BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

LƯU THỊ PHƯƠNG

1821050345

BÁO CÁO

THỰC TẬP DOANH NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: TIN HỌC KINH TẾ

ĐỀ TÀI:

Kiểm thử và đảm bảo chất lượng Phần mềm quản lý văn bằng chứng chỉ

HÀ NỘI, 5/ 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

BÁO CÁO

THỰC TẬP DOANH NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: TIN HỌC KINH TẾ

ĐỀ TÀI:

Kiểm thử và đảm bảo chất lượng Phần mềm quản lý văn bằng chứng chỉ

Sinh viên thực tập : Lưu Thị Phương

Mã sinh viên : 1821050345

Công ty thực tập : Công ty Cổ phần đầu tư phần mềm Novi

Người phụ trách : Anh Lại Đức Thuần

Giảng viên hướng dẫn: Cô Lê Thanh Huệ

HÀ NỘI, 5/ 2022.

Mục lục

[Mục lục 1](#_Toc103597621)

[DANH MỤC HÌNH VẼ BẢNG BIỂU 3](#_Toc103597622)

[MỞ ĐẦU 4](#_Toc103597623)

[1 .CHƯƠNG 1: GIỚI THIỆU VỀ CÔNG TY THỰC TẬP 5](#_Toc103597624)

[1.1 Giới thiệu tổng quan về công ty 5](#_Toc103597625)

[1.2 Vị trí thực tập tại công ty 6](#_Toc103597626)

[1.3 Đề tài thực tập 6](#_Toc103597627)

[1.4 Lịch làm việc 7](#_Toc103597628)

[2 CHƯƠNG 2: NỘI DUNG THỰC TẬP 8](#_Toc103597629)

[2.1 .Các khái niệm cơ bản về kiểm thử phần mềm 8](#_Toc103597630)

[2.1.1 .Kiểm thử hần mềm là gì? 8](#_Toc103597631)

[2.1.2 .Lỗi phần mềm là gì? Nguyên nhân sinh ra lỗi của phần mềm. 8](#_Toc103597632)

[2.1.3 Vai trò của kiểm thử phần mềm 9](#_Toc103597633)

[2.2 Quy trình kiểm thử phần mềm 9](#_Toc103597634)

[2.2.1 Lập kế hoạch kiểm thử ( Test plan) 10](#_Toc103597635)

[2.2.2 Chuẩn bị kiểm thử 10](#_Toc103597636)

[2.2.3 Thực hiện kiểm thử 10](#_Toc103597637)

[2.2.4 Viết báo cáo kiểm thử (Test report) 10](#_Toc103597638)

[2.3 Các mức kiểm thử (Test Level) 11](#_Toc103597639)

[2.3.1 Kiểm thử đơn vị (Unit test) 11](#_Toc103597640)

[2.3.2 Kiểm thử tích hợp (Intergration testing) 12](#_Toc103597641)

[2.3.3 Kiểm thử hệ thống (System test) 13](#_Toc103597642)

[2.3.4 Kiểm thử chấp nhận (Acceptance test) 14](#_Toc103597643)

[2.3.5 Kiểm thử hồi qui 14](#_Toc103597644)

[2.4 Các kỹ thuật kiểm thử phần mềm 15](#_Toc103597645)

[2.4.1 Phương pháp kiểm thử hộp đen 15](#_Toc103597646)

[2.4.2 Phương pháp kiểm thử hộp trắng 16](#_Toc103597647)

[2.4.3 Phương pháp kiểm thử hộp xám 18](#_Toc103597648)

[2.5 Các kỹ thuật kiểm thử 18](#_Toc103597649)

[2.5.1 Kỹ thuật phân vùng tương đương 19](#_Toc103597650)

[2.5.2 Bảng quyết định 19](#_Toc103597651)

[2.5.3 Kỹ thuật phân tích giá trị biên 20](#_Toc103597652)

[2.5.4 Kỹ thuật đoán lỗi 20](#_Toc103597653)

[2.6 Phân loại kiểm thử 21](#_Toc103597654)

[2.6.1 Kiểm thử thủ công (Manual Test) 21](#_Toc103597655)

[2.6.2 Kiểm thử tự động (Automation Test) 21](#_Toc103597656)

[2.7 Các công cụ hỗ trợ quá trình kiểm thử 22](#_Toc103597657)

[2.7.1 Công cụ quản lý lỗi trong Redmine 22](#_Toc103597658)

[2.7.2 Công cụ hỗ trợ kiểm thử Postman 23](#_Toc103597659)

[2.8 Bài toán thử nghiệm 25](#_Toc103597660)

[2.8.1 Mô tả chức năng đăng nhập Error! Bookmark not defined.](#_Toc103597661)

[2.8.2 Mô tả chức năng Thêm mới tài khoản người dùng 29](#_Toc103597662)

[2.8.3 Thực hiện kiểm thử và log bug 34](#_Toc103597663)

[2.8.4 Kết quả kiểm thử 37](#_Toc103597664)

[2.9 Viết hướng dẫn sử dụng phần mềm 38](#_Toc103597665)

[3 CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 40](#_Toc103597666)

[3.1 Đánh giá kết quả thực tập 40](#_Toc103597667)

[3.2 Đinh hướng phát triển của đề tài 40](#_Toc103597668)

[KẾT LUẬN 42](#_Toc103597669)

[TÀI LIỆU THAM KHẢO 43](#_Toc103597670)

DANH MỤC HÌNH VẼ BẢNG BIỂU

[Hình 1- 1: Logo Công ty cổ phần đầu tư phần mềm Novi 5](#_Toc103591537)

[Hình 1- 2: Lịch làm việc 7](#_Toc103591538)

[Hình 2- 1: Quy trình kiểm thử phần mềm 9](#_Toc103810462)

[Hình 2- 2: Các giai đoạn kiểm thử phần mềm 11](#_Toc103810463)

[Hình 2- 3: Kiểm thử đơn vị 11](#_Toc103810464)

[Hình 2- 4: Kiểm thử tích hợp 12](#_Toc103810465)

[Hình 2- 5: Kiểm thử hệ thống 13](#_Toc103810466)

[Hình 2- 6: Kiểm thử chấp nhận 14](#_Toc103810467)

[Hình 2- 7: Kiểm thử hồi qui 15](#_Toc103810468)

[Hình 2- 8: Phương pháp kiểm thử hộp đen 15](#_Toc103810469)

[Hình 2- 9: Kiểm thử hộp trắng 17](#_Toc103810470)

[Hình 2- 10: Kiểm thử hộp xám 18](#_Toc103810471)

[Hình 2- 11: Giao diện các dự án trong Redmine 22](#_Toc103810472)

[Hình 2- 13: Giao diện màn hình Postman 24](#_Toc103810473)

[Hình 2- 14: Sơ đồ tổng quan Hệ thống Quản lý văn bằng chứng chỉ 27](#_Toc103810474)

[Hình 2- 15 Màn hình thêm mới tài khoản người dùng 29](#_Toc103810475)

[Hình 2- 16 Test case chức năng Thêm mới người dùng 34](#_Toc103810476)

[Hình 2- 17: Màn hình viết một lỗi 35](#_Toc103810477)

[Hình 2- 18: Danh sách các lỗi của dự án 36](#_Toc103810478)

[Hình 2- 19: Lỗi của chức năng Thêm mới tài khoản người dùng 37](#_Toc103810479)

[Hình 2- 20: Báo cáo kiểm thử 38](#_Toc103810480)

[Hình 2- 21 Màn hình viết tài liệu hướng dẫn sử dụng 39](#_Toc103810481)

# **MỞ ĐẦU**

Trong những năm gần đây, công nghệ thông tin đã phát triển nhanh chóng, mạnh mẽ và được ứng dụng ở hầu hết các mặt của đời sống, kinh tế, văn hóa, xã hội. Công nghệ thông tin phát triển cũng đồng nghĩa với nó là ngành công nghệ phần mềm cũng phát triển. Việc xây dựng các ứng dụng web ngày càng phổ biến và nó ngày càng trở nên thân thiết với con người qua các hoạt động giới thiệu, quảng cáo, mua bán, quản lý... trên mạng. Cuộc sống càng lên cao nhu cầu về chất lượng của một sản phẩm cũng tăng lên. Một sản phẩm không chỉ dừng lại ở việc thân thiện dễ sử dụng mà nó còn phải luôn chính xác, kịp thời và đảm bảo tính bảo mật tốt. Lỗi của một phần mềm nói chung hay một ứng dụng web nói riêng có thể gây ra những hậu quả nghiêm trọng. Do đó, hoạt động đảm bảo chất lượng phần mềm là rất quan trọng.

Với mong muốn có cái nhìn xác thực, rõ ràng hơn về quy trình kiểm thử phần mềm, đảm bảo chất lượng phần mềm và tiếp cận với các công cụ hỗ trợ kiểm thử, giải quyết phần nào vấn đề về tiết kiệm thời gian, kinh phí trong việc tìm kiếm lỗi, quản lý lỗi khi tiến hành kiểm thử; đồng thời rèn kỹ năng làm việc, tạo tiền đề định hướng cho tương lai sau khi ra trường. Áp dụng kỹ năng kiểm thử vào dự án thực tế, em chọn đề tài: “Kiểm thử và đảm bảo chất lượng Phần mềm quản lý văn bằng chứng chỉ”.

# **1 .CHƯƠNG 1: GIỚI THIỆU VỀ CÔNG TY THỰC TẬP**

## **Giới thiệu tổng quan về công ty**

Công ty Cổ phần đầu tư Phần Mềm Novi, địa chỉ: Toà A1 IA20 Ciputra – P. Đông Ngạc – Q. Bắc Từ Liêm – Hà Nội.

Điện thoại: 0816210000

Email: headoffice@novi.com.vn

Website: <http://novi.com.vn/>



Hình 1- 1: Logo Công ty cổ phần đầu tư phần mềm Novi

Với thông điệp vững bước tiên phong, Công ty Cổ phần đầu tư Novi được thành lập vào tháng 3 năm 2017, hình thành và phát triển cho đến nay. Novi luôn không ngừng sáng tạo để xây dựng các sản phẩm áp dụng vào nhu cầu thực tiễn trong cuộc sống và công việc.

Lĩnh vực kinh doanh và sản phẩm/dịch vụ của công ty:

* **Lĩnh vực kinh doanh**
* Xuất bản phần mềm
* Hoạt động thiết kế chuyên dụng
* Bán buôn máy vi tính, thiết bị điện tử, viễn thông
* Nghiên cứu khoa học và phát triển công nghệ trong lĩnh vực khoa học tự nhiên
* …

Trong đó, lĩnh vực kinh doanh chủ yếu của công ty là sản xuất và kinh doanh các giải pháp phần mềm đáp ứng nhu cầu quản lý hệ thống giáo dục

Trong xu thế hội nhập và phát triển, Novi đã và đang từng bước liên doanh liên kết với các đối tác nhằm tạo nên sức mạnh tổng hợp để cung cấp cho thị trường các sản phẩm phần mềm tốt nhất, với chi phí rẻ nhất và hệ thống hỗ trợ hoàn hảo nhất

* **Sản phẩm, dịch vụ của công ty**

Sản phẩm

* Phần mềm hỗ trợ giáo viên Novi Elearning
* Phần mềm trộn đề thi EQ.net
* Phần mềm quản lý văn bằng chứng chỉ

Dịch vụ

* Triển khai ứng dụng, cài đặt và đào tạo sử dụng, hỗ trợ sử dụng, bảo hành và bảo trì hệ thống thông tin
* Nâng cấp và mở rộng cùng sự phát triển của khách hàng

## **Vị trí thực tập tại công ty**

Vị trí: Thực tập sinh Kiểm thử phần mềm

Công việc được giao:

* Tìm hiểu về công ty, các sản phẩm của công ty
* Kiểm thử phần mềm của dự án và viết báo cáo kiểm thử
* Hỗ trợ khách hàng cài đặt phần mềm của công ty
* Tìm hiểu và áp dụng các công cụ hỗ trợ kiểm thử
* Viết tài liệu hướng dẫn sử dụng

## **Đề tài thực tập**

Đề tài: Kiểm thử phần mềm và ứng dụng trong dự án Quản lý văn bằng chứng chỉ

Mục tiêu:

* Nắm được tổng quan về quá trình kiểm thử phần mềm
* Hiểu được tầm quan trọng, mục đích, vai trò của kiểm thử pần mềm
* Hiểu về các cấp độ, các nguyên tắc, các phương pháp, kỹ thuật kiểm thử phần mềm
* Biết cài đặt và sử dụng các công cụ trong quá trình kiểm thử
* Được thâm nhập vào môi trường thực tế, nâng cao trách nhiệm với nghề nghiệp của mình, thực hiện tốt công việc được giao

## **Lịch làm việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuần | Công việc | Người hướng dẫn | Mức độ hoàn thành (SV tự đánh giá) | Ghi chú |
| 1 | * Tìm hiểu quy định tại công ty * Cài đặt các phần mềm cần thiết * Tìm hiểu các kiến thức về kiểm thử phần mềm | Lại Đức Thuần | 90 % |  |
| 2 | * Tìm hiểu cách viết test cases * Tìm hiểu công cụ hỗ trợ kiểm thử Redmine | Lại Đức Thuần | 100% |  |
| 3 | * Đọc hiểu, phân tích tài liệu đặc tả phần mềm | Lại Đức Thuần | 90% |  |
| 4 | * Lập kế hoạch test * Viết test case cho các tính năng | Lại Đức Thuần | 80% |  |
| 5 | * Thực hành test * Mô tả bug trên tool quản lý bug | Lại Đức Thuần | 90% |  |
| 6 | * Thực hành test * Mô tả bug trên tool quản lý bug | Lại Đức Thuần | 90% |  |
| 7 | * Thực hiện retest * Báo cáo bug lên tool quản lý | Lại Đức Thuần | 100% |  |
| 8 | * Thực hiện retest * Báo cáo bug lên tool quản lý | Lại Đức Thuần | 100% |  |
| 9 | * Viết tài liệu hướng dẫn sử dụng phần mềm | Lại Đức Thuần | 90% |  |
| 10 | * Viết tài liệu hướng dẫn sử dụng phần mềm | Lại Đức Thuần | 90% |  |

Hình 1- 2: Lịch làm việc

# **CHƯƠNG 2: NỘI DUNG THỰC TẬP**

## **Các khái niệm cơ bản về kiểm thử phần mềm**

### **Kiểm thử hần mềm là gì?**

Kiểm thử phần mềm (Software testing) là hoạt động nhằm tìm kiếm, phát hiện các lỗi của phần mềm; đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng đã đặt ra; cung cấp mục tiêu, cái nhìn độc lập về phần mềm, cho phép việc đánh giá và hiểu rõ hơn các rủi ro khi thực thi phần mềm, tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa tìm thấy.

### **Lỗi phần mềm là gì? Nguyên nhân sinh ra lỗi của phần mềm.**

Lỗi phần mềm: là một thất bại hoặc sai sót gây ra kết quả sai hoặc không mong muốn trong một chương trình. Đó là một lỗi khiến cho ứng dụng không thể vận hành như mong muốn.

Nguyên nhân sinh ra lỗi của phần mềm: Có rất nhiều nguyên nhân dẫn đến lỗi phần mềm. Lý do thường gặp nhất đó là do sai sót của con người trong quá trình thiết kế và lập trình. Khi đã biết được nguyên nhân dẫn đến những khiếm khuyết của phần mềm, việc sửa chữa để giảm thiểu những khiếm khuyết đó sẽ trở nên dễ dàng hơn rất nhiều.

Cá lỗi phổ biến như:

* Hiểu nhầm trong giáo tiếp hoặc không có giao tiếp
* Sự phức tạp của các ứng dụng phần mềm iện hành
* Lỗi lập trình
* Thay đổi yêu cầu
* Áp lực thời gian
* Các công cụ phát triển phần mềm
* Viết code hoặc kiểm thử các trường hợp khi chưa hiểu rõ tất cả các yêu cầu.
* Thiết kế sai dẫn đến việc trong tất cả các giai đoạn của chu kỳ phát triển phần mềm đều có vấn đề.
* Dành ít thời gian hoặc bỏ qua hoàn toàn việc kiểm tra hồi quy.
* Không theo dõi quá trình phát triển và kiểm thử một cách liền mạch. Những thay đổi vào phút cuối rất có thể sẽ dẫn đến lỗi.

### **Vai trò của kiểm thử phần mềm**

Lỗi có thể xuất hiện ở bất kì giai đoạn nào trong vòng đời phát triển phần mềm, kiểm tra nghiêm ngặt là cần thiết trong quá trình phát triển và bảo trì để xác định các lỗi để giảm thất bại khi hoạt động và làm tăng chất lượng của hệ thống khi đi vào vận hành.

- Tìm các bug (lỗi) phát sinh do lập trình viên tạo ra khi code.

- Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.

- Để ngăn ngừa lỗi.

- Đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.

- Để đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

Kiểm thử phần mềm sẽ giúp hoàn thiện các ứng dụng phần mềm hoặc sản phẩm so với yêu cầu kinh doanh và người sử dụng. Nó rất quan trọng để đảm bảo kiểm thử tốt, kiểm thử các ứng dụng phần mềm hoàn toàn và chắc chắn rằng nó hoạt động tốt.

## **Quy trình kiểm thử phần mềm**

Hình 2- 1: Quy trình kiểm thử phần mềm

### **Lập kế hoạch kiểm thử ( Test plan)**

Nhiệm vụ quan trọng là xác định được các yếu tố sau

+ Các giai đoạn kiểm thử áp dụng cho dự án

+ Các phương pháp kiểm thử

+ Các công cụ kiểm thử

+ Nhân sự tham gia

+ Tài nguyên sử dụng, bao gồm các phần cứng và phần mềm

+ Kế hoạch thực hiện

+ Mốc bàn giao các tài liệu kiểm thử

### **Chuẩn bị kiểm thử**

- Tìm hiểu nghiệp vụ của hệ thống phải kiểm thử

- Xây dựng kịch bản kiểm thử, phát triển các thủ tục và các kịch bản kiểm thử tự động (nếu có)

- Chẩn bị dữ liệu kiểm thử

- Xem xét phê duyệt các tài liệu kiểm thử

### **Thực hiện kiểm thử**

- Thưc hiện test dựa trên testcase đã viết.

- Chạy lại các case bị lỗi trước đó để xác nhận là case đó đã được sửa.

- So sánh kết quả kết quả ghi nhận được khi thực thi với kết quả mong đợi.

- Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử.

- Viết báo cáo lỗi khi có bug.

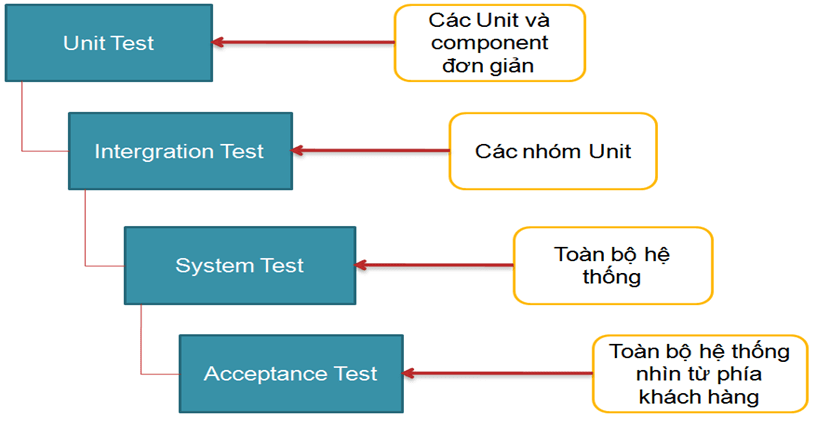
### **Viết báo cáo kiểm thử (Test report)**

Báo cáo kiểm thử thể hiện tiến độ kiểm thử, tiến độ sửa lỗi và số lượng lỗi được tìm thấy hay còn tồn của dự án.

Nó là công cụ để phục vụ cho đánh giá hay giám sát dự án có kịp tiến độ hay không, có thể bàn giao cho khách hàng hay không và các vấn đề cần giải quyết khi mà số lượng lỗi còn nhiều, gây ra các rủi ro về tiến độ hoàn thành của dự án để có những điều chỉnh kịp thời.

## **Các mức kiểm thử (Test Level)**

Quy trình kiểm thử phần mềm sẽ được thực hiện theo 4 giai đoạn như sau:



Hình 2- 2: Các giai đoạn kiểm thử phần mềm

### **Kiểm thử đơn vị (Unit test)**

Diagram

Description automatically generated

Hình 2- 3: Kiểm thử đơn vị

Đơn vị: Là thành phần nhỏ nhất của phần mềm có thể kiểm thử được. Ví dụ: Các hàm, lớp, thủ tục, phương thức. Đơn vị thường có kích thước nhỏ, chức năng hoạt động đơn giản, không gây nhiều khó khăn trong việc kiểm thử, ghi nhận và phân tích kết quả do đó nếu phát hiện lỗi việc tìm kiếm nguyên nhân và sửa lỗi cũng đơn giản và tốn ít chi phí hơn. Một nguyên lý đúc kết từ thực tiễn là thời gian dành cho kiểm thử đơn vị sẽ được đền bù bằng việc tiết kiệm được khá nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức độ kiểm thử sau đó.

- Mục đích: Đảm bảo thông tin được xử lý đúng và có đầu ra chính xác trong mối tương quan giữa dữ liệu nhập và chức năng của đơn vị.

- Người thực hiện: Do việc kiểm thử đơn vị đòi hỏi phải kiểm tra từng nhánh lệnh, nên đòi hỏi người kiểm thử có kiến thức về lập trình cũng như về thiết kế của hệ thống nên người thực hiện thường là lập trình viên.

### **Kiểm thử tích hợp (Intergration testing)**

Diagram

Description automatically generated

Hình 2- 4: Kiểm thử tích hợp

- Kiểm thử tích hợp là kiểm thử sự kết hợp và giao tiếp giữa các đơn vị của một chương trình và kiểm thử như một chương trình đã hoàn thành

- Mục đích:

* Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị cũng nhue lỗi của bản thân từng đơn vị (nếu có)
* Tích hợp các đơn vị lẻ thành các hệ thống nhỏ và cuối cùng là tích hợp các hệ thống nhỏ thành một hệ thống hoàn chỉnh để chuẩn bị cho kiểm thử hệ thống

- Người thực hiện: thường là Lập trình viên

- Lưu ý:

* Kiểm thử tích hợp chỉ nên thực hiện trên từng đơn vị đã được kiểm tra cẩn thận trước đó bằng kiểm thử đơn vị, và tất cả các lỗi mức đơn vị đã được sửa chữa
* Nên tích hợp dần từng đơn vị: Một đơn vị nên được tích hợp vào một nhóm các đơn vị khác đã được tích hợp và hoàn thành kiểm thử tích hợp trước đó, vì khi đó chỉ cần kiểm tra giao tiếp giữa đơn vị mới được thêm vào với nhóm các đơn vị đã được tích hợp trước đó

### **Kiểm thử hệ thống (System test)**

Diagram

Description automatically generated

Hình 2- 5: Kiểm thử hệ thống

* Kiểm thử hệ thống bắt đầu khi tất cả các đơn vị của hệ thống được tích hợp thành công, đây là công đoạn kiểm thử tốn nhiều công sức và thời gian nhất. Và đặc biệt, công đoạn này thường đòi hỏi được thực hiện bởi 1 nhóm nhân viên tách biệt với nhóm phát triển, có chuyên môn và kinh nghiệm kiểm thử.
* Kiểm thử hệ thống gồm nhiều loại kiểm thử khác nhau, trong số đó, các mục tiêu kiểm thử quan trọng nhất là:
  + Kiểm thử chức năng
  + Kiểm thử hiệu năng
  + Kiểm thử an toàn thông tin
* Mục đích: Kiểm tra xem hệ thống được làm ra có thỏa mãn yêu cầu hay không về nhiều khía cạnh: hoạt động, độ tin cậy, hiệu năng của hệ thống
* Người thực hiện: Nhóm nhân viên kiểm thử

### **Kiểm thử chấp nhận (Acceptance test)**

Diagram

Description automatically generated

Hình 2- 6: Kiểm thử chấp nhận

Mục đích: Kiểm thử chấp nhận còn gọi là kiểm thử nghiệm thu nhằm mục đích chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng đã chấp nhận sản phẩm. - Người thực hiện: Khách hàng.

Có 2 loại kiểm thử chấp nhận đó là kiểm thử Alpha và kiểm thử Beta:

* Kiểm thử Alpha: Người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm với sự hỗ trợ của nhân viên kiểm thử, nhân viên kiểm thử sẽ khi lại các lỗi hoặc phản hồi của khách hàng và báo lại với đơn vị phát triển phần mềm để lên kế hoạch sửa chữa
* Kiểm thử Beta: là loại kiểm thử mà khách hàng thực hiện kiểm thử ở chính môi trường của họ. Loại kiểm thử này được thực hiện sau kiểm thử Alpha.

### **Kiểm thử hồi qui**

Kiểm thử hồi qui là một hoạt động cấn thiết để chỉ ra rằng việc thay đổi mã nguồn không gây ra những ảnh hưởng bất lợi đến hệ thống nói chung.

|  |  |
| --- | --- |
| **Mục đích kiểm thử** | - Kiểm thử hồi qui dùng để kiểm tra các phần được sửa chữa và các phần liên quan đến các phần sửa chữa trong phần mềm, để đảm bảo rằng những sự thay đổi đó không gây ra lỗi trong những phần khác |
| **Cách thực hiện** | - Tái sử dụng các kịch bản kiểm thử từ những phần kiểm thử trước để kiểm thử các mô-đun đã được sửa chữa. - Sử dụng công cụ kiểm thử tự động: Tạo một số test script về chức năng. - Xây dựng một chương trình phân tích sơ sở hạ tầng. Chúng ta dựng một cơ sở hạ tầng có thể mở rộng được để thực hiện và đánh giá chương trình phân tích. Dựa vào kết quả phân tích chúng ta xác định phạm vi cần kiểm thử hồi qui. |
| **Điều kiện hoàn thành** | Toàn bộ các trường hợp kiểm thử đã chọn được thực hiện và đạt yêu cầu. |

Hình 2- 7: Kiểm thử hồi qui

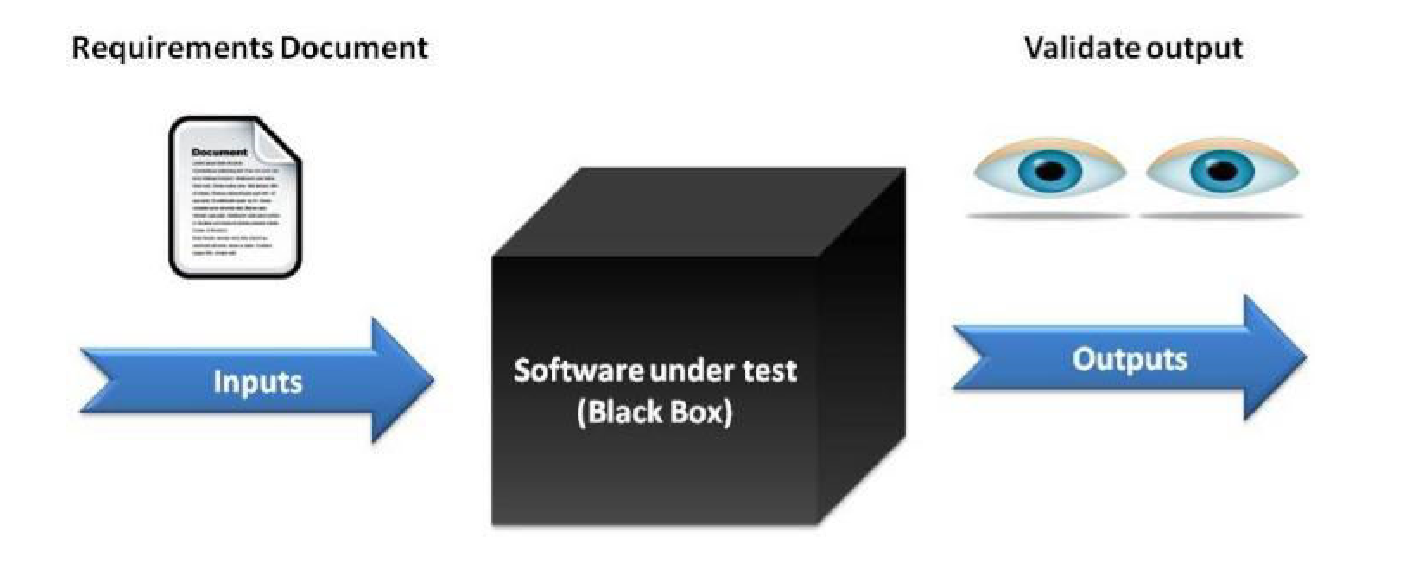
## **Các kỹ thuật kiểm thử phần mềm**

Có 3 kỹ thuật kiểm thử phần mềm chính là:

* Kiểm thử hộp đen
* Kiểm thử hộp trắng
* Kiểm thử hộp xám

### **Phương pháp kiểm thử hộp đen**

Khái niệm: Là phương pháp kiểm thử dựa trên đầu vào và đầu ra của chương trình để kiểm thử, Chỉ kiểm thử chức năng và giao diện dựa trên nghiệp vụ của hệ thống mà không quan tâm tới mã chương trình bên trong được viết ra sao. Tester xem phần mềm như là một hộp đen. Kiểm thử hộp đen không yêu cầu kỹ sư kiểm thử cần phải có bất kỳ kiến thức về mã hoặc thuật toán của chương trình. Nó kiểm tra các chức năng của hệ thống tức là những gì hệ thống được cho là cần phải làm dựa trên các đặc tả yêu cầu (Requirement document).



Hình 2- 8: Phương pháp kiểm thử hộp đen

Ưu điểm:

* Không có mối ràng buộc nào về code, và kiểm thử những thứ lập trình viên có thể bỏ qua hoặc không nhìn thấy trong quá trình lập trình.
* Người kiểm thử thực hiện tử quan điểm của người dùng và sẽ giúp đỡ trong việc sáng tỏ sự chênh lệch về thông số kĩ thuật.
* Người kiểm thử có thể không phải là một lập trình viện chuyên nghiệp, không cần phải biết ngôn ngữ lập trình hoặc làm thế nào các phần mềm đã được thực hiện.
* Người kiểm thử có thế thực hiện một cách độc lập từ các developer, cho phép một cái nhìn khách quan và tránh sự phát triển thiên vị.
* Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.

Nhược điểm:

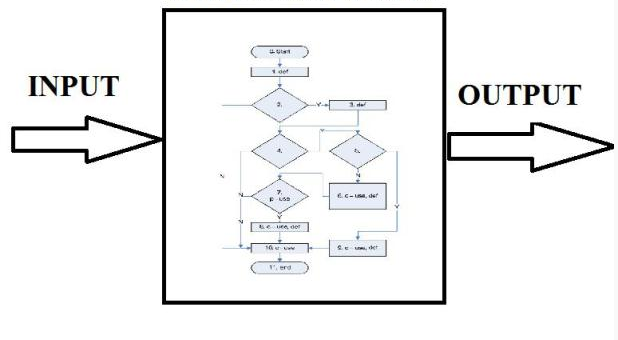
* Dữ liệu đầu vào yêu cầu một khối lượng mẫu khá lớn.
* Chỉ có thể khám phá mù (không biết phần mềm kiểm thử được xây dựng như thế nào), do đó khi áp dụng phương pháp kiểm thử hộp đen đòi hỏi người thực hiện phải làm việc vất vả hơn để khám phá được càng nhiều bug càng tốt.
* Nhiều dự án không có thông số rõ ràng thì việc thiết kế test case rất khó và do đó khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.
* Chỉ có một số nhỏ các đầu vào có thể được kiểm tra và nhiều đường dẫn chương trình sẽ được để lại chưa được kiểm tra.

### **Phương pháp kiểm thử hộp trắng**

Khái niệm: Là phương pháp kiểm thử dựa cả vào giải thuật, cấu trúc code bên trong phần mềm, việc kiểm thử được tiến hành dựa cả vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không.

Trong kiểm thử hộp trắng, cấu trúc mã hoặc giải thuật của chương trình được đưa vào xem xét, các trường hợp kiểm thử được thiết kế dựa vào cấu trúc mã hoặc cách thức làm việc của chương trình. Người kiểm thử truy cập vào mã chương trình và có thể kiểm tra nó, lấy đó làm cơ sở để hổ trợ việc kiểm thử.

Để thực hiện được phương pháp kiểm thử hộp trắng thì người kiểm thử phải có kỹ năng , kiến thức nhất định về ngôn ngữ lập trình được dùng, về thuật giải được dùng trong thành phần phần mềm để có thể thông hiểu được chi tiết về các đoạn code cần khiểm thử.



Hình 2- 9: Kiểm thử hộp trắng

Ưu điểm:

* Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test.
* Test kỹ càng hơn, có thể bao phủ hầu hết các trường hợp.
* Thích hợp trong việc tìm kiếm lỗi và các vẫn đề trong mã lệnh.
* Cho phép tìm kiếm các lỗi ẩn bên trong.
* Các lập trình viên có thể tự kiểm tra.
* Giúp tối ưu việc mã hóa.
* Do yêu cầu kiến thức cấu trúc bên trong của phần mềm nên việc kiểm soát lỗi tối đa nhất.

Nhược điểm:

* Yêu cầu người kiểm thử phải am hiểu cấu trúc mã lệnh của trương trình. Do đó đòi hỏi tài nguyên nhân lực và máy tốn kém
* Có khả năng tồn tại các tổ hợp lệnh khác nhau gây lỗi
* Không kiểm thử hết đường đi với các vòng lặp
* Khó thực hiện và chi phí thực hiện cao

### **Phương pháp kiểm thử hộp xám**

Khái niệm: Là một phương pháp kiểm thử phần mềm được kết hợp giữa phương pháp kiểm thử hộp trắng và phương pháp kiểm thử hộp đen.

Diagram

Description automatically generated

Hình 2- 10: Kiểm thử hộp xám

Trong kiểm thử hộp xám, cấu trúc bên trong sản phẩm chỉ được biết một phần, Tester có thể truy cập vào cấu trúc dữ liệu bên trong và thuật toán của chương trình với mục đích là để thiết kế testcase, nhưng khi test thì test như là người dùng cuối hoặc là ở mức hộp đen.

Được gọi là kiểm thử hộp xám vì trong chương trình phần mềm, mắt của Tester giống như hộp xám/bán trong suốt-nhìn qua hộp này ta chỉ có thể thấy được một phần.

Ví dụ: Khi code của 1 module nào đó được xem xét để thiết kế testcase (phương pháp kiểm thử hộp trắng) và khi test thực tế thì được thực hiện test trên giao diện người dùng (phương pháp kiểm thử hộp đen).

## **Các kỹ thuật kiểm thử**

Có nhiều kỹ thuật kiểm thử phần mềm, sau đây là 4 kỹ thuật phổ biến nhất trong kỹ thuật kiểm thử hộp đen là: kỹ thuật phân vùng tương đương, kỹ thuật phân tích giá trị biên, bảng quyết định và kỹ thuật đoán lỗi.

Trong quá trình kiểm thử, kiểm thử viên có thể áp dụng nhiều kỹ thuật khác nhau, kết hợp các phương pháp kiểm thử với nhau để có thể tìm các lỗi của phần mềm một cách tối đa.

### **Kỹ thuật phân vùng tương đương**

Đây là phương pháp chia đầu vào thành những nhóm tương đương nhau. Nếu một giá trị trong nhóm hoạt động đúng thì tất cả các giá trị trong nhóm đó cũng hoạt động đúng và ngược lại.

* Mục đích: Giảm đáng kể số lượng test case cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện.

Ưu điểm: Vì mỗi vùng tương đương ta chỉ cần test trên các phần tử đại diện nên số lượng test case được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

Nhược điểm: Không phải với bất kỳ bài toán nào cũng có thể áp dụng kĩ thuật này. Có thể bị hack lỗi ở biên nếu chỉ chọn giá trị ở khoảng giữa của miền tương đương.

### **Bảng quyết định**

* Bảng quyết định là một đại diện bảng đầu vào gồm có các điều kiện đầu vào, các tình huống thử nghiệm và các kết quả. Đây là một công cụ rất hiệu quả được sử dụng cho cả kiểm thử phần mềm phức tạp và quản lý các yêu cầu.
* Bảng quyết định là một kỹ thuật tốt để áp dụng cho những trường hợp cần nhiều sự kết hợp. Bảng quyết định hỗ trợ việc lựa chọn test case một cách có hệ thống và có thể đem lại nhiều lợi ích trong việc nhận biết vấn đề tiềm ẩn và sự không rõ ràng trong đặc tả (specification).
* Bảng quyết định đảm bảo bao gồm tất cả các kết hợp có thể có của các giá trị điều kiện được gọi là thuộc tính tính đầy đủ, mang lại số testcase tối thiểu với độ bao phủ tối đa.
* Phương pháp này khắc phục được khuyết điểm của 2 phương pháp phân tích giá trị biên và phân vùng tương đương, đó là kiểm soát được sự kết hợp của của các giá trị đầu vào bằng cách sử dụng mô hình quan hệ logic nguyên nhân – kết quả cho các thành phần phần mềm.

### **Kỹ thuật phân tích giá trị biên**

Đây là phương pháp kiểm thử mà chúng ta sẽ kiểm thử tất cả các giá trị ở vùng biên của dữ liệu vào và dữ liệu ra. Chúng ta sẽ tập trung vào các giá trị biên chứ không kiểm thử toàn bộ dữ liệu.

Mục đích: Thay vì chọn nhiều giá trị trong lớp đương tương để làm đại diện, phân tích giá trị biên yêu cầu chọn một hoặc vài giá trị là các cạnh của lớp tương đương để làm điều kiện kiểm thử.

Phân tích giá trị biên là kỹ thuật thiết kế test case và hoàn thành phân vùng tương đương.

Thiết kế test case bằng phương pháp phân tích giá trị biên chính là lựa chọn các test case để thực thi giá trị biên

Nguyên tắc:

* + - Giá trị nhỏ nhất
    - Giá trị ngay dưới giá trị nhỏ nhất
    - Giá trị bình thường
    - Giá trị ngay trên giá trị lớn nhất
    - Giá trị lớn nhất

### **Kỹ thuật đoán lỗi**

Phương pháp này không có quy trình cụ thể vì có tính trực giác cao và không thể dự đoán trước

Phương pháp dựa vào kinh nghiệm, trực quan của kiểm thử viên để đưa ra các trường hợp kiểm thử. Trong quá trình kiểm thử, kiểm thử viên có thể áp dụng nhiều kỹ thuật khác nhau, kết hợp các phương pháp kiểm thử với nhau để có thể tìm các lỗi của phần mềm một cách tối đa.

Ví dụ: ở màn hình login, đôi khi developer code hay gán username là “Admin” và pass là rỗng hoặc “123”, vì vậy khi thực hiện test chúng ra nên test cả case này.

## **Phân loại kiểm thử**

### **Kiểm thử thủ công (Manual Test)**

Khái niệm: Kiểm thử thủ công, Tester làm mọi công việc hoàn toàn bằng tay, từ viết test case đến thực hiện kiểm thử; mọi thao tác như: Nhập điều kiện đầu vào, thực hiện một số sự kiện khác, sau đó so sánh kết quả thực tế với kết quả mong muốn trong test case, điền kết quả kiểm thử. Hiện nay, phần lớn các tổ chức, các công ty phần mềm, hoặc các nhóm làm phần mềm đều thực hiện kiểm thử thủ công là chủ yếu.

Ưu điểm:

+ Cho phép Tester thực hiện việc kiểm thử khám phá.

+ Thích hợp kiểm tra sản phẩm lần đầu tiên.

+ Thích hợp kiểm thử trong trường hợp các test case chỉ thực hiện một số ít lần.

+ Giảm được chi phí ngắn hạn.

Nhược điểm: Tốn thời gian. Đối với mỗi lần kiểm tra lại, người kiểm thử phải thực hiện lại một tập hợp các test case đã chạy, dẫn đến sự mệt mỏi và gây ra sự nhàm chán.

### **Kiểm thử tự động (Automation Test)**

Khái niệm: Kiểm thử phần mềm tự động là thực hiện kiểm thử phần mềm bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho Tester không phải lặp đi lặp lại các bước nhàm chán.

Ưu điểm:

**+** Thay thế con người lặp đi lặp lại đúng quy tắc các bước kiểm thử.

**+** Độ tin cậy cao:Dù lặp đi lặp lại nhiều lần vẫn cho ra kết quả giống nhau, do vậy độ ổn định cao, tránh được rủi ro có thể phát sinh.

**+** Cải thiện chất lượng: Kiểm thử tự động làm giảm rủi ro về mặt chất lượng sản phẩm, việc kiểm thử được thực hiện một cách nhanh chóng, không gây mệt mỏi, tránh trường hợp chủ quan cho những trường hợp kiểm thử hồi quy, retest, kiểm thử toàn bộ hệ thống; có thể tái sử dụng các trường hợp kiểm thử.

+ Tốc độ xử lý cực nhanh.

+ Chi phí thấp: Việc rút ngắn thời gian và tiết kiệm nhân lực giúp cho việc kiểm thử tự động trở nên hiệu quả.

+ Hỗ trợ xuất ra nhiều ngôn ngữ phổ biến như Ruby, Java, Python…

Nhược điểm

+ Ban đầu thì chi phí cho kiểm thử tự động sẽ cao hơn kiểm thử thủ công.

+ Mất chi phí về bản quyền, bảo trì, tìm hiểu, training.

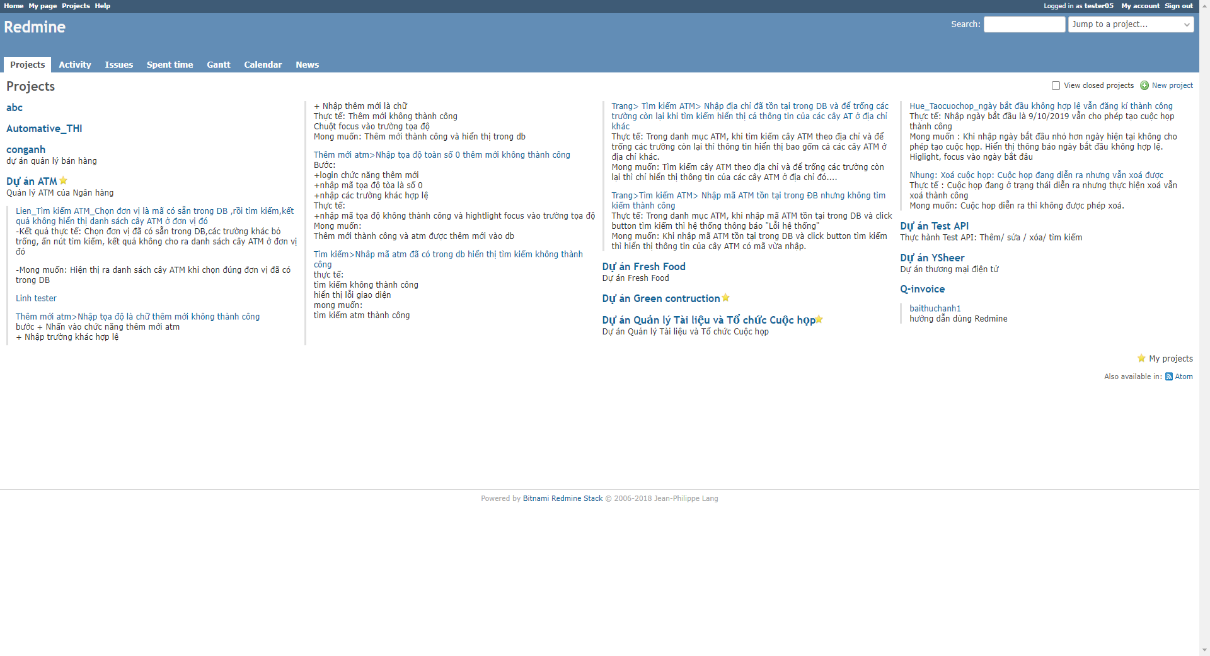
+ Khó mở rộng hơn nhiều so với kiểm thử thủ công.

+ Yêu cầu những người có trình độ chuyên môn cao mới thực hiện được.

+ Số lượng công việc phải làm để mở rộng cho kiểm thử tự động sẽ nhiều và khó hơn so với kiểm thử thủ công.

## **Các công cụ hỗ trợ quá trình kiểm thử**

### **Công cụ quản lý lỗi trong Redmine**



Hình 2- 11: Giao diện các dự án trong Redmine

Redmine là một ứng dụng web phổ biến được dùng để quản trị các dự án kỹ thuật mà thông dụng nhất là dự án phần mềm. Nó được xây dựng trên nền tảng Ruby on Rails; là một ứng dụng đa nền tảng và độc lập với các hệ quản trị cơ sở dữ liệu.

Redmine là một dự án mã nguồn mở, miễn phí theo giấy phép GPL phiên bản 2. Chỉ cần sử dụng một máy chủ cỡ nhỏ hoặc sử dụng máy tính PC thường, chúng ta sẽ có một ứng dụng quản trị dự án tập trung rất mạnh mẽ chạy trên nền web, đa người dùng và đặc biệt thích hợp cho các công ty phần mềm.

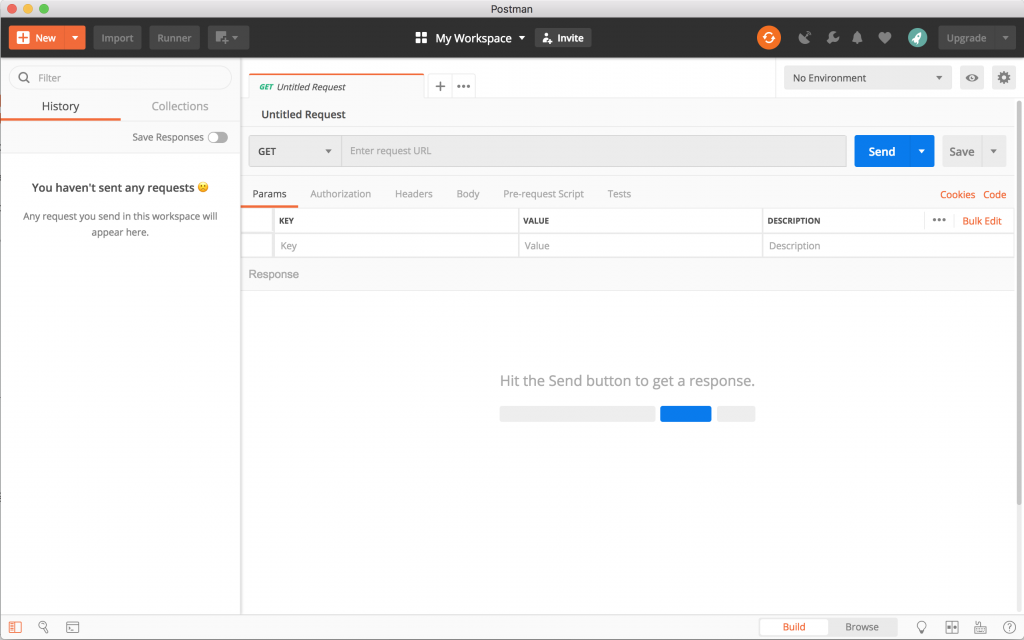
* Các đặc tính của Redmine
* Quản trị đa dự án và hỗ trợ phân cấp dự án cha – con
* Quản lý linh động vai trò của các thành viên trong nhóm
* Có hệ thống quản lý công việc mạnh mẽ
* Đồ thị Gantt và lịch
* Quản lý tin tức, tài liệu
* Hỗ trợ thông báo qua email tới các thành viên
* Hỗ trợ hệ thống từ điển (wiki)
* Có Forum riêng trong mỗi dự án
* Quản lý danh mục công việc theo thời gian
* Quản lý tài khoản người dùng, hỗ trợ tự đăng ký và xác thực tài khoản
* Hỗ trợ các hệ quản trị CSDL phổ biến SQL Server, MySQL, DB2, Oracle ..
* Đa ngôn ngữ
* Vai trò và quyền trong Redmine

Một thành viên trong dự án được phân một hay nhiều vai trò. Mỗi vai trò xác định những quyền nhất định cho thành viên ấy. Một người dùng có thể được xác định vai trò khác nhau trong những dự án khác nhau. Redmine có thể xác lập các vai trò của các thành viên trong dự án.

Redmine có 3 vai trò mặc định :

* Manager dành cho người quản lý của dự án.
* Developer dành cho những người tham gia hoàn toàn vào dự án, trực tiếp thực hiện dự án và trực tiếp tạo ra thành quả dự án.
* Reporter dành cho những người chỉ tham gia vào dự án như quan sát viên, đánh giá kết quả dự án.

### **Công cụ hỗ trợ kiểm thử Postman**



Hình 2- 12: Giao diện màn hình Postman

Postman là một công cụ cho phép chúng ta thao tác với API, phổ biến nhất là REST. Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, …). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

Postman là một công cụ hiệu quả, mạnh mẽ và dễ sử dụng với mọi đối tượng. Ngoài ra ra công cụ này còn có một và tính năng độc đáo và rất phù hợp để teamwork như:

* Collections: Postman cho phép người dùng tạo các collections bao gồm các request mà họ đã tạo để lưu lại cho việc sử dụng sau này, cùng với việc lưu trữ các request của chung một server hay mục đích cũng sẽ giúp người dùng tiết kiệm thời gian khi muốn tìm để tái sử dụng.
* Enviroments: Đây là tính năng giúp người dùng tạo được các config cho từng môi trường test API của mình bằng cách gán giá trị cho một số biến nhất định, ví dụ như url, port, …. Rất phù hợp để điều chỉnh linh hoạt giữa các môi trường khác nhau như alpha, staging, … mà không cần phải tạo request mới.
* Collaboration: tính năng cho phép những người dùng chia sẻ những requests, collections hay environments với người khác một cách dễ dàng
* API Testing: đương nhiên rồi, đây là tính năng quan trọng nhất của Postman, người dùng có thể test API của mình dễ dàng với những tính năng mạnh cùng với giao diện thân thiện của Postman.

## **Bài toán thử nghiệm**

*Mô tả hệ thống Quản lý văn bằng chứng chỉ*

Phần mềm được xây dựng và phát triển bởi Công ty Cổ phần Đầu tư Phần mềm Novi. Qua việc khảo sát các đơn vị Phòng giáo dục các quận huyện, hiện nay vẫn còn rất niều đơn vị đang áp dụng hình thức quản lý thủ công trong việc quản lý và cấp phát bằng tốt nghiệp. việc quản lý thủ công theo phương pháp sổ sách truyền thống xảy ra rất nhiều bất cập, đặc biệt là công đoạn tra cứu văn bằng còn khó khăn và in nội dung văn bằng lên phôi mẫu thường xảy ra sai lệch,.. Phần mềm quản lý văn bằng thực sự cần thiết để giải quyết các vấn đề đang tồn tại theo phương pháp quản lý sổ sách truyền thống

*Ưu điểm*

* Áp dụng các quy trình tự động giảm tiểu tối đa thao tác trong công việc
* Tra cứu và tìm kiếm đơn giản chỉ cần họ tên và mã văn bằng
* Sổ sách được lưu trữ trưc tuyến , bảo mật và thường xuyên backup, tránh tình trạng mất mát dữ liệu
* Tính năng thiết kế phôi, in phôi được thực hiện trên máy tính với giao diện trực quan, đảm bảo độ chính xác và thẩm mỹ đồng tời kiểm soát được việc in lỗi

*Đối tượng sử dụng*

* Áp dụng bạn đầu cho các đối tượng phòng giáo dục của các quận/uyện, thị xã và phân cấp sử dụng xuống các đơn vị giáo dục cấp trường(THCS)
* Có 3 đối tượng chính:

+ Quản lý hệ thống: Chuyên viên CNTT được giao nhiệm vụ quản lý hệ thống máy chủ và cơ sở dữ liệu, theo dõi và hắc phục các sự cố hệ thống

+ Quản lý: Cuyên viên Phòng GD hoặc người được PGD giao quản lý chuyên trách

+ Giáo viên: Giáo viên hoặc người được ban giám hiệu trường giao phụ trách

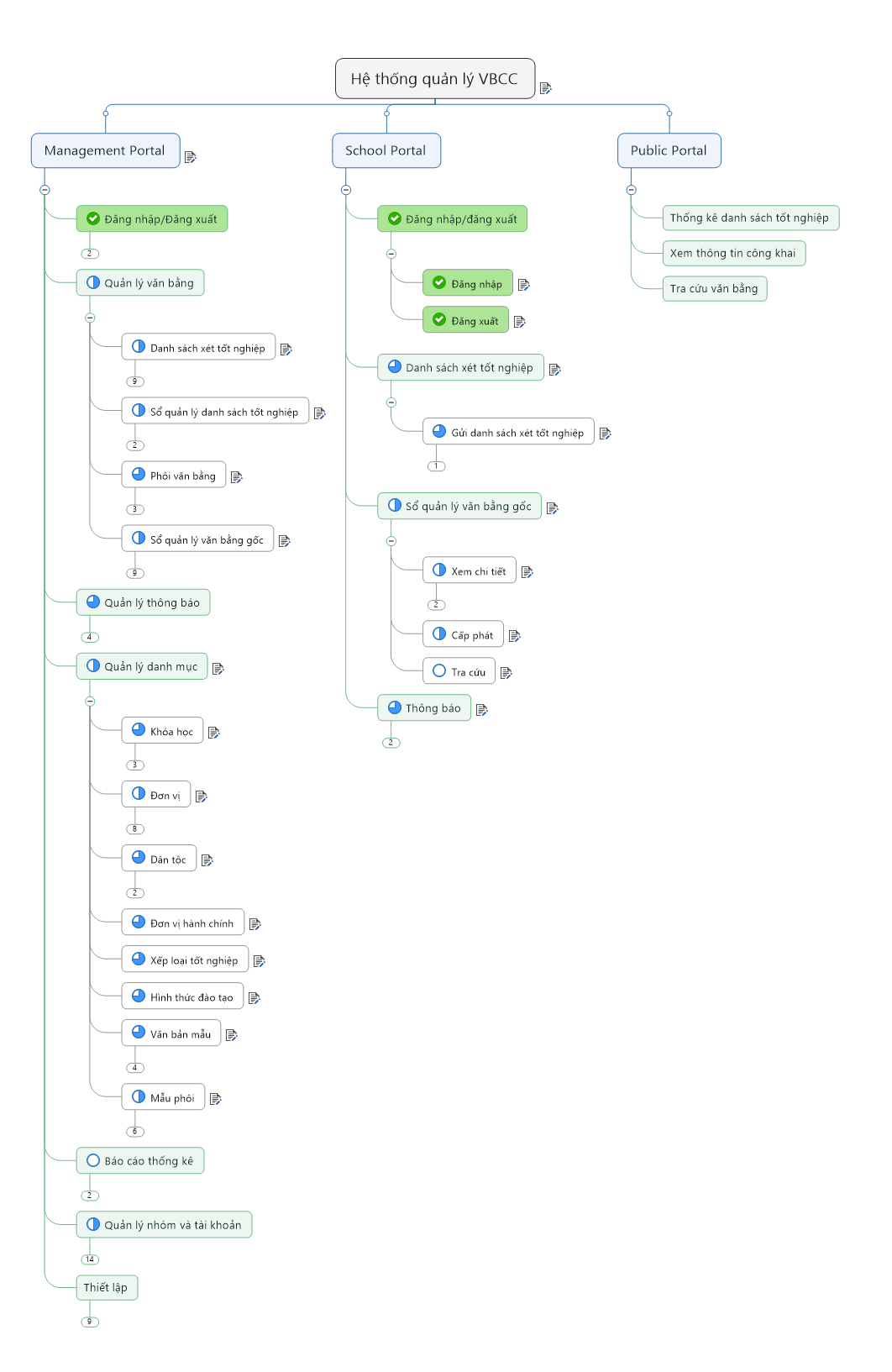
*Mô hình hệ thống*

Hệ thống được xây dựng trên mô hình webAPI bao gồm 2 thành pần chính:

* Máy chủ API: hệ quản trị SQL Server quản lý CSDL và Java endpoint webAPI/web services
* Máy củ Portal: chứa mã nguồn FrontEnd được xây dựng trên công nghệ ReactJs kết nối dữ liệu với API endpoint để hiển thị:

+ Portal tra cứu thông tin văn bằng

+ Hệ thống quản lý chức năng phần mềm



Hình 2- 13: Sơ đồ tổng quan Hệ thống Quản lý văn bằng chứng chỉ

* **Đăng nhập, đăng xuất**

Đăng nhập: Cho phép sử dụng tài khoản với tên đăng nhập và mật khẩu được cấp để đăng nhập vào hệ thống.

Đăng xuất: Cho phép người dùng chấm dứt phiên sử dụng của tài khoản đang đăng nhập hệ thống

* **Quản lý văn bằng**

Thực hiện các chức năng quản lý văn bằng trong hệ thống như:

* Danh sách xét tốt nghiệp
* Sổ quản lý danh sách học sinh tốt nghiệp
* Quản lý Phôi văn bằng
* Sổ quản lý văn bằng gốc
* **Quản lý thông báo**

Chức năng cho phép người dùng xem danh sách các thông báo tư hệ thống hoặc thông báo từ các đơn vị gửi.

Cho phép gửi thông báo tới các đơn vị

* **Quản lý danh mục**

Thực hiện chức năng quản trị các thông tin danh mục trong hệ thống như:

* Khoá tốt nghiệp
* Đơn vị
* Đơn vị hành chính
* Xếp loại tốt nghiệp
* Hình thức đào tạo
* Văn bản mẫu
* Mẫu phôi

Người dùng dễ dàng quản lý các thông tin liên quan, bên cạnh đó người dùng có thể lưu trữ file, các mẫu báo cáo liên quan đến văn bằng và cũng có thể thiết kế những mẫu phôi văn bằng phục vụ cho việc in ấn.

* **Báo cáo thống kê**
* Báo cáo tình trạng sử dụng phôi
* Lịch sử truy cập
* Lịch sử khai thác
* **Quản lý nhóm và tài khoản**

- Nhóm người dùng: Hiển thị danh sách các nhóm người dùng đang có trong hệ thống (đối tượng được phân quyền là tài khoản host hoặc được người quản trị cấp quyền quản lý nhóm người dùng).

- Tài khoản: Hiển thị danh sách tài khoản người dùng có trên hệ thống thuộc đơn vị của mình hoặc đơn vị trực thuộc.

- Cho phép thêm, sửa, xoá nhóm và các tài khoản người dùng.

* **Thiết lập**

Cho phép thiết lập hệ thống, thiết lập các tập tin tải lên và thiết lập dữ liệu.

### **Kiểm thử chức năng Thêm mới tài khoản người dùng**

Graphical user interface, application, Teams

Description automatically generated

Hình 2- 14 Màn hình thêm mới tài khoản người dùng

Mô tả:

* Các trường bắt buộc nhập không được bỏ trống
* Tài khoản không chứa các kí tự đặc biệt là duy nhất và phải lớn hơn 6 kí tự,nhỏ hơn 20 kí tự
* Họ tên không chứa kí tuej đặc biệt, loedn hoen 6 kí tự và nhỏ hơn 30 kí tự
* Số điện thoại là số và từ 10-12 kí tự và đúng định dạng 09,08,02,016[2-9]
* Email phải đúng định dạng [xxx@xxx.xxx](mailto:xxx@xxx.xxx) không chứa kí tự đặc biệt và là duy nhất
* Droplist Đơn vị: Hiển thị danh sách các đơn vị có trên hệ thống
* Droplist Nhóm người dùng: Hiển thị danh sách các nhóm người dùng gồm: Sở giáo dục, Phòng giáo dục, Trường học.

Phân tích:

* **Áp dụng phương pháp phân vùng tương đương:**
* Tài khoản phải chứa [6,20] ký tự chỉ gồm chữ và số: thực hiện nhập vào 5,21ký tự,rỗng
* Họ tên phải chứa [6,30] ký tự
* Số điện thoại phải từ [10,12] ký tự số ,đúng định dạng 09,08,02,016[2-9] : thực hiện nhập vào 9, 13 ký tự số,rỗng.

Các lớp tương đương

Tài khoản

* Tài khoản >=6 && <=20🡪 hợp lệ
* Tài khoản <6 🡪không hợp lệ
* Tài khoản >20 🡪không hợp lệ
* Tài khoản rỗng 🡪không hợp lệ
* Tài khoản có kí tự không phải số hoặc chữ🡪không hợp lệ

Họ và tên

* Họ và tên không rỗng🡪 hợp lệ
* Họ tên >=6&&<=30🡪 hợp lệ
* Họ tên < 6 🡪không hợp lệ
* Họ tên > 20 🡪không hợp lệ
* Họ tên rỗng🡪không hợp lệ
* Họ tên chứa kí tự đặc biệt 🡪không hợp lệ

Số điện thoại

* Số điện thoại >=10 && <=12 🡪 hợp lệ
* Số điện thoại <10 🡪 không hợp lệ
* Số điện thoại >12 🡪 không hợp lệ
* Số điện thoại chứa kí tự không phải số🡪 không hợp lệ
* Số điện thoại rỗng 🡪không hợp lệ

Email

* Bỏ trống email🡪không hợp lệ
* Thiếu Local-Part 🡪không hợp lệ
* Thiếu @🡪không hợp lệ
* Thiếu server name🡪không hợp lệ
* Local-Part chứa kí tự không phải chữ hoặc số🡪không hợp lệ
* Server name sai định dạng 🡪không hợp lệ

**Đơn vị**

* Đơn vị rỗng 🡪 Không hợp lệ
* Đơn vị không rỗng 🡪 Hợp lệ

**Nhóm người dùng**

* Nhóm người dùng rỗng🡪 Không hợp lệ
* Nhóm người dùng không rỗng🡪Hợp lệ

|  |  |  |
| --- | --- | --- |
| **Điều kiện đầu vào** | **Các lớp tương đương hợp lệ** | **Các lớp tương đương không hợp lệ** |
| Tài khoản | >=6 & <=20 | - >20 - <6 - bỏ trống  - có kí tự không phải số hoặc chữ:  + khoảng trắng  + kí tự đặc biệt  + chữ |
| Số điện thoại | >=10 & <=12  2 số đầu :09,08,02  3 số đầu:016[2-9] | - >12 - <10 - Bỏ trống  - Số đầu khác:09,08,02,016[2-9]  - Chứa kí tự không phải số: + khoảng trắng  + kí tự đặc biệt  +chữ |
| Họ và tên | >=6 & <=30 | - >30 - <6 - bỏ trống  - khoảng trắng  - kí tự đặc biệt |
| Đơn vị | Không trống | Bỏ trống |
| Nhóm người dùng | Không trống | Bỏ trống |
| Email | Local-Part là chữ hoặc số hoặc các kí tự(+ - .) và kí tự nối là @ và  server name có dạng xxx.xxx | -Bỏ trống email  -Thiếu Local-Part  -Thiếu @ -Thiếu server name -Local-Part chứa kí tự không phải chữ hoặc số:  + khoảng trắng  +kí tự đặc biệt  -Server name sai định dạng:  + thiếu .  + thiếu com  + chứa khoảng trắng  +chứa kí tự đặc biệt  + thiếu .com  +thiếu domain trong server name |

Testcase chức năng thêm mới tài khoản người dùng

Table

Description automatically generated

Table

Description automatically generated

Hình 2- Test case chức năng Thêm mới người dùng

### **Thực hiện kiểm thử và log bug**

Quy trình xử lý lỗi:

Bước 0: Phát hiện phần mềm có lỗi

Bước 1: Đưa lỗi lên hệ thống quản lý Redmine

Bước 2: Gián lỗi cho nhân viên phát triển

Bước 3: Xử lý lỗi

Bước 4: Kiểm thử lại

Bước 5: Đóng lỗi

Graphical user interface

Description automatically generated

Hình 2- 16: Màn hình viết một lỗi

Bước 1: Đưa lỗi lên phần Mềm quản lý lỗi

Subject: Tóm tắt nội dung của lỗi, có thể gọi là tiêu đề của lỗi

Severity (trọng số của lỗi) Thông số biểu hiện độ nghiêm trọng của lỗi, thông thường lỗi sẽ thuộc 1 trong 3 trọng số dưới đây

* Minor: Các lỗi định dạng(font chữ, cỡ chữ, màu sắc..) lỗi chính tả, validate dữ liệu
* Major: Các lỗi ràng buộc dữ liệu, lỗi chức năng nghiệp vụ của hệ thống
* Crash: Các lỗi chức năng nghiệp vụ của hệ thống gây treo hệ thống, không xử lý được tiếp

Priority: Mức độ ưu tiên trong việc sửa lỗi:

* Immediate(1): Bug sẽ phải sửa ngay lập tức, nếu không công việc sẽ không thể tiếp tục
* High (2): Độ ưu tiên cao, công việc sẽ bị ngăn trở rất nhiều nếu như lỗi vẫn chưa được sửa
* Medium (3): Độ ưu tiên trung bình, công việc sẽ gặp vài khó khăn nếu như lỗi vẫn chưa được sửa
* Low(4): Độ ưu tiên thấp nhất, công việc không bị ảnh hưởng nhưng lỗi vẫn phải được sửa

Desercription: Đây là phần mô tả lỗi, phải mô tả rõ 3 phần nội dung:

* Các bước thực hiện
* Kết quả thực tế
* Kết quả mong muốn

Bước 2: Gián lỗi cho nhân viên phát triển

* Thực hiện gán lỗi cho nhân viên phát triển
* Trạng thái của lỗi là New

Bước 3: Xử lý lỗi

* Nhân viên phát triển xem xét các lỗi được gán cho mình
* Sửa lỗi và chuyển về trạng thái Resolved nếu thấy đúng là lỗi
* Nếu thấy không phải lỗi thì chuyển về trạng thái Rejected và nêu lý do trong phần Note

Bước 4: Kiểm thử lại

* Thực hiện kiểm thử lại các lỗi đang ở trạng thái Resolved
* Nếu lỗi đã được sửa thì chuyển về trạng thái Closed
* Nếu lỗi chưa được sửa hoặc mới sửa một phần thì chuyển về trạng thái Open

Graphical user interface, text, application, email

Description automatically generated

Hình 2- 17: Danh sách các lỗi của dự án

Graphical user interface, text, application, email

Description automatically generated

Hình 2- 18: Lỗi của chức năng Thêm mới tài khoản người dùng

### **Kết quả kiểm thử**

Tổng số trường hợp kiểm thử: 60

Số trường hợp kiểm thử thành công: 52

Số trường hợp kiểm thử không thành công: 6

Xét về tổng thể giao diện:

- Giao diện dễ hiểu, dễ sử dụng, chức năng của các phím hoạt động tốt

- Khi thu phóng màn hình không bị mất dữ liệu

- Các button hoạt động tốt

Tuy nhiên vẫn còn 1 số lỗi như:

* Thông báo khi nhập trùng tên tài khoản đang ở dạng SQL
* Chưa có highlight và chuột focus vào trường lỗi

Báo cáo kiểm thử

Graphical user interface, table

Description automatically generated

Hình 2- 19: Báo cáo kiểm thử

## **Viết hướng dẫn sử dụng phần mềm**

Sử dùng phần mềm HelpNdoc để viết tài liệu hướng dẫn sử dụng phần mềm cho khách hàng

Graphical user interface, application, Word

Description automatically generated

Hình 2- Màn hình viết tài liệu hướng dẫn sử dụng

# **CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN**

## **Đánh giá kết quả thực tập**

* Về kiến thức
* Củng cố kiến thức về kiểm thử
* Học được thêm công cụ mới: Redmine, Postman,…
* Được đào tạo về các quy trình phát triển phần mềm đang được vận dụng tại công ty
* Học được những kiến thức sâu hơn ở trường và biết cách ứng dụng vào thực tế
* Về kỹ năng
* Nhận diện, phân tích và thiết kế kịch bản kiểm thử
* Củng cố khả năng giao tiếp trong môi trường công sở
* Khả năng tự học, tự nghiên cứu qua tài liệu được nâng cao
* Học được cách làm việc teamwork
* Khả năng lên kế hoạch công việc và quản lý thời gian
* Kỷ luật và chấp hành đúng giờ làm việc
* Hạn chế

Do thời gian còn hạn chế và kinh nghiệm thực tế chưa có nhiều nên không thể tránh khỏi những thiếu sót về kiến thức, kỹ năng cũng như kết quả đạt được. Em sẽ cố gắng khắc phục những hạn chế đó để có thể hoàn thành tốt bài báo cáo trong đợt Bảo vệ đồ án sắp tới.

## **Đinh hướng phát triển của đề tài**

Khi nghiên cứu về kiểm thử phần mềm, em đã hiểu được kiểm thử rất quan trọng trong quy trình sản xuất phần mềm. Sự áp dụng vơi kiến thức tìm hiểu được mới chỉ dừng lại ở 1 bài toán nhỏ. Hướng phát triển thực tập tốt nghiệp của em là

* Thực hiện kiểm thử trên mô hình bài toán phần mềm rộng hơn, phức tạp hơn
* Tìm hiểu và nghiên cứu thêm về các công cụ kiểm thử tự động, kiểm thử CSDL,…

KẾT LUẬN

Kiểm thử phần mềm là một hoạt động quan trọng trong đảm bảo chất lượng phần mềm để đảm bảo độ tin cậy và chất lượng của phần mềm .Mục đích chính của kiểm thử là phát hiện ra các lỗi phần mềm để từ đó khắc phục và sửa chữa. Việc kiểm thử không thể khẳng định được rằng các chức năng của sản phẩm đúng trong mọi điều kiện, mà chỉ có thể khẳng định rằng nó không hoạt động đúng trong những điều kiện cụ thể.

Trong thời gian tìm hiểu về kiểm thử phần mềm, em đã hiểu được kiểm thử phần mềm là giai đoạn rất quan trọng trong quy trình sản xuất phần mềm. Sự áp dụng mới chỉ thực hiện trên một bài toán nhỏ. Em xin trân thành cảm ơn cô Lê Thanh Huệ đã giúp đỡ em thực hiện tốt đề tài này.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | N. T. Thanh, Bài giảng điện tử môn học Kiểm thử và đảm bảo chất lượng phần mềm, Bộ môn Công nghệ phần mềm, Đại học Mỏ - Địa Chất. |
| [2] | Đ. D. Thu, Bài giảng Kiểm thử phần mềm, (Trung tâm đào tạo TesterTOP),Hà Nội, 2022. |
| [3] | G. Nguyen, API testing với Postman, 2018. |