

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM**

**HÀ THANH HUY – 17520568  
NGUYỄN LƯƠNG TRƯỜNG VĨ - 17521258**

**KHÓA LUẬN TỐT NGHIỆP  
PHẦN MỀM GIÁM SÁT LỖI ÚNG DỤNG  
ERROR MORNITORING APPLICATION**

**KỸ SƯ NGÀNH CÔNG NGHỆ PHẦN MỀM**

**TP. HỒ CHÍ MINH, 2021**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM**

**HÀ THANH HUY – 17520568  
NGUYỄN LƯƠNG TRƯỜNG VĨ - 17521258**

**KHÓA LUẬN TỐT NGHIỆP  
PHẦN MỀM GIÁM SÁT LỖI ÚNG DỤNG  
ERROR MORNITORING APPLICATION**

**KỸ SƯ NGÀNH CÔNG NGHỆ PHẦN MỀM**

**GIẢNG VIÊN HƯỚNG DẪN  
Th.S THÁI THỤY HÀN UYỄN**

**TP. HỒ CHÍ MINH, 2021**

## **THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP**

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số ..... ngày ..... của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

- 1.Chủ tịch.....
- 2.Thư ký.....
- 3.Ủy viên.....
- 4.Ủy viên.....

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày 15 tháng 12 năm 2021

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP**

**(CỦA CÁN BỘ HƯỚNG DẪN)**

**Tên khóa luận:**

**PHẦN MỀM GIÁM SÁT LỖI ỨNG DỤNG**

**Nhóm SV thực hiện:**

Nguyễn Lương Trường Vĩ      17521258  
Hà Thanh Huy                  17520568

**Cán bộ hướng dẫn**

Th.S Thái Thụy Hàn Uyên

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

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

Một số nhận xét về hình thức cuốn báo cáo:

.....  
.....  
.....

2. Về nội dung nghiên cứu:

.....

.....  
.....

3. Về chương trình ứng dụng:

.....  
.....  
.....

4. Về thái độ làm việc của sinh viên:

.....  
.....  
.....

**Đánh giá chung:**

.....  
.....  
.....

**Điểm từng sinh viên:**

Nguyễn Lương Trường Vĩ: ...../10

Hà Thanh Huy : ...../10

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC

Độc Lập - Tự Do - Hạnh Phúc

CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày 15 tháng 12 năm 2021

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP**

**(CỦA CÁN BỘ PHẢN BIỆN)**

**Tên khóa luận:**

**PHẦN MỀM GIÁM SÁT LỖI ỨNG DỤNG**

**Nhóm SV thực hiện:**

Nguyễn Lương Trường Vĩ      17521258  
Hà Thanh Huy                  17520568

**Cán bộ hướng dẫn**

Th.S Thái Thụy Hàn Uyên

**Đánh giá Khóa luận**

5. Về cuốn báo cáo:

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

Một số nhận xét về hình thức cuốn báo cáo:

.....  
.....  
.....

6. Về nội dung nghiên cứu:

.....

.....  
.....

7. Về chương trình ứng dụng:

.....  
.....  
.....

8. Về thái độ làm việc của sinh viên:

.....  
.....  
.....

**Đánh giá chung:**

.....  
.....  
.....

**Điểm từng sinh viên:**

Nguyễn Lương Trường Vĩ: ...../10

Hà Thanh Huy : ...../10

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

## **LỜI CẢM ƠN**

Trước hết, chúng em xin gửi lời cảm ơn đến ban giám hiệu, quý thầy cô của trường Đại học Công nghệ Thông tin, đặc biệt là các thầy cô trong khoa Công Nghệ Phần Mềm đã cung cấp cho chúng em những hành trang kiến thức cần thiết, bổ ích và phong phú trong những năm học tập tại trường.

Chúng em cũng xin bày tỏ lời cảm ơn chân thành đến cô Thái Thụy Hàn Uyên - người đã dành nhiều thời gian hướng dẫn và giúp đỡ chúng em trong suốt quá trình làm khóa luận để chúng em có thể hoàn thành tốt khóa luận tốt nghiệp này.

Chúng em đã nỗ lực rất nhiều để làm khóa luận hoàn thiện nhất có thể nhưng chắc chắn khó tránh khỏi những thiếu sót. Nên chúng em kính mong quý thầy cô thông cảm và tận tình chỉ bảo để chúng em cải thiện tốt hơn.

Sau cùng, chúng em kính chúc quý thầy cô thật nhiều sức khỏe và niềm tin để thực hiện sứ mệnh cao đẹp của mình và truyền đạt kiến thức cho các thế hệ sau này.

Chúng em trân trọng cảm ơn!

**Nhóm sinh viên**

Hà Thanh Huy

Nguyễn Lương Trường Vũ

## ĐỀ CƯƠNG CHI TIẾT

**TÊN ĐỀ TÀI:** Phần mềm giám sát lỗi ứng dụng

**Cán bộ hướng dẫn:** ThS. Thái Thụy Hàn Uyên

**Thời gian thực hiện:** Từ ngày 06/09/2021 đến ngày 01/01/2021

**Sinh viên thực hiện:**

Nguyễn Lương Trường Vĩ – 17521258

Hà Thanh Huy – 17520568

**Nội dung đề tài:**

### 1. Giới thiệu đề tài

Với công nghệ ngày càng hiện đại đòi hỏi càng nhiều sản phẩm công nghệ ra đời trong đó có ứng dụng web. Càng nhiều sản phẩm đòi hỏi người dùng càng có nhiều lựa chọn. Những sản phẩm càng ổn định thì sẽ thu hút được người dùng nhiều hơn. Vì vậy với việc phát triển một công cụ giám sát sẽ cho phép được nhà phát triển phần mềm vận hành sản phẩm một cách tối ưu nhất mang lại trải nghiệm tốt nhất cho khách hàng của mình.

Khi có hệ thống giám sát lỗi nhà phát triển sẽ tiết kiệm rất nhiều chi phí và thời gian để bảo trì sản phẩm. Phát hiện và khắc phục những lỗi ảnh hưởng nghiêm trọng đến dự án một cách kịp thời mang lại trải nghiệm tốt cho khách hàng của mình.

Nhận thấy vấn đề đó nhóm đã lên ý tưởng và thực hiện. Phần mềm chạy trên nền tảng web nơi mọi người có thể truy cập và tiện dụng, sử dụng được trên hầu hết các thiết bị.

### 2. Mục tiêu

- Tìm hiểu cách tích hợp thư viện thứ 3 vào ứng dụng của nhà phát triển để khai thác thông tin về các vấn đề xảy ra trong quá trình vận hành.
- Tìm hiểu công nghệ lập trình ReactJs, GoLang, Cơ sở dữ liệu MongoDB.
- Xây dựng phần mềm hỗ trợ thu thập, tra cứu, thông báo thông tin cho nhà phát triển mỗi khi ứng dụng của nhà phát triển phát sinh vấn đề.
- Hỗ trợ nhà phát triển phần mềm tra cứu các tình trạng kiểm thử tự động.

### 3. Phạm vi

- Tìm hiểu các công nghệ để lấy dữ liệu của ứng dụng từ thư viện.
- Tìm hiểu công nghệ quản lý lỗi.
- Nghiên cứu thuật toán phân chia công việc cho nhân viên.
- Xây dựng hệ thống giám sát lỗi cho các ứng dụng website viết bằng ReactJs.

### 4. Đối tượng nghiên cứu

- Nghiên cứu các công nghệ lấy thông tin lỗi.
- Nghiên cứu công nghệ lập trình web.
- Nghiên cứu quy trình quản lý lỗi.
- Nghiên cứu công nghệ tích hợp package vào ứng dụng.
- Nghiên cứu Framework, ứng dụng cần giám sát lỗi.

### 5. Các ứng dụng liên quan

- Raygun, Sentry: Nhận đầy đủ thông tin chi tiết chẩn đoán cho mỗi lần xảy ra lỗi hoặc sự cố, lọc các lỗi theo ngày, giờ, phiên bản, thẻ, máy chủ lưu trữ, hệ điều hành, ...

- Bugsnag, Airbrake: Thông báo lỗi theo thời gian thực, tích hợp với các ứng dụng trò chuyện khác nhau, chẩn đoán lỗi cụ thể cho các ứng dụng JavaScript, hỗ trợ tìm chi tiết và vị trí lỗi phát sinh trong mã nguồn.
- Trackjs, Logrocket, Catchjs: Hiển thị tất cả các sự kiện ứng dụng và hành động của người dùng dẫn đến lỗi, nhóm các lỗi với các quy tắc.
- Rollbar: Tích hợp với GitHub, Bitbucket và GitLab, cảnh báo tức thì thông qua các công cụ phổ biến như Slack, Pagerduty và các công cụ khác.
- Errorception: Nhóm thông minh, không thay đổi mã nguồn, tích hợp với các ứng dụng khác, tự động bỏ qua lỗi.

## 6. Ý nghĩa thực tiễn

- Tạo ra hướng mới trong phát triển sản phẩm và bảo trì thay thế cho quy trình cũ đã lỗi thời và tốn nhiều chi phí, thời gian.
- Cho phép nhà phát triển truy vết nhanh các vấn đề xảy ra trong mã nguồn để nhằm nhanh chóng khắc phục mang lại trải nghiệm tốt cho người dùng.
- Giúp doanh nghiệp giảm thiểu thời gian, chi phí thực hiện kiểm tra đầu cuối và đưa vào sử dụng thực tế.
- Thống kê các loại lỗi, tác nhân gây lỗi, số lượng lỗi theo tháng, quý, năm, ... Từ đó giúp nhà phát triển nắm bắt được tình trạng và định hướng xử lý khắc phục sản phẩm.
- Nâng cao năng suất nhóm.

## 7. Tính mới

- Hỗ trợ quản lý và theo dõi công việc.
- Hỗ trợ theo dõi quá trình Automation Test.
- Hỗ trợ đề xuất người xử lý công việc tự động.

- Gợi ý phương pháp giải quyết lỗi.
- Hỗ trợ thiết lập tự động kiểm tra sản phẩm khi nâng cấp phiên bản.

## 8. Phương pháp thực hiện

- Xây dựng website cho người dùng truy cập để giám sát ứng dụng.
- Xây dựng backend để lấy dữ liệu thông tin lỗi từ package và trả thông tin lỗi. cho người dùng cho website.
- Xây dựng cơ sở dữ liệu lưu thông tin lỗi.
- Tích hợp các kênh thông báo, kênh quản lý task như Slack, Trello, ...
- Xây dựng package tích hợp vào ứng dụng cần giám sát.

## 9. Kết quả mong đợi

- Hoàn thành các tính năng cơ bản của phần mềm giám sát lỗi.
- Tích hợp được vào các ứng dụng website ReactJs đang vận hành.
- Trả ra lỗi đúng chính xác, nhanh.

### Kế hoạch thực hiện:

Giai đoạn	Thời gian	Nội dung	Phân công
Giai đoạn 1: Thu thập yêu cầu	6/9/2021 - 15/9/2021	Thu thập các yêu cầu từ các nhà phát triển phần mềm và các lập trình viên. Tham khảo các ứng dụng về quản lý lỗi.	Huy, Vĩ
Giai đoạn 2: Phân tích hệ thống	15/9/2021- 1/10/2021	Viết tài liệu yêu cầu cho sản phẩm. Phân tích các chức năng chính của sản phẩm Tham khảo giao diện.	Huy, Vĩ

Giai đoạn 3: Tìm hiểu công nghệ và phát triển ứng dụng	1/10/2021 - 1/12/2021	<p>Front-end:</p> <ul style="list-style-type: none"> <li>- Tìm hiểu ReactJs, NextJs.</li> <li>- Tìm hiểu thư viện hỗ trợ UI Ant design.</li> </ul> <p>Back-end:</p> <ul style="list-style-type: none"> <li>- Tìm hiểu về Go Lang</li> <li>- Cách tính hợp các API bên thứ 3: Trello, Slack</li> </ul> <p>Database:</p> <ul style="list-style-type: none"> <li>- MongoDB</li> </ul> <p>Xây dựng phần mềm hoàn chỉnh dựa trên yêu cầu đã phân tích:</p> <ul style="list-style-type: none"> <li>- Xây dựng module User</li> <li>- Xây dựng module Project</li> <li>- Xây dựng module Issue</li> <li>- Xây dựng module Automation test</li> </ul>	<ul style="list-style-type: none"> <li>- Xây dựng Database: Huy, Vĩ</li> <li>- Xây dựng Front-End: Vĩ</li> <li>- Xây dựng Back-End: Huy</li> </ul>
Giai đoạn 4: Kiểm thử, sửa lỗi, chạy thử sản phẩm	1/12/2021 - 15/12/2022	<ul style="list-style-type: none"> <li>- Thực hiện kiểm thử các chức năng, luồng hoạt động của ứng dụng.</li> <li>- Kiểm tra hệ thống thu thập lỗi có chính xác không.</li> <li>- Sửa các lỗi của phần mềm trong quá trình kiểm thử.</li> <li>- Chạy thử sản phẩm trên môi trường thực tế.</li> </ul>	Huy, Vĩ
Giai đoạn 5:	15/12/2021- 1/1/2021	Hoàn thiện báo cáo, chuẩn bị phản biện và bảo vệ khóa luận.	Huy, Vĩ

Hoàn thiện sản phẩm và báo cáo		
--------------------------------	--	--

**Xác nhận của CBHD**  
(Ký tên và ghi rõ họ tên)

**TP. HCM, ngày 24 tháng 08 năm 2021**  
**Sinh viên 1**  
(Ký tên và ghi rõ họ tên)  
**Nguyễn Lương Trường Vĩ**

**Sinh viên 2**  
(Ký tên và ghi rõ họ tên)  
**Hà Thanh Huy**

## MỤC LỤC

<b>Chương 1. MỞ ĐẦU .....</b>	1
<b>1.1. Lý do chọn đề tài.....</b>	1
<b>1.2. Mục tiêu và phạm vi .....</b>	1
<b>1.2.1. Mục tiêu .....</b>	1
<b>1.2.2. Phạm vi .....</b>	1
<b>1.3. Đối tượng nghiên cứu .....</b>	2
<b>1.4. Kết quả mong đợi .....</b>	2
<b>Chương 2. TỔNG QUAN .....</b>	3
<b>2.1. Phân tích các ứng dụng liên quan .....</b>	3
<b>2.1.1. Raygun .....</b>	3
<b>2.1.2. Sentry .....</b>	4
<b>2.1.3. Bugsnag.....</b>	6
<b>2.1.4. Airbrake.....</b>	7
<b>2.1.5. RollBar .....</b>	7
<b>2.1.6. TrackJs .....</b>	8
<b>2.1.7. LogRocket.....</b>	9
<b>2.1.8. Errorception .....</b>	11
<b>2.1.9. CatchJs.....</b>	12
<b>2.1.10. Firebase Crashlytics .....</b>	13
<b>2.2. Tổng hợp.....</b>	14
<b>Chương 3. CÁC CÔNG NGHỆ SỬ DỤNG .....</b>	16
<b>3.1. Front end .....</b>	16
<b>3.1.1. Ngôn ngữ.....</b>	16
<b>3.1.2. Library và Framework.....</b>	23
<b>3.2. Back end .....</b>	26

<b>3.2.1. Golang</b> .....	26
<b>3.3. Database</b> .....	29
<b>3.3.1. Giới thiệu NoSQL</b> .....	29
<b>3.3.2. Vì sao nên dùng cơ sở dữ liệu NoSQL?</b> .....	29
<b>3.3.3. Các loại cơ sở dữ liệu NoSQL?</b> .....	30
<b>3.3.4. Giới thiệu MongoDB</b> .....	31
<b>3.4. Server</b> .....	33
<b>3.4.1. Vercel</b> .....	33
<b>3.4.2. MongoDB Atlas</b> .....	34
<b>3.5. Các công cụ hỗ trợ</b> .....	35
<b>3.5.1. Visual Studio Code</b> .....	35
<b>3.5.2. MongoDB Compass</b> .....	36
<b>3.5.3. Goland</b> .....	37
<b>3.5.4. Heroku</b> .....	38
<b>3.5.5. Git</b> .....	41
<b>3.5.6. Source tree</b> .....	43
<b>Chương 4. PHÂN TÍCH, THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG</b> .....	45
<b>4.1. Đặt vấn đề</b> .....	45
<b>4.1.1. Thực trạng ứng dụng xảy ra lỗi khi người dùng thao tác</b> .....	45
<b>4.1.2. Khó khăn khi sửa lỗi ứng dụng</b> .....	45
<b>4.1.3. Tối ưu quy trình quản lý lỗi</b> .....	46
<b>4.2. Phương pháp</b> .....	46
<b>4.2.1. Xây dựng Package nhúng vào code của ứng dụng cần giám sát lỗi</b> ....	46
<b>4.2.2. Tối ưu hóa thời gian xử lý lỗi với giao diện Kanban Board</b> .....	47
<b>4.2.3. Tích hợp các ứng dụng bên thứ 3 hỗ trợ thông báo: Trello, Slack</b> .....	48
<b>4.3. Đặc tả yêu cầu</b> .....	49
<b>4.3.1. Đặc tả tổng quát</b> .....	49
<b>4.3.2. Danh sách các Actor</b> .....	49
<b>4.3.3. Danh sách các Use-case</b> .....	50

<b>4.4. Đặc tả Use-case.....</b>	<b>51</b>
<b>4.4.1. Đặc tả Use-case “Bảo mật hệ thống”.....</b>	<b>51</b>
<b>4.4.2. Đặc tả Use-case “Quản lý dự án cần giám sát”.....</b>	<b>55</b>
<b>4.4.3. Đặc tả Use-case “Quản lý lỗi của dự án” .....</b>	<b>59</b>
<b>4.4.4. Đặc tả Use-case “Quản lý Automation test case” .....</b>	<b>63</b>
<b>4.4.5. Đặc tả Use-case “Tích hợp tự động” .....</b>	<b>65</b>
<b>4.5. Đặc tả Squence Diagram.....</b>	<b>67</b>
<b>4.5.1. Sơ đồ Seq Diagram “Đăng ký” .....</b>	<b>67</b>
<b>4.5.2. Sơ đồ Seq Diagram “Đăng nhập”.....</b>	<b>68</b>
<b>4.5.3. Sơ đồ Seq Diagram “Thiết lập lại mật khẩu” .....</b>	<b>69</b>
<b>4.5.4. Sơ đồ Seq Diagram “Thêm mới dự án” .....</b>	<b>70</b>
<b>4.5.5. Sơ đồ Seq Diagram “Thêm mới thành viên vào dự án” .....</b>	<b>71</b>
<b>4.5.6. Sơ đồ Seq Diagram “Tạm dừng giám sát dự án” .....</b>	<b>72</b>
<b>4.5.7. Sơ đồ Seq Diagram “Thay đổi vai trò thành viên trong dự án” .....</b>	<b>73</b>
<b>4.5.8. Sơ đồ Seq Diagram “Thêm lỗi” .....</b>	<b>74</b>
<b>4.5.9. Sơ đồ Seq Diagram “Giao sửa lỗi” .....</b>	<b>74</b>
<b>4.5.10. Sơ đồ Seq Diagram “Cập nhật lỗi” .....</b>	<b>75</b>
<b>4.5.11. Sơ đồ Seq Diagram “Thêm automation test” .....</b>	<b>75</b>
<b>4.5.12. Sơ đồ Seq Diagram “Tích hợp thông báo Trello, Slack, Gmail .....</b>	<b>76</b>
<b>4.5.13. Sơ đồ Seq Diagram “Tự động đề xuất người xử lý” .....</b>	<b>77</b>
<b>4.6. Sơ đồ lớp (Class diagram).....</b>	<b>78</b>
<b>4.7. Cơ sở dữ liệu (NoSql) .....</b>	<b>79</b>
<b>4.7.1. Cơ sở dữ liệu tổng quát .....</b>	<b>79</b>
<b>4.7.2. Bảng “users”.....</b>	<b>80</b>
<b>4.7.3. Bảng “projects” .....</b>	<b>80</b>
<b>4.7.4. Bảng “suites” .....</b>	<b>81</b>
<b>4.7.5. Bảng “issues” .....</b>	<b>82</b>
<b>4.8. Thiết kế User-flow .....</b>	<b>84</b>
<b>4.8.1. Security .....</b>	<b>84</b>

<b>4.8.2. Project manager .....</b>	<b>85</b>
<b>4.8.3. Issues manager .....</b>	<b>86</b>
<b>4.8.4. Automation test manager .....</b>	<b>87</b>
<b>4.9. Giao diện.....</b>	<b>88</b>
<b>4.9.1. Giao diện trang giới thiệu .....</b>	<b>88</b>
<b>4.9.2. Giao diện trang đăng ký .....</b>	<b>88</b>
<b>4.9.3. Giao diện trang quên mật khẩu.....</b>	<b>89</b>
<b>4.9.4. Giao diện trang đăng nhập .....</b>	<b>90</b>
<b>4.9.5. Giao diện trang cập nhật thông tin lần đầu .....</b>	<b>90</b>
<b>4.9.6. Giao diện trang dự án.....</b>	<b>92</b>
<b>4.9.7. Giao diện trang danh sách lỗi .....</b>	<b>96</b>
<b>4.9.8. Giao diện trang danh sách automation test.....</b>	<b>97</b>
<b>4.9.9. Giao diện trang tài liệu hướng dẫn .....</b>	<b>98</b>
<b>Chương 5. ÁP DỤNG THUẬT TOÁN ĐỂ XUẤT NGƯỜI PHỤ TRÁCH LỖI.....</b>	<b>99</b>
<b>5.1. Đặt vấn đề.....</b>	<b>99</b>
<b>5.2. Giải pháp .....</b>	<b>99</b>
<b>5.2.1. Tính toán.....</b>	<b>99</b>
<b>5.2.2. Đưa ra gợi ý .....</b>	<b>102</b>
<b>Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>103</b>
<b>6.1. Kết quả đạt được .....</b>	<b>103</b>
<b>6.2. Hạn chế và khó khăn .....</b>	<b>103</b>
<b>6.3. Hướng phát triển .....</b>	<b>103</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>104</b>

## DANH MỤC HÌNH

Hình 2.1: Ứng dụng Raygun .....	3
Hình 2.2: Ứng dụng Sentry .....	5
Hình 2.3: Ứng dụng Bugsnag.....	6
Hình 2.4: Ứng dụng Airbrake .....	7
Hình 2.5: Ứng dụng RollBar.....	8
Hình 2.6: Ứng dụng TrackJs .....	9
Hình 2.7: Ứng dụng LogRocket.....	10
Hình 2.8: Ứng dụng Errorception .....	11
Hình 2.9: Ứng dụng CatchJs .....	12
Hình 2.10: Ứng dụng Firebase Crashlytics .....	14
Hình 3.1: Ngôn ngữ JavaScript.....	16
Hình 3.2: Ngôn ngữ Html .....	18
Hình 3.3: Ngôn ngữ Css.....	21
Hình 3.4: Thư viện ReactJs .....	23
Hình 3.5: Framework NextJs .....	25
Hình 3.6: MongoDB.....	31
Hình 3.7: Server Vercel.....	33
Hình 3.8: MongoDB Atlas .....	34
Hình 3.9: Visual Studio Code .....	35
Hình 3.10: MongoDB Compass .....	36
Hình 3.11: Source tree.....	44
Hình 4.1: Sơ đồ Use-case tổng quát.....	49
Hình 4.2: Đặc tả Use-case “Bảo mật hệ thống” .....	51
Hình 4.3: Đặc tả Use-case “Quản lý dự án cần giám sát” .....	55
Hình 4.4: Đặc tả Use-case “Quản lý lỗi của dự án” .....	59
Hình 4.5: Đặc tả Use-case “Quản lý Automation test case” .....	63

Hình 4.6: Đặc tả Use-case “Tích hợp tự động” .....	65
Hình 4.7: Sơ đồ Seq Diagram “Đăng ký” .....	67
Hình 4.8: Sơ đồ Seq Diagram “Đăng nhập” .....	68
Hình 4.9: Sơ đồ Seq Diagram “Thiết lập lại mật khẩu”.....	69
Hình 4.10: Sơ đồ Seq Diagram “Thêm mới dự án” .....	70
Hình 4.11: Sơ đồ Seq Diagram “Thêm mới thành viên vào dự án” .....	71
Hình 4.12: Sơ đồ seq Diagram “Tạm dừng giám sát dự án” .....	72
Hình 4.13: Sơ đồ Seq Diagram “Thay đổi vai trò thành viên trong dự án”.....	73
Hình 4.14: Sơ đồ Seq Diagram “Thêm lỗi” .....	74
Hình 4.15: Sơ đồ Seq Diagram “Giao sửa lỗi” .....	74
Hình 4.16: Sơ đồ Seq Diagram “Cập nhật lỗi” .....	75
Hình 4.17: Sơ đồ Seq Diagram “Thêm Automation test”.....	75
Hình 4.18: Sơ đồ Seq Diagram “Tích hợp thông báo Trello, Slack, Gmail”.....	76
Hình 4.19: Sơ đồ Seq Diagram “Tự động đề xuất người xử lý” .....	77
Hình 4.20: Sơ đồ lớp (Class diagram).....	79
Hình 4.21: Sơ đồ cơ sở dữ liệu tổng quan.....	79
Hình 4.22: User-flow Security .....	84
Hình 4.23: User-flow Project manager .....	85
Hình 4.24: User-flow Issues manager .....	86
Hình 4.25: User-flow Automation test manager .....	87
Hình 4.26: Giao diện trang giới thiệu .....	88
Hình 4.27: Giao diện trang đăng ký .....	88
Hình 4.28: Giao diện trang nhập email cần khôi phục lại mật khẩu.....	89
Hình 4.29: Giao diện trang “Nhập mật khẩu mới” .....	89
Hình 4.30: Giao diện bước “Cập nhật thông tin cá nhân” .....	90
Hình 4.31: Giao diện bước “Tạo mới dự án cần giám sát” .....	91
Hình 4.32: Giao diện bước “Thông báo tạo mới và cập nhật thành công” .....	91
Hình 4.33: Giao diện trang dự án.....	92

Hình 4.34: Giao diện trang chi tiết dự án.....	92
Hình 4.35: Giao diện trang cài đặt chung dự án .....	93
Hình 4.36: Giao diện trang cập nhật thành viên dự án .....	93
Hình 4.37: Giao diện trang bật tắt tính năng đề xuất người xử lý .....	94
Hình 4.38: Giao diện trang tích hợp thông báo trello .....	94
Hình 4.39: Giao diện trang tích hợp thông báo Slack.....	95
Hình 4.40: Giao diện trang tích hợp thông báo Email .....	95
Hình 4.41: Giao diện trang danh sách lỗi.....	96
Hình 4.42: Giao diện trang chi tiết lỗi .....	96
Hình 4.43: Giao diện trang danh sách automation test .....	97
Hình 4.44: Giao diện trang chi tiết automation test .....	97
Hình 4.45: Giao diện trang tài liệu hướng dẫn.....	98

## **DANH MỤC BẢNG**

Bảng 1: Danh sách các Actor.....	49
Bảng 2: Danh sách các Use-case.....	50
Bảng 3: Use-case “Bảo mật hệ thống” .....	51
Bảng 4: Use-case “Quản lý dự án cần giám sát” .....	55
Bảng 5: Use-case “Quản lý lỗi của dự án”.....	59
Bảng 6: Use-case “Quản lý Automation test case”.....	64
Bảng 7: Use-case “Tích hợp tự động” .....	66
Bảng 8: Cơ sở dữ liệu Tổng quan .....	80
Bảng 9: Bảng “users” .....	80
Bảng 10: Bảng “projects” .....	81
Bảng 11: Bảng “suites” .....	81
Bảng 12: Bảng “issues” .....	82

## **DANH MỤC TỪ VIẾT TẮT**

Từ viết tắt	Ý nghĩa
UI	User Interface – Giao diện người dùng
FE	Front end – Lập trình phía giao diện
BE	Back end – Lập trình phía server
SQL	Structured Query Language – Ngôn ngữ truy vấn
DB	Database – Cơ sở dữ liệu
PM	Project Manager – Người quản lý dự án
DEV	Developer – Lập trình viên
PE	Project Editor – Người chỉnh sửa dự án
PA	Project Admin – Người quản trị dự án

# **Chương 1. MỞ ĐẦU**

## **1.1. Lý do chọn đề tài**

Với công nghệ ngày càng hiện đại đòi hỏi càng nhiều sản phẩm công nghệ ra đời trong đó có ứng dụng web. Càng nhiều sản phẩm đòi hỏi người dùng càng có nhiều lựa chọn, những sản phẩm càng ổn định, ít lỗi thì sẽ thu hút được người dùng nhiều hơn. Vì vậy với việc phát triển một công cụ giám sát lỗi sẽ cho phép được nhà phát triển phần mềm vận hành sản phẩm một cách tối ưu nhất mang lại trải nghiệm tốt nhất cho khách hàng của mình.

Khi có hệ thống giám sát lỗi nhà phát triển sẽ tiết kiệm rất nhiều chi phí và thời gian để bảo trì sản phẩm. Phát hiện và khắc phục những lỗi ảnh hưởng nghiêm trọng đến dự án một cách kịp thời mang lại trải nghiệm tốt cho khách hàng của mình.

Nhận thấy vấn đề đó nhóm đã lên ý tưởng và thực hiện đề tài “Phần mềm giám sát lỗi ứng dụng”. Phần mềm chạy trên nền tảng website nơi mọi người có thể truy cập và tiện dụng, sử dụng hầu hết trên các thiết bị.

## **1.2. Mục tiêu và phạm vi**

### **1.2.1. Mục tiêu**

- Tìm hiểu cách tích hợp thư viện thứ 3 vào ứng dụng của nhà phát triển để khai thác thông tin về các vấn đề xảy ra trong quá trình vận hành.
- Tìm hiểu công nghệ lập trình ReactJs, GoLang, Cơ sở dữ liệu MongoDB.
- Xây dựng phần mềm hỗ trợ thu thập, tra cứu, thông báo thông tin cho nhà phát triển mỗi khi ứng dụng của nhà phát triển phát sinh vấn đề.
- Hỗ trợ nhà phát triển phần mềm tra cứu các tình trạng kiểm thử tự động.

### **1.2.2. Phạm vi**

- Tìm hiểu các công nghệ để lấy dữ liệu của ứng dụng từ thư viện.
- Tìm hiểu công nghệ quản lý lỗi.
- Nghiên cứu thuật toán phân chia công việc cho nhân viên.

- Xây dựng hệ thống giám sát lỗi cho các ứng dụng website viết bằng ReactJs.

### **1.3. Đối tượng nghiên cứu**

- Nghiên cứu các công nghệ lấy thông tin lỗi.
- Nghiên cứu công nghệ lập trình web.
- Nghiên cứu quy trình quản lý lỗi.
- Nghiên cứu công nghệ tích hợp package vào ứng dụng.
- Nghiên cứu Framework, ứng dụng cần giám sát lỗi.

### **1.4. Kết quả mong đợi**

- Hoàn thành các tính năng cơ bản của phần mềm giám sát lỗi.
- Tích hợp được vào các ứng dụng website ReactJs đang vận hành.
- Trả ra lỗi đúng chính xác, nhanh.

## Chương 2. TỔNG QUAN

### 2.1. Phân tích các ứng dụng liên quan

#### 2.1.1. Raygun

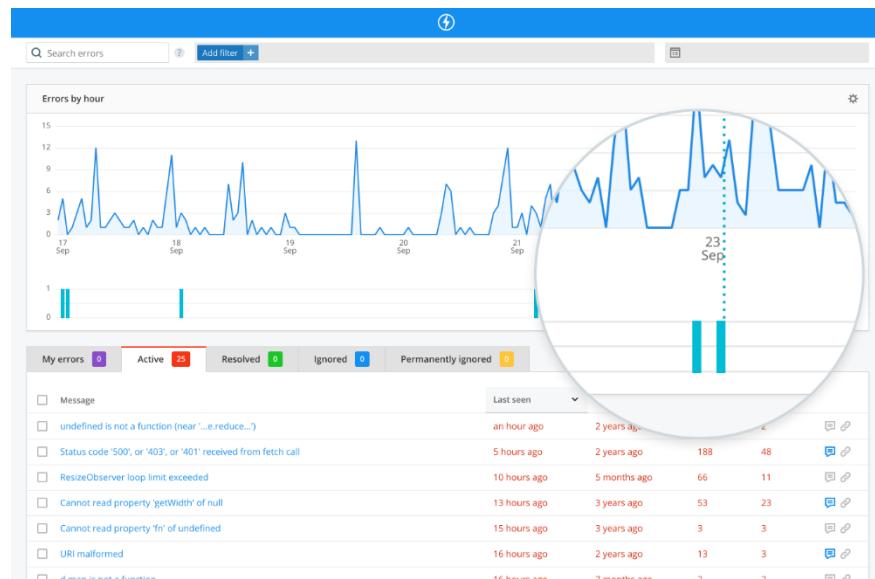
##### *Mô tả chung*

Giám sát lỗi Raygun là một công cụ mạnh mẽ, cung cấp thông tin chi tiết về các lỗi và sự cố ảnh hưởng đến người dùng.

Raygun giám sát toàn bộ hệ thống người dùng trong thời gian thực, từ phía máy khách và phía máy chủ, đến máy tính để bàn và thiết bị di động.

Raygun sẽ phát hiện và hiển thị mọi lỗi, cùng với ai bị ảnh hưởng, nơi họ gặp sự cố, cũng như những thông tin chi tiết về người dùng thực khác.

##### *Hình ảnh*



Hình 2.1: Ứng dụng Raygun

##### *Các chức năng*

- Xem vị trí chính xác gây ra lỗi.
- Theo dõi việc triển khai để xác định điều gì đã gây ra sự gia tăng đột biến hoặc giảm số lượng lỗi.

- Lọc các lỗi theo ngày, giờ, phiên bản, thẻ, máy chủ lưu trữ, hệ điều hành, trình duyệt, thẻ tùy chỉnh, v.v.
- Giảm nhiễu với các bộ lọc có thể định cấu hình cho tên máy, phiên bản, địa chỉ IP, tên máy chủ, v.v.
- Nhóm các lỗi do người dùng bị ảnh hưởng.
- Lưu giữ dữ liệu 180 ngày.
- Hỗ trợ cho tất cả các ngôn ngữ.

### **Nhận xét**

Raygun là một giải pháp giám sát lỗi hiện đại dành cho các nhà phát triển muốn có được khả năng hiển thị đầy đủ về các lỗi trên hệ thống công nghệ của họ, đi sâu vào nguyên nhân gốc rễ của lỗi và tiết kiệm thời gian và tiền bạc. Raygun là một lựa chọn mạnh mẽ và giá cả phải chăng cho các nhóm phần mềm.

#### **2.1.2. Sentry**

##### **Mô tả chung**

Sentry là một trong những phần mềm lớn nhất trong thị trường theo dõi lỗi. Là một giải pháp giám sát lỗi được thiết lập tốt, họ đã xây dựng một sản phẩm ổn định với nhiều tính năng và chức năng. Các nhà phát triển sử dụng Sentry để xem dấu vết lỗi, ngữ cảnh người dùng, dữ liệu phát hành và các sự kiện trước đó.

## Hình ảnh

The screenshot shows the Sentry interface for an application error. At the top, it displays user information (DC, dchow@goldenmothchemical.com, ID: 7036), browser details (Chrome Version: 80.0.3987), and system details (Mac OS X Version: 10.15.4). Below this is a section titled 'TAGS' containing various metadata tags such as browser, device, os, and url. The main area is titled 'EXCEPTION (most recent call first)' and shows a 'ReferenceError' where 'getCardInfo' is not defined. The error occurred in a file named 'Form.jsx' at line 13:4. The code snippet shows the following lines:

```
JS ./app/components/Form.jsx in onSubmit at line 13:4
09.     let { cart } = this.form;
10.
11.     cart = {
12.       ...cart,
13.       creditCard: getCardInfo()
14.     };
15.
16.     App.confirmPurchase({ cart }).then(({ invoice }) => {
17.       this.onPurchaseComplete({ invoice });
18.     }
19.   }
```

Below the code, there are two more sections of the stack trace: 'Checkout.jsx' at line 126:9 and 'EmpowerPlant.jsx' at line 94:40.

Hình 2.2: Ứng dụng Sentry

### Các chức năng

- Tùy chỉnh câu truy vấn.
- Có trang tổng quan.
- Truy tìm phân tán.
- Gợi ý các vấn đề liên quan đến ứng dụng.

### Nhận xét

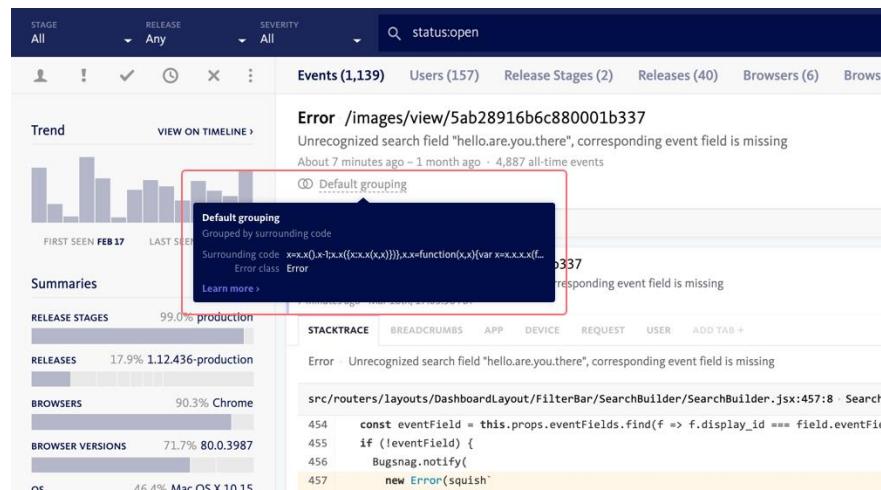
Sentry Là một công cụ theo dõi lỗi được thiết lập tốt, Sentry có một loạt chức năng ấn tượng nhưng lại có mức giá cao hơn. Tuy nhiên, Sentry sẽ chỉ cung cấp khả năng hiển thị các lỗi và sự cố mà không có thông tin ngữ cảnh lớn hơn về người dùng thực và trải nghiệm kỹ thuật số của họ.

### 2.1.3. Bugsnag

#### *Mô tả chung*

Bugsnag là một nền tảng SaaS tuyệt vời, tự động phát hiện sự cố trong các ứng dụng được phát triển bằng Ruby, Python, PHP, Java, Android, iOS, Node.js hoặc Unity (bao gồm cả Android và iOS).

#### *Hình ảnh*



Hình 2.3: Ứng dụng Bugsnag

#### *Các chức năng*

- Dễ dàng kiểm soát khi được thông báo về các trường hợp xảy ra lỗi
- Lọc các lỗi ứng dụng một cách dễ dàng bằng cách sử dụng tìm kiếm toàn văn.
- Theo dõi thời gian thực
- Theo dõi hệ thống liên tục.
- Nhóm Các lỗi được theo cách thông minh nhất có thể.

#### *Nhận xét*

Bugsnag là một lựa chọn tốt để theo dõi lỗi. Với tính năng giám sát toàn diện từ đầu đến cuối, tạo quy trình làm việc mạnh mẽ giúp ưu tiên, chẩn đoán và sửa lỗi trong phần mềm của mình.

## 2.1.4. Airbrake

### *Mô tả chung*

Airbrake cung cấp các dấu vết chi tiết giúp sửa lỗi và cho phép đi sâu vào các trường hợp ngoại lệ định kỳ và lọc theo tham số, người dùng hoặc biến môi trường. Tính năng nhóm thông minh và khả năng phát hiện trùng lặp.

### *Hình ảnh*



Hình 2.4: Ứng dụng Airbrake

### *Các chức năng*

- Thông báo thời gian thực.
- Định nghĩa ngữ cảnh lỗi.
- Thông tin chi tiết lỗi.

### *Nhận xét*

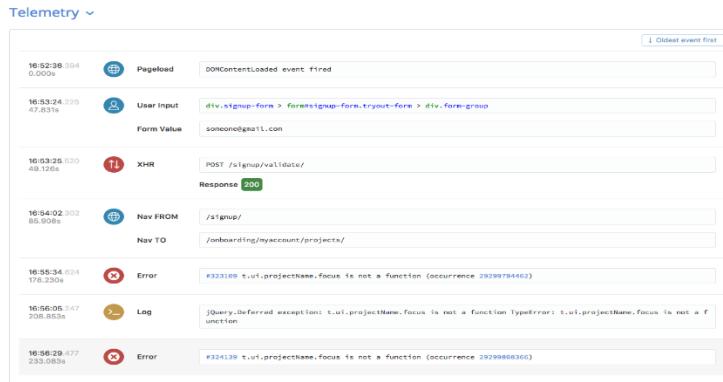
Airbrake, giống như Raygun, cho biết ai đang gặp lỗi và chính xác những gì họ đang làm dẫn đến khi nó xảy ra. Công cụ này cực kỳ hữu ích cho những người đang tìm cách chủ động hỗ trợ khách hàng gặp sự cố và lỗi trong ứng dụng. Giao diện người dùng hơi lỗi thời, nhưng chức năng được cung cấp bởi công cụ này rất mạnh mẽ khi đã quen với giao diện.

## 2.1.5. RollBar

### *Mô tả chung*

Rollbar được tự mô tả là một nền tảng cải tiến code liên tục. Chúng cho phép các nhà phát triển chủ động phát hiện, dự đoán và khắc phục lỗi với quy trình làm việc được hỗ trợ bởi AI theo thời gian thực.

## Hình ảnh



Hình 2.5: Ứng dụng RollBar

### Các chức năng

- Hiển thị trong thời gian thực.
- Chủ động phân loại lỗi.
- Phân tích nguyên nhân gốc rễ.
- Bảo mật mạnh mẽ.
- Hỗ trợ nền tảng rộng.

### Nhận xét

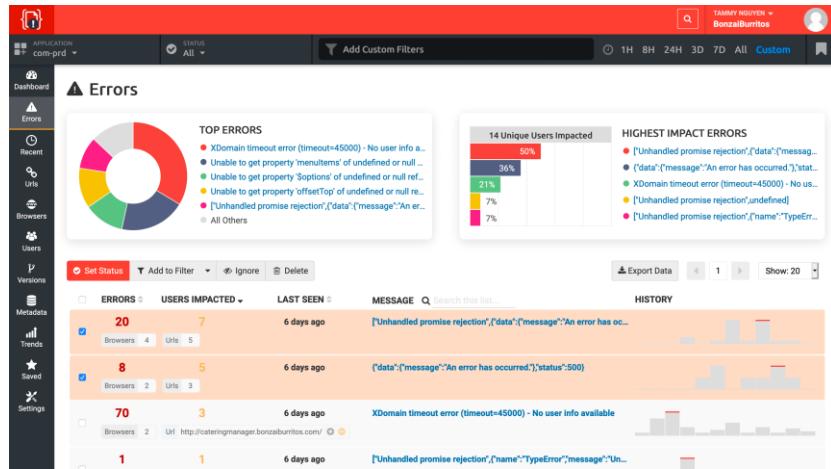
Rollbar là một giải pháp hiệu quả, giá cả phải chăng, trực quan và dễ sử dụng. Sự kết hợp của một giao diện người dùng đơn giản, cùng với khả năng xem tất cả các ứng dụng và dịch vụ được báo cáo trên một bảng điều khiển dễ sử dụng làm cho nó trở thành một công cụ mạnh mẽ. Tuy nhiên, vấn đề chính xung quanh Rollbar là chức năng nhóm kém, có thể là một điểm mâu thuẫn khó hiểu.

## 2.1.6. TrackJs

### Mô tả chung

Trackjs sử dụng Error Tracking để theo dõi và ghi lại các lỗi JavaScript trên các trang web và ứng dụng, ghi lại Telemetry về ứng dụng, mạng, bảng điều khiển và người có thể dễ dàng hiểu và tạo lại các lỗi.

## Hình ảnh



Hình 2.6: Ứng dụng TrackJs

### Các chức năng

- Nhóm lỗi theo quy tắc.
- Bảng điều khiển siêu dữ liệu.
- Báo cáo tác động lỗi.
- Xu hướng và phân tích lỗi.

### Nhận xét

Trackjs là một tùy chọn dễ sử dụng nhưng đắt tiền để báo cáo lỗi JavaScript. Một trong những điểm nổi bật của công cụ này là cách họ định giá theo lưu lượng truy cập trang web, không phải theo lỗi, nghĩa là sẽ trả nhiều tiền hơn khi quy mô khách truy cập và không bị phạt vì trải nghiệm lỗi.

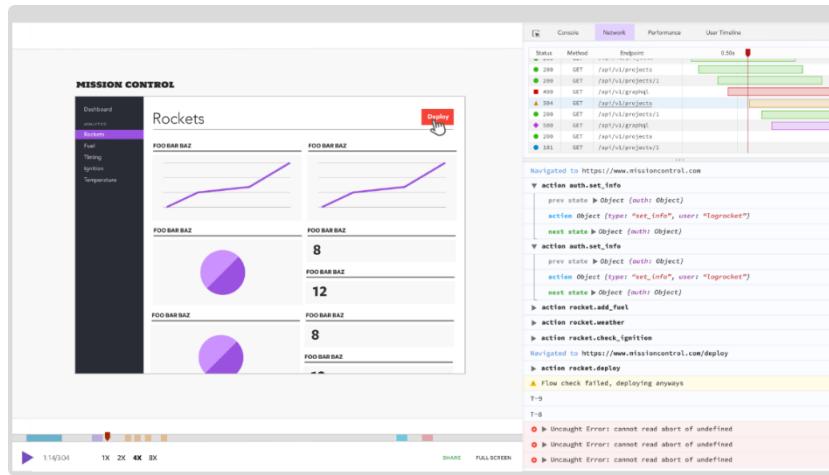
## 2.1.7. LogRocket

### Mô tả chung

LogRocket là một giải pháp giám sát giao diện người dùng kết hợp phát lại, giám sát hiệu suất và phân tích sản phẩm. Theo dõi các lỗi Javascript, lỗi mạng và các

vấn đề về hiệu suất, xem lại dấu vết hệ thống cùng với video phát lại mọi vấn đề và loại bỏ chi tiết không cần thiết để tìm và phân loại các vấn đề có mức độ ưu tiên cao nhất.

## Hình ảnh



Hình 2.7: Ứng dụng LogRocket

## Các chức năng

- Giám sát theo phiên đăng nhập.
- Truy vết hệ thống.
- Bộ lọc nâng cao.

## Nhận xét

Phiên bản miễn phí của LogRocket là người bạn tốt nhất của người khởi nghiệp. Tuy nhiên, việc mở rộng quy mô trong giá cả có thể là một bước ngoặt. Một trong những công cụ hữu ích nhất mà LogRocket cung cấp là khả năng xem và tương tác với các phiên trực tiếp trước đây của người dùng. Khả năng hiển thị lỗi bằng điều khiển giúp việc xác định và sửa lỗi giao diện người dùng trở nên cực kỳ dễ

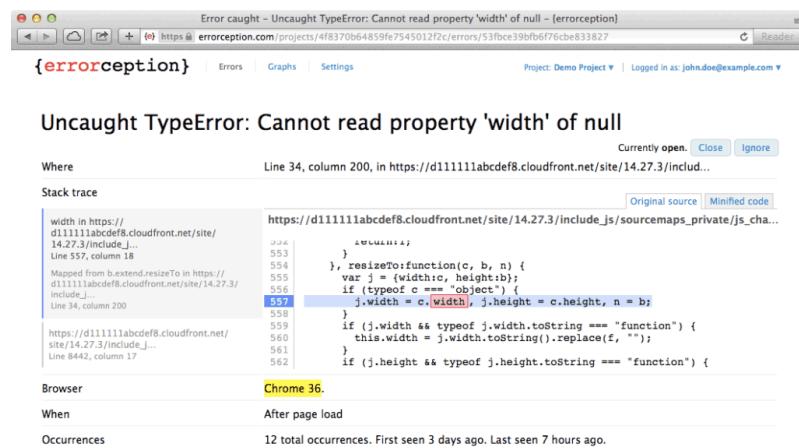
dàng. Nếu có đủ ngân sách, LogRocket có thể là một bổ sung mạnh mẽ cho ngăn xếp công cụ.

## 2.1.8. Errorception

### Mô tả chung

Errorception là một công cụ do một người tên là Rakesh Pai tạo ra và nó giúp tìm ra các lỗi JavaScript và nhóm các lỗi lại với nhau. Vì là công cụ do một nhà phát triển duy nhất tạo ra, nên công cụ này không có nhiều chức năng như những công cụ khác trong danh sách này. Thay vào đó, nó cung cấp một giải pháp giám sát lỗi JavaScript cơ bản mà không gây ra các ảnh hưởng tiêu cực đến hệ thống.

### Hình ảnh



Hình 2.8: Ứng dụng Errorception

### Các chức năng

- Nhóm thông minh.
- Không có thay đổi code.
- Tự động bỏ qua lỗi.

## Nhận xét

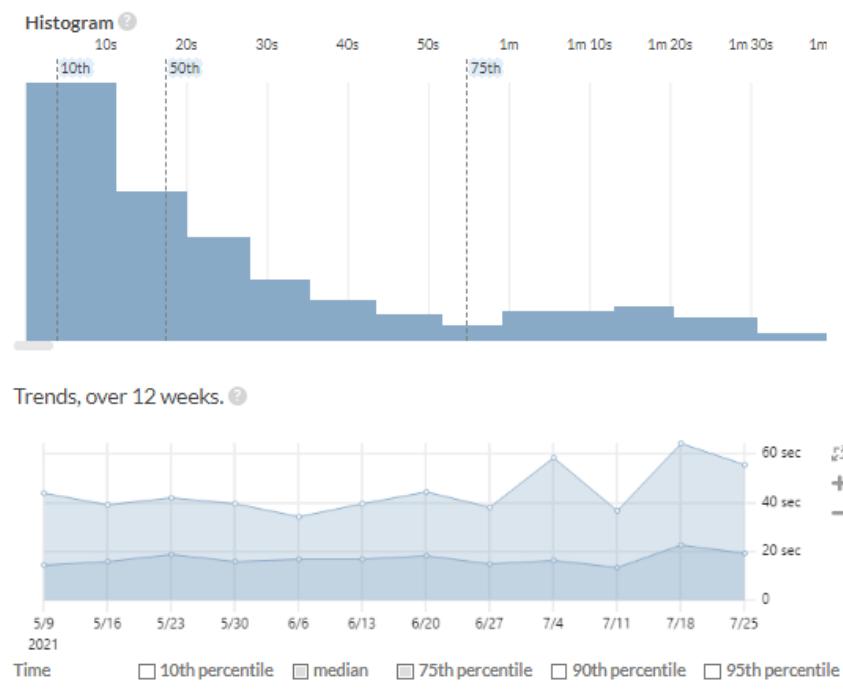
Errorception là một ứng viên thú vị trong danh sách này. Nó rất cơ bản và dễ sử dụng, nhưng do một nhà phát triển duy nhất đang chạy nó. Nếu đang tìm kiếm một giải pháp theo dõi lỗi JavaScript đơn giản và không muốn trả tiền cho tất cả các thông báo mà các công cụ khác trong danh sách này cung cấp, thì Errorception có thể là lựa chọn thích hợp.

### 2.1.9. CatchJs

#### Mô tả chung

CatchJS là một trình theo dõi lỗi JavaScript để cảnh báo khi có bất kỳ sự cố nào xảy ra. Ghi lại bất kỳ lỗi nào gặp phải trên trang web và xem nó ngay lập tức trong nhật ký lỗi. CatchJS cho phép phát lại các lần nhấp dẫn đến lỗi, với ảnh chụp màn hình trực tiếp từ thiết bị của người dùng.

#### Hình ảnh



Hình 2.9: Ứng dụng CatchJs

### **Các chức năng**

- Cài đặt nhỏ gọn (chỉ 1,77KB gzipped).
- Toàn cảnh lỗi.
- Ảnh chụp màn hình của khách hàng.
- Bảo vệ quyền riêng tư của người dùng.
- Những thông báo thư điện tử.

### **Nhận xét**

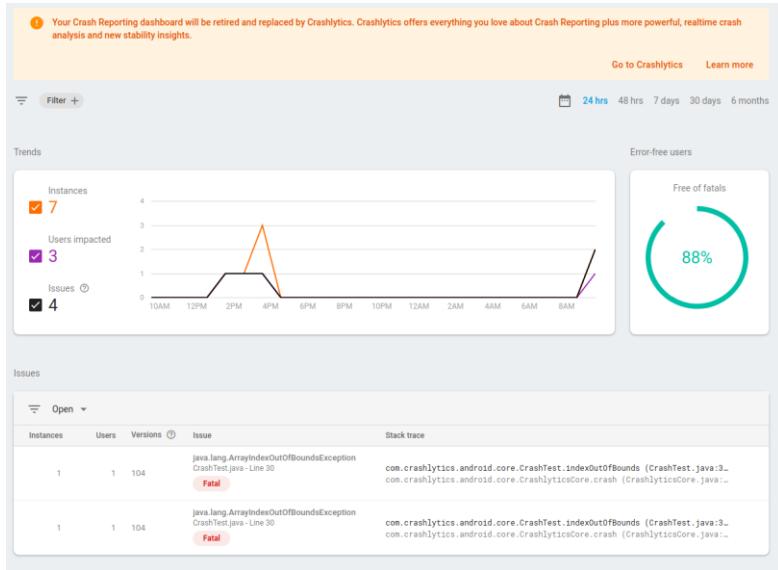
CatchJS là công cụ ghi lỗi cuối cùng dành riêng cho JavaScript. Một trong những điều ấn tượng nhất về CatchJS là khả năng chụp ảnh màn hình của người dùng thực khi họ gặp sự cố. Tuy nhiên, nó bị giới hạn ở việc chỉ cung cấp giải pháp end-to-end cho Windows, khiến nó kém linh hoạt hơn so với các công cụ khác.

## **2.1.10. Firebase Crashlytics**

### **Mô tả chung**

Firebase là một nền tảng do Google phát triển để tạo các ứng dụng web và di động. Crashlytics là công cụ báo cáo sự cố miễn phí nằm trong bộ Firebase. Có thể sử dụng các sản phẩm Firebase cùng nhau để giải quyết những thách thức phức tạp và tối ưu hóa trải nghiệm ứng dụng, cá nhân hóa quy trình giới thiệu, tăng mức độ tương tác của người dùng hoặc thêm chức năng mới.

## Hình ảnh



Hình 2.10: Ứng dụng Firebase Crashlytics

## Các chức năng

- A/B Testing.
- Phân tích lỗi.
- Lập chỉ mục ứng dụng.
- Cơ sở dữ liệu thời gian thực.

## Nhận xét

Firebase cung cấp một loạt các dịch vụ hoạt động song song với nhau, giúp đơn giản hóa việc phát triển ứng dụng nói chung. Tuy nhiên, điều này có nghĩa là chức năng theo dõi lỗi của họ chỉ là một trong nhiều chức năng mà họ làm, có nghĩa là nó không phải là ưu tiên chính như các đối thủ cạnh tranh khác.

## 2.2. Tổng hợp

Sau khi sử dụng, phân tích các ứng dụng giám sát lỗi hiện có thì hầu hết các ứng dụng đều được phát triển bởi các lập trình viên nước ngoài. Đa số các ứng dụng đã phát triển

rất lâu, thu thập được nhiều vấn đề xảy ra khi lập trình viên phát triển ứng dụng, nên hầu hết đều hỗ trợ nhiều tính năng cho các nhà phát triển về việc giám sát, kiểm tra, phân tích, gom nhóm lỗi, cho phép tích hợp nhiều công cụ thứ 3. Nhiều ứng dụng có giao diện bắt mắt dễ dùng, dễ tương tác, mang lại các tiện ích giúp kiểm soát lỗi dễ dàng và nhanh hơn. Nhưng nhìn chung vẫn còn khá phức tạp để cho một người mới bắt đầu sử dụng, để triển khai nhanh một công cụ giám sát người dùng cần phải mất thời gian để nghiên cứu bên cạnh đó đối với những công cụ giám sát lỗi tốt nhiều chức năng thì lại có mức chi phí rất cao, còn đối với những công cụ với mức chi phí thấp thì ít tính năng hơn, có nhiều tính năng không phù hợp với nhu cầu giám sát lỗi của nhà phát triển.

Từ những gì đã phân tích được từ các ứng dụng đi trước, nhóm đưa ra những tính năng cần làm cho đề tài khóa luận như sau:

- Nhận dạng lỗi trùng lặp
- Tích hợp kênh giao tiếp Trello, Slack
- Tích hợp gửi mail khi ứng dụng phát sinh lỗi
- Cung cấp Karban dasboard cho phép thành viên thao tác quản lý lỗi dễ dàng, trực quan.
- Cung cấp tính năng quản lý thành viên trong dự án
- Quản lý công việc cho thành viên
- Gợi ý giao task cho thành viên
- Hiển thị biểu đồ tổng quan của mỗi dự án
- Tích hợp theo dõi automation test
- Cung cấp các tính năng quản lý dự án

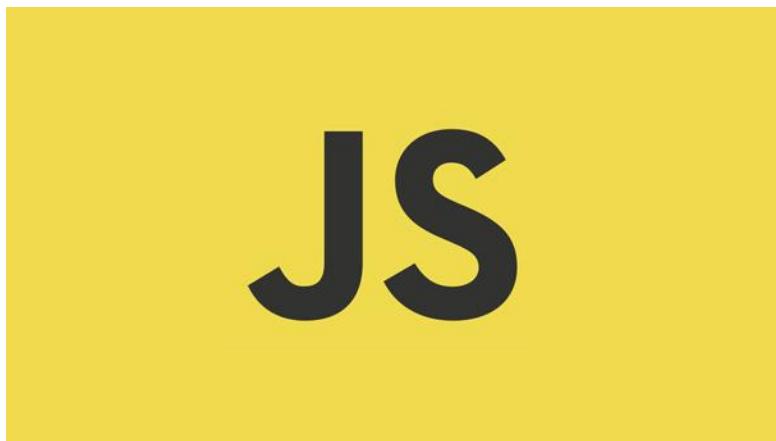
## Chương 3. CÁC CÔNG NGHỆ SỬ DỤNG

### 3.1. Front end

#### 3.1.1. Ngôn ngữ

Về phía front end nhóm sử dụng ngôn ngữ lập trình chính là Javascript. Ngoài ra nhóm còn sử dụng ngôn ngữ HTML, CSS để xây dựng giao diện cho ứng dụng.

*JavaScript*



Hình 3.1: Ngôn ngữ JavaScript

JavaScript là ngôn ngữ lập trình được tạo bởi Brandan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi trở thành tên JavaScript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.

Trong năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lục đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web.

### **Ưu điểm:**

- Không cần một compiler vì web browser có thể biên dịch nó bằng HTML
- Dễ học hơn các ngôn ngữ lập trình khác
- Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn
- Giúp website tương tác tốt hơn với khách truy cập
- Nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác

### **Nhược điểm:**

- Dễ bị khai thác
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng
- Nhiều khi không được hỗ trợ trên mọi trình duyệt
- Js code snippets lớn

### **Các thuật ngữ thông dụng:**

- Không đồng bộ (Asynchronous): Mã không đồng bộ khi bạn bắt đầu một cái gì đó, hãy quên nó đi và khi kết quả sẵn sàng, bạn lấy lại nó mà không cần phải đợi nó.
- Khối (Block): Trong JavaScript, một khối được phân tách bằng dấu ngoặc nhọn ({}). Một câu lệnh chứa một khối, một vòng lặp chứa một khối.

- Xác định phạm vi (Scope): Với chức năng Xác định phạm vi, bất kỳ biến nào được xác định trong một khối đều có thể nhìn thấy và truy cập được từ bên trong toàn bộ khối, nhưng không nằm ngoài nó.
- Gọi lại (Callback): Gọi lại là một hàm được gọi khi có điều gì đó xảy ra. Sự kiện nhấp chuột được liên kết với một phần tử có chức năng gọi lại được gọi khi người dùng nhấp vào phần tử. Một yêu cầu tìm nạp có một cuộc gọi lại được gọi khi tài nguyên được tải xuống.
- Lexical Scoping: là một loại phạm vi cụ thể có nghĩa là giá trị của một biến được xác định theo vị trí của nó khi nó được viết. Không phải khi nó được gọi, đó là điều gì đó xảy ra với phương án thay thế, phạm vi động (được sử dụng trong một số ngôn ngữ lập trình khác).
- Pure function: Một hàm không có tác dụng phụ (không sửa đổi các tài nguyên bên ngoài) và đầu ra của nó chỉ được xác định bởi các đối số. Bạn có thể gọi hàm này 1 triệu lần và cho cùng một tập hợp các đối số, kết quả đầu ra sẽ luôn giống nhau.

### *Html*



Hình 3.2: Ngôn ngữ Html

HTML là chữ viết tắt của Hypertext Markup Language. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, v.v.

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản `<p>` và `</p>`

### **Ưu điểm:**

- Ngôn ngữ được sử dụng rộng lớn này có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
- Sử dụng mượt mà trên hầu hết mọi trình duyệt.
- Có quá trình học đơn giản và trực tiếp.
- Mã nguồn mở và hoàn toàn miễn phí.
- Markup gọn gàng và đồng nhất.
- Chuẩn chính của web được vận hành bởi World Wide Web Consortium (W3C).
- Dễ dàng tích hợp với các ngôn ngữ backend như PHP và Node.js.

### **Nhược điểm:**

- Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động, bạn cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.

- Nó có thể thực thi một số logic nhất định cho người dùng. Vì vậy, hầu hết các trang đều cần được tạo riêng biệt, kể cả khi nó sử dụng cùng các yếu tố, như là headers hay footers.
- Một số trình duyệt chậm hỗ trợ tính năng mới.
- Khó kiểm soát cảnh thực thi của trình duyệt (ví dụ, những trình duyệt cũ không render được tag mới)

### **Các thuật ngữ thông dụng:**

- **Elements:** Các element là các chỉ định xác định cấu trúc và nội dung của các đối tượng trong một trang. Các element được xác định bằng cách sử dụng dấu ngoặc nhỏ hơn và lớn hơn < > bao quanh tên element.
- **Tags:** Việc sử dụng các dấu ngoặc nhỏ hơn và lớn hơn bao quanh một element sẽ tạo ra cái được gọi là thẻ. Thẻ thường xảy ra nhất trong các cặp thẻ mở và đóng.
- **Attributes:** Các attribute là các thuộc tính được sử dụng để cung cấp thông tin bổ sung về một element. Các thuộc tính phổ biến nhất bao gồm thuộc tính id, xác định một element; các thuộc tính class, phân loại một element; thuộc tính src, trong đó xác định một nguồn cho nội dung nhúng; và thuộc tính href, cung cấp một tham chiếu hyperlink đến một tài nguyên được liên kết (Vd: <a href="http://shayhowe.com/">Shay Howe</a>).

## Css



Hình 3.3: Ngôn ngữ Css

CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

### **Ưu điểm:**

- Tiết kiệm thời gian: CSS khi sử dụng có thể viết lần đầu, đồng thời sử dụng lại trong các trang HTML tiếp theo. Không chỉ vậy, chúng ta hoàn toàn có thể xác

định được một kiểu cho từng thành phần HTML và cũng áp dụng được cho nhiều trang web khác khi cần thiết.

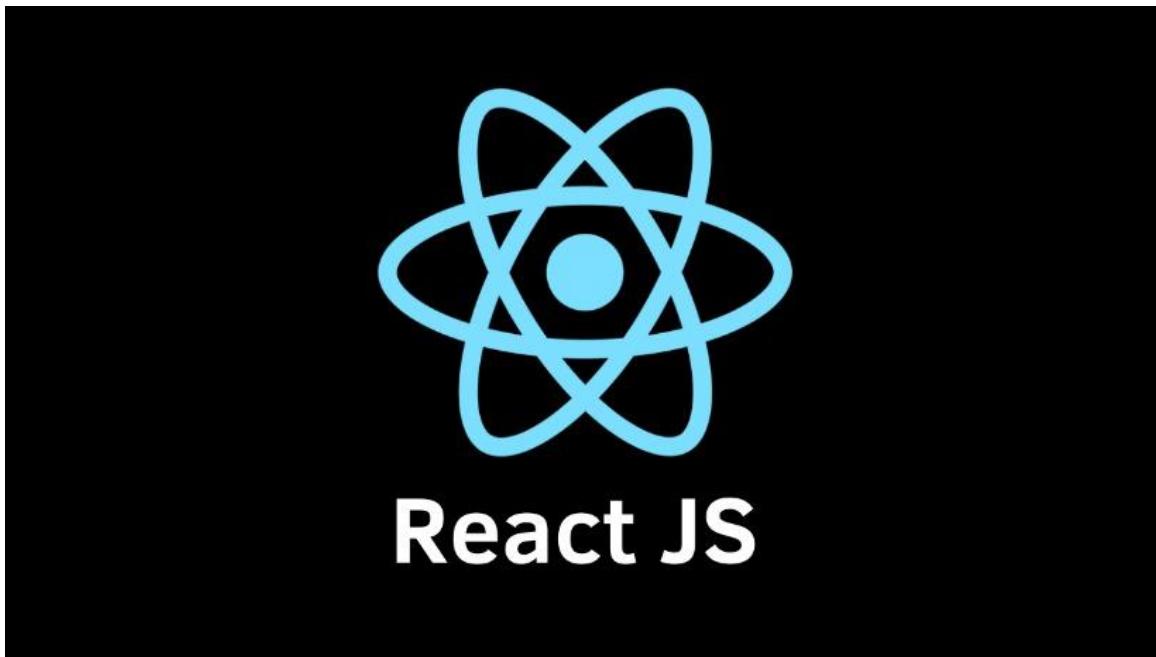
- Tải trang nhanh chóng: Đặc trưng của CSS khi sử dụng là ít mã hơn. Bởi thế mà thời gian tải xuống nhanh chóng, giúp tiết kiệm thời gian đáng kể
- Dễ dàng khi thực hiện bảo trì: Khi cần chúng ta chỉ cần thay đổi một kiểu và mọi thành phần trong tất cả những website thì lúc này CSS sẽ hỗ trợ cập nhật hoàn toàn tự động.

### Các thuật ngữ thông dụng:

- Inline css: Nhúng trực tiếp vào tài liệu HTML thông qua cặp <></>
- Internal css: Dùng thẻ <style> bên trong thẻ <head> của HTML để tạo ra nơi viết mã CSS.
- External css: Tạo một tập tin .css riêng và nhúng vào tài liệu HTML thông qua cặp thẻ <link>.
- Selectors: Khi các yếu tố được thêm vào một trang web, chúng có thể được tạo kiểu bằng CSS. Selector chỉ định chính xác thành phần nào trong HTML để nhắm mục tiêu và áp dụng các kiểu (như màu sắc, kích thước và vị trí) vào.
- Properties: Khi một phần tử được chọn, một thuộc tính sẽ xác định các kiểu sẽ được áp dụng cho phần tử đó. Tên thuộc tính nằm sau selector, trong dấu ngoặc nhọn {} và ngay trước dấu hai chấm, .: Có rất nhiều thuộc tính chúng ta có thể sử dụng, chẳng hạn như background, color, font-size, height, và width, và các thuộc tính mới thường được thêm vào.
- Values: Cho đến nay, chúng tôi đã chọn một element với selector và xác định kiểu chúng tôi muốn áp dụng với property.

### 3.1.2. Library và Framework

#### *ReactJs*



Hình 3.4: Thư viện ReactJs

Là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web.

Components của công cụ này được phát triển bởi Facebook. Nó được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất thời bấy giờ.

Nó cũng sử dụng khái niệm là Virtual DOM (DOM ảo). Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là Reactjs chỉ render những component hay subcomponent nào thực sự thay đổi.

## **Tại sao dùng ReactJs:**

- Dễ sử dụng
- Hỗ trợ Reusable component trong Javascript
- Viết component dễ dàng hơn
- Hiệu suất tốt hơn với Virtual DOM
- Thân thiện với SEO

## **Ưu điểm:**

- Viết bằng reactjs dễ dàng chia và sử dụng lại component
- Code được webpack mã hóa, up lên google chrome không sợ bị dịch ngược ...
- Hàng trăm ngàn thư viện hỗ trợ code các module chức năng.
- Reactjs có Dom ảo, giúp cải thiện hiệu suất làm việc rất nhiều, khi có tính toán cần thay đổi hoặc cần cập nhật những gì lên Dom thì ReactJS đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên Dom, làm như vậy sẽ giúp cho ReactJS tránh được những thao tác cần có trên một Dom mà không cần tốn thêm bất cứ chi phí nào.
- Source code extension có thể build thành web luôn khi bạn không muốn nó là 1 Extension đơn thuần.

## **Nhược điểm:**

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax

- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nề so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
- Khó tiếp cận cho người mới học Web

### **NextJs**



Hình 3.5: Framework NextJs

React là một thư viện của javascript để xây dựng Frontend. React có rất nhiều ưu điểm từ việc chia nhỏ các thành phần của web ra nhiều phần khác nhau, modules hoá css, xử lý sự kiện... Thế nhưng dường như react chỉ phù hợp để render data phía client, việc này ảnh hưởng rất lớn đến SEO của website khi mà hầu hết những data mà google đánh index đều phải được render ở server.

Các nhà phát triển của react đã đưa ra một công cụ để hỗ trợ việc render phía server cho react, đó là Next-Js.

Next.js là một framework dùng để phát triển các ứng dụng React theo tư tưởng Isomorphic (Universal ), được phát triển bởi Zeit. Nhờ Next.js chúng ta có thể dễ dàng

tạo được một ứng dụng React theo tư tưởng Isomorphic, có bao gồm chức năng Server Side Redering.

### 3.2. Back end

#### 3.2.1. Golang

Go hay còn gọi là Golang là ngôn ngữ lập trình mã nguồn mở, được thiết kế tại Google bởi Robert Griesemer, Rob Pike, and Ken Thompson.

Cú pháp của ngôn ngữ lập trình Golang (syntax) khá tinh gọn, mặc dù thật sự có những cú pháp mà người mới nhìn vào thật sự hơi khó chịu.

Golang giống như C++ hay Java, nó cũng là một ngôn ngữ dùng để lập trình. Tuy nhiên điểm khiến ngôn ngữ Go trở nên khác biệt nằm ở sự đơn giản của nó. Nó sở hữu những cú pháp khá tinh gọn. Điều này có thể khiến những người mới học thấy khó khăn. Dù vậy cú pháp của Go có độ tương đồng rất lớn với C++. Vì thế nếu bạn đã quen thuộc với C++ thì việc học Golang không còn là điều khó khăn.

#### Các khái niệm quan trọng:

##### 1. Goroutine

Goroutine là một hàm có thể chạy đồng thời với các hàm khác.

Goroutines là những luồng gọn nhẹ, được khởi tạo với chỉ 2KB trong stack size có thể tăng hoặc giảm vùng nhớ tùy yêu cầu sử dụng.

Những ứng dụng Go có thể có rất nhiều Goroutines chạy đồng thời với nhau.

Cơ chế của goroutine khá là đơn giản: 1 function tồn tại một cách đa luồng với các goroutine khác trên cùng một không gian bộ nhớ, Go có bộ điều khiển quản lý các goroutine rồi phân phối chúng vào các bộ xử lý logic và gắn mỗi bộ xử lý logic này với một thread hệ thống được tạo ra trước đó để thực thi các goroutine này. Nói cách khác, mỗi thread hệ thống sẽ xử lý một nhóm goroutine được điều phối thông qua bộ

xử lý logic. Với bộ điều khiển quản lý tác vụ đồng thời và cơ chế bộ xử lý logic, những cái khó khăn, phức tạp khi khai báo thread Go đã xử lý hết giúp chúng ta rồi.

Để khởi tạo một goroutine ta chỉ cần thêm phía trước một function call hay method call từ khóa go

## 2. Channel

**Channel** là để các tiến trình bất đồng bộ (trong **Go** hay gọi là **Goroutines**) có thể giao tiếp với nhau mà không cần phải dùng biến từ bên ngoài. Nó không được thiết kế để giải quyết mọi trường hợp của concurrent nhưng nó giúp việc này dễ dàng, an toàn và vui hơn. Thậm chí tới những trường hợp không cần dùng channel nhưng các gopher vẫn dùng channel chỉ vì họ thực sự thích nó.

## 3. Mutex

Mutex là hình thức ngắn gọn để loại trừ lẫn nhau. Mutex được sử dụng khi bạn không muốn một tài nguyên bị truy cập bởi nhiều Routine con trong cùng một thời điểm. Mutex có 2 phương thức Lock và Unlock. Mutex được đặt trong package. Nên chúng ta phải import Package Sync. Các câu lệnh phải được thực hiện riêng rẽ lẫn nhau có thể được đặt trong mutex.Lock() hoặc Mutex.Unlock() do đó tránh được điều kiện Race.

- **Ưu điểm**

- + **Tiện ích:**

Ngôn ngữ Golang sở hữu một tính năng nổi bật và độc quyền mang tên goroutines. Goroutines tồn tại như một công cụ tích cực giúp giải quyết rất nhiều vấn đề còn tồn tại. Nó có thời gian khởi động nhanh hơn threads thông thường. Công cụ này cũng sở hữu đa kênh và có khả năng cho phép sự giao tiếp giữa các kênh này. Ngoài ra, goroutines còn có mutex locking, một tính năng cho phép khóa lại các cấu trúc dữ liệu để việc đọc và ghi nhớ không xảy ra xung đột.

#### + Tốc độ:

Tốc độ nhị phân của ngôn ngữ Golang đương nhiên sẽ có thể chậm hơn C++.

Tuy nhiên hiệu suất làm việc giữa chúng lại không có chênh lệch nhiều. Và Golang cũng nhanh hơn các ngôn ngữ phổ biến khác như Java, Python hay Ruby... Vì thế đây được coi như một điểm cộng của Golang.

#### + Linh hoạt:

Không cần có sự xuất hiện bắt buộc của external dependencies, các tệp dữ liệu executable được tạo bằng toolchain của Golang vẫn có thể tự hoạt động một cách trơn tru. Toolchain của Go cho phép hỗ trợ rất nhiều hệ điều hành cũng như các loại phần cứng khác nhau và cũng được dùng như một công cụ biên dịch hệ nhị phân qua các nền tảng.

Tương thích: Một trong những điều kiện cần thiết ở một ngôn ngữ lập trình là sự tương thích của nó đối với những hệ thống khác nhau. Ngôn ngữ Golang có thể liên kết với thư viện C bên ngoài lẫn hệ thống native bên trong.

#### + Hỗ trợ:

Golang được liên kết với rất nhiều bên thứ ba khác nhau. Từ đó mang đến cho người dùng những trải nghiệm mới lạ và độc đáo.

### o Nhược điểm

+ **Không hỗ trợ công cụ Generics:** Điều này làm người dùng đánh mất đi cơ hội kiến tạo nên những code rõ ràng. Thay vào đó, các mã được tạo ra rất khó hiểu và trừu tượng. Go vẫn đang trong quá trình xem xét trang bị thêm Generics. Điều này có thể không cần thiết với những nhà lập trình lão làng, những người vốn không ngại những đoạn mã khó.

+ **Quy tắc vô cùng nghiêm ngặt:** Đối với nhiều người đây là lợi thế bởi nó giúp hệ thống hóa các quy trình bạn đang thực hiện. Tuy nhiên nó cũng sẽ giới hạn khả

năng sáng tạo của người dùng. Bạn có thể khắc phục bằng cách ghi đè lên những quy tắc này bằng những plugin sáng tạo.

+ **Ít thư viện hỗ trợ:** Điều đó làm hạn chế khả năng lưu trữ của người dùng.

### 3.3. Database

#### 3.3.1. Giới thiệu NoSQL

Cơ sở dữ liệu NoSQL là Cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn

Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

#### 3.3.2. Vì sao nên dùng cơ sở dữ liệu NoSQL?

Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời.

- **Linh hoạt:** Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.
- **Khả năng thay đổi quy mô:** Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém.

- **Hiệu năng cao:** Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.
- **Cực kỳ thiết thực:** Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

### 3.3.3. Các loại cơ sở dữ liệu NoSQL?

- **Khóa–giá trị:** Cơ sở dữ liệu khóa–giá trị có khả năng phân mảnh cao và cho phép thay đổi quy mô theo chiều ngang ở các quy mô lớn mà các loại hình cơ sở dữ liệu khác không thể làm được
- **Tài liệu:** Trong mã ứng dụng, dữ liệu thường được biểu diễn dưới dạng một đối tượng hoặc văn bản dạng JSON vì đây là mô hình dữ liệu hiệu quả và trực quan cho các nhà phát triển. Cơ sở dữ liệu văn bản giúp nhà phát triển dễ dàng lưu trữ và truy vấn dữ liệu trong một cơ sở dữ liệu bằng cách sử dụng cùng một định dạng mô hình văn bản họ sử dụng trong mã ứng dụng của mình.
- **Đồ thị:** Mục đích của cơ sở dữ liệu đồ thị là giúp việc dựng và chạy ứng dụng hoạt động với các bộ dữ liệu có khả năng kết nối cao trở nên dễ dàng. Cơ sở dữ liệu đồ thị thường được sử dụng cho các đồ thị tri thức, mạng xã hội, công cụ đề xuất và phát hiện lừa đảo.
- **Trong bộ nhớ:** Ứng dụng trò chơi và công nghệ quảng cáo đã sử dụng loại Cơ sở dữ liệu này trong nhiều trường hợp như bảng xếp hạng, kho lưu trữ phiên và phân tích thời gian thực yêu cầu thời gian phản hồi đạt mức micro giây và lưu lượng có thể đạt mức đỉnh điểm vào bất kỳ thời điểm nào.
- **Tìm kiếm:** Nhiều ứng dụng xuất nhật ký để giúp nhà phát triển khắc phục sự cố. Amazon Elasticsearch Service (Amazon ES) là Cơ sở dữ liệu được xây dựng chuyên dụng để cung cấp khả năng ảo hóa và phân tích dữ liệu do máy tạo gần như

theo thời gian thực bằng cách tạo chỉ mục, tập hợp và tìm kiếm nhặt ký và số liệu có cấu trúc chưa hoàn chỉnh

### 3.3.4. Giới thiệu MongoDB



Hình 3.6: MongoDB

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

#### Ưu điểm:

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

#### Nhược điểm:

- Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
- Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

### **Các thuật ngữ trong MongoDB:**

- **\_id:** Là trường bắt buộc có trong mỗi document. Trường \_id đại diện cho một giá trị duy nhất trong document MongoDB. Trường \_id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một \_id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.
- **Collection:** Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.
- **Cursor:** Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.
- **Database:** Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều Database.
- **Document:** Một bản ghi thuộc một Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường tên và giá trị.

- **Field:** Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.
- **JSON:** Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.
- **Index:** Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn.

### 3.4. Server

#### 3.4.1. Vercel



Hình 3.7: Server Vercel

Vercel (trước đây là zeit) là một nền tảng đám mây cho các trang web tĩnh và serverless . Nó cho phép các nhà phát triển lưu trữ các trang web và công nghệ Jamstack triển khai ngay lập tức, tự động mở rộng quy mô và không yêu cầu giám sát, tất cả đều không có cấu hình.

#### Ưu điểm:

- Khả năng phát triển đơn giản

- Dễ dàng cài đặt
- Miễn phí hoàn toàn cho người dùng cá nhân
- Miễn phí SSL
- Dễ dàng cài đặt domain
- Xây dựng và đẩy sản phẩm dễ dàng (chỉ cần dùng 1 lệnh duy nhất vercel)
- Có Edge Network (cho phép chọn địa điểm)

### 3.4.2. MongoDB Atlas



Hình 3.8: MongoDB Atlas

Database-as-a-Service (DBaaS) là một dịch vụ cho phép người dùng thiết lập, triển khai và mở rộng cơ sở dữ liệu mà không cần lo lắng về phần cứng vật lý tại chỗ, các bản cập nhật phần mềm và các chi tiết về cấu hình cho hiệu suất. Với DBaaS, một nhà cung cấp dịch vụ đám mây sẽ thực hiện tất cả những điều đó và giúp người dùng thiết lập và chạy ngay lập tức.

MongoDB Atlas là cơ sở dữ liệu đám mây được quản lý hoàn toàn có thể xử lý tất cả sự phức tạp của việc triển khai, quản lý và sửa chữa các triển khai trên nhà cung cấp dịch vụ đám mây mà bạn chọn (AWS, Azure và GCP). MongoDB Atlas là cách tốt nhất để triển khai, chạy và mở rộng MongoDB trên đám mây. Với Atlas, bạn sẽ có cơ sở dữ liệu MongoDB chạy chỉ với vài cú nhấp chuột và chỉ trong vài phút.

### **3.5. Các công cụ hỗ trợ**

#### **3.5.1. Visual Studio Code**



Hình 3.9: Visual Studio Code

Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cài tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

#### **Một số tính năng của Visual studio code:**

- Hỗ trợ nhiều ngôn ngữ lập trình
- Hỗ trợ đa nền tảng

- Cung cấp kho tiện ích mở rộng
- Kho lưu trữ an toàn
- Hỗ trợ web
- Lưu trữ dữ liệu dạng phân cấp
- Hỗ trợ thiết bị đầu cuối
- Màn hình đa nhiệm

### 3.5.2. MongoDB Compass



Hình 3.10: MongoDB Compass

MongoDB Compass là công cụ dành cho người dùng Windows PC và Mac, cung cấp phương thức nhanh và trực quan để chạy các truy vấn, hơn hết là tìm kiếm và phân tích cấu trúc cơ sở dữ liệu.

Dành cho hệ điều hành Linux, tiện tích qMongoFront for Linux cho phép bạn kết nối cơ sở dữ liệu MongoDB, ưu điểm của qMongoFront for Linux là khả năng thêm và

loại bỏ các cơ sở dữ liệu, cũng như xem trạng thái của bất kỳ cơ sở dữ liệu và máy chủ MongoDB.

Mục đích của tiện ích MongoDB Compass là hỗ trợ người dùng tương tác tốt hơn với dữ liệu MongoDB và tối ưu hiệu suất truy vấn với bộ công cụ hiển thị chuyên dụng.

MongoDB Compass yêu cầu người dùng nhập địa chỉ host, số cổng và lựa chọn trong số ba phương pháp chứng thực, Kerberos, LDAP và tổ hợp Username/Password thông thường.

### **Những tính năng chính của tiện ích MongoDB Compass:**

- Chạy các truy vấn
- Tìm kiếm và phân tích cấu trúc cơ sở dữ liệu MongoDB
- Hoạt động trên Windows PC và máy Mac

### **3.5.3. Goland**

Phần mềm GoLand JetBrains là một IDE đa nền tảng được xây dựng đặc biệt cho các nhà phát triển Go. Việc phân tích mã tĩnh kết hợp với thiết kế ergonomic giúp phát triển không chỉ có hiệu quả mà còn là trải nghiệm thú vị. Các công cụ thiết yếu như rest-runner, coverage tool, full-featured debugger và tích hợp kiểm soát phiên bản – không bao gồm plugin.

### **Chức năng chính**

- **Code Insight mạnh mẽ, GoLand giúp bạn dễ dàng đọc, viết và thay đổi mã Go:** Phát hiện lỗi nhanh chóng và đề xuất các bản sửa lỗi, tái cấu trúc nhanh chóng và an toàn với hoàn tác một bước, hoàn thành mã thông minh, phát hiện mã chênh và gợi ý tài liệu giúp tất cả các nhà phát triển Go, từ người mới đến các chuyên gia có kinh nghiệm, tạo ra nhanh chóng, hiệu quả, và mã đáng tin cậy.
- **Hệ thống plugin IntelliJ:** Trong trường hợp bạn cần bất cứ thứ gì ngoài các bộ công cụ đã được xây dựng sẵn, IntelliJ plugin ecosystem sẽ giúp bạn.

- **Run & Debug :** Các công cụ tích hợp mạnh mẽ giúp chạy và gỡ lỗi các ứng dụng của bạn. Bạn có thể viết và gỡ lỗi các bài kiểm tra mà không cần bất kỳ plugin bổ sung hoặc nỗ lực cấu hình nào và kiểm tra các ứng dụng của bạn ngay trong IDE. Công cụ Code Coverage được tích hợp sẵn để đảm bảo rằng các thử nghiệm của bạn không bỏ lỡ bất kỳ điều gì quan trọng. IDE phân tích mã của bạn, tìm kiếm các kết nối giữa các symbol, để cung cấp code completion, điều hướng nhanh, phân tích lỗi thông minh, định dạng và tái cấu trúc.
- **Tích hợp với VCSs – Hỗ trợ ngoại vi cho Git, GitHub và Mercurial:** GoLand cung cấp hỗ trợ ngoại vi cho Git, GitHub và Mercurial. Perforce, ClearCase và những người khác được hỗ trợ bởi các plugin người dùng có thể cài đặt.

## **Khả năng mở rộng**

Hệ sinh thái phong phú của GoLand bao gồm hơn 1000 plugin có thể được sử dụng để điều chỉnh IDE cho phù hợp với nhu cầu cụ thể của bạn. Tùy chỉnh, mở rộng và thay đổi mọi thứ theo sở thích của bạn: từ việc thêm hỗ trợ cho các dự án Angular và Vue.js, hoặc nhận thanh tiến trình Nyan Cat, đến thiết lập một chủ đề IDE hoàn toàn khác.

### **3.5.4. Heroku**

#### **Giới thiệu chung**

Heroku là một nền tảng đám mây cho phép các cá nhân, doanh nghiệp xây dựng, triển khai, quản lý và mở rộng ứng dụng. Lợi thế chính của nền tảng này nằm ở tính linh hoạt và dễ sử dụng, đây là giải pháp giúp các nhà phát triển đưa ứng dụng của họ ra thị trường đơn giản và nhanh chóng nhất.

Được quản lý bởi một công ty cùng tên, Heroku mang lại một trải nghiệm tuyệt vời cho nhà phát triển ứng dụng, khi họ chỉ cần tập trung vào phát triển, nâng cấp sản phẩm cốt lõi của mình mà không bị phân tâm trong việc duy trì hoạt động máy chủ, phần cứng hoặc các cơ sở hạ tầng.

Một số ngôn ngữ lập trình mà Heroku có khả năng hỗ trợ gồm: NodeJS, Ruby, Python, PHP, Java, Scala, Clojure, Go, Kotlin,... Bên cạnh đó, Heroku còn hỗ trợ cả Postgresql database.

## Cách hoạt động

Heroku có cách thức hoạt động khá giống với các dịch vụ git (github, gitlab). Để sử dụng nền tảng này, bạn cần phải đăng ký một tài khoản, sau đó tạo các git repository (heroku gọi là app) và dùng các git command để đẩy code lên. Điểm khác biệt của Heroku so với các dịch vụ git ở chỗ với mỗi git repository trên Heroku bạn sẽ có một domain dạng <https://ten-repository.herokuapp.com> – đây là nơi show demo cho repository của bạn.

Ngoài ra, bạn cũng có thể sử dụng một git repository trên github để deploy lên Heroku. Tuy nhiên, bạn không nên sử dụng Heroku để thay thế cho các dịch vụ git như github hay gitlab.

## Các khái niệm cơ bản

**Heroku Runtime:** Hỗ trợ cung cấp các smart container mà ứng dụng của bạn sẽ chạy trong đó. Bên cạnh đó, tính năng này còn giúp xử lý mọi thứ quan trọng trong ứng dụng bao gồm cấu hình, điều phối, cân bằng tải, backup, log, bảo mật...

**Heroku Postgres (SQL):** Đây là một dịch vụ cơ sở dữ liệu SQL được quản lý trực tiếp bởi Heroku nên rất đáng tin cậy và rất an toàn cho người dùng. Chúng được thiết lập dễ dàng với mã hóa nhanh chóng, mở rộng quy mô đơn giản,...

**Scale:** Tính năng Scale mang lại cho bạn khả năng mở rộng quy mô của ứng dụng ngay lập tức, theo cả chiều dọc lẫn chiều ngang. Hơn nữa, bạn có thể điều hành mọi thứ một cách đơn giản từ các dự án sở thích nhỏ lẻ cho đến thương mại điện tử cấp doanh nghiệp.

**Heroku Redis:** Hỗ trợ dịch vụ Redis phục vụ cho lập trình viên sử dụng. Dịch vụ này cho phép lưu trữ trên bộ nhớ với tốc độ nhanh phô biến

**Add-ons:** Hỗ trợ bạn quản lý các ứng dụng với dịch vụ được tích hợp sẵn như New Relic, Mongo DB, SendGrid, Searchify, Fasty, ClearDB,...

**Code/data rollback:** Giúp người dùng có thể dễ dàng khôi phục mã nguồn hoặc cơ sở dữ liệu của mình về trạng thái trước đó.

**App Metrics:** Đây là tính năng giúp giám sát lưu lượng, thời gian phản hồi, bộ nhớ, tải CPU và lỗi. Giúp bạn quản lý ứng dụng chặt chẽ

**Continuous Delivery:** Heroku Flow sử dụng Heroku Pipeline, review Apps và GitHub hỗ trợ xây dựng quy trình pipeline CI/CD gồm xây dựng, test, deploy

## Ưu điểm

- Database hoàn toàn miễn phí
- Cung cấp nhiều gói xây dựng đa dạng
- Chứa nhiều ngôn ngữ lập trình phổ biến
- SSL sử dụng miễn phí
- Có khả năng hỗ trợ làm việc nhóm
- Cho phép liên kết với các loại GitHub đơn giản nhất.

## Nhược điểm

- Heroku chỉ cho bạn sử dụng miễn phí 550 giờ/tháng. Tuy nhiên, bạn có thể gia tăng số giờ sử dụng lên đến 1000 giờ/tháng qua việc bạn trả phí. Với 1000 giờ này sẽ đủ để cho một blog cá nhân hoạt động mượt mà.
- Chỉ sau từ 2 đến 3 giờ, nếu server không có người truy cập thì nó sẽ tự động chuyển sang trạng thái ngủ.

### 3.5.5. Git

Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thẻ ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.

#### Các lệnh cơ bản:

- **git config:** Để set user name và email của bạn trong main configuration file. Để kiểm tra tên và kiểu email trong cấu hình.
- **git init:** tạo một thư mục mới có tên .git, thư mục này chứa tất cả các tập tin cần thiết cho kho chứa.
- **git clone:** Copy 1 git repository từ remote source.
- **git status:** Để check trạng thái của những file bạn đã thay đổi trong thư mục làm việc.
- **git add:** thêm mới, sửa, xoá files,... Bạn cần phải cập nhật lên Staging Area. Để cập nhật hết các files:
- **git commit :** Lưu lại một snapshot của các sự thay đổi trong thư mục làm việc. Và các tập tin, thư mục được thay đổi đã phải nằm trong Staging Area.
- **git push:** Cập nhật lên server thì bạn phải sử dụng câu lệnh push
- **git pull:** Pull từ remote repository. Lệnh trên sẽ gộp những thay đổi mới kéo về từ máy chủ từ xa với nhánh hiện tại trên máy local.

- **git branch:** Tạo ra nhiều nhánh (branch) khác nhau
- **git checkout:** Chuyển sang branch khác:
- **git merge:** merge code lại vào nhánh gốc (master).
- **git log:** Xem lại lịch sử commit
- **git diff:** Xem thay đổi trước khi push \$ git diff Lệnh này giúp bạn biết những gì đã được thay đổi giữa nhánh hiện tại và nhánh trước nó.
- **git rebase:** Gộp commit.

## **Ưu điểm**

- Sắp xếp công việc tốt hơn
- Linh hoạt hơn khi phải làm cùng lúc nhiều task
- Tự tin hơn khi thử nghiệm những ý tưởng mới
- Git cho phép chúng ta làm việc offline
- Git miễn phí

## **Nhược điểm**

- **Điểm yếu SHA1:** Thời gian trôi đi, những nhà mật mã đã phát hiện ra ngày càng nhiều điểm yếu của thuật toán SHA1. Thực tế người ta đã đã phát hiện thấy sự va chạm giá trị băm. Trong khoảng vài năm, có lẽ những chiếc PC thông thường cũng đủ sức để âm thầm làm hư hỏng một kho Git.
- **Microsoft Windows:** Sử dụng Git trên hệ điều hành Microsoft Windows có vẻ hơi cồng kềnh một chút: Cygwin, mô phỏng Linux dành cho Windows, có chứa Git đã chuyển đổi để chạy trên Windows.
- **Các tập tin không liên quan:** Nếu dự án của bạn rất lớn và chứa rất nhiều tập tin không có liên quan mà luôn luôn bị thay đổi, Git có thể chịu thiệt thòi hơn các hệ

thống khác bởi vì các tập tin không được giữ dấu viết từng cái riêng lẻ. Git giữ các dấu vết thay đổi cho toàn bộ dự án, điều này thường là có lợi.

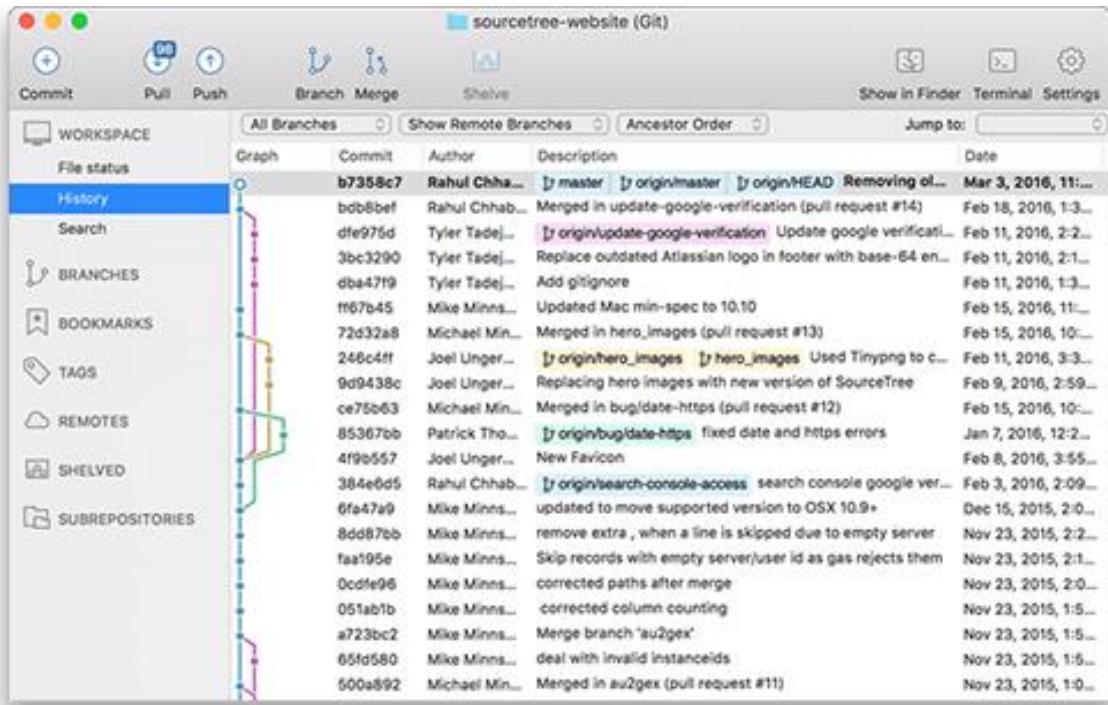
- **Lịch sử tập tin:** Sau khi Git ghi lại các thay đổi cho các dự án lớn, việc cấu trúc lại lịch sử của một tập tin đơn lẻ yêu cầu phải làm việc nhiều hơn các chương trình quản lý mã nguồn giữ dấu vết theo các tập tin riêng lẻ. Thiệt hại thường là không đáng kể, và thứ đáng giá mà nó nhận được là các tác vụ khác hoạt động hiệu quả đến không ngờ
- **Khởi tạo bản sao :** Việc tạo một bản sao có vẻ hơi xa xỉ hơn là việc *checkout* trong các hệ thống quản lý mã nguồn khác khi phần mềm có lịch sử phát triển lâu dài.
- **Các dự án hay thay đổi:** Git được viết ra với mục đích chú tâm đến kích thước tạo ra bởi các thay đổi. Con người chỉ tạo ra sự thay đổi rất nhỏ giữa các phiên bản. Như là bỏ xung lời nhận xét là có sửa lỗi ở đây, có đặc tính mới ở đây, sửa lỗi chủ thích, v.v.. Nhưng nếu các tập tin của bạn căn bản khác nhau, thì trong mỗi lần commit, nó sẽ ghi lại toàn bộ các thay đổi vào lịch sử và làm cho dự án của bạn tất yếu sẽ tăng kích cỡ.
- **Với thư mục rỗng:** Các thư mục rỗng không được theo dõi. Tạo ra các thư mục giả để thử trực trặc này.
- **Lần commit khởi tạo:** Hệ thống số đếm khoa học của máy tính đếm từ 0, thay vì 1. Thật không may, có liên quan đến các lần commit, Git không tôn trọng quy ước này. Rất nhiều lệnh bất lợi trước lần commit khởi tạo.Thêm nữa, các trường hợp ngoại lệ phải được xử lý theo một cách đặc biệt, như là việc rebasing một nhánh với lần commit khởi tạo khác.

### 3.5.6. Source tree

Sourcetree là một ứng dụng máy tính để bàn giao diện người dùng đồ họa (GUI) miễn phí giúp đơn giản hóa cách bạn tương tác với các kho lưu trữ Git để bạn có thể hoàn toàn tập trung vào việc viết mã.

## Các đặc điểm nổi bật

- **Tự động theo dõi code:** Dấu trang cung cấp cho bạn cái nhìn tóm tắt về tất cả các dự án và lưu trữ trong thời gian thực.



Hình 3.11: Source tree

- **Dễ dàng cho người mới bắt đầu:** Sourcetree là một GUI đầy đủ chức năng hoạt động với Git và Mercurial.

Nó tương thích với cả hệ điều hành Windows hoặc Mac.

Hướng dẫn chi tiết giúp bạn bắt đầu nhanh chóng.

Tất cả các lệnh đều có thể truy cập được chỉ với một cú nhấp chuột. Nó cung cấp kiểm soát phiên bản phân tán được đơn giản hóa.

- **Mạnh mẽ cho các chuyên gia:** Thực hiện đánh giá và lưu các thay đổi. Kiểm tra các tập thay đổi đi và đến của bạn. Tùy chỉnh các bộ thay đổi và lưu trữ. Lựa chọn tự

do giữa các nhánh. Áp dụng các bản vá lỗi, làm lại cơ sở hoặc đặt các bộ thay đổi của bạn lên kệ.

## **Chương 4. PHÂN TÍCH, THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG**

### **4.1. Đặt vấn đề**

#### **4.1.1. Thực trạng ứng dụng xảy ra lỗi khi người dùng thao tác**

Hiện tại trên thế giới có rất nhiều phần mềm ứng dụng, ứng dụng web được đông đảo nhiều người sử dụng như: Facebook, Microsoft Office, Trello, Figma, .v.v. Hầu hết những ứng dụng lớn đều có tính ổn định rất cao nhưng thỉnh thoảng người dùng vẫn sẽ gặp một vài trường hợp ứng dụng chạy không đúng, thao tác không được, ứng dụng bị đơ gây lại cảm giác khó chịu cho người dùng. Đối với các ứng dụng có lượng người sử dụng cao nếu không khắc phục ngay sẽ dẫn tới tình trạng bị đối thủ cạnh tranh mất đi lượng lớn khách hàng. Lấy ví dụ Facebook ngày 4/10 bị lỗi nghiêm trọng toàn cầu. Khi vào bằng máy tính, trình duyệt hiện lỗi "This site can't be reached", trong khi vào bằng ứng dụng di động sẽ không thể cập nhật nội dung mới. Điều này dẫn đến một số người dùng phải đi tìm những ứng dụng khác để thay thế và có thể chuyển sang ứng dụng mới đó nếu nó tốt hơn.

Đối với những ứng dụng vừa và nhỏ phục vụ trong nội bộ công ty hay cá nhân, nếu hoạt động sai như là phần mềm tính công, tính lương, quản lý ngày phép sẽ mang lại tổn thất rất nhiều.

#### **4.1.2. Khó khăn khi sửa lỗi ứng dụng**

Khi nhà phát triển hoặc lập trình viên mỗi khi sửa lỗi đều đối mặt với những vấn đề sau:

- Không biết lỗi xuất phát từ đâu.
- Rất mất thời gian để truy vết lỗi.
- Không biết ai, nguyên nhân gây ra lỗi.
- Cách khắc phục để hạn chế xảy ra những lỗi tương tự.

### **4.1.3. Tối ưu quy trình quản lý lỗi**

Khi ứng dụng phát sinh lỗi thông thường người quản lý dự án hoặc nhà phát triển cần phải biết:

- Thời gian lỗi xuất hiện.
- Loại lỗi và mức độ nghiêm trọng của lỗi.
- Lỗi phát sinh ở chỗ nào.
- Ai là người xử lý lỗi.
- Lỗi cần xử lý đang ở trong giai đoạn nào: Mới, Đang xử lý, Đã xử lý.
- Ai là người kiểm tra lại sau khi lỗi đã được xử lý.
- Ghi nhận bao nhiêu lỗi phát sinh trong 1 tháng, 1 quý, 1 năm.

## **4.2. Phương pháp**

### **4.2.1. Xây dựng Package nhúng vào code của ứng dụng cần giám sát lỗi**

Hiện tại để có thể tự động ghi nhận lỗi của một ứng dụng, nhóm chúng em cần tạo ra một đoạn mã có thể nhúng vào dự án cần giám sát. Ở đây nhóm em tạo một package được viết bằng ngôn ngữ JavaScript và sẽ đưa lên NPM (Node package manager). Người dùng chỉ cần cài đặt và nhúng vào dự án, package sẽ tự động ghi nhận mỗi khi ứng dụng phát sinh lỗi. Hiện tại package này chỉ mới được hỗ trợ cho các ứng dụng web viết bằng JavaScript, ReactJs.

Để sử dụng người dùng chỉ cần cài đặt và gắn đoạn mã này như hình dưới đây:

```
# Using npm  
npm i @error-monitoring/package  
# Using yarn  
yarn add @error-monitoring/package
```

```

import React from "react";
import ReactDOM from "react-dom";
import Error from "@error-monitoring/package";
import App from "./App";

Error.init({
  projectId: "your-project-id",
  env: "development", // There are three config env: development, staging, production
});

ReactDOM.render(<App />, document.getElementById("root"));

```

Trong đoạn mã, sẽ sử dụng hàm có sẵn của browser để tự động bắt lỗi window.error, ở đoạn mã trên sẽ ghi ra những thông tin sau:

- message: thông báo lỗi.
- source: URL của tập lệnh nơi lỗi đã được phát sinh.
- lineno: Số dòng nơi lỗi đã được phát sinh.
- colno: Số cột cho dòng xảy ra lỗi.
- error: Thông tin chi tiết của lỗi.

#### **4.2.2. Tối ưu hóa thời gian xử lý lỗi với giao diện Kanban Board**

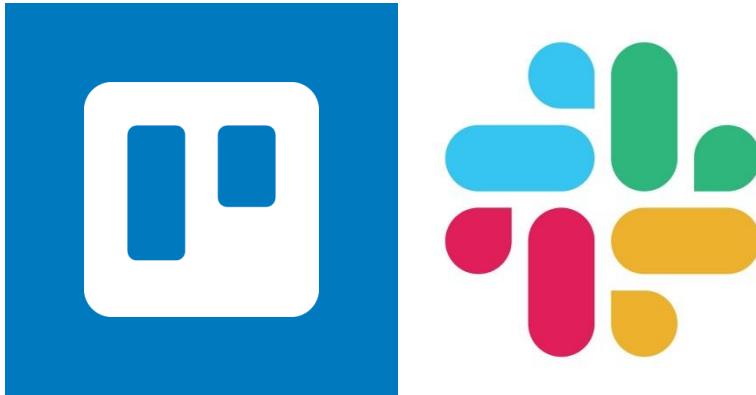
Bảng Kanban là công cụ để trực quan hóa công việc. Bảng Kanban bao gồm các cột tương ứng với trạng thái của công việc. Mỗi công việc khi ở trạng thái nào thì được đặt ở cột tương ứng. Chúng ta có thể dùng một bảng vật lý hoặc một phần mềm hỗ trợ Kanban như Trello.

Ở đây nhóm em sử dụng Bản Kanban để trực quan hóa giao diện lỗi hơn, dễ dàng thao tác với công việc xử lý lỗi như: thêm người xử lý lỗi, thay đổi trạng thái công việc, thay đổi độ ưu tiên v.v.

The screenshot shows a dashboard titled "Issues" with three main sections: "UnResolved", "Processing", and "Reviewing".

- UnResolved:**
  - Uncaught Error**: "Do phat son huy" (Priority: Medium) - Last updated 3 months ago on 20/09/2021. Status: Sep 26 11:11 → Sep 30 11:11. Reviewer: ha thanh huy.
  - TypeError**: "Nguyễn Trường Vũ" (Priority: High) - Last updated 3 months ago on 15/09/2021. Status: Start date → Due date. Reviewer: Ark 01.
  - TypeError**: "Ark 01" (Priority: Medium) - Last updated 3 months ago on 15/09/2021. Status: Start date → Due date. Reviewer: Ark 01.
- Processing:**
  - TypeError**: "Nguyễn Trường Vũ" (Priority: High) - Last updated 3 months ago on 15/09/2021. Status: Start date → Due date. Reviewer: Ark 01.
  - SyntaxError**: "Ark 01" (Priority: Medium) - Last updated 3 months ago on 20/09/2021. Status: Start date → Due date. Reviewer: Ark 01.
- Reviewing:**
  - TypeError**: "Nguyễn Trường Vũ" (Priority: High) - Last updated 3 months ago on 15/09/2021. Status: Start date → Due date. Reviewer: Ark 01.
  - TypeError**: "Ark 01" (Priority: Medium) - Last updated 3 months ago on 20/09/2021. Status: Start date → Due date. Reviewer: Ark 01.

#### 4.2.3. Tích hợp các ứng dụng bên thứ 3 hỗ trợ thông báo: Trello, Slack



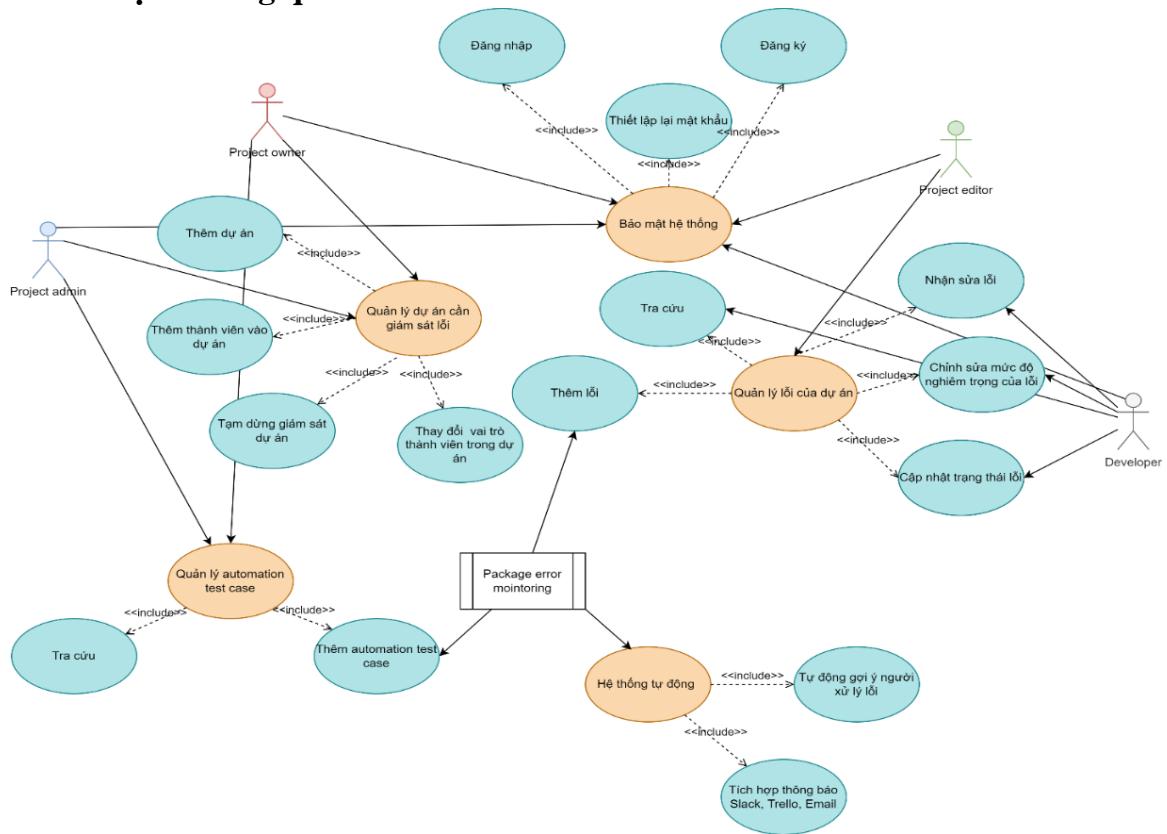
Slack là một ứng dụng chat nhóm, bạn hoàn toàn có thể trao đổi thông tin với bạn bè, đồng nghiệp đặc biệt đây là một lựa chọn hoàn hảo cho bạn trong việc phục vụ công việc, trao đổi thông tin công việc với đồng nghiệp thông qua Slack thay vì **dùng mail**,

Trello là công cụ quản lý công việc trực quan hỗ trợ các nhóm lên ý tưởng, lập kế hoạch, quản lý và ăn mừng thành tựu cùng nhau một cách hợp tác, hiệu quả và có tổ chức.

Slack và Trello là hai công cụ lớn và được đa số doanh nghiệp công ty dùng, việc tích hợp thông báo qua 2 kênh này giúp nhà phát triển nắm bắt được khi nào ứng dụng phát sinh lỗi hỗ trợ xử lý nhanh hơn, dễ dàng hơn.

### 4.3. Đặc tả yêu cầu

#### 4.3.1. Đặc tả tổng quát



Hình 4.1: Sơ đồ Use-case tổng quát

#### 4.3.2. Danh sách các Actor

Bảng 1: Danh sách các Actor

STT	Tên Actor	Ý nghĩa/Ghi chú
1	Project Owner	Chủ dự án có toàn quyền trong dự án cần giám sát: Thêm, cập nhật dự án, Thêm thành viên vào dự án, quản lý lỗi, automation test của dự án.
2	Project Admin	Có Mọi quyền như Project owner ngoại trừ quyền ẩn dự án, thay đổi quyền hạn của project owner.

3	Project Editor	Là người có quyền quản lý lỗi, tra cứu lỗi, giao công việc sửa lỗi cho developer, Không có quyền quản lý dự án.
4	Developer	Chỉ có quyền nhận sửa lỗi, cập nhật trạng thái của lỗi.
5	Package Error monitoring	Là thư viện được nhúng vào dự án cần giám sát lỗi, Tự động tạo ra lỗi, automation test từ dự án, tích hợp thông báo trello, slack, gmail, tự động đề xuất người sửa lỗi.

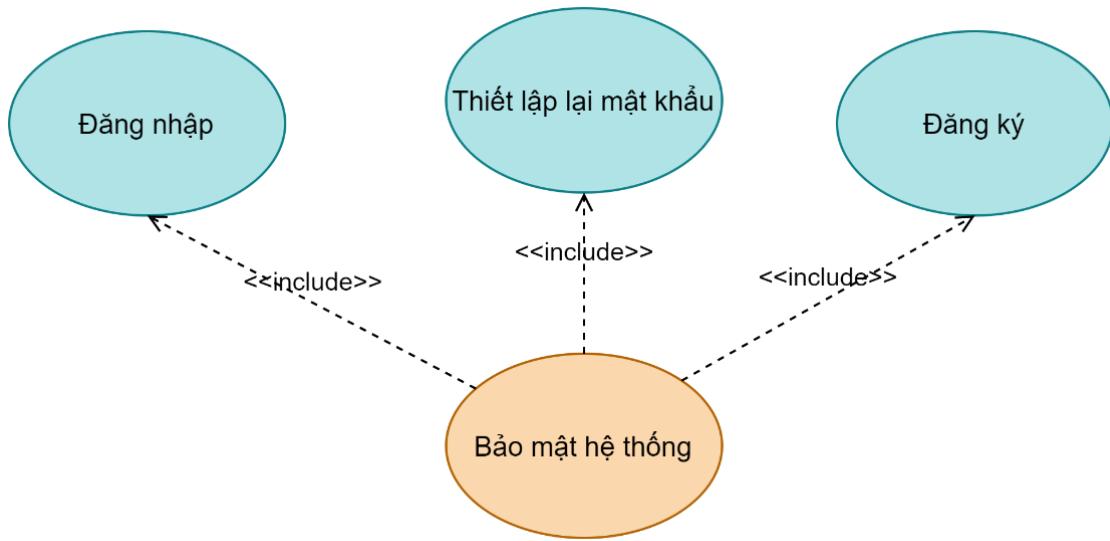
#### 4.3.3. Danh sách các Use-case

Bảng 2: Danh sách các Use-case

STT	Tên Use-case	Ý nghĩa/Ghi chú
1	Use-case bảo mật hệ thống	Tạo chức năng đăng ký đăng nhập quên mật khẩu cho người dùng
2	Use-case Quản lý Dự án cần giám sát	Thêm dự án, Cập nhật thông tin dự án, Thêm thành viên vào dự án, ẩn dự án, Tra cứu
3	Use-case Quản lý lỗi của Dự án cần giám sát	Tự động thêm lỗi. Cập nhật trạng thái lỗi, giao quyền sửa lỗi cho thành viên, nhận quyền sửa lỗi, cập nhật mức độ nghiêm trọng của lỗi, điều chỉnh hạn sửa lỗi.
4	Use-case Quản lý Automation test của dự án	Tự động thêm automation test case, Cho phép người dùng tra cứu.
5	Use-case Tích hợp, Tự động	Tự động thông báo đến các kênh giao tiếp Slack, Trello, Gmail khi có phát sinh lỗi. Tự động đề xuất người xử lý.

#### 4.4. Đặc tả Use-case

##### 4.4.1. Đặc tả Use-case “Bảo mật hệ thống”



Hình 4.2: Đặc tả Use-case “Bảo mật hệ thống”

Bảng 3: Use-case “Bảo mật hệ thống”

Mã Use case	UC001	Tên Use case	Bảo mật hệ thống
Tác Nhân	Project owner, Project Admin, Project Editor, Developer		
Mô tả	Thực hiện các tác vụ bảo mật như: Tạo tài khoản, Đăng nhập, Thiết lập lại mật khẩu		
Sự kiện kích hoạt	Nhấn nút Đăng nhập, Nhấn nút Đăng Ký, Nhấn nút quên mật khẩu		
Tiền điều kiện			
<b>Đăng ký</b>			
Luồng sự kiện chính (Thành công)	Stt	Thực hiện bởi	Hành động
	1	Hệ thống	Hiển thị giao diện đăng ký

	2	Khách	Nhập các thông tin tài khoản
	3	Khách	Yêu cầu đăng ký
	4	Hệ thống	Kiểm tra xem Khách đã điền thông tin bắt buộc chưa
	5	Hệ thống	Kiểm tra email đã nhập hợp lệ chưa
	6	Hệ thống	Kiểm tra mật khẩu nhập lại và mật khẩu có trùng nhau hay không
	7	Hệ thống	Lưu thông tin tài khoản và thông báo đăng ký thành công
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	4a	Hệ thống	Thông báo lỗi: Cần nhập các trường bắt buộc nếu khách nhập thiếu
	5a	Hệ thống	Thông báo lỗi: Địa chỉ email không hợp lệ nếu địa chỉ email không hợp lệ
	6a	Hệ thống	Thông báo lỗi: Mật khẩu xác nhận không trùng với Mật khẩu nếu hai mật khẩu không trùng nhau
<b>Hậu điều kiện</b>	Tài khoản được lưu trữ vào hệ thống		

### Đăng nhập

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	Hệ thống	Hiển thị giao diện đăng nhập
	2	Khách	Nhập email và mật khẩu
	3	Khách	Yêu cầu đăng nhập

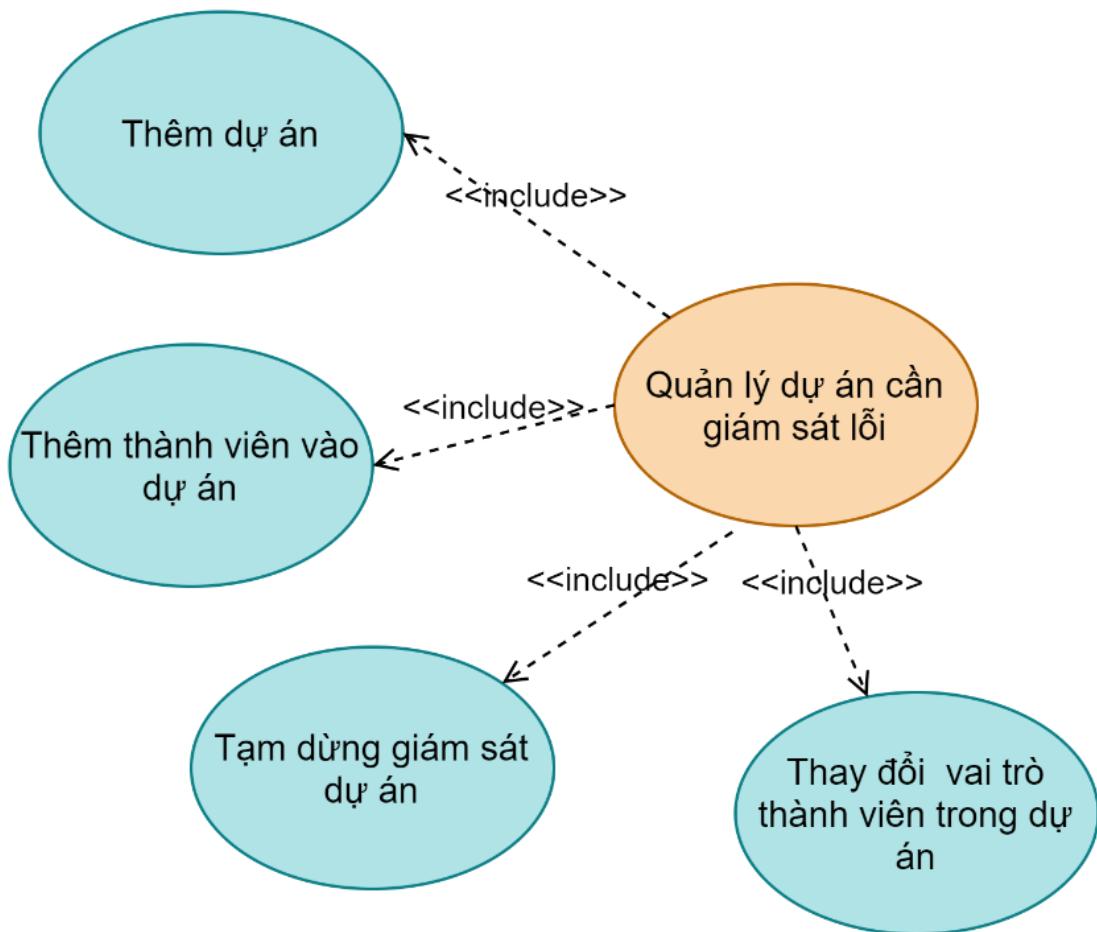
	4	Hệ thống	Kiểm tra xem Khách đã điền thông tin bắt buộc chưa
	5	Hệ thống	Kiểm tra email đã nhập hợp lệ chưa
	6	Hệ thống	Hiển thị chức năng tương tác đối với người dùng
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	5a	Hệ thống	Thông báo lỗi: Cần nhập các trường bắt buộc nếu khách nhập thiếu
	6a	Hệ thống	Thông báo lỗi: Email hoặc mật khẩu chưa đúng nếu không tìm thấy email và mật khẩu hợp lệ với email trong hệ thống
<b>Hậu điều kiện</b>	Tác nhân đăng nhập được vào hệ thống		

### Thiết lập lại mật khẩu

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút quên mật khẩu ở màn hình đăng nhập
	2	Hệ thống	Hiển thị giao diện nhập email tài khoản quên
	3	Người dùng	Nhập email tài khoản quên
	4	Người dùng	Yêu cầu thiết lập lại mật khẩu (Nhấn nút send mail)
	5	Hệ thống	Kiểm tra email đã nhập hợp lệ chưa

	6	Hệ thống	Kiểm tra định dạng email có hợp lệ và có tồn tại trong hệ thống không nếu có gửi mã đk vào email và chuyển sang màn hình thiết lập lại mật khẩu
	7	Người dùng	Người dùng nhập mật khẩu mới, xác nhận mật khẩu và mã xác nhận nhận được từ mail
	8	Hệ thống	Kiểm tra xem người dùng có nhập đúng mã xác nhận và mật khẩu xác nhận có trùng với mật khẩu mới hay không
	9	Hệ thống	Cập nhật mật khẩu thông báo thành công và chuyển người dùng về màn hình đăng nhập
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	8a	Hệ thống	Thông báo lỗi: Mã xác nhận không hợp lệ và mật khẩu xác nhận không hợp lệ khi người dùng nhập sai mã xác nhận và sai mật khẩu xác nhận
<b>Hậu điều kiện</b>	Hệ thống cập nhật mật khẩu mới của người dùng thành công		

#### 4.4.2. Đặc tả Use-case “Quản lý dự án cần giám sát”



Hình 4.3: Đặc tả Use-case “Quản lý dự án cần giám sát”

Bảng 4: Use-case “Quản lý dự án cần giám sát”

Mã Use case	UC002	Tên Use case	Quản lý dự án cần giám sát lõi
Tác Nhân	Project owner, Project Admin (PO, PA)		
Mô tả	Thực hiện các thao tác với dự án như: Tạo mới dự án, Thêm thành viên vào dự án, Tạm dừng giám sát dự án, Thay đổi vai trò thành viên trong dự án		

<b>Sự kiện kích hoạt</b>	Khi Người dùng lần đầu đăng nhập, Khi người dùng nhấn vào nút Tạo mới dự án, Thêm thành viên vào dự án, ẩn dự án, Sửa đổi vai trò thành viên trong dự án
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập thành công

### Tạo mới dự án cần giám sát

<b>Luồng sự kiện chính (Thành công)</b>	<b>Stt</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1	PO	Lần đầu đăng nhập, hoặc nhấn vào nút tạo mới dự án
	2	Hệ thống	Hiển thị giao diện tạo mới dự án cần giám sát
	3	PO	Nhập thông tin dự án cần tạo (Tên Dự án, Nền tảng)
	4	Hệ thống	Kiểm tra xem PO có nhập đầy đủ thông tin chưa
	5	Hệ thống	Lưu trữ thông tin dự án vào dữ liệu
<b>Luồng sự kiện thay thế</b>	<b>Stt</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	4a	Hệ thống	Thông báo lỗi: Cần nhập các trường bắt buộc nếu PO nhập thiếu
	5a	Hệ thống	Nếu PO lần đầu đăng nhập sẽ chuyển sang trang “Cập nhật thông tin và khởi tạo project mới” và thông báo tạo dự án thành công
	5b	Hệ thống	Nếu PO Nhấn vào nút khởi tạo mới dự án , hệ thống sẽ truyền về trang “Danh sách dự án”
<b>Hậu điều kiện</b>	Dự án cần giám sát được tạo mới		

### Thêm thành viên vào dự án

<b>Luồng sự kiện chính (Thành công)</b>	<b>Stt</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1	PO, PA	Người dùng chọn vào project cần thêm thành viên, chọn vào “Cài đặt”, và ấn vào “Thành viên”.

	2	Hệ thống	Hiển thị trang quản lý thành viên của dự án
	3	PO, PA	Nhập tên thành viên muốn thêm vào dự án ở ô “Tìm kiếm thành viên”
	4	Hệ thống	Hiển thị danh sách các thành viên trùng với tên của người dùng tìm kiếm
	5	PO, PA	Chọn thành viên cần thêm vào dự án và chọn vai trò cho thành viên mới.
	6	Hệ thống	Lưu lại thành viên mới được thêm vào dữ liệu và hiển thị lại danh sách thành viên của dự án
<b>Hậu điều kiện</b>	Thêm mới thành công thành viên vào dự án		

Thay đổi vai trò thành viên trong dự án

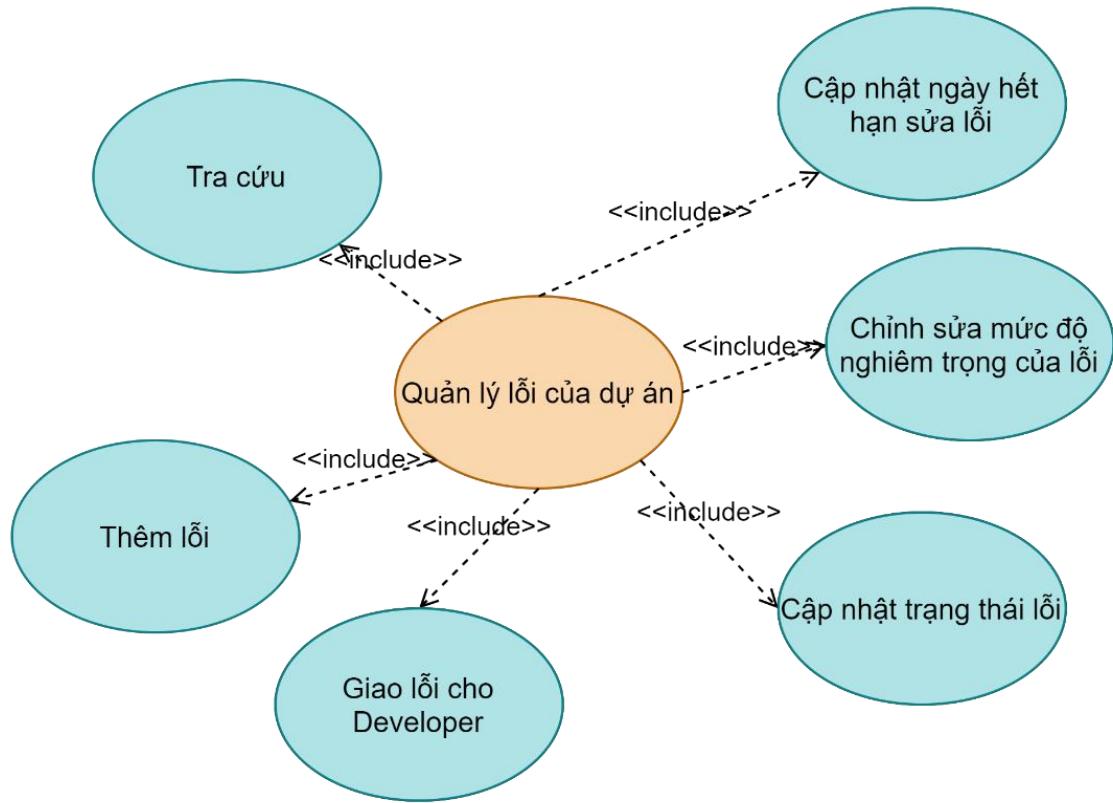
Luồng sự kiện chính (Thành công)	Stt	Thực hiện bởi	Hành động
	1	PO	Người dùng chọn vào project cần thêm thành viên, chọn vào “Cài đặt”, và ấn vào “Cài đặt chung”.
	2	Hệ thống	Hiển thị trang quản lý thành viên của dự án
	3	PO, PA	Chọn vào thành viên cần thay đổi vai trò và nhấn vào nút “Chỉnh sửa”
	4	Hệ thống	Hiển thị popup danh sách các vai trò của thành viên
	5	PO, PA	Chọn vai trò cần thay đổi cho thành viên và nhấn nút xác nhận
	6	Hệ thống	Xử lý yêu cầu và hiện thông báo “Thay đổi vai trò thành viên thành công”
	7	Hệ thống	Đóng popup và cập nhật lại trang quản lý thành viên

<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	6a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Hệ thống tạm dừng giám sát lỗi cho dự án		

### Tạm dừng giám sát dự án

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA	Người dùng chọn vào project cần thêm thành viên, chọn vào “Cài đặt”, và ấn vào “Cài đặt chung”.
	2	Hệ thống	Hiển thị giao diện thông tin chung của dự án
	3	PO	Nhấn vào nút “Ấn dự án” để tạm dừng giám sát lỗi cho dự án
	4	Hệ thống	Thực thi yêu cầu và đổi trạng thái dự án, Thông báo “Tạm dừng giám sát dự án thành công”
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	4a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Hệ thống tạm dừng giám sát lỗi cho dự án		

#### 4.4.3. Đặc tả Use-case “Quản lý lỗi của dự án”



Hình 4.4: Đặc tả Use-case “Quản lý lỗi của dự án”

Bảng 5: Use-case “Quản lý lỗi của dự án”

Mã Use case	UC003	Tên Use case	Quản lý lỗi của dự án
<b>Tác Nhân</b>	Project owner, Project Admin, Project Editor, Developer , Package Error Library(PO, PA, PE, DEV, PEL)		
<b>Mô tả</b>	Thực hiện các thao tác với lỗi như: Thêm lỗi, Tra cứu lỗi, Nhận sửa lỗi, Chỉnh sửa mức độ nghiêm trọng của lỗi, Cập nhật trạng thái của lỗi, Giao lỗi,		
<b>Sự kiện kích hoạt</b>	Khi người dùng Nhấn vào “Giao lỗi”, “Mức độ”, “Ngày hết hạn”. Kéo thả lỗi vào các ô trạng thái tương ứng (Chưa xử lý, đang xử lý, Thành công )		

<b>Tiền điều kiện</b>	Người dùng đã đăng nhập thành công, Đã tạo thành công dự án, Cài đặt thành công Package Error vào dự án cần được giám sát		
-----------------------	---	--	--

### Thêm lỗi

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PEL	Tự động thêm mới lỗi vào hệ thống khi dự án phát sinh lỗi
	2	Hệ thống	Hệ thống nhận thông tin lỗi từ PEL và lưu vào hệ thống
<b>Hậu điều kiện</b>	Lỗi mới được thêm vào hệ thống		

### Giao lỗi cho DEV

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE	Người dùng chọn vào lỗi cần thực thi
	2	PO, PA, PE	Người dùng nhấn vào nút “Giao lỗi”
	3	Hệ thống	Hệ thống hiện danh sách các thành viên trong dự án
	4	PO, PA, PE	Chọn Thành viên cần giao lỗi
	5	Hệ thống	Cập nhật lại thành viên được giao lỗi và cập nhật vào dữ liệu rồi cập nhật lại trang danh sách lỗi
<b>Luồng sự kiện thay thế</b>	5a	Hệ thống	Khi cập nhật lại thành viên không thành công phải thông báo lỗi
<b>Hậu điều kiện</b>	Giao lỗi thành công cho thành viên trong dự án		

### Cập nhật trạng thái lỗi

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE, DEV	Người dùng chọn vào lỗi cần cập nhật trạng thái
	2	PO, PA, PE, DEV	Kéo lỗi vào trạng thái tương ứng cần cập nhật (“Chưa xử lý”, “Đang xử lý”, “Đã xử lý”)
	3	Hệ thống	Hệ Thống Cập nhật lại trạng thái của lỗi, cập nhật vào dữ liệu và cập nhật lại trang danh sách lỗi
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	3a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Cập nhật trạng thái lỗi thành công		

### **Chỉnh sửa mức độ nghiêm trọng của lỗi**

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE	Người dùng chọn vào lỗi cần cập nhật mức độ nghiêm trọng
	2	Hệ thống	Hiển thị danh sách các mức độ (“Tháp”, “Trung bình”, “Cao”)
	3	PO, PA, PE	Chọn vào mức độ cần cập nhật cho lỗi
	4	Hệ thống	Thực thi yêu cầu, đổi mức độ nghiêm trọng của lỗi và cập nhật lại trang danh sách lỗi
	Stt	Thực hiện bởi	Hành động

<b>Luồng sự kiện thay thế</b>	4a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Lỗi được cập nhật thành công mức độ nghiêm trọng		

### Cập nhật ngày hết hạn sửa lỗi

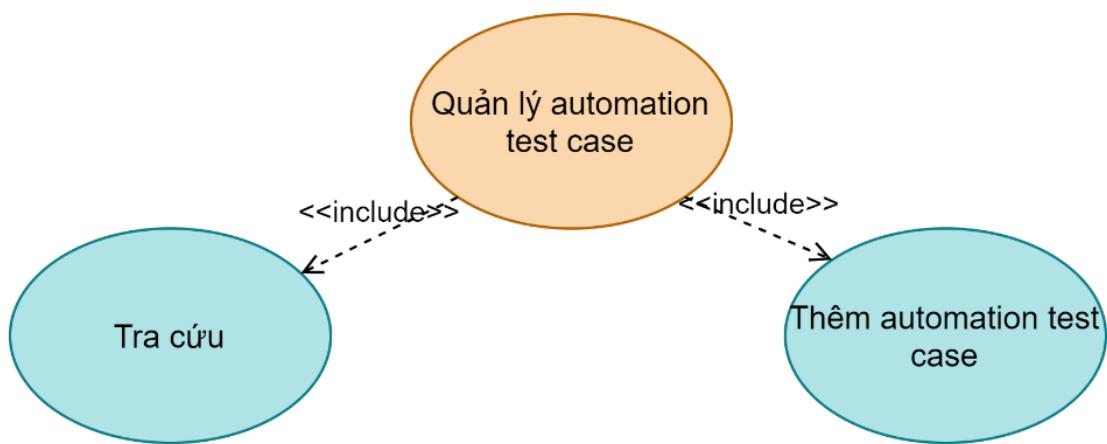
<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE, DEV	Người dùng chọn vào lỗi cần cập nhật ngày hết hạn sửa lỗi
	2	Hệ thống	Hiển thị popup cho người dùng chọn ngày
	3	PO, PA, PE, DEV	Chọn ngày cần cập nhật
<b>Luồng sự kiện thay thế</b>			Thực thi yêu cầu, cập nhật lại ngày hết hạn sửa lỗi cho lỗi và cập nhật lại trang danh sách lỗi
<b>Hậu điều kiện</b>	Stt	Thực hiện bởi	Hành động
	4a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Ngày hết sửa lỗi của lỗi được cập nhật		

### Tra cứu lỗi

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE, DEV	Người dùng truy cập vào trang danh sách lỗi

	2	PO, PA, PE, DEV	Chọn trường cần tra cứu: Tra cứu theo “Dự án”, Tra cứu theo “Môi trường”: Development, Production, Staging, Tra cứu theo ngày tạo.
	3	Hệ Thống	Hệ thống thực thi yêu cầu tra cứu của người dùng và tra ra danh sách tương ứng với yêu cầu tra cứu.
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	3a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Hiện thị danh sách lỗi theo yêu cầu tra cứu của người dùng		

#### 4.4.4. Đặc tả Use-case “Quản lý Automation test case”



Hình 4.5: Đặc tả Use-case “Quản lý Automation test case”

Bảng 6: Use-case “Quản lý Automation test case”

Mã Use case	UC004	Tên Use case	Quản lý dự án cần giám sát lỗi
<b>Tác Nhân</b>	Project owner, Project Admin, Project Editor, Developer , Package Error Library(PO, PA, PE, DEV, PEL)		
<b>Mô tả</b>	Thực hiện các thao tác với automation test case như: Thêm automation test case, Tra cứu, Xem chi tiết Automation test case		
<b>Sự kiện kích hoạt</b>	Khi hệ thống được deploy hoặc chạy automation test, Khi người dùng vào trang danh sách automation test case để tra cứu		
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập thành công, Đã tạo thành công dự án, Cài đặt thành công Package Error vào dự án cần được giám sát		

#### Thêm automation test case

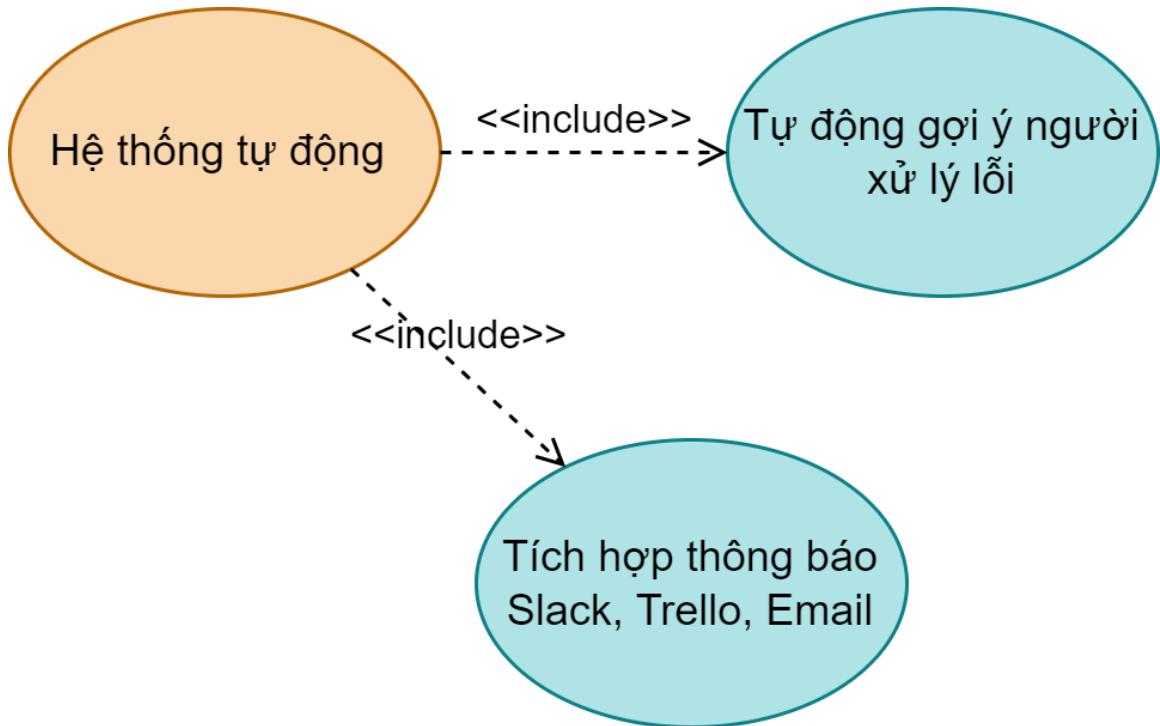
Luồng sự kiện chính (Thành công)	Stt	Thực hiện bởi	Hành động
	1	PEL	Tự động thêm mới automation test case vào hệ thống khi dự án deploy hoặc chạy automation testing
	2	Hệ thống	Hệ thống nhận thông tin automation test case từ PEL và lưu vào hệ thống
<b>Hậu điều kiện</b>	Automation test case mới được thêm vào hệ thống		

#### Tra cứu automation test case

Luồng sự kiện chính (Thành công)	Stt	Thực hiện bởi	Hành động
	1	PO, PA, PE, DEV	Người dùng truy cập vào trang danh sách automation test case
	2	PO, PA, PE, DEV	Chọn trường cần tra cứu: Tra cứu theo “Dự án”, Tra cứu theo “Môi trường”:

			Development, Production, Staging, Tra cứu theo ngày tạo.
	3	Hệ Thống	Hệ thống thực thi yêu cầu tra cứu của người dùng và tra ra danh sách tương ứng với yêu cầu tra cứu.
<b>Luồng sự kiện thay thế</b>	Stt	Thực hiện bởi	Hành động
	3a	Hệ thống	Thực thi yêu cầu không thành công sẽ thông báo lỗi cho người dùng
<b>Hậu điều kiện</b>	Hiện thị danh sách automation test theo yêu cầu tra cứu của người dùng		

#### 4.4.5. Đặc tả Use-case “Tích hợp tự động”



Hình 4.6: Đặc tả Use-case “Tích hợp tự động”

Bảng 7: Use-case “Tích hợp tự động”

<b>Mã Use case</b>	UC005	<b>Tên Use case</b>	Tích hợp, tự động
<b>Tác Nhân</b>	Package Error Library (PEL)		
<b>Mô tả</b>	Thực hiện thông báo tới các kênh Trello, Slack, Gmail. Tự động để xuất người xử lý.		
<b>Sự kiện kích hoạt</b>	Khi hệ thống của người dùng phát sinh lỗi		
<b>Tiền điều kiện</b>	Người dùng đã cài đặt PEL vào hệ thống, đã bật tính năng thông báo Trello, Slack, Gmail, tự động để xuất người xử lý.		

### Tích hợp thông báo Slack, Trello, Gmail.

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PEL	Tiếp nhận và gửi thông tin lỗi tới hệ thống
	2	Hệ thống	Hệ thống nhận thông tin từ PEL, xử lý và gửi thông báo đến các kênh đã tích hợp.
<b>Hậu điều kiện</b>	Thông báo được gửi đến hệ thống tích hợp thành công.		

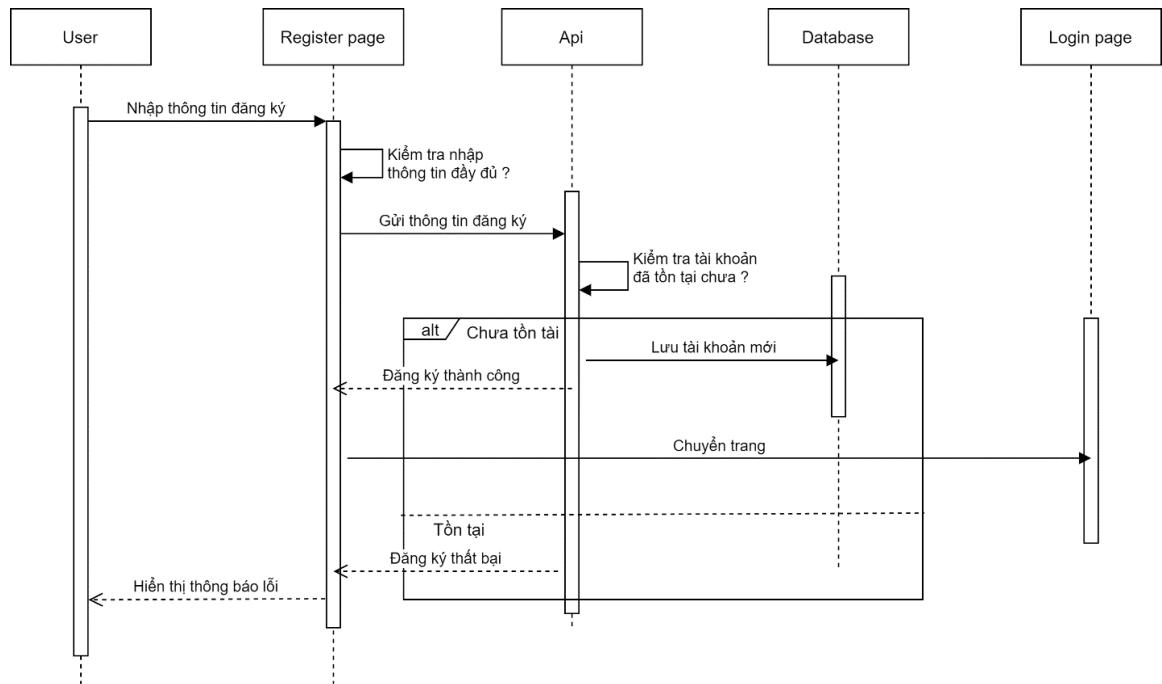
### Tự động để xuất người xử lý

<b>Luồng sự kiện chính (Thành công)</b>	Stt	Thực hiện bởi	Hành động
	1	PEL	Tiếp nhận và gửi thông tin lỗi tới hệ thống
	2	Hệ thống	Hệ thống tiếp nhận thông tin từ PEL, xử lý tính toán và đưa ra để xuất người xử lý.
	Stt	Thực hiện bởi	Hành động

<b>Luồng sự kiện thay thế</b>	3a	Hệ thống	Nếu không bật tính năng tự động đề xuất thì sẽ bỏ qua đề xuất người xử lý.
-------------------------------	----	----------	--

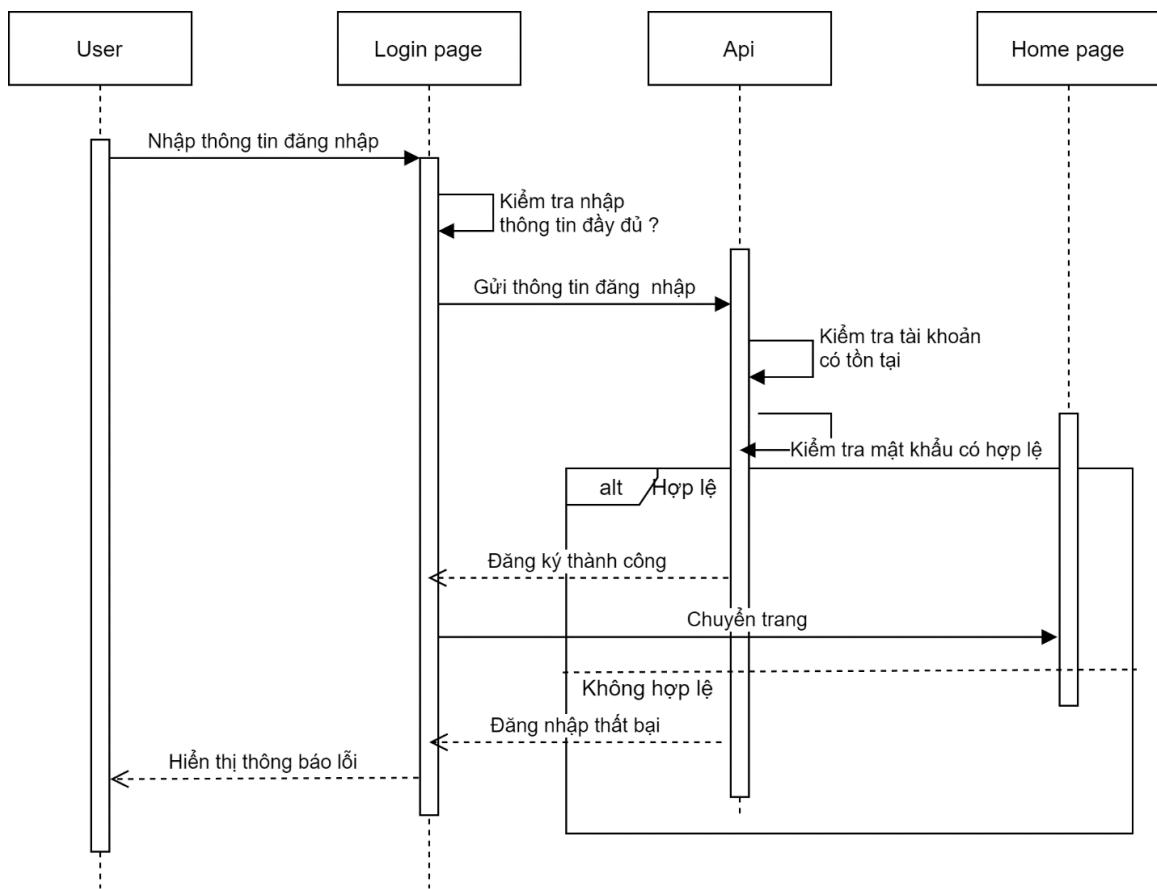
#### 4.5. Đặc tả Sequence Diagram

##### 4.5.1. Sơ đồ Seq Diagram “Đăng ký”



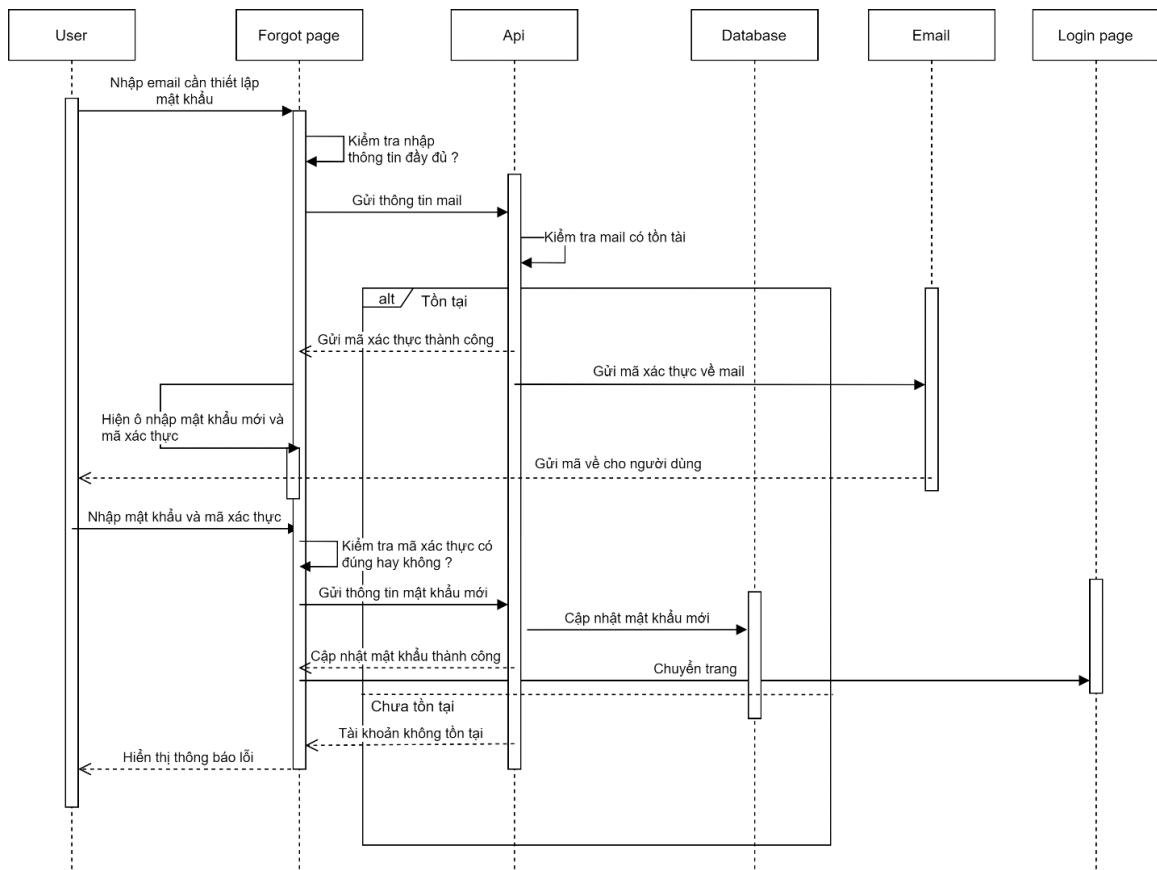
Hình 4.7: Sơ đồ Seq Diagram “Đăng ký”

#### 4.5.2. Sơ đồ Seq Diagram “Đăng nhập”



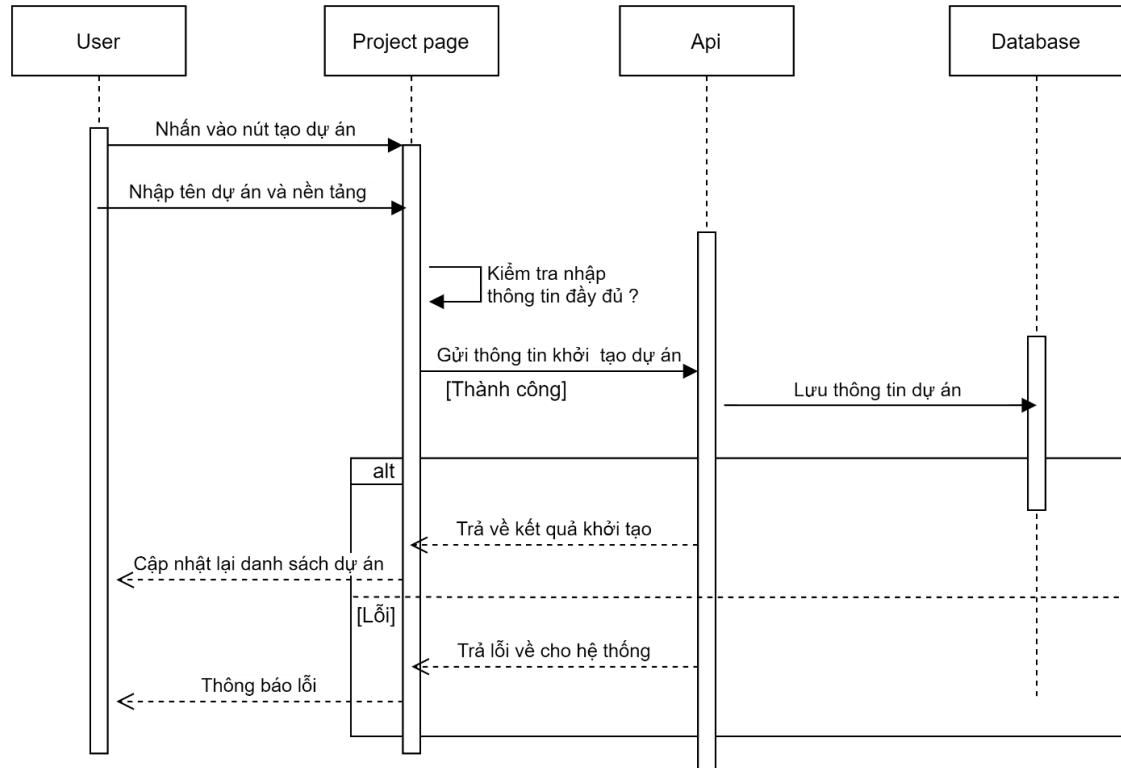
Hình 4.8: Sơ đồ Seq Diagram “Đăng nhập”

### 4.5.3. Sơ đồ Seq Diagram “Thiết lập lại mật khẩu”



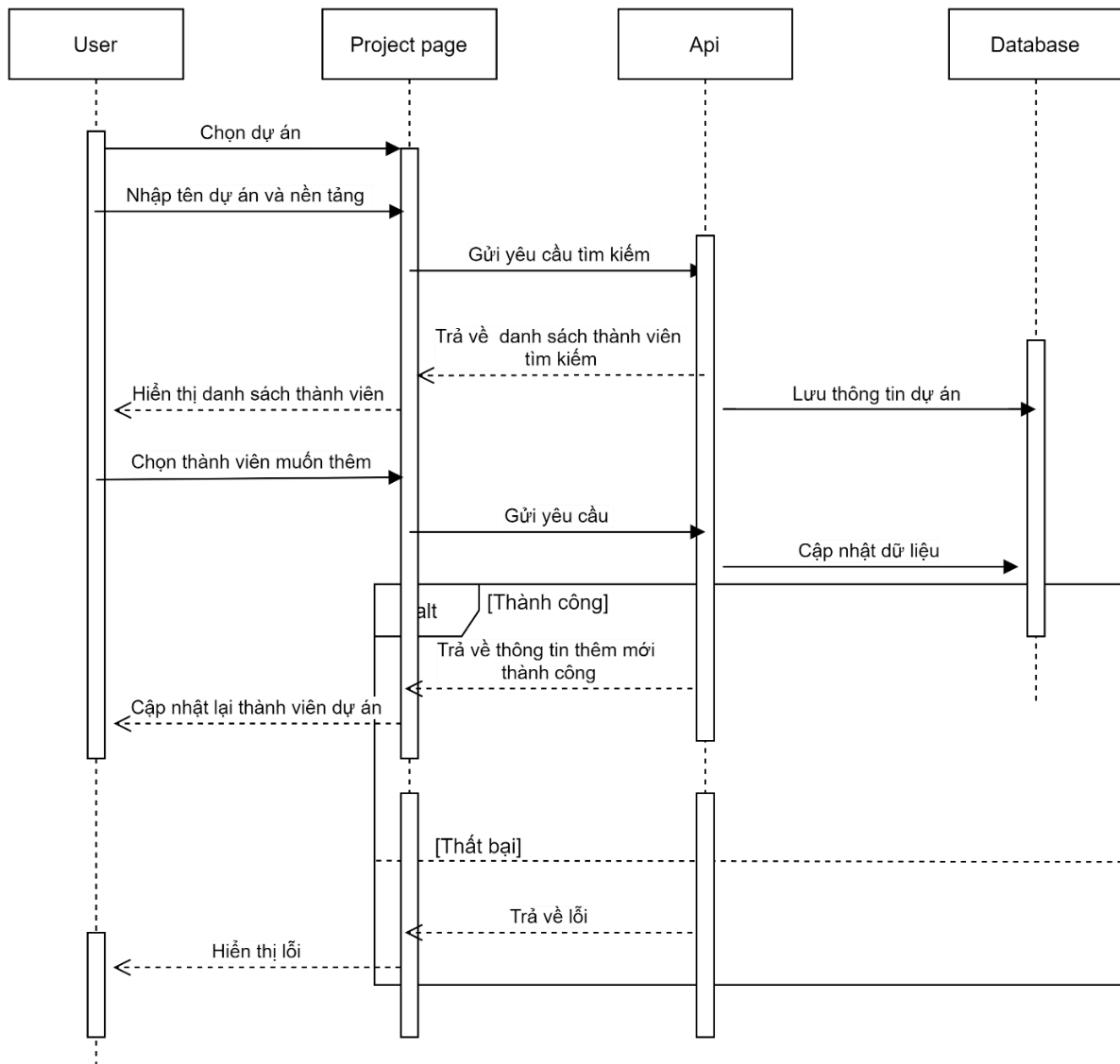
Hình 4.9: Sơ đồ Seq Diagram “Thiết lập lại mật khẩu”

#### 4.5.4. Sơ đồ Seq Diagram “Thêm mới dự án”



Hình 4.10: Sơ đồ Seq Diagram “Thêm mới dự án”

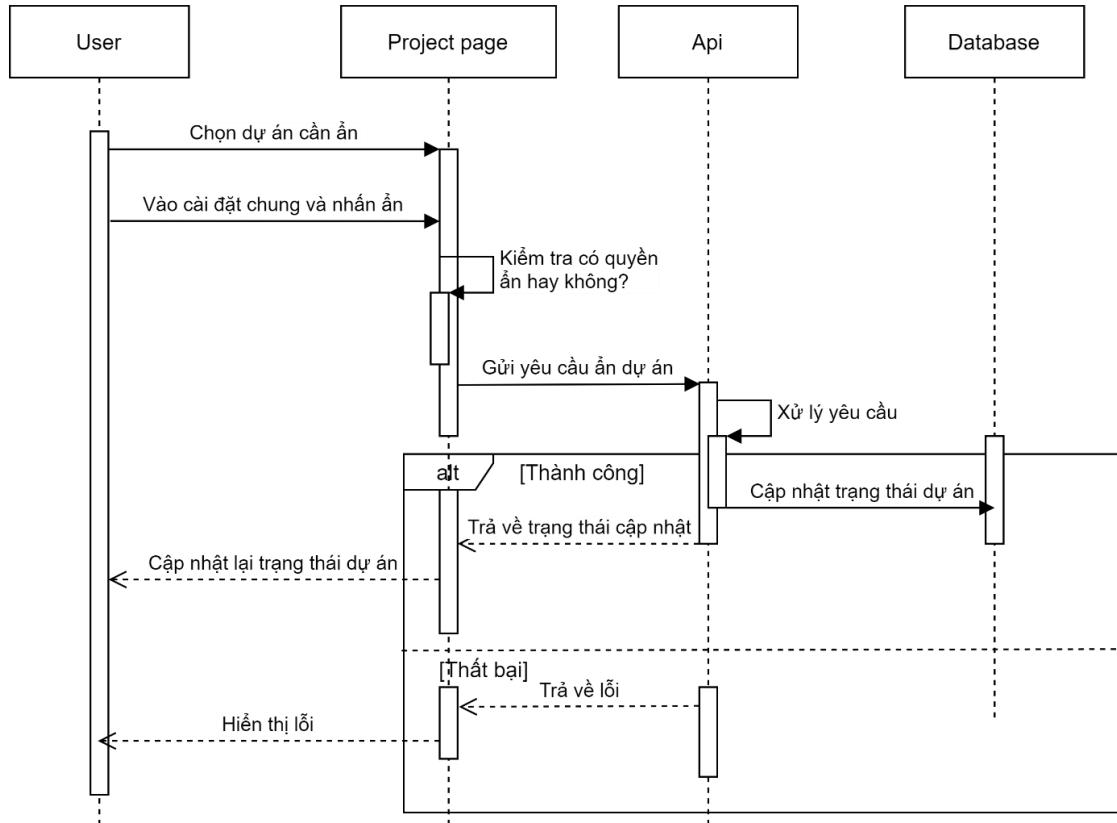
#### 4.5.5. Sơ đồ Seq Diagram “Thêm mới thành viên vào dự án”



Hình 4.11: Sơ đồ Seq Diagram “Thêm mới thành viên vào dự án”

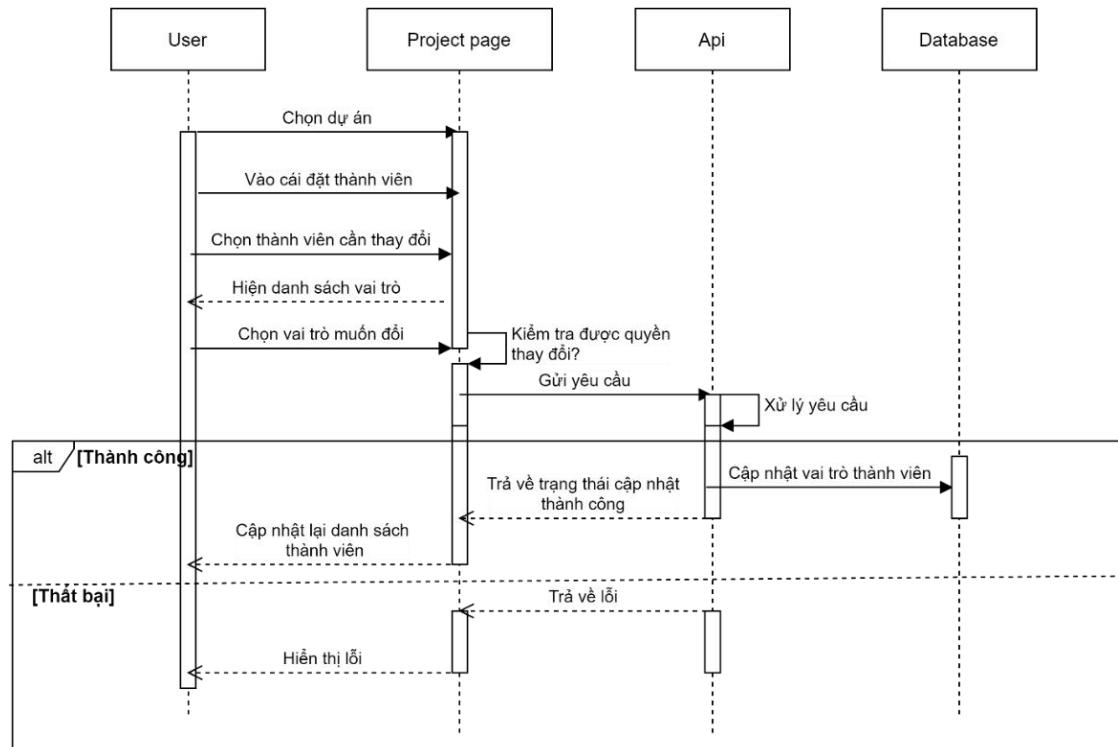
#### 4.5.6. Sơ đồ Seq Diagram “Tạm dừng giám sát dự án”

Tạm dừng giám sát dự án



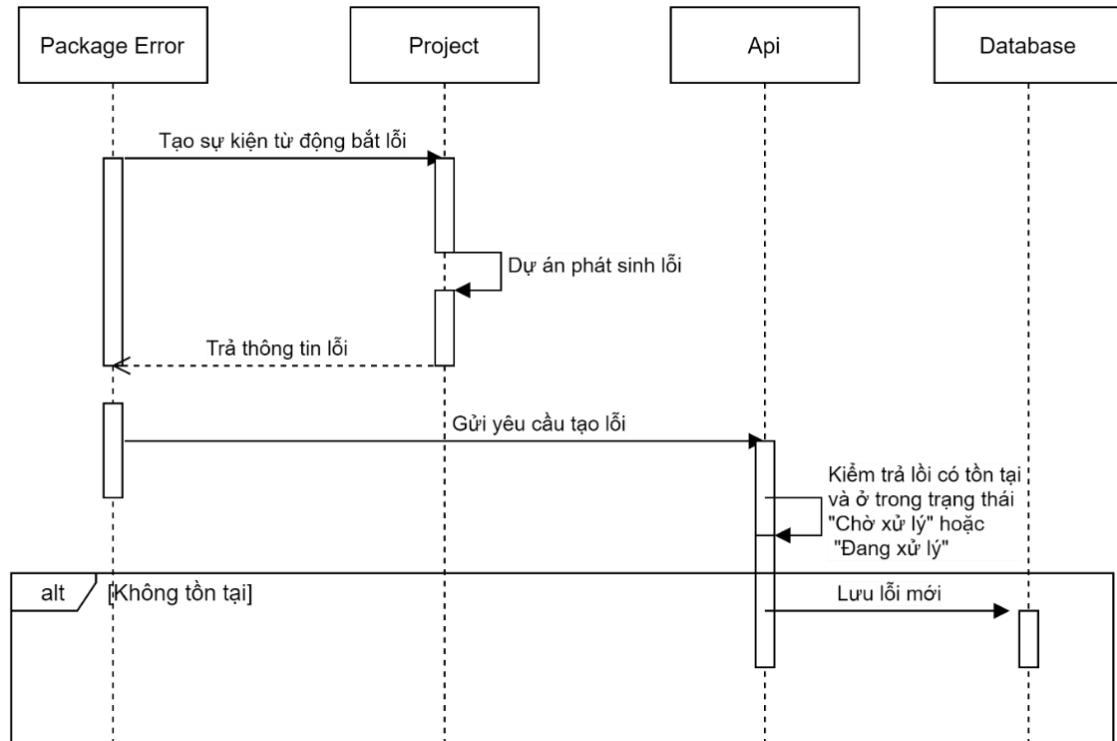
Hình 4.12: Sơ đồ seq Diagram “Tạm dừng giám sát dự án”

#### 4.5.7. Sơ đồ Seq Diagram “Thay đổi vai trò thành viên trong dự án”



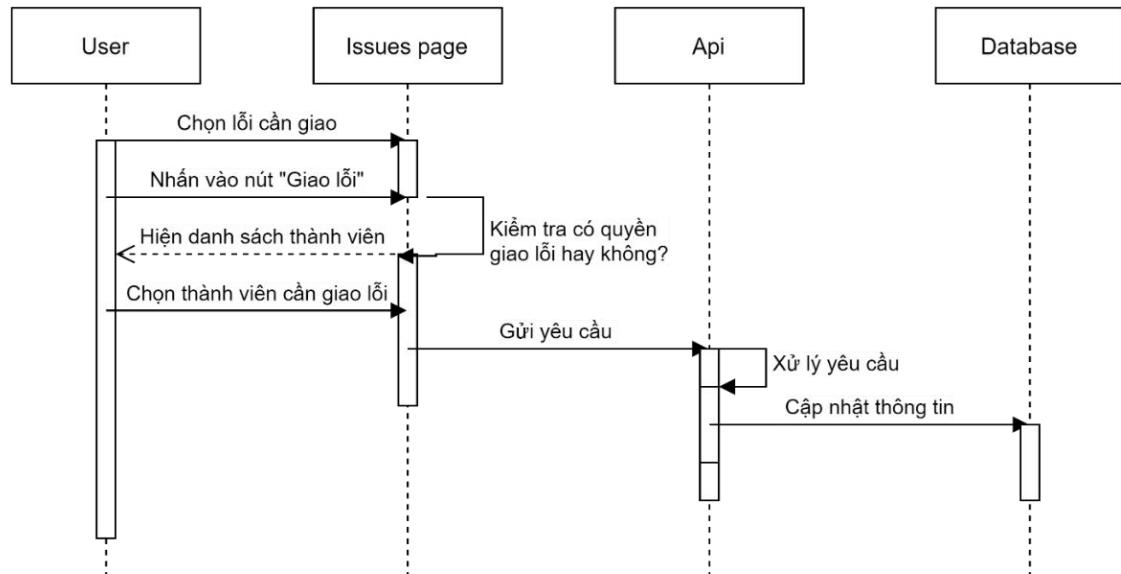
Hình 4.13: Sơ đồ Seq Diagram “Thay đổi vai trò thành viên trong dự án”

#### 4.5.8. Sơ đồ Seq Diagram “Thêm lỗi”



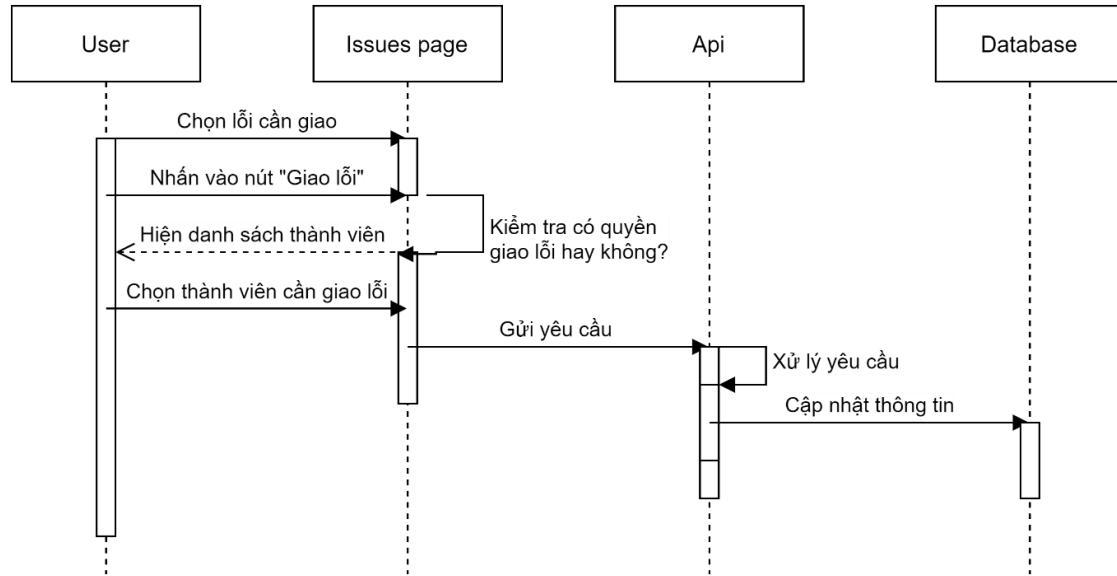
Hình 4.14: Sơ đồ Seq Diagram “Thêm lỗi”

#### 4.5.9. Sơ đồ Seq Diagram “Giao sửa lỗi”



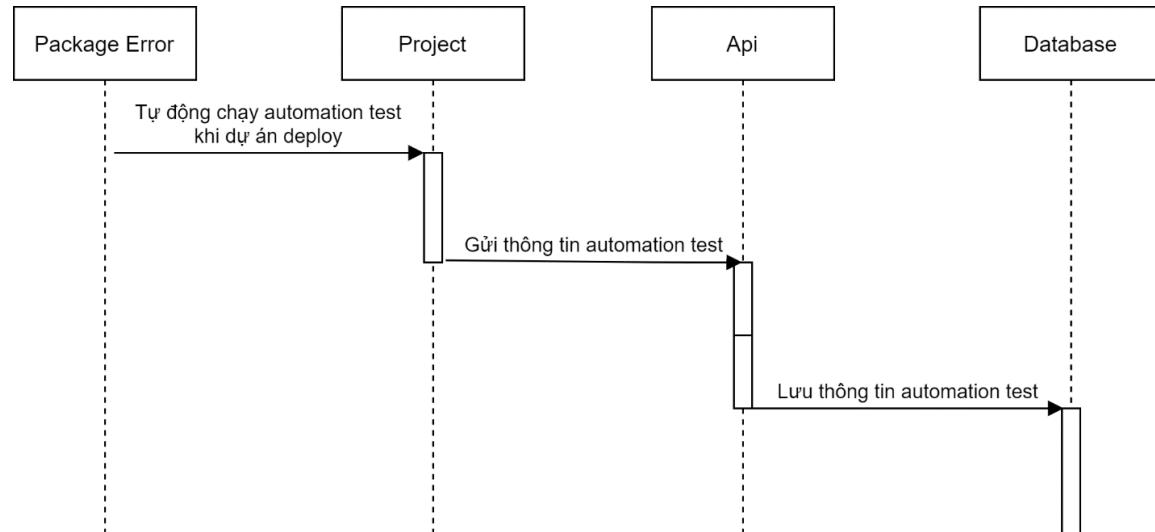
Hình 4.15: Sơ đồ Seq Diagram “Giao sửa lỗi”

#### 4.5.10. Sơ đồ Seq Diagram “Cập nhật lỗi”



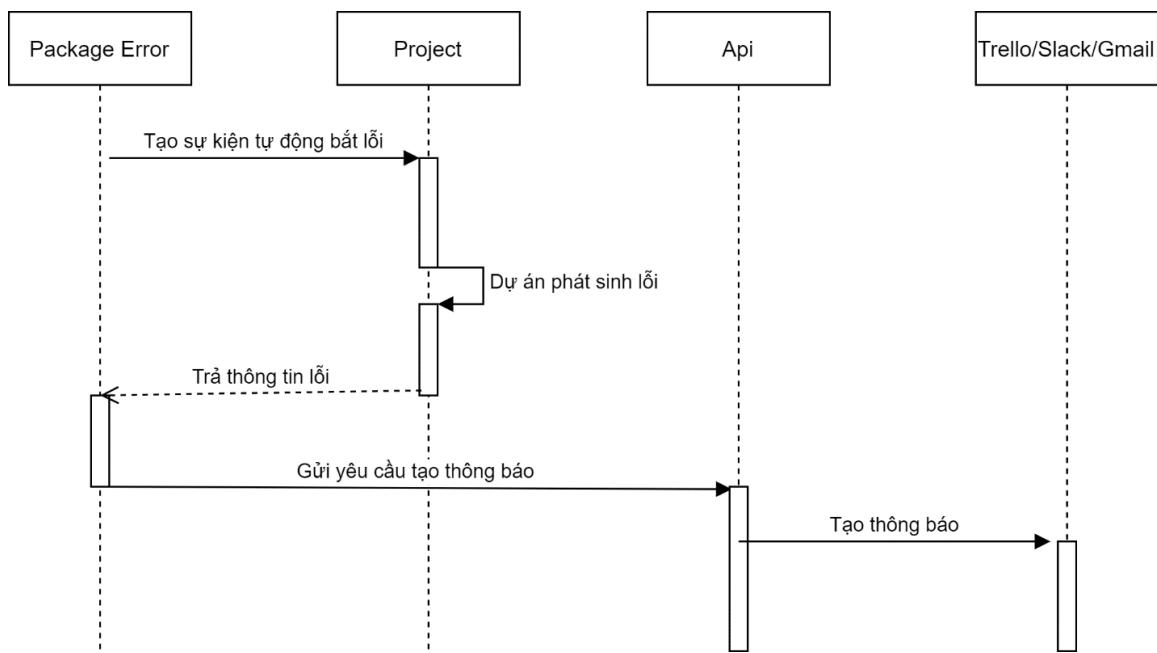
Hình 4.16: Sơ đồ Seq Diagram “Cập nhật lỗi”

#### 4.5.11. Sơ đồ Seq Diagram “Thêm automation test”



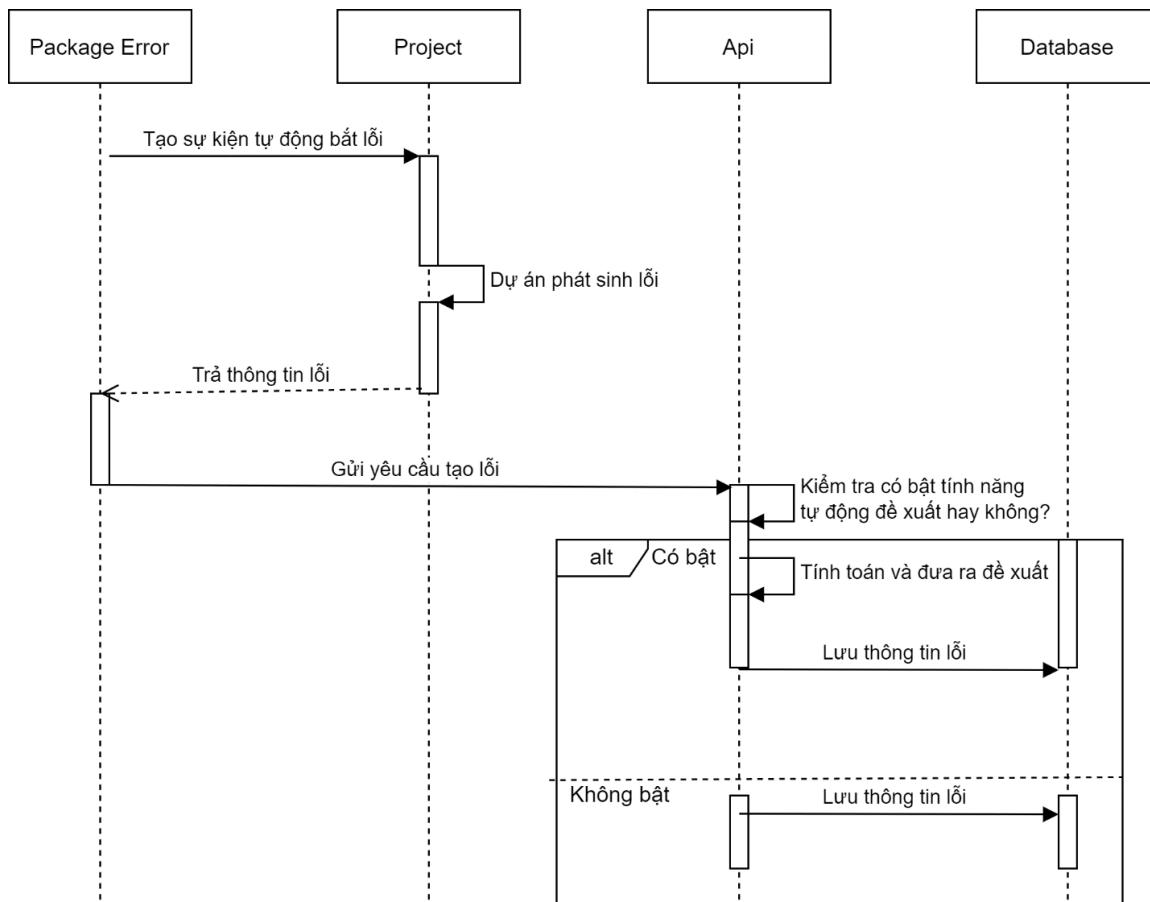
Hình 4.17: Sơ đồ Seq Diagram “Thêm Automation test”

#### 4.5.12. Sơ đồ Seq Diagram “Tích hợp thông báo Trello, Slack, Gmail



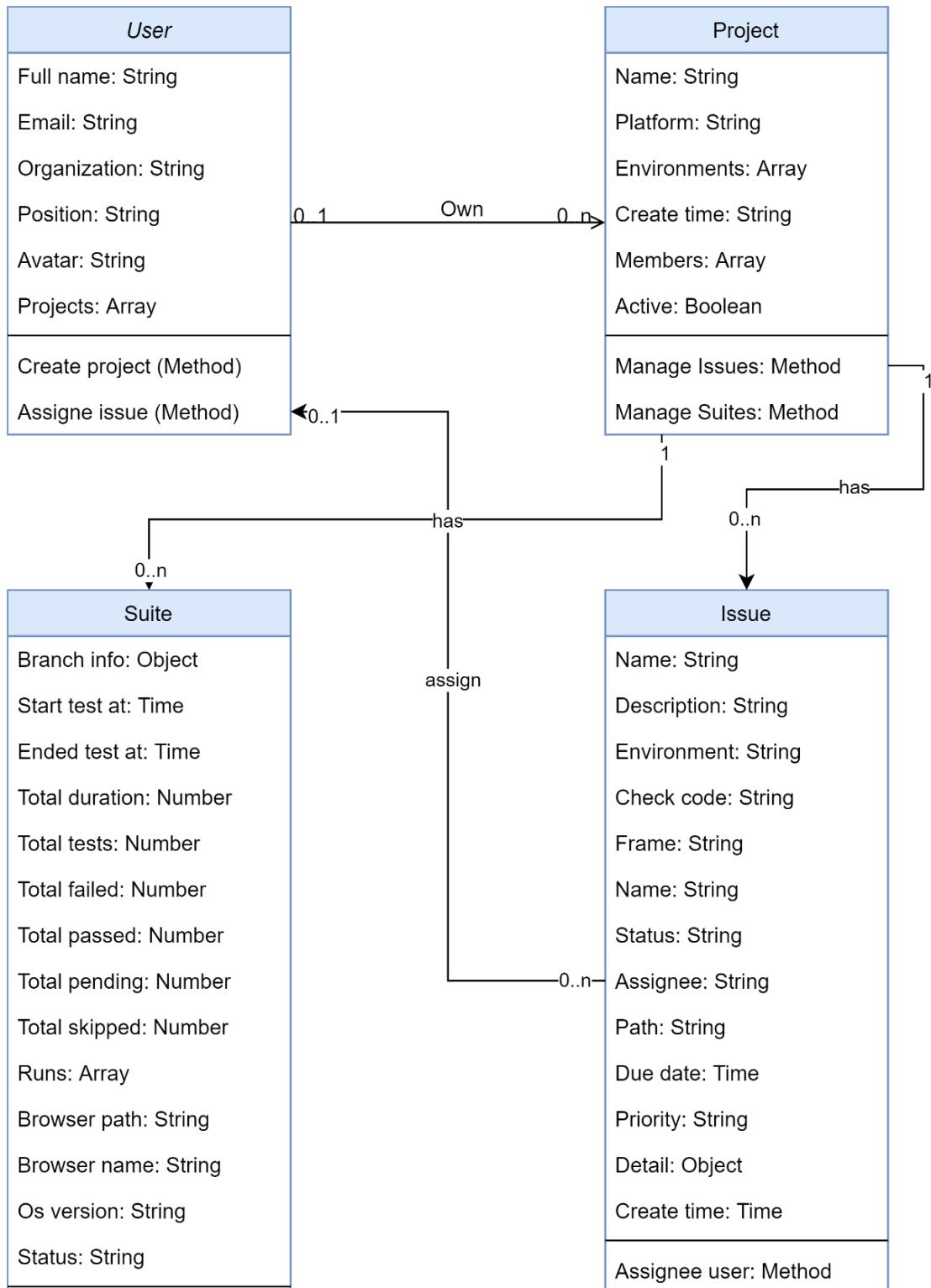
Hình 4.18: Sơ đồ Seq Diagram “Tích hợp thông báo Trello, Slack, Gmail”

#### 4.5.13. Sơ đồ Seq Diagram “Tự động đề xuất người xử lý”



Hình 4.19: Sơ đồ Seq Diagram “Tự động đề xuất người xử lý”

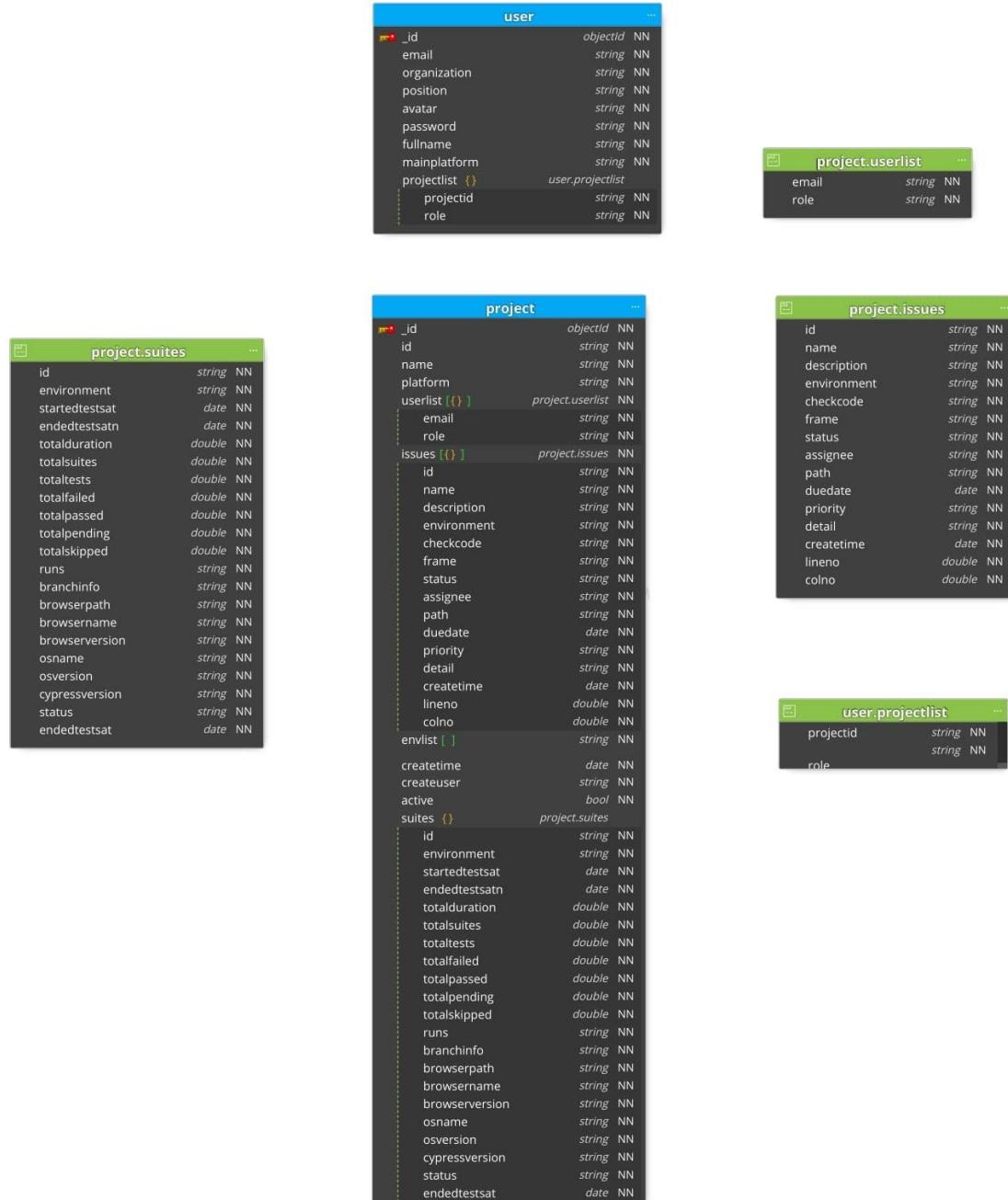
## 4.6. Sơ đồ lớp (Class diagram)



Hình 4.20: Sơ đồ lớp (Class diagram)

## 4.7. Cơ sở dữ liệu (NoSql)

### 4.7.1. Cơ sở dữ liệu tổng quát



Hình 4.21: Sơ đồ cơ sở dữ liệu tổng quan

Bảng 8: Cơ sở dữ liệu Tổng quan

STT	Tên bảng dữ liệu	Diễn giải
1	users	Lưu thông tin người dùng
2	projects	Lưu thông tin các dự án cần giám sát
3	projects.suites	Lưu trữ thông tin các automation test case của dự án
4	project.issues	Lưu danh sách lỗi của dự án

#### 4.7.2. Bảng “users”

Bảng 9: Bảng “users”

STT	Tên trường	Kiểu dữ liệu	Chú thích
1	_id	objectId	Định danh user
2	Email	String	Thông tin email
3	Organization	String	Tổ chức
4	Avatar	String	Hình ảnh đại diện của người dùng
5	Password	String	Mật khẩu
6	fullName	String	Tên người dùng
7	mainplatform	String	Nền tảng công nghệ chính
8	Projectlist	Array	Danh sách các project người dùng đã tạo

#### 4.7.3. Bảng “projects”

Bảng 10: Bảng “projects”

STT	Tên trường	Kiểu dữ liệu	Chú thích
1	_id	objectId	Định danh dự án
2	Name	String	Tên Dự án
3	Platform	String	Nền tảng công nghệ
4	UserList	Array	Danh sách thành viên
5	Issues	Array<Issue>	Danh sách lỗi
6	envList	Array	Danh sách môi trường của dự án
7	Createtime	Date	Ngày tạo dự án
8	Createuser	String	Người tạo dự án
9	Active	Boolean	Trạng thái dự án
10	Suites	Array	Danh sách automation test case
11	SlackInfo	Object	Thông tin Slack để tích hợp thông báo
12	TrelloInfo	Object	Thông tin Trello để tích hợp thông báo

#### 4.7.4. Bảng “suites”

Bảng 11: Bảng “suites”

STT	Tên trường	Kiểu dữ liệu	Chú thích
1	_id	objectId	Định danh suite
2	Environment	String	Môi trường
3	startedtestatn	Date	Ngày bắt đầu automation test
4	Endedtestsatn	Date	Ngày kết thúc automation test

5	totalduration	String	Tổng thời gian chạy test
6	totalsuites	double	Tổng automation test
7	totaltests	double	Tổng test case
8	totalfailed	double	Tổng test case lỗi
9	totalpassed	double	Tổng test case thành công
10	totalpending	double	Tổng test case tạm dừng
11	totalskipped	double	Tổng test case bỏ qua
12	runs	String	Chi tiết của automation test
13	branchinfo	String	Thông tin commit
14	browserpath	String	Đường dẫn trình duyệt
15	browsername	String	Tên trình duyệt chạy automation test
16	browsersversion	String	Version của trình duyệt
17	osname	String	Tên hệ điều hành
18	osversion	String	Version hệ điều hành
19	cypressversion	String	Version cypress
20	status	String	Tình trạng test case

#### 4.7.5. Bảng “issues”

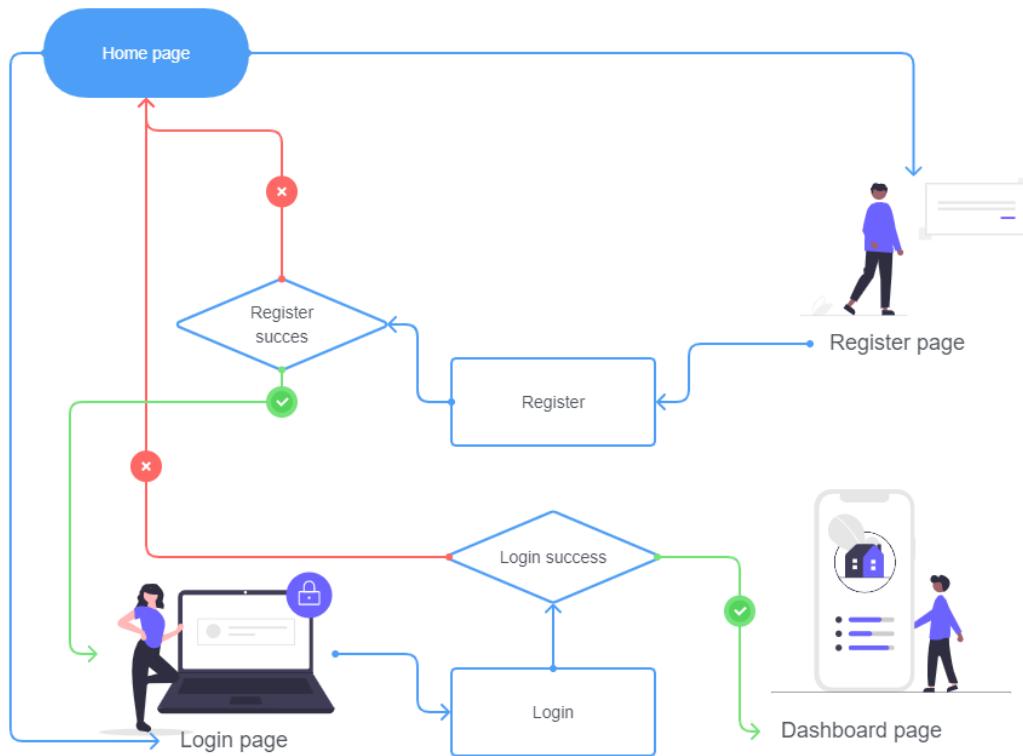
Bảng 12: Bảng “issues”

STT	Tên trường	Kiểu dữ liệu	Chú thích
1	id	objectId	Định danh lỗi
2	Name	String	Tên lỗi

3	Environment	String	Môi trường lỗi hiển thị
4	Description	String	Mã xác định lỗi đã tồn tại hay chưa
5	Checkcode	String	Lưu stack chi tiết của lỗi
6	Frame	String	Trạng thái của lỗi “Chưa xử lý”, “Đang xử lý”, “Đã xử lý”
7	Status	String	Nền tảng công nghệ chính
8	assignee	String	Danh sách các project người dùng đã tạo
9	Path	String	Đường dẫn lỗi
10	Duedate	Date	Ngày hết hạn lỗi
11	Priority	String	Mức độ nghiêm trọng
12	Detail	String	Chi tiết lỗi
13	createtime	date	Ngày tạo lỗi
14	lineno	Double	Dòng bị lỗi
15	colno	Double	Cột bị lỗi
16	Reviewer	String	Người đánh giá

## 4.8. Thiết kế User-flow

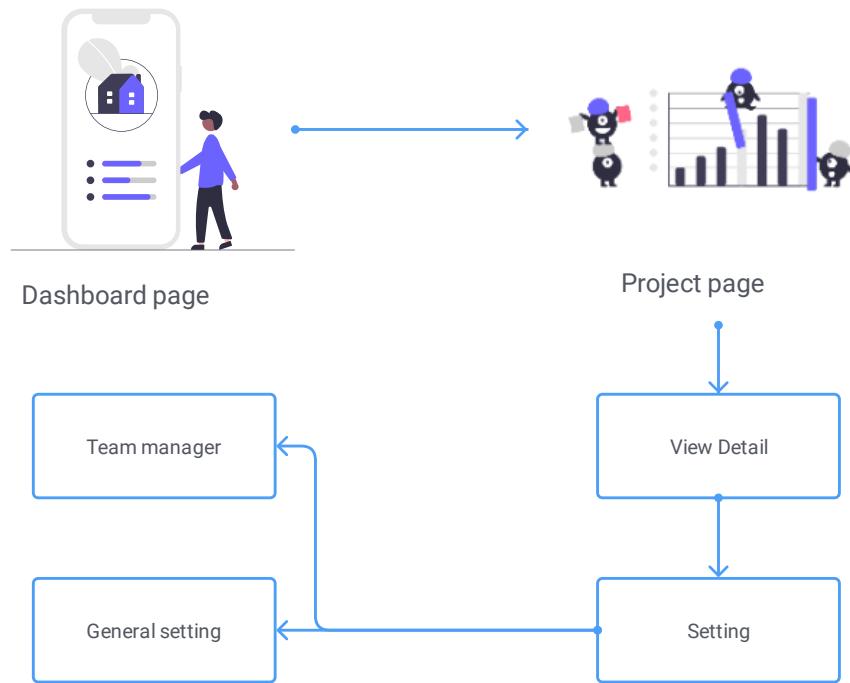
### 4.8.1. Security



09 July 2021

Hình 4.22: User-flow Security

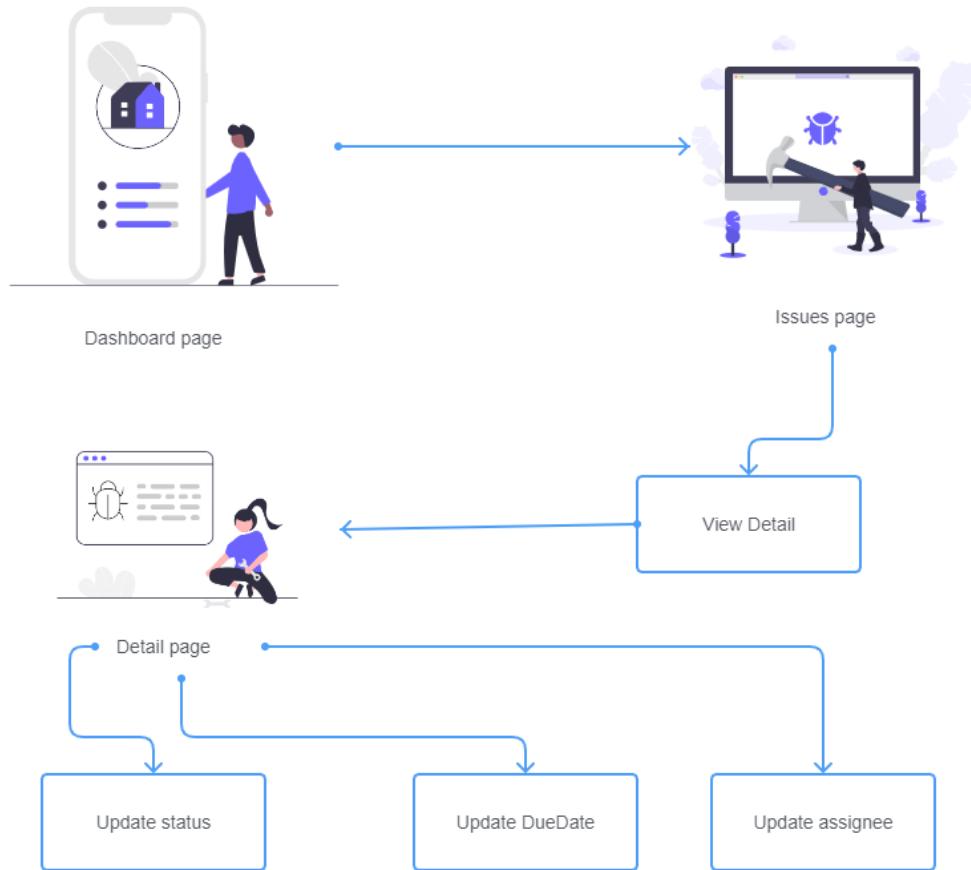
#### 4.8.2. Project manager



09 July 2021

Hình 4.23: User-flow Project manager

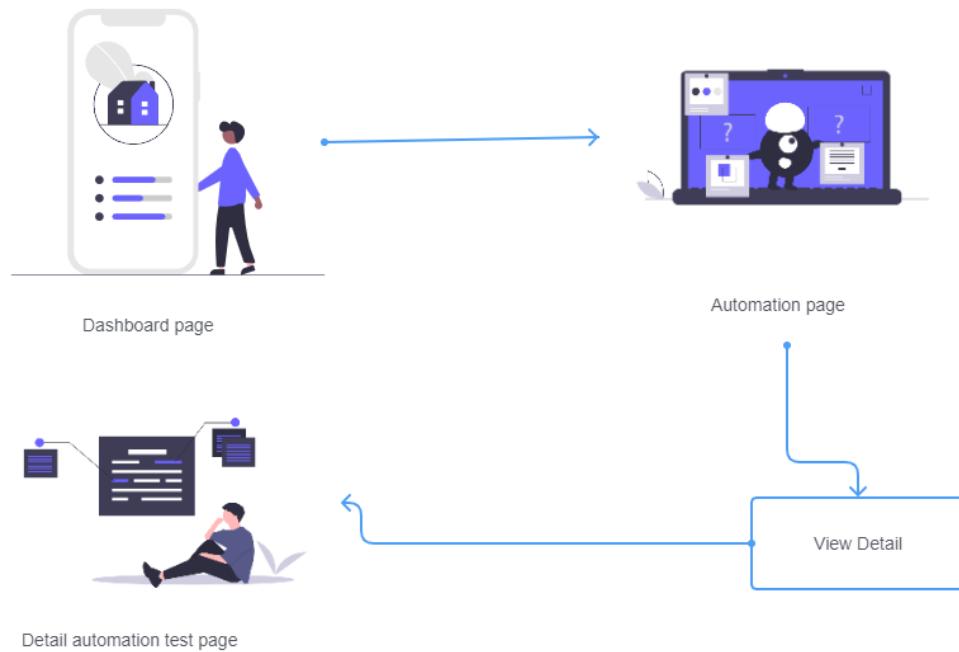
#### 4.8.3. Issues manager



09 July 2021

Hình 4.24: User-flow Issues manager

#### 4.8.4. Automation test manager

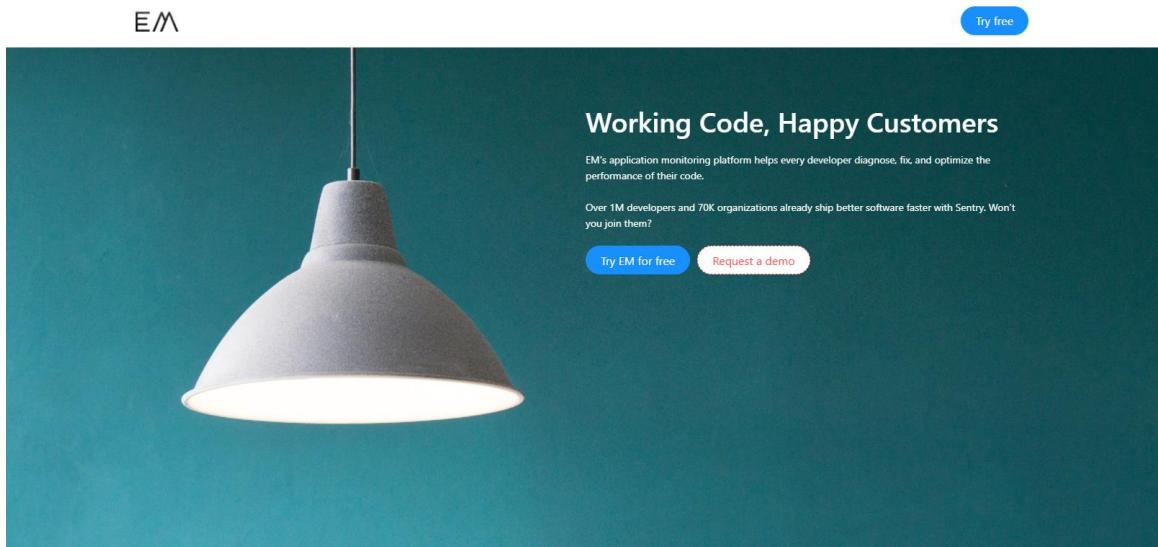


09 July 2021

Hình 4.25: User-flow Automation test manager

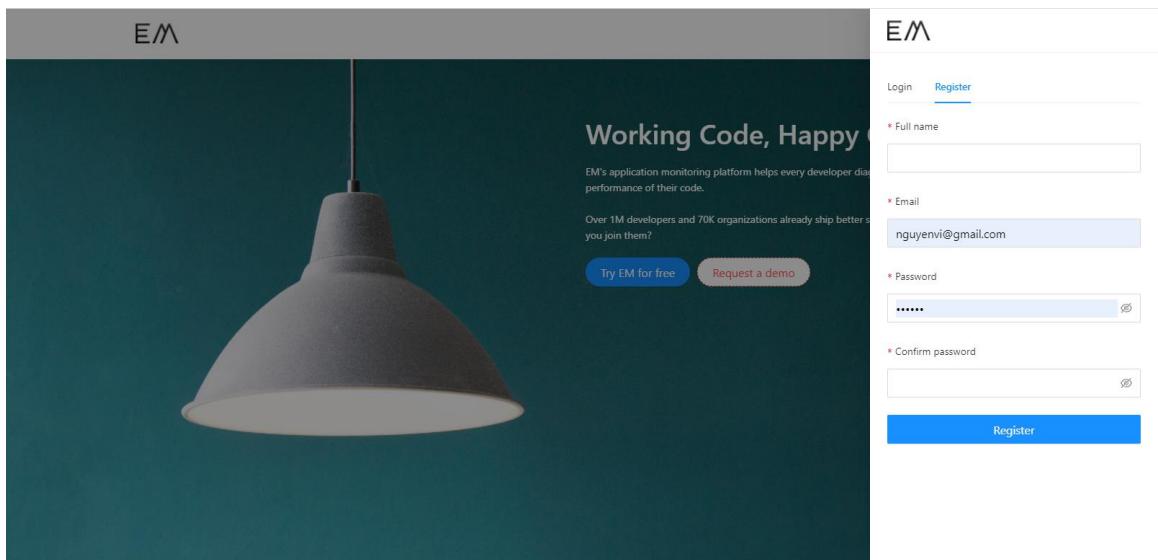
## 4.9. Giao diện

### 4.9.1. Giao diện trang giới thiệu



Hình 4.26: Giao diện trang giới thiệu

### 4.9.2. Giao diện trang đăng ký



Hình 4.27: Giao diện trang đăng ký

#### 4.9.3. Giao diện trang quên mật khẩu

---

The screenshot shows a web page titled 'EM' at the top. Below it, a section titled 'Recover Account' contains the text 'We will send a email include code to this address:'. There is a text input field labeled 'Your email' and a blue button labeled 'Send email' below it.

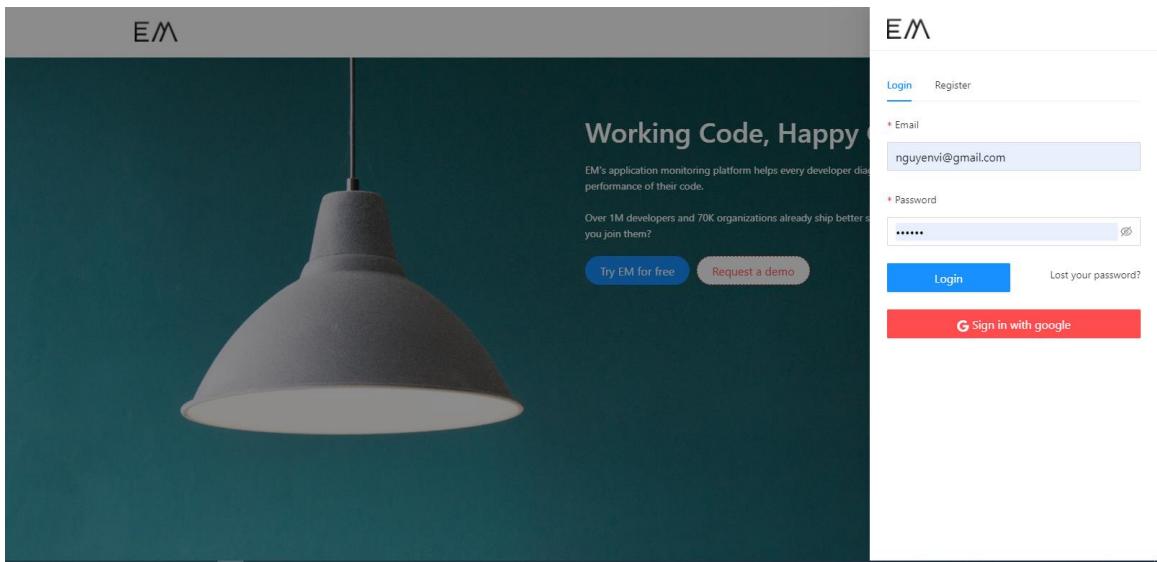
Hình 4.28: Giao diện trang nhập email cần khôi phục lại mật khẩu

---

The screenshot shows a web page titled 'EM' at the top. Below it, a section titled 'Recover Account' contains the text 'Please input your new password'. It includes three input fields: one for 'Password' and one for 'Confirm password', both with eye icon password reveal buttons. Below these is a field for 'Code' with a text input field and a blue button labeled 'Update password' at the bottom.

Hình 4.29: Giao diện trang “Nhập mật khẩu mới”

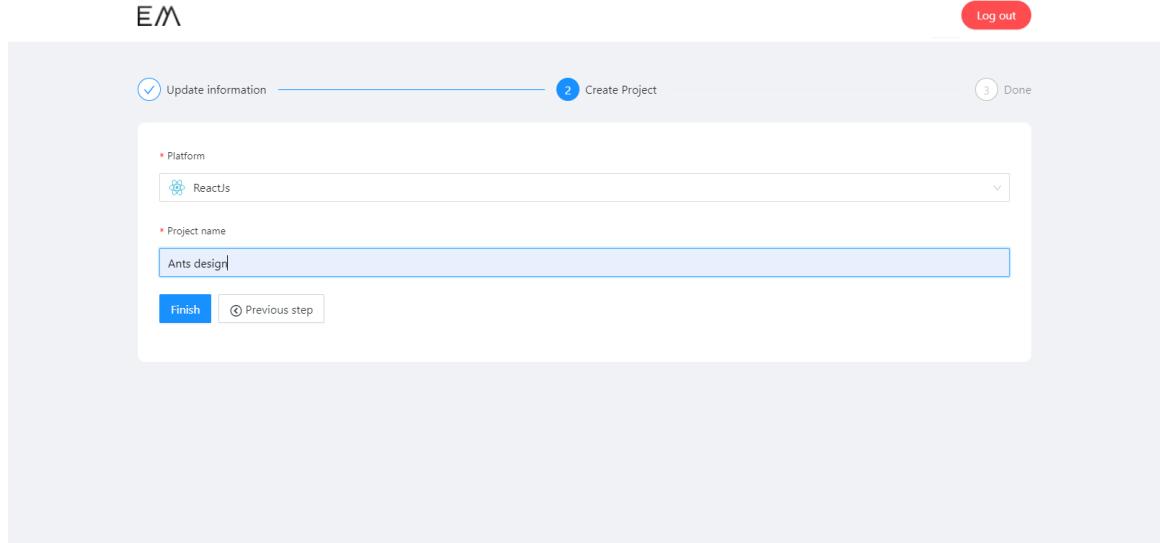
#### 4.9.4. Giao diện trang đăng nhập



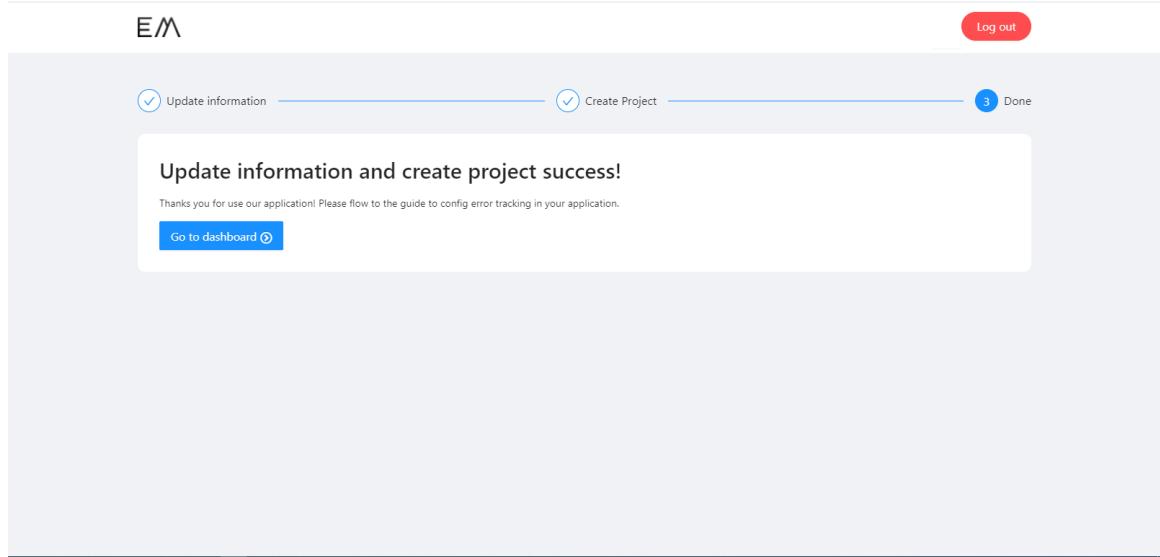
#### 4.9.5. Giao diện trang cập nhật thông tin lần đầu

A screenshot of the 'Update information' step of the onboarding process. The interface is clean with a white background. At the top, there are three circular icons labeled 1) 'Update information', 2) 'Create Project', and 3) 'Done'. The main area contains four form fields with validation stars: 'Email' (nguyenvi@work.com), 'Full Name' (Nguyen Luong truong vi), 'Organization' (Organization), and 'Main platform' (Main platform). Below the form is a blue 'Next step' button.

Hình 4.30: Giao diện bước “Cập nhật thông tin cá nhân”

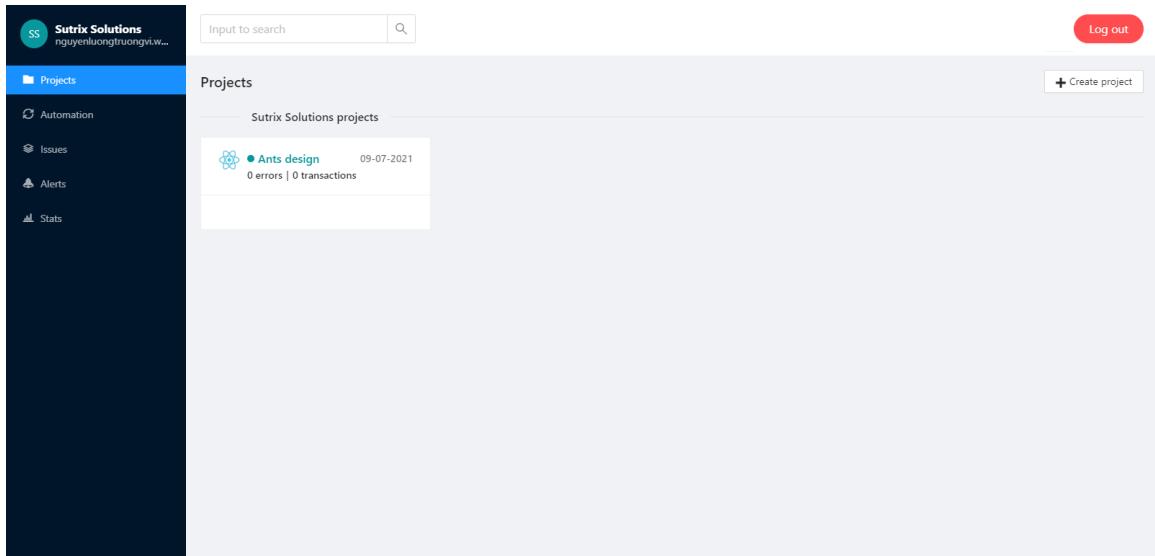


Hình 4.31: Giao diện bước “Tạo mới dự án cần giám sát”



Hình 4.32: Giao diện bước “Thông báo tạo mới và cập nhật thành công”

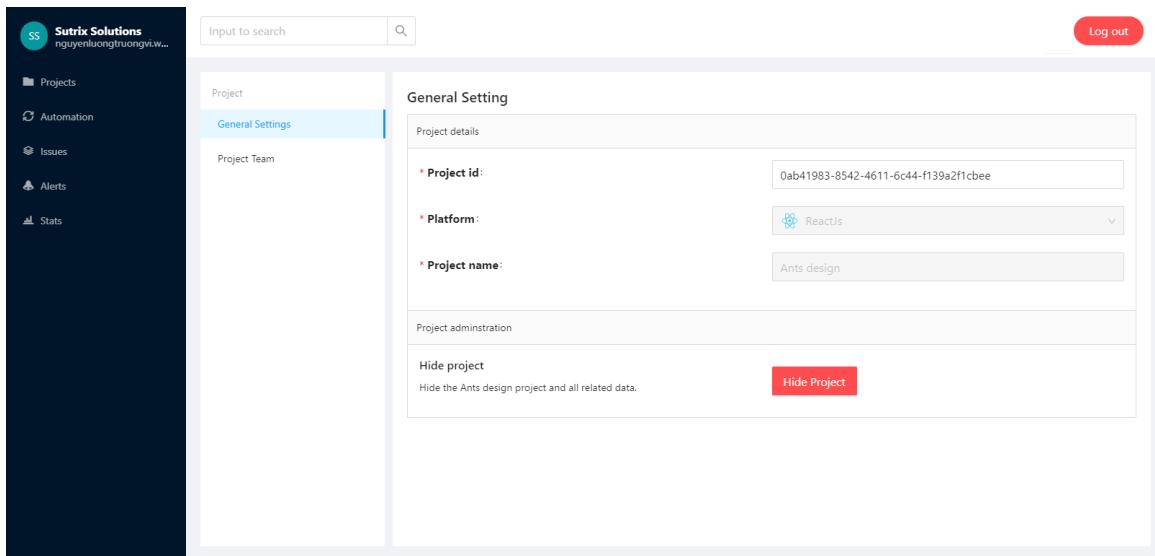
#### 4.9.6. Giao diện trang dự án



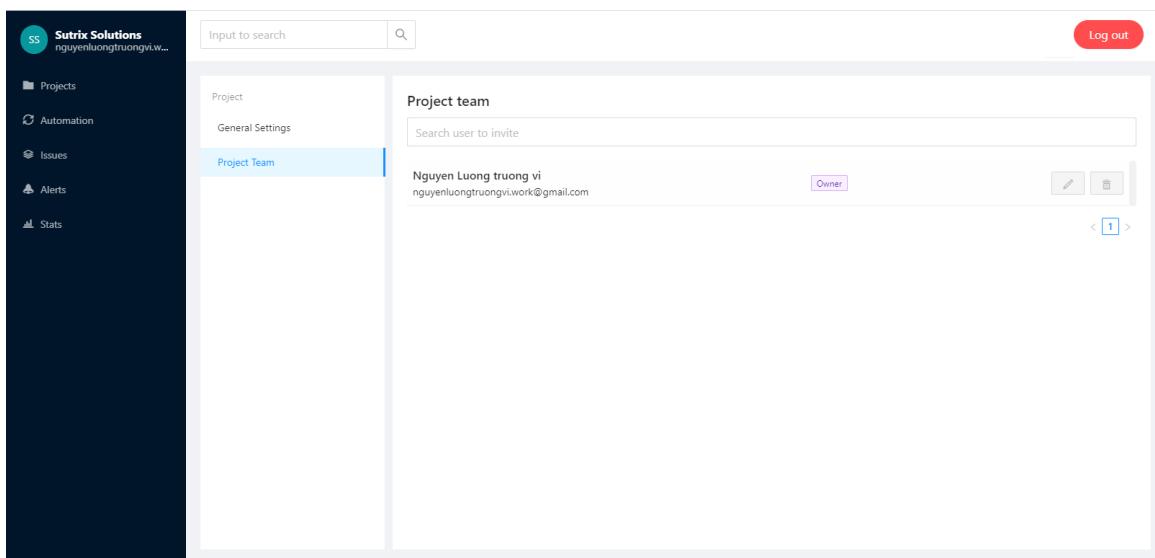
Hình 4.33: Giao diện trang dự án



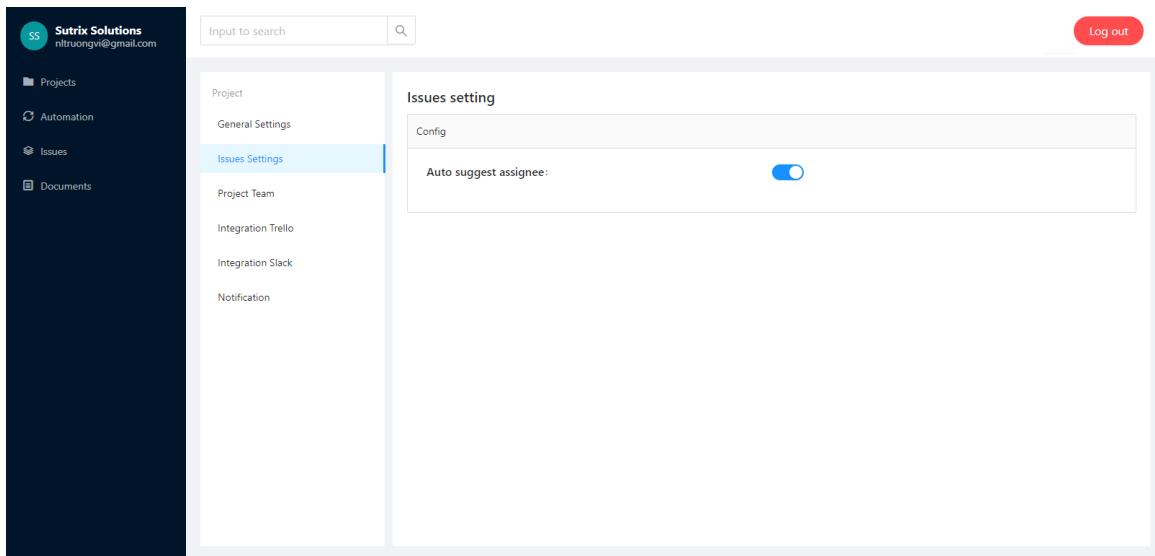
Hình 4.34: Giao diện trang chi tiết dự án



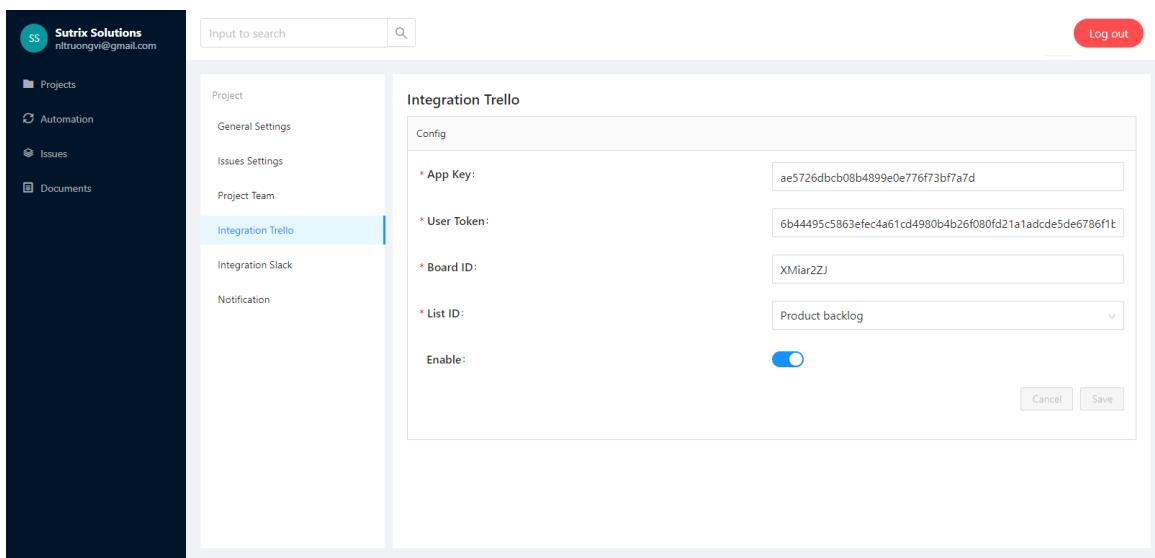
Hình 4.35: Giao diện trang cài đặt chung dự án



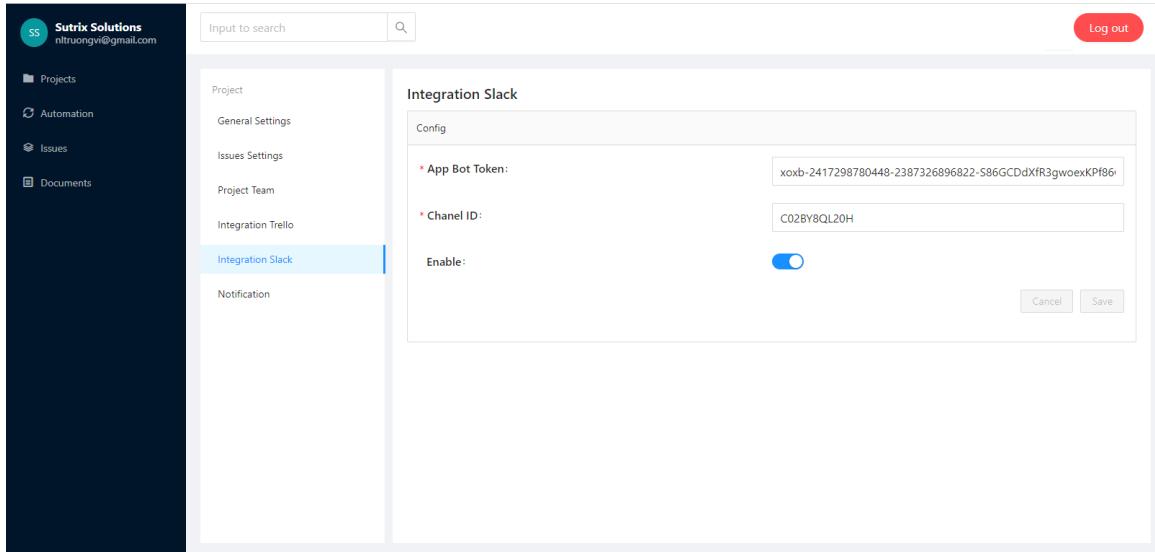
Hình 4.36: Giao diện trang cập nhật thành viên dự án



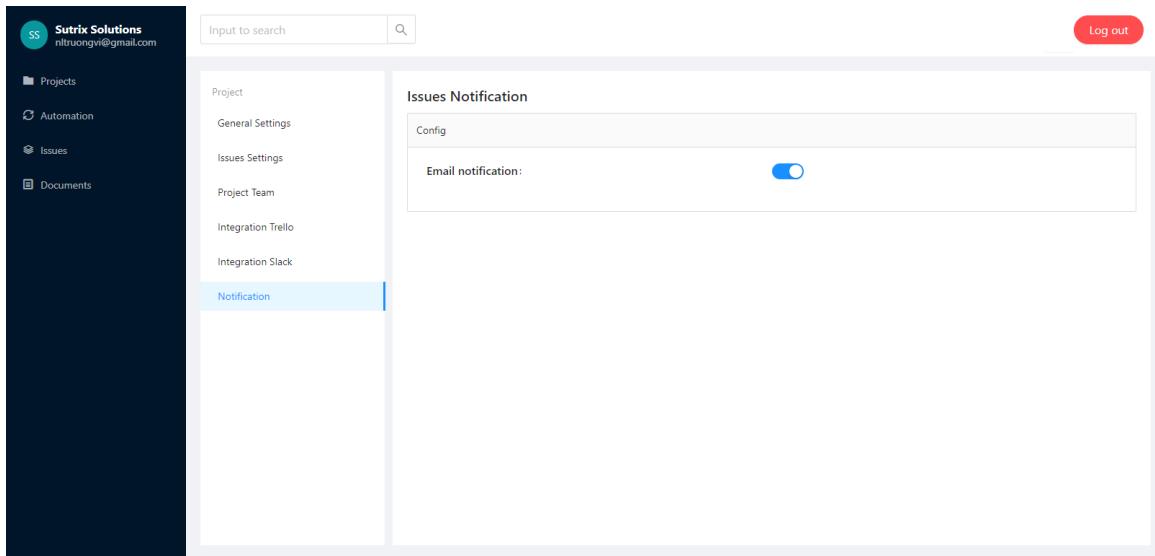
Hình 4.37: Giao diện trang bật tắt tính năng đề xuất người xử lý



Hình 4.38: Giao diện trang tích hợp thông báo trello



Hình 4.39: Giao diện trang tích hợp thông báo Slack



Hình 4.40: Giao diện trang tích hợp thông báo Email

#### 4.9.7. Giao diện trang danh sách lỗi

Hình 4.41: Giao diện trang danh sách lỗi

```

Error: A cross-origin error was thrown. React doesn't have access to the actual error object in development. See https://reactjs.org/link/crossorigin-error for more information.
    at Object.invokeGuardedCallbackDev (http://localhost:8080/index.js:7763:19)
    at invokeGuardedCallback (http://localhost:8080/index.js:7814:31)
    at invokeGuardedCallbackAndCatchFirstError (http://localhost:8080/index.js:7828:25)
    at executeDispatch (http://localhost:8080/index.js:12001:3)
    at processDispatchQueueItemsInOrder (http://localhost:8080/index.js:12033:7)
    at processDispatchQueue (http://localhost:8080/index.js:12046:5)
    at dispatchEventsForPlugins (http://localhost:8080/index.js:12057:3)
    at http://localhost:8080/index.js:12066:17
  
```

Hình 4.42: Giao diện trang chi tiết lỗi

#### 4.9.8. Giao diện trang danh sách automation test

The screenshot shows a web-based project management interface. On the left is a dark sidebar with navigation links: 'Project 1' (with the email 'nguyenenvi@gmail.com'), 'Projects', 'Automation' (which is highlighted in blue), 'Issues', 'Alerts', and 'Stats'. The main content area has a light gray background. At the top, there's a search bar with placeholder text 'Input to search' and a magnifying glass icon. To the right of the search bar is a red 'Log out' button. Below the search bar, the title 'Automation' is displayed in bold. Underneath it is a search/filter section with dropdown menus for 'No pa hello' and 'All', and date range fields for 'Start date' and 'End date'. A small calendar icon is also present. The main list area contains four items, each representing a different automation run:

- Update project**  
ntruongvi • Ran a month ago • 00:00:02 • master  
Status: 0 green, 0 yellow, 2 red
- Update project**  
ntruongvi • Ran a month ago • 00:00:03 • master  
Status: 0 green, 0 yellow, 2 red
- Initialize project using Create React App**  
Nguyen Luong Truong Vi • Ran a month ago • 00:00:40 • master  
Status: 0 green, 0 yellow, 8 red
- Initialize project using Create React App**  
Nguyen Luong Truong Vi • Ran a month ago • 00:00:32 • master  
Status: 0 green, 0 yellow, 8 red

Hình 4.43: Giao diện trang danh sách automation test

This screenshot shows a detailed view of automation test results. The left sidebar is identical to the one in Figure 4.43. The main content area has a light gray background. At the top, there's a search bar with placeholder text 'Input to search' and a magnifying glass icon. To the right of the search bar is a red 'Log out' button. Below the search bar, the title 'Dashboard / Automation / Suite' is displayed. Underneath it is a list item:  
**Initialize project using Create React App**  
Nguyen Luong Truong Vi • Ran a month ago • 00:00:40 • master  
Status: 0 green, 0 yellow, 8 red

Below this, a section titled 'Test results (8)' is shown. It lists two test files with their results:

- pages/login/login.spec.js**  
4 green, 0 yellow, 0 red, 0 error  
Test login social chat → Require email and password  
Test login social chat → Check validate email  
Test login social chat → Check login failed  
Test login social chat → Check login success
- pages/register/regiser.spec.js**  
4 green, 0 yellow, 0 red, 0 error

Hình 4.44: Giao diện trang chi tiết automation test

#### 4.9.9. Giao diện trang tài liệu hướng dẫn

The screenshot shows a web-based documentation page. At the top left is a dark sidebar with a user icon and the text 'Sutrix Solutions nlttruongv@gmail.com'. To the right is a search bar with placeholder text 'Input to search' and a magnifying glass icon. On the far right is a red 'Log out' button. The main content area has a light gray background. At the top of this area is a navigation bar with four items: 'Overview' (gray), 'Install package' (blue, indicating the current section), 'Integration Slack', and 'Integration Trello'. Below this is a section titled 'Instructions for installing packages into the application'. It contains a sub-section titled 'Install' with the following code examples:

```
# Using npm
npm i @error-monitoring/package
# Using yarn
yarn add @error-monitoring/package
```

Below the 'Install' section is another sub-section titled 'Configure' with the following configuration code:

```
import React from "react";
import ReactDOM from "react-dom";
import Error from "@error-monitoring/package";
import App from "./App";

Error.init({
  projectId: "your-project-id",
  env: "development", // There are three config env: development, staging, production
});

ReactDOM.render(, document.getElementById("root"));
```

Hình 4.45: Giao diện trang tài liệu hướng dẫn

## **Chương 5. ÁP DỤNG THUẬT TOÁN ĐỀ XUẤT NGƯỜI PHỤ TRÁCH LỖI**

### **5.1. Đặt vấn đề**

Các issues được tạo ra trong hệ thống và những người quản lý gặp khó khăn trong việc phân chia cho nhân viên phụ trách. Vì vậy nhóm đã đưa ra giải pháp hỗ trợ đề xuất người phụ trách công việc, chức năng này sẽ giúp người quản lý chọn được nhân viên phù hợp dựa trên các tiêu chí sau:

- Có thể bắt đầu công việc đó sớm nhất
- Có thể xử lý công việc đó nhanh nhất
- Có thể cân bằng lượng công việc của các nhân viên.
- Có thể cân bằng thời gian xử lý toàn bộ các lỗi của dự án.
- Có thể nhận diện được nhân viên nào có liên quan đến công việc mới, từ đó giảm thiểu được thời gian xử lý công việc mới.

### **5.2. Giải pháp**

Giải pháp Đề xuất người phụ trách hoạt động với điều kiện người dùng đã kích hoạt tính năng này trên mục cài đặt (Hình 0.46: Giao diện trang bật tắt tính năng đề xuất người xử lý).

Tính năng này gồm 2 giai đoạn : Tính toán và đưa ra gợi ý.

#### **5.2.1. Tính toán**

Sau khi người dùng kích hoạt tính năng tự động đề xuất thì hệ thống sẽ thực hiện 2 công việc: Tính toán ước tính năng suất của mỗi nhân viên, ước lượng thời gian sẵn sàng cho công việc mới.

Các đơn vị phục vụ cho tính toán:

Thuộc tính	Ký hiệu viết tắt	Tập giá trị	Giá trị mặc định	Chú thích
------------	------------------	-------------	------------------	-----------

Status	<b>S</b>	Unresolve, Processing, Reviewing, Resolved	Unresolve	Tương ứng với tỷ lệ hoàn thành công việc đó: chưa xử lý - 0 %, đang xử lý - 50 %, đang đánh giá lại - 90 %, đã hoàn thành - 100 %.
Duedate	<b>D</b>		Ngày tháng năm	Thời hạn để xử lý công việc đó.
Priority	<b>P</b>	Low (1), Medium (2), High (3)	0	các mức độ ưu tiên cho công việc đó.
CreateTime	<b>C</b>		Ngày tháng năm	Thời gian bắt đầu thực hiện công việc đó.
Enviroment	<b>E</b>	Dev (1), Staging (1.5), Product (2)	dev	Môi trường xảy ra sự cố đó.
Author	<b>A</b>	Yes/No	No	Nhận diện nhân viên nào thực hiện công việc đó.

#### 5.2.1.1. Tính toán năng suất của mỗi nhân viên

**Mục đích:** Tính toán tương đối thời gian hoàn thành 1 công việc cơ bản của nhân viên đó từ đó làm cơ sở ước lượng thời gian hoàn thành các công việc mới.

**Điều kiện kích hoạt:** Khi nhân viên có phát sinh cập nhật tình trạng công việc(trong màn hình Issues).

**Phạm vi dữ liệu :** tất cả các công việc mà nhân viên đó đã hoàn thành (bảng Issues).

**Công thức tính toán:**

$$K = \frac{\sum_1^n (D-C) / (P \times E)}{n}$$

### **Giải thích công thức:**

D – C: là khoảng thời gian xử lý công việc đó. Sử dụng đơn vị là giờ hành chính.

P x E: tích của độ ưu tiên và môi trường. Là đại lượng tương đối của độ phức tạp của công việc đó.

n: tổng số lượng công việc đã hoàn thành

K: Trung bình thời gian hoàn thành 1 công việc cơ bản.

#### **5.2.1.2. Ước lượng thời gian sẵn sàng cho công việc xử lý lỗi mới**

Sau khi đã tính toán năng lực của nhân viên đó, thuật toán sẽ tiến hành ước tính thời gian sẵn sàng cho công việc mới của nhân viên đó

Mục đích: tính toán thời gian hoàn thành các công việc đang được giao sớm nhất, từ đó làm cơ sở để gợi ý nhân viên phù hợp.

**Điều kiện kích hoạt:** Khi nhân viên có phát sinh cập nhật tình trạng công việc (trong màn hình Issues).

Phạm vi dữ liệu: tất cả các công việc mà nhân viên đó chưa hoàn thành (bảng Issues).

### **Công thức tính toán:**

$$T = \sum_1^n P \times E \times K \times (100 - S)$$

### **Giải thích công thức:**

P x E: Tích của độ ưu tiên và môi trường. Là đại lượng tương đối của độ phức tạp của công việc đó.

K: Trung bình thời gian hoàn thành 1 công việc cơ bản.

100 – S: Tiến độ còn lại của công việc đang làm.

n: Tổng số lượng công việc đã hoàn thành

T: Thời gian mà nhân viên đó sẽ hoàn thành tất cả các công việc đang được giao.

### **5.2.2. Đưa ra gợi ý**

Ở giai đoạn này, hệ thống sẽ sử dụng các tính toán từ trước để đề xuất nhanh nhân viên phù hợp nhất

Mục đích: đưa ra đề xuất nhân viên phụ trách công việc mới phù hợp nhất.

Điều kiện kích hoạt: Khi hệ thống khách hành phát sinh lỗi và được gửi đến hệ thống (trong màn hình Issues).

Nguyên lý hoạt động: hệ thống sẽ sử dụng thời gian đã tính toán của từng nhân viên(T). So sánh và chọn người có (T) nhỏ nhất và tự động gán tên cho nhân viên đó phụ trách.

Ưu tiên nhân viên: Hệ thống nhận diện được nhân viên đã từng có liên quan đến lỗi phát sinh đó, từ đó sẽ ưu tiên cho nhân viên đó xử lý công việc đó. Hệ thống sẽ ưu tiên bằng cách giảm trực tiếp thông số (T) cho nhân viên đó đi 0.5(K), như vậy sẽ tăng khả năng cho nhân viên đó nhận được công việc này.

## **Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1. Kết quả đạt được**

Sau thời gian thực hiện đề tài, chương trình đã hoàn thành và đạt được một số kết quả sau:

- Hiểu rõ hơn về các giai đoạn phân tích, xác định yêu cầu, thiết kế và triển khai một dự án phần mềm và các công cụ hỗ trợ.
- Hiểu rõ quy trình quy trình quản lý lỗi dự án.
- Giao diện của chương trình thân thiện, dễ sử dụng đã có cải tiến rất nhiều so với các đồ án trước (reactjs, golang, mongoose)
- Tìm hiểu được nhiều công nghệ hỗ trợ cho việc lập trình web
- Hiểu hơn về việc phân chia và làm việc nhóm một cách hiệu quả.

### **6.2. Hạn chế và khó khăn**

- Vì phải học thêm công nghệ mới nên trong quá trình vừa phải làm và research nên thời gian làm dự án bị kéo dài so với dự kiến
- Làm việc online nên năng suất không được tối đa, nhiều vấn đề rất khó giải quyết qua meeting.
- Không có chi phí để thuê server nên phải dùng server miễn phí nên tốc độ chậm dẽ chệch
- Ứng dụng vẫn một số lỗi do mót bắt đầu phát triển

### **6.3. Hướng phát triển**

Nhóm dự định trong tương lai sẽ hoàn thiện và phát triển thêm các chức năng cho ứng dụng với các nội dung sau:

- Tích hợp đa ngôn ngữ.
- Tự động đề xuất hướng giải quyết lỗi.
- Tích hợp thêm các kênh thông báo mới Microsoft Team, OutLook,
- Tích hợp giám sát lỗi cho một số ngôn ngữ: GoLang, Java, Python, ...
- Cho phép đăng nhập ứng dụng bằng Google, Facebook

## TÀI LIỆU THAM KHẢO

### Tài liệu Tiếng Việt

- [1] V. X. Phong, “Golang là gì và tại sao bạn nên học Go?” 2020. [Trực tuyến]. Địa chỉ: <https://topdev.vn/blog/golang-la-gi-va-tai-sao-ban-nen-hoc-go>. [Truy cập: 07/09/2021].
- [2] Teky, “Ngôn ngữ Golang là gì? Tại sao bạn nên sử dụng ngôn ngữ này?” 2019. [Trực tuyến]. Địa chỉ: <https://teky.edu.vn/blog/ngon-ngu-golang>. [Truy cập: 06/09/2021].
- [3] N. V. Thịnh, “[Go Lang] Lập trình golang căn bản - Goroutine” 2020. [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/go-lang-lap-trinh-golang-can-ban-goroutine-djeZ1pvQKWz>. [Truy cập: 06/09/2021].
- [4] V. Trần, “Channel trong Golang là gì? So sánh Callback function và mutex lock với channel”. [Trực tuyến]. Địa chỉ: <https://topdev.vn/blog/golang-channel-la-gi>. [Truy cập: 06/09/2021].
- [5] N. D. Thuc, “Go – Mutex” 2019. [Trực tuyến]. Địa chỉ: <https://vngeeks.com/go-mutex>. [Truy cập: 12/10/2021].
- [6] Đ. Tùng, “Heroku là gì? Cách đăng ký Heroku và đưa ứng dụng lên” 2018. [Trực tuyến]. Địa chỉ: <https://wiki.tino.org/heroku-la-gi>. [Truy cập: 12/10/2021].
- [7] ViHoth Corporation, “Phần mềm GoLand - IDE mới cho phát triển Go” 2021. [Trực tuyến]. Địa chỉ: <https://vihoth.com/phan-mem-van-phong/jetbrains/phan-mem-goland-ide-moi-cho-phat-trien-go>. [Truy cập: 12/10/2021].

### Tài liệu Tiếng Anh

- [9] ReactJs, “Getting started” 2021. [Online] Available: <https://reactjs.org/docs/getting-started.html>. . [Accessed 09/10/2021].
- [8] GlobalLogic, “Source tree - Simplicity & Performance in One GUI” 2021, [Online] Available: <https://www.globallogic.com/offerings/atlassian/products/sourcetree>. [Accessed 12/11/2021].
- [9] MongoDB, “MongoDB Atlas Tutorial” 2020. [Online] Available: <https://www.mongodb.com/basics/mongodb-atlas-tutorial>. [Accessed 14/11/2021]

- [10] Jamie Birss, “10 best error monitoring tools to use in 2021 - A comparison report” 2021. [Online] Available: <https://raygun.com/blog/best-error-monitoring-tools>. [Accessed 20/11/2021]