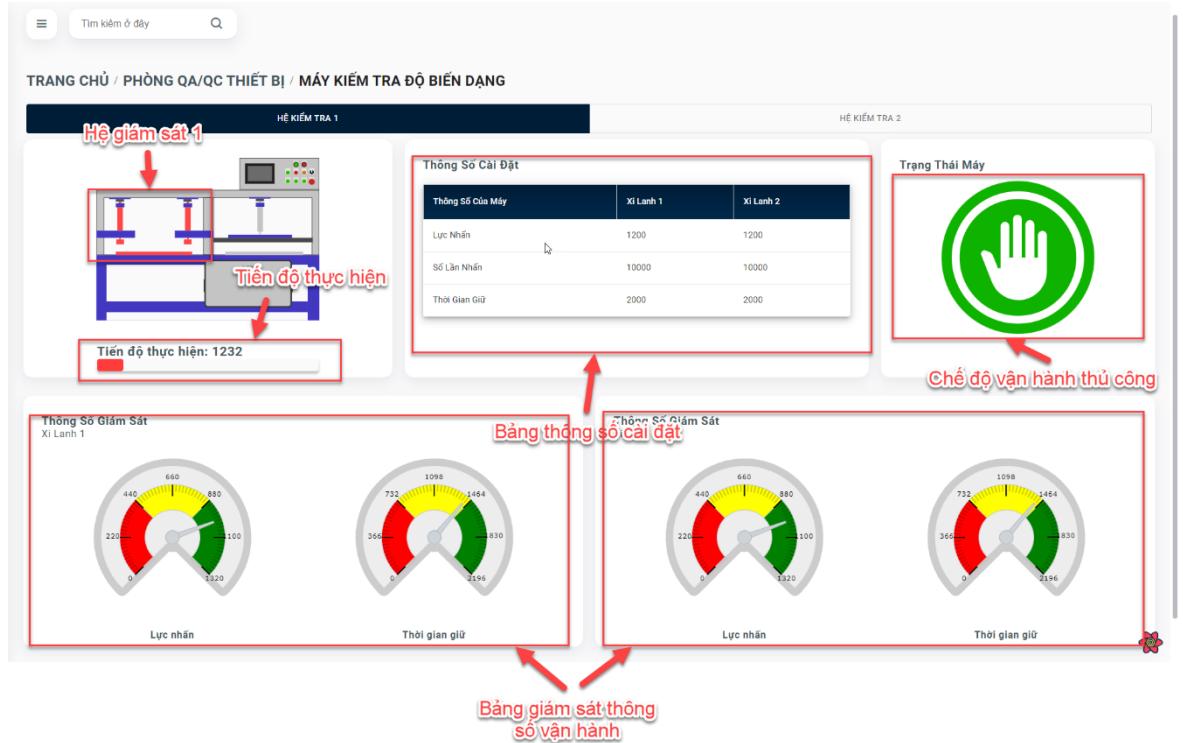
Hình 6.13 Kết quả hiển thị dữ liệu cập nhật từ khu vực lắp ráp

6.1.4. Phòng KTCLSP

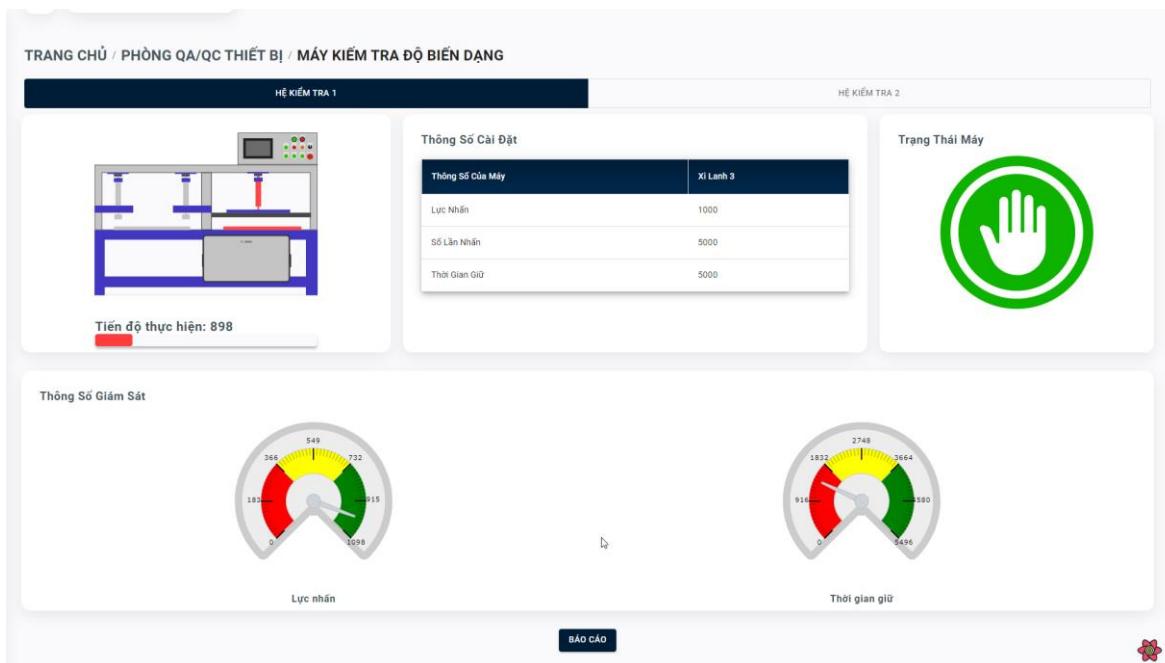
Nhóm thực hiện đọc dữ liệu máy kiểm tra độ biến dạng với hệ 1 và hệ 2 và thu được kết quả như hình.



Hình 6.14 Kết quả hiển thị dữ liệu cập nhật từ phòng QA/QC thiết bị



Hình 6.15 Trang giám sát chi tiết máy kiểm tra độ biến dạng hệ 1



Hình 6.16 Trang giám sát chi tiết máy kiểm tra độ biến dạng hệ 2

6.1.5. Truy xuất dữ liệu báo cáo và tính toán chỉ số OEE

Đối với khu vực kho vận, phần mềm web có thể truy xuất thành công hàng tồn, và ở trang chi tiết, ta có thể quan sát được quá trình xuất nhập của mã sản phẩm theo khoảng thời gian nhất định, tùy theo thông tin người dùng muốn truy xuất. Kết quả được thể hiện như hình dưới.

TRANG CHỦ / KHO VẬN

Tìm Kiếm

Kết Quả

Mã Sản Phẩm	Tên Sản Phẩm	Số Lượng
DVO1-HA1	Van chia nước xi HA1	385
DVO1-HA2	Van chia nước xi HA2	0
ABS001	Kéo ABS CP	0
LT041	Màng co 480*580	0
LT015	Đây dài	300

Xóa tất cả

Xem chi tiết

Hình 6.17 Kết quả truy xuất của trang tổng quát kho vận có chú thích

TRANG CHỦ / KHO VẬN / MÃ CHI TIẾT DVO1-HA1

Tìm Kiếm Theo Ngày

Từ ngày: 05/10/2022

Đến ngày: 05/18/2022

Biểu đồ cập nhật số lượng mã chi tiết DVO1-HA1

Đữ liệu tồn kho 1 ngày

Bảng thông tin xuất nhập kho các ngày

Vị trí của sản phẩm trong kho hiện tại

Thời Gian Xuất Kho Nhập Kho Số Lượng Trong Kho

16/05/2022 12:00:00	65	450	385
---------------------	----	-----	-----

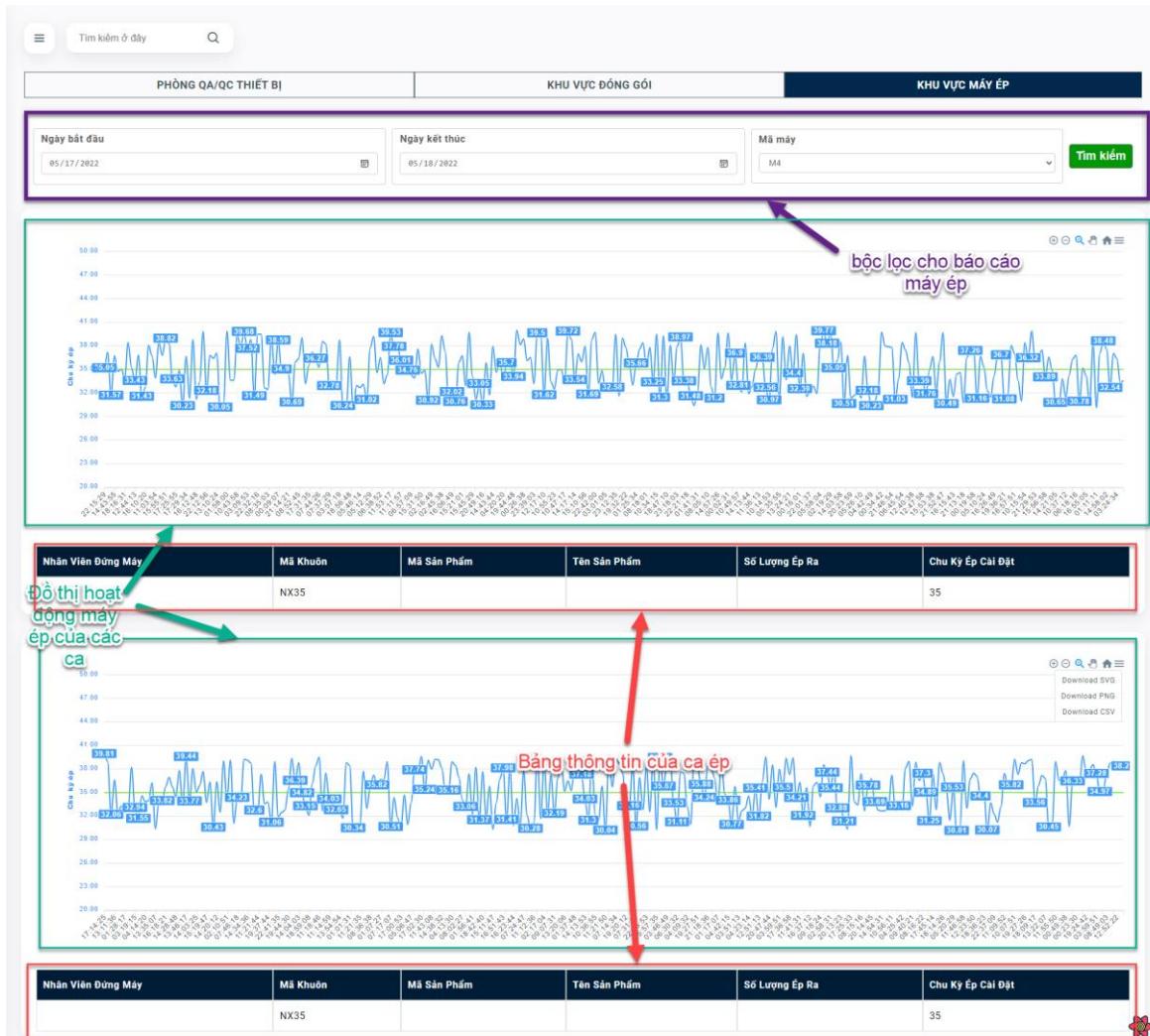
Vị Trí Trong Kho Vị Trí Trong Kệ Số Lượng

A.1.1	1 - 1	5
A.1.1	1 - 1	10
A.1.1	1 - 1	10
A.1.1	1 - 2	5

Hình 6.18 Kết quả truy xuất của trang chi tiết mã sản phẩm kho vận có chú thích

Về phần báo cáo từng khu vực, ở khu vực máy ép, phần mềm có thể đọc các dữ liệu liên quan đến thông số vận từ database của máy chủ, từ đó có thể vẽ đồ

thì thể hiện sự chênh lệch giữa chu kỳ ép tiêu chuẩn và chu kỳ ép thực tế (chu kỳ ép bao gồm chu kỳ ép của máy và thời gian mở cửa). Kết quả được thể hiện như hình dưới đây.



Hình 6.19 Trang báo cáo máy ép kèm chú thích

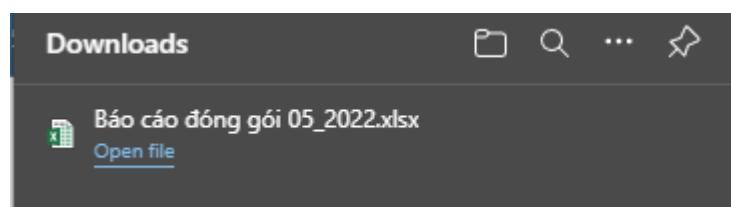
Ở khu vực đóng gói, phần mềm cung cấp tính năng truy xuất lại tiến độ thực hiện theo tháng của nhân viên, thể hiện dữ liệu dưới dạng các cột như file báo cáo tháng được quy định của nhân viên, phần mềm có thể xuất file excel theo dạng báo cáo kết quả. Kết quả thực hiện được thể hiện như hình dưới đây.

Bộ lọc truy xuất và đặt tên file excel

STT	Ngày	Mã Công Việc	Kết quả truy xuất báo cáo	Dvt	Số Lượng	Người Thực Hiện	Thời Gian Thực Hiện	Ghi Chú
Kết quả Tim Kiếm								
1	23/07/2021	HA0201T-HA5	Nắp Bán Cầu HA-03 Biển TT1	Bộ	891	Lê Văn Dũng	22.22	Không
2	23/07/2021	HA0201T-HA2	Nắp Bán Cầu HA-02 Trắng HA4	Bộ	896	Lê Văn Dũng	22.22	Không
1	31/10/2021	HA0201T-C05	Nắp Bán Cầu HA-03 Biển TT1	Cái	891	Lê Văn Lê	22.47	Không
2	31/10/2021	HA0201T-C02	Nắp Bán Cầu HA-03 Biển HA6	Cái	689	Lê Văn Lê	22.47	Không
3	31/10/2021	HA0201T-C05	Nắp Bán Cầu HA-03 Trắng TT1	Bộ	662	Lê Văn Lê	22.47	Không
1	01/02/2022	HA0101T-HA1	Nắp Bán Cầu HA-03 Biển HA6	Bộ	834	Trịnh Hoài	22.72	Không
2	01/02/2022	HA0101T-HA3	Nắp Bán Cầu HA-02 Trắng HA4	Bộ	621	Trịnh Hoài	22.72	Không
3	01/02/2022	HA0101T-HA4	Nắp Bán Cầu HA-02 Trắng HA4	Bộ	661	Trịnh Hoài	22.72	Không
4	01/02/2022	HA0201T-HA3	Nắp Bán Cầu HA-03 Trắng HA2	Cái	846	Trịnh Hoài	22.72	Không
1	09/10/2021	HA0101T-HA3	Nắp Bán Cầu HA-03 Trắng HA4	Bộ	725	Lê Văn Dũng	22.93	Không
2	09/10/2021	HA0101T-HA4	Nắp Bán Cầu HA-03 Trắng TT1	Cái	982	Lê Văn Dũng	22.93	Không
3	09/10/2021	HA0101T-HA3	Nắp Bán Cầu HA-03 Trắng HA4	Cái	978	Lê Văn Dũng	22.93	Không
1	04/07/2021	HA0101T-HA2	Nắp Bán Cầu HA-03 Ngọc TT1	Bộ	985	Hoài Văn Hoài	22.17	Không
2	04/07/2021	HA0201T-HA3	Nắp Bán Cầu HA-03 Trắng TT1	Bộ	942	Hoài Văn Hoài	22.17	Không
3	04/07/2021	HA0201T-C02	Nắp Bán Cầu HA-03 Ngọc HA6	Cái	600	Hoài Văn Hoài	22.17	Không
4	04/07/2021	HA0101T-HA2	Nắp Bán Cầu HA-03 Ngọc TT1	Cái	644	Hoài Văn Hoài	22.17	Không

Trang 1 của 1 << Trang trước Trang sau >>

Hình 6.20 Truy xuất và thể dữ liệu trong trang báo cáo khu vực đóng gói kèm chú thích



Hình 6.21 Xuất file báo cáo thành công

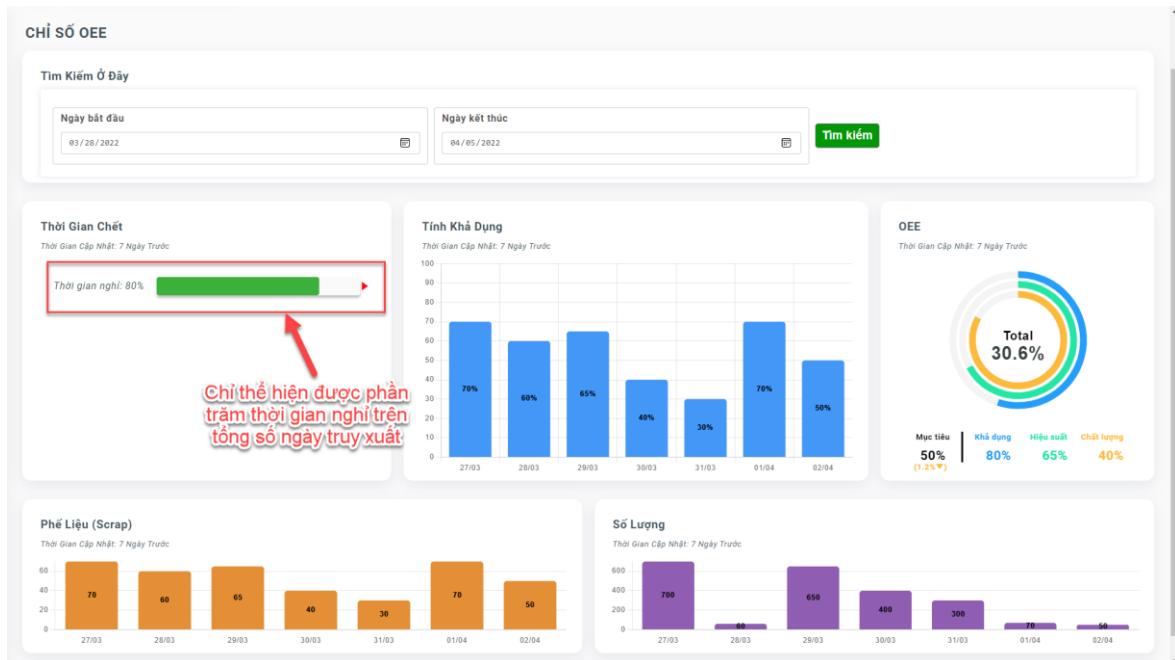
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1																Mã tài liệu	BM07 QT02 SX		
2																Ngày ban hành	01/06/2021		
3																Ngày sửa đổi	01/06/2021		
4																Lần sửa đổi	0		
5																			
6	Tổ Nhóm																		
7	1.Báo cáo kết quả thực hiện																		
8	Stt	Ngày	Mã công việc (mã hàng)		Tên công việc		Đvt	Số lượng	KQ kiểm tra SP	Thiết bị sử dụng	KQ bảo dưỡng	Người thực hiện	Thời gian thực hiện					Ghi chú	
9	1	23/07/2021	HA0201T-HA5	Nắp bát cầu HA-03 biến TT1		Bộ	891					Lê Văn Dũng	22.22						
10		23/07/2021	HA0201T-HA2	Nắp bát cầu HA-07 trắng HA4		Bộ	896					Lê Văn Dũng	22.22						
11																			
12	1	31/10/2021	HA0201T-C02	Nắp bát cầu HA-09 biến TT1		Cái	891					Lê Văn Lê	22.47						
13	2	31/10/2021	HA0201T-C02	Nắp bát cầu HA-09 biến HA6		Cái	689					Lê Văn Lê	22.47						
14	3	31/10/2021	HA0201T-C05	Nắp bát cầu HA-09 trắng TT1		Bộ	662					Lê Văn Lê	22.47						
15																			
16	1	01/02/2022	HA0101T-HA1	Nắp bát cầu HA-03 biến HA6		Bộ	834					Trịnh Hoài	22.72						
17	2	01/02/2022	HA0101T-HA3	Nắp bát cầu HA-02 trắng HA4		Bộ	621					Trịnh Hoài	22.72						
18	3	01/02/2022	HA0101T-HA4	Nắp bát cầu HA-02 trắng TT4		Bộ	661					Trịnh Hoài	22.72						
19	4	01/02/2022	HA0201T-HA3	Nắp bát cầu HA-03 ngọc TT1		Cái	846					Trịnh Hoài	22.72						
20																			
21	1	09/10/2021	HA0201T-HA2	Nắp bát cầu HA-03 ngọc TT1		Bộ	725					Lê Văn Dũng	22.93						
22	2	09/10/2021	HA0101T-HA4	Nắp bát cầu HA-03 ngọc TT1		Cái	982					Lê Văn Dũng	22.93						
23	3	09/10/2021	HA0101T-HA3	Nắp bát cầu HA-03 ngọc TT1		Cái	978					Lê Văn Dũng	22.93						
24																			
25	1	04/07/2021	HA0101T-HA2	Nắp bát cầu HA-03 ngọc TT1		Bộ	985					Hoài Văn Hoài	22.17						
26	2	04/07/2021	HA0201T-HA3	Nắp bát cầu HA-03 trắng TT1		Bộ	942					Hoài Văn Hoài	22.17						
27	3	04/07/2021	HA0201T-C02	Nắp bát cầu HA-03 ngọc HA6		Cái	600					Hoài Văn Hoài	22.17						
28	4	04/07/2021	HA0101T-HA2	Nắp bát cầu HA-03 ngọc TT1		Cái	644					Hoài Văn Hoài	22.17						

Hình 6.22 Hình ảnh thể hiện kết quả truy xuất dữ liệu báo cáo tháng.

Ở báo cáo phòng QA/QC thiết bị, có tổng cộng 6 mục báo cáo bao gồm báo cáo KT độ bền rơi êm, báo cáo KT độ bền cưỡng bức, báo cáo KT Rocktest, báo cáo KT chịu tải tĩnh, báo cáo KT chịu lực uốn và báo cáo KT khả năng chống thấm có chức năng tương tự nhau là xem lại kết quả bài kiểm tra và xuất file excel. Khi truy xuất, kết quả sẽ trả về kết quả đúng nhất với bộ lọc truy xuất. Kết quả thực hiện được thể hiện như hình dưới.

Hình 6.23 Kết quả báo cáo kiểm tra độ biến dạng với bài kiểm tra chịu tải tĩnh kèm ghi chú

Về phần tính toán chỉ số OEE, nhóm đã thực hiện được tính toán các chỉ số quan trọng: chỉ số OEE, biểu đồ “tính khả dụng” theo ngày, biểu đồ thể hiện phế liệu theo ngày, biểu đồ thể hiện kết quả sản xuất theo ngày, tất cả đều thuộc về khu vực máy ép, tuy nhiên đối với thời gian chết, vì giới hạn về mặt phần cứng có sẵn, nhóm chỉ có thể biết được thời gian làm việc và thời gian nghỉ của máy, không thể chỉ ra phần trăm lý do máy tạm dừng. Kết quả thu được thể hiện như hình.



Hình 6.24 Kết quả truy xuất trang báo cáo chỉ số OEE kèm chú thích

6.2. Đánh giá

Nhóm đã đáp ứng được mục tiêu đề ra:

- Đảm bảo đề tài bám sát với dự án đã tham khảo ý kiến từ phía công ty để tăng cao tính ứng dụng.
- Xác thực và phân quyền cho người sử dụng bằng OpenID Connect.
- Xây dựng được ứng dụng web chạy trên nhiều nền tảng.
- Xây dựng được các Windows Service chạy ngầm để thu thập dữ liệu liên tục.
- Đọc và phân tích các dữ liệu có sẵn trong cơ sở dữ liệu của công ty để xuất báo cáo và chỉ số sản xuất phù hợp.

Các mục tiêu chưa đạt được:

- Chưa triển khai ứng dụng lên máy tính công nghiệp để kiểm tra công suất của máy.
- Chưa xây dựng đầy đủ tính năng cảnh báo.

Hạn chế:

- Chưa mô phỏng được chuyển động của các máy trên ứng dụng.
- Chưa mở rộng được cho các khu vực khác: Pha màu, Gia công nắp cầu.
- Thời gian cập nhật dữ liệu các máy kể từ khi gọi hàm bằng phương thức invoke có trễ.
- Không đưa ra các cảnh báo tức thời được cho người sử dụng khi có lỗi xảy ra tại nhà máy.
- Phần mềm web chưa có tính responsive để sử dụng trong điện thoại, tablet...
- Không có dữ liệu riêng cho từng mục đích xuất, phần mềm phải sàng lọc và tính toán dữ liệu.
- Chỉ sử dụng web được trong phạm vi nhà máy, không thể sử dụng từ xa vì phần mềm sử dụng mạng LAN là chủ yếu.

6.3. Hướng phát triển

- Tích hợp với các phần mềm ERP như Odoo để có thể bám sát với kế hoạch sản xuất hơn.
- Tăng tải của máy chủ trong trường hợp công ty mở rộng tin học hóa, số hóa toàn bộ nhà máy.
- Tích hợp với phần mềm Mobile App để thông báo cho người sử dụng trong trường hợp có sự kiện cảnh báo xảy ra.
- Xây dựng tính năng responsive hoặc PWA cho ứng dụng để có truy cập trên mọi phương tiện.
- Mở rộng giám sát quy trình kiểm tra chất lượng ở các khâu: lắp ráp, đóng gói, ép máy.
- Tích hợp hệ thống điều khiển từ xa cho khu vực máy ép và phải phù hợp với người sử dụng.

TÀI LIỆU THAM KHẢO

- [1] "ASP.NET: Open-source web framework for .NET," *Microsoft*. [Online]. Available: <https://dotnet.microsoft.com/apps/aspnet>. [Accessed: 15-Feb-2022].
- [2] Bradygaster, "Overview of ASP.NET core SIGNALR," *Microsoft Docs*. [Online]. Available: <https://docs.microsoft.com/vi-vn/aspnet/core/signalr/introduction?view=aspnetcore-3.1>. [Accessed: 01-Mar-2022].
- [3] Dotnet, "Dotnet/mqtt.net: Mqtt.net is a high performance .net library for MQTT based communication. It provides a MQTT client and a MQTT server (broker). the implementation is based on the documentation from <http://mqtt.org/>," *GitHub*. [Online]. Available: <https://github.com/dotnet/MQTTnet>. [Accessed: 28-Feb-2022].
- [4] "Get started with INFLUXDB OSS 2.2," *InfluxDB OSS 2.2 Documentation*. [Online]. Available: <https://docs.influxdata.com/influxdb/v2.2/>. [Accessed: 01-Mar-2022].
- [5] Influxdata, "Influxdata/influxdb-client-csharp: Influxdb 2.x C# client," *GitHub*. [Online]. Available: <https://github.com/influxdata/influxdb-client-csharp>. [Accessed: 01-May-2022].
- [6] Jchristn, "JCHRISTN/SuperSimpleTcp: Simple wrapper for TCP client and server in C# with SSL Support," *GitHub*. [Online]. Available: <https://github.com/jchristn/SuperSimpleTcp>. [Accessed: 12-Apr-2022].
- [7] Mắt Bão, Chuyên gia SEO và yêu thích lập trình Website, "Socket LÀ GÌ? Khái Niệm Cần Biết Về Giao Thức TCP/IP Và UDP," *MATBAO.NET*, 27-Dec-2021. [Online]. Available: <https://wiki.matbao.net/socket-la-gi-khai-niem-can-biet-ve-giao-thuc-tcp-ip-va-udp/>. [Accessed: 19-May-2022].
- [8] OPCFoundation, "OPCFoundation/UA-.Netstandard: OPC Unified Architecture .NET standard," *GitHub*. [Online]. Available: <https://github.com/OPCFoundation/UA-.NETStandard>. [Accessed: 01-May-2022].
- [9] "Overall Equipment Effectiveness," *OEE*. [Online]. Available: <https://www.oee.com/>. [Accessed: 13-Apr-2022].
- [10] "React – a JavaScript library for building user interfaces," – *A JavaScript library for building user interfaces*. [Online]. Available: <https://reactjs.org/>. [Accessed: 19-May-2022].
- [11] *Redux Toolkit*. [Online]. Available: <https://redux-toolkit.js.org/>. [Accessed: 24-Apr-2022].

- [12] SeppPenner, "Seppenner/SparkplugNet: SparkplugNet is a library to use the sparkplug industrial IOT (IIoT) standard in .net. it uses mqttnet in the background," *GitHub*. [Online]. Available: <https://github.com/SeppPenner/SparkplugNet>. [Accessed: 19-May-2022].
- [13] "Sparkplug topic namespace and State Management - Eclipse." [Online]. Available: <https://www.eclipse.org/tahu/spec/Sparkplug%20Topic%20Namespace%20and%20State%20ManagementV2.2-with%20appendix%20B%20format%20-%20Eclipse.pdf>. [Accessed: 15-Feb-2022].
- [14] "Tìm Hiểu Về Giao Thức MQTT - IOT protocol," *kipalog.com*. [Online]. Available: <https://kipalog.com/posts/Tim-hieu-ve-giao-thuc-MQTT---IoT-protocol>. [Accessed: 02-Apr-2022].