

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**THIẾT KẾ PHÁT TRIỂN PHẦN MỀM HỆ THỐNG
PHÁT HIỆN NGUỒN PHÓNG XẠ NGOÀI KIỂM SOÁT**

Sinh viên thực hiện: NGUYỄN NGỌC HIẾU

Lớp ĐT7 – K59

Giảng viên hướng dẫn: TS. TRẦN QUANG VINH

Hà Nội, 5-2019

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**THIẾT KẾ PHÁT TRIỂN PHẦN MỀM HỆ THỐNG
PHÁT HIỆN NGUỒN PHÓNG XẠ NGOÀI KIỂM SOÁT**

Sinh viên thực hiện: NGUYỄN NGỌC HIẾU

Lớp ĐT7 – K59

Giảng viên hướng dẫn: TS. TRẦN QUANG VINH

Cán bộ phản biện:

Hà Nội, 5-2019

ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho giảng viên hướng dẫn)

Tên giảng viên đánh giá:

Họ và tên Sinh viên: Nguyễn Ngọc HiếuMSSV: 20141528.....

Tên đồ án: **Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát.**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)					
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4 5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4 5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4 5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4 5
Có khả năng phân tích và đánh giá kết quả (15)					
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4 5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4 5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4 5
Kỹ năng viết quyền đồ án (10)					
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4 5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4 5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)					
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế	5			
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.	2			
10c	Không có thành tích về nghiên cứu khoa học	0			
Điểm tổng		/50			
Điểm tổng quy đổi về thang 10					

Nhận xét khác (về thái độ và tinh thần làm việc của sinh viên)

.....

.....

.....

.....

.....

.....

Ngày: ... / ... / 2019

Người nhận xét
(Ký và ghi rõ họ tên)

ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho cán bộ phản biện)

Giảng viên đánh giá:

Họ và tên sinh viên: Nguyễn Ngọc Hiếu MSSV: 20141528

Tên đồ án: **Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát**

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)					
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4 5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4 5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4 5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4 5
Có khả năng phân tích và đánh giá kết quả (15)					
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4 5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng.	1	2	3	4 5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai.	1	2	3	4 5
Kỹ năng viết quyền đồ án (10)					
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4 5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4 5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)					
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế	5			
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest.	2			
10c	Không có thành tích về nghiên cứu khoa học	0			
Điểm tổng		/50			
Điểm tổng quy đổi về thang 10					

Nhận xét khác của cán bộ phản biện

.....

.....

.....

.....

.....

.....

Ngày: ... / ... / 2019

Người nhận xét
(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong những năm gần đây, an ninh nguồn phóng xạ nói riêng và an ninh trong lĩnh vực năng lượng nguyên tử nói chung hiện tại đang là mối quan tâm lớn của cộng đồng quốc tế. Mặc dù đã có các biện pháp quản lý an toàn và an ninh nguồn phóng xạ, nhưng các sự cố mất nguồn phóng xạ vẫn xảy ra thường xuyên trên thế giới và cả ở Việt Nam. Các nguồn phóng xạ bị mất thường sẽ được bán cho cơ sở thu mua phế liệu sắt thép để tái chế. Nếu để xảy ra sự cố nấu chảy nguồn phóng xạ vào sản phẩm kim loại tái chế thì hậu quả sẽ rất nghiêm trọng cho con người và môi trường, gây ảnh hưởng không nhỏ đến kinh tế và ổn định xã hội. Trong khi hầu hết các nước đều có các hệ thống kiểm soát rất nghiêm ngặt cả bằng các hệ thống kỹ thuật và biện pháp quản lý nghiệp vụ để ngăn chặn khả năng ô nhiễm phóng xạ từ tái chế kim loại phế liệu, thì ở Việt Nam vấn đề này vẫn chưa được kiểm soát chặt chẽ, mặc dù đã có những quy định pháp luật hiện hành.

Trước những yêu cầu đó, em quyết định chọn đề tài ***“Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát”***. Đây là một module trong đề tài ứng dụng công nghệ 4.0 để xây dựng hệ phát hiện nguồn phóng xạ nằm ngoài kiểm soát pháp quy trong các cơ sở thu mua và tái chế sắt thép phế liệu. Với những kết quả nghiên cứu đạt được, chúng ta có thể ứng dụng đề tài vào việc theo dõi phát hiện nguồn phóng xạ và đưa ra cảnh báo kịp thời cho các cơ sở, từ đó đảm bảo tăng cường an ninh đối với các nguồn phóng xạ ngoài kiểm soát.

Qua đây, em cũng xin chân thành cảm ơn thầy giáo Tiến sĩ TRẦN QUANG VINH đã trực tiếp định hướng đồ án và tận tình hướng dẫn, tạo điều kiện cần thiết để em có thể hoàn thành đồ án của mình. Em cũng xin chân thành cảm ơn các thành viên trong phòng nghiên cứu SANSLAB – Smart Applications and Network System Laboratory đã tận tình hỗ trợ, giúp đỡ em trong việc giải quyết những vấn đề gặp phải trong quá trình làm đồ án.

Trong quá trình làm đồ án, do kiến thức của em còn nhiều hạn chế, hiểu biết chưa rộng nên đồ án không tránh khỏi thiếu sót. Em rất mong nhận được sự chỉ bảo và nhận xét quý báu của các thầy cô.

Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Em là Nguyễn Ngọc Hiếu, mã số sinh viên 20141528, sinh viên lớp Điện tử 07, khóa 59. Người hướng dẫn là TS. Trần Quang Vinh. Em xin cam đoan toàn bộ nội dung được trình bày trong đề án *Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát* là kết quả quá trình tìm hiểu và nghiên cứu của em. Các dữ liệu được nêu trong đề án là hoàn toàn trung thực, phản ánh đúng kết quả đo đạc thực tế. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ; các tài liệu tham khảo được liệt kê rõ ràng. Em xin chịu hoàn toàn trách nhiệm với những nội dung được viết trong đề án này.

Hà nội, ngày ... tháng ... năm 2019

Người cam đoan

Nguyễn Ngọc Hiếu

TÓM TẮT ĐỒ ÁN

Trong thời gian vừa qua đã xảy ra một số sự cố mất nguồn phóng xạ tại một số địa phương. Khi xảy ra sự cố thì các địa phương hoàn toàn bị động, không có hệ thống giám sát nguồn phóng xạ tại các cơ sở thu mua và các cơ sở tái chế sắt thép phế liệu để có ngay một kênh thông tin quan trọng phục vụ công tác cảnh báo tự động trực tuyến (online).

Xuất phát từ thực tế trên, trong đồ án này, em đã xây dựng một phần mềm hệ thống quản lý đáp ứng được yêu cầu xác thực thiết bị và quản trị hệ thống phát hiện phóng xạ; phân tích các quy trình nghiệp vụ phát hiện và xử lý phát hiện nguồn phóng xạ ngoài kiểm soát; tự động cảnh báo qua SMS, Email; phân tích, báo cáo và thống kê số liệu. Ngoài ra, hệ thống còn cung cấp một bộ API để phát triển phần mềm trên smartphone.

Qua đó, đề tài của em sẽ được chia ra thành các mục dưới đây:

- **CHƯƠNG 1: Đặt vấn đề**

Chương này đưa ra những mục chính là lí do chọn đề tài, mục tiêu mà đề tài hướng đến cũng như các phương pháp được sử dụng trong đề tài.

- **CHƯƠNG 2: Cơ sở lý thuyết**

Chương này đưa ra những kiến thức và công cụ cần có để xây dựng một website mà em tìm hiểu được để phục vụ cho những vấn đề nêu ra ở chương 1.

- **CHƯƠNG 3: Phân tích thiết kế hệ thống**

Chương này đưa ra những phân tích về yêu cầu thiết kế chi tiết và các quy trình xây dựng hệ thống phần mềm.

- **CHƯƠNG 4: Kết quả thực nghiệm**

Chương này trình bày những kết quả đã đạt được của đề tài khi được đưa vào sử dụng.

- **CHƯƠNG 5: Kết luận và hướng phát triển**

Chương này đưa ra kết luận về những kết quả làm được, chưa làm được và hướng phát triển trong tương lai của đề tài.

ABSTRACT

During the past, there have been some incidents of losing radioactive sources in some localities. When incidents occur, these localities are completely passive, there is no system for monitoring radioactive sources at purchasing facilities and recycling facilities for scrap steel to immediately have an important information channel to serve automatic online warning (on-line).

Due to this fact, in this project, I designed a management system software to meet the requirements of device authentication and radiation detection system management; analysis of business processes for detection and processing of out-of-control detection of radioactive sources; automatically alert via SMS, Email; analysis, reporting and statistics. In addition, the system also provides APIs set to develop software on smartphones.

There by, my topic is divided into the following sections:

- CHAPTER 1: Overview of the topic

This chapter provides the main sections: the reasons for choosing the topic , the target that the topic is directed to and the methods used in the topic.

- CHAPTER 2: Theoretical basis

This chapter provides the knowledge and tools needed to build a website that I learn to serve the problems raised in chapter 1.

- CHAPTER 3: Analysis requires the design and construction of software systems

This chapter provides analysis of detailed design requirements and software system construction processes.

- CHAPTER 4: Experimental results

This chapter presents the results of the project when it is established.

- CHAPTER 5: Conclusion and Development Strategy

This chapter provides conclusions about the results of the work done, not yet done and the future development strategy of the topic.

MỤC LỤC

LỜI NÓI ĐẦU.....	v
LỜI CAM ĐOAN.....	vi
TÓM TẮT ĐỒ ÁN.....	vii
ABSTRACT	viii
MỤC LỤC.....	ix
DANH MỤC HÌNH VẼ	xi
DANH MỤC BẢNG BIỂU	xiii
DANH MỤC CÁC TỪ VIẾT TẮT.....	xiv
CHƯƠNG 1 ĐẶT VẤN ĐỀ.....	1
1.1 Tổng quan đề tài	1
1.2 Mục tiêu	1
1.3 Các phương pháp sử dụng trong nghiên cứu và thiết kế.....	2
1.4 Phương hướng giải quyết	2
1.4.1 Xây dựng website	2
1.4.2 Xây dựng công cụ xử lý dữ liệu	3
1.5 Kết luận.....	3
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	4
2.1 Cơ sở lý thuyết.....	4
2.1.1 Ngôn ngữ lập trình C#.....	4
2.1.2 .NET Framework	5
2.1.3 ASP.NET MVC	6
2.1.4 .NET Core	10
2.1.5 CSHTML.....	12
2.1.6 JQuery Ajax.....	12
2.1.7 SQL.....	13
2.2 Các công cụ hỗ trợ.....	16
2.2.1 Visual Studio	16
2.2.2 Visual Studio Code	16
2.2.3 Microsoft SQL Server Management Studio	17

2.2.4 Postman	19
2.3 Kết luận.....	20
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG	21
3.1 Phân tích yêu cầu.....	21
3.1.1 Yêu cầu chức năng	21
3.1.2 Yêu cầu phi chức năng	21
3.2 Thiết kế hệ thống.....	22
3.2.1 Tổng quan hệ thống	22
3.2.2 Sơ đồ use – case.....	22
3.2.3 Sơ đồ activity	34
3.2.4 Thiết kế cơ sở dữ liệu	39
3.2.5 Thiết kế bộ API phát triển ứng dụng trên Smartphone	46
3.3 Kết luận.....	48
CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM	50
4.1 Nhận dữ liệu từ thiết bị truyền thông.....	50
4.2 Giám sát trực tuyến.....	50
4.3 Thống kê	52
4.4 Quản trị.....	54
4.4.1 Quản lý quyền.....	54
4.4.2 Lịch sử hoạt động.....	55
4.4.3 Quản lý tài khoản.....	56
4.4.4 Quản lý cơ sở.....	58
4.5 Cấu hình	60
4.6 Service gửi SMS.....	61
4.7 API.....	62
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	64
TÀI LIỆU THAM KHẢO	65

DANH MỤC HÌNH VẼ

Hình 2.1: Kiến trúc của .NET Framework.....	5
Hình 2.2: Minh hoạ cấu trúc lập trình web	6
Hình 2.3: Minh hoạ mô hình MVC	7
Hình 2.4: Mô hình trong .NET	8
Hình 2.5: Cơ chế hoạt động của ASP.NET MVC.....	9
Hình 2.6: Vị trí của .NET Core trong mô hình phát triển .NET	11
Hình 2.7: Cấu trúc dữ liệu trong SQL Server	18
Hình 3.1: Sơ đồ tổng quan website.....	22
Hình 3.2: Sơ đồ use – case quản trị viên	23
Hình 3.3: Sơ đồ use – case người dùng cấp Bộ	30
Hình 3.4: Sơ đồ use – case người dùng cấp Sở.....	31
Hình 3.5: Sơ đồ use – case người dùng cấp Cơ sở.....	33
Hình 3.6: Sơ đồ activity chức năng gửi và xác thực dữ liệu.....	34
Hình 3.7: Sơ đồ activity chức năng cảnh báo	35
Hình 3.8: Sơ đồ activity chức năng cấu hình cảnh báo	36
Hình 3.9: Giao diện chức năng cấu hình cảnh báo.....	36
Hình 3.10: Giao diện chức năng thêm mới cấu hình cảnh báo.....	37
Hình 3.11: Sơ đồ activity chức năng tự động gửi tin nhắn cảnh báo qua SMS, Email	38
Hình 3.12: Sơ đồ activity chức năng phân tích, báo cáo và thống kê	39
Hình 3.13: Mô hình thực thể liên kết nhóm hệ thống	40
Hình 3.14: Mô hình thực thể liên kết nhóm chức năng.....	43
Hình 3.15: Sơ đồ tổng quan API	48
Hình 4.1: Kết quả nhận dữ liệu từ thiết bị truyền thông.....	50
Hình 4.2: Giao diện đăng nhập.....	51
Hình 4.3: Giao diện xác thực 2 lớp	51
Hình 4.4: Giao diện giám sát trực tuyến.....	52
Hình 4.5: Giao diện chi tiết marker thiết bị	52
Hình 4.6: Giao diện chức năng thống kê	53

Hình 4.7: Màn hình thông báo lỗi	53
Hình 4.8: Biểu đồ thống kê lịch sử tham số.....	54
Hình 4.9: Giao diện quản lý quyền.....	54
Hình 4.10: Giao diện gán chức năng cho quyền	55
Hình 4.11: Giao diện lịch sử hoạt động.....	56
Hình 4.12: Giao diện quản lý tài khoản.....	56
Hình 4.13: Giao diện thêm mới tài khoản.....	57
Hình 4.14: Giao diện quản lý cơ sở.....	58
Hình 4.15: Giao diện thêm mới cơ sở.....	59
Hình 4.16: Giao diện thêm mới thiết bị.....	60
Hình 4.17: Giao diện cấu hình cảnh báo.....	60
Hình 4.18: Giao diện thêm mới cấu hình cảnh báo.....	61
Hình 4.19: Service gửi tin nhắn.....	62
Hình 4.20: Tin nhắn cảnh báo được gửi từ server đến số điện thoại cảnh báo.....	62
Hình 4.21: Giao diện kiểm thử API.....	63

DANH MỤC BẢNG BIỂU

Bảng 2-1: Ngôn ngữ định nghĩa dữ liệu	15
Bảng 2-2: Ngôn ngữ thao tác dữ liệu.....	15
Bảng 2-3: Ngôn ngữ điều khiển dữ liệu	15
Bảng 2-4: Các chức năng của Visual Studio Code	17
Bảng 3-1: Bảng đặc tả use – case đăng nhập	24
Bảng 3-2: Bảng đặc tả use – case quản lý người sử dụng	24
Bảng 3-3: Bảng đặc tả use – case quản lý quyền và phân quyền cho người dùng	25
Bảng 3-4: Bảng đặc tả use – case xem biểu đồ thống kê lịch sử tham số	26
Bảng 3-5: Bảng đặc tả chi tiết use – case quản lý cấu hình và cảnh báo	26
Bảng 3-6: Bảng đặc tả use – case quản lý thiết bị.....	27
Bảng 3-7: Bảng đặc tả chi tiết use – case xem bản đồ giám sát trực tuyến.....	28
Bảng 3-8: Bảng đặc tả chi tiết use – case quản lý hoạt động người dùng	28
Bảng 3-9: Bảng đặc tả chi tiết use – case quản lý người dùng cấp Sở.....	30
Bảng 3-10: Bảng đặc tả chi tiết use – case quản lý người dùng cấp Cơ sở.....	32
Bảng 3-11: Bảng đặc tả chi tiết use – case xem cảnh báo	32
Bảng 3-12: Bảng AppUsers	40
Bảng 3-13: Bảng AppRoles	41
Bảng 3-14: Bảng Functions.....	41
Bảng 3-15: Bảng AppUserRoles	41
Bảng 3-16: Bảng Permissions	42
Bảng 3-17: Bảng ActivityLogs	42
Bảng 3-18: Bảng Enterprises	43
Bảng 3-19: Bảng Decentralizations	44
Bảng 3-20: Bảng Projectors	44
Bảng 3-21: Bảng DataPackets.....	45
Bảng 3-22: Bảng WarningProfiles	45
Bảng 3-23: Bảng WarningNews.....	46

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
IOT	Internet of Things	Internet vạn vật
CSDL		Cơ sở dữ liệu
MVC	Model View Controller	
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
RDBMS	Relational Database Management System	Hệ quản trị cơ sở dữ liệu quan hệ
UML	Unified Modeling Language	Ngôn ngữ mô hình hoá thống nhất

CHƯƠNG 1 ĐẶT VẤN ĐỀ

Trong chương 1, em sẽ trình bày khái quát về đề tài, mục tiêu mà đề tài hướng đến cũng như các phương pháp được sử dụng để hoàn thành được đề tài.

1.1 Tổng quan đề tài

Đề tài “*Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát*” là đề tài nghiên cứu và ứng dụng công nghệ thông tin vào trong hệ thống IOT. Trong cuộc cách mạng công nghiệp 4.0, IoT là một trong các xu thế, với Internet of Things, tất cả mọi thứ đều được kết nối với nhau qua mạng internet, người dùng có thể kiểm soát mọi đồ vật của mình qua mạng mà chỉ bằng một thiết bị thông minh, chẳng hạn như smart phone, tablet, PC hay thậm chí chỉ bằng một smart watch nhỏ bé trên tay. Với những thành tựu mà IOT mang lại, có thể nói Internet of Things chính là xu hướng của tương lai. [1]

Với phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát, người dùng có thể theo dõi và giám sát sự thay đổi vị trí và thông số đo đạc của thiết bị, nhận cảnh báo từ những thiết bị thuộc quyền quản lý, đồng thời hệ thống có trách nhiệm xử lý và gửi dữ liệu giám sát theo thời gian thực đến các cơ sở quản lý thiết bị, các cơ quan chức năng hoặc cán bộ phụ trách an toàn bức xạ của Sở Khoa học Công nghệ.

Phần mềm được xây dựng dựa trên những công nghệ tiên tiến như sử dụng ngôn ngữ lập trình C# .NET, ASP.NET Core, hệ quản trị cơ sở dữ liệu Microsoft SQL Server.

1.2 Mục tiêu

Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát. Dựa trên những mục đích trên, em đưa ra những mục tiêu và nhiệm vụ nghiên cứu sau:

- Tìm hiểu tổng quan về các công nghệ được sử dụng trong thiết kế xây dựng phần mềm
- Tìm hiểu và phối hợp với phần truyền thông để đảm bảo thông tin nhận được là chính xác và liên tục
- Xây dựng phần mềm với những tính năng xác thực và bảo mật, tăng tính an toàn cho người sử dụng phần mềm
- Xây dựng phần mềm với khung thiết kế chuẩn, tăng tính kế thừa và tái sử dụng đối với những project trong tương lai.

1.3 Các phương pháp sử dụng trong nghiên cứu và thiết kế

Để đảm bảo thực hiện đúng tiến độ cũng như hoàn thành tốt các công việc đã đề ra, em đã sử dụng một số phương pháp sau:

- Phương pháp đọc - hiểu lý thuyết: Dựa trên những lý thuyết tổng hợp được từ quá trình học tập trên giảng đường đại học, từ những môn học có liên quan đến đồ án như Ngôn ngữ lập trình, Kỹ thuật phần mềm ứng dụng, Lập trình nâng cao. Những kiến thức này giúp em rất nhiều trong việc xây dựng thiết kế phần mềm và nâng cao hiệu suất lập trình.
- Phương pháp thực hành: Xây dựng mô hình CSDL, cấu trúc các module dựa trên các đề tài nghiên cứu đã được thực hiện trước đó, tiến hành sửa đổi bổ sung nhiều lần sau đó chọn ra hướng phát triển tối ưu nhất.
- Phương pháp tham khảo tài liệu: Tham khảo tài liệu từ sách báo, internet, từ kết quả nghiên cứu của các anh khoá trước trong Lab nhằm có được những kiến thức tổng quan, những dẫn chứng chính xác phục vụ cho đề tài.
- Phương pháp quan sát, học hỏi: Quan sát, học hỏi, xin ý kiến – lời khuyên từ các thầy cô, các chuyên gia, bạn bè có chuyên môn trong lĩnh vực lập trình nhằm đưa ra được hướng đi tốt nhất và tránh mắc phải những khuyết điểm không đáng có.

1.4 Phương hướng giải quyết

Dựa trên những mục tiêu và nhiệm vụ nghiên cứu của đề tài, trong phần này em xin đưa ra những phương hướng giải quyết về mặt thiết kế, công nghệ để đáp ứng được yêu cầu của phần mềm.

1.4.1 Xây dựng website

Đối với bất kỳ một website nào, để vận hành trơn tru và hiệu quả đều phải chia ra thành các phần : Back - end, Front – end và CSDL. Trong phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát, website được chia ra làm các mục sau:

- Back – end: Được xây dựng trên nền tảng ASP. NET Core 2.1, một mã nguồn mở được phát triển bởi Microsoft với nhiều thư viện tích hợp giúp việc xây dựng website ngày càng dễ dàng, đồng thời tăng tính bảo mật cho hệ thống.
- Front – end: Sử dụng ngôn ngữ HTML, CSS kết hợp với JQuery – Ajax để tạo hiệu ứng, SPA để tăng trải nghiệm người dùng.
- CSDL: Sử dụng hệ quản trị CSDL Microsoft SQL Server 2012 tương thích rất tốt với công nghệ ASP. NET, đảm bảo website chạy ổn định, hiệu quả.

1.4.2 Xây dựng công cụ xử lý dữ liệu

Trên thực tế, dữ liệu sau khi đo đạc từ cảm biến sẽ được gửi lên server thông qua giao thức truyền thông LORAWAN. Vì thế đòi hỏi hệ thống phải có một công cụ nhận và xử lý dữ liệu thô từ thiết bị và đảm bảo độ bảo mật cho nguồn dữ liệu thô này. Chính vì những yêu cầu đó, nhóm đã phát triển một công cụ cho phép xử lý dữ liệu thô nhận được từ thiết bị. Công cụ được viết bằng ngôn ngữ C#, được tích hợp các chức năng giải mã, bóc tách dữ liệu ra bản tin rõ đồng thời đảm bảo nguồn dữ liệu này là chính xác trước khi tham gia vào hệ thống.

1.5 Kết luận

Qua chương 1, em đã giới thiệu cơ bản về đề tài, mục đích thực hiện đề tài cùng với một số yêu cầu và phương pháp cơ bản để thực hiện. Để chuẩn bị những kiến thức để hoàn thành đề tài, em xin trình bày những lý thuyết cơ sở trong chương 2.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

Chương 2 em sẽ trình bày những kiến thức và các công cụ cần có để xây dựng một website mà em tìm hiểu được để phục vụ cho những vấn đề nêu ra ở chương 1.

2.1 Cơ sở lý thuyết

Các kiến thức được sử dụng bao gồm các công nghệ tiên tiến, hiện đại như : Ngôn ngữ lập trình C#, .NET Framework, ASP. NET MVC, .NET Core, CSHTML, JQuery Ajax, SQL.

2.1.1 Ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg, kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC.

Tiêu chuẩn ECMA liệt kê các mục tiêu của việc thiết kế ngôn ngữ C#: [2]

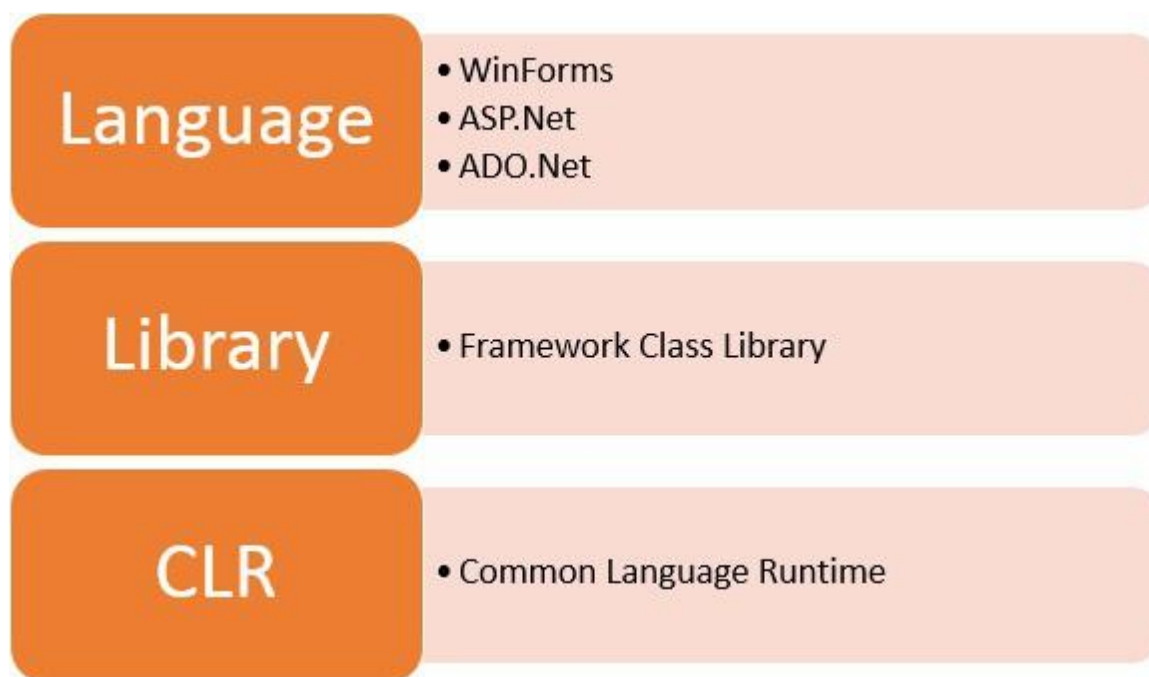
- Ngôn ngữ được dự định là một ngôn ngữ lập trình đơn giản, hiện đại, hướng đến nhiều mục đích sử dụng, và là một ngôn ngữ lập trình hướng đối tượng.
- Ngôn ngữ và việc triển khai đáp ứng các nguyên tắc của ngành kỹ thuật phần mềm như kiểm tra chặt chẽ kiểu dữ liệu, kiểm tra giới hạn mảng, phát hiện các trường hợp sử dụng các biến chưa có dữ liệu, và tự động thu gom rác. Tính mạnh mẽ, sự bền bỉ, và năng suất của việc lập trình là rất quan trọng đối với ngôn ngữ này.
- Ngôn ngữ sẽ được sử dụng để phát triển các thành phần của phần mềm theo hướng thích hợp cho việc triển khai trong các môi trường phân tán.
- Khả năng di chuyển (portability) là rất quan trọng, đặc biệt là đối với những lập trình viên đã quen với C và C++.
- Hỗ trợ quốc tế hóa.
- Ngôn ngữ sẽ được thiết kế để phù hợp với việc viết các ứng dụng cho cả hai hệ thống: hosted và nhúng, từ các phần mềm quy mô lớn, đến các phần mềm chỉ có các chức năng đơn giản.

- Mặc dù các ứng dụng C# có tính kinh tế đối với các yêu cầu về bộ nhớ và chế độ xử lý, ngôn ngữ này không cạnh tranh trực tiếp về hiệu năng và kích thước đối với ngôn ngữ C hoặc assembly.

2.1.2 .NET Framework

.NET framework là một nền tảng phát triển phần mềm do Microsoft phát triển. Framework này tạo ra các ứng dụng chạy trên nền tảng Windows. Phiên bản đầu tiên của .NET framework được phát hành vào năm 2002. .NET framework có thể được sử dụng để tạo cả những ứng dụng dựa trên biểu mẫu (Form-based) và dựa trên Web (Web-based). Các web service cũng có thể được phát triển bằng cách sử dụng .NET framework. Framework cũng hỗ trợ các ngôn ngữ lập trình khác nhau như Visual Basic và C#. Vì vậy, các nhà phát triển có thể lựa chọn ngôn ngữ để phát triển ứng dụng cần thiết.

Kiến trúc của .NET Framework:



Hình 2.1: Kiến trúc của .NET Framework

Những thành phần của .NET Framework bao gồm:

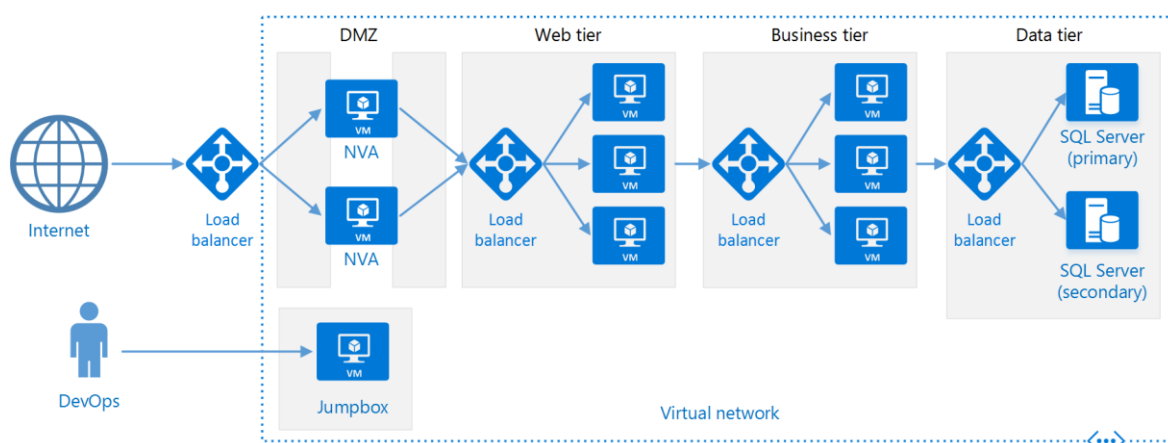
- Môi trường thực thi code CLR (Common Language Runtime)
- Class Library: NET Framework bao gồm một bộ các class libraries. Một class library là một tập hợp các phương thức và các hàm có thể được sử dụng cho mục đích cốt lõi. Ví dụ, có một class library với các phương thức để xử lý tất cả các hoạt động cấp tệp. Vì vậy, có một phương pháp có thể được sử dụng để đọc văn bản từ một tập tin. Tương tự, có một phương pháp để viết văn bản vào một tập

tin. Hầu hết các phương thức được chia thành các vùng tên System. *Hoặc Microsoft.*. (Dấu hoa thị * chỉ có nghĩa là tham chiếu đến tất cả các phương thức nằm trong vùng tên System hoặc Microsoft). Một namespace là một sự tách biệt logic của các phương thức.

- Ngôn ngữ: Các loại ứng dụng có thể được xây dựng trong .NET framework được phân loại rộng rãi thành các loại sau.
 - Winforms: Được sử dụng để phát triển các ứng dụng Forms-based, quá trình này chạy trên end user machine. Notepad là một ví dụ về ứng dụng dựa trên ứng dụng khách.
 - ASP .NET: Được sử dụng để phát triển các ứng dụng dựa trên web, được tạo để chạy trên bất kỳ trình duyệt nào như Internet Explorer, Chrome hoặc Firefox. Ứng dụng Web sẽ được xử lý trên một máy chủ, sẽ được cài đặt Dịch vụ thông tin Internet. Dịch vụ thông tin Internet hoặc IIS là một thành phần của Microsoft được sử dụng để thực thi một ứng dụng Asp.Net. Kết quả của việc thực hiện sau đó được gửi đến các máy khách và kết quả đầu ra được hiển thị trong trình duyệt.
 - ADO.NET: Công nghệ này được sử dụng để phát triển các ứng dụng tương tác với Cơ sở dữ liệu như Oracle hoặc Microsoft SQL Server. Microsoft luôn đảm bảo rằng các .NET framework tuân thủ tất cả các hệ điều hành Windows được hỗ trợ.

2.1.3 ASP.NET MVC

2.1.3.1 Tổng quan mô hình MVC



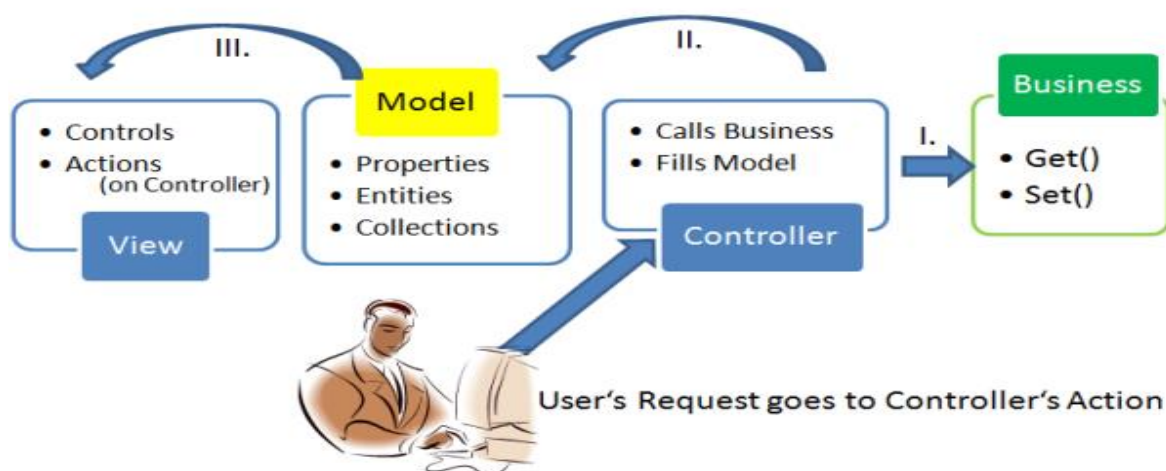
Hình 2.2: Minh họa cấu trúc lập trình web

Trong lập trình web, chúng ta chia ra ba tầng như sau:

- Tầng Presentation: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi...
- Tầng Business Logic: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, ... Tầng này còn di chuyển, xử lý thông tin giữa 2 tầng trên dưới.
- Tầng Data: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu.

Mô hình MVC:

MVC – Model View Controller



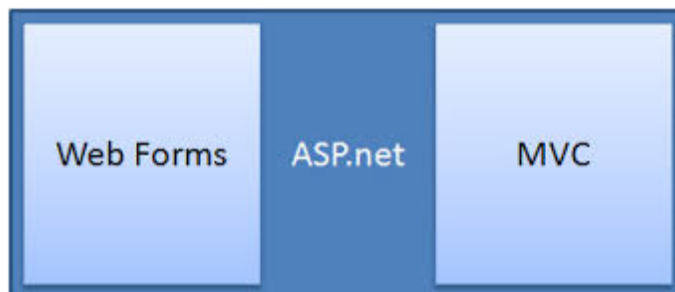
Hình 2.3: Minh họa mô hình MVC

MVC (Model – View - Controller) là một design pattern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller. Giống như trong cấu trúc Three – Tier, mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web, vì vậy giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện

- Controller: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.
- Model: Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...
- View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images... Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

2.1.3.2 Tổng quan mô hình ASP.NET MVC

ASP.NET là một framework lập trình web được cung cấp bởi Microsoft và nó sử dụng ngôn ngữ C# làm ngôn ngữ phát triển. ASP.NET bao gồm 2 model đó là Web Forms và MVC.



Hình 2.4: Mô hình trong .NET

ASP.NET WebForms là một phần của khung ứng dụng ASP.NET. Nó là một trong ba mô hình lập trình khác nhau mà bạn có thể sử dụng để tạo ứng dụng web ASP.NET, phần còn lại là ASP.NET MVC và ASP.NET Web Pages. Web Forms là trang mà người dùng yêu cầu thông qua trình duyệt của họ và hình thức giao diện người dùng (UI) cung cấp cho các ứng dụng của bạn có cái nhìn và cảm nhận. Những trang này được viết bằng cách sử dụng một sự kết hợp của HTML, Server Control và Server code. Khi người dùng yêu cầu từ một trang, nó được biên dịch và thực thi từ máy chủ và sau đó nó được tạo ra các HTML markup (đánh dấu) mà trình duyệt có thể render. ASP.NET WebForms ra đời và nhanh chóng nhận được sự yêu thích của mọi công ty, nó trở nên khá quen thuộc với những tín đồ của .NET. Tuy nhiên, công nghệ này tồn tại rất nhiều nhược điểm dưới đây:

- Web Forms không có sự tách biệt rõ ràng 3 tầng trong mô hình lập trình web.
- View State của WebForm thường khá lớn, đôi khi lên đến cả MB (chứa thông tin về toàn bộ các field của form), làm việc gửi nhận dữ liệu trở nên nặng nề.
- Không quản lý được ID của các element được tạo ra nên khó tích hợp với jQuery (Các bản WebForm mới đã khắc phục lỗi này).
- Page Cycle phức tạp khó hiểu với 7 step và gần 10 event.
- Một số control hỗ trợ viết code nhanh dẫn đến code ẩu (Ví dụ như cho phép kết nối tới Database trực tiếp từ view).
- Không thể áp dụng DI (Dependency Injection), không viết được unit test.
- Giao diện của Web Forms được thiết kế dựa trên những toolbox có sẵn mà web forms cung cấp mà hiện nay đa phần designer sử dụng html, css để thiết kế

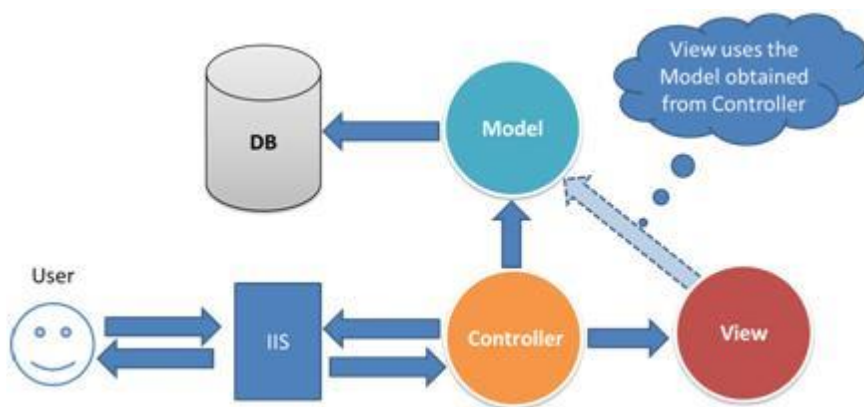
website vì nó gây khó khăn trong việc thiết kế web hiện nay, việc thiết kế trở nên mất linh hoạt.

Chính vì những hạn chế đó dẫn đến sự ra đời của ASP.NET MVC. Đây là một framework lập trình web mới của Microsoft cho phép ứng dụng MVC vào trong .NET. Mới ra đời từ năm 2009, trải qua các lần cập nhật thường xuyên, đến nay ASP.NET đã phát triển mạnh mẽ và không thua kém gì các công nghệ lập trình web sử dụng Java và PHP.

ASP.NET MVC là framework hỗ trợ pattern MVC cho ASP.NET. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.

- Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
- View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.
- Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Cơ chế hoạt động của ASP.NET MVC:



Hình 2.5: Cơ chế hoạt động của ASP.NET MVC

- User gửi yêu cầu tới server bằng cách truyền vào url trong browser.
- Yêu cầu được gửi tới controller. Tại đây controller xử lý yêu cầu và truy xuất dữ liệu sau đó chuyển dữ liệu đến model.
- Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller.
- Controller sẽ giúp dữ liệu chuyển được từ model qua view.

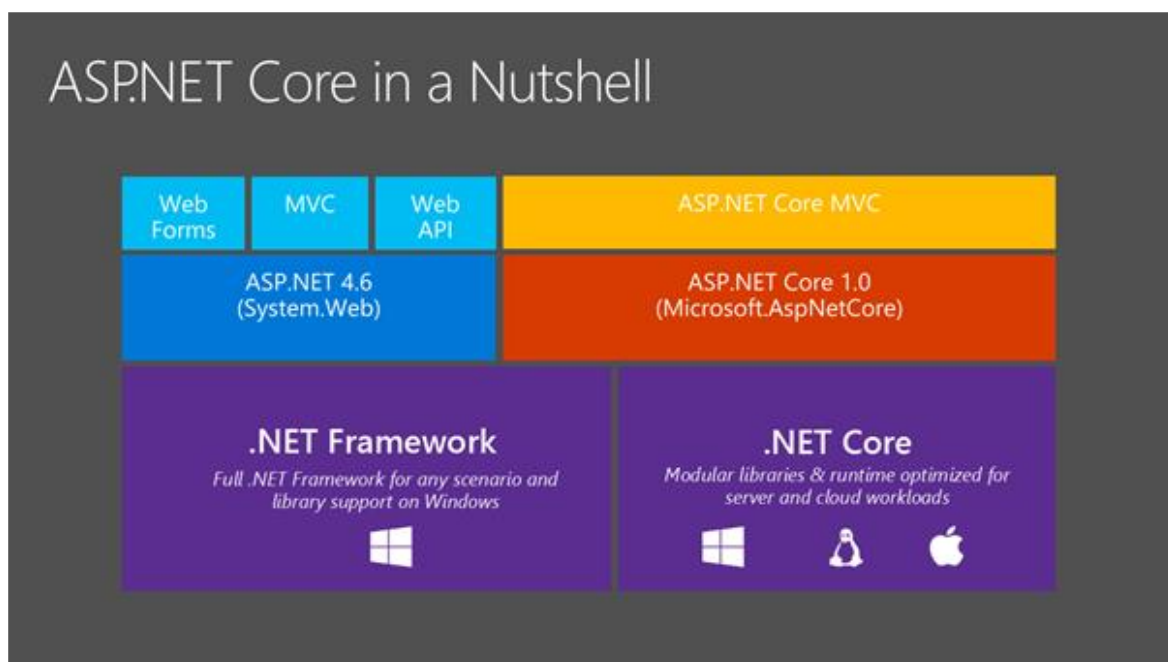
- View là tầng cuối cùng giao tiếp với user, mọi dữ liệu sẽ được chuyển tới user thông qua tầng view.

Ưu điểm của ASP. NET MVC:

- Do sử dụng mô hình MVC nên trong ASP.NET MVC đã tách biệt được các tầng trong mô hình lập trình web vì vậy giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện.
- Giao diện trong ASP.NET MVC sử dụng công nghệ thiết kế web HTML, CSS nên việc thiết kế giao diện trở nên dễ dàng và giúp cho designer linh hoạt trong việc thiết kế.
- ASP.NET MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị giảm.

2.1.4 .NET Core

ASP.NET Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile [3]. Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cụ mà được triển khai trên đám mây (cloud) hoặc chạy on-premise. Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm dẻo trong việc xây dựng giải pháp của bạn. Chúng có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux. Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo em là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.



Hình 2.6: Vị trí của .NET Core trong mô hình phát triển .NET

ASP.NET Core còn được xem là sự kết hợp giữa ASP.NET MVC và ASP.NET Web API tạo thành một mô hình lập trình đơn. Mặc dù được xây dựng mới, ASP.NET Core vẫn có tính tương thích cao với ASP.NET MVC. Hơn nữa, các ứng dụng ASP.NET Core hỗ trợ kiểu phiên bản “side by side”, tức là cùng chạy trên một máy tính với việc lựa chọn nhiều phiên bản ASP.NET Core khác nhau. Điều này là không thể với các phiên bản ASP.NET trước kia.

Hiện nay, có hàng triệu lập trình viên đã và đang sử dụng ASP.NET 4.x để xây dựng các ứng dụng Web. ASP.NET Core là một phiên bản thiết kế lại của ASP.NET 4.x, với nhiều thay đổi kiến trúc giúp framework nhẹ hơn và có tính module nhiều hơn. Do đó, các lập trình viên có thể tiếp tục xây dựng ứng dụng bằng ASP.NET Core với nền tảng hiệu suất và tính tương thích tốt hơn.

ASP.NET Core mang lại các lợi ích như sau:

- Dùng để xây dựng giao diện Web (Web UI) cũng như các API Web.
- Tích hợp các framework phía client hiện đại và các quy trình làm việc phát triển.
- Hệ thống cấu hình sẵn có trên đám mây.
- Tích hợp sẵn những phụ thuộc.
- Đường ống (pipeline) yêu cầu HTTP mang tính module, hiệu suất cao và nhẹ ký.
- Có khả năng lưu trữ (host) ở IIS, Nginx, Apache, Docker, hoặc tự host ở các tiến trình riêng.
- Tạo mới phiên bản app side-by-side với .NET Core.

- Tạo công cụ đơn giản hóa phát triển web hiện đại.
- Khả năng xây dựng, chạy trên Windows, macOS, và Linux.
- Mã nguồn mở và tập trung vào cộng đồng phát triển mã nguồn.

2.1.5 CSHTML

CSHTML là một file HTML được render thành các module ASP. NET, cho phép tạo mã động và sử dụng ngôn ngữ HTML. Nó cung cấp các ánh xạ để trao đổi dữ liệu qua các MVC Engine. Những file có phần mở rộng là .CSHTML sẽ thông báo cho ASP. NET biết rằng có mã trên trang đó cần được thực thi bằng cách sử dụng Razor View Engine trong ASP. NET [4].

Razor View Engine là 1 ngôn ngữ ngắn gọn, rõ ràng và hữu ích, mà nó cho phép bạn tạo ra các giao diện cho ứng dụng ASP.NET MVC trong khi vẫn giữ được sự phân chia rõ ràng, khả năng có thể kiểm tra, và phát triển dựa trên pattern. Các lập trình viên ASP.NET MVC đang tìm kiếm cho mình 1 ngôn ngữ có cú pháp rõ ràng, ngắn gọn, và bây giờ nó đã được xây dựng sẵn với ngôn ngữ quen thuộc là C# [5].

2.1.6 JQuery Ajax

2.1.6.1 Ajax

Ajax (Asynchronous JavaScript and XML) là một bộ công cụ cho phép load dữ liệu từ server mà không yêu cầu tải lại trang. Nó sử dụng chức năng sẵn có XMLHttpRequest (XHR) của trình duyệt để thực hiện một yêu cầu đến server và xử lý dữ liệu server trả về.

Lấy một ví dụ như sau : khi một người dùng viết một nhận xét trên bài viết đăng trên trang Facebook. Sau khi người dùng gửi nhận xét thành công trang Facebook mà người đó đang truy cập cần phải được cập nhật để hiển thị nhận xét vừa mới được tạo ra này. Nếu load lại toàn bộ trang mà người dùng đang truy cập thì sẽ không hiệu quả do tất cả những gì chúng ta muốn là hiển thị nhận xét mới được tạo ra, Ajax được tạo ra để giải quyết vấn đề này, thay vì tải lại toàn bộ trang trình duyệt sẽ chỉ 1 những phần được thay đổi để tiết kiệm thời gian chờ đợi một lượng thông tin lớn về từ server .

Một số ứng dụng sử dụng Ajax như : Gmail , Google Maps , Youtube , Facebook.

2.1.6.2 JQuery Ajax

Jquery cung cấp một số phương thức để thực hiện các chức năng ajax. Chúng ta có thể yêu cầu các text, HTML, XML và JSON từ server sử dụng cả giao thức HTTP GET và HTTP POST, chúng ta cũng có thể lấy dữ liệu từ bên ngoài trực tiếp vào trong phần tử được chọn.

- Phương thức jQuery load(): Lấy dữ liệu từ server và trả về cho phần tử được chọn
- Cú pháp:
- `$(selector).load(URL,data,callback);`
 - URL: Url muốn lấy dữ liệu
 - Data: Cặp key/value gửi đi cùng với yêu cầu
 - Callback: Tên của hàm sẽ được thực thi sau khi phương thức load hoàn thành.
- Phương thức POST trong jQuery Ajax: Có tác dụng gửi request đến server bằng phương thức HTTP POST Request
- Cú pháp:
- `$(selector).post(URL,data,function(data,status,xhr),dataType)`
 - Url: Url muốn gửi request
 - data: không bắt buộc ,là một đối tượng object gồm các key : value sẽ gửi lên server
 - function(data, status , xhr): là function sẽ xử lý khi thực hiện thành công với các parameters:
 - data : bao gồm các dữ liệu trả về từ request
 - status : gồm trạng thái request (“success” , “notmodified” , “error” “timeout” , or “parsererror”)
 - xhr : gồm các đối tượng XMLHttpRequest
 - dataType: là dạng dữ liệu trả về. (text, json, script, xml,html,jsonp)
- Phương thức Get trong jQuery Ajax: Là phương pháp lấy dữ liệu từ server bằng phương thức HTTP GET
- Cú pháp:
- `$.get(URL,data,function(data,status,xhr),dataType) [6]`

2.1.7 SQL

2.1.7.1 SQL là gì?

SQL là viết tắt của Structured Query Language có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. [7]

SQL là ngôn ngữ chuẩn cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

Ngoài ra, họ đang sử dụng các ngôn ngữ khác nhau, chẳng hạn như:

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format)...

2.1.7.2 Tại sao sử dụng SQL

SQL được sử dụng phổ biến vì nó có các ưu điểm sau:

- Cho phép người dùng truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.
- Cho phép người dùng mô tả dữ liệu.
- Cho phép người dùng xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.
- Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện và trình biên dịch trước.
- Cho phép người dùng tạo và thả các cơ sở dữ liệu và bảng.
- Cho phép người dùng tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.
- Cho phép người dùng thiết lập quyền trên các bảng, thủ tục và view.

2.1.7.3 Lịch sử ngắn gọn của SQL

- 1970 – Tiến sĩ Edgar F. “Ted” Codd của IBM được biết đến như là cha đẻ của các cơ sở dữ liệu quan hệ. Ông mô tả một mô hình quan hệ cho cơ sở dữ liệu.
- 1974 – Đã xuất hiện Structured Query Language.
- 1978 – IBM đã làm việc để phát triển các ý tưởng của Codd và đưa ra một sản phẩm có tên System / R.
- 1986 – IBM đã phát triển nguyên mẫu đầu tiên của cơ sở dữ liệu quan hệ và được tiêu chuẩn hóa bởi ANSI (American National Standards Institute). Cơ sở dữ liệu quan hệ đầu tiên được phát hành bởi Relational Software sau này được gọi là Oracle.

2.1.7.4 Các câu lệnh SQL

Các lệnh SQL tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ là CREATE, SELECT, INSERT, UPDATE, DELETE và DROP. Các lệnh này có thể được phân thành các nhóm sau dựa trên bản chất của chúng

- DDL – Ngôn ngữ định nghĩa dữ liệu(Data Definition Language)

Bảng 2-1: Ngôn ngữ định nghĩa dữ liệu

Lệnh	Mô tả
CREATE	Tạo ra một bảng mới, một view của một bảng, hoặc các đối tượng khác trong cơ sở dữ liệu.
ALTER	Sửa đổi một đối tượng cơ sở dữ liệu hiện có, chẳng hạn như một bảng.
DROP	Xoá toàn bộ một bảng, view của một bảng hoặc các đối tượng khác trong cơ sở dữ liệu.

- DML – Ngôn ngữ thao tác dữ liệu(Data Manipulation Language)

Bảng 2-2: Ngôn ngữ thao tác dữ liệu

Lệnh	Mô tả
SELECT	Lấy ra các bản ghi nhất định từ một hoặc nhiều bảng.
INSERT	Tạo một bản ghi.
UPDATE	Chỉnh sửa bản ghi.
DELETE	Xóa bản ghi.

- DCL – Ngôn ngữ điều khiển dữ liệu (Data Control Language)

Bảng 2-3: Ngôn ngữ điều khiển dữ liệu

Lệnh	Mô tả
GRANT	Cung cấp một đặc quyền cho người dùng.
REVOKE	Lấy lại các đặc quyền được cấp từ người dùng.

2.2 Các công cụ hỗ trợ

2.2.1 Visual Studio

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý. [8]

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

2.2.2 Visual Studio Code

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và cá tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính thì là có giấy phép [9].

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").

Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual

Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

Các chức năng phụ thuộc vào ngôn ngữ:

Bảng 2-4: Các chức năng của Visual Studio Code

Chức năng	Ngôn ngữ
Syntax highlighting	Batch, C++, Clojure, CoffeeScript, DockerFile, Elixir, F#, Go, Pug template language, Java, HandleBars, Ini, Lua, Makefile, Objective-C, Perl, PowerShell, Python, R, Razor, Ruby, Rust, SQL, Visual Basic, XML
Snippets	Groovy, Markdown, Nim, PHP, Swift
Tự động hoàn thành mã thông minh	CSS, HTML, JavaScript, JSON, Less, Sass, TypeScript
Cải tiến mã nguồn	C#, TypeScript
Debugging	JavaScript và TypeScript cho Node.js ; C# và F# cho Mono trên Linux và macOS; C và C++ trên Windows, Linux và macOS Python

2.2.3 Microsoft SQL Server Management Studio

2.2.3.1 Khái niệm SQL Server

SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ, (relational database management system viết tắt là RDBMS), hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty IT. SQL Server là một trong 3 công nghệ dữ liệu dẫn đầu hiện nay cùng với Oracle Database và IBM's DB2 [10].

Cũng giống như các phần mềm RDBMS khác, Microsoft SQL Server được xây dựng bên trên lớp SQL - ngôn ngữ lập trình tiêu chuẩn hóa mà quản trị viên cơ sở dữ liệu (DBAs) và các chuyên gia CNTT sử dụng để quản lý cơ sở dữ liệu và truy vấn dữ

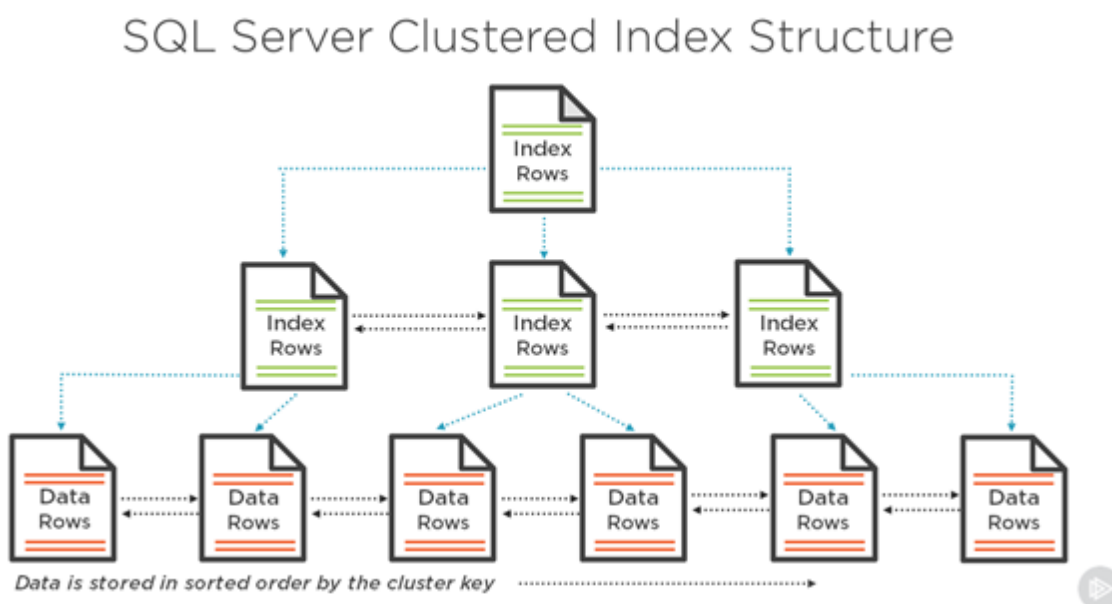
liệu nằm bên trong. SQL Server thường gắn với Transact-SQL (T-SQL), một cài đặt SQL của Microsoft bổ sung một bộ chương trình mở rộng ngôn ngữ lập trình chuẩn.

2.2.3.2 Lịch sử hình thành

Mã SQL Server phiên bản đầu tiên được Sybase Inc phát triển vào những năm 1980. Đến năm 1994, SQL Server chính thức về tay Microsoft và được hãng quảng bá là SQL Server của hệ điều hành Microsoft. Microsoft cũng hỗ trợ phần mềm trên phiên bản Windows NT mới được phát hành sau khi sửa đổi code base OS / 2 16 bit sang triển khai 32-bit với các tính năng bổ sung.

2.2.3.3 Cấu trúc của SQL Server

SQL Server cơ bản dựa trên một cấu trúc bảng biểu, bao gồm các dòng kết nối các phần tử dữ liệu có liên quan trong các bảng khác nhau, do đó loại bỏ nhu cầu lưu trữ dữ liệu ở nhiều vị trí trong cơ sở dữ liệu. Mô hình quan hệ cũng cung cấp các tham chiếu và ràng buộc toàn vẹn nhằm duy trì độ chính xác của dữ liệu; các kiểm tra này là một phần thúc đẩy sự tuân thủ các nguyên tắc về tính nhất quán, tính độc lập và độ tin cậy - được gọi chung là các thuộc tính ACID và được thiết kế để đảm bảo cho các giao dịch cơ sở dữ liệu được xử lý ổn thỏa.



Hình 2.7: Cấu trúc dữ liệu trong SQL Server

2.2.3.4 Mục đích sử dụng SQL Server

- Tạo cơ sở dữ liệu
- Duy trì cơ sở dữ liệu

- Phân tích dữ liệu bằng SSAS - SQL Server Analysis Services.
- Tạo báo cáo bằng SSRS - SQL Server Reporting Services.
- Thực hiện quá trình ETL (Extract-Transform-Load) bằng SSIS SQL Server Integration Services.

2.2.3.5 Các phiên bản của SQL Server

- Enterprise - bản cao cấp nhất với đầy đủ tính năng.
- Standard - ít tính năng hơn Enterprise, sử dụng khi không cần dùng tới các tính năng nâng cao.
- Workgroup - phù hợp cho các công ty lớn với nhiều văn phòng làm việc từ xa.
- Web - thiết kế riêng cho các ứng dụng web.
- Developer - tương tự như Enterprise nhưng chỉ cấp quyền cho một người dùng duy nhất để phát triển, thử nghiệm, demo. Có thể dễ dàng nâng cấp lên bản Enterprise mà không cần cài lại.
- Express - bản này chỉ dùng ở mức độ đơn giản, tối đa 1 CPU và bộ nhớ 1GB, kích thước tối đa của cơ sở dữ liệu là 10GB.
- Compact - nhúng miễn phí vào các môi trường phát triển ứng dụng web. Kích thước tối đa của cơ sở dữ liệu là 4GB.
- Datacenter - thay đổi lớn trên SQL Server 2008 R2 chính là bản Datacenter Edition. Không giới hạn bộ nhớ và hỗ trợ hơn 25 bản cài.
- Business Intelligence - Business Intelligence Edition mới được giới thiệu trên SQL Server 2012. Phiên bản này có các tính năng của bản Standard và hỗ trợ một số tính năng nâng cao về BI như Power View và PowerPivot nhưng không hỗ trợ những tính năng nâng cao về mức độ sẵn sàng như AlwaysOn Availability Groups...
- Enterprise Evaluation - bản SQL Server Evaluation Edition là lựa chọn tuyệt vời để dùng được mọi tính năng và có được bản cài miễn phí của SQL Server để học tập và phát triển. Phiên bản này có thời gian hết hạn là 6 tháng từ ngày cài.

2.2.4 Postman

POSTMAN là một công cụ cho phép chúng ta làm việc với API, nhất là REST. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào. Làm chủ Postman, chúng ta sẽ thấy việc gọi các Rest API (như Facebook, Google, Youtube) không có gì phức tạp cả.

POSTMAN hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE...). POSTMAN cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

Các chức năng cơ bản:

- Cho phép gửi HTTP Request với các method GET, POST, PUT, DELETE.
- Cho phép post dữ liệu dưới dạng form (key-value), text, json
- Hiện kết quả trả về dạng text, hình ảnh, XML, JSON
- Hỗ trợ authorization (Oauth1, 2)
- Cho phép thay đổi header của các request

2.3 Kết luận

Trong chương 2 em đã trình bày những kiến thức lý thuyết về ngôn ngữ lập trình và những công cụ sử dụng để thiết kế phần mềm. Từ những lý thuyết đó, em sẽ trình bày kiến trúc hệ thống phần mềm ở chương 3.

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Từ những cơ sở lý thuyết thu thập ở chương 2, chương này em sẽ trình bày những bước phân tích thiết kế và xây dựng hệ thống phần mềm.

3.1 Phân tích yêu cầu

3.1.1 Yêu cầu chức năng

- Truyền thông:
 - Xây dựng tool cho phép nhận dữ liệu từ thiết bị truyền thông (Lora, GSM) qua giao thức TCP, UDP.
 - Truyền và nhận thông tin đảm bảo tin cậy liên tục.
 - Cho phép truy vấn, thiết lập cấu hình từ xa.
- Hiển thị vị trí và tham số đo đạc từ thiết bị:
 - Hiển thị vị trí trên nền bản đồ (Google Maps).
 - Hiển thị thông tin thiết bị (tên, Imei, cơ sở quản lý, địa chỉ cơ sở).
 - Hiển thị tham số đo đạc từ thiết bị (Gamma, Neutron, Mức pin).
- Cảnh báo:
 - Cảnh báo mức nguy hiểm.
 - Cảnh báo mức rất nguy hiểm.
 - Cảnh báo mức đặc biệt nguy hiểm.
- Phân cấp, phân quyền quản lý:
 - Cấp quốc gia (Cục ATBXHN): toàn quyền truy vấn, theo dõi, quản lý.
 - Cấp tỉnh (Sở KH-CN ở 63 tỉnh thành): quyền tương ứng với tỉnh thành.
 - Cấp cơ sở: theo dõi và truy vấn đối với nguồn thuộc quyền sở hữu
- Xác thực và bảo mật:
 - Cấp quyền và xác thực thiết bị mới tham gia vào hệ thống.
 - Cấp quyền và quản lý người dùng.
 - Bảo mật thông tin, an toàn thông tin và an toàn hệ thống.

3.1.2 Yêu cầu phi chức năng

- Dễ quản lý, sử dụng đối với người dùng cuối.

- Có tính bảo mật cao, dễ bảo trì, nâng cấp đối với đội ngũ kỹ thuật.
- Sử dụng mã hoá các thông tin nhạy cảm như mật khẩu, imei thiết bị, email...
- Đảm bảo website chạy ổn định, tốc độ xử lý nhanh.

3.2 Thiết kế hệ thống

3.2.1 Tổng quan hệ thống

Phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát bao gồm các nội dung chính dưới đây:

- Xác thực thiết bị và quản trị hệ thống phát hiện phóng xạ.
- Phân tích các quy trình nghiệp vụ phát hiện và xử lý phát hiện nguồn phóng xạ ngoài kiểm soát.
- Tự động cảnh báo qua SMS, Email.
- Phân tích, báo cáo và thống kê số liệu.
- API (Application Programming Interface) cung cấp cho phát triển ứng dụng trên smartphone.



Hình 3.1: Sơ đồ tổng quan website

3.2.2 Sơ đồ use – case

Use case là một kỹ thuật được dùng trong kỹ thuật phần mềm và hệ thống để nắm bắt yêu cầu chức năng của hệ thống. Use case mô tả sự tương tác đặc trưng giữa người dùng bên ngoài (actor) và hệ thống. Nó thể hiện ứng xử của hệ thống đối với bên ngoài, trong một hoàn cảnh nhất định, xét từ quan điểm của người sử dụng. Nó mô tả các yêu cầu đối với hệ thống, có nghĩa là những gì hệ thống phải làm chứ không phải mô tả hệ thống làm như thế nào. Tập hợp tất cả Use case của hệ thống sẽ mô tả tất cả các trường hợp mà hệ thống có thể được sử dụng.

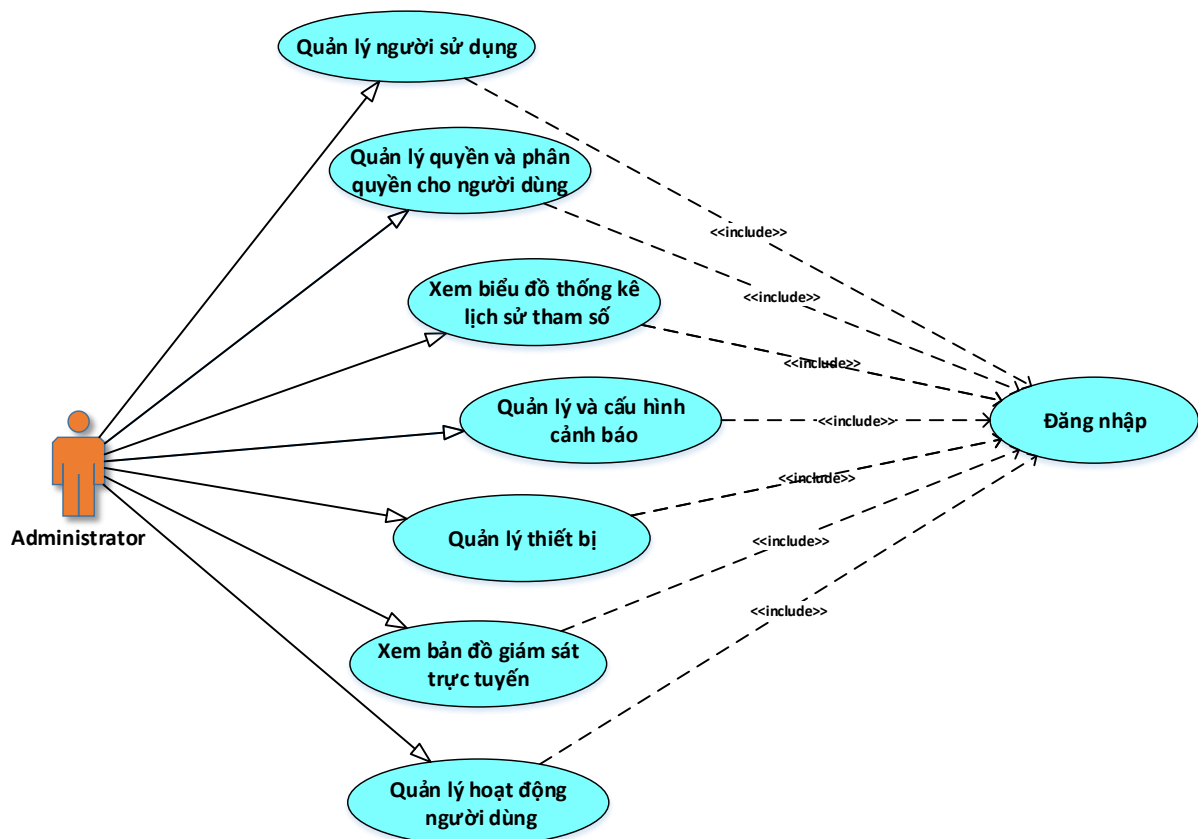
Mỗi use case mô tả cách thức actor tương tác với hệ thống để đạt được mục tiêu nào đó. Một hoặc nhiều kịch bản (scenario) có thể được tạo ra từ mỗi use case, tương

ứng với chi tiết về mỗi cách thức đạt được mục tiêu nào đó. Khi mô tả Use case, người ta thường tránh dùng thuật ngữ kỹ thuật, thay vào đó họ sử dụng ngôn ngữ của người dùng cuối hoặc chuyên gia về lĩnh vực đó. Để tạo ra use case, cần phải có sự hợp tác chặt chẽ giữa người phân tích hệ thống và người dùng cuối. Một trong những cách biểu diễn trực quan phổ biến hiện nay là lược đồ use case của UML.

3.2.2.1 Quản trị viên

Quản trị viên là người có vai trò cao nhất trong hệ thống, là người có đầy đủ các quyền truy cập, thao tác đối với hệ thống. Quản trị viên có các chức năng dưới đây:

- Quản lý người sử dụng.
- Quản lý quyền và phân quyền cho người dùng.
- Xem biểu đồ thống kê lịch sử tham số.
- Quản lý và cấu hình cảnh báo.
- Quản lý thiết bị.
- Xem bản đồ giám sát trực tuyến.
- Quản lý hoạt động người dùng.



Hình 3.2: Sơ đồ use – case quản trị viên

Từ Hình 3.2 ta có bảng đặc tả chi tiết cho các use – case quản trị viên:

- ***Use – case đăng nhập***

Bảng 3-1: Bảng đặc tả use – case đăng nhập

Use - case	Nội dung
Tên use - case	Đăng nhập
Mô tả	Cho phép quản trị viên đăng nhập vào hệ thống, là điều kiện tiên quyết để quản trị viên thao tác với hệ thống.
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Không có
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống
Hậu điều kiện	Quản trị viên đăng nhập thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hiện thị màn hình đăng nhập. 2. Quản trị viên nhập tên tài khoản (Số điện thoại) và mật khẩu. 3. Hệ thống kiểm tra thông tin đăng nhập và quyền hạn. 4. Nếu thành công => nhập mật khẩu xác thực 2 lớp (nếu tài khoản có bật xác thực). 5. Nếu mã xác thực đúng => Hệ thống chuyển hướng đến trang quản trị. 6. Kết thúc use – case.

- ***Use – case quản lý người sử dụng***

Bảng 3-2: Bảng đặc tả use – case quản lý người sử dụng

Use - case	Nội dung
Tên use - case	Quản lý người sử dụng
Mô tả	Cho phép quản trị viên thêm, sửa hoặc xóa đối với người dùng trong hệ thống.
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền hạn quản trị viên

Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng quản lý tài khoản.
Hậu điều kiện	Quản trị viên thêm, sửa, xoá người dùng thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn chức năng quản lý tài khoản. 2. Quản trị viên xem danh sách các tài khoản trong hệ thống (đối với chức năng xem danh sách), hoặc nhập đầy đủ thông tin người dùng (đối với chức năng thêm, sửa người dùng). 3. Hệ thống kiểm tra thông tin quản trị viên đã nhập. 4. Nếu thành công => Hiện thị thống báo thêm/sửa người dùng thành công. 5. Nếu thất bại => Hiện thị lỗi ra màn hình và thông báo nhập lại. 6. Kết thúc use – case.

- *Use – case quản lý quyền và phân quyền cho người dùng*

Bảng 3-3: Bảng đặc tả use – case quản lý quyền và phân quyền cho người dùng

Use - case	Nội dung
Tên use - case	Quản lý quyền và phân quyền
Mô tả	Cho phép quản trị viên thêm sửa xoá quyền và phân quyền cho người dùng trong hệ thống
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng quản lý quyền
Hậu điều kiện	Quản trị viên phân quyền cho người dùng thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn chức năng quản lý quyền. 2. Quản trị viên nhập đầy đủ thông tin quyền (đối với chức năng quản lý quyền) hoặc tích chọn các chức năng (đối với chức năng phân quyền). 3. Hệ thống kiểm tra thông tin quyền và chức năng.

	<p>4. Nếu thành công => Hiển thị thông báo thêm/sửa/phân quyền người dùng thành công</p> <p>5. Nếu thất bại => Hiển thị lỗi ra màn hình và thông báo nhập lại.</p> <p>6. Kết thúc use – case.</p>
--	---

- *Use – case xem biểu đồ thống kê lịch sử tham số*

Bảng 3-4: Bảng đặc tả use – case xem biểu đồ thống kê lịch sử tham số

Use - case	Nội dung
Tên use - case	Xem biểu đồ thống kê lịch sử tham số
Mô tả	Cho phép quản trị viên xem biểu đồ thống kê lịch sử tham số
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng xem biểu đồ thống kê lịch sử tham số
Hậu điều kiện	Quản trị viên xem biểu đồ thống kê thành công
Luồng sự kiện chính	<p>1. Quản trị viên chọn chức năng xem biểu đồ thống kê.</p> <p>2. Quản trị viên chọn các thông tin về cơ sở, thiết bị, ngày thống kê, tham số thống kê được hiển thị dưới dạng dropdownlist và nhấn theo dõi.</p> <p>3. Hệ thống kiểm tra các thông tin đã chọn.</p> <p>4. Nếu thông tin chọn đầy đủ => Hệ thống hiển thị biểu đồ thống kê tham số dưới dạng biểu đồ đường</p> <p>5. Nếu chọn thiếu thông tin hoặc thông tin sai lệch => Hiển thị lỗi ra màn hình và thông báo chọn lại.</p> <p>6. Kết thúc use – case.</p>

- *Use – case quản lý cấu hình và cảnh báo*

Bảng 3-5: Bảng đặc tả chi tiết use – case quản lý cấu hình và cảnh báo

Use - case	Nội dung
Tên use - case	Quản lý cấu hình và cảnh báo

Mô tả	Cho phép quản trị viên xem cảnh báo và cấu hình cảnh báo
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng xem cảnh báo / cấu hình cảnh báo
Hậu điều kiện	Quản trị viên cấu hình cảnh báo thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn chức năng xem cảnh báo / cấu hình cảnh báo. 2. Quản trị viên nhập đầy đủ các thông tin cấu hình cảnh báo (ngưỡng trên, ngưỡng dưới, tên cấu hình, mức cảnh báo...) và chọn Save . 3. Hệ thống kiểm tra các thông tin đã nhập. 4. Nếu thành công => Hệ thống hiển thị thông báo cấu hình thành công 5. Nếu thất bại => Hiển thị lỗi ra màn hình và thông báo nhập lại. 6. Kết thúc use – case.

- *Use – case quản lý thiết bị*

Bảng 3-6: Bảng đặc tả use – case quản lý thiết bị

Use - case	Nội dung
Tên use - case	Quản lý thiết bị
Mô tả	Cho phép quản trị viên quản lý thiết bị
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng quản lý thiết bị
Hậu điều kiện	Quản trị viên thêm mới/ sửa thiết bị thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn chức năng quản lý thiết bị

	<p>2. Quản trị viên nhập đầy đủ các thông tin thiết bị (tên thiết bị, imei, application key, application Id...) và chọn Save .</p> <p>3. Hệ thống kiểm tra các thông tin đã nhập.</p> <p>4. Nếu thành công => Hệ thống hiển thị thông báo cấu hình thành công</p> <p>5. Nếu thất bại => Hiển thị lỗi ra màn hình và thông báo nhập lại.</p> <p>6. Kết thúc use – case.</p>
--	--

- ***Use – case xem bản đồ giám sát trực tuyến***

Bảng 3-7: Bảng đặc tả chi tiết use – case xem bản đồ giám sát trực tuyến

Use - case	Nội dung
Tên use - case	Xem bản đồ giám sát trực tuyến
Mô tả	Cho phép quản trị viên xem bản đồ giám sát trực tuyến
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng xem bản đồ giám sát trực tuyến.
Hậu điều kiện	Quản trị viên xem bản đồ thành công
Luồng sự kiện chính	<p>1. Quản trị viên chọn chức năng xem bản đồ giám sát trực tuyến</p> <p>2. Chọn thiết bị để xem thông tin chi tiết thông số đo đặc của thiết bị được hiển thị dưới dạng marker google map.</p> <p>3. Tìm kiếm thiết bị theo vị trí bằng cách nhập địa chỉ vào ô tìm kiếm.</p> <p>4. Kết thúc use – case.</p>

- ***Use – case quản lý hoạt động người dùng***

Bảng 3-8: Bảng đặc tả chi tiết use – case quản lý hoạt động người dùng

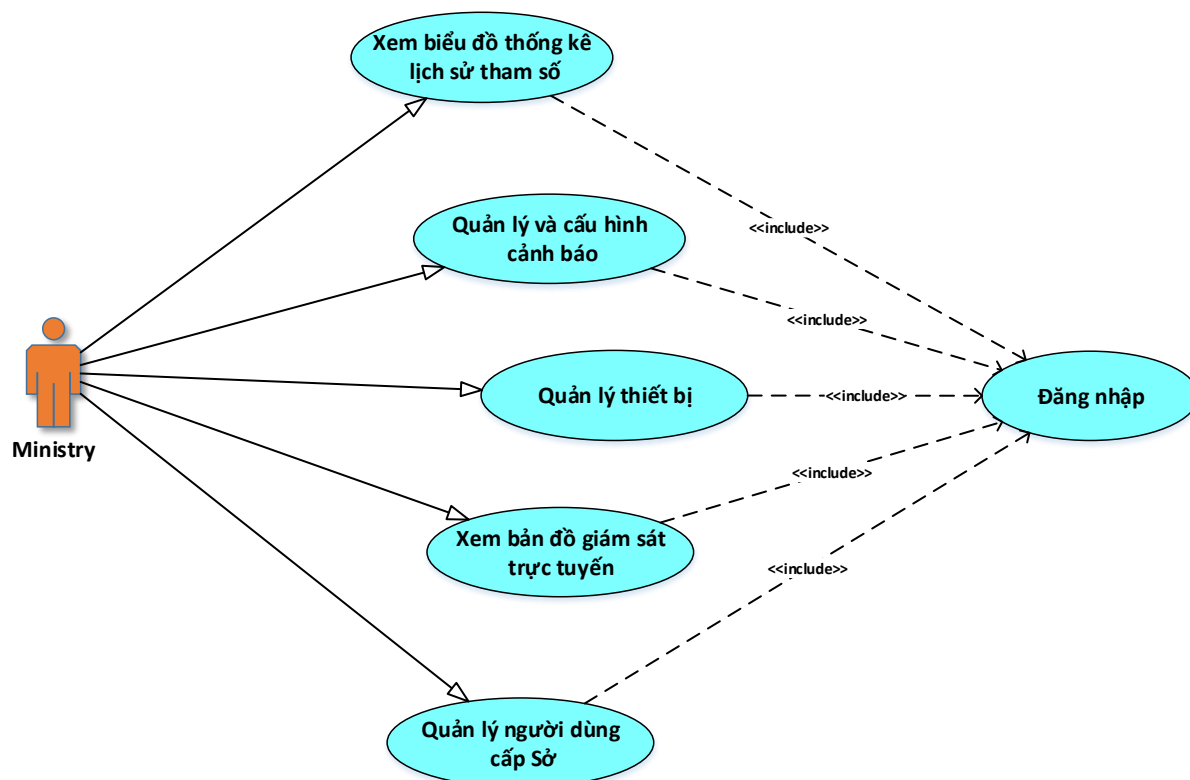
Use - case	Nội dung
Tên use - case	Quản lý hoạt động người dùng

Mô tả	Cho phép quản trị viên quản lý hoạt động người dùng
Tên tác nhân	Quản trị viên
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền quản trị viên
Tiền điều kiện	Quản trị viên phải có tài khoản với quyền quản trị trên hệ thống => chọn chức năng quản lý hoạt động người dùng.
Hậu điều kiện	Quản trị viên xem lịch sử hoạt động người dùng thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn chức năng quản lý hoạt động người dùng 2. Hệ thống hiển thị danh sách lịch sử hoạt động người dùng bao gồm nội dung lịch sử hoạt động, người thao tác và thời gian xảy ra hoạt động . 3. Kết thúc use – case.

3.2.2.2 Người dùng cấp Bộ

Người dùng cấp Bộ là người có quyền cao thứ hai trong hệ thống, vai trò tương đương Cục ATBXHN, có toàn quyền theo dõi, quản lý. Người dùng cấp Bộ có các chức năng dưới đây:

- Xem biểu đồ thống kê lịch sử tham số
- Quản lý và cấu hình cảnh báo
- Quản lý thiết bị
- Xem bản đồ giám sát trực tuyến
- Quản lý người dùng cấp Sở



Hình 3.3: Sơ đồ use – case người dùng cấp Bộ

Từ Hình 3.3 ta có bảng đặc tả chi tiết cho các use – case người dùng cấp Bộ

- *Use – case quản lý người dùng cấp Sở*

Bảng 3-9: Bảng đặc tả chi tiết use – case quản lý người dùng cấp Sở

Use - case	Nội dung
Tên use - case	Quản lý người dùng cấp Sở
Mô tả	Cho phép người dùng cấp Bộ thêm, sửa hoặc xoá đối với người dùng cấp Sở.
Tên tác nhân	Người dùng cấp Bộ.
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền hạn người dùng cấp Bộ
Tiền điều kiện	Người dùng phải có tài khoản với quyền người dùng cấp Bộ trên hệ thống => chọn chức năng quản lý người dùng cấp Sở.
Hậu điều kiện	Người dùng cấp Bộ thêm, sửa, xoá người dùng thành công
Luồng sự kiện chính	1. Người dùng cấp Bộ chọn chức năng quản lý người dùng cấp Sở.

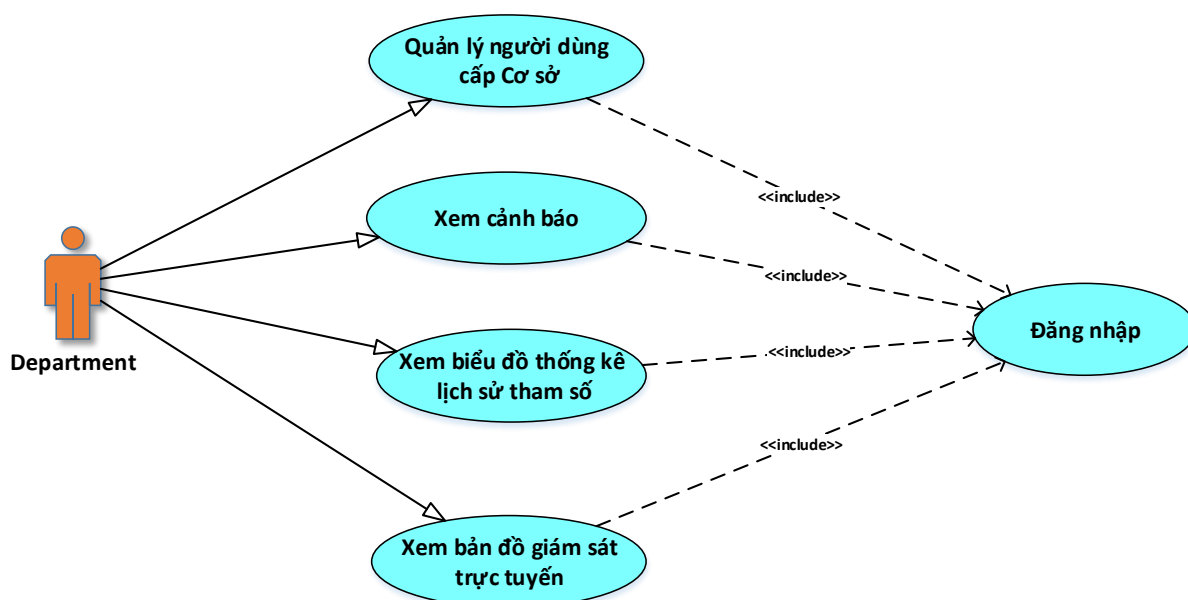
	<p>2. Người dùng cấp Bộ xem danh sách các tài khoản cấp Sở trong hệ thống (đối với chức năng xem danh sách), hoặc nhập đầy đủ thông tin người dùng cấp Sở (đối với chức năng thêm, sửa người dùng).</p> <p>3. Hệ thống kiểm tra thông tin Người dùng cấp Bộ đã nhập.</p> <p>4. Nếu thành công => Hiện thị thông báo thêm/sửa người dùng thành công.</p> <p>5. Nếu thất bại => Hiện thị lỗi ra màn hình và thông báo nhập lại.</p> <p>6. Kết thúc use – case.</p>
--	--

Các use – case còn lại : Xem biểu đồ thống kê lịch sử tham số; Quản lý và cấu hình cảnh báo; Quản lý thiết bị; Xem bản đồ giám sát trực tuyến có bảng đặc tả chi tiết giống với quản trị viên.

3.2.2.3 Người dùng cấp Sở

Người dùng cấp Sở là người có quyền cao thứ ba trong hệ thống, vai trò tương đương các Sở KHCN. Người dùng cấp Sở có các chức năng dưới đây:

- Quản lý người dùng cấp Cơ sở
- Xem cảnh báo
- Xem biểu đồ thống kê lịch sử tham số
- Xem biểu đồ giám sát trực tuyến



Hình 3.4: Sơ đồ use – case người dùng cấp Sở

Từ Hình 3.4 ta có bảng đặc tả chi tiết cho các use – case người dùng cấp Sở.

- *Use – case quản lý người dùng cấp Cơ sở*

Bảng 3-10: Bảng đặc tả chi tiết use – case quản lý người dùng cấp Cơ sở

Use - case	Nội dung
Tên use - case	Quản lý người dùng cấp Cơ sở
Mô tả	Cho phép người dùng cấp Sở thêm, sửa hoặc xóa đối với người dùng cấp Cơ sở.
Tên tác nhân	Người dùng cấp Sở.
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền hạn người dùng cấp Sở
Tiền điều kiện	Người dùng phải có tài khoản với quyền người dùng cấp Sở trên hệ thống => chọn chức năng quản lý người dùng cấp Cơ sở.
Hậu điều kiện	Người dùng cấp Sở thêm, sửa, xóa người dùng thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng cấp Sở chọn chức năng quản lý người dùng cấp Cơ sở. 2. Người dùng cấp Sở xem danh sách các tài khoản cấp Cơ sở trong hệ thống (đối với chức năng xem danh sách), hoặc nhập đầy đủ thông tin người dùng cấp Cơ sở (đối với chức năng thêm, sửa người dùng). 3. Hệ thống kiểm tra thông tin Người dùng cấp Sở đã nhập. 4. Nếu thành công => Hiện thị thông báo thêm/sửa người dùng thành công. 5. Nếu thất bại => Hiện thị lỗi ra màn hình và thông báo nhập lại. 6. Kết thúc use – case.

- *Use – case xem cảnh báo*

Bảng 3-11: Bảng đặc tả chi tiết use – case xem cảnh báo

Use - case	Nội dung
Tên use - case	Xem cảnh báo
Mô tả	Cho phép người dùng cấp Sở xem cảnh báo
Tên tác nhân	Người dùng cấp Sở.

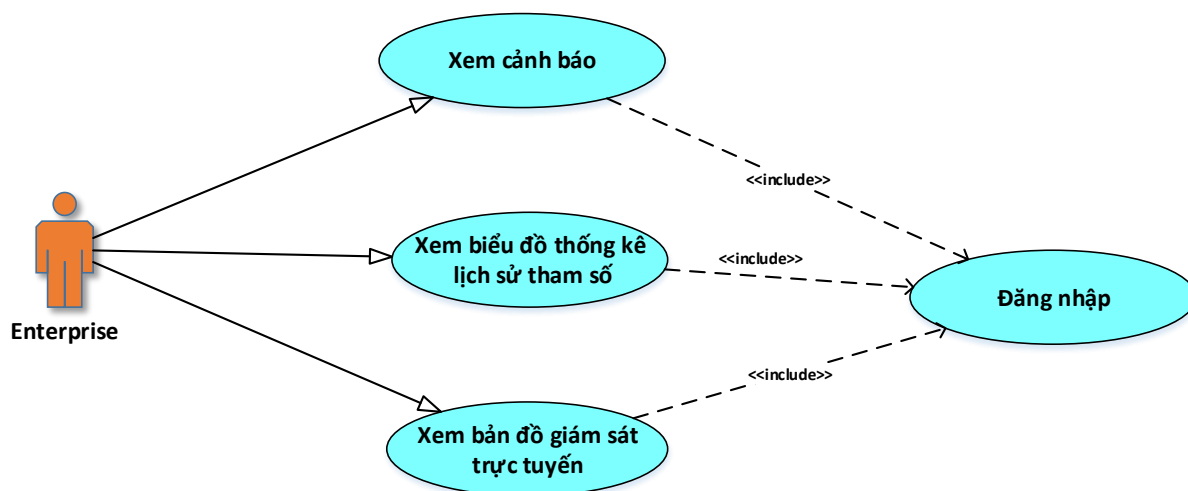
Điều kiện kích hoạt	Đã đăng nhập hệ thống với quyền hạn người dùng cấp Sở
Tiền điều kiện	Người dùng phải có tài khoản với quyền người dùng cấp Sở trên hệ thống => chọn chức năng xem cảnh báo.
Hậu điều kiện	Người dùng cấp Sở xem cảnh báo thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng cấp Sở chọn chức năng xem cảnh báo. 2. Hệ thống hiển thị các cảnh báo bao gồm tên tham số cảnh báo, nội dung cảnh báo, thời gian nhận cảnh báo. 3. Kết thúc use – case.

Các use – case còn lại: Xem biểu đồ thống kê lịch sử tham số, xem bản đồ giám sát trực tuyến có các bảng đặc tả chi tiết với quản trị viên.

3.2.2.4 Người dùng cấp Cơ sở

Người dùng cấp Cơ sở là người có quyền hạn thấp nhất trong hệ thống, tương đương với các cơ sở đặt thiết bị phát hiện nguồn phóng xạ. Người dùng cấp Cơ sở có các chức năng sau đây:

- Xem cảnh báo
- Xem biểu đồ thống kê lịch sử tham số
- Xem bản đồ giám sát trực tuyến



Hình 3.5: Sơ đồ use – case người dùng cấp Cơ sở

Các use – case người dùng cấp Cơ sở có các bảng đặc tả chi tiết giống với người dùng cấp Sở.

3.2.3 Sơ đồ activity

Activity Diagram (Sơ đồ activity) là một mô hình logic dùng để mô hình hoá các hoạt động trong một quy trình nghiệp vụ. Hay có thể hiểu, Activity – Diagram là sơ đồ luồng xử lý của hệ thống. Bao gồm luồng đi của dòng dữ liệu, dòng sự kiện.

Sơ đồ Activity dùng để mô tả các hoạt động trong một chức năng của hệ thống. Hay có thể hiểu là mô tả luồng xử lý của một Use – Case.

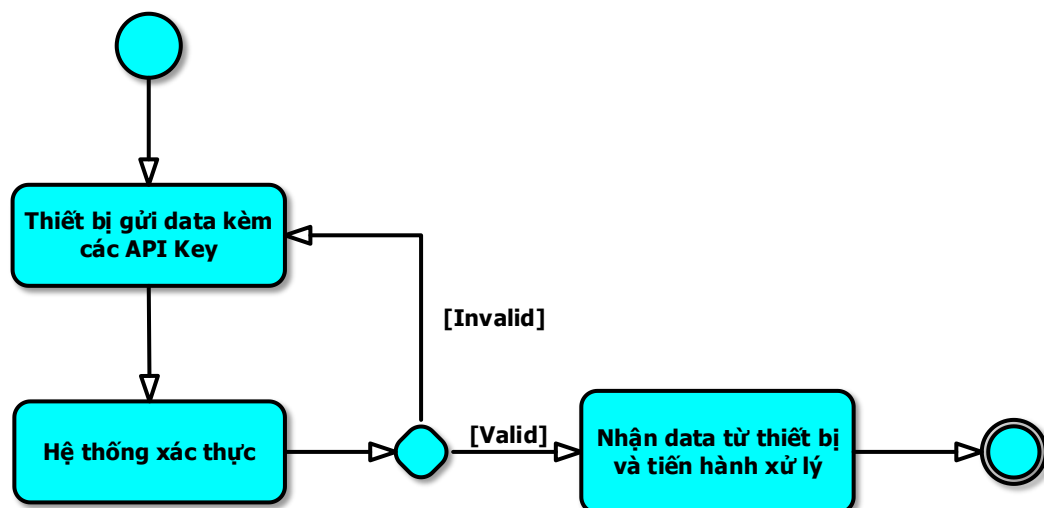
Mô tả hoạt động chính và mối quan hệ giữa các hoạt động này trong quy trình. Hay có thể hiểu là mô tả cả luồng xử lý chính của hệ thống bao gồm các luồng con, luồng xử lý của các Use – Case gom lại mà thành.

Dưới đây là một số sơ đồ activity của hệ thống:

Sơ đồ activity gửi và xác thực dữ liệu

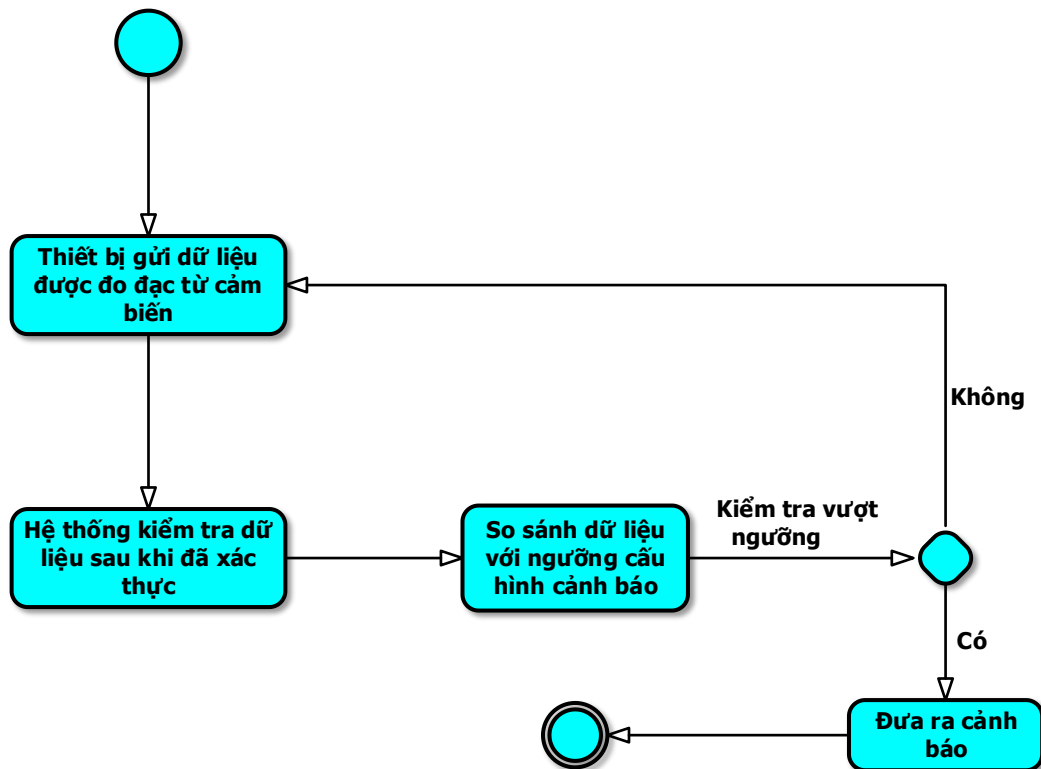
Hình 3.6 mô tả sơ đồ activity chức năng gửi và xác thực dữ liệu, bao gồm các bước sau:

- Bước 1: Thiết bị gửi dữ liệu đã được mã hoá (thường được mã hoá dạng AES và Encode Base 64 String) lên server kèm theo các API Key. Các API Key này có nhiệm vụ xác thực và bắt tay kết nối đến server.
- Bước 2: Tại server, xây dựng một phương thức để xác thực thiết bị gửi lên. Nếu các API Key thiết bị đưa ra hoàn toàn trùng khớp với phía server, dữ liệu sẽ được nhận và tiến hành xử lý. Nếu không, dữ liệu sẽ được loại bỏ và tiến hành quá trình gửi dữ liệu tiếp theo.



Hình 3.6: Sơ đồ activity chức năng gửi và xác thực dữ liệu

Sơ đồ activity chức năng cảnh báo

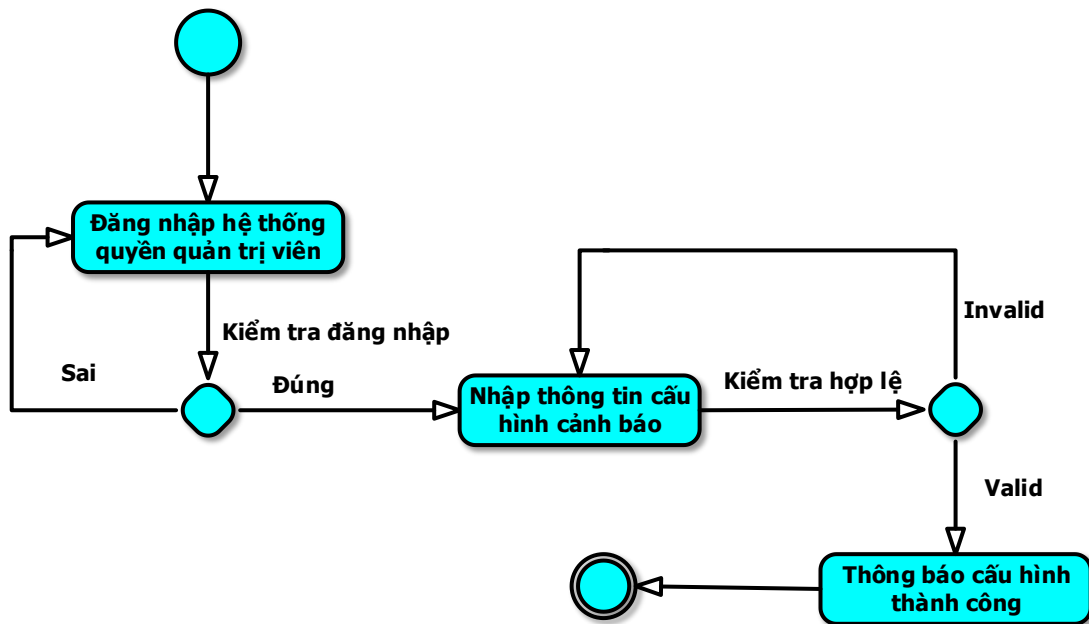


Hình 3.7: Sơ đồ activity chức năng cảnh báo

Hình 3.7 mô tả sơ đồ activity chức năng cảnh báo, bao gồm các bước sau:

- Thiết bị truyền thông gửi dữ liệu đo đạc từ cảm biến qua giao thức LoraWan đến server. Server nhận dữ liệu sau khi đã xác thực thiết bị.
- Tại server, tiến hành so sánh dữ liệu nhận được với cấu hình cảnh báo. Cảnh báo được chia thành 3 cấp độ : Nguy hiểm, Rất nguy hiểm, Đặc biệt nguy hiểm
 - Cấp độ Nguy hiểm: Với cấp độ này, tiến hành gửi tin nhắn kèm theo nội dung cảnh báo nguy hiểm đến cơ sở quản lý đang đặt thiết bị đo. Tin nhắn được gửi qua 2 hình thức : SMS, Email
 - Cấp độ Rất nguy hiểm: Với cấp độ này, tiến hành gửi tin nhắn kèm theo nội dung cảnh báo rất nguy hiểm tới uỷ ban chỉ huy thực hiện UPSC của Tỉnh. Tin nhắn được gửi qua 2 hình thức : SMS, Email.
 - Cấp độ đặc biệt nguy hiểm: Với cấp độ này, tiến hành gửi tin nhắn kèm theo nội dung cảnh báo đặc biệt nguy hiểm tới Cục ATBXHN. Tin nhắn được gửi qua 2 hình thức : SMS, Email.

Sơ đồ activity chức năng cấu hình cảnh báo



Hình 3.8: Sơ đồ activity chức năng cấu hình cảnh báo

Hình 3.8 mô tả sơ đồ activity chức năng cấu hình cảnh báo, bao gồm các bước sau:

- Quản trị viên nhấn vào link đăng nhập trang quản trị, sau đó nhập vào tên đăng nhập và mật khẩu. Nếu tài khoản và mật khẩu trùng khớp => chuyển hướng đến trang quản trị, ngược lại => đưa ra thông báo lỗi.
- Sau khi đăng nhập thành công vào trang quản trị, quản trị viên chọn chức năng cấu hình cảnh báo.

Cấu hình cảnh báo Thêm mới

Show entries Search... Search!

Total records count: 4

Tên	Nội dung	Từ	Đến	Mức cảnh báo	Nhận cảnh báo SMS	Khoảng thời gian cảnh báo(phút)	Trạng thái	
Battery	Pin yếu	0	30	Nguy hiểm	Kích hoạt	60	Kích hoạt	✕
Radiation	Cảnh báo phóng xạ vượt ngưỡng (Đơn vị: micro si vơ trên giờ uSv/h) min : 0.34	10	11.41	Nguy hiểm	Kích hoạt	60	Kích hoạt	✕
Radiation	Cảnh báo phóng xạ vượt ngưỡng, cấp độ rất nguy hiểm	11.41	114.15	Rất nguy hiểm	Kích hoạt	30	Kích hoạt	✕
Radiation	Cảnh báo phóng xạ vượt ngưỡng, cấp độ đặc biệt nguy hiểm	114.15	570.77	Rất nghiêm trọng	Kích hoạt	10	Kích hoạt	✕

Đầu Trước 1 Tiếp Cuối

Hình 3.9: Giao diện chức năng cấu hình cảnh báo

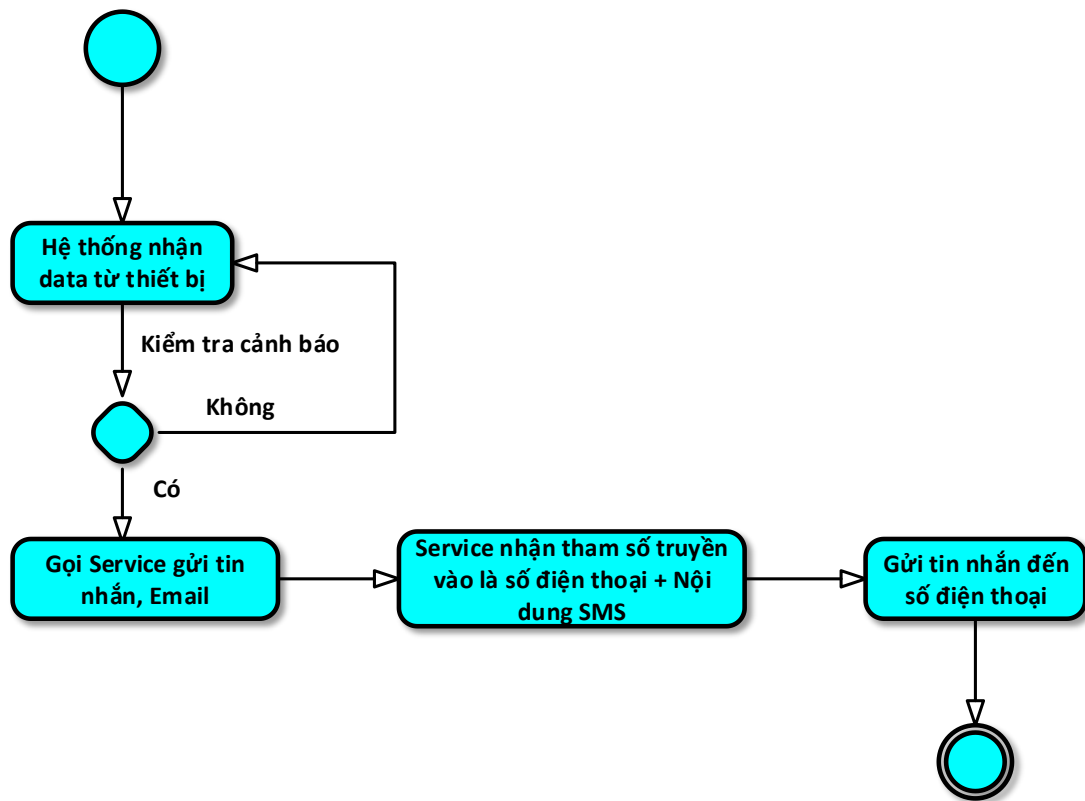
- Quản trị viên nhấn vào button “Thêm mới”. Giao diện trang thêm mới cấu hình được hiển thị ra. Quản trị viên nhập đầy đủ thông tin cấu hình và nhấn “Lưu” để lưu lại thông tin cấu hình cảnh báo. Thông tin cấu hình cảnh báo bao gồm Tên cấu hình, Nội dung cảnh báo, ngưỡng cảnh báo (max, min), Mức cảnh báo (nguy hiểm, rất nguy hiểm, đặc biệt nguy hiểm), Khoảng thời gian giữa hai lần cảnh báo (phút), có cho phép gửi tin nhắn cảnh báo hay không?

The screenshot shows a web form for adding a new alert configuration. It includes the following fields and elements:

- Tên cấu hình**: A text input field for the configuration name.
- Nội dung cảnh báo**: A text input field for the alert content.
- Ngưỡng cảnh báo (Từ)**: A text input field for the alert threshold (From).
- Ngưỡng cảnh báo (Đến)**: A text input field for the alert threshold (To).
- Mức cảnh báo**: A dropdown menu for the alert level.
- Khoảng thời gian cảnh báo(phút)**: A text input field for the alert interval in minutes.
- Gửi tin nhắn cảnh báo?**: A checkbox with a blue checkmark icon, indicating whether to send alert messages.

Hình 3.10: Giao diện chức năng thêm mới cấu hình cảnh báo

Sơ đồ activity chức năng tự động gửi tin nhắn cảnh báo qua SMS, Email



Hình 3.11: Sơ đồ activity chức năng tự động gửi tin nhắn cảnh báo qua SMS, Email

Hình 3.11 mô tả sơ đồ activity chức năng tự động gửi tin nhắn cảnh báo qua SMS, Email, bao gồm các bước sau:

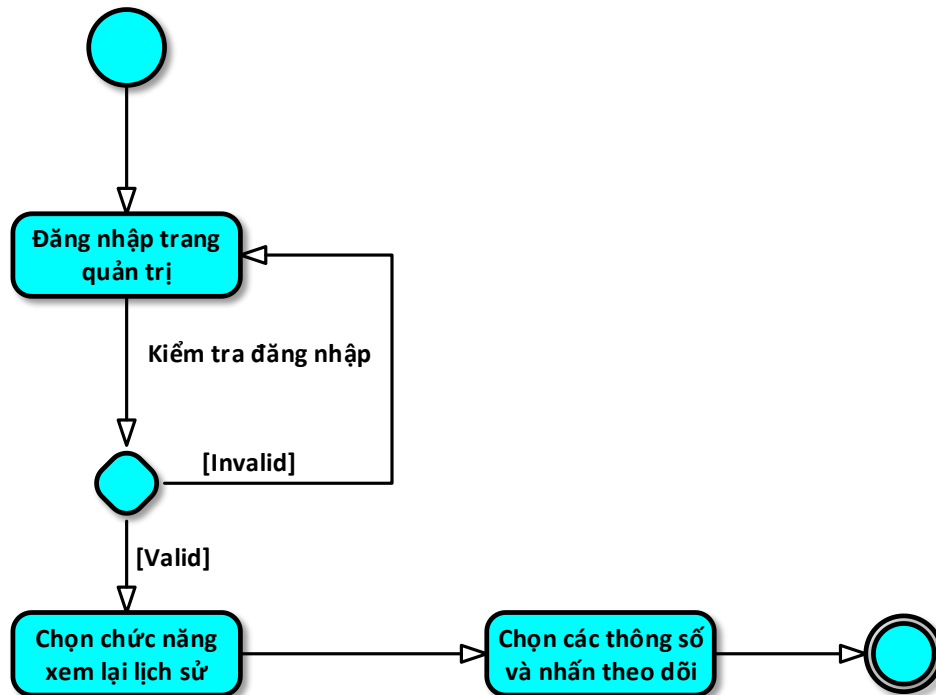
- Hệ thống nhận data từ thiết bị sau khi đã xác thực thiết bị.
- Kiểm tra dữ liệu so với mức ngưỡng. Nếu giá trị dữ liệu đo được vượt ngưỡng => Gọi service gửi tin nhắn email trên máy chủ.
- Service nhận tham số truyền vào là số điện thoại + nội dung tin nhắn (tin nhắn cảnh báo).
- Gửi tin nhắn đến số điện thoại cần gửi tin nhắn cảnh báo.

Sơ đồ activity chức năng phân tích, báo cáo và thống kê

Hình 3.12 mô tả sơ đồ chức năng phân tích, báo cáo và thống kê, bao gồm các bước sau:

- Đăng nhập vào trang quản trị. Hệ thống kiểm tra đăng nhập, nếu thông tin đăng nhập không trùng khớp thì trả về trang báo lỗi. Ngược lại nếu đăng nhập thành công => chuyển hướng đến trang quản trị.
- Tại trang quản trị, người dùng chọn chức năng xem lại lịch sử
- Tại đây, người dùng chọn cơ sở quản lý, thiết bị quản lý, ngày muốn xem lại thống kê, tham số cần xem và nhấn theo dõi.

- Đối với những ngày không có dữ liệu trả về, hệ thống sẽ đưa ra thông báo tới người dùng.
- Khi người dùng chọn đầy đủ các thông số và nhấn theo dõi, với những ngày có dữ liệu => biểu đồ thống kê sẽ được hiển thị ra.
- Ngoài ra, hệ thống còn cho phép xuất báo cáo dữ liệu tham số đo đạc dưới dạng file csv, excel...



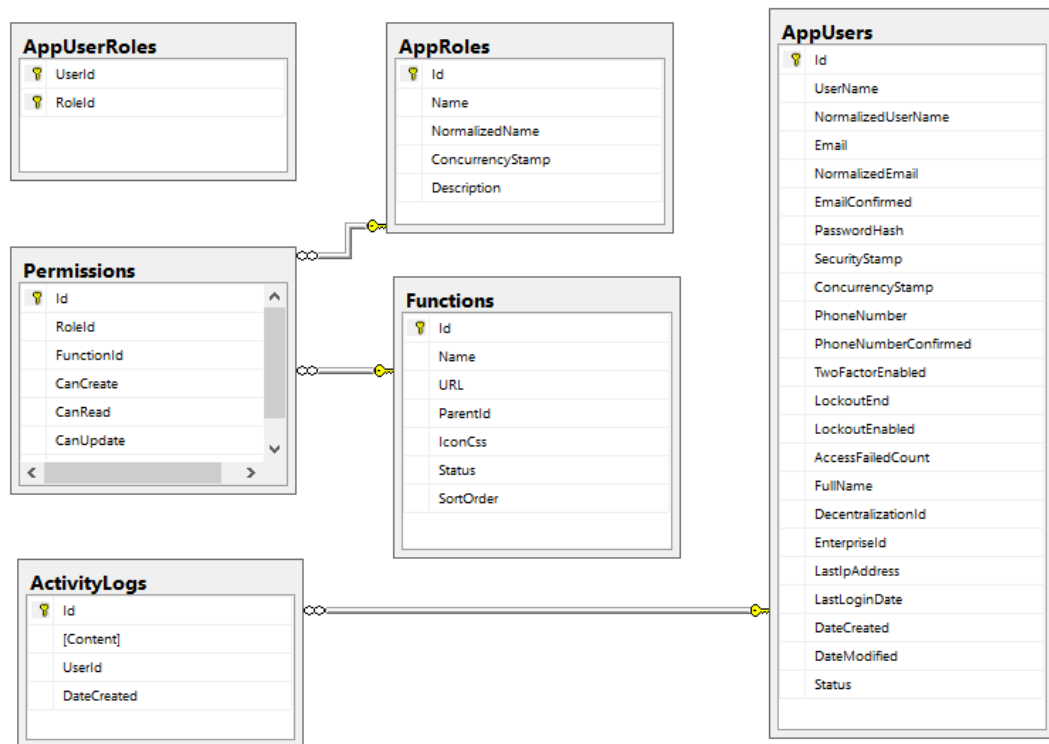
Hình 3.12: Sơ đồ activity chức năng phân tích, báo cáo và thống kê

3.2.4 Thiết kế cơ sở dữ liệu

Sau khi phân tích yêu cầu chức năng, phi chức năng, thiết kế sơ đồ use case, sơ đồ activity, em thiết kế cơ sở dữ liệu với 2 nhóm chức năng:

- Nhóm hệ thống : bao gồm các dbo liên quan đến người dùng và phân quyền người dùng.
- Nhóm chức năng: bao gồm các dbo liên quan đến quy trình nghiệp vụ.

3.2.4.1 Nhóm hệ thống



Hình 3.13: Mô hình thực thể liên kết nhóm hệ thống

- Bảng AppUsers : lưu trữ thông tin của tất cả người dùng trong hệ thống

Bảng 3-12: Bảng AppUsers

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	uniqueidentifier	Khoá chính	<input type="checkbox"/>
UserName	nvarchar(MAX)	Tên đăng nhập	<input type="checkbox"/>
Email	nvarchar(MAX)	Email	<input type="checkbox"/>
PasswordHash	nvarchar(MAX)	Mật khẩu đã mã hoá	<input type="checkbox"/>
PhoneNumber	nvarchar(MAX)	Số điện thoại	<input type="checkbox"/>
TwoFactorEnabled	bit	Bật xác thực 2 lớp	<input checked="" type="checkbox"/>
LockoutEnabled	bit	Bật tính năng khoá tài khoản	<input checked="" type="checkbox"/>
FullName	nvarchar(128)	Tên đầy đủ	<input checked="" type="checkbox"/>

DecentralizationId	nvarchar(128)	Phân cấp Id	<input checked="" type="checkbox"/>
EnterpriseId	int	Id Cơ sở	<input checked="" type="checkbox"/>
DateCreated	datetime2(7)	Ngày tạo	<input checked="" type="checkbox"/>
Status	bit	Trạng thái	<input checked="" type="checkbox"/>

- Bảng AppRoles: lưu trữ thông tin tất cả các quyền trong hệ thống.

Bảng 3-13: Bảng AppRoles

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	uniqueidentifier	Khoá chính	<input type="checkbox"/>
Name	nvarchar(MAX)	Tên quyền	<input type="checkbox"/>
Description	nvarchar(250)	Mô tả	<input checked="" type="checkbox"/>

- Bảng Functions: lưu trữ thông tin các functions, mỗi functions được tính là một hàm tương ứng với một đường dẫn url để người dùng thao tác.

Bảng 3-14: Bảng Functions

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	uniqueidentifier	Khoá chính	<input type="checkbox"/>
Name	nvarchar(MAX)	Tên quyền	<input type="checkbox"/>
URL	nvarchar(250)	URL	<input type="checkbox"/>
ParentId	nvarchar(128)	Id function cha	<input checked="" type="checkbox"/>
IconCss	nvarchar(MAX)	Css cho icon	<input checked="" type="checkbox"/>
Status	int	Trạng thái	<input type="checkbox"/>
SortOrder	int	Vị trí	<input type="checkbox"/>

- Bảng AppUserRoles: là bảng mapping giữa bảng AppUsers và Bảng AppRoles. Một User có nhiều Role và ngược lại => Quan hệ n – n.

Bảng 3-15: Bảng AppUserRoles

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
UserId	uniqueidentifier	Id User	<input type="checkbox"/>

RoleId	uniqueidentifier	Id quyền	<input type="checkbox"/>
--------	------------------	----------	--------------------------

- Bảng Permissions: lưu thông tin các permission, mỗi permission được tính là một chức năng gắn với một quyền và một functions. Permission xác định với một function và một quyền xác định thì user được phép làm gì (xem, tạo, sửa, xoá). Một Role có nhiều Permission => Quan hệ 1 – n. Một function có nhiều permission => Quan hệ 1 – n.

Bảng 3-16: Bảng Permissions

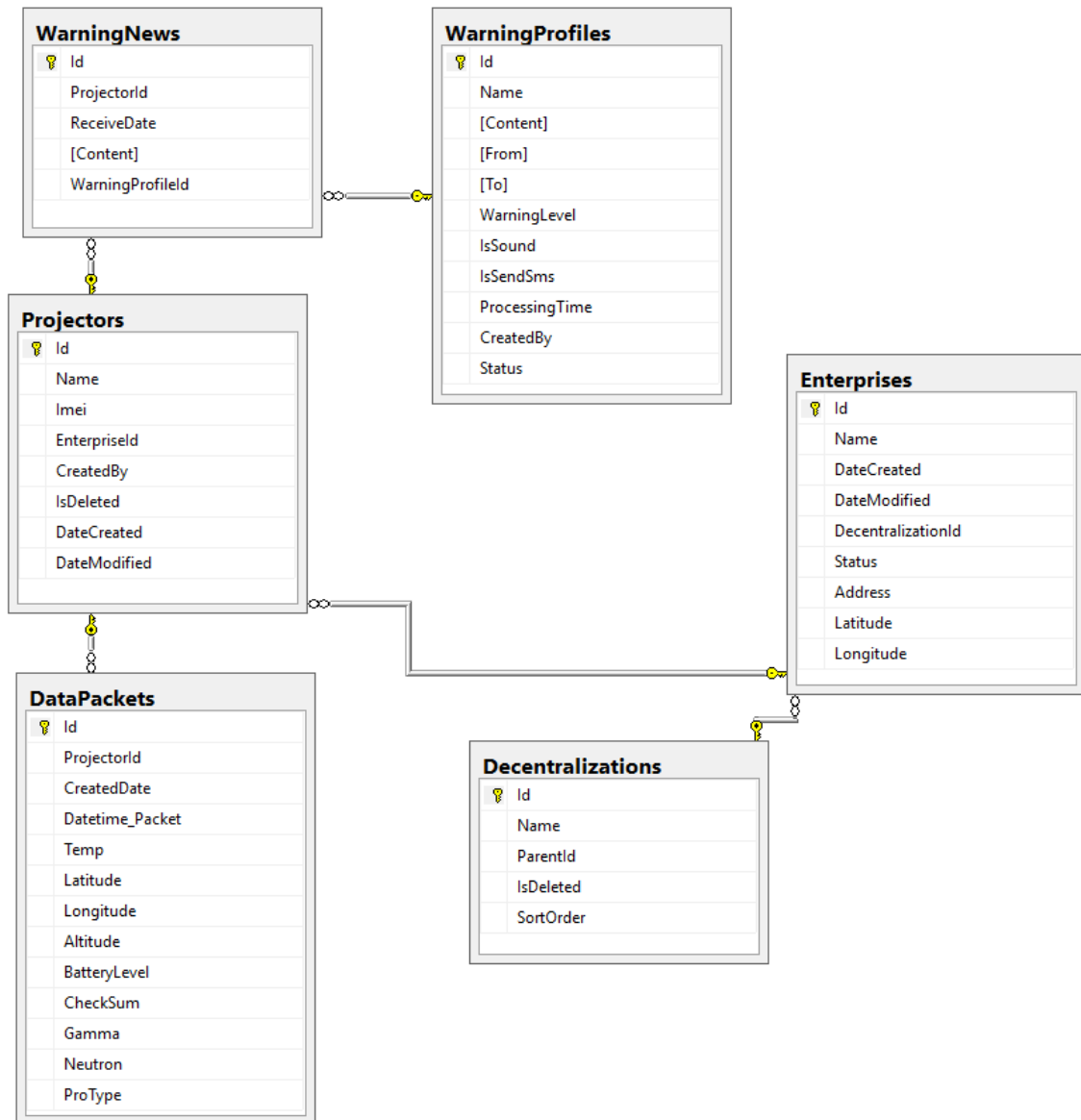
Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
RoleId	uniqueidentifier	Id quyền	<input type="checkbox"/>
FunctionId	varchar(128)	Id Function	<input type="checkbox"/>
CanCreate	bit	Có thể tạo mới?	<input checked="" type="checkbox"/>
CanRead	bit	Có thể xem	<input checked="" type="checkbox"/>
CanUpdate	bit	Có thể cập nhật	<input checked="" type="checkbox"/>
CanDelete	bit	Có thể xoá	<input checked="" type="checkbox"/>

- Bảng ActivityLogs: lưu thông tin lịch sử hoạt động của tất cả người dùng trong hệ thống. Một User có nhiều Activity => Quan hệ 1 – n.

Bảng 3-17: Bảng ActivityLogs

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
[Content]	nvarchar(MAX)	Nội dung	<input checked="" type="checkbox"/>
UserId	uniqueidentifier	Id User	<input type="checkbox"/>
DateCreated	datetime2(7)	Ngày tạo	<input checked="" type="checkbox"/>

3.2.4.2 Nhóm chức năng



Hình 3.14: Mô hình thực thể liên kết nhóm chức năng

- Bảng Enterprises: lưu trữ thông tin các cơ sở. Cơ sở là nơi đặt các thiết bị để đo đạc các tham số phóng xạ.

Bảng 3-18: Bảng Enterprises

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
Name	nvarchar(MAX)	Tên Cơ sở	<input type="checkbox"/>
DateCreated	datetime2(7)	Ngày tạo	<input checked="" type="checkbox"/>
DateModified	datetime2(7)	Ngày sửa	<input checked="" type="checkbox"/>

DecentralizationId	varchar(128)	Id phân cấp	<input checked="" type="checkbox"/>
Status	int	Trạng thái	<input checked="" type="checkbox"/>
Address	nvarchar(128)	Địa chỉ	<input checked="" type="checkbox"/>
Latitude	decimal(18, 9)	Vĩ độ	<input type="checkbox"/>
Longitude	decimal(18, 9)	Kinh độ	<input type="checkbox"/>

- Bảng Decentralizations: lưu trữ thông tin các phân cấp (Toàn quốc => Miền Bắc/ Trung/ Nam => Sở KHCN).

Bảng 3-19: Bảng Decentralizations

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	varchar(128)	Khoá chính	<input type="checkbox"/>
Name	nvarchar(500)	Tên Phân cấp	<input type="checkbox"/>
ParentId	nvarchar(128)	Id phân cấp cha	<input checked="" type="checkbox"/>
IsDeleted	bit	Đã xoá?	<input checked="" type="checkbox"/>
SortOrder	int	Vị trí	<input type="checkbox"/>

- Bảng Projectors: lưu thông tin các thiết bị đo phóng xạ. Mỗi thiết bị thuộc một cơ sở duy nhất. Một cơ sở có nhiều thiết bị => Quan hệ 1 – n.

Bảng 3-20: Bảng Projectors

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
Name	nvarchar(500)	Tên Thiết bị	<input type="checkbox"/>
Imei	varchar(50)	Imei định danh thiết bị	<input type="checkbox"/>
EnterpriseId	int	Id Cơ sở	<input type="checkbox"/>
CreatedBy	uniqueidentifier	Người tạo	<input type="checkbox"/>
IsDeleted	bit	Đã xoá?	<input checked="" type="checkbox"/>
DateCreated	datetime2(7)	Ngày tạo	<input checked="" type="checkbox"/>
DateModified	datetime2(7)	Ngày sửa	<input checked="" type="checkbox"/>

- Bảng DataPackets: lưu giá trị tham số đo đạc của các thiết bị. Mỗi giá trị của một lần đo tương ứng với một thiết bị. Một Projector có nhiều Datapacket => Quan hệ 1 – n.

Bảng 3-21: Bảng DataPackets

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
ProjectorId	int	Id Thiết bị	<input type="checkbox"/>
CreatedDate	datetime2(7)	Ngày tạo	<input checked="" type="checkbox"/>
Datetime_Packet	datetime2(7)	Thời gian nhận bản tin	<input type="checkbox"/>
Temp	real	Nhiệt độ	<input type="checkbox"/>
Latitude	decimal(18, 9)	Vĩ độ	<input type="checkbox"/>
Longitude	decimal(18, 9)	Kinh độ	<input type="checkbox"/>
Altitude	smallint	Độ cao so với mực nước biển	<input type="checkbox"/>
BatteryLevel	smallint	Mức pin	<input type="checkbox"/>
Checksum	smallint	Check Sum	<input type="checkbox"/>
Gamma	real	Nồng độ Gamma	<input type="checkbox"/>
Neutron	real	Nồng độ Neutron	<input type="checkbox"/>
ProType	tinyint	Loại bản tin	<input type="checkbox"/>

- Bảng WarningProfiles: lưu thông tin các cấu hình cảnh báo nếu tham số vượt ngưỡng.

Bảng 3-22: Bảng WarningProfiles

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
Name	nvarchar(128)	Tên Cấu hình	<input type="checkbox"/>
[Content]	nvarchar(500)	Nội dung cảnh báo	<input checked="" type="checkbox"/>
[From]	float	Ngưỡng Từ	<input type="checkbox"/>

[To]	float	Ngưỡng Đến	<input type="checkbox"/>
WarningLevel	int	Mức cảnh báo	<input type="checkbox"/>
IsSound	bit	Có cảnh báo âm thanh?	<input type="checkbox"/>
IsSendSms	bit	Có gửi tin nhắn cảnh báo?	<input type="checkbox"/>
ProcessingTime	int	Thời gian xử lý	<input type="checkbox"/>
CreatedBy	uniqueidentifier	Người tạo	<input type="checkbox"/>
Status	int	Trạng thái	<input type="checkbox"/>

- Bảng WarningNews: lưu các bản tin cảnh báo. Mỗi bản tin cảnh báo ứng với một thiết bị. Một thiết bị có nhiều bản tin cảnh báo => Quan hệ 1 – n. Mỗi bản tin cảnh báo ứng với một cấu hình cảnh báo. Một cấu hình cảnh báo có nhiều bản tin cảnh báo => Quan hệ 1 – n.

Bảng 3-23: Bảng WarningNews

Tên hiển thị	Kiểu dữ liệu	Mô tả	Cho phép NULL
Id	int	Khoá chính	<input type="checkbox"/>
ProjectorId	int	Id Thiết bị	<input type="checkbox"/>
ReceiveDate	datetime2(7)	Thời gian nhận	<input type="checkbox"/>
[Content]	nvarchar(128)	Nội dung cảnh báo	<input checked="" type="checkbox"/>
WarningProfileId	int	Id Cấu hình cảnh báo	<input type="checkbox"/>

3.2.5 Thiết kế bộ API phát triển ứng dụng trên Smartphone

3.2.5.1 Tổng quan về API

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành. api Mỗi bộ API dành cho các hệ điều hành khác nhau là hoàn toàn khác nhau và không có sự tương thích với nhau. API dành cho các hệ điều hành Windows và Linux là hoàn toàn khác nhau. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm

hay dùng. Nói nôm na API là một loại công cụ để tạo ra phần mềm, cũng giống như các viên gạch xây nên một toà nhà. Chúng ta chia việc xây nhà ra các phần khác nhau, API cũng như viên gạch là một bộ phận của ngôi nhà, nhưng việc tạo ra nó là một công đoạn riêng hoàn toàn, chúng ta có thể tự làm hoặc đơn giản là đi mua của các nhà cung cấp. Web API là một trong những công nghệ mới của Microsoft dùng để xây dựng dịch vụ thành phần phân tán. Web API là mô hình dùng để hỗ trợ MVC bao gồm: routing, controller, action result, filter, filter, loc container, model binder, unit test, injection. Bên cạnh đó nó còn hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu [11].

3.2.5.2 *Json Web Token*

JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

Cấu trúc của Json Web Token gồm 3 phần: Header, Payload, Signature:

- Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

- Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ.

```
{  
  "iss": "jira:1314039",  
  "iat": 1300819370,  
  "exp": 1300819380,  
  "context": {  
    "user": {  
      "username": "bwayne",  
      "displayName": "Bruce Wayne"  
    }  
  }  
}
```

```

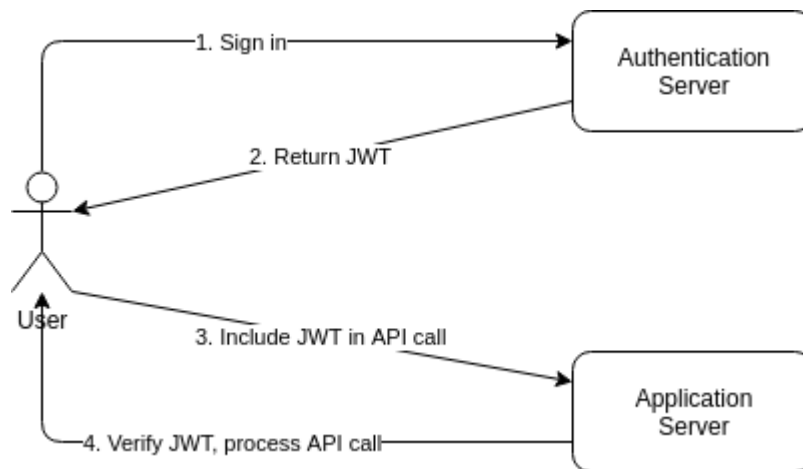
    }
  }
}

```

- Signature là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

$\text{HMACSHA256}(\text{base64UrlEncode}(\text{header}) + "." + \text{base64UrlEncode}(\text{payload}), \text{secret})$

3.2.5.3 Thiết kế API



Hình 3.15: Sơ đồ tổng quan API

Hình 3.15 mô tả sơ đồ tổng quan API, bao gồm các bước sau:

- Bước 1: User gửi request đến Server để xác thực bằng cách gọi API login. Sau khi login thành công => Server trả về một chuỗi json dùng để xác thực gọi là JWT (Json Web Token).
- Bước 2: User gửi request để lấy data từ server bằng cách gọi các API đồng thời đính kèm JWT vào header.
- Bước 3: Server nhận request và phân tích chuỗi JWT để xác thực. Nếu xác thực thành công => Server trả về data. Ngược lại nếu thất bại => Server trả về response thông báo chưa được xác thực và mã lỗi 401 (Unauthorized).

3.3 Kết luận

Trong chương 3, em đã trình bày chi tiết các yêu cầu chức năng và phi chức năng của hệ thống, đồng thời đưa ra các quy trình thiết kế hệ thống phần mềm. Từ cơ sở này, em sẽ tiến hành triển khai thực nghiệm hệ thống và trình bày trong chương 4.

CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM

Sau khi đã phân tích các yêu cầu chức năng và phi chức năng đồng thời đưa ra các quy trình thiết kế hệ thống, trong chương này em sẽ trình bày các kết quả thực nghiệm của hệ thống phần mềm.

4.1 Nhận dữ liệu từ thiết bị truyền thông

Tiến hành gửi một số bản tin mẫu từ thiết bị phản cứng qua giao thức truyền thông LoraWan. Sau khi gửi, phía server nhận được bản tin như hình dưới đây:

```

1  [{"rssi": -63, "lat": 21.00344, "lon": 105.84394, "altitude": 781, "frequency": 868100000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
2  {"rssi": -60, "lat": 21.00355, "lon": 105.84398, "altitude": 701, "frequency": 868300000, "dr": 5, "ad": true, "true": 1, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
3  {"rssi": -69, "lat": 21.00355, "lon": 105.84398, "altitude": 701, "frequency": 868100000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
4  {"rssi": -62, "lat": 21.00361, "lon": 105.84398, "altitude": 681, "frequency": 868100000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
5  {"rssi": -66, "lat": 21.00363, "lon": 105.84403, "altitude": 641, "frequency": 867100000, "dr": 5, "ad": true, "true": 1, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
6  {"rssi": -61, "lat": 21.00335, "lon": 105.84397, "altitude": 681, "frequency": 868300000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
7  {"rssi": -64, "lat": 21.00339, "lon": 105.84399, "altitude": 681, "frequency": 868500000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
8  {"rssi": -69, "lat": 21.00346, "lon": 105.844, "altitude": 791, "frequency": 867300000, "dr": 5, "ad": true, "true": 4, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
9  {"rssi": -65, "lat": 21.00349, "lon": 105.84402, "altitude": 741, "frequency": 867700000, "dr": 5, "ad": true, "true": 5, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
10 {"rssi": -64, "lat": 21.00356, "lon": 105.84402, "altitude": 691, "frequency": 867900000, "dr": 5, "ad": true, "true": 6, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
11 {"rssi": -65, "lat": 21.00355, "lon": 105.84403, "altitude": 691, "frequency": 867500000, "dr": 5, "ad": true, "true": 7, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
12 {"rssi": -64, "lat": 21.00402, "lon": 105.84412, "altitude": 641, "frequency": 867700000, "dr": 5, "ad": true, "true": 20, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
13 {"rssi": -79, "lat": 21.00384, "lon": 105.84415, "altitude": 301, "frequency": 867900000, "dr": 5, "ad": true, "true": 21, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
14 {"rssi": -75, "lat": 21.00363, "lon": 105.84412, "altitude": 421, "frequency": 867300000, "dr": 5, "ad": true, "true": 22, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
15 {"rssi": -66, "lat": 21.00366, "lon": 105.84412, "altitude": 441, "frequency": 868100000, "dr": 5, "ad": true, "true": 23, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
16 {"rssi": -63, "lat": 21.00361, "lon": 105.84414, "altitude": 431, "frequency": 867500000, "dr": 5, "ad": true, "true": 24, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
17 {"rssi": -65, "lat": 21.00355, "lon": 105.84416, "altitude": 461, "frequency": 867500000, "dr": 5, "ad": true, "true": 25, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
18 {"rssi": -60, "lat": 21.00352, "lon": 105.84417, "altitude": 481, "frequency": 869500000, "dr": 5, "ad": true, "true": 26, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
19 {"rssi": -60, "lat": 21.00336, "lon": 105.84417, "altitude": 571, "frequency": 867700000, "dr": 5, "ad": true, "true": 27, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
20 {"rssi": -64, "lat": 21.00322, "lon": 105.84415, "altitude": 621, "frequency": 867100000, "dr": 5, "ad": true, "true": 28, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
21 {"rssi": -63, "lat": 21.00322, "lon": 105.8441, "altitude": 691, "frequency": 867900000, "dr": 5, "ad": true, "true": 29, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
22 {"rssi": -61, "lat": 21.0033, "lon": 105.84408, "altitude": 661, "frequency": 868500000, "dr": 5, "ad": true, "true": 30, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
23 {"rssi": -60, "lat": 21.00331, "lon": 105.84406, "altitude": 681, "frequency": 867300000, "dr": 5, "ad": true, "true": 31, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
24 {"rssi": -61, "lat": 21.00327, "lon": 105.84402, "altitude": 711, "frequency": 868500000, "dr": 5, "ad": true, "true": 33, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
25 {"rssi": -61, "lat": 21.00331, "lon": 105.844, "altitude": 611, "frequency": 867500000, "dr": 5, "ad": true, "true": 34, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
26 {"rssi": -58, "lat": 21.00353, "lon": 105.84399, "altitude": 641, "frequency": 867700000, "dr": 5, "ad": true, "true": 35, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
27 {"rssi": -63, "lat": 21.00357, "lon": 105.84396, "altitude": 621, "frequency": 867900000, "dr": 5, "ad": true, "true": 36, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
28 {"rssi": -61, "lat": 21.00346, "lon": 105.84388, "altitude": 521, "frequency": 867100000, "dr": 5, "ad": true, "true": 37, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
29 {"rssi": -61, "lat": 21.00348, "lon": 105.84398, "altitude": 521, "frequency": 868100000, "dr": 5, "ad": true, "true": 38, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
30 {"rssi": -61, "lat": 21.00346, "lon": 105.84398, "altitude": 521, "frequency": 868300000, "dr": 5, "ad": true, "true": 39, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
31 {"rssi": -61, "lat": 21.00363, "lon": 105.84394, "altitude": 421, "frequency": 867300000, "dr": 5, "ad": true, "true": 40, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
32 {"rssi": -58, "lat": 21.00356, "lon": 105.84391, "altitude": 441, "frequency": 868500000, "dr": 5, "ad": true, "true": 41, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
33 {"rssi": -59, "lat": 21.00357, "lon": 105.84398, "altitude": 371, "frequency": 867700000, "dr": 5, "ad": true, "true": 42, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
34 {"rssi": -58, "lat": 21.00343, "lon": 105.84388, "altitude": 351, "frequency": 867500000, "dr": 5, "ad": true, "true": 43, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
35 {"rssi": -60, "lat": 21.00337, "lon": 105.8439, "altitude": 561, "frequency": 867900000, "dr": 5, "ad": true, "true": 44, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
36 {"rssi": -60, "lat": 21.00346, "lon": 105.84394, "altitude": 441, "frequency": 868100000, "dr": 5, "ad": true, "true": 45, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
37 {"rssi": -55, "lat": 21.00357, "lon": 105.84397, "altitude": 291, "frequency": 867100000, "dr": 5, "ad": true, "true": 46, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
38 {"rssi": -55, "lat": 21.00344, "lon": 105.84399, "altitude": 251, "frequency": 868300000, "dr": 5, "ad": true, "true": 47, "float": 1, "data": "S0Vab08iHdvcmkiQ=="},
39 {"rssi": -61, "lat": 21.00341, "lon": 105.84409, "altitude": 481, "frequency": 867700000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "AMCQ=="},
40 {"rssi": -63, "lat": 21.0035, "lon": 105.84409, "altitude": 481, "frequency": 868100000, "dr": 5, "ad": true, "true": 0, "float": 1, "data": "AMCQ=="},
41 {"rssi": -59, "lat": 21.0035, "lon": 105.84409, "altitude": 431, "frequency": 868500000, "dr": 5, "ad": true, "true": 3, "float": 1, "data": "AQCC=="},
42 {"rssi": -61, "lat": 21.00362, "lon": 105.84409, "altitude": 321, "frequency": 867300000, "dr": 5, "ad": true, "true": 4, "float": 1, "data": "AQCC=="},
43 {"rssi": -60, "lat": 21.00362, "lon": 105.84402, "altitude": 301, "frequency": 867900000, "dr": 5, "ad": true, "true": 16, "float": 1, "data": "AQCC=="},
44 {"rssi": -61, "lat": 21.00361, "lon": 105.84402, "altitude": 341, "frequency": 868100000, "dr": 5, "ad": true, "true": 18, "float": 1, "data": "AQCC=="},
45 {"rssi": -60, "lat": 21.0035, "lon": 105.844, "altitude": 281, "frequency": 868300000, "dr": 5, "ad": true, "true": 19, "float": 1, "data": "AQCC=="},
46 {"rssi": -61, "lat": 21.00331, "lon": 105.844, "altitude": 241, "frequency": 867100000, "dr": 5, "ad": true, "true": 10, "float": 1, "data": "AQCC=="},
47 {"rssi": -57, "lat": 21.0033, "lon": 105.84403, "altitude": 311, "frequency": 868500000, "dr": 5, "ad": true, "true": 11, "float": 1, "data": "AQCC=="},
48 {"rssi": -63, "lat": 21.00345, "lon": 105.84407, "altitude": 311, "frequency": 867700000, "dr": 5, "ad": true, "true": 12, "float": 1, "data": "AQCC=="},
49 {"rssi": -61, "lat": 21.00338, "lon": 105.84408, "altitude": 311, "frequency": 867500000, "dr": 5, "ad": true, "true": 13, "float": 1, "data": "AQCC=="},

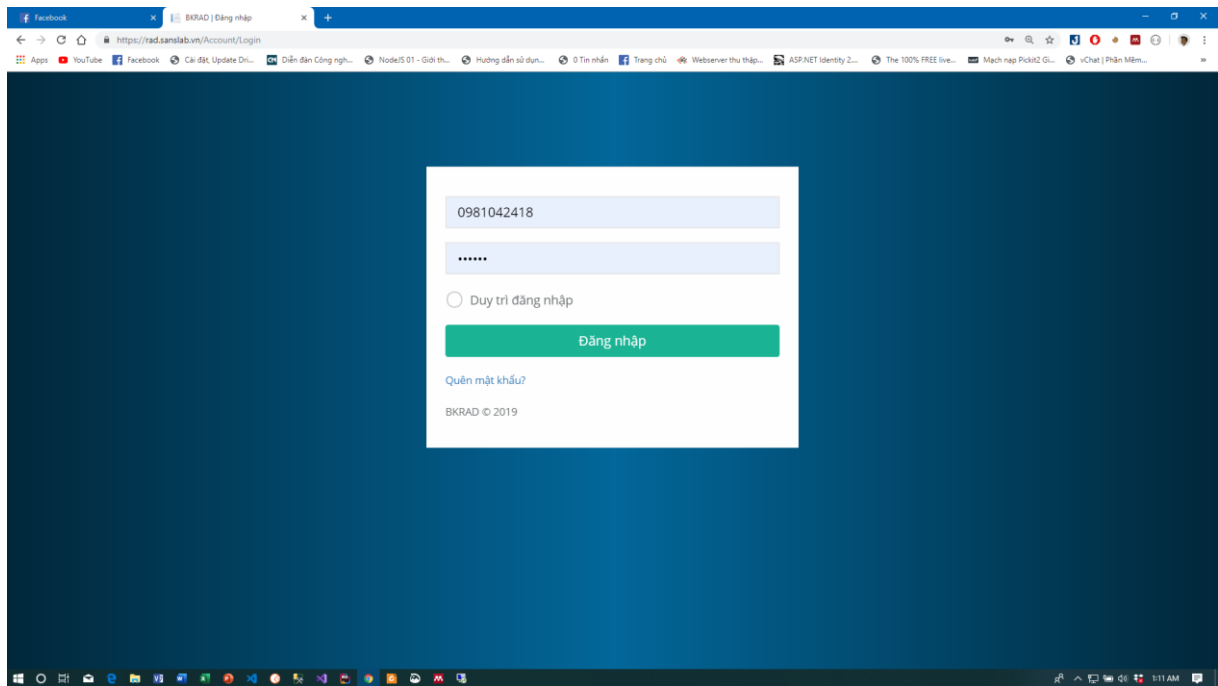
```

Hình 4.1: Kết quả nhận dữ liệu từ thiết bị truyền thông

Hình 4.1 thể hiện dữ liệu từ thiết bị truyền thông. Kết quả cho thấy dữ liệu nhận được trùng khớp với dữ liệu đã gửi đi.

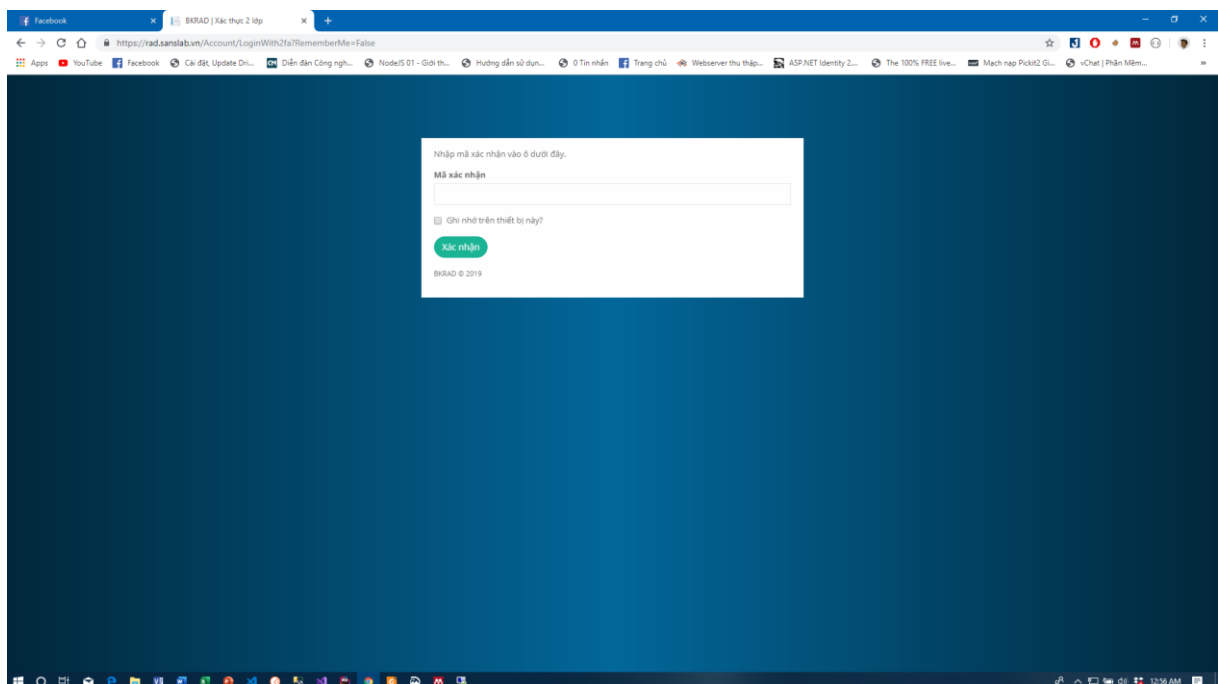
4.2 Giám sát trực tuyến

Để có thể thực hiện chức năng giám sát trực tuyến, hệ thống yêu cầu đăng nhập để xác thực người dùng.



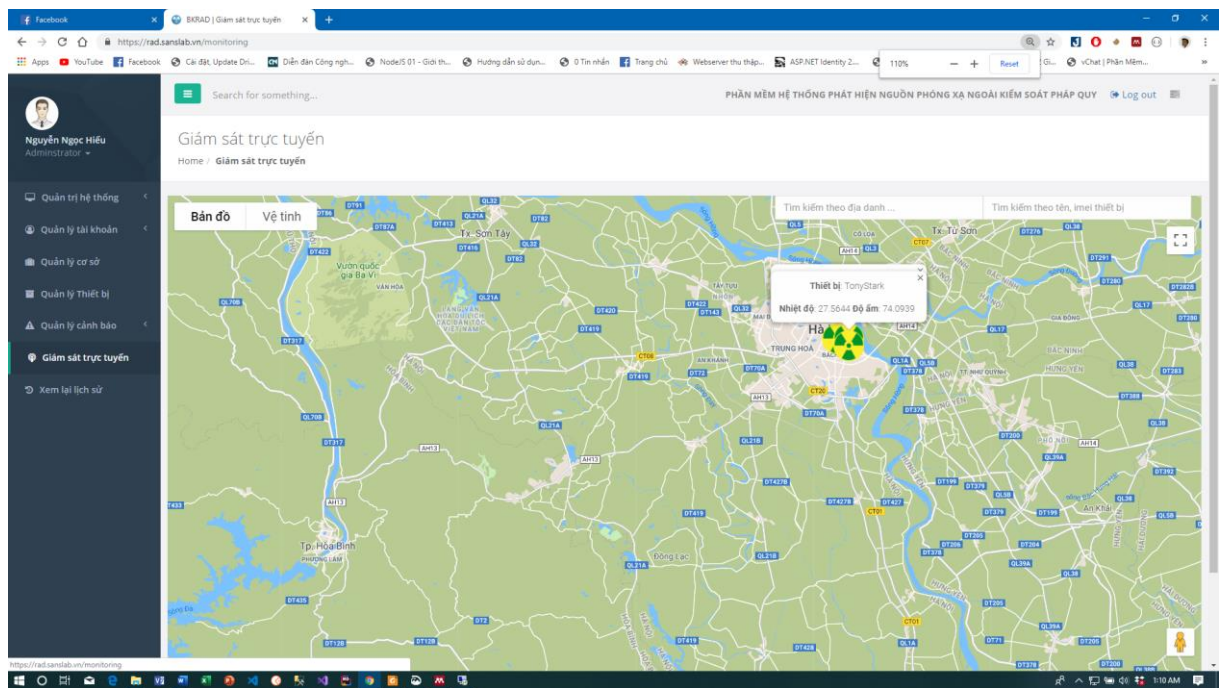
Hình 4.2: Giao diện đăng nhập

Hệ thống yêu cầu nhập mã OTP (mã xác thực 2 lớp) do người dùng đã bật chức năng xác thực 2 lớp:



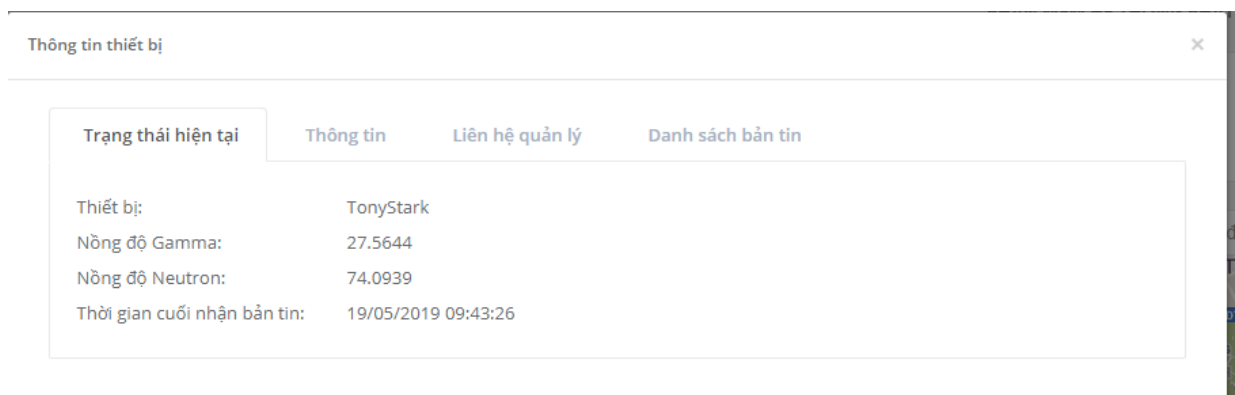
Hình 4.3: Giao diện xác thực 2 lớp

Sau khi xác thực thành công, hệ thống chuyển qua màn hình giám sát trực tuyến. Đây cũng chính là giao diện chính của trang web.



Hình 4.4: Giao diện giám sát trực tuyến

Hình 4.4 mô tả màn hình giám sát trực tuyến. Tại đây hiển thị vị trí đặt các thiết bị đo phóng xạ đồng thời hiển thị giá trị các tham số đo đạc trên nền google map. Ngoài ra người dùng có thể tìm kiếm nhanh thiết bị bằng cách gõ từ khóa vào ô tìm kiếm. Đồng thời vị trí của các thiết bị này cũng sẽ được cập nhật realtime.

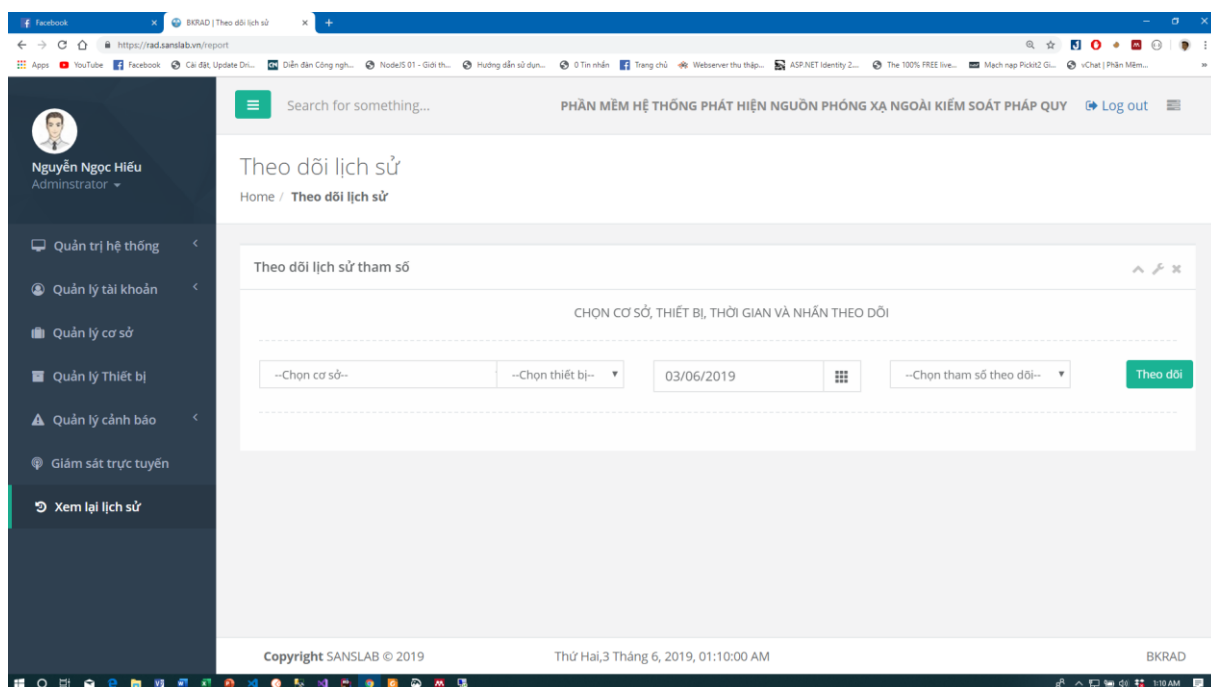


Hình 4.5: Giao diện chi tiết marker thiết bị

Khi ấn vào một marker trên bản đồ, phần mềm sẽ hiển thị thông tin chi tiết của thiết bị tương ứng với marker đó, bao gồm trạng thái hiện tại, thông tin thiết bị, liên hệ quản lý và danh sách bản tin.

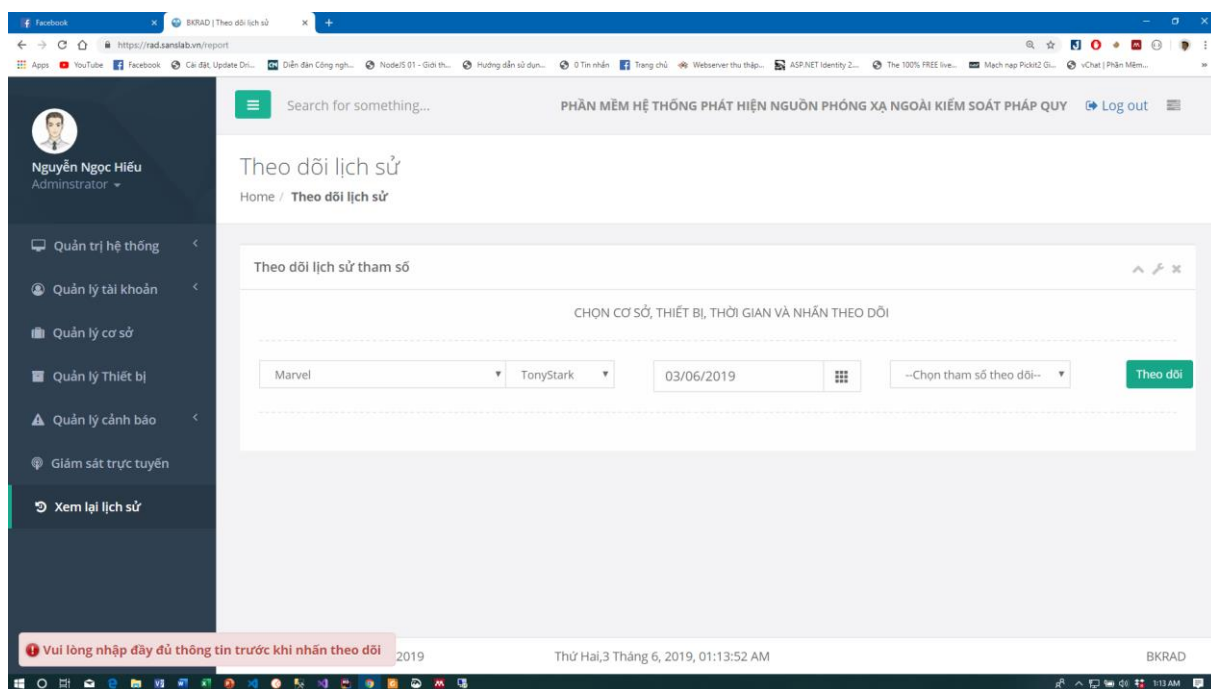
4.3 Thống kê

Người dùng chọn chức năng xem lại lịch sử để xem biểu đồ thống kê lịch sử tham số.



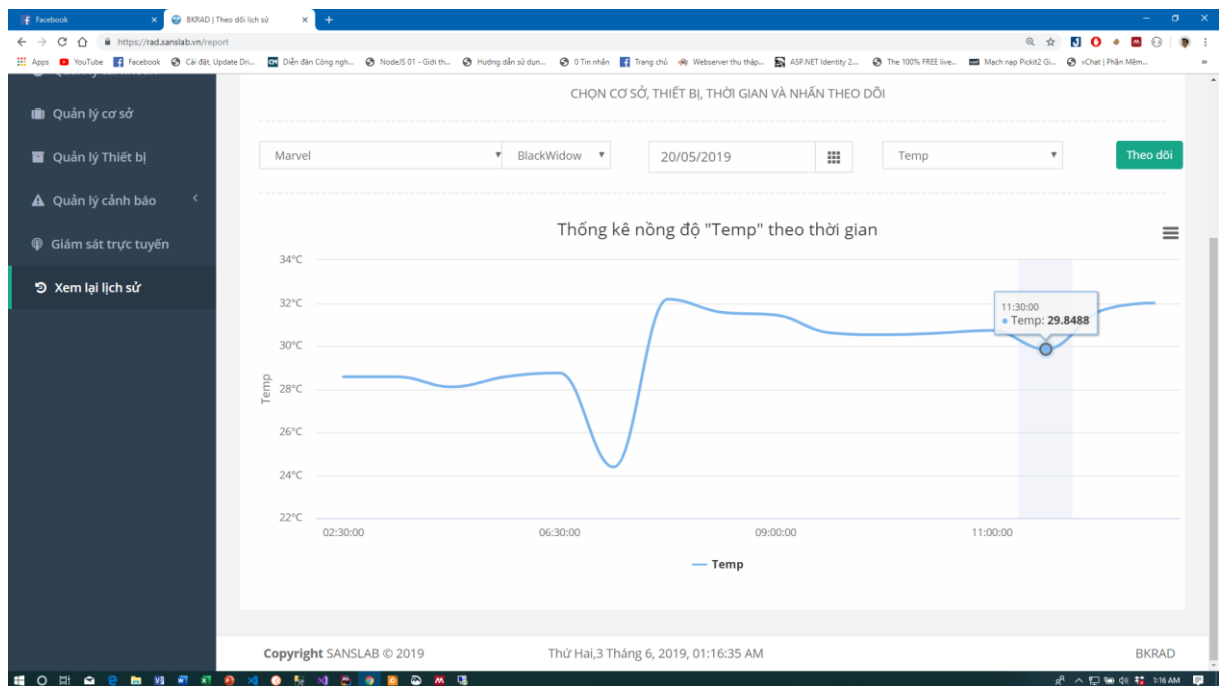
Hình 4.6: Giao diện chức năng thống kê

Khi người dùng chưa nhập đầy đủ thông tin mà nhấn nút theo dõi, hệ thống sẽ đưa ra thông báo lỗi và yêu cầu người dùng nhập lại.



Hình 4.7: Màn hình thông báo lỗi

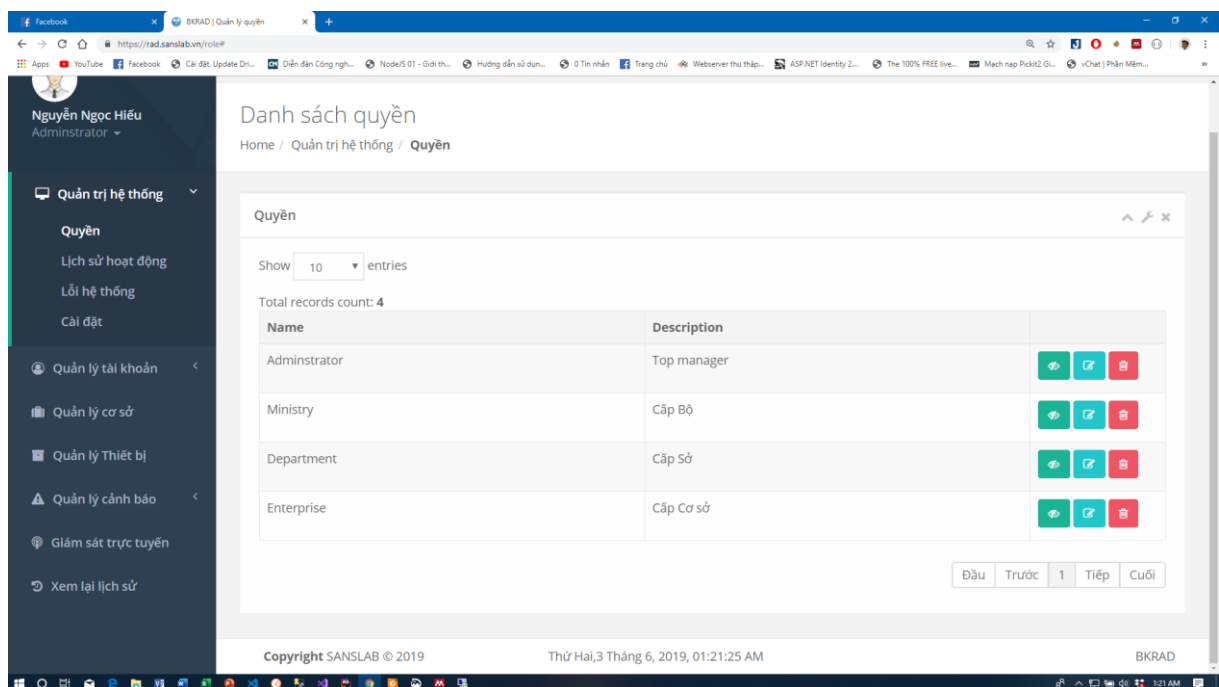
Trường hợp người dùng đã nhập đầy đủ thông tin, hệ thống sẽ trả về biểu đồ thống kê lịch sử tham số như Hình 4.8.



Hình 4.8: Biểu đồ thống kê lịch sử tham số

4.4 Quản trị

4.4.1 Quản lý quyền



Hình 4.9: Giao diện quản lý quyền

Tại màn hình quản lý quyền, người dùng có thể chọn chức năng gán permission cho quyền bằng cách nhấn vào button.

Assign permission ×

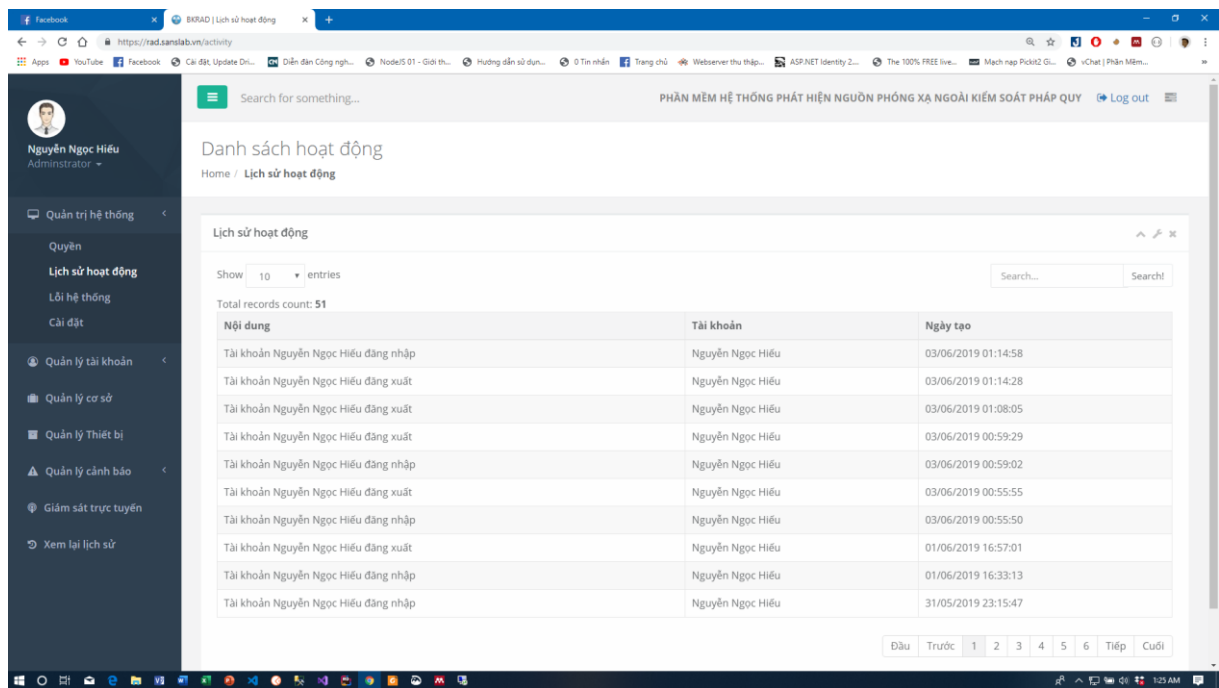
Function name	<input checked="" type="checkbox"/> Xem	<input checked="" type="checkbox"/> Thêm mới	<input checked="" type="checkbox"/> Cập nhật	<input checked="" type="checkbox"/> Xoá
▼ Quản lý tài khoản	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Tài khoản cấp Sở	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Tài khoản cấp Cục	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Quản lý cơ sở	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Giám sát trực tuyến	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Quản lý Thiết bị	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Xem lại lịch sử	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
▼ Quản trị hệ thống	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Quyền	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Cài đặt	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Lỗi hệ thống	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Lịch sử hoạt động	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
▼ Quản lý cảnh báo	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Lịch sử cảnh báo	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow
Cấu hình cảnh báo	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow	<input checked="" type="checkbox"/> Allow

Save
Close

Hình 4.10: Giao diện gán chức năng cho quyền

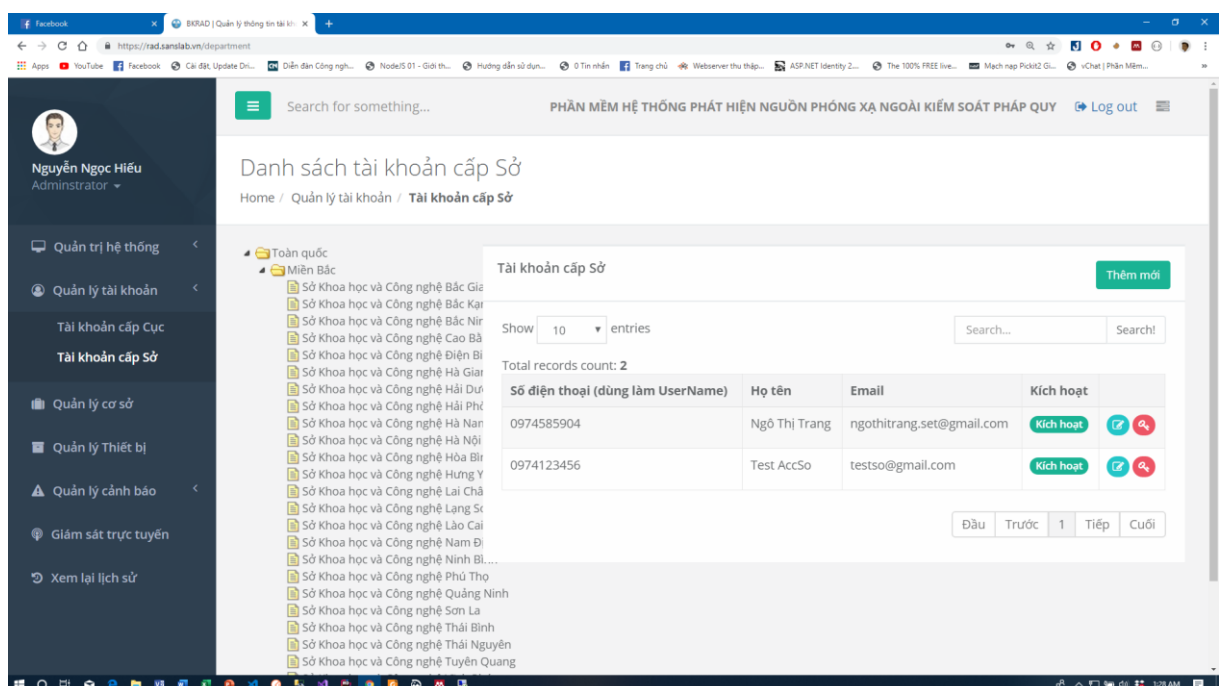
4.4.2 Lịch sử hoạt động

Tại màn hình theo dõi lịch sử hoạt động, hệ thống sẽ hiển thị danh sách các hoạt động của tất cả các người dùng trong hệ thống, bao gồm các thông tin về nội dung hoạt động, tài khoản thao tác và ngày tạo.



Hình 4.11: Giao diện lịch sử hoạt động

4.4.3 Quản lý tài khoản



Hình 4.12: Giao diện quản lý tài khoản

Hình 4.12 mô tả giao diện quản lý tài khoản. Hệ thống hiển thị danh sách tài khoản theo phân cấp, người dùng có thể chọn nhanh tài khoản thuộc phân cấp nào bằng cách nhấn vào phân cấp ở cột bên trái.

Khi nhấn vào button thêm mới, hệ thống sẽ hiển thị màn hình thêm mới tài khoản.

Thêm mới/ Chỉnh sửa tài khoản

Thuộc phân cấp

Chọn phân cấp

Số điện thoại (dùng làm UserName)

0981042418

Email

Mật khẩu

.....

Xác nhận mật khẩu

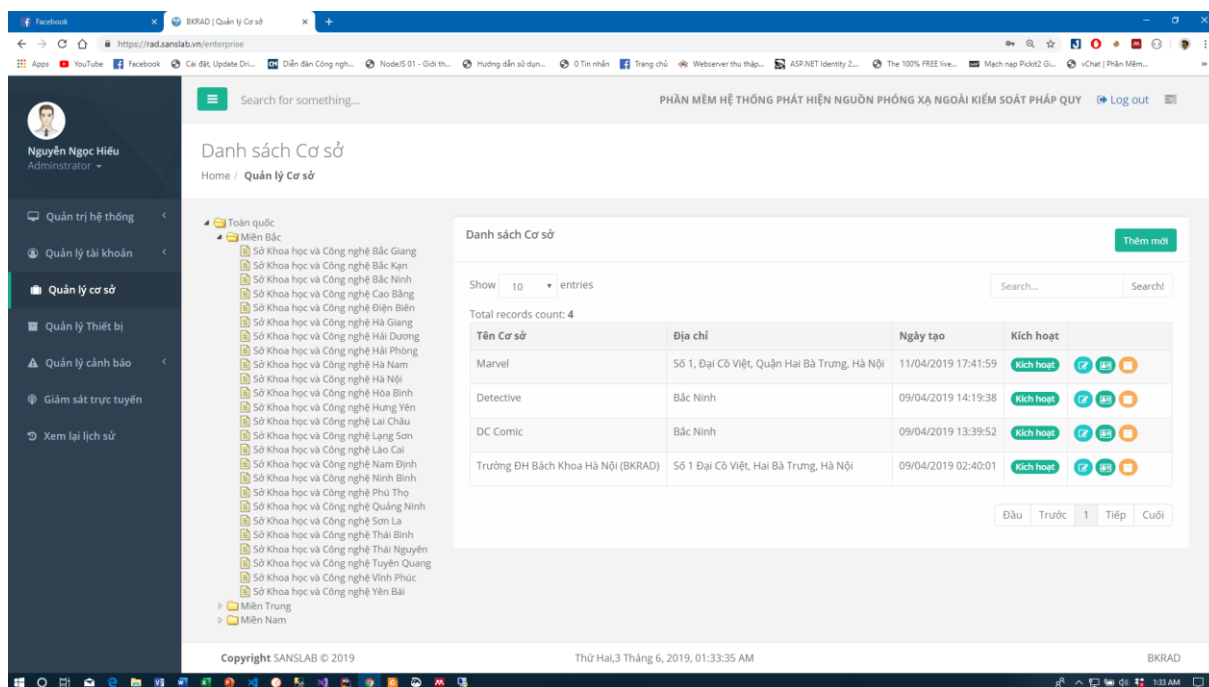
Họ tên

☒ Kích hoạt

SaveClose

Hình 4.13: Giao diện thêm mới tài khoản

4.4.4 Quản lý cơ sở

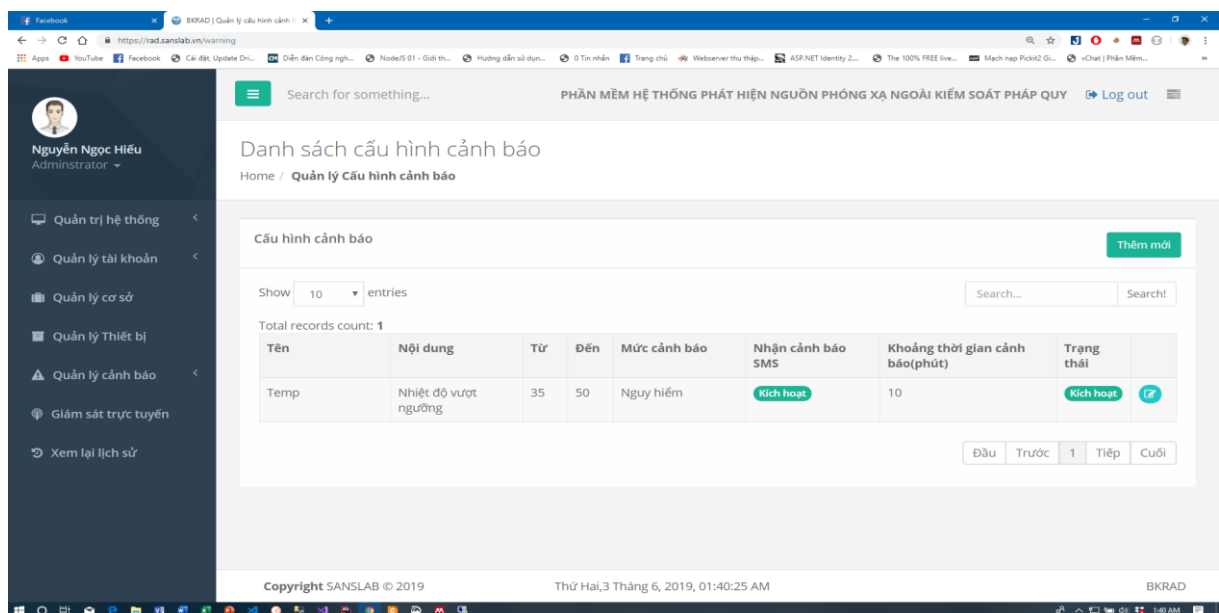


Hình 4.14: Giao diện quản lý cơ sở

Hình 4.14 mô tả giao diện quản lý cơ sở được thể hiện theo từng phân cấp. Tại đây người dùng có thể thêm mới một cơ sở:

hoặc thêm mới một thiết bị ứng với cơ sở đó:

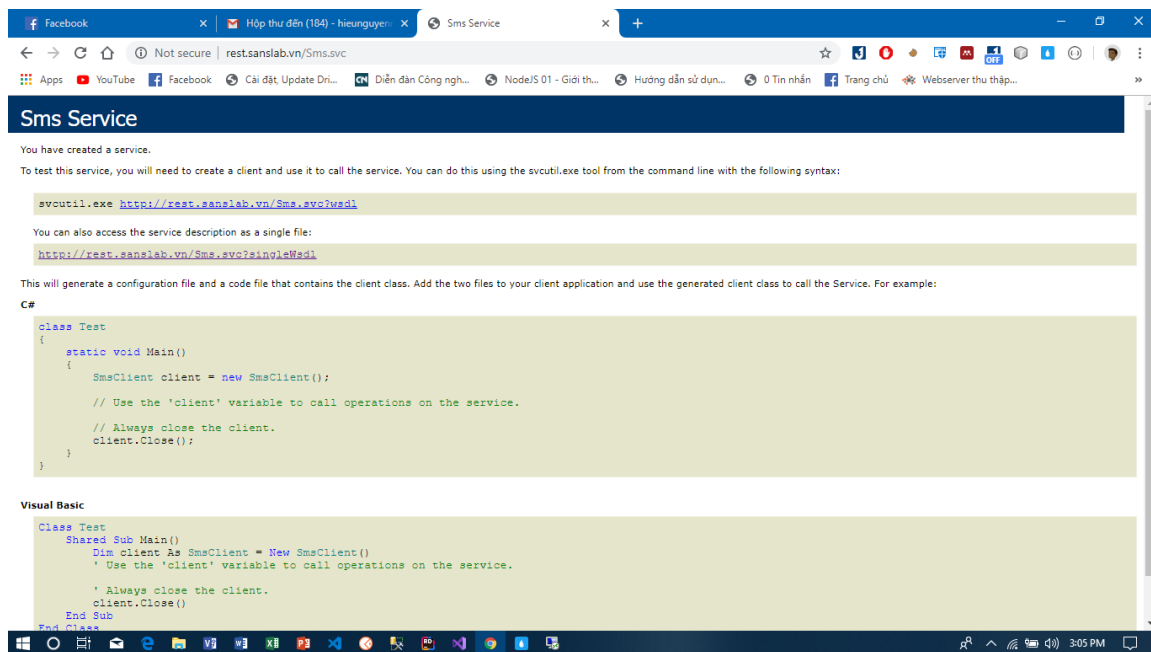
4.5 Cấu hình



NGUYỄN NGỌC HIỀU

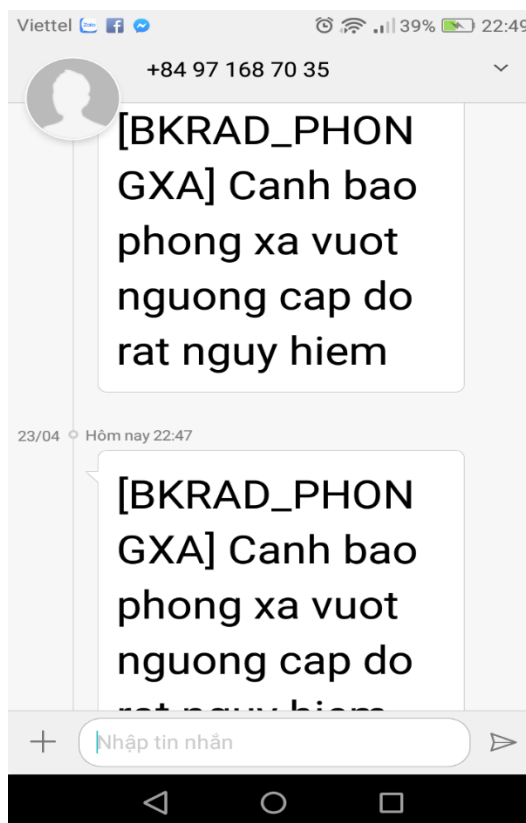
Hình 4.18: Giao diện thêm mới cấu hình cảnh báo

Hình 4.19 mô tả giao diện service gửi tin nhắn SMS. Tại đây người dùng có thể nhìn thấy cách dùng và gọi đến service.



Hình 4.19: Service gửi tin nhắn

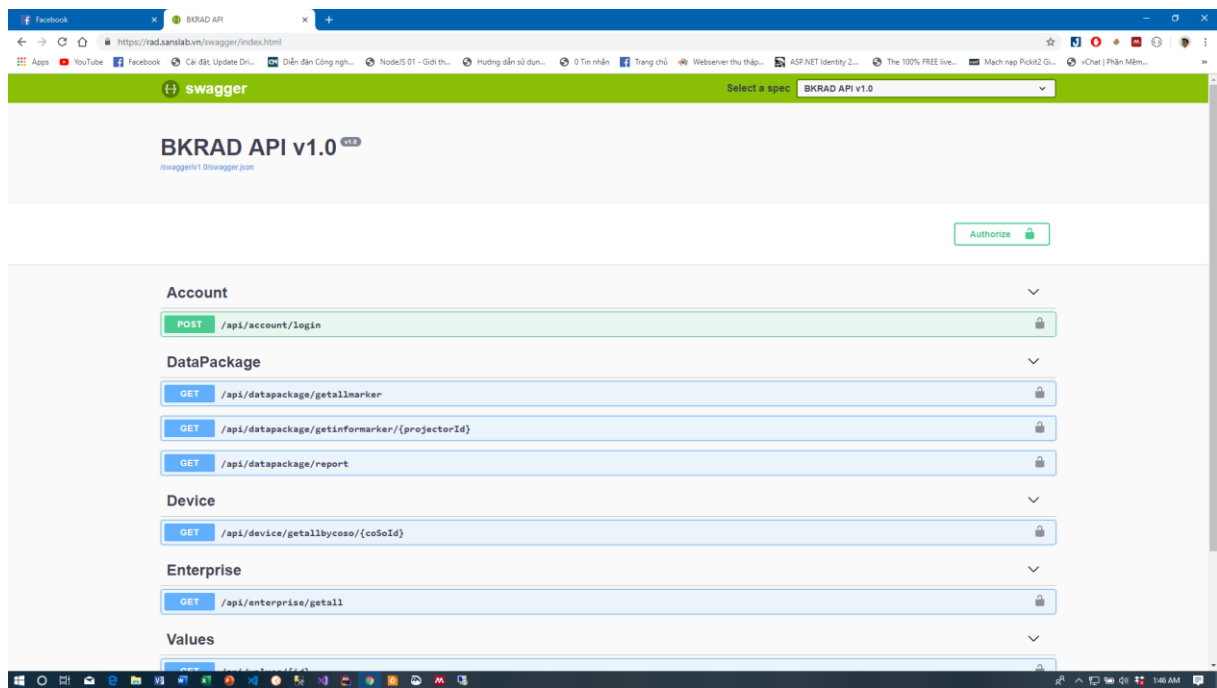
Sau khi gọi thành công service gửi SMS, kết quả thể hiện như Hình 4.20.



Hình 4.20: Tin nhắn cảnh báo được gửi từ server đến số điện thoại cảnh báo

4.7 API

Hình 4.21 mô tả giao diện kiểm thử API. Tại đây người dùng có thể kiểm tra các API có trong hệ thống và xem các dữ liệu trả về từ các API này.



Hình 4.21: Giao diện kiểm thử API

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau quá trình tìm hiểu, nghiên cứu và thực hiện đề tài “***Thiết kế phát triển phần mềm hệ thống phát hiện nguồn phóng xạ ngoài kiểm soát***”, em đã đạt được một số kết quả rất khả quan. Phần mềm đã được đưa vào sử dụng và có một số tiến triển tốt. Dữ liệu nhận được từ thiết bị truyền thông một cách ổn định và cập nhật theo thời gian thực. Hệ thống phân cấp người dùng đã giúp cho quản trị viên dễ dàng quản lý. Chức năng xác thực bằng mã OTP đã giúp hệ thống trở nên bảo mật hơn. Ngoài ra, việc tích hợp bộ API làm tăng khả năng mở rộng hệ thống và tiếp cận được nhiều người dùng, không chỉ trên nền tảng website mà còn trên nền tảng di động.

Mặc dù vậy, đề tài vẫn còn có một số vấn đề cần giải quyết. Việc chưa tối ưu được phần tài nguyên css, javascript đã khiến cho website chạy khá chậm so với thông thường (website mất tầm 3-5s để load hết trang). Điều này là không cho phép đối với một website chạy thương mại. Giao diện website còn đơn điệu, chưa bắt mắt, thu hút người dùng. Ngoài ra, chức năng xác thực 2 lớp bằng ứng dụng Google Authenticator còn chưa chạy ổn định (chỉ chạy trên local mà không chạy trên host).

Trong tương lai, em sẽ tiếp tục hoàn thành và khắc phục các lỗi còn gặp phải của đề tài. Đồng thời, em sẽ áp dụng những công nghệ mới nhất như sử dụng ngôn ngữ NoSQL thay cho SQL để tăng tốc độ truy vấn, sử dụng docker để chạy đa nền tảng, sử dụng máy chủ Linux thay cho Window để giảm tải cho server hiện tại.

TÀI LIỆU THAM KHẢO

- [1] “Bắt kịp xu thế ứng dụng in-tơ-nét vạn vật - Báo Nhân Dân điện tử.” [Online]. Available: <http://www.nhandan.com.vn/khoahoc-congnghe/item/39999802-bat-kip-xu-the-ung-dung-in-to-net-van-vat.html>. [Accessed: 29-May-2019].
- [2] “ECMA-334 5 th Edition / C# Language Specification,” 2017.
- [3] “Asp.net Core những thay đổi lớn cho lập trình viên .NET - Viblo.” [Online]. Available: <https://viblo.asia/p/aspnet-core-nhung-thay-doi-lon-cho-lap-trinh-vien-net-Qbq5QqxL5D8>. [Accessed: 29-May-2019].
- [4] “What is CSHTML – Web Development.” [Online]. Available: <http://johnharbison.net/what-is-cshtml/>. [Accessed: 29-May-2019].
- [5] “Lập trình MVC với Razor view engine - M21Love.” [Online]. Available: <https://m21love.blogspot.com/2012/07/lap-trinh-mvc-voi-razor-view-engine.html>. [Accessed: 29-May-2019].
- [6] “Tìm hiểu tổng quan về JQuery Ajax - Viblo.” [Online]. Available: <https://viblo.asia/p/tim-hieu-tong-quan-ve-jquery-ajax-PwRGgmmrkEd>. [Accessed: 29-May-2019].
- [7] “SQL là gì? - VietTuts.” [Online]. Available: <https://viettuts.vn/sql/sql-la-gi>. [Accessed: 30-May-2019].
- [8] A. Cangialosi and A. Cangialosi, “The Visual Studio Gallery gets a little more community friendly,” *Anthony @ MS*, 2014.
- [9] “Language Support in Visual Studio Code,” *Vis. Stud. Code*, 2016.
- [10] “SQL server là gì? Cập nhật hệ tính năng mới cho Microsoft SQL Server - Techblog của VCCloud.” [Online]. Available: <https://tech.vccloud.vn/sql-server-la-gi-cap-nhat-he-tinh-nang-moi-cho-microsoft-sql-server-20181018112935543.htm>. [Accessed: 29-Mar-2019].
- [11] “API là gì? Những đặc điểm nổi bật của Web API - Viblo.” [Online]. Available: <https://viblo.asia/p/api-la-gi-nhung-dac-diem-noi-bat-cua-web-api-Qpmle9L9lrd>. [Accessed: 02-Jun-2019].