

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG QUẢN LÝ RỦI RO VÀ CẢNH BÁO SỰ CỐ TRONG DOANH NGHIỆP

Người hướng dẫn: THS. NGUYỄN THỊ XUÂN THỦY

Sinh viên thực hiện: LÊ ĐĂNG HUY

Mã số sinh viên: 2151220304

Lớp: 21CT3

Đà Nẵng, 11/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG QUẢN LÝ RỦI RO VÀ
CẢNH BÁO SỰ CỐ TRONG DOANH NGHIỆP**

Người hướng dẫn: **THS. NGUYỄN THỊ XUÂN THỦY**

Sinh viên thực hiện: **LÊ ĐĂNG HUY**

Mã số sinh viên: 2151220304

Lớp: **21CT3**

Đà Nẵng, 11/2025

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

TÓM TẮT

Tên đề tài: **Xây dựng hệ thống quản lý rủi ro và cảnh báo sự cố trong doanh nghiệp**

Sinh viên thực hiện: **Lê Đăng Huy**

Mã số SV: 2151220304 Lớp: 21CT3

Đề tài “**Xây dựng hệ thống quản lý rủi ro và cảnh báo sự cố trong doanh nghiệp**” nhằm thiết kế và phát triển một hệ thống web hỗ trợ doanh nghiệp phát hiện, theo dõi và xử lý rủi ro một cách hiệu quả. Hệ thống cho phép nhân viên báo cáo sự cố nhanh chóng, quản lý nhận cảnh báo theo thời gian thực (Realtime) và bộ phận xử lý được phân công tự động dựa trên loại rủi ro. Bên cạnh đó, hệ thống cung cấp các chức năng theo dõi tiến độ xử lý, ghi log quá trình xử lý, phân tích mức độ nghiêm trọng, thống kê số lượng sự cố và hỗ trợ nhà quản lý đưa ra quyết định kịp thời.

Đề tài ứng dụng các công nghệ như Spring Boot, WebSocket, Angular và PostgreSQL để đảm bảo khả năng hoạt động ổn định, realtime và thân thiện với người dùng. Kết quả đạt được cho thấy hệ thống hỗ trợ tốt việc quản lý rủi ro trong doanh nghiệp, giảm thiểu thời gian phản hồi khi xảy ra sự cố, nâng cao hiệu quả vận hành và góp phần đảm bảo an toàn trong môi trường làm việc.

LỜI NÓI ĐẦU

Trong quá trình hoạt động, các doanh nghiệp thường phải đối mặt với nhiều rủi ro và sự cố như hỏng hóc máy móc, chập điện, cháy nổ hay tai nạn lao động. Việc quản lý những vấn đề này chủ yếu dựa vào ghi chép thủ công hoặc báo cáo rời rạc, dẫn đến chậm trễ và thiếu đồng bộ trong xử lý.

Từ nhu cầu thực tế đó, đề tài “Xây dựng hệ thống quản lý rủi ro và cảnh báo sự cố trong doanh nghiệp” được triển khai nhằm hỗ trợ doanh nghiệp số hóa quy trình báo cáo và xử lý sự cố. Hệ thống giúp:

Rút ngắn thời gian phản hồi và xử lý.

Cảnh báo sự cố theo thời gian thực.

Theo dõi tiến độ minh bạch, tránh bỏ sót.

Thông kê và đánh giá rủi ro để cải thiện an toàn.

Nhờ vậy, doanh nghiệp có thể chủ động hơn trong công tác phòng ngừa, giảm thiểu thiệt hại và nâng cao hiệu quả vận hành.

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến cô ThS. Nguyễn Thị Xuân Thủy, giảng viên hướng dẫn đã tận tình chỉ bảo, định hướng và hỗ trợ em trong suốt quá trình thực hiện đồ án tốt nghiệp. Sự tận tâm, trách nhiệm và những góp ý quý báu của cô đã giúp em hoàn thiện đề tài một cách hiệu quả.

Em xin chân thành cảm ơn quý thầy cô Trường Đại học Kiến trúc Đà Nẵng đã truyền đạt cho em những kiến thức và kỹ năng cần thiết trong suốt quá trình học tập. Đây là nền tảng quan trọng để em có thể thực hiện và hoàn thành tốt đồ án này.

Em cũng xin gửi lời cảm ơn đến bạn bè và những người đã luôn đồng hành, chia sẻ và hỗ trợ em trong suốt thời gian thực hiện đồ án. Sự động viên và giúp đỡ của mọi người là nguồn động lực lớn giúp em vượt qua những khó khăn trong quá trình làm việc.

Dù đã cố gắng trong quá trình thực hiện, nhưng với kiến thức và kinh nghiệm còn hạn chế, đồ án chắc chắn vẫn còn những điểm chưa thật sự hoàn thiện. Em hy vọng sẽ nhận được sự góp ý chân thành từ thầy cô để có thể rút kinh nghiệm và hoàn thiện bản thân hơn trong con đường học tập và công việc sau này.

CAM ĐOAN

Em xin cam đoan rằng tất cả các kết quả và thông tin trình bày trong đề tài này là thành quả của công việc của em dưới sự hướng dẫn của ThS.Nguyễn Thị Xuân Thuý. Em đã tuân thủ mọi nguyên tắc và quy định về nghiên cứu khoa học và đạo đức trong việc thực hiện khóa luận này.

Tất cả các nguồn tham khảo và công trình nghiên cứu của người khác đã được em trích dẫn và trình bày một cách rõ ràng và chi tiết trong danh mục tài liệu tham khảo của khóa luận. Em đã tuân thủ nguyên tắc không sao chép, không vi phạm bản quyền, và đã tuân thủ đúng quy định về việc chỉ rõ nguồn gốc thông tin từ nguồn khác.

Các kết quả đánh giá và thống kê được trình bày trong khóa luận này đều là kết quả của những thực nghiệm mà em đã tự thực hiện, và không có sự can thiệp từ bất kỳ nguồn nào khác. Em cam đoan tính trung thực và chính xác trong việc trình bày kết quả và thông tin trong khóa luận này.

Sinh viên thực hiện

Lê Đăng Huy

MỤC LỤC

LỜI NÓI ĐẦU	i
LỜI CẢM ƠN	ii
CAM ĐOAN.....	iii
MỤC LỤC	iv
DANH MỤC BẢNG BIỂU	vii
DANH MỤC HÌNH ẢNH	ix
DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT	xii
MỞ ĐẦU	1
CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	2
1.1. Tổng quan đề tài	2
1.2. Đối tượng nghiên cứu	2
1.3. Phạm vi nghiên cứu	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN	4
2.1. Cơ sở lý thuyết	4
2.1.1. Tổng quan về hệ thống thông tin quản lý	4
2.1.2. Tìm hiểu về rủi ro, quản lý rủi ro và sự cố trong doanh nghiệp	4
2.1.3. Chu trình quản lý rủi ro	4
2.1.4. Tìm hiểu về hệ thống cảnh báo sự cố	5
2.1.5. Công nghệ Web Realtime	5
2.1.6. Trí tuệ nhân tạo trong quản lý rủi ro	9
2.1.7. Công nghệ Frontend Angular	11
2.1.8. Công nghệ Backend Spring Boot	12
2.2. Công cụ sử dụng	15
2.2.1. IntelliJ IDEA	15
2.2.2. Visual Studio Code	19

2.2.3. PostgreSQL	22
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN	26
3.1. Mô tả và phân tích bài toán	26
3.1.1. Bối cảnh bài toán.....	26
3.1.2. Mục tiêu của bài toán.....	26
3.1.3. Các tác nhân trong hệ thống.....	27
3.2. Phương pháp xử lý bài toán	28
3.3. Biểu đồ Use Case.....	29
3.3.1. Use Case tổng quát	29
3.3.2. Mô tả usecase mức chi tiết	30
3.4. Biểu đồ hoạt động	50
3.4.1. Biểu đồ hoạt động chức năng báo cáo sự cố.....	50
3.4.2. Biểu đồ hoạt động chức năng nhận cảnh báo và phân công bộ phận xử lý	52
3.4.3. Biểu đồ hoạt động chức năng bộ phận xử lý nhận phân công sự cố và phân công nhiệm vụ đến người xử lý.....	54
3.4.4. Biểu đồ hoạt động chức năng người xử lý nhận phân công nhiệm vụ, xử lý sự cố và hoàn thành nhiệm vụ.	56
3.5. Biểu đồ tuần tự.....	57
3.6. Thiết lập cơ sở dữ liệu	63
3.6.1. Mô hình ERD.....	63
3.6.2. Cấu trúc các bảng.....	63
CHƯƠNG 4. XÂY DỰNG HỆ THỐNG QUẢN LÝ RỦI RO VÀ CẢNH BÁO SỰ CỐ	71
4.1. Môi trường phát triển và công cụ lập trình	71
4.2. Triển khai hệ thống.....	71
4.2.1. Triển khai Frontend Angular.....	71
4.2.2. Triển khai backend Spring Boot.....	71

4.3. Kết quả đạt được	73
4.3.1. Giao diện nhân viên	73
4.3.2. Giao diện quản lý.....	75
4.3.3. Giao diện Bộ phận xử lý.....	78
4.3.4. Giao diện người xử lý.....	81
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	83
TÀI LIỆU THAM KHẢO	84

DANH MỤC BẢNG BIỂU

Bảng 2.1. So sánh WebSocket, SSE và REST Polling	7
Bảng 2.2. Một số mô hình AI phổ biến	10
Bảng 3.1. Đặc tả usecase đăng nhập.....	30
Bảng 3.2. Đặc tả usecase báo cáo sự cố	31
Bảng 3.3. Đặc tả usecase xem danh sách sự cố	33
Bảng 3.4. Đặc tả usecase xem chi tiết sự cố.....	34
Bảng 3.5. Đặc tả usecase phân công bộ phận xử lý	35
Bảng 3.6. Đặc tả usecase xem danh sách sự cố.....	36
Bảng 3.7. Đặc tả usecase phân công người xử lý.....	38
Bảng 3.8. Đặc tả usecase danh sách nhiệm vụ được giao	40
Bảng 3.9. Đặc tả usecase nhận nhiệm vụ	41
Bảng 3.10. Đặc tả usecase thêm bình luận	43
Bảng 3.11. Đặc tả usecase hoàn thành nhiệm vụ.....	44
Bảng 3.12. Đặc tả usecase đóng sự cố.....	45
Bảng 3.13. Đặc tả usecase từ chối sự cố	47
Bảng 3.14. Đặc tả usecase quản lý user.....	49
Bảng 3.15. User	63
Bảng 3.16. roles	64
Bảng 3.17. departments	64
Bảng 3.18. risk_categories	65
Bảng 3.19. locations	65
Bảng 3.20. incidents	65
Bảng 3.21. incident_media	66
Bảng 3.22. incident_comments	66
Bảng 3.23. incident_logs	67

Bảng 3.24. incident_alerts	67
Bảng 3.25. notifications.....	67
Bảng 3.26. assignments	68
Bảng 3.27. department_assignments	68
Bảng 3.28. sla_config	69
Bảng 3.29. support_requests.....	69

DANH MỤC HÌNH ẢNH

Hình 2.1. Nguyên lý hoạt động của WebSocket.....	6
Hình 2.2. Logo Spring Boot	12
Hình 2.3. Logo IntelliJ IDEA	15
Hình 2.4. Trang download IntelliJ IDEA	17
Hình 2.5. Cấu hình IntelliJ IDEA.....	18
Hình 2.6. Giao diện khi hoàn thành.....	18
Hình 2.7. Logo VS Code	19
Hình 2.8. Giao diện trang download	20
Hình 2.9. Tạo dự án mới	21
Hình 2.10. Logo PostgreSQL	22
Hình 2.11. Giao diện download.....	23
Hình 2.12. Chạy file vừa download.....	23
Hình 2.13. Cài đặt cấu hình	24
Hình 2.14. Giao diện kết nối cơ sở dữ liệu.....	25
Hình 2.15. Cài đặt thành công	25
Hình 3.1. Luồng thực thi hệ thống	28
Hình 3.2. Usecase tổng quát	29
Hình 3.3. Usecase đăng nhập	30
Hình 3.4. Usecase báo cáo sự cố	31
Hình 3.5. Usecase xem danh sách sự cố.....	32
Hình 3.6. Usecase xem chi tiết sự cố.....	34
Hình 3.7. Usecase phân công bộ phận xử lý	35
Hình 3.8. Usecase xem danh sách sự cố của bộ phận xử lý	36
Hình 3.9. Usecase phân công người xử lý.....	38
Hình 3.10. Usecase danh sách nhiệm vụ được giao	39

Hình 3.11. Usecase nhận nhiệm vụ	41
Hình 3.12. Usecase thêm bình luận	42
Hình 3.13. Usecase hoàn thành nhiệm vụ	44
Hình 3.14. Usecase đóng sự cố	45
Hình 3.15. Usecase từ chối sự cố	47
Hình 3.16. Usecase quản lý user	49
Hình 3.17. Biểu đồ hoạt động chức năng báo cáo sự cố	51
Hình 3.18. Biểu đồ hoạt động chức năng nhận cảnh báo và phân công bộ phận xử lý	52
Hình 3.19. Biểu đồ hoạt động chức năng bộ phận xử lý nhận cảnh báo và phân công người xử lý.....	54
Hình 3.20. Biểu đồ hoạt động các chức năng của người xử lý	56
Hình 3.21. Biểu đồ tuần sự nhân viên báo cáo sự cố và quản lý nhận cảnh báo	57
Hình 3.22. Biểu đồ tuần sự quản lý phân công và bộ phận xử lý nhận realtime.....	59
Hình 3.23. Biểu đồ tuần sự bộ phận xử lý phân công người xử lý	60
Hình 3.24. Biểu đồ tuần sự người xử lý hoàn thành nhiệm vụ và quản lý đóng sự cố	61
Hình 3.25. Thiết kế cơ sở dữ liệu cho hệ thống	63
Hình 4.1. Giao diện trang đăng nhập.....	73
Hình 4.2. Giao diện form báo cáo sự cố.....	74
Hình 4.3. Giao diện khi sử dụng AI.....	74
Hình 4.4. Giao diện danh sách sự cố của tôi	75
Hình 4.5. Giao diện hiển thị cảnh báo sự cố mới	75
Hình 4.6. Giao diện danh sách sự cố	76
Hình 4.7. Giao diện từ chối sự cố.....	76
Hình 4.8. Giao diện chi tiết sự cố và chức năng phân công	77
Hình 4.9. Giao diện hiển thị giao thành công.....	77
Hình 4.10. Giao diện dashboard	78
Hình 4.11. Giao diện nhận cảnh báo sự cố mới được giao.....	78
Hình 4.12. Giao diện danh sách sự cố	79

Hình 4.13. Giao diện danh sách yêu cầu hỗ trợ.....	79
Hình 4.14. Giao diện hiển thị chi tiết sự cố được giao.....	80
Hình 4.15. Giao diện thông báo phân công.....	80
Hình 4.16. Giao diện danh sách nhiệm vụ.....	81
Hình 4.17. Giao diện chi tiết nhiệm vụ	81
Hình 4.18. Giao diện hiển thị thông báo	82

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Từ viết tắt	Tiếng Anh	Nghĩa tiếng Việt
1	MIS	Management Information System	Hệ thống thông tin quản lý
2	AI	Artificial Intelligence	Trí tuệ nhân tạo
3	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
4	ML	Machine Learning	Học máy
5	SLA	Service Level Agreement	Cam kết mức độ dịch vụ
6	SPA	Single Page Application	Ứng dụng một trang
7	API	Application Programming Interface	Giao diện lập trình ứng dụng
8	JWT	JSON Web Token	Chuẩn token xác thực
9	JPA	Java Persistence API	API thao tác cơ sở dữ liệu
10	DB	Database	Cơ sở dữ liệu
11	CRUD	Create – Read – Update – Delete	Các thao tác dữ liệu cơ bản
12	UI	User Interface	Giao diện người dùng

13	UX	User Experience	Trải nghiệm người dùng
14	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
15	MVCC	Multi-Version Concurrency Control	Kiểm soát đồng thời nhiều phiên bản
16	IoT	Internet of Things	Internet vạn vật
17	SSE	Server-Sent Events	Gửi sự kiện từ máy chủ
18	REST	Representational State Transfer	Kiến trúc truyền tải tài nguyên
19	VS Code	Visual Studio Code	Trình soạn thảo mã nguồn
20	WS	WebSocket	Giao thức WebSocket

MỞ ĐẦU

Trong bối cảnh công nghiệp hóa và hiện đại hóa diễn ra mạnh mẽ, các doanh nghiệp phải đối mặt với nhiều loại rủi ro trong quá trình sản xuất và vận hành như hỏng hóc máy móc, chập điện, cháy nổ, tai nạn lao động hay sự cố kho bãi. Việc phát hiện và xử lý kịp thời những rủi ro này đóng vai trò quan trọng trong việc đảm bảo an toàn lao động, bảo vệ tài sản và duy trì hoạt động sản xuất ổn định. Do đó, nhu cầu ứng dụng các hệ thống quản lý rủi ro và cảnh báo sự cố theo thời gian thực (Realtime) ngày càng trở nên cấp thiết.

Trên thế giới, nhiều tập đoàn lớn như Siemens, General Electric (GE) hay Honeywell đã triển khai các hệ thống giám sát rủi ro ứng dụng IoT và trí tuệ nhân tạo nhằm phát hiện bất thường và cảnh báo sớm. Các nghiên cứu của những tổ chức học thuật như Đại học Stanford hay MIT cũng cho thấy rằng công nghệ Web Realtime có thể rút ngắn thời gian phản hồi và giảm thiểu thiệt hại đáng kể trong môi trường công nghiệp. Tại các quốc gia phát triển, các giải pháp như Industrial IoT Monitoring System hay Incident Management Platform đã trở thành công cụ quan trọng trong vận hành nhà máy và chuỗi cung ứng.

Tại Việt Nam, các doanh nghiệp trong lĩnh vực điện lực, dầu khí và logistics cũng đã quan tâm đến việc xây dựng hệ thống cảnh báo và giám sát rủi ro theo thời gian thực. Chẳng hạn, Tập đoàn Điện lực Việt Nam (EVN) triển khai giám sát trực tuyến tình trạng lưới điện, hay các đơn vị kho vận ứng dụng hệ thống cảnh báo cháy nổ tự động. Tuy nhiên, phần lớn các giải pháp hiện nay vẫn mang tính cục bộ, chưa tích hợp sâu với nền tảng web trực tuyến hoặc công nghệ AI để tự động phân loại và dự đoán mức độ nghiêm trọng của sự cố.

Chính vì vậy, việc nghiên cứu và xây dựng “**Hệ thống quản lý rủi ro và cảnh báo sự cố trong doanh nghiệp**” là cần thiết cả về lý luận và thực tiễn. Hệ thống hướng đến việc hỗ trợ nhân viên báo cáo sự cố nhanh chóng, giúp quản lý nhận cảnh báo ngay lập tức, bộ phận xử lý được phân công rõ ràng, đồng thời cung cấp công cụ thống kê và phân tích rủi ro để doanh nghiệp xây dựng chiến lược phòng ngừa hiệu quả hơn.

CHƯƠNG 1. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

1.1. Tổng quan đề tài

Trong bối cảnh chuyển đổi số và tự động hóa ngày càng mạnh mẽ, các doanh nghiệp phải đối mặt với nhiều rủi ro trong quá trình vận hành như sự cố kỹ thuật, an toàn lao động, vận hành thiết bị và sự cố hệ thống. Việc phát hiện và xử lý chậm trễ không chỉ ảnh hưởng đến hiệu suất làm việc mà còn tiềm ẩn nguy cơ thiệt hại lớn cho doanh nghiệp. Dự án **“Xây dựng hệ thống quản lý rủi ro và cảnh báo sự cố trong doanh nghiệp”** nhằm giúp doanh nghiệp chủ động phát hiện, theo dõi và xử lý sự cố một cách nhanh chóng và hiệu quả hơn. Thông qua việc ứng dụng công nghệ thông tin, hệ thống hỗ trợ tự động hóa quy trình quản lý rủi ro, rút ngắn thời gian phản hồi, đồng thời giúp nhà quản lý có cái nhìn tổng quan hơn về tình hình sự cố, từ đó đưa ra biện pháp phòng ngừa và cải tiến kịp thời.

1.2. Đối tượng nghiên cứu

- Nhân viên: Nghiên cứu cách nhân viên ghi nhận sự cố trong quá trình làm việc bao gồm quy trình báo cáo sự cố nhanh và đơn giản, cách cung cấp thông tin về sự cố, trải nghiệm giao diện báo cáo và theo dõi tình trạng xử lý.

- Quản lý hệ thống: Tìm hiểu nhu cầu toàn hệ thống bao gồm xem danh sách sự cố của toàn doanh nghiệp, phân công sự cố cho các bộ phận xử lý, theo dõi tiến độ và tình trạng xử lý, thống kê và phân tích báo cáo rủi ro.

- Quản lý bộ phận xử lý: Nghiên cứu yêu cầu về việc tiếp nhận sự cố liên quan đến bộ phận mình quản lý bao gồm nhận thông báo khi có sự cố được giao, kiểm tra thông tin sự cố, phân công kỹ thuật viên phù hợp để xử lý, theo dõi tiến trình xử lý và hạn SLA của kỹ thuật viên.

- Người xử lý: Tìm hiểu nhu cầu thực tế của người xử lý bao gồm nhận nhiệm vụ được phân công qua realtime, xác nhận nhận việc, cập nhật quá trình xử lý, cập nhật trạng thái hoàn thành của sự cố sau khi xử lý và theo dõi các nhiệm vụ đang được phụ trách.

1.3. Phạm vi nghiên cứu

Dự án tập trung nghiên cứu và xây dựng những chức năng cốt lõi phục vụ quy trình quản lý sự cố trong doanh nghiệp bao gồm:

- Ghi nhận, báo cáo và mô tả sự cố từ nhân viên.

- Phân loại rủi ro và đánh giá mức độ nghiêm trọng của sự cố.
- Phân công xử lý cho các bộ phận liên quan và kỹ thuật viên.
- Xử lý sự cố, cập nhật trạng thái và theo dõi tiến độ.
- Cảnh báo realtime khi có sự cố mới, phân công mới hoặc quá hạn SLA.
- Thống kê, tổng hợp và báo cáo tình hình rủi ro.

Phạm vi đối tượng sử dụng tập trung vào 4 nhóm người dùng chính: Nhân viên báo cáo sự cố, quản lý hệ thống, quản lý bộ phận xử lý, nhân viên xử lý sự cố (kỹ thuật viên).

Phạm vi nghiệp vụ của dự án tập trung vào quy trình quản lý rủi ro và cảnh báo sự cố nội bộ bao gồm:

- Sự cố kỹ thuật trong vận hành máy móc.
- Sự cố an toàn lao động.
- Sự cố liên quan đến thiết bị văn phòng.
- Sự cố hệ thống, môi trường làm việc.

Hệ thống được xây dựng trong phạm vi:

- Backend: Spring Boot
- Frontend: Angular
- Cơ sở dữ liệu: PostgreSQL
- Realtime: WebSocket
- AI: NLP + Machine Learning cơ bản để phân loại sự cố và dự đoán mức độ nghiêm trọng.
- Triển khai: Môi trường cục bộ (local) và có thể mở rộng triển khai cloud.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN

2.1. Cơ sở lý thuyết

2.1.1. Tổng quan về hệ thống thông tin quản lý

Hệ thống thông tin quản lý (MIS) là một hệ thống dùng để thu thập, lưu trữ, xử lý và cung cấp thông tin nhằm hỗ trợ các nhà quản lý ra quyết định, lập kế hoạch, kiểm soát và điều hành hoạt động trong tổ chức. MIS kết hợp giữa con người, quy trình, dữ liệu và công nghệ để đảm bảo thông tin được truyền tải chính xác, kịp thời và có ý nghĩa.

MIS đóng vai trò quan trọng trong doanh nghiệp vì nó giúp tự động hóa các quy trình làm việc, cải thiện hiệu suất, giảm sai sót và cung cấp cái nhìn tổng quan về hoạt động tổ chức. Hệ thống thông tin quản lý thường bao gồm các thành phần như cơ sở dữ liệu, phần mềm ứng dụng, phần cứng, mạng, và nhân sự vận hành hệ thống.

Nhờ MIS, doanh nghiệp có thể phân tích dữ liệu, theo dõi hiệu suất, dự báo xu hướng, và tối ưu hóa các nguồn lực. Đây là nền tảng quan trọng hỗ trợ hoạt động quản lý trong hầu hết các lĩnh vực như tài chính, nhân sự, bán hàng, sản xuất, chuỗi cung ứng và chăm sóc khách hàng.

2.1.2. Tìm hiểu về rủi ro, quản lý rủi ro và sự cố trong doanh nghiệp

Rủi ro (Risk) là những sự kiện hoặc các điều kiện tiềm tàng có thể xảy ra, gây ra những thiệt hại, tổn thất (tài chính, thị trường, nhân sự, ...) hoặc ảnh hưởng tiêu cực đến hoạt động sản xuất, kinh doanh như gián đoạn vận hành, hư hỏng máy móc, mất an toàn lao động, cháy nổ, lỗi phần mềm, ... và khả năng đạt được các mục tiêu đã đề ra của một tổ chức.

Quản lý rủi ro (Risk Management) là quá trình có hệ thống nhằm nhận diện – phân tích – đánh giá – xử lý – theo dõi các rủi ro có thể phát sinh trong quá trình hoạt động của doanh nghiệp. Mục tiêu của quản lý rủi ro là giảm thiểu tác động tiêu cực, đảm bảo hoạt động an toàn, liên tục và hiệu quả.

Sự cố (Incident) là những sự kiện đã xảy ra làm gián đoạn tiến trình bình thường của các hoạt động, quy trình hoặc hệ thống và có thể thậm chí dẫn đến thiệt hại, tổn thất, thương tích hoặc nguy hiểm.

2.1.3. Chu trình quản lý rủi ro

Quy trình quản lý rủi ro tiêu chuẩn gồm 5 bước:

- Bước 1: Nhận diện rủi ro (Risk Identification): Xác định những sự cố có thể xảy ra thông qua khảo sát, điều tra, thống kê hoặc thông tin phản ánh từ nhân viên.
- Bước 2: Phân tích rủi ro (Risk Analysis): Xác định nguyên nhân, xác suất và mức độ ảnh hưởng của từng rủi ro.
- Bước 3: Đánh giá và phân loại rủi ro (Risk Evaluation): Xếp hạng theo mức độ ưu tiên như low, medium, high. Critical.
- Bước 4: Xử lý rủi ro (Risk Treatment): Các phương án xử lý phổ biến gồm giảm thiểu (Mitigate), tránh rủi ro (Avoid), chuyển giao (Transfer), chấp nhận rủi ro (Accept).
- Bước 5: Giám sát và báo cáo rủi ro (Monitoring & Review): Theo dõi tiến độ xử lý, cập nhật trạng thái (New – Processing – Completed – Rejected) và phân tích xu hướng sự cố để cải thiện hệ thống.

2.1.4. Tìm hiểu về hệ thống cảnh báo sự cố

Hệ thống cảnh báo sự cố là nền tảng thu thập thông tin sự cố từ người dùng, tự động phân tích mức độ nguy hiểm và gửi thông báo đến bộ phận xử lý thông qua nhiều hình thức như: Email, SMS, thông báo đẩy (push notification), WebSocket real-time.

Một hệ thống cảnh báo hiệu quả cần đảm bảo: Tốc độ phản hồi nhanh, cảnh báo đúng đối tượng, ghi log đầy đủ, theo dõi thời gian thực (Realtime), có phân quyền để đảm bảo bảo mật.

2.1.5. Công nghệ Web Realtime

2.1.5.1. Tổng quan về công nghệ Web Realtime

Công nghệ Web Realtime cho phép truyền dữ liệu giữa máy khách (client) và máy chủ (server) theo thời gian thực mà không cần tải lại trang hoặc gửi yêu cầu liên tục. Đây là nền tảng quan trọng trong các hệ thống cần cập nhật tức thời như: chat, giám sát, theo dõi sự cố, dashboard realtime, cảnh báo nguy hiểm, IoT...

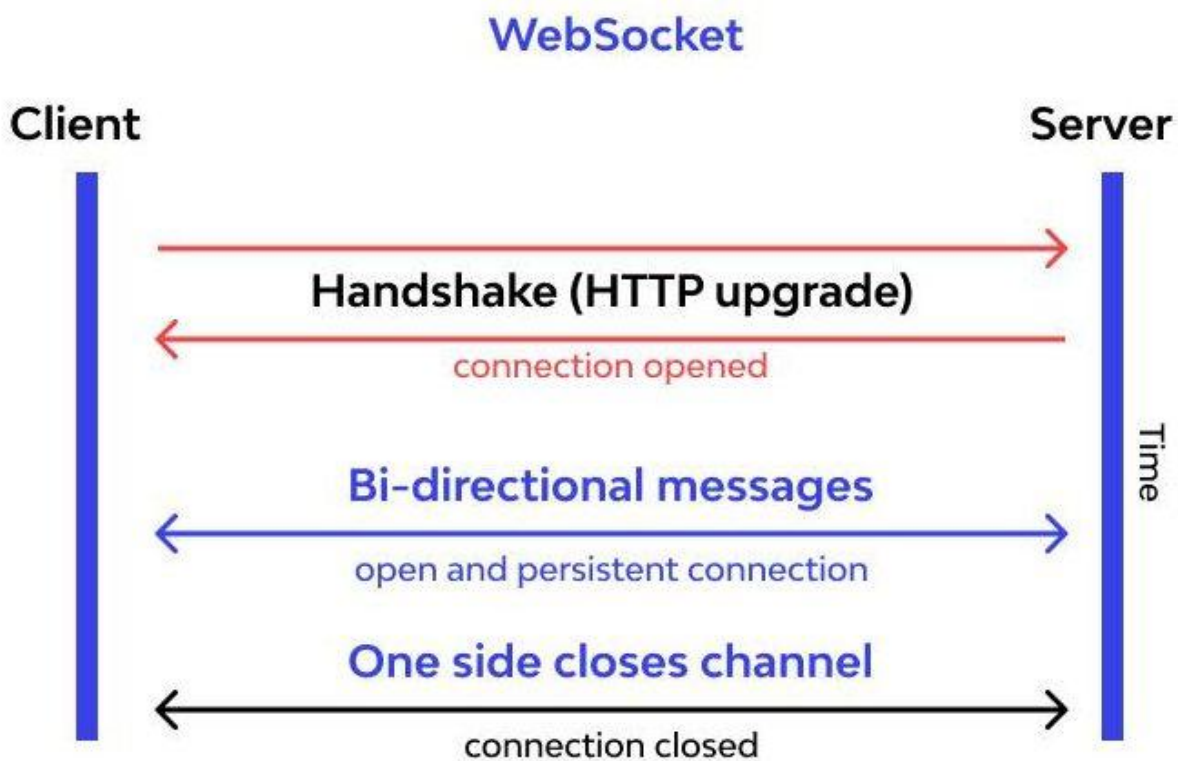
Trong các dự án hiện nay, phổ biến nhất là WebSocket, Server-Sent Events (SSE) và REST Polling/Long Polling.

2.1.5.2. Nguyên lý hoạt động của WebSocket

WebSocket là giao thức truyền thông hai chiều (full-duplex) hoạt động trên nền TCP, cho phép client và server trao đổi dữ liệu liên tục thông qua một kết nối duy nhất.

Cách thức hoạt động của giao thức WebSocket:

- Bước 1: Handshake – Thiết lập kết nối
 - Client gửi yêu cầu HTTP đặc biệt yêu cầu nâng cấp giao thức sang WebSocket.
 - Server phản hồi bằng mã 101 Switching Protocols.
 - Sau khi được chấp nhận, kết nối được nâng cấp thành kênh WebSocket.
- Bước 2: Truyền dữ liệu hai chiều (Full-duplex Communication)
 - Khi kết nối WebSocket được thiết lập, cả client và server đều có thể gửi dữ liệu bất kỳ lúc nào mà không cần chờ request từ bên còn lại.
 - Server có thể chủ động đẩy dữ liệu đến client khi có sự kiện mới.
 - Dữ liệu được truyền qua lại thông qua các frame (khung dữ liệu), kích thước nhỏ và tối ưu băng thông, cho phép thời gian phản hồi rất nhanh.
- Bước 3: Duy trì và đóng kết nối
 - Kết nối WebSocket được giữ nguyên cho đến khi một trong hai bên yêu cầu đóng.
 - Khi muốn đóng, client hoặc server gửi một close frame để thông báo kết thúc phiên làm việc.
 - Sau khi cả hai xác nhận, kết nối được giải phóng hoàn toàn.



Hình 2.1. Nguyên lý hoạt động của WebSocket

➤ Ưu điểm của WebSocket

- Độ trễ thấp, tốc độ cao
 - Trao đổi dữ liệu hai chiều
 - Tối ưu băng thông
 - Phù hợp các hệ thống yêu cầu realtime liên tục
- Nhược điểm của WebSocket
- Tốn tài nguyên nếu phải giữ nhiều kết nối cùng lúc
 - Việc scale hệ thống cần thêm công cụ (Redis Pub/Sub, MQ...)

2.1.5.3. So sánh WebSocket, SSE và REST Polling

Bảng sau thể hiện sự khác biệt giữa ba kỹ thuật truyền dữ liệu realtime phổ biến nhất:

Bảng 2.1. So sánh WebSocket, SSE và REST Polling

Tiêu chí	WebSocket	SSE (Server-Sent Events)	REST Polling / Long Polling
Cơ chế	Hai chiều (full-duplex)	Một chiều (server → client)	Client hỏi liên tục
Giao thức	WS/WSS (TCP)	HTTP/HTTPS	HTTP/HTTPS
Độ trễ	Rất thấp	Thấp	Cao
Băng thông	Thấp	Thấp	Cao
Tính realtime	Tốt nhất	Tốt	Kém
Trường hợp sử dụng	Chat, cảnh báo, IoT	Thông báo sự kiện	Hệ thống nhỏ
Ưu điểm	Nhanh, hiệu quả	Dễ triển khai	Đơn giản
Nhược điểm	Cần giữ kết nối	Chỉ 1 chiều	Lãng phí tài nguyên

2.1.5.4. Ứng dụng Web Realtime trong hệ thống cảnh báo sự cố

Trong hệ thống quản lý rủi ro và cảnh báo sự cố, tốc độ truyền tải thông tin giữa các vai trò như nhân viên, quản lý và bộ phận xử lý có ý nghĩa quan trọng trong việc rút ngắn thời gian phản hồi. Công nghệ Web Realtime cho phép dữ liệu được cập nhật ngay lập tức thay vì phải tải lại trang hoặc gửi yêu cầu lặp lại.

Quy trình ứng dụng Web Realtime trong hệ thống có thể mô tả như sau:

(1) Nhân viên báo cáo sự cố → Gửi cảnh báo realtime đến quản lý

Khi nhân viên tạo một báo cáo sự cố:

- Dữ liệu sự cố được gửi lên server.
- Server lập tức đẩy thông báo realtime đến giao diện của quản lý thông qua WebSocket.
- Màn hình quản lý ngay lập tức hiển thị sự cố mới (không cần refresh).

Điều này giúp quản lý nhận biết tình huống ngay khi sự cố phát sinh.

(2) Quản lý phân công xử lý → Gửi nhiệm vụ realtime đến bộ phận xử lý

Sau khi xem xét sự cố, quản lý sử dụng hệ thống để giao nhiệm vụ cho bộ phận xử lý phù hợp.

- Hệ thống đẩy thông tin phân công đến bộ phận xử lý theo thời gian thực.
- Các thành viên bộ phận xử lý nhận được thông báo nhiệm vụ ngay lập tức.
- Dashboard bộ phận xử lý được cập nhật realtime số lượng sự cố cần xử lý.

Việc phân công không còn phụ thuộc vào email hoặc hệ thống chờ đợi truyền thông.

(3) Bộ phận xử lý giao nhiệm vụ cho người xử lý → Cập nhật realtime

Bộ phận xử lý tiếp tục giao nhiệm vụ cho người xử lý trực tiếp (cá nhân chịu trách nhiệm xử lý sự cố):

- Khi giao nhiệm vụ, hệ thống gửi thông báo realtime đến tài khoản của người xử lý.
- Giao diện hiển thị ngay công việc mới, mức độ sự cố, thời hạn SLA, yêu cầu xử lý...
- Người xử lý có thể phản hồi hoặc gửi log xử lý ngay lập tức.

Điều này giúp giảm thời gian truyền thông nội bộ và hạn chế thiếu sót thông tin.

(4) Cập nhật trạng thái xử lý sự cố theo thời gian thực

Trong suốt quá trình xử lý:

- Người xử lý thay đổi trạng thái (Processing → Completed → Verified...)
- Bộ phận xử lý, quản lý và người báo cáo đều nhận được cập nhật realtime.

- Log xử lý (bình luận, hình ảnh, file đính kèm) cũng được đồng bộ theo thời gian thực.

Mọi bên liên quan đều theo dõi được tiến độ mà không cần reload trang.

(5) Dashboard thống kê realtime cho quản lý tổng và ban điều hành

Web Realtime giúp:

- Hiện thị số sự cố mới theo thời gian thực
- Cập nhật số lượng sự cố đang xử lý
- Tự động tăng số lượng sự cố đã hoàn thành
- Cảnh báo realtime khi có sự cố nghiêm trọng (High/Critical)

Điều này hỗ trợ lãnh đạo đưa ra quyết định nhanh và chính xác hơn.

2.1.6. Trí tuệ nhân tạo trong quản lý rủi ro

2.1.6.1. NLP để phân loại mô tả sự cố

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực của trí tuệ nhân tạo cho phép máy tính hiểu, phân tích và xử lý ngôn ngữ của con người.

Trong hệ thống quản lý rủi ro, NLP có vai trò quan trọng trong việc phân tích và phân loại tự động mô tả sự cố dựa trên nội dung văn bản mà người dùng nhập vào.

Cụ thể, khi nhân viên báo cáo sự cố, mô tả hoặc tiêu đề sẽ được mô hình NLP xử lý để xác định loại rủi ro tương ứng như sự cố kỹ thuật, sự cố an toàn lao động, cháy nổ, sức khỏe, môi trường,... mà không cần thao tác chọn thủ công.

Quy trình xử lý thường bao gồm:

- Tiền xử lý ngôn ngữ: loại bỏ ký tự đặc biệt, chuẩn hóa văn bản, tách từ và loại bỏ từ dừng.
- Biểu diễn văn bản: chuyển mô tả sự cố thành vector số bằng các phương pháp như TF-IDF, Word2Vec, hoặc BERT Embedding.
- Phân loại nội dung: sử dụng các mô hình học máy như Naive Bayes, Support Vector Machine (SVM) hoặc mô hình ngôn ngữ tiên tiến như GPT/BERT để gán nhãn loại sự cố.

Việc ứng dụng NLP giúp hệ thống tự động hóa bước phân loại, đảm bảo tính nhất quán trong dữ liệu và tăng tốc độ xử lý khi có nhiều sự cố được ghi nhận đồng thời.

2.1.6.2. Machine Learning trong dự đoán mức độ nghiêm trọng

Học máy (Machine Learning – ML) là lĩnh vực cho phép hệ thống máy tính học từ dữ liệu quá khứ để dự đoán hoặc phân loại dữ liệu mới mà không cần lập trình tường minh.

Trong quản lý rủi ro, ML được sử dụng để dự đoán mức độ nghiêm trọng của sự cố (severity) dựa trên các yếu tố như:

- Loại sự cố và danh mục rủi ro;
- Tiêu đề và mô tả chi tiết sự cố;
- Vị trí, thời gian và tần suất xảy ra;
- Dữ liệu lịch sử các sự cố tương tự trước đó.

Một số thuật toán phổ biến:

- Decision Tree, Random Forest: dễ giải thích, phù hợp với dữ liệu có cấu trúc.
- XGBoost, LightGBM: đạt độ chính xác cao trong các bài toán phân loại.
- Neural Network (MLP): xử lý được mối quan hệ phi tuyến phức tạp giữa các thuộc tính.

Việc áp dụng ML giúp hệ thống tự động xác định mức độ rủi ro cao hay thấp, hỗ trợ nhà quản lý ưu tiên xử lý và cảnh báo sớm đối với các sự cố nghiêm trọng, góp phần nâng cao hiệu quả quản trị.

2.1.6.3. Một số mô hình AI tham khảo

Để triển khai các chức năng phân tích và dự đoán trong hệ thống quản lý rủi ro, có thể tham khảo và ứng dụng nhiều mô hình AI khác nhau, cụ thể như sau:

Bảng 2.2. Một số mô hình AI phổ biến

Nhóm mô hình	Mục đích sử dụng
BERT / PhoBERT / RoBERTa	Mô hình ngôn ngữ dựa trên Transformer, dùng để hiểu ngữ cảnh và trích xuất thông tin từ mô tả sự cố.
GPT (Generative Pre-trained Transformer)	Dùng trong Chat AI để sinh phản hồi, hỗ trợ phân loại sự cố và gợi ý mức độ rủi ro.
Random Forest / XGBoost	Phân loại và dự đoán mức độ nghiêm trọng của sự cố dựa trên dữ liệu lịch sử.

K-Means / DBSCAN	Phân cụm các sự cố có đặc điểm tương tự, phát hiện điểm bất thường (anomaly detection).
RNN / LSTM	Phân tích chuỗi thời gian để dự đoán xu hướng và tần suất sự cố trong tương lai.

Các mô hình này có thể được huấn luyện và tích hợp vào hệ thống để tự động phân tích dữ liệu sự cố, giúp ra quyết định nhanh hơn, chính xác hơn và hướng tới quản trị rủi ro chủ động thay vì phản ứng bị động.

2.1.7. Công nghệ *Frontend Angular*

2.1.7.1. Giới thiệu tổng quan *Angular*

Angular là một nền tảng và framework mã nguồn mở do Google phát triển, được sử dụng để xây dựng các ứng dụng web đơn trang (SPA – Single Page Application). Angular hoạt động dựa trên kiến trúc hướng thành phần (Component-based), cho phép chia ứng dụng thành các module và component nhỏ, dễ quản lý, dễ bảo trì và tái sử dụng.

Angular sử dụng TypeScript làm ngôn ngữ chính, cung cấp khả năng kiểm tra kiểu (type-checking), giúp phát hiện lỗi sớm và tăng tính ổn định của ứng dụng. Framework này tích hợp nhiều tính năng mạnh mẽ như:

- Two-way Data Binding – đồng bộ dữ liệu hai chiều giữa view và model
- Dependency Injection (DI) – quản lý phụ thuộc giữa các thành phần
- Directive – mở rộng HTML để tạo giao diện động
- Routing – điều hướng giữa các trang trong SPA
- HTTP Client – giao tiếp với API backend
- RxJS – xử lý dữ liệu bất đồng bộ dựa trên mô hình Observable

Nhờ sự kết hợp toàn diện này, Angular trở thành lựa chọn phổ biến cho các ứng dụng web hiện đại, đặc biệt khi yêu cầu tính mở rộng, hiệu suất và khả năng bảo trì cao.

2.1.7.2. Lợi ích khi sử dụng *Angular*

- Phát triển nhanh và hiệu quả: Angular cung cấp Angular CLI, hệ thống module rõ ràng và công cụ tự động hoá giúp lập trình viên tạo, build, test và triển khai ứng dụng nhanh chóng.
- Mở rộng dễ dàng: Kiến trúc component và module cho phép ứng dụng mở rộng theo chiều ngang hoặc dọc mà không ảnh hưởng cấu trúc tổng thể.
- Bảo trì tốt: Cấu trúc mã rõ ràng, phân tầng hợp lý giúp việc bảo trì, sửa lỗi hoặc nâng cấp tính năng dễ dàng hơn.

- Giao diện mượt mà: Cơ chế binding và cơ chế change detection giúp cải thiện hiệu năng và trải nghiệm người dùng.
- Hệ sinh thái phong phú: Angular có cộng đồng lớn, tài liệu đầy đủ, nhiều thư viện hỗ trợ như Angular Material, NgRx, ngx-chart,...

2.1.7.3. Ứng dụng của Angular

- Ứng dụng SPA (Single Page Application)
- Ứng dụng web doanh nghiệp: CRM, CMS, hệ thống nội bộ
- Ứng dụng web thương mại điện tử
- PWA (Progressive Web App) – ứng dụng web hoạt động như ứng dụng di động
- Dashboard quản trị, hiển thị dữ liệu realtime

2.1.8. Công nghệ Backend Spring Boot

2.1.8.1. Khái niệm

Spring Boot được xây dựng dựa trên nền của Spring Framework, phát triển bởi ngôn ngữ Java. Nó là một phiên bản có tính tự động hoá cao hơn và đơn giản hơn của Spring. Spring Boot khiến cho việc tạo và khởi chạy ứng dụng chỉ diễn ra trong vòng vài phút.



Hình 2.2. Logo Spring Boot

2.1.8.2. Ưu điểm nổi bật của Spring Boot

- Tăng tốc quá trình phát triển: Spring Boot cung cấp hệ thống cấu hình mặc định thông minh và cơ chế tự động cấu hình (Auto Configuration), giúp giảm thiểu cấu

hình thủ công. Nhờ đó, lập trình viên có thể tập trung vào logic nghiệp vụ thay vì xử lý cấu hình phức tạp.

- Cấu hình đơn giản – Không cần XML: Spring Boot loại bỏ phần lớn cấu hình XML truyền thống, sử dụng các annotation hiện đại, giúp mã nguồn gọn nhẹ và dễ quản lý hơn.
- Máy chủ nhúng (Embedded Server): Spring Boot đi kèm các máy chủ như Tomcat, Jetty, Undertow, cho phép chạy ứng dụng trực tiếp mà không cần triển khai lên server ngoài, hỗ trợ phát triển và triển khai nhanh chóng.
- Khả năng tích hợp mạnh mẽ: Vì được xây dựng trên hệ sinh thái Spring, Spring Boot dễ dàng tích hợp với:
 - Spring Data, Spring Security
 - JPA, Hibernate
 - WebSocket, Thymeleaf
 - Các dịch vụ bên ngoài (Kafka, Redis, RabbitMQ...)

Điều này giúp phát triển hệ thống lớn mà vẫn giữ cấu trúc rõ ràng, dễ mở rộng.

- Tự động cấu hình tối ưu: Spring Boot tự động cấu hình dựa trên thư viện có trong project. Ví dụ: nếu phát hiện spring-data-jpa → tự cấu hình datasource, entity manager... Điều này giúp giảm đáng kể sai sót cấu hình.
- Quản lý phụ thuộc hiệu quả: Spring Boot sử dụng Maven/Gradle với các “Starter Dependencies”, giúp quản lý thư viện dễ dàng, đồng bộ phiên bản và hạn chế xung đột phụ thuộc.
- Giám sát và quản lý ứng dụng dễ dàng: Với Spring Boot Actuator, nhà phát triển có thể theo dõi:
 - Health Check
 - Metrics
 - Log
 - Trạng thái ứng dụng
 - Thông tin cấu hình

Điều này hỗ trợ rất tốt cho môi trường sản xuất.

- Tính sẵn sàng cho môi trường production: Spring Boot cung cấp các tính năng như:
 - Logging mặc định
 - Quản lý cấu hình bên ngoài
 - Kiểm tra trạng thái hệ thống

- Metrics phục vụ giám sát
- Phù hợp cho các hệ thống lớn và yêu cầu độ ổn định cao
- Hỗ trợ kiểm thử mạnh mẽ: Spring Boot hỗ trợ kiểm thử tích hợp và kiểm thử web thông qua máy chủ HTTP nhúng, giúp viết và chạy test nhanh chóng.
- Hoạt động tốt trên môi trường Đám mây và Container: Spring Boot tích hợp mượt mà với:
 - Docker
 - Kubernetes
 - AWS, GCP, Azure

Giúp triển khai và mở rộng ứng dụng dễ dàng.

- Cộng đồng lớn – hệ sinh thái phong phú: Với cộng đồng rộng lớn và tài liệu phong phú, Spring Boot được hỗ trợ mạnh mẽ và liên tục cập nhật, giúp giảm thời gian tìm hiểu và xử lý vấn đề.

2.1.8.3. Kiến trúc của Spring Boot

Spring Boot dựa trên nguyên tắc "Convention over Configuration" với các thành phần chính:

- Spring IoC Container: Quản lý các beans và dependencies của ứng dụng.
- Auto-Configuration: Tự động cấu hình ứng dụng dựa trên các dependencies có trên classpath.
- Starter Dependencies: Cung cấp các bộ dependencies được cấu hình sẵn cho các tác vụ cụ thể.
- Actuator: Cung cấp các tính năng sản xuất như monitoring, metrics, health checks.
- Embedded Servers: Tomcat, Jetty hoặc Undertow được tích hợp sẵn, không cần triển khai riêng.

2.1.8.4. Spring data JPA

Spring Data JPA đơn giản hóa việc truy cập dữ liệu trong Spring Boot thông qua:

- Repository pattern
- Method name query generation
- Các annotation để ánh xạ entities với bảng dữ liệu
- JPQL (Java Persistence Query Language)
- Query by Example
- Auditing tự động

2.1.8.5. Spring Security

Spring Security cung cấp khung bảo mật toàn diện cho ứng dụng Spring Boot:

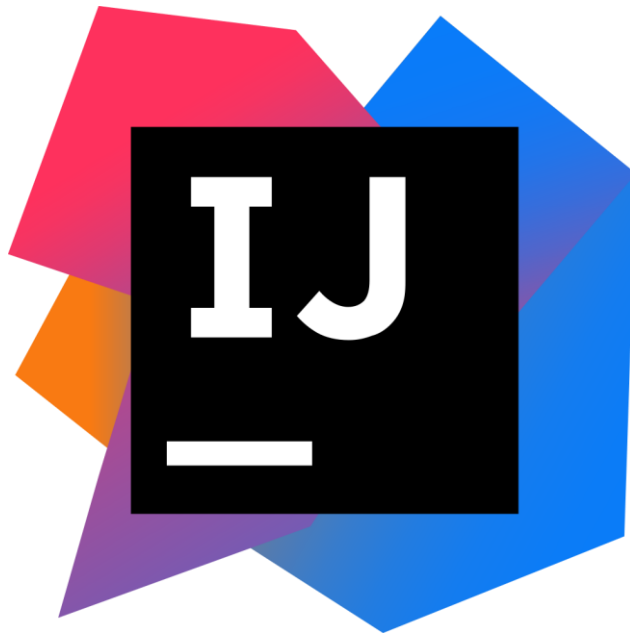
- Authentication: xác thực người dùng
- Authorization: phân quyền người dùng
- Protection against attacks (CSRF, Session Fixation)
- Servlet API integration
- Integration với các cơ chế xác thực như OAuth2, JWT

2.2. Công cụ sử dụng

2.2.1. IntelliJ IDEA

2.2.1.1. Giới thiệu về IntelliJ IDEA

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) mạnh mẽ được phát triển bởi JetBrains. Nó được thiết kế chủ yếu cho lập trình Java và Kotlin nhưng cũng hỗ trợ nhiều ngôn ngữ khác thông qua các plugin, với mục đích tăng năng suất lập trình bằng các tính năng thông minh như tự động hoàn thành mã, gỡ lỗi mạnh mẽ, phân tích mã và tái cấu trúc. IntelliJ có hai phiên bản chính: bản miễn phí (Community) và bản trả phí (Ultimate).



Hình 2.3. Logo IntelliJ IDEA

2.2.1.2. Lợi ích khi sử dụng IntelliJ IDEA

- Tập trung quản lý tác vụ: Cho phép quản lý toàn bộ tác vụ lập trình trong một IDE duy nhất, giúp làm việc hiệu quả hơn.

- Hỗ trợ viết và gỡ lỗi một cách linh hoạt: Cung cấp công cụ mạnh mẽ để viết mã, gỡ lỗi, tái cấu trúc, kiểm tra và tìm hiểu mã nguồn của bạn một cách linh hoạt mà không bị ảnh hưởng.
- Đa ngôn ngữ và đa công nghệ: Hỗ trợ nhiều ngôn ngữ lập trình như Java, Ruby, Kotlin, Groovy, Python và Scala, đồng thời hỗ trợ nhiều framework và công nghệ khác nhau.
- Tự động đảm bảo chất lượng mã nguồn: Phát hiện và gợi ý sửa lỗi ở mọi cấp độ, giúp giữ code sạch và chuẩn.
- Tạo mã nhanh và tối ưu: Cung cấp các công cụ giúp viết mã nhanh hơn và dễ duy trì hơn.
- Phù hợp mọi quy mô: Sử dụng tốt cho cá nhân, doanh nghiệp từ nhỏ đến lớn.
- Tích hợp mạnh mẽ với Git và CI/CD: Hỗ trợ Git, GitHub, TeamCity... giúp quy trình phát triển linh hoạt và chuyên nghiệp hơn.

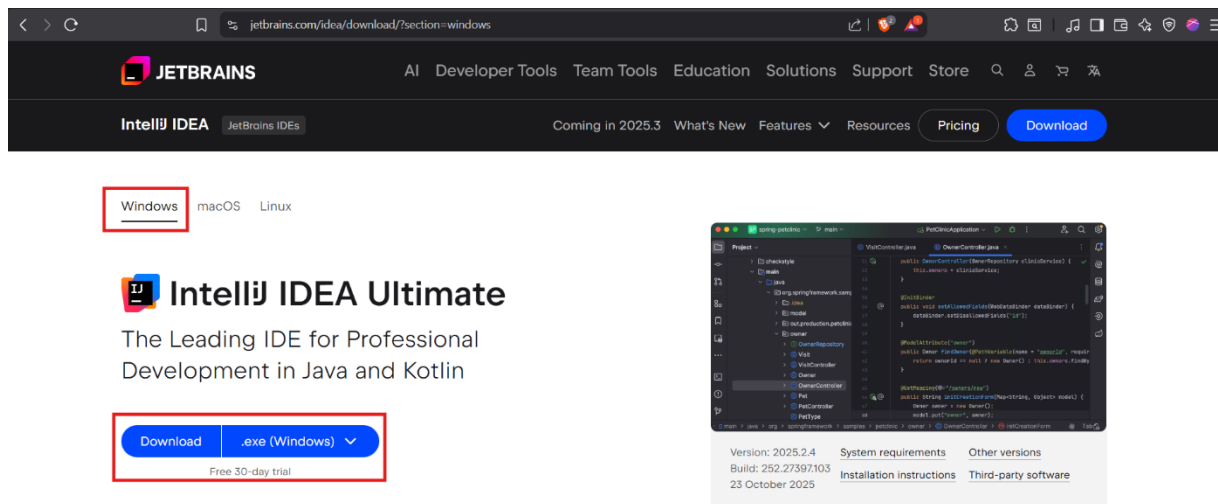
2.2.1.3. Các tính năng nổi bật của IntelliJ IDEA

- Hỗ trợ mã nguồn thông minh: Kiểm tra hơn 60 loại lỗi và phân tích mã cực nhanh giúp nhà phát triển phát hiện lỗi một cách nhanh chóng. Tự động gợi ý sửa lỗi và tạo code thống nhất, sạch và hiệu quả. Ngoài ra hỗ trợ nhiều ngôn ngữ và framework khác nhau bao gồm Java, JavaScript, HTML/CSS, PHP, Ruby, SQL,...
- Phân tích luồng dữ liệu (Data Flow Analysis): Công cụ tự động phân tích cách dữ liệu di chuyển trong chương trình, đưa ra gợi ý hoàn thành mã và phát hiện logic chính xác. IntelliJ còn tự thêm kiểu chuyển đổi (cast) khi cần, giúp mã chạy ổn định và hạn chế lỗi runtime.
- Tiêm ngôn ngữ (Language Injection): tính năng này cho phép nhúng các đoạn SQL, HTML, CSS, JavaScript trực tiếp trong chuỗi java một cách tự nhiên và linh hoạt, giúp làm việc đa ngôn ngữ trong cùng một file mà không cần đổi môi trường, tăng tốc độ phát triển.
- Tái cấu trúc ngôn ngữ chéo: Khi đổi tên class, phương thức hoặc entity, IntelliJ IDEA tự cập nhật tất cả vị trí liên quan trong toàn bộ dự án. Điều này giúp refactor nhanh, chính xác và giảm rủi ro lỗi do sửa thủ công.
- Phát hiện bản sao: IntelliJ tự nhận diện các đoạn mã trùng lặp và cảnh báo cho lập trình viên, giúp dễ dàng tái sử dụng mã, giảm lặp lại không cần thiết và cải thiện khả năng bảo trì.
- Soát lỗi và sửa nhanh: IDE cảnh báo ngay khi bạn viết sai và cung cấp gợi ý sửa lỗi thông qua phím **Alt + Enter**, giúp khắc phục nhanh các vấn đề từ cú pháp đến logic, tiết kiệm nhiều thời gian.

- Phím tắt cho mọi thao tác: IntelliJ IDEA hỗ trợ nhiều phím tắt mạnh mẽ cho điều hướng, tạo file, mở cửa sổ và chỉnh sửa, giúp tăng tốc độ làm việc và giảm phụ thuộc vào chuột.
- Trình gỡ lỗi inline: Khi debug, giá trị biến được hiển thị trực tiếp ngay tại dòng code, và tự động làm nổi bật khi thay đổi, giúp hiểu rõ luồng chạy và xử lý lỗi nhanh hơn.
- Hoàn thành chuỗi thông minh: IDE có khả năng gợi ý chính xác các chuỗi gọi phương thức phức tạp (như `module.getProject()`) chỉ với vài phím tắt, giúp viết mã nhanh và giảm sai sót cú pháp.
- Môi trường tích hợp nhóm: IntelliJ IDEA hỗ trợ Git, SVN, Maven, Gradle và kết nối CI/CD với TeamCity, giúp làm việc nhóm hiệu quả hơn, quản lý mã nguồn chặt chẽ và đẩy nhanh quy trình phát triển phần mềm.

2.2.1.4. Hướng dẫn download IntelliJ IDEA

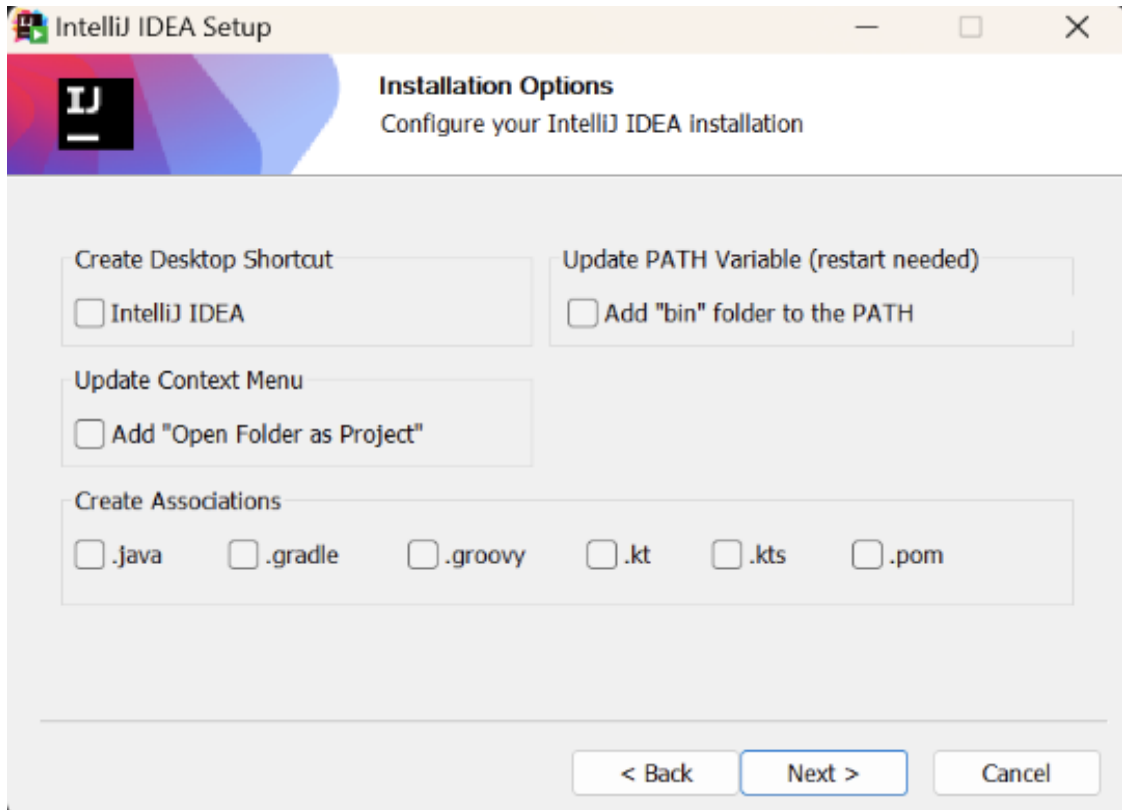
- Bước 1: Truy cập vào trang web chính thức của IntelliJ IDEA tại địa chỉ <https://www.jetbrains.com/idea/download>.
- Bước 2: Tại trang download, bạn sẽ thấy các phiên bản của IntelliJ IDEA được liệt kê. Chọn phiên bản phù hợp và sau đó ấn download. Sẽ có 2 phiên bản:
 - **Ultimate** (trả phí – hỗ trợ Spring Boot, Database...)
 - **Community** (miễn phí – đủ dùng cho Java cơ bản)



Hình 2.4. Trang download IntelliJ IDEA

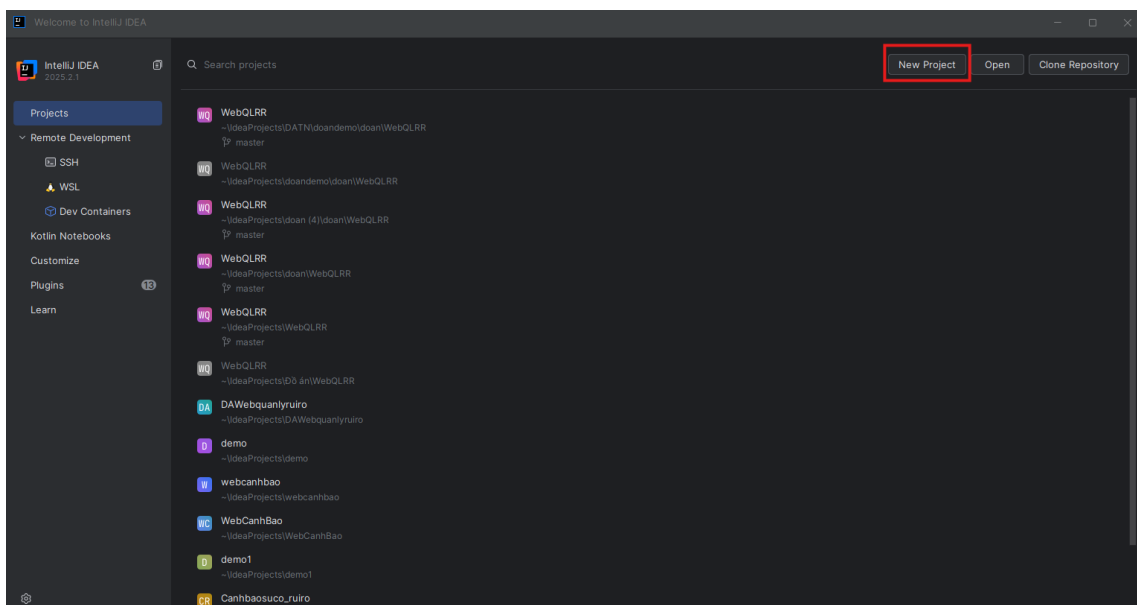
- Bước 3: Sau khi tải xuống hoàn tất, nhấn đúp vào file .exe (Windows) hoặc file .dmg (Mac) để bắt đầu quá trình cài đặt.
- Bước 4: Tiến hành chọn folder để cài đặt IntelliJ IDEA trên máy tính của bạn.

- Bước 5: Liên kết các phần mở rộng tệp cụ thể với IntelliJ IDEA để mở chúng bằng một cú nhấp đúp.



Hình 2.5. Cấu hình IntelliJ IDEA

- Bước 6: Hoàn thành cài đặt và cấu hình, bạn có thể bắt đầu sử dụng IntelliJ IDEA để lập trình.



Hình 2.6. Giao diện khi hoàn thành

2.2.2. Visual Studio Code

2.2.2.1. Giới thiệu về Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn rất phổ biến nhờ sự nhanh, nhẹ, hỗ trợ đa nền tảng và hoàn toàn mã nguồn mở. Với nhiều tính năng mạnh mẽ và hệ sinh thái plugin phong phú, VS Code ngày càng được các lập trình viên ưa chuộng và trở thành lựa chọn hàng đầu trong lập trình.



Hình 2.7. Logo VS Code

2.2.2.2. Ưu điểm nổi bật của Visual Studio Code

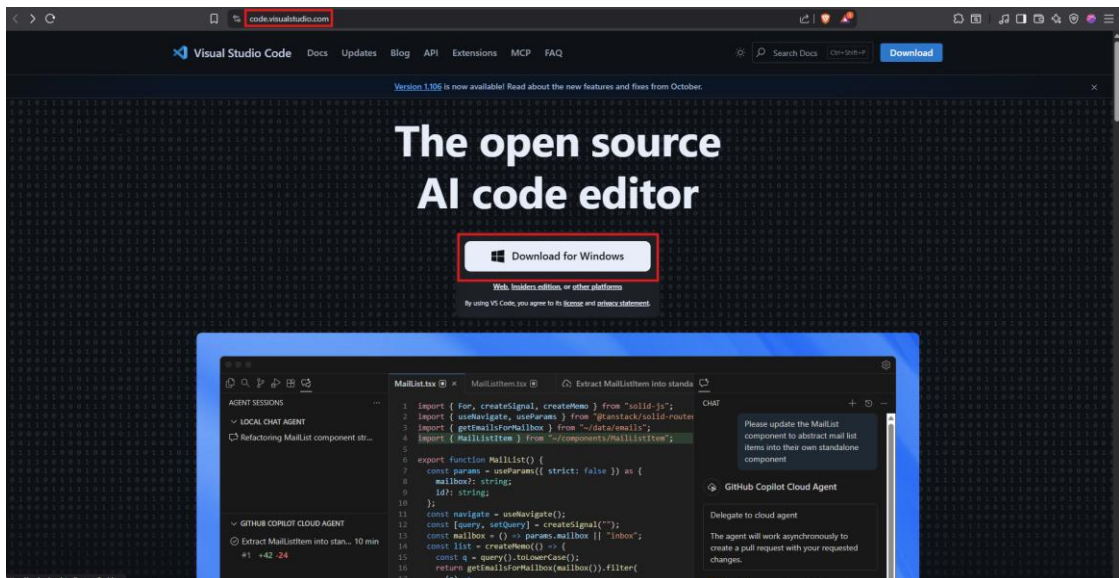
- Miễn phí và mã nguồn mở: VS Code là công cụ hoàn toàn miễn phí và mã nguồn mở theo giấy phép MIT. Người dùng cũng có thể đóng góp vào dự án bằng cách sửa lỗi, bổ sung tính năng hoặc dịch thuật. Chính tính chất mở và miễn phí này đã thu hút đông đảo cộng đồng lập trình viên và giúp VS Code phát triển mạnh mẽ.
- Đa nền tảng: Hỗ trợ đa nền tảng trên Windows, macOS và Linux, giúp lập trình viên có thể sử dụng cùng một công cụ trên nhiều thiết bị và hệ điều hành khác nhau. Nhờ đó, việc giữ nguyên môi trường phát triển trở nên thuận tiện và nhất quán hơn, không cần thay đổi công cụ khi chuyển máy.
- Giao diện thân thiện: Sở hữu giao diện đơn giản, trực quan và dễ sử dụng, phù hợp cho cả người mới học lập trình lẫn các nhà phát triển giàu kinh nghiệm, giúp việc làm quen và thao tác trở nên nhanh chóng và thuận tiện.
- Hỗ trợ nhiều ngôn ngữ lập trình: Hỗ trợ hơn 30 ngôn ngữ lập trình phổ biến như JavaScript, Python, Java, C++, C#, Go, PHP..., giúp bạn có thể dùng nó cho hầu hết mọi loại dự án. Mỗi ngôn ngữ đều được cung cấp cú pháp, gợi ý mã thông minh và tính năng gỡ lỗi đầy đủ, giúp lập trình hiệu quả hơn. Ví dụ, với JavaScript, VS Code hỗ trợ IntelliSense, debug tích hợp và cả TypeScript, mang lại trải nghiệm phát triển mạnh mẽ.
- Đa dạng Extension hỗ trợ coder: VS Code sở hữu hệ thống extension rất phong phú, cho phép người dùng bổ sung tính năng và tùy chỉnh trình soạn thảo theo nhu

cầu. Nhờ kho extension đa dạng, lập trình viên có thể dễ dàng mở rộng VS Code để hỗ trợ nhiều ngôn ngữ, công cụ và quy trình khác nhau, đặc biệt hữu ích cho cả người mới bắt đầu lẫn lập trình viên chuyên nghiệp.

- Hỗ trợ gỡ lỗi hiệu quả: cung cấp công cụ gỡ lỗi mạnh mẽ và dễ dùng, hỗ trợ nhiều ngôn ngữ như Node.js, Python, C++, Java. Người dùng có thể đặt breakpoint, xem giá trị biến, điều khiển luồng chạy và sử dụng console tích hợp để kiểm tra lỗi hiệu quả. So với Sublime Text, VS Code vượt trội về khả năng gỡ lỗi, extension và hỗ trợ Git. So với Atom, VS Code ổn định và hiệu năng tốt hơn. So với IntelliJ IDEA, dù IntelliJ mạnh về Java và hệ sinh thái JVM, VS Code vẫn đa năng hơn nhờ hỗ trợ rộng rãi nhiều ngôn ngữ lập trình khác nhau.
- Kết nối với các dịch vụ phổ biến: Tích hợp mạnh mẽ với các dịch vụ phổ biến như Git, Azure, GitHub và Docker, giúp bạn dễ dàng quản lý mã nguồn, triển khai ứng dụng và làm việc nhóm. Nhờ khả năng kết nối này, lập trình viên có thể thao tác với Git, làm việc với container Docker hoặc triển khai lên Azure ngay trong VS Code, giúp tối ưu quy trình làm việc và tập trung hơn vào việc viết mã và phát triển sản phẩm.

2.2.2.3. Hướng dẫn download Visual Studio Code

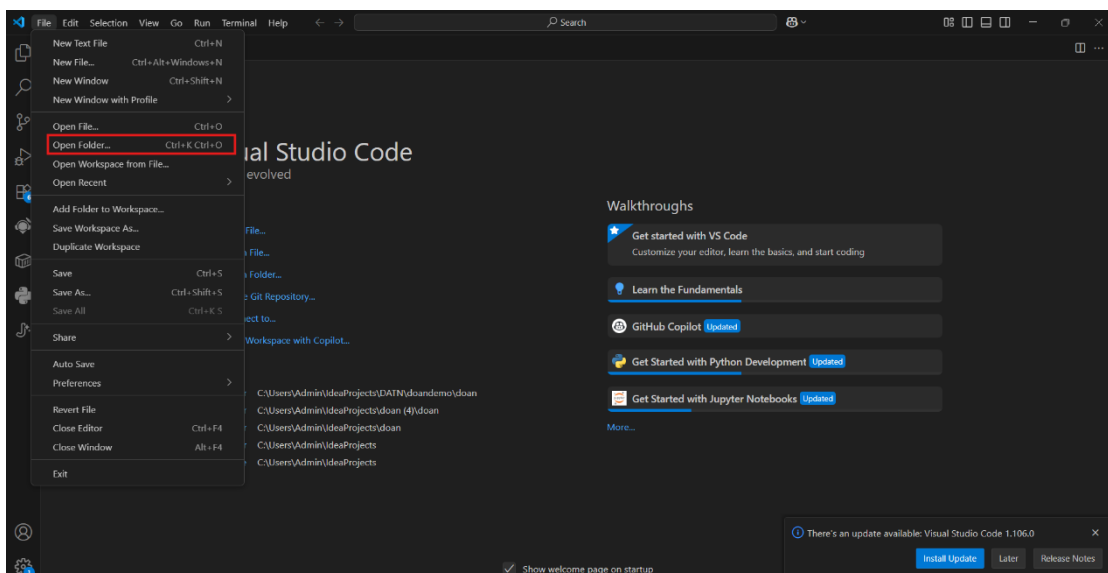
- Bước 1: Truy cập vào trang web chính thức của Visual Studio Code tại địa chỉ: <https://code.visualstudio.com>. Nhấn nút download và chọn phiên bản phù hợp với hệ điều hành của bạn (Windows, macOS, hoặc Linux).



Hình 2.8. Giao diện trang download

- Bước 2: Sau khi tải xuống, hãy làm theo các bước cài đặt ở bên dưới:

- Bấm vào và chạy tệp cài đặt (VSCodeUserSetup-{version}.exe) mà bạn đã tải về trên máy.
 - Bấm "Next" để bắt đầu quá trình cài đặt VsCode.
 - Đọc và chấp nhận các điều khoản sử dụng, sau đó nhấp "Next" để tiếp tục quá trình cài đặt.
 - Chọn thư mục cài đặt hoặc giữ nguyên đường dẫn mặc định, rồi bấm "Next" để tiếp tục.
 - Tùy chọn cài đặt thêm:
 - Thêm Visual Studio Code vào PATH (giúp bạn có thể mở VS Code từ dòng lệnh).
 - Tạo biểu tượng trên màn hình Desktop.
 - Tích hợp các tùy chọn vào menu ngữ cảnh của Windows Explorer.
 - Nhấn "Install" để bắt đầu quá trình tải.
 - Sau khi quá trình cài đặt hoàn tất, nhấp vào "Finish" để khởi động VS Code.
- Bước 3: Tạo dự án mới
- Mở Visual Studio Code và chọn File > Open Folder để mở thư mục dự án.
 - Nhấp chuột phải vào thanh bên trái (Explorer) và chọn New File để tạo tệp mới.



Hình 2.9. Tạo dự án mới

2.2.3. PostgreSQL

2.2.3.1. Tìm hiểu về PostgreSQL

PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ – hướng đối tượng (ORDBMS) miễn phí và mã nguồn mở, nổi tiếng là một trong những hệ thống tiên tiến và mạnh mẽ nhất hiện nay. Nó có khả năng mở rộng cao, tuân thủ nhiều tiêu chuẩn kỹ thuật và được thiết kế để xử lý khối lượng công việc đa dạng, từ ứng dụng trên máy cá nhân đến hệ thống kho dữ liệu và dịch vụ web với lượng người dùng lớn truy cập đồng thời.



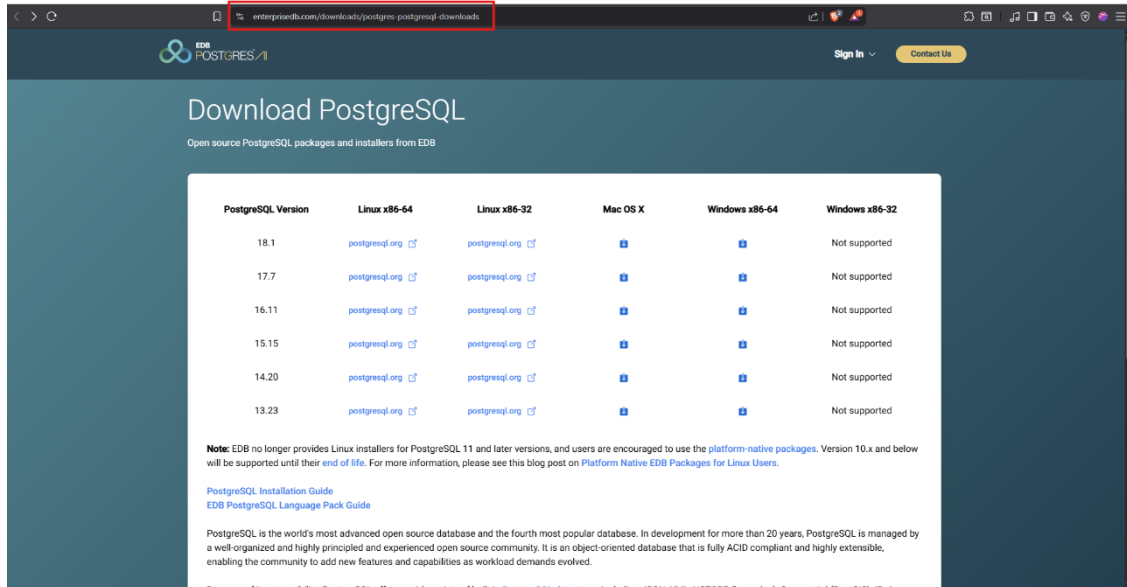
Hình 2.10. Logo PostgreSQL

2.2.3.2. Tính năng nổi bật của PostgreSQL

- Tính năng kiểm soát đồng thời nhiều phiên bản (MVCC): PostgreSQL đã triển khai tính năng MVCC trước cả Oracle. MVCC cho phép các phiên bản snapshot riêng biệt, giúp đạt được tính nhất quán và đồng thời trong quá trình truy cập cơ sở dữ liệu.
- Hỗ trợ đối tượng trong cơ sở dữ liệu: PostgreSQL là hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng, cho phép mở rộng và tùy chỉnh các tính năng bằng cách sử dụng ngôn ngữ chương trình như C/C++, Java, và nhiều ngôn ngữ khác.
- Sự hỗ trợ từ cộng đồng: PostgreSQL có một cộng đồng nhiệt tình, sẵn sàng giúp đỡ. Người dùng có thể tìm thấy câu trả lời cho các vấn đề liên quan đến PostgreSQL từ cộng đồng. Ngoài ra, có nhiều công ty cung cấp dịch vụ hỗ trợ chuyên nghiệp cho PostgreSQL khi cần thiết.

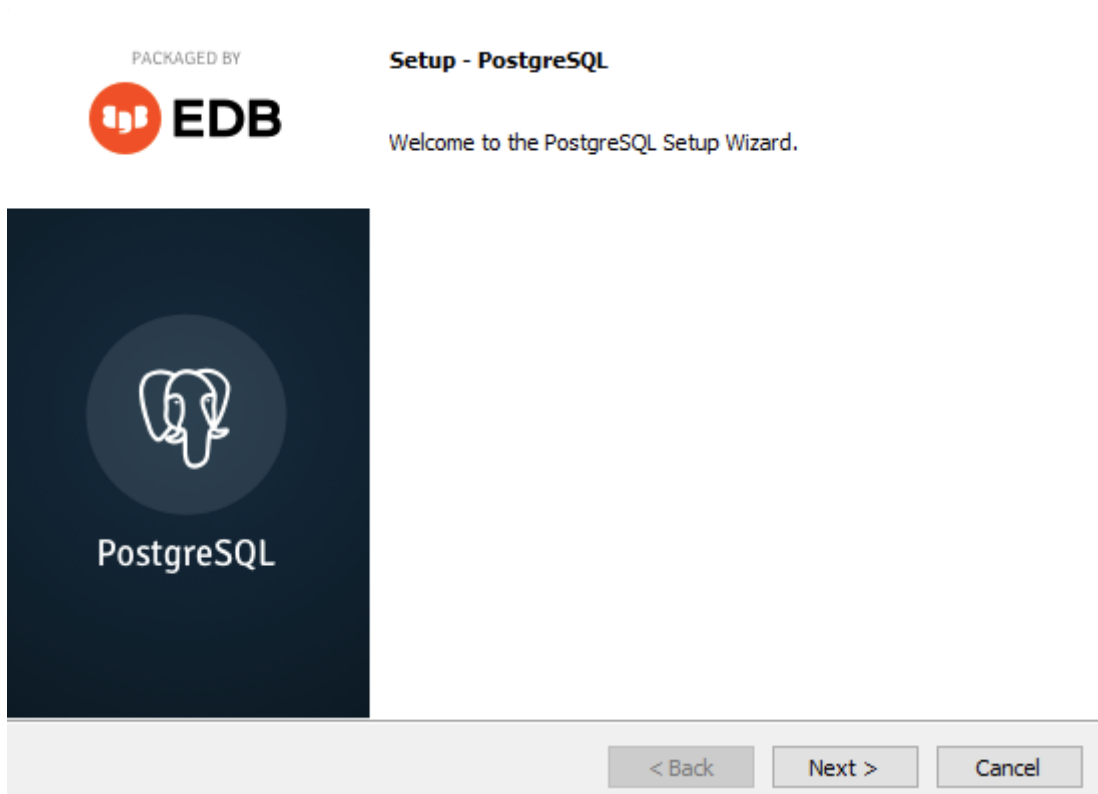
2.2.3.3. Hướng dẫn download PostgreSQL

- Bước 1: truy cập vào địa chỉ <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> và tải xuống phiên bản mới nhất tương thích với hệ điều hành của bạn.



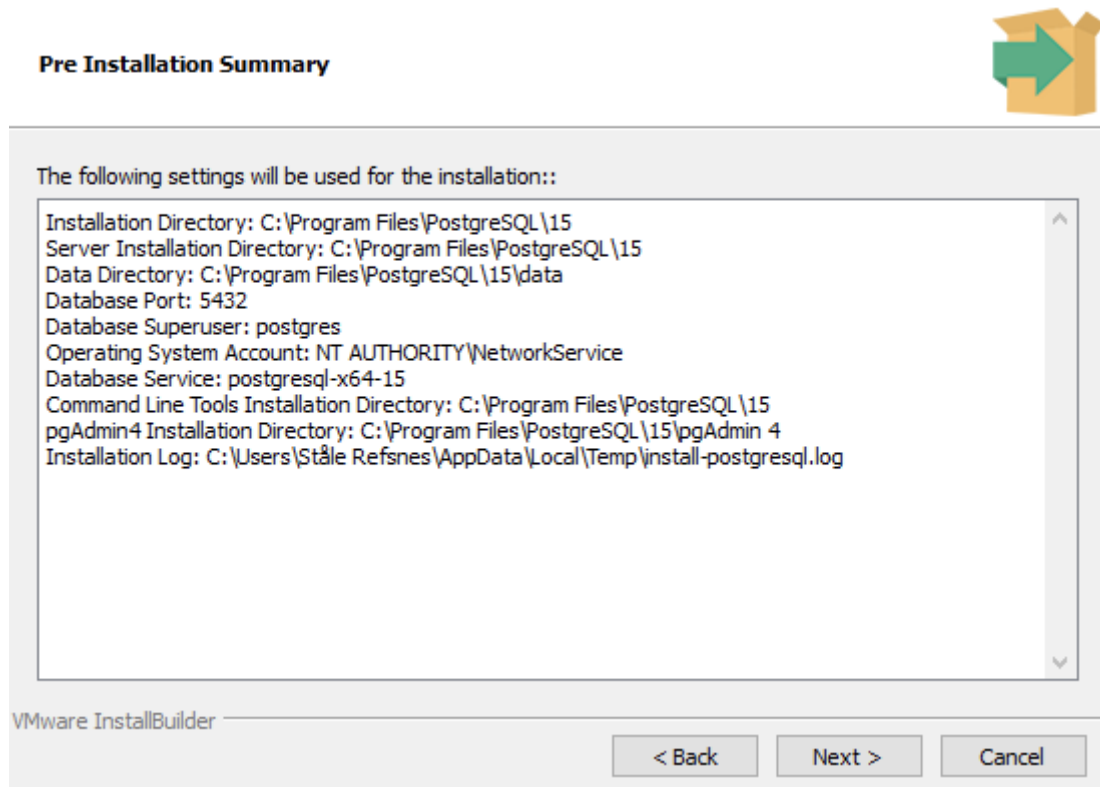
Hình 2.11. Giao diện download

- Bước 2: Chạy trình cài đặt mở file .exe đã tải xuống



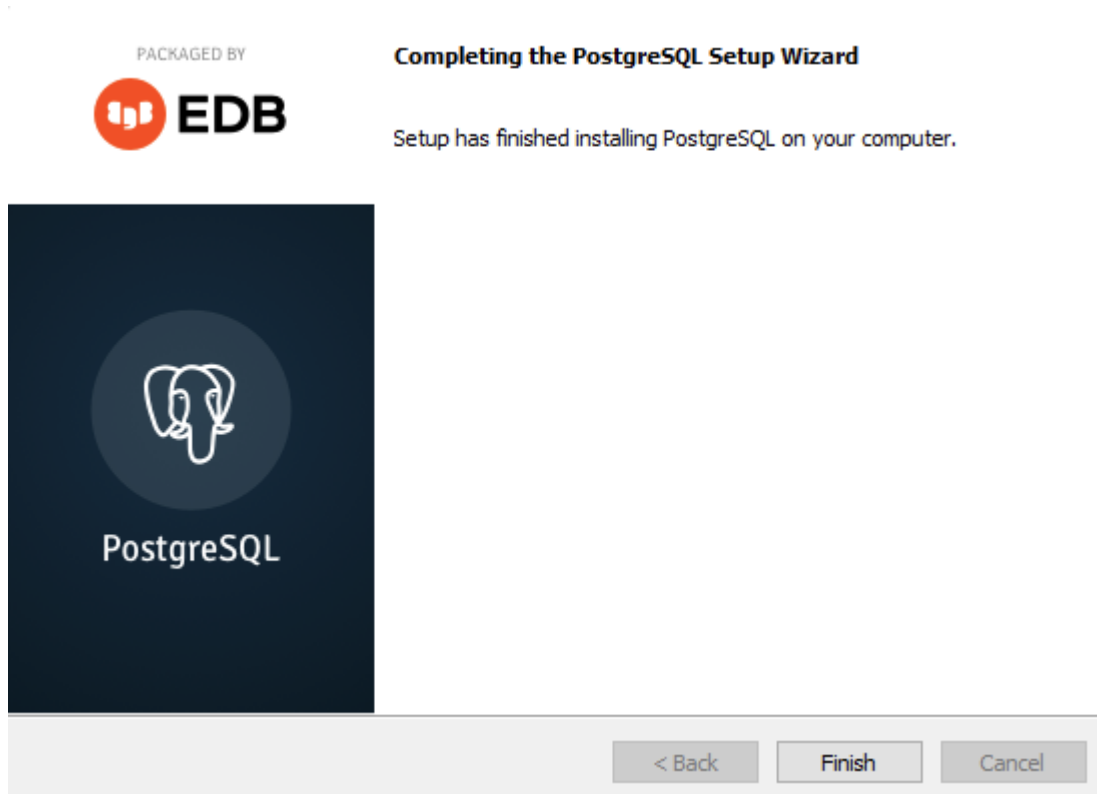
Hình 2.12. Chạy file vừa download

- Bước 3: Cấu hình cài đặt:
 - Chọn thư mục cài đặt và các thành phần (PostgreSQL Server, pgAdmin 4).
 - Đặt mật khẩu cho user mặc định “postgresql” (rất quan trọng).
 - Giữ cổng mặc định 5432 hoặc thay đổi nếu cần.



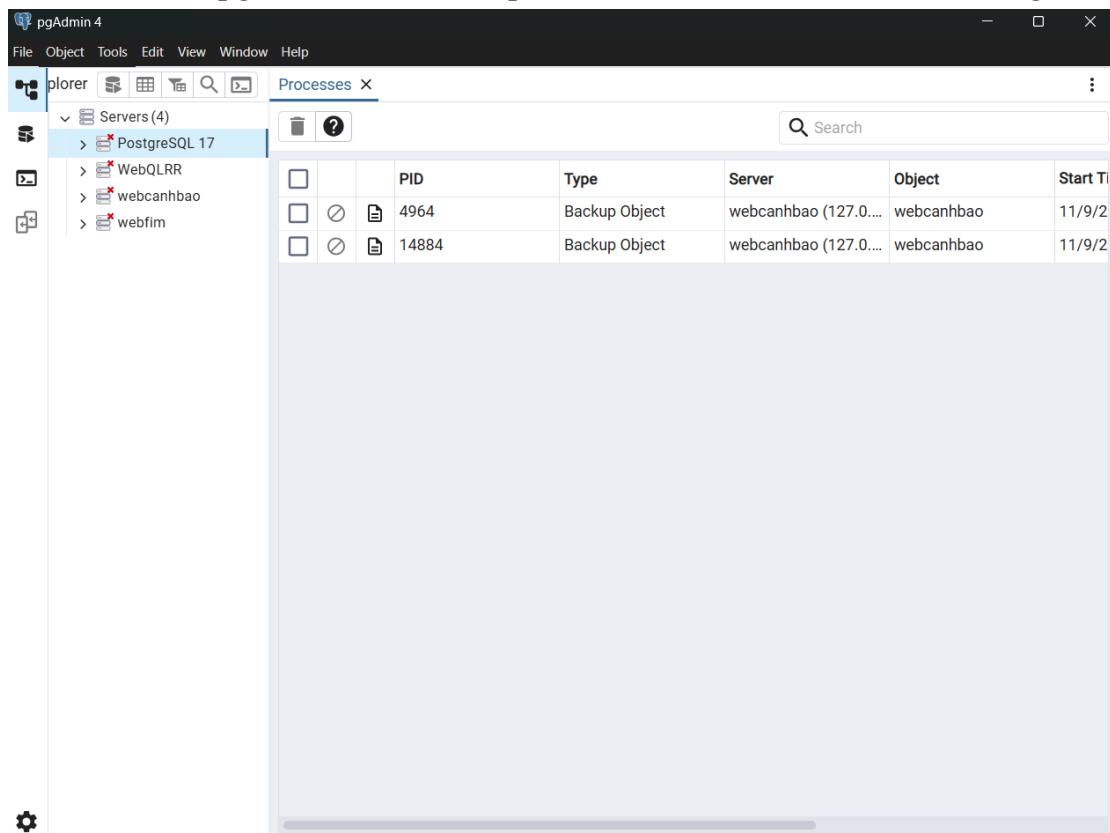
Hình 2.13. Cài đặt cấu hình

- Bước 4: Bắt đầu cài đặt
- Bước 5: Hoàn thành



Hình 2.15. Cài đặt thành công

- Bước 6: Mở pgAdmin4 và nhập mật khẩu đã tạo để kết nối vào PostgreSQL.



Hình 2.14. Giao diện kết nối cơ sở dữ liệu

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN

3.1. Mô tả và phân tích bài toán

3.1.1. Bối cảnh bài toán

Trong hoạt động của các doanh nghiệp, các sự cố như hỏng hóc máy móc, chập điện, cháy nổ, tai nạn lao động hoặc sự cố kho bãi xảy ra thường xuyên và gây ảnh hưởng lớn đến an toàn và hiệu suất sản xuất. Việc báo cáo và xử lý sự cố trong nhiều doanh nghiệp hiện nay vẫn còn mang tính thủ công, phụ thuộc vào điện thoại, giấy tờ hoặc ghi nhận nội bộ rời rạc, dẫn đến tình trạng:

- Phát hiện sự cố chậm
- Thông tin truyền đạt không đồng bộ
- Bộ phận xử lý không nhận thông tin kịp thời
- Khó theo dõi tiến độ xử lý
- Không có hệ thống dữ liệu để đánh giá rủi ro lâu dài

Điều này tạo ra nhu cầu cấp thiết về một hệ thống quản lý rủi ro và cảnh báo sự cố tự động – realtime – minh bạch – dễ theo dõi.

3.1.2. Mục tiêu của bài toán

Hệ thống cần giải quyết các mục tiêu sau:

- Tiếp nhận và ghi nhận sự cố nhanh chóng
 - Nhân viên có thể gửi báo cáo sự cố ngay lập tức thông qua giao diện web.
- Gửi cảnh báo realtime tới quản lý và bộ phận liên quan
 - Khi có sự cố, quản lý nhận thông báo thời gian thực (WebSocket).
- Quản lý phân công nhiệm vụ xử lý sự cố
 - Quản lý chỉ định bộ phận xử lý.
 - Bộ phận xử lý giao nhiệm vụ cho người xử lý trực tiếp.
- Theo dõi trạng thái xử lý sự cố
 - Từ khi phát sinh → đang xử lý → hoàn thành → đóng sự cố.
- Quản lý log xử lý và trao đổi thông tin realtime
 - Gửi bình luận, hình ảnh, file kèm theo → đồng bộ ngay lập tức.
- Cung cấp dashboard và thống kê

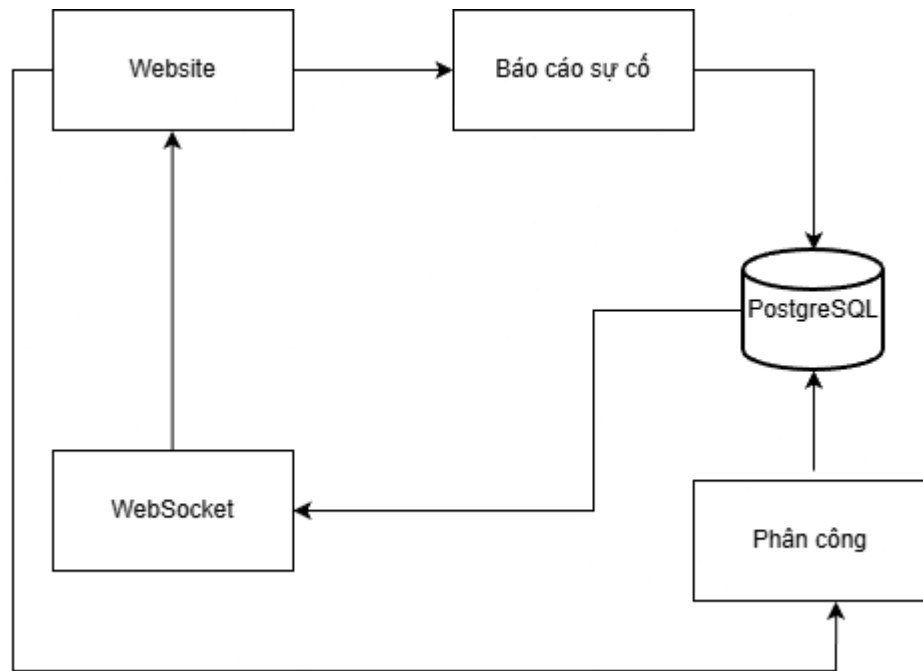
- Thống kê theo ngày, tháng, mức độ nghiêm trọng.
- Biểu đồ realtime phục vụ quản lý và lãnh đạo.

3.1.3. Các tác nhân trong hệ thống

Hệ thống có bốn tác nhân chính:

- Nhân viên báo cáo sự cố
 - Phát hiện và gửi báo cáo sự cố.
 - Cập nhật thông tin bổ sung (hình ảnh, mô tả).
 - Theo dõi tiến độ xử lý.
- Quản lý
 - Nhận cảnh báo realtime khi có sự cố mới.
 - Xem chi tiết sự cố → phân loại → phân công bộ phận xử lý.
 - Theo dõi toàn bộ tiến trình.
 - Xác nhận hoàn thành sự cố
- Bộ phận xử lý
 - Nhận nhiệm vụ được phân công từ quản lý.
 - Giao nhiệm vụ cho người xử lý trực tiếp.
 - Gửi log xử lý realtime.
- Người xử lý
 - Nhận nhiệm vụ xử lý trực tiếp.
 - Cập nhật tiến độ, tình trạng thực tế.
 - Yêu cầu hỗ trợ
 - Đính kèm minh chứng (ảnh, tài liệu).
 - Đánh dấu hoàn thành.

3.2. Phương pháp xử lý bài toán

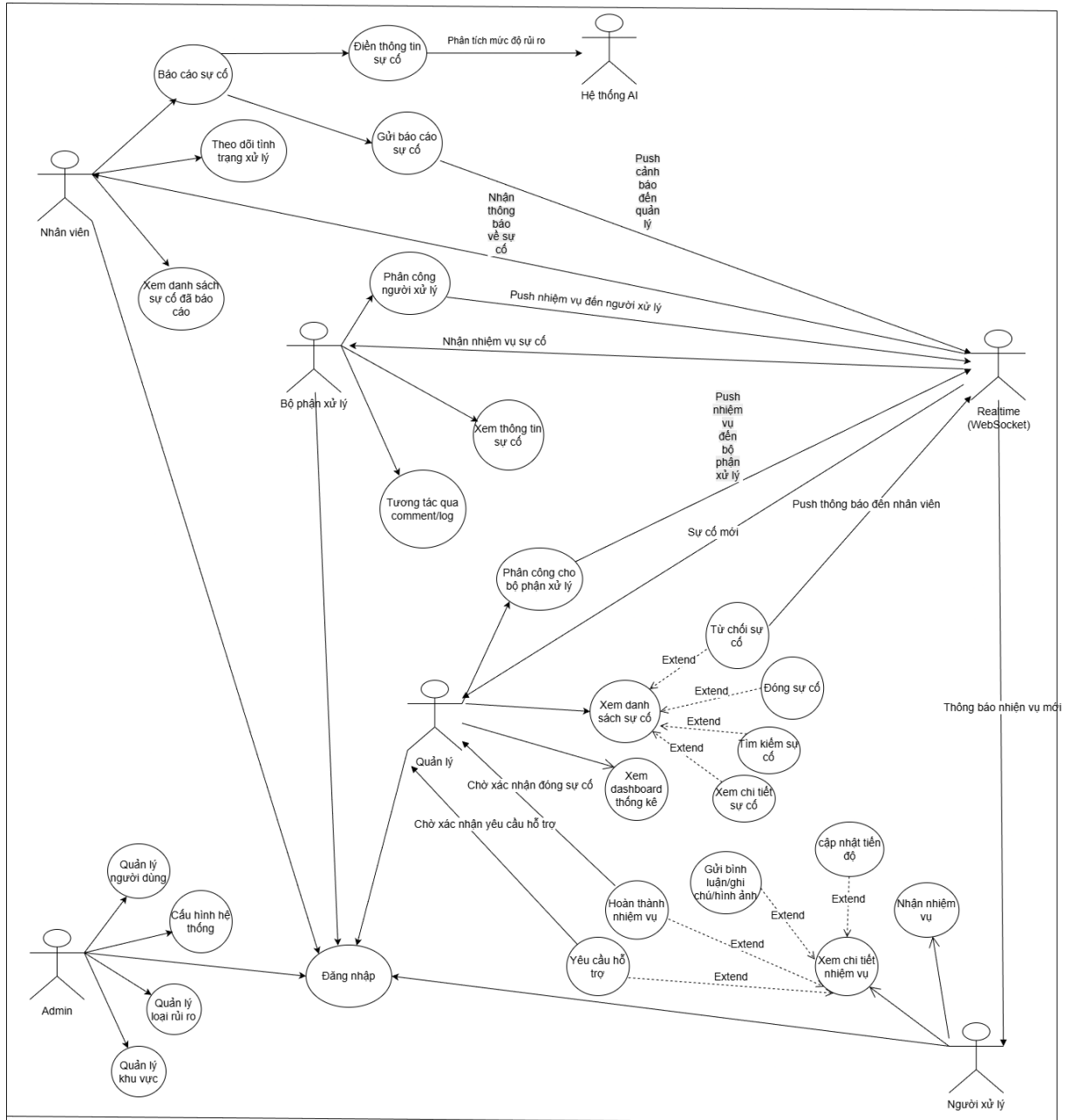


Hình 3.1. Luồng thực thi hệ thống

- Đăng nhập: Người dùng đăng nhập vào hệ thống. Mỗi tài khoản được cấp quyền sử dụng các chức năng khác nhau.
- Báo cáo sự cố: Nhân viên báo cáo sự cố nhập tiêu đề, mô tả và các thông tin liên quan. Hệ thống lưu trữ sự cố đồng thời phát sự kiện WebSocket để gửi thông báo realtime đến Quản lý.
- Phân công: Quản lý hoặc Bộ phận xử lý phân công sự cố. Hệ thống lưu phân công và phát sự kiện WebSocket để bộ phận xử lý hoặc người xử lý nhận thông báo realtime.
- Xử lý và hoàn thành sự cố: Người xử lý xem thông tin sự cố, tiến hành xử lý như thêm bình luận, hình ảnh về xử lý sự cố. Khi xử lý xong người xử lý nhấn hoàn thành nhiệm vụ. Hệ thống phát sự kiện WebSocket và quản lý nhận thông báo về sự cố hoàn thành.
- Thông báo: Hệ thống sử dụng WebSocket để gửi mọi thông báo realtime như Sự cố mới, được phân công, được giao nhiệm vụ, nhiệm vụ hoàn thành, đóng sự cố.

3.3. Biểu đồ Use Case

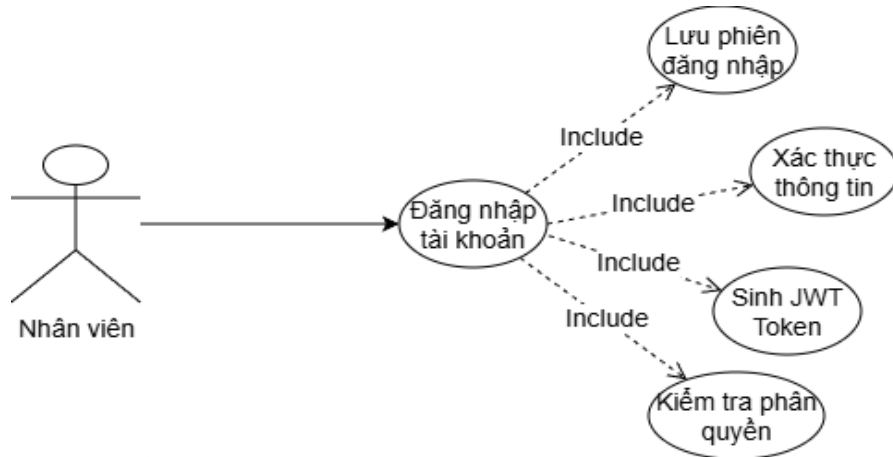
3.3.1. Use Case tổng quát



Hình 3.2. Usecase tổng quát

3.3.2. Mô tả usecase mức chi tiết

3.3.2.1. Usecase đăng nhập



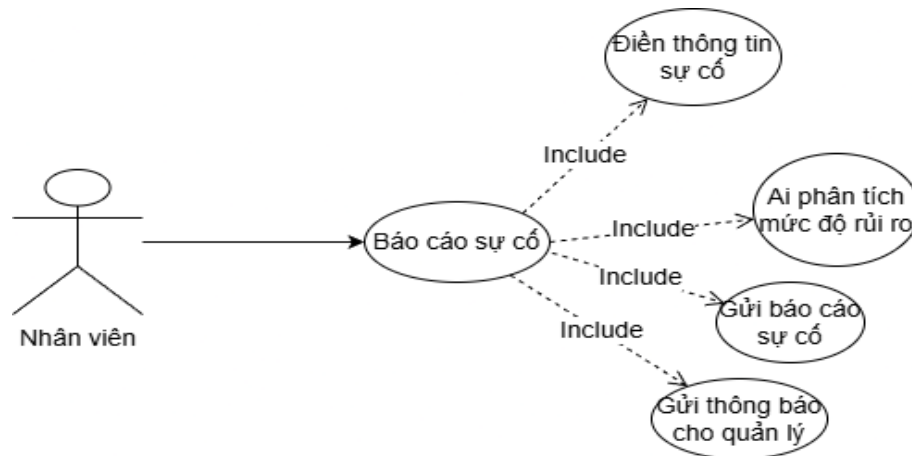
Hình 3.3. Usecase đăng nhập

Bảng 3.1. Đặc tả usecase đăng nhập

Tác nhân	Nhân viên, Quản lý, Bộ phận xử lý, người xử lý, Admin
Mô tả	Người dùng đăng nhập vào hệ thống để thực hiện các tác vụ.
Điều kiện trước	Người dùng phải có tài khoản của hệ thống được lưu trong database. Tài khoản được gán đúng vai trò
Luồng sự kiện chính	1. Hệ thống hiển thị form đăng nhập 2. Nhập thông tin lên form. 3. Người dùng nhấn đăng nhập 4. Hệ thống tiến hành kiểm tra thông tin. 5. Nếu hợp lệ: Hệ thống sinh JWT Token và gán Role vào Token. 6. Người dùng đăng nhập thành công và được điều hướng đến giao diện phù hợp theo vai trò.
Luồng sự kiện phụ	Nhập sai thông tin sẽ thông báo lỗi phần nhập sai và yêu cầu nhập lại. Nếu Token không hợp lệ sẽ yêu cầu đăng nhập lại

Điều kiện sau	Người dùng đăng nhập thành công sẽ được cấp JWT Access Token. Hệ thống cho phép truy cập các chức năng đúng theo role.
----------------------	--

3.3.2.2. Usecase báo cáo sự cố



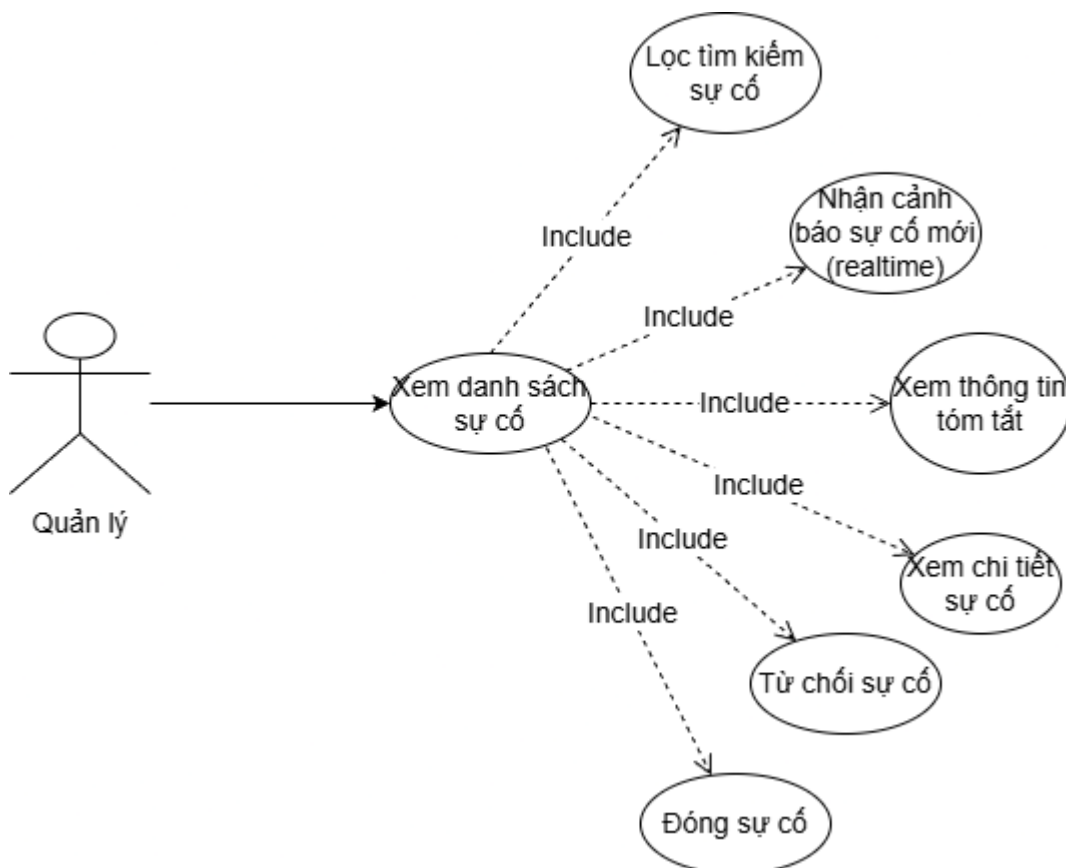
Hình 3.4. Usecase báo cáo sự cố

Bảng 3.2. Đặc tả usecase báo cáo sự cố

Tác nhân	Nhân viên
Mô tả	Nhân viên phát hiện và báo cáo sự cố.
Điều kiện trước	Nhân viên đã đăng nhập hệ thống. Nhân viên có quyền tạo sự cố
Luồng sự kiện chính	1. Nhân viên chọn chức năng báo cáo sự cố 2. Hệ thống hiển thị form để nhập thông tin sự cố. 3. Nhân viên nhập thông tin sự cố (tiêu đề, mô tả, mức độ rủi ro, loại sự cố, khu vực, hình ảnh nếu có) 4. Nhân viên chọn chức năng AI phân tích mức độ rủi ro dựa trên tiêu đề và mô tả. 5. Hệ thống gợi ý mức độ(Low/Medium/High). 6. Nhân viên kiểm tra thông tin và nhấn “Gửi báo cáo”. 7. Hệ thống lưu sự cố vào cơ sở dữ liệu. 8. WebSocket gửi cảnh báo sự cố mới đến quản lý.

	9. Hệ thống thông báo “Báo cáo sự cố thành công.
Luồng sự kiện phụ	Nếu nhân viên bỏ trống tiêu đề, mô tả hoặc khu vực thì hệ thống báo lỗi “Vui lòng nhập đầy đủ thông tin”. Nếu AI không thể phân tích mức độ rủi ro thì hệ thống sẽ báo lỗi “Vui lòng chọn mức độ thủ công”.
Điều kiện sau	Sự cố mới được lưu vào hệ thống. Quản lý nhận được cảnh báo realtime về sự cố mới.

3.3.2.3. Usecase xem danh sách sự cố

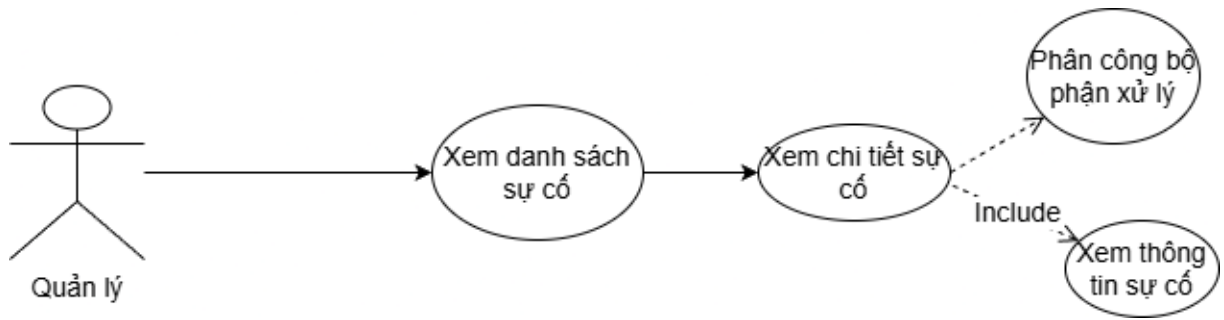


Hình 3.5. Usecase xem danh sách sự cố

Bảng 3.3. Đặc tả usecase xem danh sách sự cố

Tác nhân	Quản lý
Mô tả	Quản lý xem toàn bộ sự cố trong hệ thống, các chức năng trong danh sách sự cố và nhận cảnh báo khi có sự cố mới.
Điều kiện trước	Quản lý đã đăng nhập hệ thống. Có dữ liệu trong hệ thống.
Luồng sự kiện chính	1. Quản lý truy cập chức năng “Danh sách sự cố” 2. Hệ thống hiển thị danh sách sự cố bao gồm các thông tin như: Tiêu đề, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại, phân công. 3. Quản lý nhấn vào xem chi tiết sự cố khi có cảnh báo sự cố mới. 4. Quản lý có thể chọn tìm kiếm sự cố. 5. Quản lý có thể nhấn “Chi tiết” để xem thông tin đầy đủ. 6. Quản lý nhấn “Từ chối” ngay tại danh sách nếu sự cố không hợp lệ. 7. Quản lý nhấn đóng sự cố nếu đã hoàn thành.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị không có sự cố nào. Nếu quản lý tìm kiếm không có kết quả hệ thống sẽ báo không tìm thấy sự cố.
Điều kiện sau	Quản lý xem được danh sách sự cố. Có thể thực hiện các hành động khác: Xem chi tiết, từ chối, lọc, tìm kiếm và đóng sự cố.

3.3.2.4. Usecase xem chi tiết sự cố

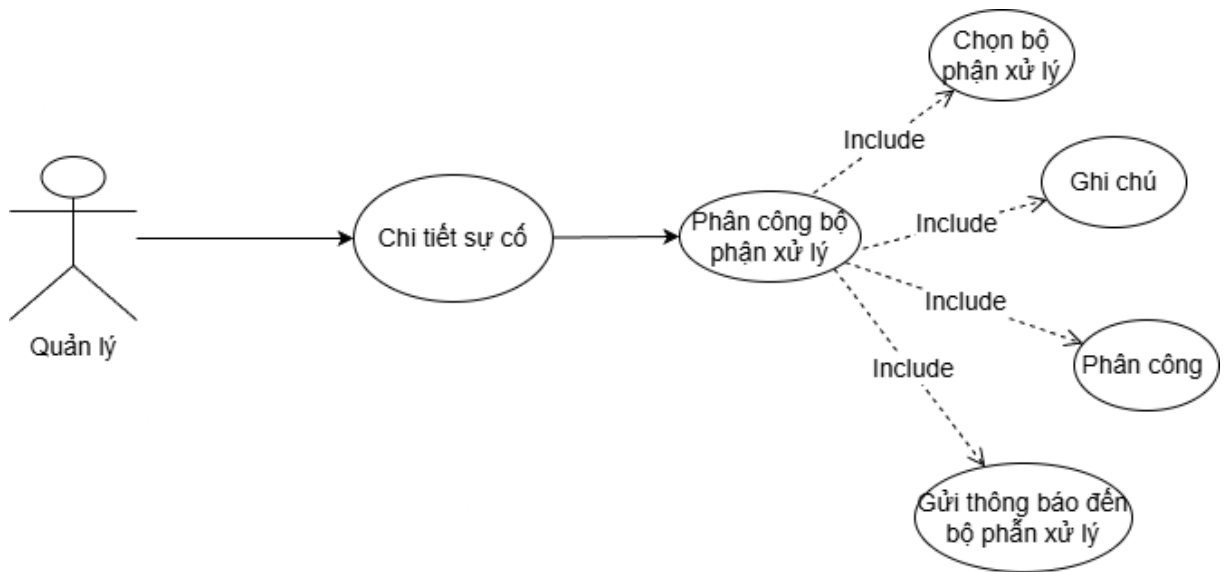


Hình 3.6. Usecase xem chi tiết sự cố

Bảng 3.4. Đặc tả usecase xem chi tiết sự cố

Tác nhân	Quản lý
Mô tả	Quản lý xem chi tiết về sự cố bao gồm tiêu đề, mô tả, khu vực, mức độ, hình ảnh, trạng thái, lịch sử xử lý và chức năng phân công.
Điều kiện trước	Quản lý đã đăng nhập hệ thống. Sự cố tồn tại trong hệ thống. Quản lý mở chức năng “Xem danh sách sự cố”.
Luồng sự kiện chính	1. Quản lý truy cập chức năng “Xem chi tiết” 2. Hệ thống hiển thị các thông tin về sự cố như: Tiêu đề, mô tả, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại. 3. Quản lý có thể chọn chức năng “phân công bộ phận xử lý”.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị sự cố không tồn tại.
Điều kiện sau	Quản lý xem chi tiết sự cố. Có thể thực hiện các hành động khác như phân công.

3.3.2.5. Usecase phân công bộ phận xử lý



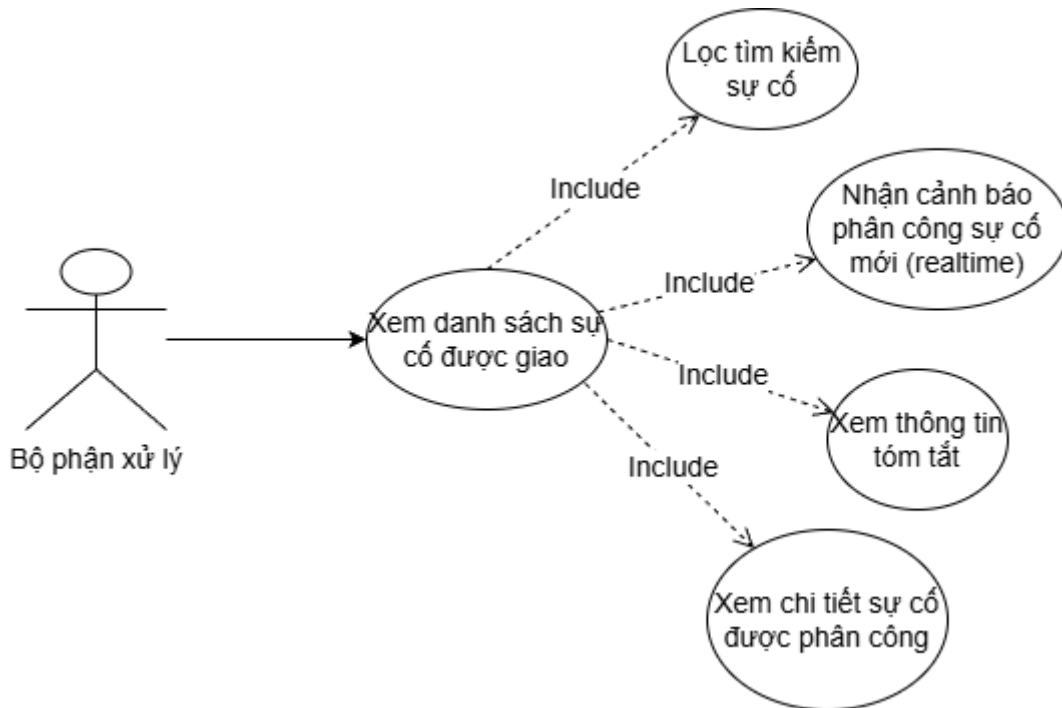
Hình 3.7. Usecase phân công bộ phận xử lý

Bảng 3.5. Đặc tả usecase phân công bộ phận xử lý

Tác nhân	Quản lý
Mô tả	Quản lý phân công sự cố đến bộ phận xử lý. Hệ thống lưu lại phân công và gửi thông báo realtime đến bộ phận xử lý.
Điều kiện trước	Quản lý đã đăng nhập hệ thống. Sự cố tồn tại trong hệ thống. Quản lý mở chức năng “Xem chi tiết sự cố”.
Luồng sự kiện chính	1. Quản lý truy cập chức năng “Xem chi tiết” 2. Hệ thống hiển thị các thông tin về sự cố như: Tiêu đề, mô tả, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại. 3. Quản lý chọn phân công cho bộ phận xử lý phù hợp từ danh sách kèm ghi chú. 4. Quản lý xác nhận phân công. 5. Hệ thống lưu phân công sự cố vào cơ sở dữ liệu. 6. WebSocket gửi cảnh báo sự cố mới đến bộ phận xử lý liên quan.

	7. Hệ thống thông báo “Phân công sự cố thành công.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị sự cố không tồn tại. Không chọn bộ phận xử lý: Hệ thống yêu cầu: “Vui lòng chọn bộ phận xử lý trước khi phân công.”
Điều kiện sau	Sự cố mới phân công thành công và được lưu vào hệ thống. Bộ phận xử lý nhận được cảnh báo realtime về sự cố mới.

3.3.2.6. Usecase xem danh sách sự cố của bộ phận xử lý



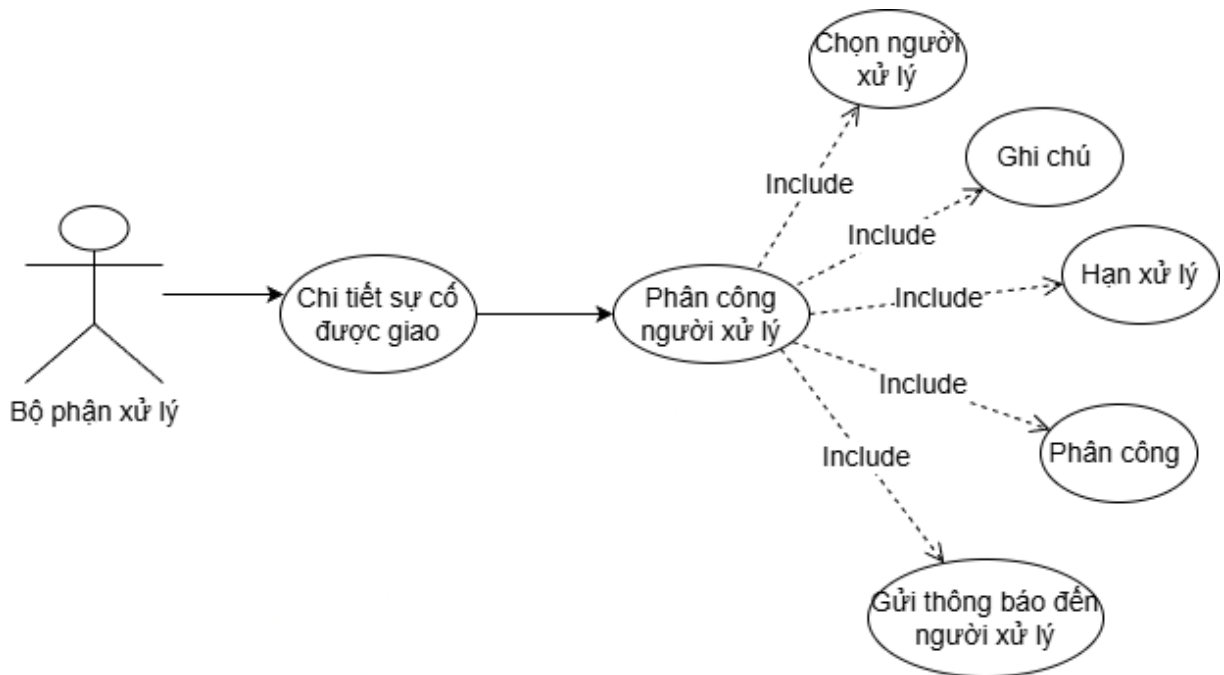
Hình 3.8. Usecase xem danh sách sự cố của bộ phận xử lý

Bảng 3.6. Đặc tả usecase xem danh sách sự cố

Tác nhân	Bộ phận xử lý
Mô tả	Bộ phận xử lý xem toàn bộ sự cố được phân công cho bộ phận của mình. Hệ thống hỗ trợ tìm kiếm, sắp xếp, lọc và

	cập nhật realtime khi có sự cố mới được giao hoặc khi người xử lý không nhận nhiệm vụ đúng hạn.
Điều kiện trước	Bộ phận xử lý đã đăng nhập hệ thống. Có sự cố được phân công.
Luồng sự kiện chính	1. Bộ phận xử lý truy cập chức năng “Danh sách phân công” 2. Hệ thống hiển thị danh sách sự cố được phân công bao gồm các thông tin như: Tiêu đề, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại, bộ phận được giao. 3. Bộ phận xử lý nhấn vào xem chi tiết sự cố khi có cảnh báo phân công sự cố mới. 4. Bộ phận xử lý chọn một sự cố để xem chi tiết. 5. Bộ phận xử lý tìm kiếm hoặc sắp xếp danh sách theo nhu cầu.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị không có sự cố nào được giao cho bộ phận. Nếu Bộ phận xử lý tìm kiếm không có kết quả hệ thống sẽ báo không tìm thấy sự cố. Người xử lý không nhận nhiệm vụ : Hệ thống tự động đánh dấu sự cố màu đỏ và hiển thị cảnh báo cho bộ phận xử lý.
Điều kiện sau	Bộ phận xử lý xem được danh sách sự cố được giao. Danh sách luôn cập nhật realtime khi có sự cố mới hoặc sự cố bị chậm xử lý. Có thể thực hiện các chức năng tiếp theo như: xem chi tiết sự cố, phân công người xử lý, thêm bình luận, giám sát tiến độ.

3.3.2.7. Usecase phân công người xử lý



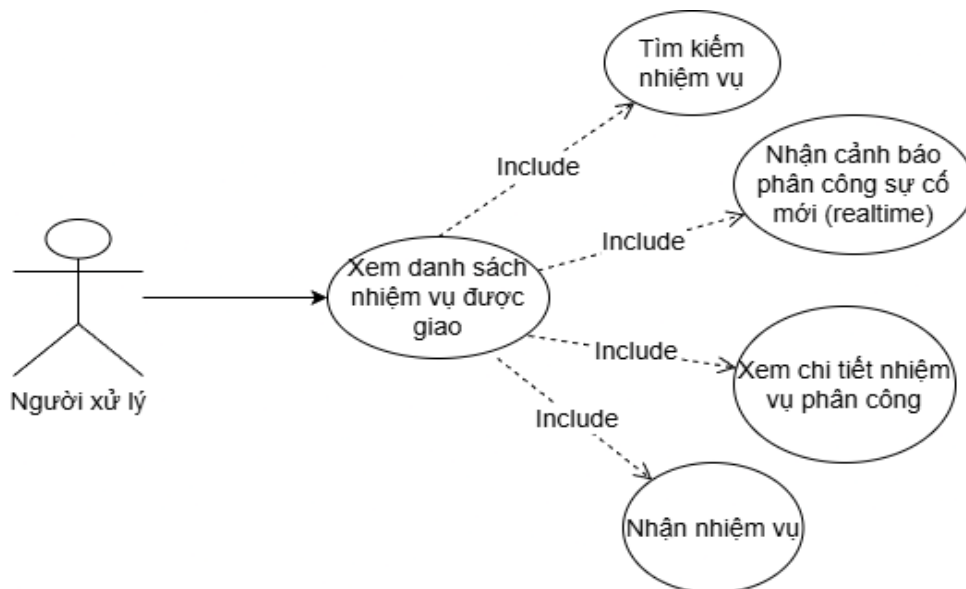
Hình 3.9. Usecase phân công người xử lý

Bảng 3.7. Đặc tả usecase phân công người xử lý

Tác nhân	Bộ phận xử lý
Mô tả	Bộ phận xử lý phân công nhiệm vụ đến người xử lý trực tiếp. Hệ thống lưu lại phân công và gửi thông báo realtime đến người xử lý để tiếp nhận nhiệm vụ.
Điều kiện trước	Bộ phận xử lý đã đăng nhập hệ thống. Sự cố đã được phân công đến bộ phận xử lý và tồn tại trong hệ thống. Bộ phận xử lý mở chức năng “Xem chi tiết sự cố”.
Luồng sự kiện chính	- Bộ phận xử lý truy cập chức năng “Xem chi tiết” - Hệ thống hiển thị các thông tin về sự cố như: Tiêu đề, mô tả, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại. - Bộ phận xử lý chọn phân công cho người xử lý phù hợp từ danh sách, ghi chú và hạn xử lý.

	4. Bộ phận xử lý xác nhận phân công. 5. Hệ thống lưu thông tin phân công vào cơ sở dữ liệu và cập nhật trạng thái sự cố.. 6. Hệ thống gửi thông báo realtime (WebSocket) đến người xử lý được giao nhiệm vụ. 7. Hệ thống thông báo “Phân công sự cố thành công.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị sự cố không tồn tại. Không chọn người xử lý: Hệ thống yêu cầu: “Vui lòng chọn người xử lý trước khi phân công.” Không chọn hạn xử lý: Hệ thống yêu cầu: “Vui lòng chọn hạn xử lý trước khi phân công.”
Điều kiện sau	Sự cố mới phân công thành công và được lưu vào hệ thống. Người xử lý nhận được cảnh báo realtime về sự cố mới.

3.3.2.8. Usecase danh sách nhiệm vụ được giao

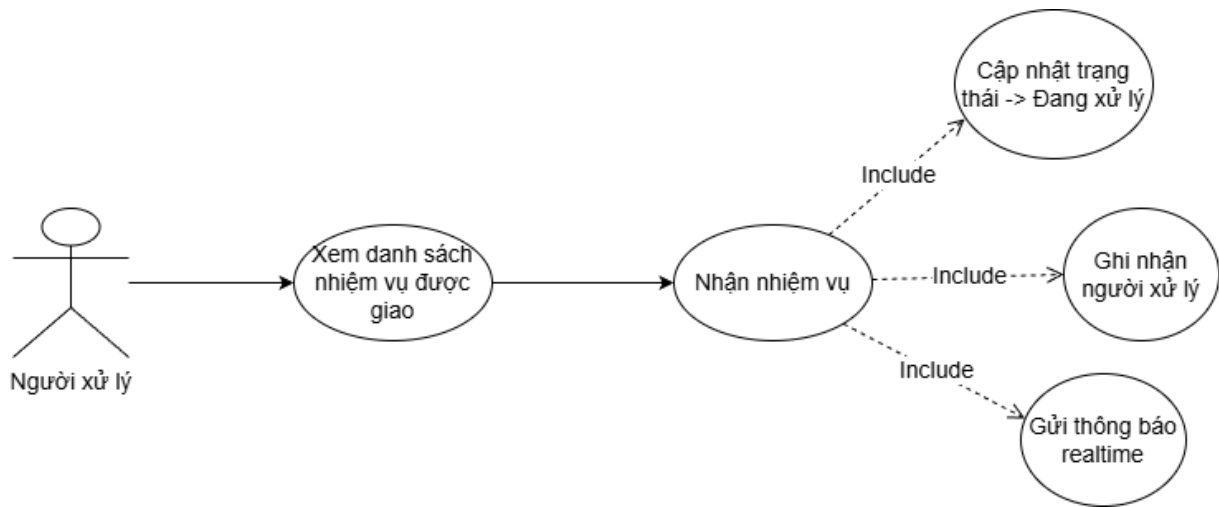


Hình 3.10. Usecase danh sách nhiệm vụ được giao

Bảng 3.8. Đặc tả usecase danh sách nhiệm vụ được giao

Tác nhân	Người xử lý
Mô tả	Người xử lý xem toàn bộ nhiệm vụ mà bộ phận xử lý đã phân công cho mình. Hệ thống hỗ trợ lọc, tìm kiếm, sắp xếp và cập nhật realtime khi có nhiệm vụ mới được giao hoặc nhiệm vụ thay đổi trạng thái.
Điều kiện trước	Người xử lý đã đăng nhập hệ thống. Có nhiệm vụ được phân công.
Luồng sự kiện chính	1. Người xử lý truy cập chức năng “Nhiệm vụ của tôi” 2. Hệ thống hiển thị danh sách nhiệm vụ được phân công bao gồm các thông tin như: Tiêu đề, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại. 3. Hệ thống tự động cập nhật realtime nếu có nhiệm vụ mới được phân công từ bộ phận xử lý. 4. Người xử lý chọn một nhiệm vụ để xem chi tiết. 5. Người xử lý tìm kiếm hoặc sắp xếp danh sách theo nhu cầu.
Luồng sự kiện phụ	Không có dữ liệu hệ thống sẽ hiển thị không có sự cố nào được giao cho người xử lý. Nếu người xử lý tìm kiếm không có kết quả hệ thống sẽ báo không tìm thấy sự cố.
Điều kiện sau	Người xử lý xem được danh sách nhiệm vụ được giao. Danh sách luôn cập nhật realtime khi có nhiệm vụ mới. Có thể thực hiện các chức năng tiếp theo như: xem chi tiết sự cố, nhận nhiệm vụ.

3.3.2.9. Usecase nhận nhiệm vụ



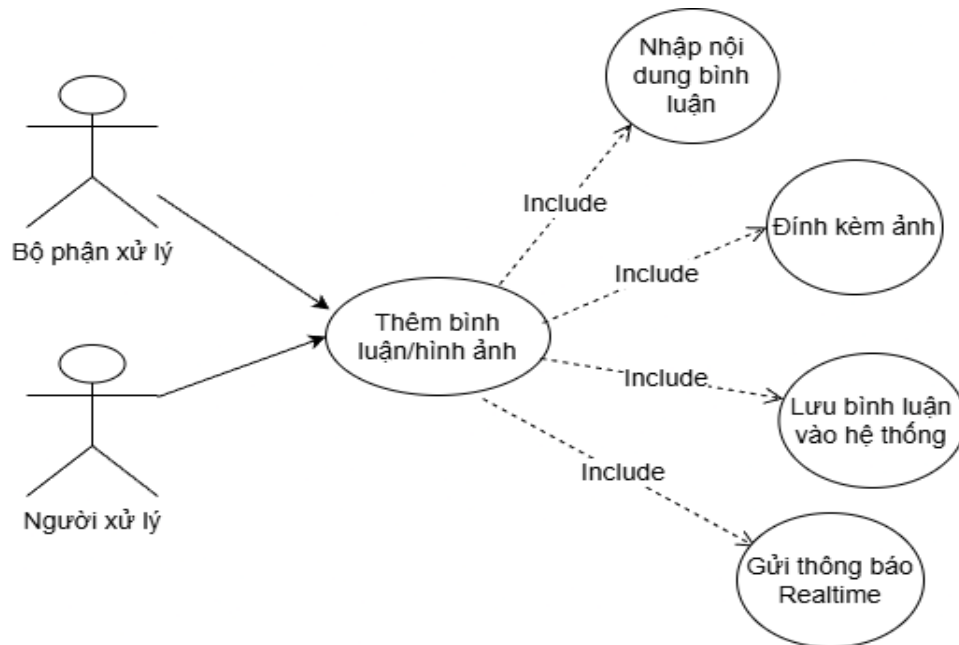
Hình 3.11. Usecase nhận nhiệm vụ

Bảng 3.9. Đặc tả usecase nhận nhiệm vụ

Tác nhân	Người xử lý
Mô tả	Người xử lý tiếp nhận nhiệm vụ được phân công từ bộ phận xử lý bằng cách nhấn nút “Nhận nhiệm vụ”. Khi thực hiện, hệ thống cập nhật trạng thái nhiệm vụ, ghi nhận người xử lý chính và gửi thông báo realtime đến các tác nhân liên quan.
Điều kiện trước	Người xử lý đã đăng nhập hệ thống. Nhiệm vụ đang ở trạng thái “Chờ nhận”. Người xử lý thấy nhiệm vụ trong danh sách nhiệm vụ của mình.
Luồng sự kiện chính	1. Người xử lý mở màn hình “Nhiệm vụ của tôi”. 2. Hệ thống hiển thị danh sách nhiệm vụ được phân công bao gồm các thông tin như: Tiêu đề, khu vực, loại sự cố, mức độ rủi ro, trạng thái hiện tại. 3. Người xử lý chọn một nhiệm vụ và nhấn “Nhận nhiệm vụ”. 4. Hệ thống xác nhận hành động. 5. Hệ thống cập nhật trạng thái nhiệm vụ thành “Đang xử lý” và lưu người xử lý chính.

	6. Hệ thống gửi thông báo realtime đến những người liên quan. 7. Hệ thống hiển thị thông báo: “Nhận nhiệm vụ thành công.”
Luồng sự kiện phụ	Nếu nhiệm vụ không tồn tại hoặc đã bị hủy → hệ thống thông báo lỗi.
Điều kiện sau	Trạng thái nhiệm vụ đổi sang “Đang xử lý”. Bộ phận xử lý, Quản lý và Nhân viên nhận được thông báo qua realtime. Nếu người xử lý tìm kiếm không có kết quả hệ thống sẽ báo không tìm thấy sự cố. Người xử lý có thể tiếp tục các chức năng khác: xem chi tiết nhiệm vụ.

3.3.2.10. Usecase thêm bình luận

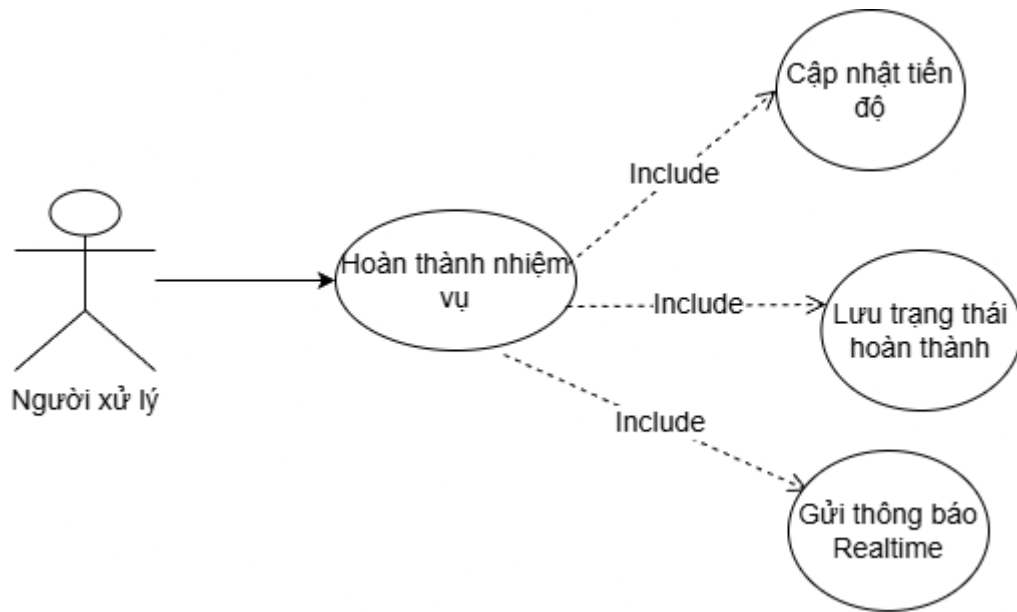


Hình 3.12. Usecase thêm bình luận

Bảng 3.10. Đặc tả usecase thêm bình luận

Tác nhân	Người xử lý, Bộ phận xử lý
Mô tả	Tác nhân thêm bình luận, ghi chú hoặc đính kèm hình ảnh trong quá trình xử lý sự cố. Bình luận được lưu vào hệ thống và gửi thông báo realtime đến các bên liên quan.
Điều kiện trước	Tác nhân đã đăng nhập hệ thống. Tác nhân đang ở màn hình “Xem chi tiết sự cố”. Sự cố còn hiệu lực và chưa đóng.
Luồng sự kiện chính	1. Tác nhân mở màn hình “Chi tiết nhiệm vụ”. 2. Tác nhân nhập nội dung bình luận hoặc hình ảnh. 3. Tác nhân nhấn nút Gửi bình luận. 4. Hệ thống kiểm tra dữ liệu hợp lệ. 5. Hệ thống lưu bình luận / hình ảnh vào cơ sở dữ liệu. 6. Hệ thống gửi thông báo realtime đến: quản lý, bộ phận xử lý, người xử lý liên quan. 7. Hệ thống hiển thị bình luận mới trong danh sách bình luận.
Luồng sự kiện phụ	Không nhập nội dung → báo lỗi: “Vui lòng nhập nội dung bình luận”. - Ảnh không hợp lệ (định dạng sai hoặc quá dung lượng) → hệ thống báo lỗi. - Sự cố đã đóng → hệ thống không cho phép bình luận.
Điều kiện sau	Bình luận / hình ảnh được lưu thành công. Tất cả người dùng liên quan nhận được thông báo realtime. Danh sách bình luận trong sự cố được cập nhật.

3.3.2.11. Usecase hoàn thành nhiệm vụ



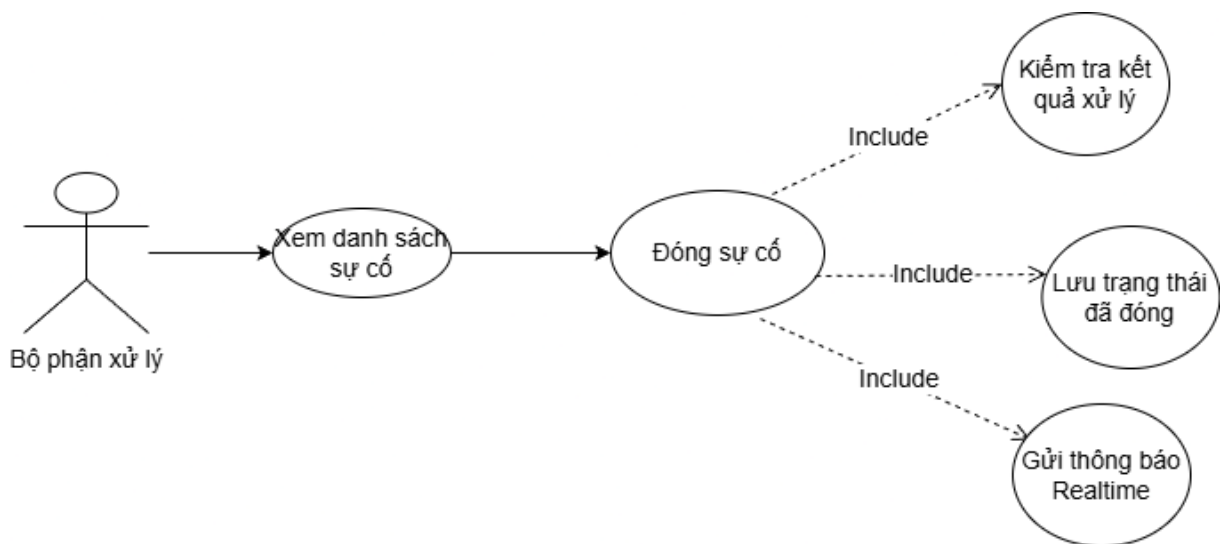
Hình 3.13. Usecase hoàn thành nhiệm vụ

Bảng 3.11. Đặc tả usecase hoàn thành nhiệm vụ

Tác nhân	Người xử lý
Mô tả	Người xử lý sau khi thực hiện công việc sẽ hoàn thành nhiệm vụ bằng cách cập nhật trạng thái, hoàn thành nhiệm vụ. Hệ thống lưu trạng thái và gửi thông báo realtime đến bộ phận xử lý và quản lý.
Điều kiện trước	Người xử lý đã đăng nhập hệ thống. Nhiệm vụ tồn tại và đang ở trạng thái "Đang xử lý". Người xử lý đang ở trang "Chi tiết nhiệm vụ".
Luồng sự kiện chính	1. Người xử lý mở màn hình "Chi tiết nhiệm vụ". 2. Người xử lý nhấn nút Hoàn thành nhiệm vụ. 3. Người xử lý xác nhận hoàn thành. 4. Hệ thống kiểm tra hợp lệ. 5. Hệ thống cập nhật trạng thái nhiệm vụ thành "Hoàn thành". 6. Hệ thống lưu nhật ký xử lý. 7. WebSocket gửi thông báo realtime đến quản lý và bộ phận xử lý.

	8. Hệ thống hiển thị thông báo “Hoàn thành nhiệm vụ thành công”.
Luồng sự kiện phụ	Nhiệm vụ đã bị hủy hoặc đóng → hệ thống không cho phép hoàn thành.
Điều kiện sau	Nhiệm vụ được cập nhật sang “Hoàn thành”. - Tất cả bên liên quan nhận được thông báo realtime: Quản lý, Bộ phận xử lý.

3.3.2.12. Usecase đóng sự cố



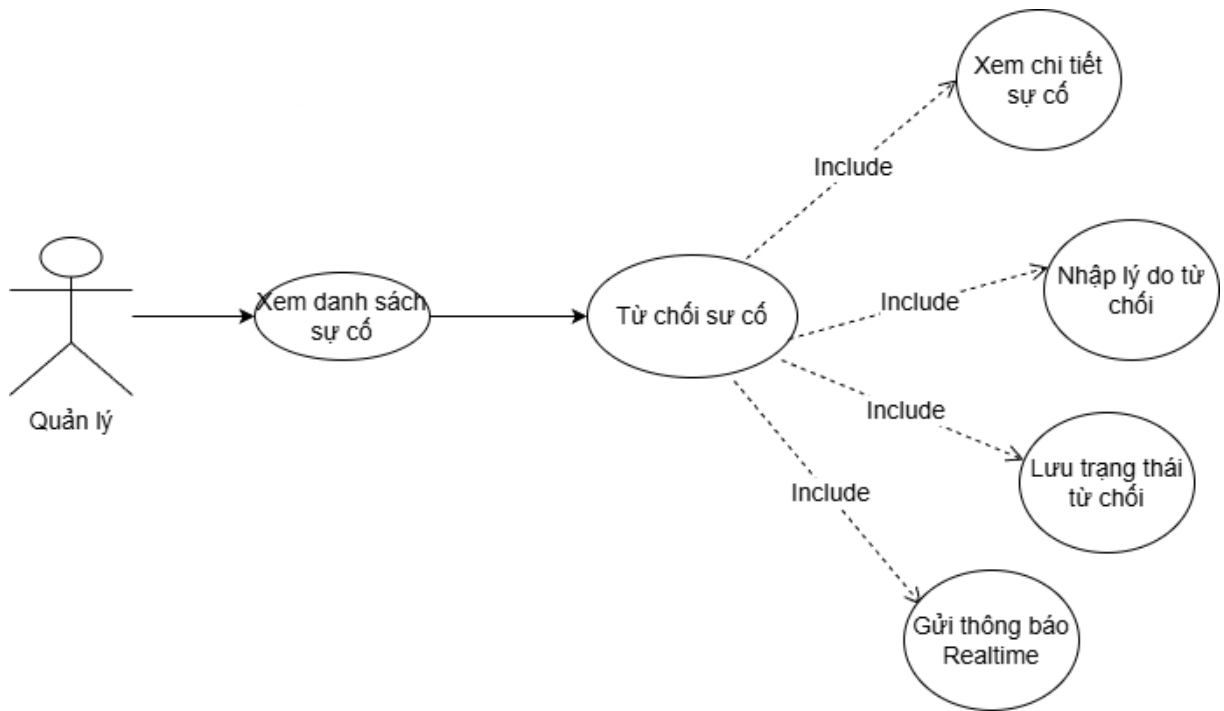
Hình 3.14. Usecase đóng sự cố

Bảng 3.12. Đặc tả usecase đóng sự cố

Tác nhân	Bộ phận xử lý
Mô tả	Bộ phận xử lý sau khi kiểm tra kết quả xử lý của người xử lý có thể đóng sự cố. Khi sự cố được đóng, hệ thống lưu trạng thái, ghi nhật ký và gửi thông báo realtime đến tất cả các tác nhân liên quan.

Điều kiện trước	Bộ phận xử lý đã đăng nhập hệ thống. - Sự cố tồn tại. - Người xử lý đã hoàn thành nhiệm vụ (trạng thái “Hoàn thành”). - Bộ phận xử lý đang ở màn hình “Danh sách sự cố”.
Luồng sự kiện chính	1. Bộ phận xử lý truy cập Danh sách sự cố. 2. Hệ thống hiển thị thông tin sự cố và trạng thái sự cố. 3. Bộ phận xử lý nhấn nút Đóng sự cố. 4. Hệ thống yêu cầu xác nhận “Bạn có chắc muốn đóng sự cố?” 5. Bộ phận xử lý xác nhận đóng sự cố. 6. Hệ thống cập nhật trạng thái sự cố → Đã đóng. 7. Hệ thống lưu nhật ký hoạt động (history log). 8. WebSocket gửi thông báo realtime đến Nhân viên báo cáo, Quản lý, Người xử lý. 9. Hệ thống hiển thị thông báo: “Đóng sự cố thành công.”
Luồng sự kiện phụ	Sự cố chưa được người xử lý hoàn thành → hệ thống không cho phép đóng và báo: “Nhiệm vụ chưa hoàn thành, không thể đóng sự cố.” - Sự cố đã ở trạng thái Đã đóng → hệ thống báo: “Sự cố này đã được đóng trước đó.”
Điều kiện sau	Sự cố chuyển sang trạng thái Đã đóng. - Các bên liên quan nhận được thông báo realtime. - Sự cố không còn hiển thị trong danh sách nhiệm vụ đang xử lý.

3.3.2.13. Usecase từ chối sự cố



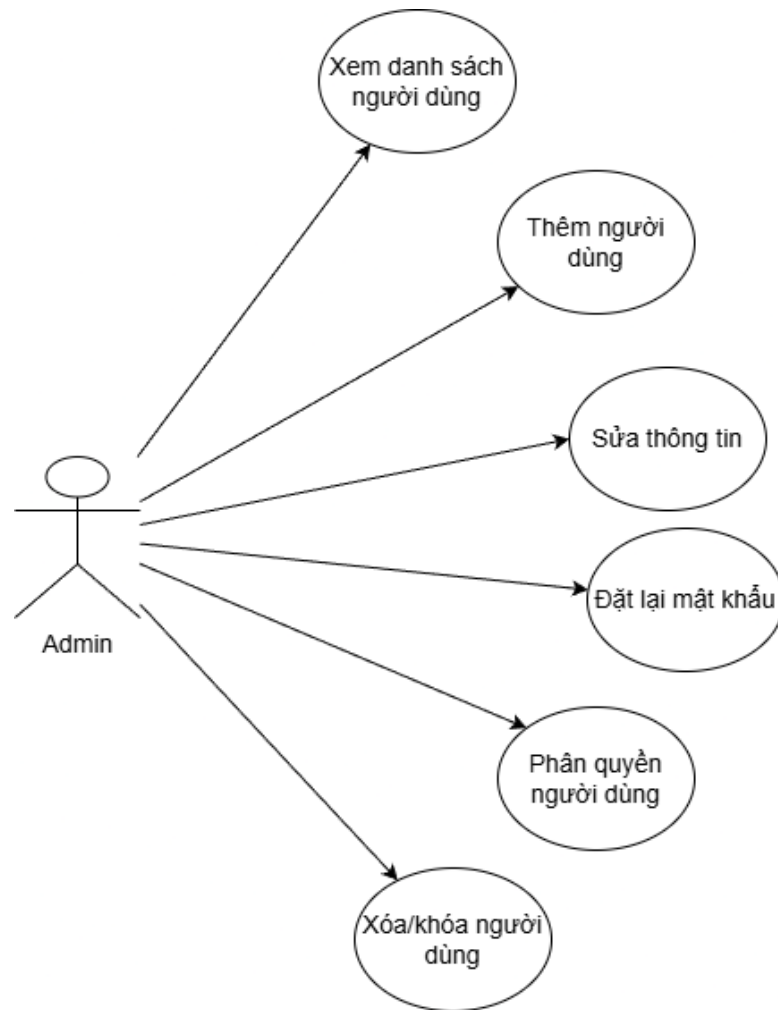
Hình 3.15. Usecase từ chối sự cố

Bảng 3.13. Đặc tả usecase từ chối sự cố

Tác nhân	Quản lý
Mô tả	Quản lý có thể từ chối sự cố khi thông tin không hợp lệ, thiếu dữ liệu, hoặc không đúng sự thật. Hệ thống cập nhật trạng thái sự cố và gửi thông báo realtime đến nhân viên báo cáo.
Điều kiện trước	Quản lý đã đăng nhập hệ thống. - Sự cố tồn tại trong hệ thống. - Sự cố đang ở trạng thái “Chờ duyệt”. - Quản lý đang ở màn hình “Danh sách sự cố”.
Luồng sự kiện chính	1. Quản lý truy cập Danh sách sự cố. 2. Hệ thống hiển thị thông tin sự cố và trạng thái sự cố. 3. Quản lý nhấn nút Từ chối sự cố. 4. Hệ thống yêu cầu nhập lý do từ chối. 5. Quản lý xác nhận từ chối sự cố.

	6. Hệ thống cập nhật trạng thái sự cố thành “Đã bị từ chối”. 7. Hệ thống lưu nhật ký hoạt động (history log). 8. WebSocket gửi thông báo realtime đến Nhân viên báo cáo. 9. Hệ thống hiển thị thông báo: “Từ chối sự cố thành công.”
Luồng sự kiện phụ	Nếu sự cố đã được phân công và người xử lý đang xử lý → hệ thống cảnh báo: “Sự cố đang xử lý, không thể từ chối.” - Nếu sự cố đã bị đóng → không thể từ chối. - Nếu lý do bị bỏ trống → hệ thống có thể hiển thị: “Bạn có muốn từ chối mà không nhập lý do?”
Điều kiện sau	Sự cố chuyển sang trạng thái Đã từ chối. - Nhân viên báo cáo nhận thông báo realtime. - Sự cố không hiển thị trong danh sách xử lý của bộ phận xử lý.

3.3.2.14. Usecase quản lý user



Hình 3.16. Usecase quản lý user

Bảng 3.14. Đặc tả usecase quản lý user

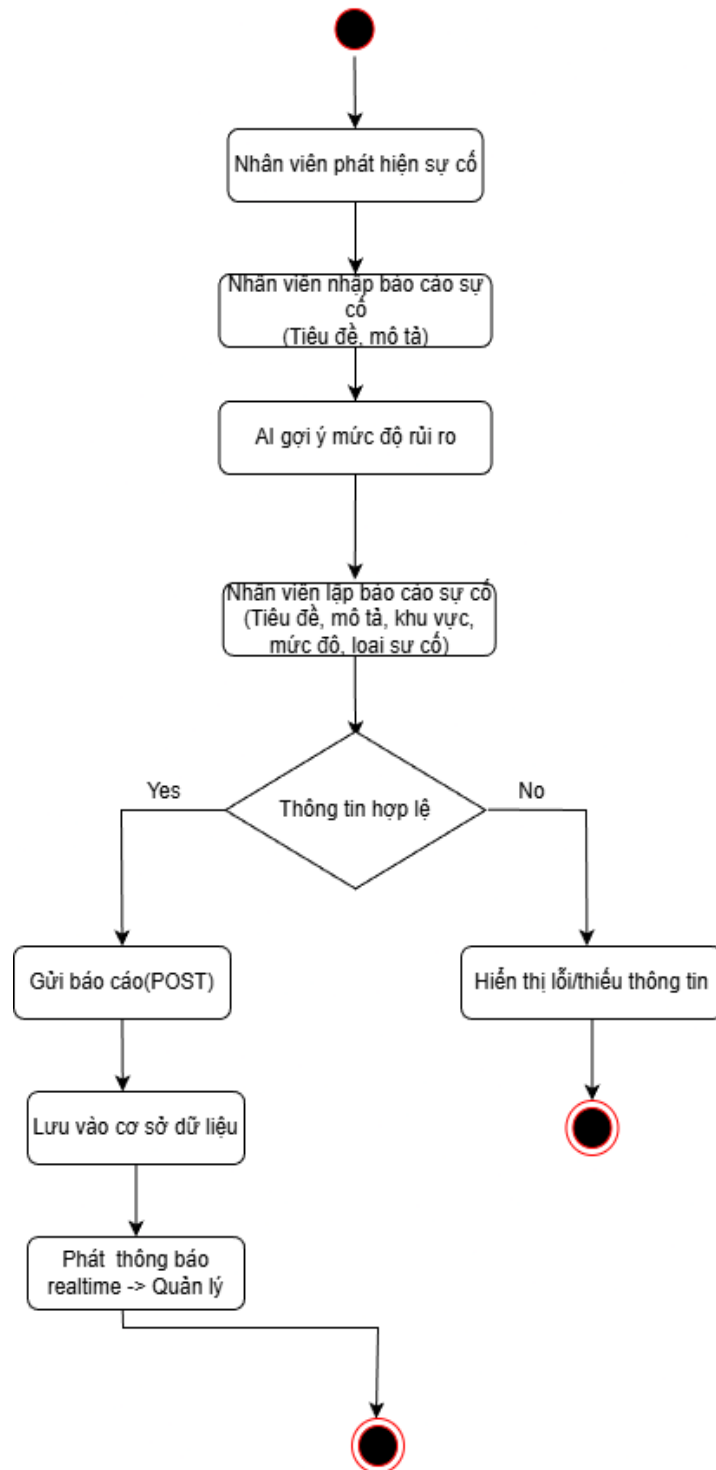
Tác nhân	Admin
Mô tả	Admin quản lý toàn bộ tài khoản người dùng trong hệ thống: thêm mới, sửa thông tin, phân quyền, khóa/xóa tài khoản và đặt lại mật khẩu. Chức năng này đảm bảo hệ thống hoạt động đúng vai trò và bảo mật.
Điều kiện trước	Admin đã đăng nhập hệ thống. Có dữ liệu người dùng trong hệ thống. Admin có quyền “Quản lý người dùng”.

Luồng sự kiện chính	1. Admin truy cập chức năng “Quản lý người dùng”. 2. Hệ thống hiển thị danh sách người dùng (tên, email, vai trò, trạng thái hoạt động). 3. Admin có thể thực hiện một trong các hành động sau: • Thêm người dùng • Chỉnh sửa • Khóa người dùng • Phân quyền • Đặt lại mật khẩu 4. Hệ thống lưu các thay đổi và hiển thị thông báo thành công.
Luồng sự kiện phụ	Email đã tồn tại → hệ thống cảnh báo: “Email này đã được sử dụng”. - Không chọn vai trò khi tạo người dùng → hệ thống yêu cầu chọn. - Xóa tài khoản không thành công do đang xử lý nhiệm vụ → hệ thống báo lỗi. - Lỗi kết nối CSDL → thông báo “Không thể lưu dữ liệu”.
Điều kiện sau	- Danh sách người dùng được cập nhật theo thay đổi. - Người dùng được thêm, sửa, phân quyền hoặc khóa thành công. - Hệ thống ghi lại log hoạt động để theo dõi bảo mật.

3.4. Biểu đồ hoạt động

3.4.1. Biểu đồ hoạt động chức năng báo cáo sự cố

Cho phép nhân viên báo cáo sự cố mới khi phát hiện sự cố.

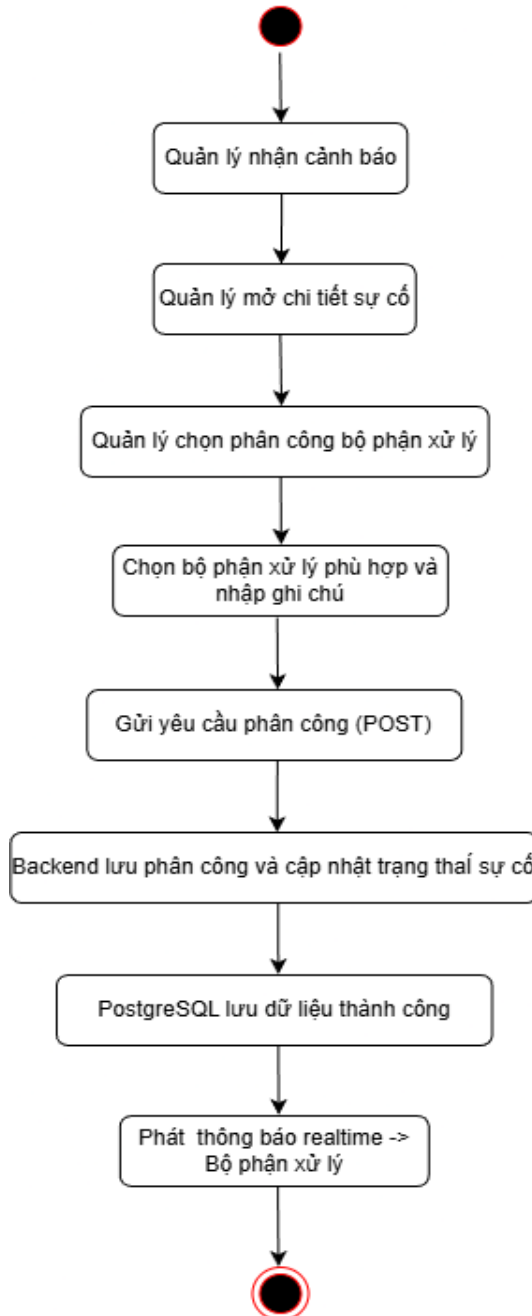


Hình 3.17. Biểu đồ hoạt động chức năng báo cáo sự

Biểu đồ này mô tả quy trình khi nhân viên báo cáo một sự cố trong hệ thống quản lý rủi ro. Toàn bộ luồng xử lý được mô tả như sau: Quy trình bắt đầu khi người dùng (nhân viên) phát hiện ra sự cố trong quá trình làm việc và cần gửi báo cáo lên hệ thống. Nhân viên nhập thông tin báo cáo sự cố gồm tiêu đề và mô tả chi tiết sự cố sau đó chọn chức năng AI gợi ý mức độ rủi ro và hoàn thiện báo cáo. Hệ thống kiểm tra tính hợp lệ nếu dữ liệu không hợp lệ quy trình kết thúc và hệ thống yêu cầu nhân viên sửa lại thông

tin. Nếu thông tin hợp lệ sẽ gửi yêu cầu API đến Backend để tạo sự cố mới, lưu báo cáo vào cơ sở dữ liệu và phát thông báo đến quản lý.

3.4.2. Biểu đồ hoạt động chức năng nhận cảnh báo và phân công bộ phận xử lý

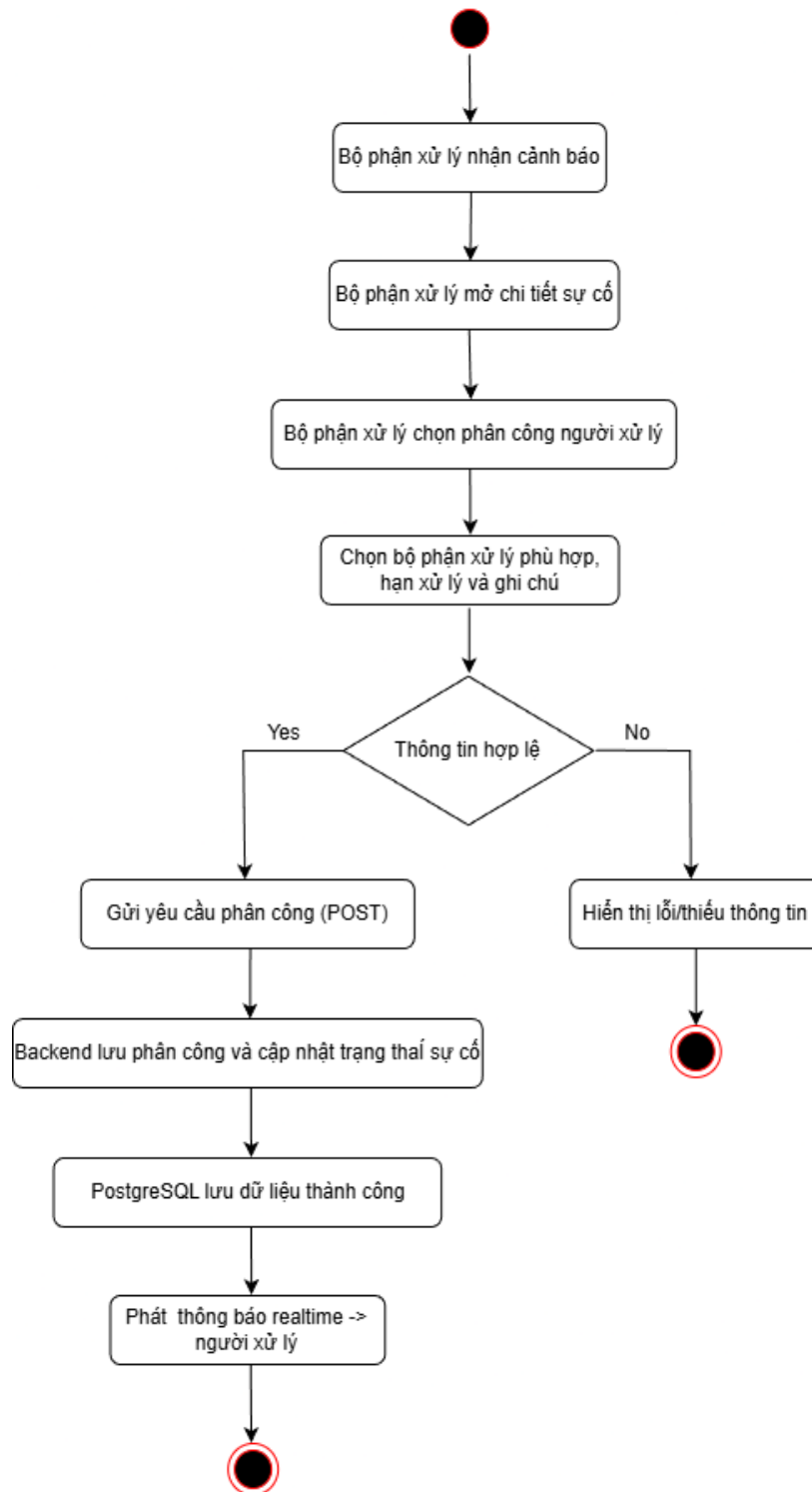


Hình 3.18. Biểu đồ hoạt động chức năng nhận cảnh báo và phân công bộ phận xử

Biểu đồ này mô tả toàn bộ luồng hoạt động khi Quản lý nhận cảnh báo sự cố mới và tiến hành phân công bộ phận xử lý. Quy trình gồm các bước sau: Hệ thống WebSocket gửi thông báo realtime đến tài khoản Quản lý khi có sự cố mới được nhân viên báo cáo. Quản lý nhấn chi tiết sự cố → hệ thống gọi API để lấy thông tin chi tiết sự cố từ cơ sở dữ liệu. Tại giao diện chi tiết sự cố, Quản lý chọn chức năng “Phân công bộ phận xử

lý”. Quản lý chọn bộ phận xử lý phù hợp và có thể nhập ghi chú kèm đề bộ phận hiểu rõ hơn. Sau đó Quản lý xác nhận phân công và thông tin phân công được lưu vào cơ sở dữ liệu và phát thông báo realtime đến bộ phận xử lý.

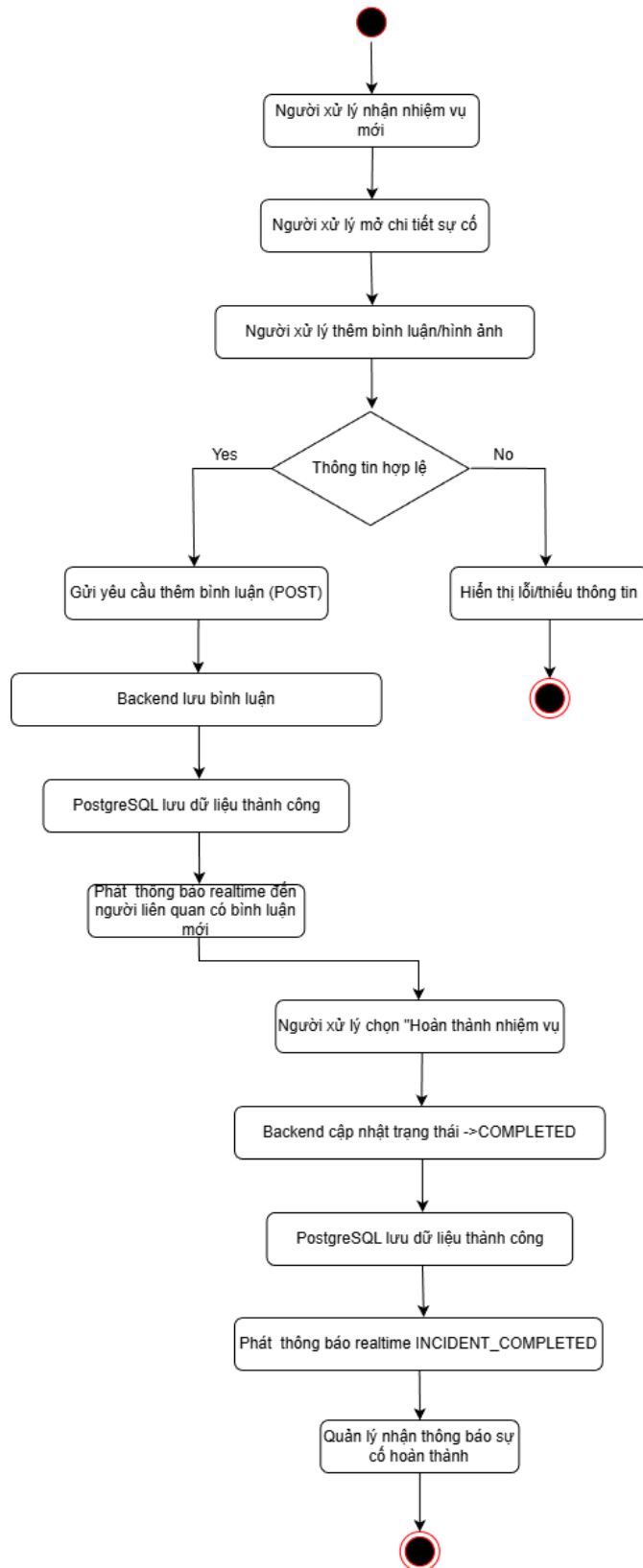
3.4.3. Biểu đồ hoạt động chức năng bộ phận xử lý nhận phân công sự cố và phân công nhiệm vụ đến người xử lý



Hình 3.19. Biểu đồ hoạt động chức năng bộ phận xử lý nhận cảnh báo và phân công người xử lý

Biểu đồ này mô tả quy trình bộ phận xử lý tiếp nhận sự cố từ quản lý và thực hiện giao nhiệm vụ xuống người xử lý phù hợp. Toàn bộ quy trình bao gồm các bước sau: Khi quản lý phân công sự cố, hệ thống WebSocket phát cảnh báo realtime đến bộ phận xử lý. Điều này đảm bảo bộ phận xử lý biết ngay khi có sự cố mới cần được xử lý. Bộ phận xử lý nhấn chi tiết phân công sự cố → hệ thống gọi API để lấy thông tin chi tiết sự cố từ cơ sở dữ liệu. Tại giao diện chi tiết phân công sự cố, Bộ phận xử lý chọn chức năng “Phân công người xử lý”. Bộ phận xử lý chọn người xử lý phù hợp, hạn xử lý kèm ghi chú. Sau đó Bộ phận xử lý xác nhận phân công và thông tin phân công được lưu vào cơ sở dữ liệu và phát thông báo realtime đến người xử lý.

3.4.4. Biểu đồ hoạt động chức năng người xử lý nhận phân công nhiệm vụ, xử lý sự cố và hoàn thành nhiệm vụ.

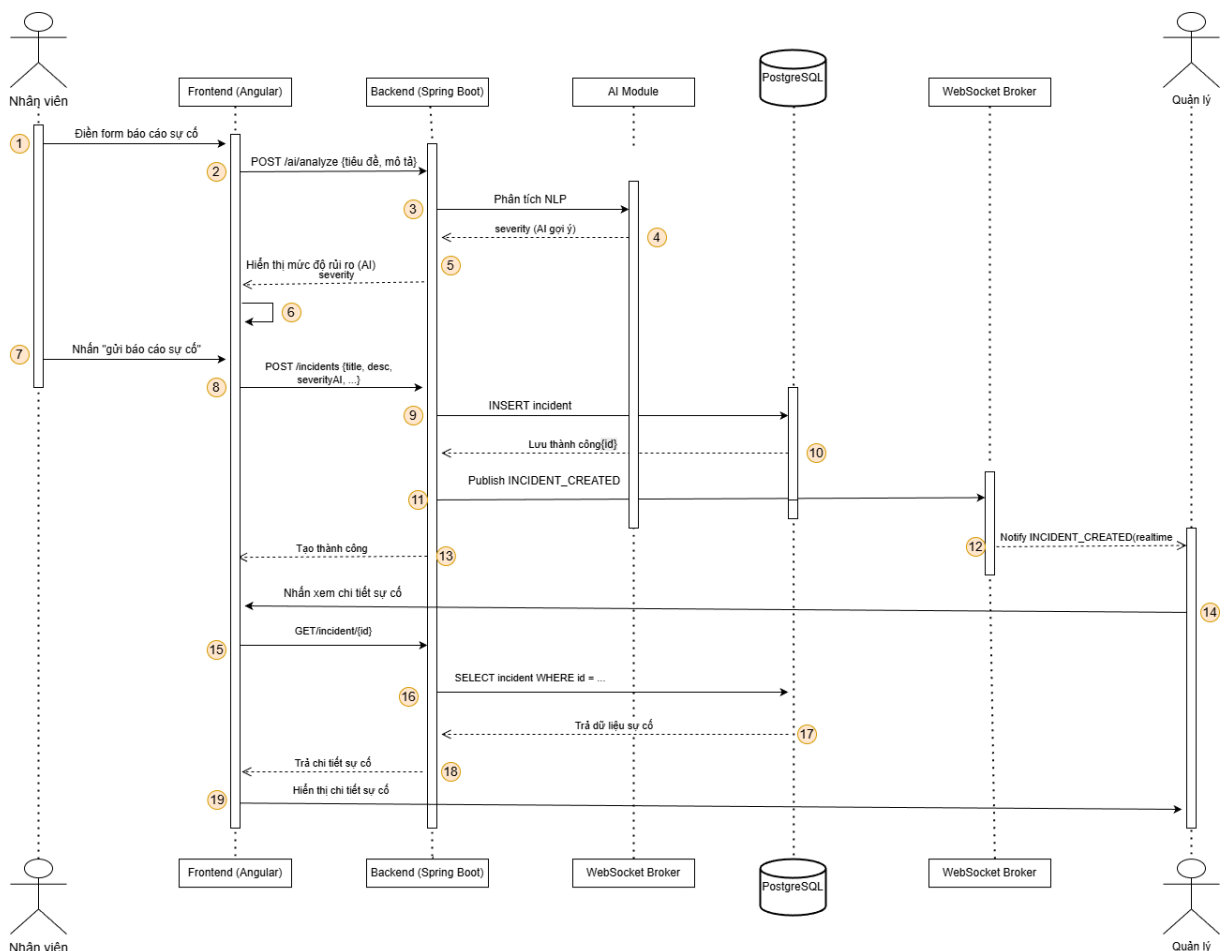


Hình 3.20. Biểu đồ hoạt động các chức năng của người xử lý

Biểu đồ này mô tả toàn bộ quy trình mà người xử lý thực hiện sau khi được phân công một sự cố bởi Bộ phận xử lý: Người xử lý nhận cảnh báo phân công mới, Người xử lý mở chi tiết sự cố xem đầy đủ thông tin mô tả, mức độ rủi ro, thời gian, hạn xử lý,... Người xử lý cập nhật tiến độ, báo cáo hiện trạng bằng văn bản hoặc hình ảnh. Hệ thống kiểm tra xem bình luận/hình ảnh có đầy đủ, đúng định dạng không nếu thiếu sẽ thông báo lỗi và yêu cầu sửa lại. Nếu đúng Gửi yêu cầu thêm bình luận và lưu vào cơ sở dữ liệu, phát thông báo realtime đến người liên quan. Khi xử lý xong người xử lý chọn “Hoàn thành nhiệm vụ” thì Backend cập nhật trạng thái sự cố sang “Hoàn thành” và lưu log. Quản lý nhận thông báo về sự cố.

3.5. Biểu đồ tuần tự

3.5.1.1. Nhân viên báo cáo sự cố & quản lý nhận cảnh báo sự cố



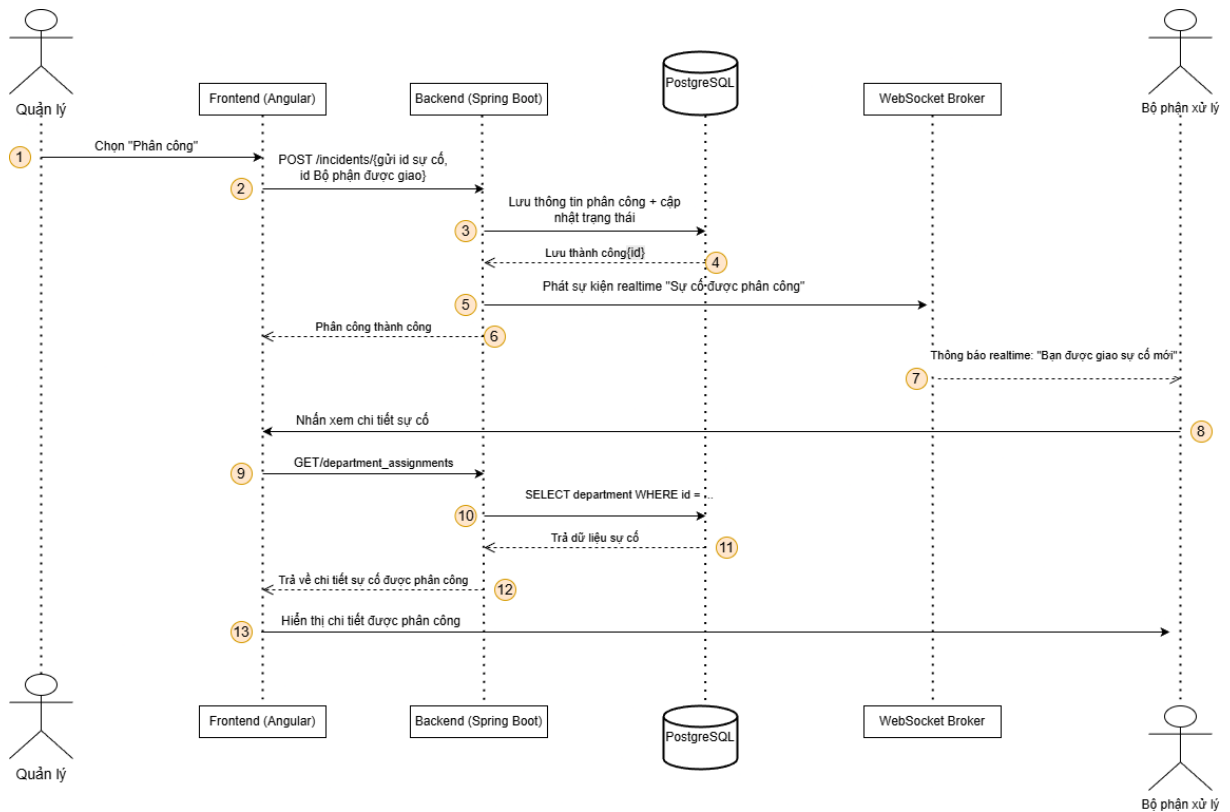
Hình 3.21. Biểu đồ tuần sự nhân viên báo cáo sự cố và quản lý nhận cảnh báo

Nhân viên muốn thực hiện chức năng báo cáo sự cố thì thực hiện như sau:

- (1) Người dùng (Nhân viên) nhập tiêu đề, mô tả, khu vực, hình ảnh (tùy chọn) vào giao diện Angular.

- (2) Hệ thống frontend gửi yêu cầu phân tích rủi ro đến Backend kèm tiêu đề và mô tả.
- (3) Backend nhận yêu cầu và gửi dữ liệu đến AI Module.
- (4) AI Module phân tích và trả về mức độ rủi ro.
- (5) Backend gửi kết quả severity về Frontend.
- (6) Frontend hiển thị mức độ rủi ro gợi ý (AI Suggestion).
- (7) Nhân viên nhấn nút “Gửi báo cáo sự cố”.
- (8) Frontend gửi API tạo sự cố đến Backend kèm theo dữ liệu: tiêu đề, mô tả, mức độ, khu vực, loại rủi ro, file đính kèm nếu có.
- (9) Backend xử lý yêu cầu và lưu sự cố vào PostgreSQL.
- (10) PostgreSQL trả về id sự cố sau khi lưu thành công.
- (11) Backend publish sự kiện INCIDENT_CREATED lên WebSocket Broker.
- (12) WebSocket Broker gửi thông báo realtime đến Quản lý.
- (13) Backend phản hồi lại Frontend “Tạo thành công”.
- (14) Quản lý nhấn xem chi tiết sự cố.
- (15) Frontend gửi GET /incidents/{id}.
- (16) Backend truy vấn PostgreSQL.
- (17) PostgreSQL trả dữ liệu sự cố.
- (18) Backend trả dữ liệu về Frontend
- (19) Quản lý xem toàn bộ thông tin sự cố.

3.5.1.2. Quản lý phân công và bộ phận xử lý nhận realtime



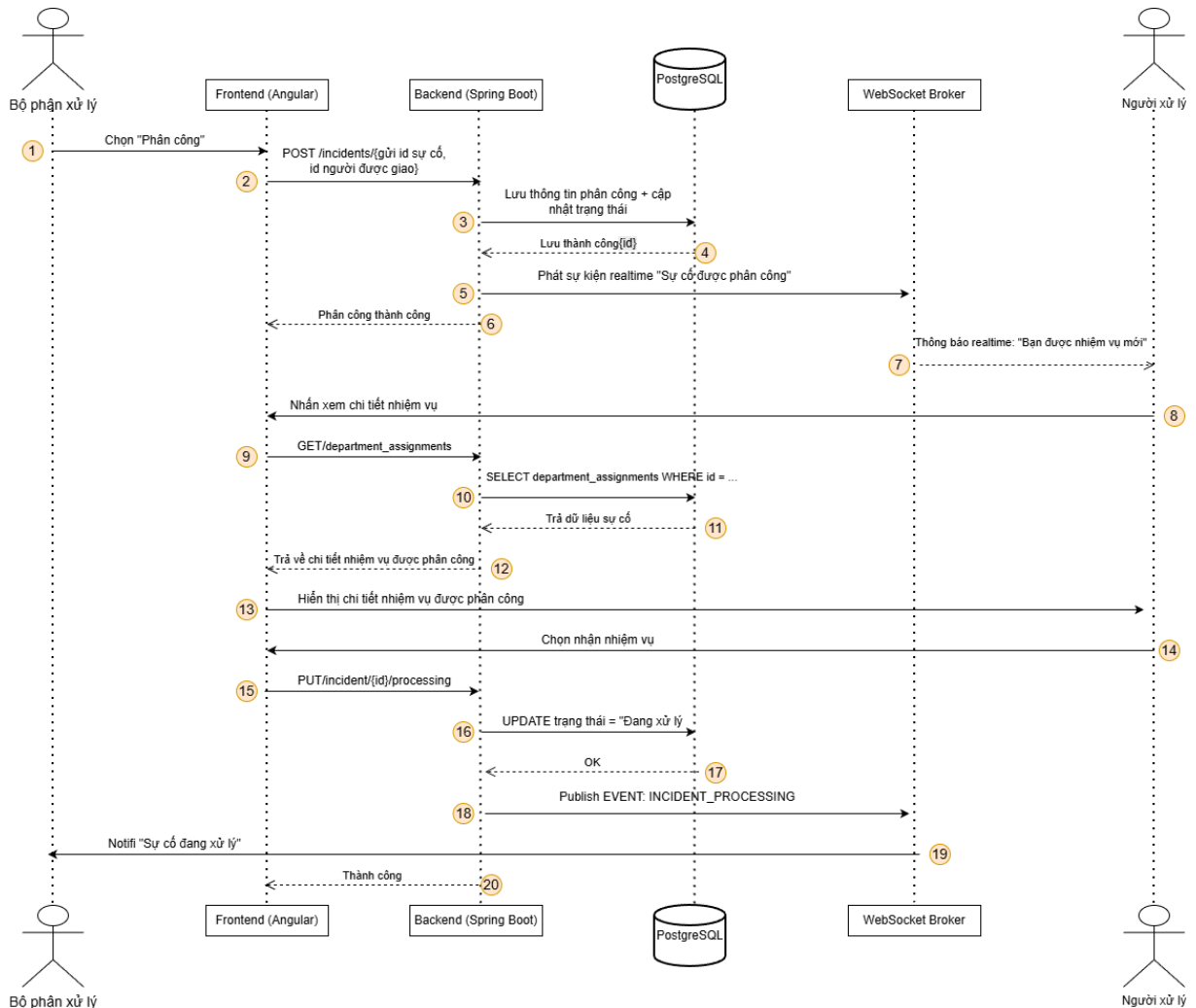
Hình 3.22. Biểu đồ tuần sự quản lý phân công và bộ phận xử lý nhận realtime

Quản lý muốn thực hiện chức năng phân công cho bộ phận xử lý thì thực hiện như sau:

- (1) Quản lý chọn Phân công sự cố trên giao diện Angular.
- (2) Frontend gửi API phân công.
- (3) Backend xử lý phân công: Thực hiện ghi nhận thông tin phân công, cập nhật trạng thái sự cố, ghi log phân công, trả về kết quả thành công.
- (4) Lưu kết quả phân công vào PostgreSQL.
- (5) Backend Publish sự kiện realtime lên WebSocket Broker.
- (6) Backend trả thông báo phân công thành công.
- (7) Bộ phận xử lý nhận thông báo realtime “Bạn được giao sự cố mới”
- (8) Bộ phận xử lý nhấn Xem chi tiết sự cố ngay sau khi nhận thông báo realtime.
- (9) Frontend gửi yêu cầu lấy thông tin chi tiết phân công.
- (10) Backend truy vấn DB lấy thông tin phân công.
- (11) PostgreSQL trả dữ liệu phân công về Backend.
- (12) Backend trả chi tiết sự cố đã phân công.

(13) Giao diện Angular hiển thị đầy đủ thông tin phân công.

3.5.1.3. Bộ phận xử lý phân công người xử lý



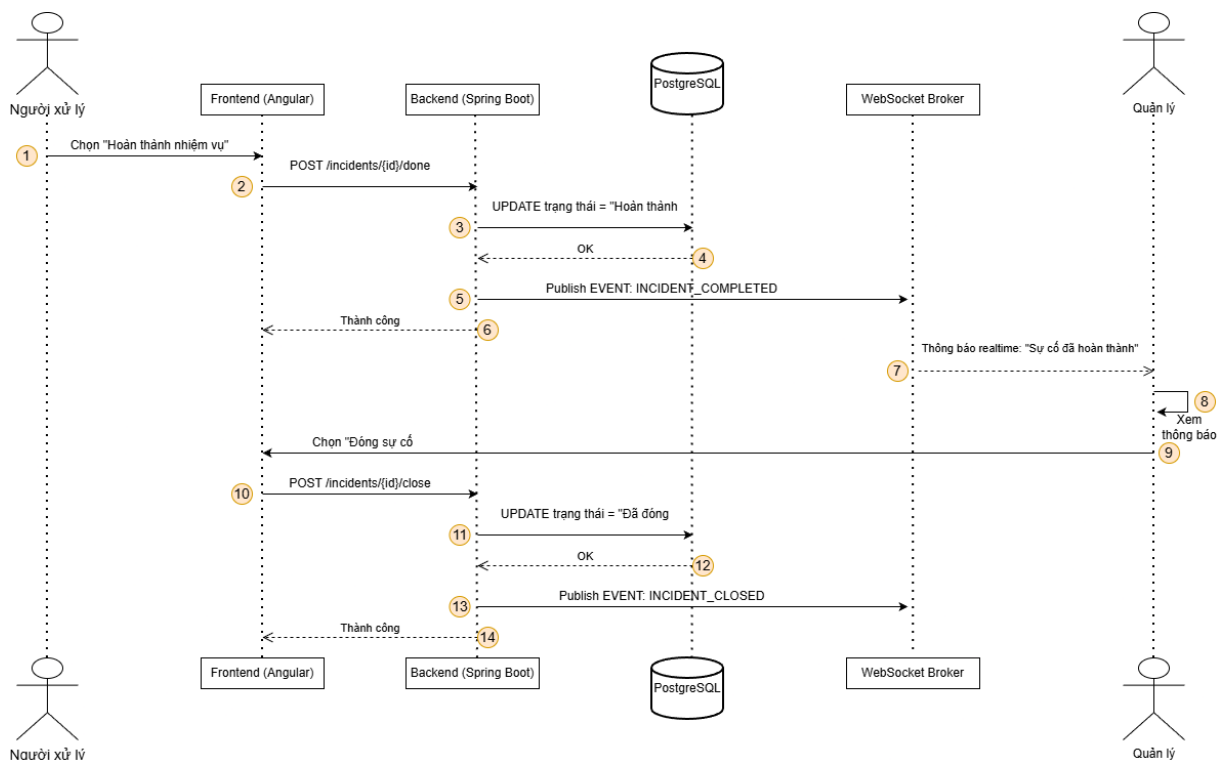
Hình 3.23. Biểu đồ tuần sự bộ phận xử lý phân công người xử lý

Bộ phận xử lý muốn thực hiện chức năng phân công cho người xử lý thì thực hiện như sau:

- (1) Bộ phận xử lý chọn “Phân công” cho người xử lý.
- (2) Frontend gửi API phân công người xử lý.
- (3) Backend xử lý phân công: Lưu thông tin phân công người xử lý, cập nhật trạng thái sự cố, ghi log phân công.
- (4) Lưu kết quả phân công vào PostgreSQL.
- (5) Backend Publish sự kiện realtime lên WebSocket Broker.
- (6) Backend trả thông báo phân công thành công.
- (7) Người xử lý nhận thông báo realtime “Bạn được giao nhiệm vụ mới”

- (8) Người xử lý nhân Xem chi tiết nhiệm vụ ngay sau khi nhận thông báo realtime.
- (9) Frontend gửi yêu cầu lấy thông tin chi tiết phân công.
- (10) Backend truy vấn DB lấy thông tin nhiệm vụ.
- (11) PostgreSQL trả dữ liệu nhiệm vụ về Backend.
- (12) Backend trả chi tiết nhiệm vụ đã phân công.
- (13) Giao diện Angular hiển thị đầy đủ thông tin nhiệm vụ được phân công.
- (14) Người xử lý chọn “Nhận nhiệm vụ”.
- (15) Frontend gửi yêu cầu cập nhật trạng thái.
- (16) Backend xử lý yêu cầu nhận nhiệm vụ: cập nhật trạng thái sự cố, thời gian bắt đầu xử lý, ghi log xử lý.
- (17) PostgreSQL lưu trạng thái mới.
- (18) Backend publish sự kiện realtime.
- (19) WebSocket Broker gửi thông báo “Sự cố đang được xử lý” đến Bộ phận xử lý.
- (20) Frontend hiển thị thông báo thành công: Người xử lý chính thức bắt đầu nhiệm vụ.

3.5.1.4. Người xử lý hoàn thành nhiệm vụ và quản lý đóng sự cố



Hình 3.24. Biểu đồ tuần sự người xử lý hoàn thành nhiệm vụ và quản lý đóng sự cố

Người xử lý muốn thực hiện chức năng hoàn thành sự cố thì thực hiện như sau:

- (1) Người xử lý chọn “Hoàn thành nhiệm vụ”
- (2) Frontend gửi API hoàn thành nhiệm vụ.
- (3) Backend xử lý yêu cầu hoàn thành nhiệm vụ: cập nhật trạng thái sự cố → COMPLETED, ghi thời gian hoàn thành, lưu log.
- (4) PostgreSQL cập nhật trạng thái và trả về ok.
- (5) Backend publish sự kiện realtime EVENT: INCIDENT_COMPLETED đến WebSocket Broker.
- (6) Backend trả kết quả “Thành công” cho Frontend.
- (7) WebSocket Broker gửi thông báo realtime đến quản lý : Sự cố đã hoàn thành.
- (8) Quản lý nhận được thông báo và mở giao diện xem thông báo.
- (9) Quản lý chọn “Đóng sự cố”.
- (10) Frontend gửi API: POST /incidents/{id}/close đến Backend.
- (11) Backend cập nhật trạng thái sự cố: cập nhật trạng thái → CLOSED, ghi log.
- (12) Backend publish sự kiện realtime EVENT: INCIDENT_CLOSED lên WebSocket Broker.
- (13) Backend trả thông báo “Đóng sự cố thành công” cho Frontend.
- (14) Frontend hiển thị “Đóng sự cố thành công”.

department_id	integer	Bộ phận	FK → departments.id
is_active	boolean	Trạng thái tài khoản	
created_at	timestamp	Thời gian tạo	

Bảng roles: Chứa danh sách các vai trò (ADMIN, MANAGER, TECH, STAFF, DEPT_MANAGER). Hệ thống dựa trên bảng này để cấp quyền truy cập các chức năng.

Bảng 3.16. roles

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID vai trò	PK
code	varchar(30)	Mã vai trò (ADMIN, MANAGER...)	UNIQUE
name	varchar(100)	Tên vai trò	
description	text	Mô tả	

Bảng departments: Lưu thông tin các bộ phận xử lý trong doanh nghiệp (Bộ phận y tế, bộ phận máy móc, bộ phận PCCC,...). Phục vụ chức năng phân công sự cố theo bộ phận.

Bảng 3.17. departments

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID bộ phận	PK
name	varchar(100)	Tên bộ phận	UNIQUE
description	text	Mô tả	
manager_id	integer	Trưởng bộ phận	FK → users.id

Bảng risk_categories: Lưu nhóm rủi ro: cháy nổ, máy móc, tai nạn lao động.

Bảng 3.18. risk_categories

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
name	varchar(120)	Tên danh mục rủi ro	
description	text	Ghi chú	

Bảng locations: Lưu danh sách các khu vực trong doanh nghiệp (Khu A, Kho, văn phòng,...). Giúp người báo cáo chọn vị trí xảy ra sự cố.

Bảng 3.19. locations

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID khu vực	PK
name	varchar(120)	Tên khu vực	
description	text	Mô tả	

Bảng incidents: Lưu toàn bộ thông tin sự cố từ khi tạo → phân công → xử lý → hoàn thành. Đây là bảng quan trọng nhất phục vụ quy trình xử lý sự cố.

Bảng 3.20. incidents

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID sự cố	PK
incident_code	varchar(30)	Mã sự cố	UNIQUE
title	varchar(200)	Tiêu đề	
description	text	Mô tả	
severity	varchar(10)	Mức độ (LOW/MEDIUM/HIGH)	Constraint
status	varchar(30)	Trạng thái (NEW/ASSIGNED/IN_PROGRESS/RESOLVED/...)	Constraint
reporter_id	integer	Người báo cáo	FK → users.id

category_id	integer	Loại rủi ro	FK → risk_categories.id
location_id	integer	Khu vực	FK → locations.id
created_at	timestamp	Ngày tạo	
updated_at	timestamp	Ngày cập nhật	Trigger auto-update

Bảng incident_media: Chứa file hình ảnh và video do nhân viên hoặc người xử lý tải lên. Giúp minh họa chi tiết sự cố.

Bảng 3.21. incident_media

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
incident_id	integer	Sự cố gắn với file	FK → incidents.id
file_url	varchar(500)	Link file	
file_type	varchar(10)	IMAGE/VIDEO	CHECK
created_at	timestamp	Thời gian tạo	

Bảng incident_comments: Lưu bình luận, ghi chú, phản hồi giữa quản lý – bộ phận xử lý – quản lý. Hỗ trợ giao tiếp trong quá trình xử lý sự cố.

Bảng 3.22. incident_comments

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
incident_id	integer	ID sự cố	FK → incidents.id
author_id	integer	Người viết bình luận	FK → users.id
content	text	Nội dung	
created_at	timestamp	Ngày tạo	
file_url	varchar(255)	File đính kèm	

Bảng incident_logs: Ghi lại mọi thay đổi trạng thái, phân công, cập nhật... của sự cố. Dùng để theo dõi quy trình và chứng minh trách nhiệm xử lý.

Bảng 3.23. incident_logs

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID log	PK
incident_id	integer	Sự cố	FK → incidents.id
actor_id	integer	Người thực hiện	FK → users.id
action	varchar(40)	Hành động (ASSIGNED, UPDATED, CLOSED...)	
old_status	varchar(30)	Trạng thái cũ	
new_status	varchar(30)	Trạng thái mới	
note	text	Ghi chú	
created_at	timestamp	Thời gian	

Bảng incident_alerts: Lưu thông tin cảnh báo được gửi tới các bộ phận/nhóm liên quan. Phục vụ tính năng cảnh báo realtime qua WebSocket.

Bảng 3.24. incident_alerts

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	bigint	ID	PK
incident_id	bigint	Sự cố	FK → incidents.id
department_id	bigint	Bộ phận nhận cảnh báo	FK → departments.id
type	varchar(50)	Loại cảnh báo	
created_at	timestamp	Khi tạo	
cleared_at	timestamp	Khi xóa cảnh báo	

Bảng notifications: Lưu thông báo cá nhân gửi qua WebSocket hoặc giao diện web. Giúp người dùng nhận thông báo sự cố mới, phân công, bình luận...

Bảng 3.25. notifications

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
user_id	integer	Người nhận	FK → users.id
ref_incident_id	integer	Sự cố liên quan	FK → incidents.id

type	varchar(40)	Kiểu thông báo	
channel	varchar(20)	web/mobile	
payload	jsonb	Nội dung JSON	
read	boolean	Đã đọc?	
read_at	timestamp	Thời gian đọc	
created_at	timestamp	Thời gian tạo	

Bảng assignments: Quản lý việc phân công trực tiếp tới người xử lý cụ thể. Bao gồm người được giao, người giao, hạn xử lý, trạng thái xác nhận.

Bảng 3.26. assignments

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID phân công	PK
incident_id	integer	Sự cố	FK → incidents.id
department_id	integer	Bộ phận (nếu có)	FK → departments.id
assignee_id	integer	Người được phân công	FK → users.id
assigned_by_id	integer	Người phân công	FK → users.id
note	text	Ghi chú	
acknowledged_at	timestamp	Thời gian người xử lý xác nhận	
due_at	timestamp	Hạn xử lý	
active	boolean	Phân công hiện tại hay không	
created_at	timestamp	Ngày tạo	
UNIQUE	(incident_id, assignee_id, active)	Chỉ 1 phân công active mỗi người	

Bảng department_assignments: Quản lý phân công cấp bộ phận xử lý.

Bảng 3.27. department_assignments

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
incident_id	integer	Sự cố	FK → incidents.id

department_id	integer	Bộ phận được phân công	FK → departments.id
assigned_by_id	integer	Người phân công	FK → users.id
note	text	Ghi chú	
assigned_at	timestamp	Ngày phân công	
status	varchar(20)	ASSIGNED / ACCEPTED / REJECTED / COMPLETED	CHECK

Bảng sla_config: Lưu thời gian phản hồi & thời gian giải quyết tối đa theo mức độ rủi ro.

Bảng 3.28. sla_config

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID	PK
severity	varchar(10)	LOW/MEDIUM/HIGH	UNIQUE
max_response_minutes	integer	Thời gian phản hồi	
max_resolution_minutes	integer	Thời gian giải quyết	
created_at	timestamp	Ngày tạo	

Bảng support_requests: Lưu trữ các yêu cầu hỗ trợ giữa các bộ phận trong quá trình xử lý sự cố

Bảng 3.29. support_requests

Trường	Kiểu dữ liệu	Mô tả	Khóa
id	integer	ID yêu cầu hỗ trợ	PK
incident_id	bigint	Mã sự cố liên quan	FK → incidents.id
requested_by	bigint	Người yêu cầu hỗ trợ	FK → users.id
department_id	bigint	Bộ phận được yêu cầu hỗ trợ	FK → departments.id

message	text	Nội dung yêu cầu hỗ trợ	
status	varchar(20)	Trạng thái: PENDING / APPROVED / REJECTED	
created_at	timestamp	Ngày tạo yêu cầu	
approved_by_id	bigint	Người duyệt yêu cầu	FK → users.id
approved_at	timestamp	Thời gian duyệt	
rejected_by_id	bigint	Người từ chối yêu cầu	FK → users.id
rejected_at	timestamp	Thời gian từ chối	
reject_reason	text	Lý do từ chối	

CHƯƠNG 4. XÂY DỰNG HỆ THỐNG QUẢN LÝ RỦI RO VÀ CẢNH BÁO SỰ CỐ

4.1. Môi trường phát triển và công cụ lập trình

Đề tài được xây dựng trên môi trường phát triển:

- Vi xử lý: Intel Core i5-12500H
- RAM: 16GB
- Hệ điều hành: Windows 11 – Build 26200 – 64-bit

Công cụ lập trình:

- Phần mềm viết code: Visual Studio Code, IntelliJ IDEA.
- Cơ sở dữ liệu: PostgreSQL.
- Framework: Spring Boot, Angular.
- Ngôn ngữ: Java, TypeScript.

4.2. Triển khai hệ thống

4.2.1. Triển khai Frontend Angular

Frontend của hệ thống được xây dựng bằng Angular 20, kết hợp với Bootstrap 5, AdminLTE 4, và một số thư viện hỗ trợ như Chart.js, SweetAlert2, STOMP/SockJS để xử lý realtime. Đây là bộ công nghệ hiện đại giúp giao diện trực quan, mượt mà và dễ mở rộng.

Môi trường lập trình: Mã nguồn Frontend được phát triển trên Visual Studio Code, IDE phổ biến cho lập trình web, cung cấp khả năng linting, IntelliSense, hỗ trợ Angular CLI và tích hợp terminal mạnh mẽ.

Ứng dụng được chạy mặc định trên cổng 4200, cấu hình bởi Angular CLI: `ng serve` → `http://localhost:4200`

4.2.2. Triển khai backend Spring Boot

4.2.2.1. Cấu trúc và môi trường Backend

Backend được xây dựng bằng Spring Boot 3.3.4, sử dụng Java 21 và hệ quản trị cơ sở dữ liệu PostgreSQL. Bộ dự án dùng Maven để quản lý phụ thuộc và Flyway để quản lý phiên bản cơ sở dữ liệu.

Môi trường lập trình: Toàn bộ mã nguồn Backend được phát triển và chạy trên IDE IntelliJ IDEA Ultimate, hỗ trợ tốt cho Spring Boot, Maven, Swagger và tích hợp chạy thử trực tiếp bằng Spring Boot Run.

Ứng dụng được chạy mặc định trên cổng 8080 Khi khởi chạy Backend qua IntelliJ, hệ thống sẽ hoạt động tại địa chỉ <http://localhost:8080>.

Các dependency chính trong backend bao gồm:

- spring-boot-starter-web: xây dựng API RESTful.
- spring-boot-starter-security + JWT: xác thực người dùng.
- spring-boot-starter-data-jpa: truy vấn dữ liệu bằng Hibernate JPA.
- spring-boot-starter-websocket: triển khai WebSocket realtime.
- spring-ai-starter-model-openai (Groq): tích hợp AI phân tích mức độ rủi ro.
- springdoc-openapi: tự động sinh trang Swagger UI.
- flyway-core: quản lý database migration.

4.2.2.2. Cấu hình kết nối cơ sở dữ liệu

- spring.datasource.url=jdbc:postgresql://localhost:5432/webcanhbao
- spring.datasource.username=postgres
- spring.datasource.password=123456
- spring.datasource.driver-class-name=org.postgresql.Driver

4.2.2.3. Cấu hình bảo mật và JWT

- jwt.secret=EGYcY4xyT...
 - jwt.expiration=86400000
- Luồng hoạt động:
- Người dùng đăng nhập (email + password).
 - Backend tạo accessToken.
 - Mỗi request gửi kèm token trong header.

4.2.2.4. Cấu hình AI Groq – Spring AI

- Thêm dependency Spring AI vào dự án: Trong file pom.xml, khai báo thư viện Spring AI và module OpenAI model (được Groq hỗ trợ tương thích OpenAI API):

```
<dependency>  
  
    <groupId>org.springframework.ai</groupId>  
  
    <artifactId>spring-ai-openai-spring-boot-starter</artifactId>
```

<version>1.0.2</version>

</dependency>

Thư viện này cung cấp sẵn các cấu hình tự động và đối tượng ChatClient giúp gọi mô hình AI thông qua API mà không cần xử lý thủ công.

- Khai báo thông tin cấu hình trong file application.properties: Cấu hình thông tin API của Groq để Spring AI có thể kết nối tới mô hình ngôn ngữ (LLM):

- spring.ai.openai.api-key=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
- spring.ai.openai.chat.base-url=https://api.groq.com/openai
- spring.ai.openai.chat.options.model=llama-3.1-8b-instant

Trong đó:

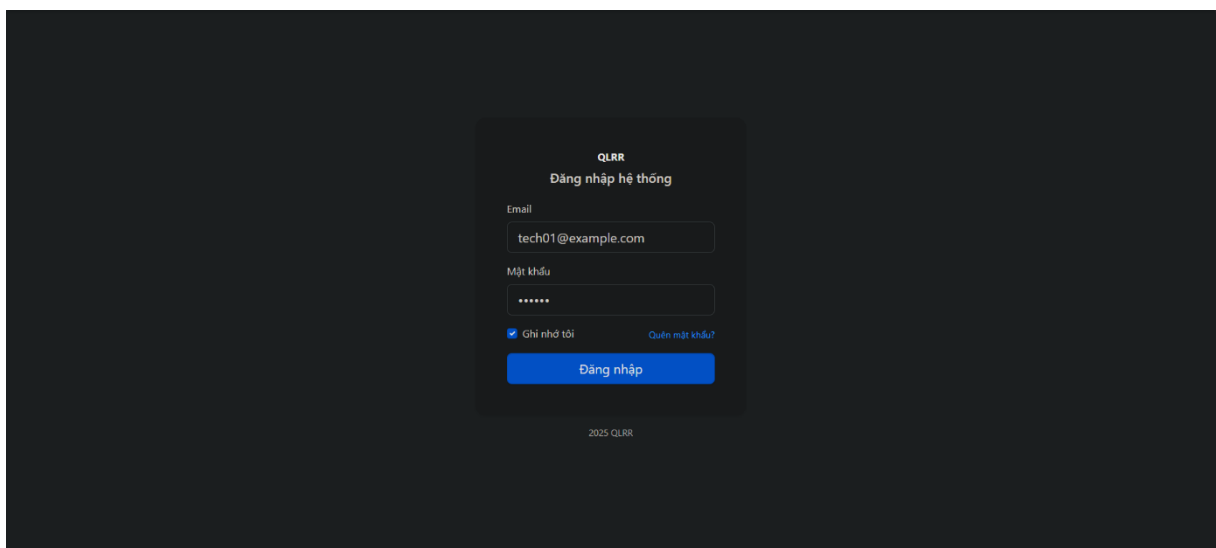
- api-key: khóa truy cập miễn phí hoặc do hệ thống Groq cung cấp;
- base-url: địa chỉ Groq tương thích với giao thức OpenAI;
- model: tên mô hình ngôn ngữ sử dụng (ở đây là LLaMA 3.1 8B Instant – mô hình tốc độ cao, tối ưu cho phản hồi nhanh).

4.3. Kết quả đạt được

4.3.1. Giao diện nhân viên

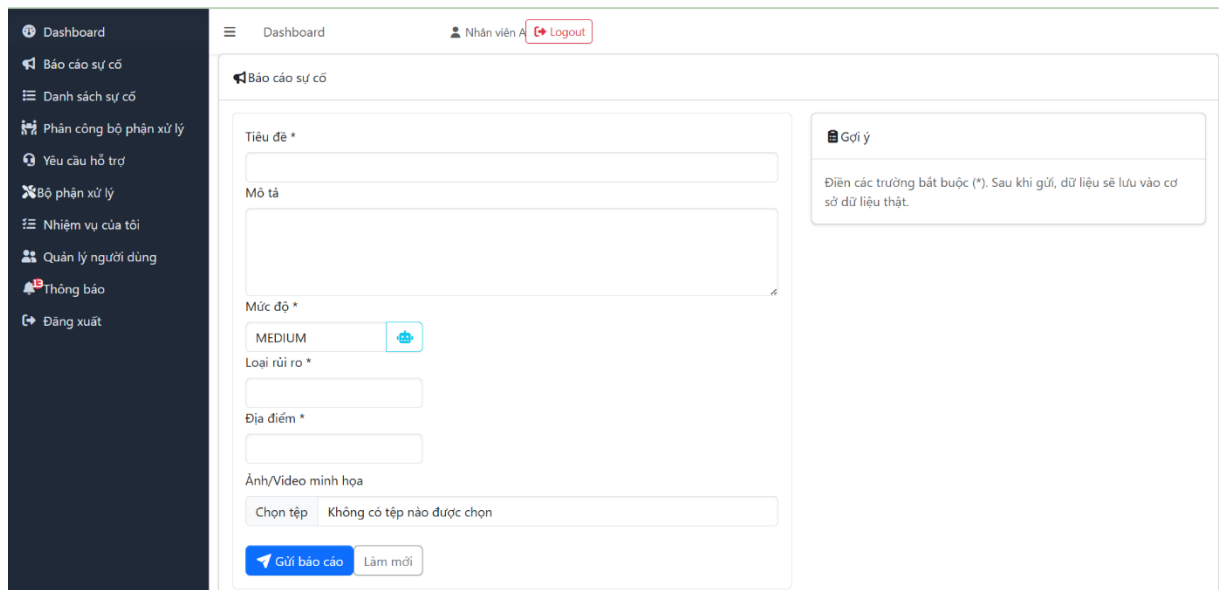
4.3.1.1. Giao diện đăng nhập

Nhân viên, Quản lý, Bộ phận xử lý, Người xử lý, Admin đều sử dụng chung giao diện đăng nhập này:



Hình 4.1. Giao diện trang đăng nhập

4.3.1.2. Giao diện form báo cáo sự cố

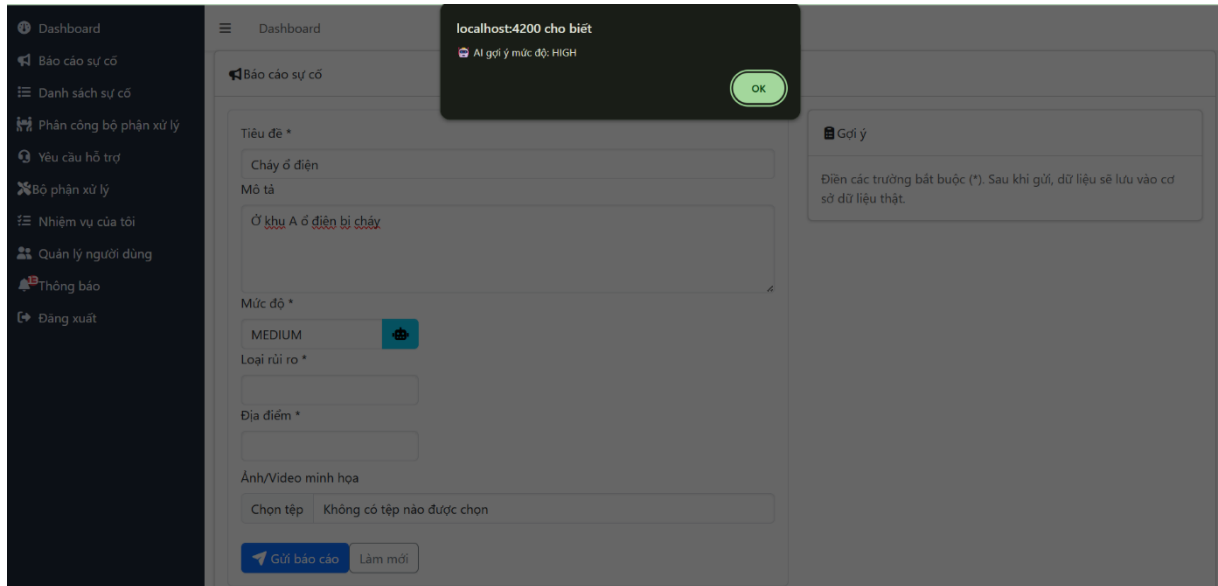


The screenshot shows a web application interface for reporting incidents. On the left is a dark sidebar with a menu containing: Dashboard, Báo cáo sự cố (Incident Report), Danh sách sự cố (Incident List), Phân công bộ phận xử lý (Assign department for handling), Yêu cầu hỗ trợ (Request support), Bộ phận xử lý (Handling department), Nhiệm vụ của tôi (My tasks), Quản lý người dùng (User management), Thông báo (Notification), and Đăng xuất (Logout). The main content area is titled 'Báo cáo sự cố' (Incident Report). It contains several input fields: 'Tiêu đề *' (Title), 'Mô tả' (Description), 'Mức độ *' (Severity) with a dropdown menu currently showing 'MEDIUM', 'Loại rủi ro *' (Risk type), 'Địa điểm *' (Location), and 'Ảnh/Video minh họa' (Illustration photo/video) with a file selection button. At the bottom are two buttons: 'Gửi báo cáo' (Submit report) and 'Làm mới' (Refresh). On the right side of the form, there is a 'Gợi ý' (Suggestion) box with the text: 'Điền các trường bắt buộc (*). Sau khi gửi, dữ liệu sẽ lưu vào cơ sở dữ liệu thật.' (Fill in the required fields (*). After sending, the data will be saved to the real database.)

Hình 4.2. Giao diện form báo cáo sự cố

Màn hình giao diện này khi nhân viên phát hiện có sự cố sẽ báo cáo tại form này, điền đầy đủ các thông tin về sự cố.

4.3.1.3. Giao diện nhân viên sử dụng chức năng AI

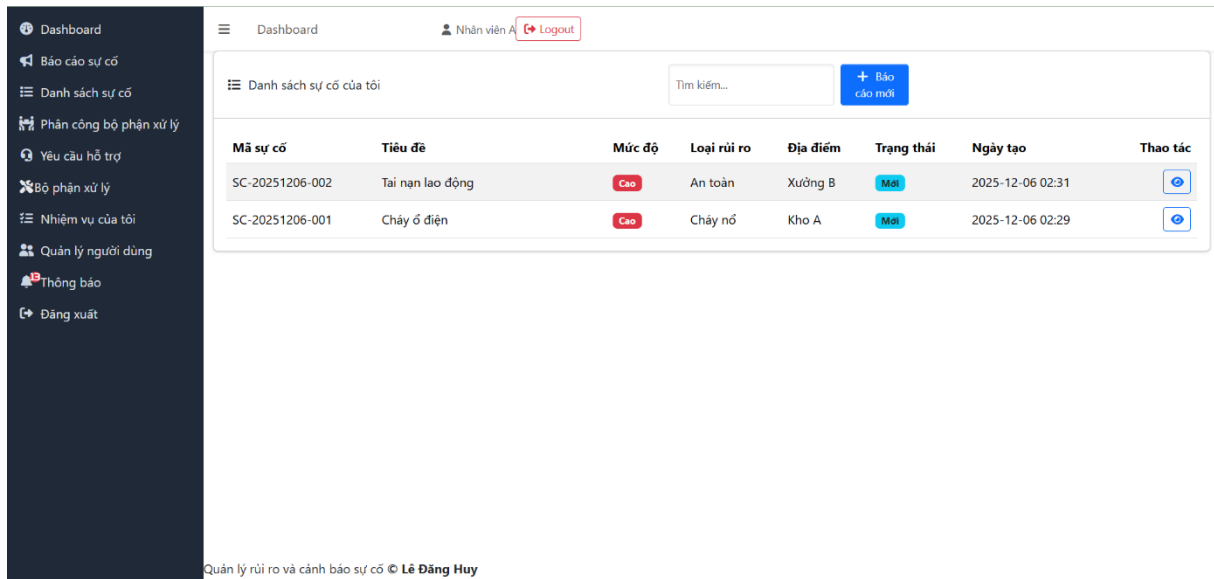


This screenshot shows the same incident reporting form as in Figure 4.2, but with an AI suggestion overlay. The overlay is a dark box with the text 'localhost:4200 cho biết' (localhost:4200 informs) and 'AI gợi ý mức độ: HIGH' (AI suggests severity: HIGH), with an 'OK' button. The form fields are partially filled: 'Tiêu đề *' contains 'Cháy ổ điện' (Power supply fire), 'Mô tả' contains 'Ở khu A ổ điện bị cháy' (In area A, the power supply is on fire), and 'Mức độ *' is set to 'MEDIUM'. The 'Loại rủi ro *' field is empty. The 'Ảnh/Video minh họa' section shows a file selection button with the text 'Không có tệp nào được chọn' (No files selected).

Hình 4.3. Giao diện khi sử dụng AI

Màn hình giao diện này cho phép nhân viên sử dụng chức năng AI dự đoán mức độ rủi ro khi nhân viên nhập tiêu đề và mô tả về sự cố.

4.3.1.4. Giao diện danh sách sự cố của tôi

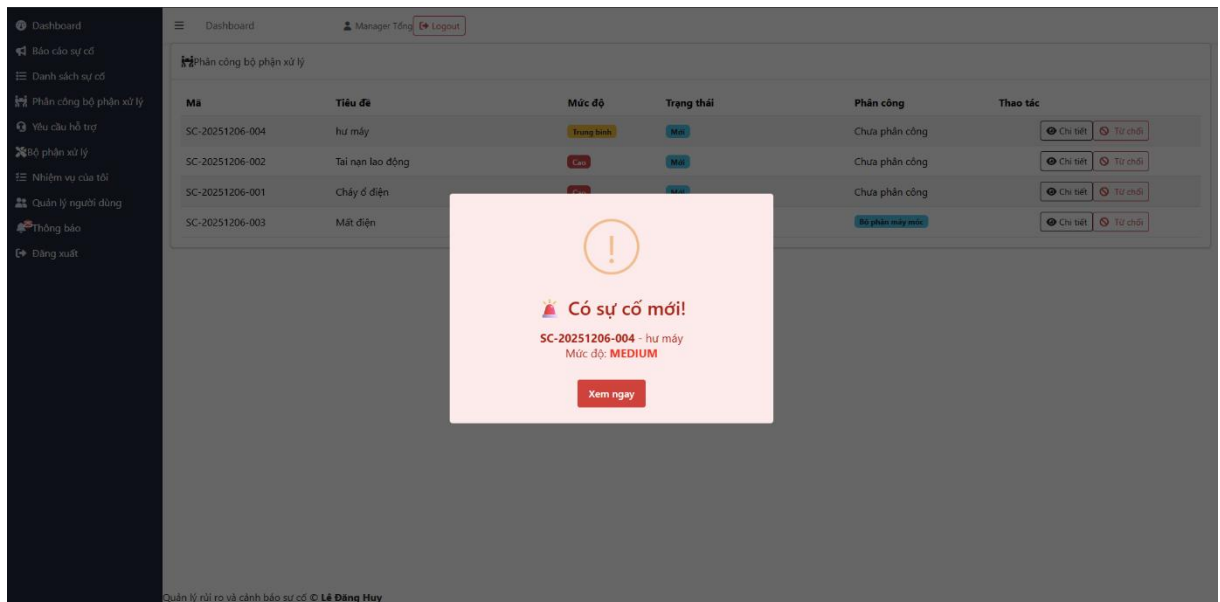


Hình 4.4. Giao diện danh sách sự cố của tôi

Màn hình giao diện này cho phép nhân viên xem danh sách sự cố đã báo cáo

4.3.2. Giao diện quản lý

4.3.2.1. Giao diện hiển thị khi có sự cố mới



Hình 4.5. Giao diện hiển thị cảnh báo sự cố mới

Màn hình giao diện quản lý nhận được cảnh báo realtime khi có sự cố mới

4.3.2.2. Giao diện danh sách sự cố

Mã	Tiêu đề	Mức độ	Trạng thái	Phản công	Thao tác
SC-20251206-004	hư máy	Trung bình	Mới	Chưa phản công	☐ Chi tiết ☐ Từ chối
SC-20251206-002	Tai nạn lao động	Cao	Mới	Chưa phản công	☐ Chi tiết ☐ Từ chối
SC-20251206-001	Chảy ố điện	Cao	Mới	Chưa phản công	☐ Chi tiết ☐ Từ chối
SC-20251206-003	Mất điện	Cao	Chờ người xử lý nhận việc	Bộ phận máy móc	☐ Chi tiết ☐ Từ chối

Hình 4.6. Giao diện danh sách sự cố

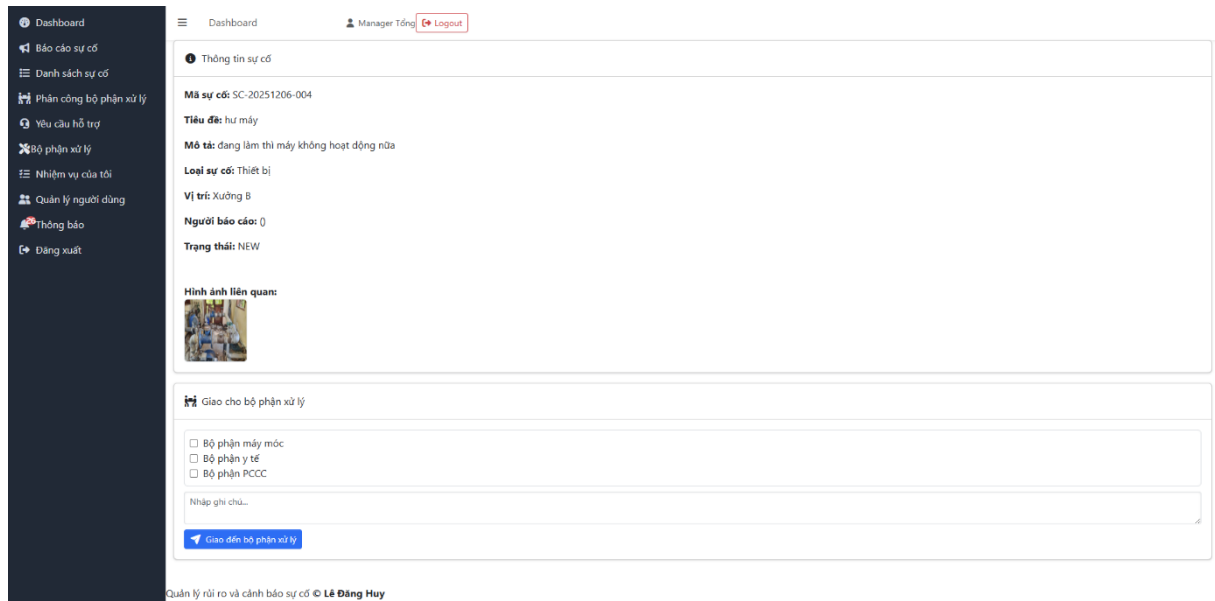
Màn hình giao diện này hiển thị danh sách sự cố vừa được báo cáo, sự cố phân công, các sự cố đã xử lý xong.

4.3.2.3. Giao diện từ chối sự cố

Hình 4.7. Giao diện từ chối sự cố

Màn hình giao diện này cho phép quản lý từ chối sự cố nếu sự cố thiếu thông tin hoặc thông tin không chính xác.

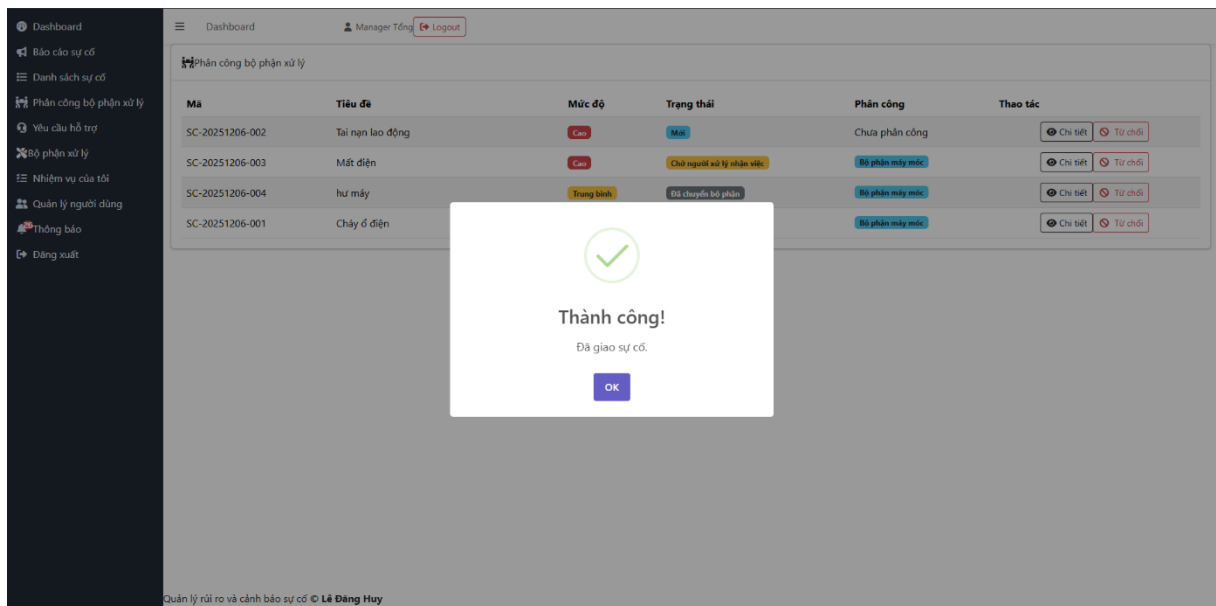
4.3.2.4. Giao diện xem chi tiết sự cố



Hình 4.8. Giao diện chi tiết sự cố và chức năng phân công

Màn hình giao diện này cho phép quản lý xem thông tin về sự cố và phân công đến bộ phận xử lý để xử lý sự cố.

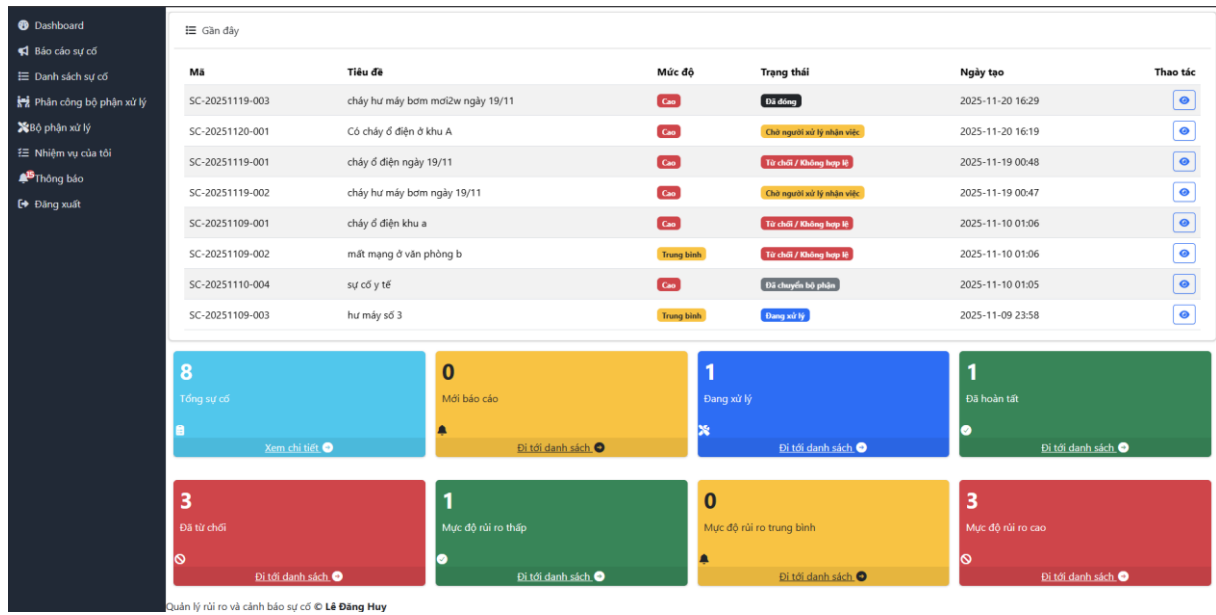
4.3.2.5. Giao diện giao thành công



Hình 4.9. Giao diện hiển thị giao thành công

Màn hình giao diện này hiển thị thông báo khi quản lý giao sự cố đến bộ phận xử lý thành công.

4.3.2.6. Giao diện dashboard

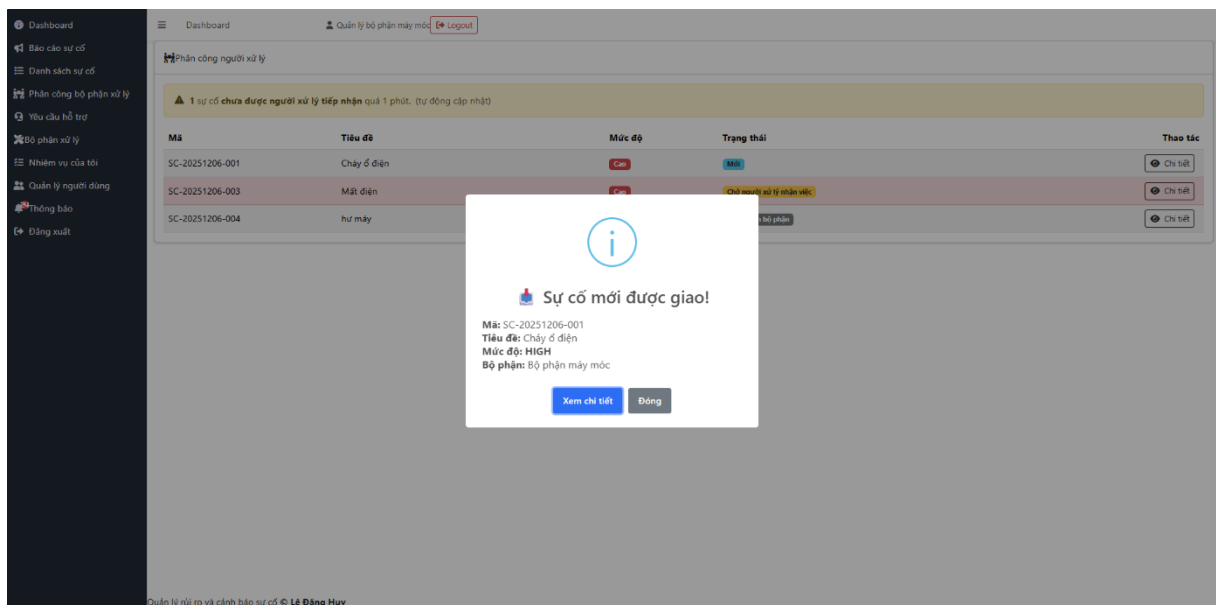


Hình 4.10. Giao diện dashboard

Màn hình giao diện này hiển thị các sự cố gần đây và tổng quan về trạng thái của sự cố, mức độ rủi ro của sự cố.

4.3.3. Giao diện Bộ phận xử lý

4.3.3.1. Giao diện nhận cảnh báo sự cố từ quản lý



Hình 4.11. Giao diện nhận cảnh báo sự cố mới được giao

Màn hình giao diện này hiển thị thông báo khi có sự cố mới được phân công từ quản lý.

4.3.3.2. Giao diện hiển thị danh sách sự cố được giao

The screenshot shows a dashboard for incident management. On the left is a sidebar with navigation links: Dashboard, Báo cáo sự cố, Danh sách sự cố, Phân công bộ phận xử lý, Yêu cầu hỗ trợ, Bộ phận xử lý, Nhiệm vụ của tôi, Quản lý người dùng, Thông báo, and Đăng xuất. The main content area is titled 'Phân công người xử lý' and shows a warning: '1 sự cố chưa được người xử lý tiếp nhận quá 1 phút. (tự động cấp nhật)'. Below this is a table of incidents:

Mã	Tiêu đề	Mức độ	Trạng thái	Thao tác
SC-20251206-001	Cháy ổ điện	Cao	Mới	Chi tiết
SC-20251206-003	Mất điện	Cao	Chờ người xử lý nhận việc	Chi tiết
SC-20251206-004	hư máy	Trung bình	Đã chuyển bộ phận	Chi tiết

At the bottom left, it says 'Quản lý rủi ro và cảnh báo sự cố © Lê Đăng Huy'.

Hình 4.12. Giao diện danh sách sự cố

Màn hình giao diện này hiển thị danh sách sự cố được giao và hiển thị cảnh báo khi người xử lý không nhận nhiệm vụ.

4.3.3.3. Giao diện hiển thị danh sách yêu cầu hỗ trợ

The screenshot shows a dashboard for support request management. On the left is a sidebar with navigation links: Dashboard, Báo cáo sự cố, Danh sách sự cố, Phân công bộ phận xử lý, Yêu cầu hỗ trợ, Bộ phận xử lý, Nhiệm vụ của tôi, Quản lý người dùng, Thông báo, and Đăng xuất. The main content area is titled 'Danh sách yêu cầu hỗ trợ' and has a dropdown menu set to 'Tất cả'. Below this is a table of support requests:

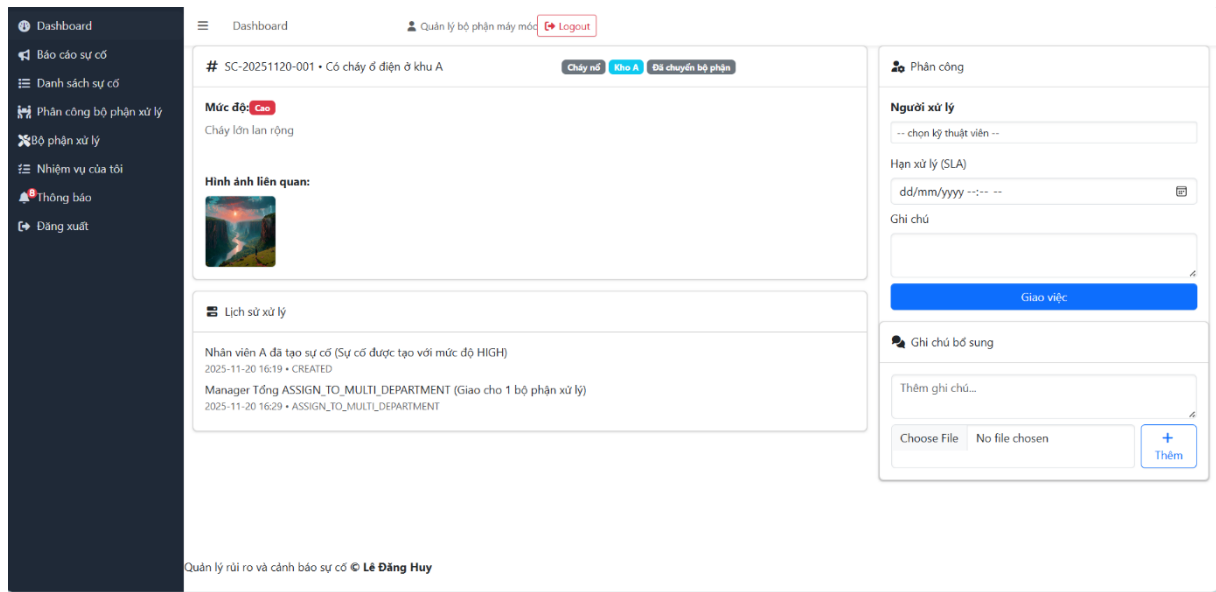
Mã SC	Tiêu đề	Người yêu cầu	Nội dung	Trạng thái	Ngày yêu cầu	Thao tác
SC-20251206-003	Mất điện	Kỹ thuật viên A	cần 2 thêm người giúp	Chờ duyệt	2025-12-06 02:36	Chi tiết Duyệt Hủy

At the bottom left, it says 'Quản lý rủi ro và cảnh báo sự cố © Lê Đăng Huy'.

Hình 4.13. Giao diện danh sách yêu cầu hỗ trợ

Màn hình giao diện này hiển thị danh sách yêu cầu hỗ trợ khi người xử lý có yêu cầu.

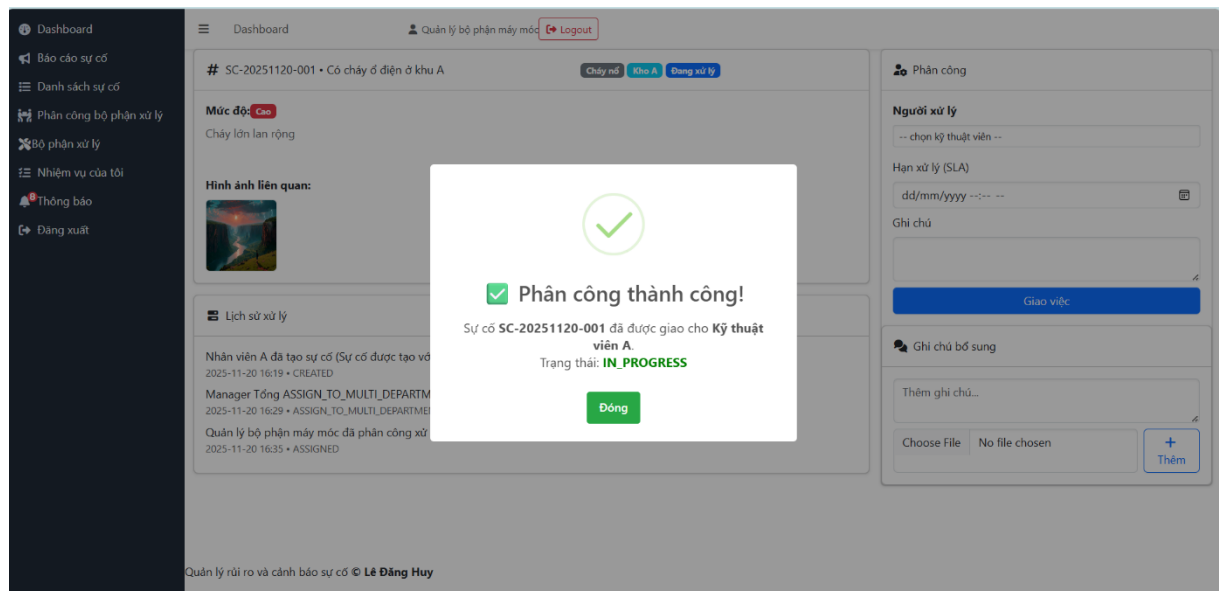
4.3.3.4. Giao diện hiển thị chi tiết sự cố của bộ phận xử lý



Hình 4.14. Giao diện hiển thị chi tiết sự cố được giao

Màn hình giao diện này hiển thị chi tiết sự cố được giao và chức năng phân công người xử lý của bộ phận xử lý.

4.3.3.5. Giao diện thông báo phân công nhiệm vụ

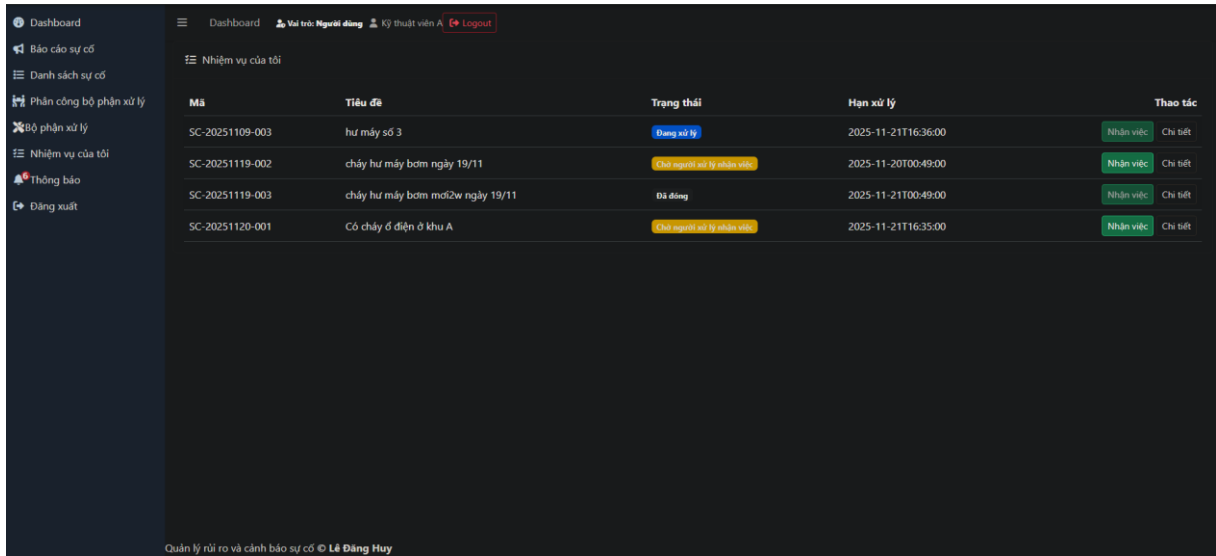


Hình 4.15. Giao diện thông báo phân công

Màn hình giao diện này thông báo cho bộ phận xử lý biết đã giao nhiệm vụ cho người xử lý thành công.

4.3.4. Giao diện người xử lý

4.3.4.1. Giao diện danh sách nhiệm vụ

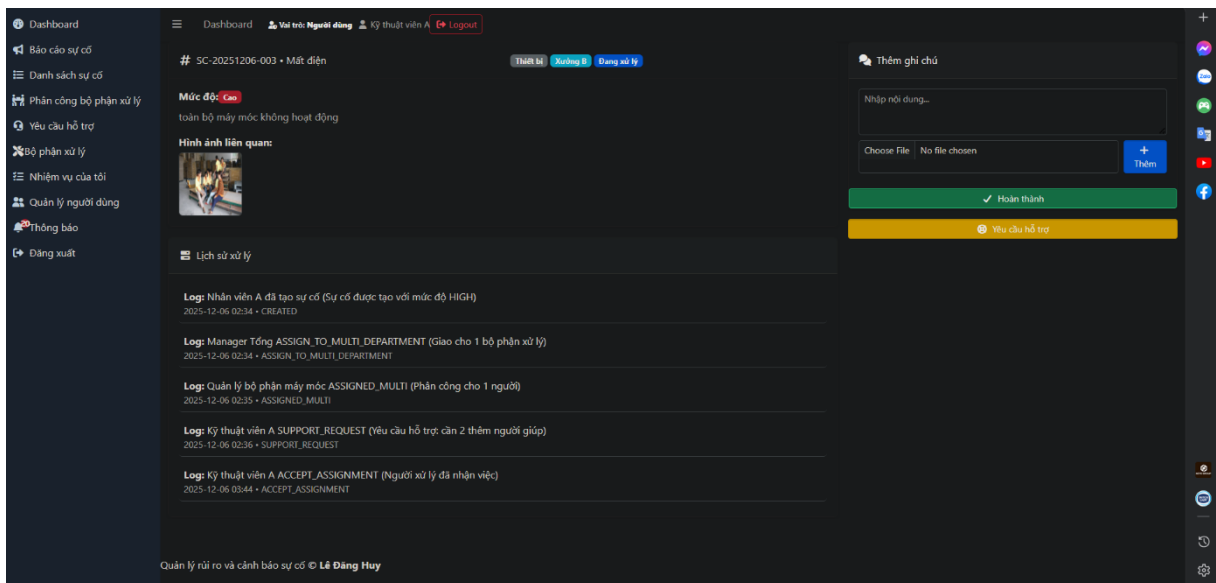


Mã	Tiêu đề	Trạng thái	Hạn xử lý	Thao tác
SC-20251109-003	hư máy số 3	Đang xử lý	2025-11-21T16:36:00	Nhận việc Chi tiết
SC-20251119-002	cháy hư máy bơm ngày 19/11	Chờ người xử lý nhận việc	2025-11-20T00:49:00	Nhận việc Chi tiết
SC-20251119-003	cháy hư máy bơm mđ2w ngày 19/11	Đã đóng	2025-11-21T00:49:00	Nhận việc Chi tiết
SC-20251120-001	Có cháy ổ điện ở khu A	Chờ người xử lý nhận việc	2025-11-21T16:35:00	Nhận việc Chi tiết

Hình 4.16. Giao diện danh sách nhiệm vụ

Màn hình giao diện hiển thị danh sách nhiệm vụ được giao của người xử lý.

4.3.4.2. Giao diện chi tiết nhiệm vụ



Mức độ: Cao

toàn bộ máy móc không hoạt động

Hình ảnh liên quan:

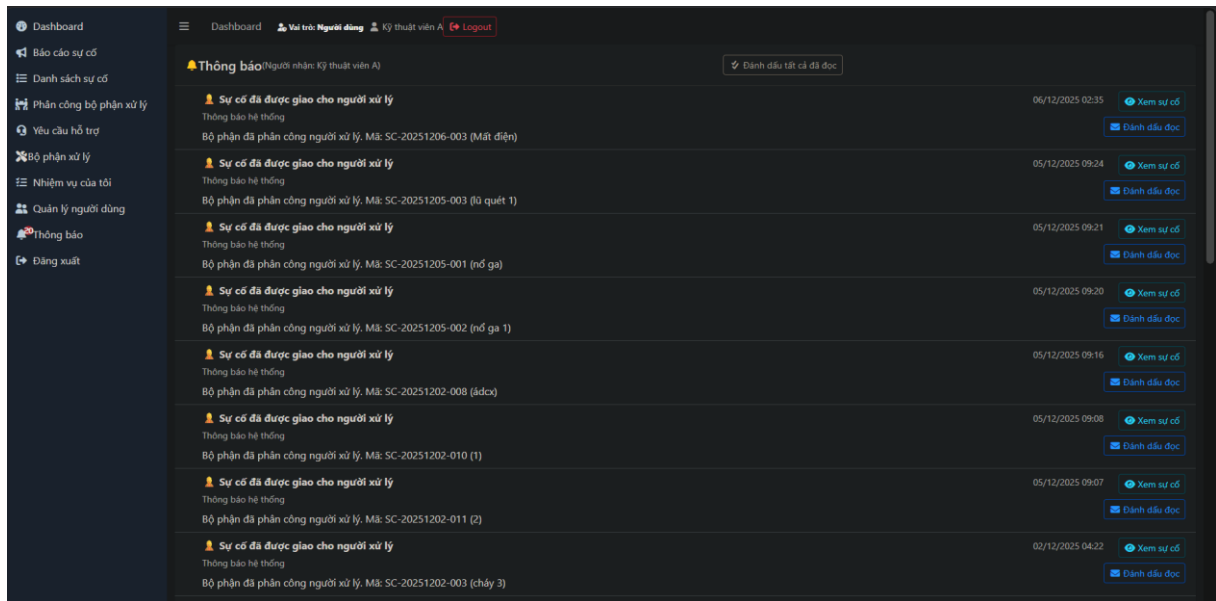
Lịch sử xử lý:

- Log:** Nhân viên A đã tạo sự cố (Sự cố được tạo với mức độ HIGH)
2025-12-06 02:34 • CREATED
- Log:** Manager Tổng ASSIGN_TO_MULTI_DEPARTMENT (Giao cho 1 bộ phận xử lý)
2025-12-06 02:34 • ASSIGN_TO_MULTI_DEPARTMENT
- Log:** Quản lý bộ phận máy móc ASSIGNED_MULTI (Phân công cho 1 người)
2025-12-06 02:35 • ASSIGNED_MULTI
- Log:** Kỹ thuật viên A SUPPORT_REQUEST (Yêu cầu hỗ trợ cần 2 thêm người giúp)
2025-12-06 02:36 • SUPPORT_REQUEST
- Log:** Kỹ thuật viên A ACCEPT_ASSIGNMENT (Người xử lý đã nhận việc)
2025-12-06 03:44 • ACCEPT_ASSIGNMENT

Hình 4.17. Giao diện chi tiết nhiệm vụ

Màn hình này cho phép người xử lý có thể xem chi tiết nhiệm vụ được giao, các bình luận, ghi chú khi xử lý sự cố, và các chức năng khi hoàn thành nhiệm vụ.

4.3.4.3. Giao diện thông báo



Hình 4.18. Giao diện hiển thị thông báo

Màn hình giao diện này hiển thị các thông báo liên quan về sự cố liên quan đến người nhận thông báo.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận đạt được

Trong quá trình thực hiện dự án, tôi đã mở rộng kiến thức về Java, AI, WebSocket và hiểu sâu hơn về các framework Spring Boot, Angular. Đồng thời, tôi nắm vững được luồng hoạt động và quy trình vận hành của hệ thống. Kết quả là hệ thống đã đáp ứng đầy đủ các yêu cầu cốt lõi của bài toán thực tế như:

- Tiếp nhận sự cố nhanh chóng từ nhân viên thông qua giao diện trực quan.
- Tự động phân tích mức độ rủi ro, hỗ trợ nâng cao chất lượng báo cáo.
- Phân công công việc rõ ràng, từ quản lý → bộ phận xử lý → người xử lý.
- Cảnh báo realtime thông qua WebSocket, giúp các bộ phận liên quan nhận thông báo ngay lập tức.
- Theo dõi toàn bộ vòng đời của sự cố, ghi log chi tiết phục vụ quản lý và đánh giá sau xử lý.
- Thống kê, báo cáo, hỗ trợ nhà quản lý đánh giá tình hình và cải thiện quy trình an toàn – vận hành.

2. Hướng phát triển

Dù đã đáp ứng được các chức năng cốt lõi, hệ thống vẫn còn nhiều tiềm năng mở rộng trong tương lai. Một số định hướng phát triển bao gồm:

- Tích hợp trí tuệ nhân tạo (AI): Gợi ý bộ phận xử lý phù hợp dựa trên dữ liệu lịch sử, phân tích dashboard dự đoán nguy cơ theo thời gian để bảo trì.
- Tích hợp IoT và cảm biến cảnh báo: Nhận dữ liệu từ cảm biến: nhiệt độ, độ ẩm, báo cháy, rung chấn... Tự động tạo sự cố khi phát hiện bất thường. Ứng dụng trong nhà máy, khu công nghiệp, hạ tầng kỹ thuật.
- Xây dựng ứng dụng Mobile App: Nhận thông báo đẩy (push notification). Chụp hình, quay video và gửi sự cố trực tiếp. Nhận việc và cập nhật tiến độ nhanh hơn.

TÀI LIỆU THAM KHẢO

- [1] Aven, T. (2015), *"Risk Analysis"*, Wiley.
- [2] Microsoft (2023), *"Introduction to Incident Management"*.
- [3] Mozilla Developer Network (MDN) (2024), *"WebSocket API Documentation"*.
- [4] VMware (2024), *"Spring Boot Reference Documentation"*.
- Spring AI Team (2024), *"Spring AI Documentation – Building AI-powered Applications"*.
- [5] EVN – Tập đoàn Điện lực Việt Nam (2022), *Báo cáo giám sát vận hành lưới điện*.
- [6] Google Developers (2024), *"Angular Documentation"*.
- [7] SweetAlert2 (2024), *"Documentation"*.
- [8] AdminLTE (2024), *"AdminLTE 4 Documentation"*.
- [9] PostgreSQL Global Development Group (2024), *PostgreSQL 17 Documentation*.
- [10] JetBrains (2023), *"IntelliJ IDEA Documentation"*.
- [11] Microsoft (2024), *"Visual Studio Code Documentation"*.
- [12] Devlin, J. et al. (2019), *"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"*.
- [13] ISO (2018), *"ISO 31000:2018 – Risk Management — Guidelines"*.