TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG HÀ NỘI KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

Học phần : Kiểm Thử Và Đảm Bảo Chất Lượng Phần Mềm.

ĐỀ TÀI: KIỂM THỬ WEBSITE

QUẢN LÝ NHÂN VIÊN

Giảng viên hướng dẫn : ThS. Vương Minh Phương

Lớp : DH11C13

Nhóm sinh viên thực hiện : Nhữ Công Việt Hoàng (trưởng nhóm)

Lê Đức Anh

Nguyễn Văn Hải

Hoàng Văn Quý

Đinh Tuấn Duyd

**Hà Nội – 2024**

[MỤC LỤC](#_Toc167647859)

[BẢNG PHÂN CÔNG CÔNG VIỆC 5](#_Toc167647860)

[LỜI NÓI ĐẦU 7](#_Toc167647861)

[TÓM TẮT NỘI DUNG 8](#_Toc167647862)

[CHƯƠNG 1: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 9](#_Toc167647863)

[1.1 Khái niệm phần mềm 9](#_Toc167647864)

[1.1.1 Phân loại phần mềm 9](#_Toc167647865)

[1.1.2 Quy trình phát triển phần mềm 10](#_Toc167647866)

[1.1.3 Một số mô hình phát triển phần mềm 11](#_Toc167647867)

[1.1.4 Lỗi phần mềm 14](#_Toc167647868)

[1.2 Khái niệm về kiểm thử phần mềm 15](#_Toc167647869)

[1.2.1 Kiểm thử phần mềm là gì? 15](#_Toc167647870)

[1.2.2 Các phương pháp kiểm thử 17](#_Toc167647871)

[1.2.3 Các chiến lược kiểm thử 17](#_Toc167647872)

[1.2.4 Các cấp độ kiểm thử phần mềm 20](#_Toc167647873)

[1.2.5 Kiểm thử tự động 25](#_Toc167647874)

[1.2.6 Nguyên tắc kiểm thử phần mềm 28](#_Toc167647875)

[CHƯƠNG 2: ĐẶC TẢ YÊU CẦU PHẦN MỀM (SRS) 32](#_Toc167647876)

[2.1. Các tác nhân 33](#_Toc167647877)

[2.2. Các chức năng của hệ thống 33](#_Toc167647878)

[2.3. Biểu đồ use case tổng quát. 34](#_Toc167647879)

[2.5. Đặc tả các use case 36](#_Toc167647880)

[2.5.1 Đăng nhập 36](#_Toc167647881)

[2.5.2 Đăng ký 39](#_Toc167647882)

[2.5.3 Quản lý nhân viên 43](#_Toc167647883)

[3. Các yêu cầu phi chức năng 47](#_Toc167647884)

[CHƯƠNG 3: TRIỂN KHAI KIỂM THỬ 49](#_Toc167647885)

[3.1 Kiểm thử hộp đen 49](#_Toc167647886)

[3.2 Kiểm thử dịch chuyển trạng thái 63](#_Toc167647887)

[3.3 Kiểm thử hộp trắng 67](#_Toc167647888)

[CHƯƠNG 4. KIỂM THỬ TỰ ĐỘNG 75](#_Toc167647889)

[4.1 Công cụ kiểm thử Selenium 75](#_Toc167647890)

[4.2 Công cụ kiểm thử Katalon 82](#_Toc167647891)

[4.3 Công cụ kiểm thử Jmeter 93](#_Toc167647892)

[4.4 Công cụ kiểm thử Telerik Test Studio 97](#_Toc167647893)

[4.5 Công cụ kiểm thử LoadStom 104](#_Toc167647894)

[LoadStorm tạo tải HTTP từ đám mây dựa trên kế hoạch thử nghiệm và kịch bản người dùng . Loadstorm được sử dụng như một thử nghiệm tải, kiểm tra căng thẳng hoặc công cụ kiểm tra độ bền 105](#_Toc167647895)

[KẾT LUẬN 111](#_Toc167647896)

[TÀI LIỆU THAM KHẢO 112](#_Toc167647897)

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **TÊN SINH VIÊN** | **MSSV** | **Phân công nhiệm vụ** | **Tự đánh giá điểm** | **Đánh giá mức độ hoàn thành** |
| 1 | Nhữ Công Việt Hoàng | 21111064612 | -Phân tích các yêu cầu  -Thiết kế Website  -Mô tả các Module  -Tìm hiểu vai trò, qui trình thiết kế Test case  -Kiểm thử hộp đen  - Demo công cụ kiểm thử Nessus  -Phân tích chiến lược kiểm thử  -Tìm hiểu và kiểm thử công cụ Katalon | 8.5 | 95 |
| 2 | Đinh Tuấn Duy | 21111064581 | -Kiểm thử thủ công: Thiết kế test case  -Phân tích các yêu cầu  -Mô tả các Module  -Tìm hiểu vai trò, qui trình thiết kế Test case  -Kiểm thử hộp đen  - Demo công cụ kiểm thử Nessus  -Phân tích chiến lược kiểm thử  -Tìm hiểu và kiểm thử công cụ Katalon | 8.5 | 95 |
| 3 | Nguyễn Văn Hải | 21111064598 | -Kiểm thử thủ công: Thiết kế test case  -Phân tích các yêu cầu  -Mô tả các Module  -Tìm hiểu vai trò, qui trình thiết kế Test case  -Kiểm thử hộp đen  - Demo công cụ kiểm thử Nessus  -Phân tích chiến lược kiểm thử  -Tìm hiểu và kiểm thử công cụ Katalon | 8.5 | 95 |
| 4 | Lê Đức Anh | 21111065061 | -Kiểm thử thủ công: Thiết kế test case  -Phân tích các yêu cầu  -Mô tả các Module  -Tìm hiểu vai trò, qui trình thiết kế Test case  -Kiểm thử hộp đen  - Demo công cụ kiểm thử Nessus  -Phân tích chiến lược kiểm thử  -Tìm hiểu và kiểm thử công cụ Katalon | 8.5 | 95 |
| 5 | Hoàng Văn Quý | 21111064651 | - Phân tích các yêu cầu  - Mô tả chương trình và nội dung kiểm thử  - Tìm hiểu các khái niệm về kiểm thử  - Tìm hiểu các công cụ kiểm thử  - Tìm hiểu các các nguyên tắc kiểm thử phần mềm  - Lên kế hoạch kiểm thử  - Mô tả chương trình và nội dung kiểm thử  - Kiểm thử thủ công: Thiết kế test case  - Tổng hợp nội dung làm Word đồ án | 8.5 | 95 |

LỜI NÓI ĐẦU

Trong ngành kỹ nghệ phần mềm, năm 1979, có một quy tắc nổi tiếng là: “Trong một dự án lập trình điển hình, thì xấp xỉ 50% thời gian và hơn 50% tổng chi phí được sử dụng trong kiểm thử các chương trình hay hệ thống đã được phát triển”. Và cho đến nay, sau gần một phần 3 thế kỷ, quy tắc đó vẫn còn đúng. Đã có rất nhiều ngôn ngữ, hệ thống phát triển mới với các công cụ tích hợp cho các lập trình viên sử dụng phát triển ngày càng linh động. Nhưng kiểm thử vẫn đóng vai trò hết sức quan trọng trong bất kỳ dự án phát triển phần mềm nào.

Rất nhiều các giáo sư, giảng viên đã từng than phiền rằng: “ Sinh viên của chúng ta tốt nghiệp và đi làm mà không có được những kiến thực thực tế cần thiết về cách để kiểm thử một chương trình. Hơn nữa, chúng ta hiếm khi có được những lời khuyên bổ ích để cung cấp trong các khóa học mở đầu về cách một sinh viên nên làm về kiểm thử và gỡ lỗi các bài tập của họ”.

Báo cáo này được chuẩn bị để cung cấp cái nhìn tổng quan về quy trình kiểm thử phần mềm và những thách thức phổ biến mà chúng ta thường gặp phải. Qua việc phân tích và thảo luận các phương pháp, ưu điểm và nhược điểm của kiểm thử tự động, chúng ta sẽ có cái nhìn sâu hơn về cách thức áp dụng và tối ưu hóa quy trình kiểm thử trong môi trường làm việc của mình.

Với sự phát triển không ngừng của công nghệ và sự yêu cầu ngày càng tăng về chất lượng phần mềm, việc hiểu rõ và áp dụng các phương pháp kiểm thử hiệu quả là chìa khóa để đảm bảo sản phẩm của chúng ta đáp ứng được mong đợi của người dùng cuối.

Chúng tôi hy vọng rằng báo cáo này sẽ cung cấp cho bạn cái nhìn toàn diện và hữu ích về kiểm thử phần mềm, đồng thời giúp bạn nắm vững những khái niệm cơ bản cũng như tiên tiến nhất trong lĩnh vực này.

TÓM TẮT NỘI DUNG

* Bản báo cáo được chia thành 4 chương với nội dung như sau:

Chương 1: Tổng quan về kiểm thử phần mềm.

Chương này là cái nhìn tổng quan về kiểm thử phần mềm: các khái niệm cơ bản về kiểm thử phần mềm, các quy tắc trong kiểm thử, và các phương pháp kiểm thử phần mềm tiêu biểu.

Chương 2: Thiết kế test – case trong kiểm thử phần mềm.

Trong chương này, em đi tìm hiểu các phương pháp thiết kế test – case có hiệu quả. Từ đó rút ra nhận xét về ưu nhược điểm của từng phương pháp.

Chương 3: Triển khai kiểm thử.

Từ những phương pháp thiết kế test – case đã tìm hiểu trong Chương 2, em áp dụng để xây dựng tập các test – case cho một bài toán cụ thể : tiến hành kiểm thử hộp đen, hộp trắng cho websie “Rick Story”.

Chương 4: Kiểm thử tự động

Trong chương này nhóm sẽ ứng dụng một số công cụ kiểm thử tự động vào các chức năng của website sau đó đưa ra kết quả thực nghiệm

1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

## Khái niệm phần mềm

Phần mềm (Software) có thể hiểu là một tập hợp các tập tin có mối liên hệ chặt chẽ với nhau, đảm bảo thực hiện một số nhiệm vụ, chức năng nào đó trên thiết bị điện tử. Các tập tin này có thể bao gồm: các file mã nguồn viết bằng một hoặc nhiều ngôn ngữ lập trình, các file dữ liệu (thư viện), các file hướng dẫn.

### Phân loại phần mềm

Phần mềm được chia thành 2 loại chính xét theo từng mục đích sử dụng khác nhau:

* Theo phương thức hoạt động: Phân theo phương thức hoạt động thì được chia thành các loại khác nhau bao gồm các loại như sau :
  + Phần mềm hệ thống dùng để vận hành máy tính nói riêng và các thiết bị điện tử nói chung. Ví dụ: hệ điều hành máy tính Windows, Linux, Unix; Các trình điều khiển (driver), phần sụn (firmware) và BIOS. Hệ điều hành di động ios, Android, Windows Phone,…
  + Phần mềm ứng dụng – phần mềm máy tính : Các phần mềm văn phòng (Microsoft Office, openoffice), trò chơi điện tử (game), các công cụ & tiện ích khác (ví dụ như phần mềm quản lý chi tiêu cá nhân, phần mềm quản lý công việc,…).
  + Phần mềm dịch mã (trình dịch) Bao gồm trình biên dịch và trình thông dịch, cụ thể là chúng dịch các câu lệnh từ mã nguồn của ngôn ngữ lập trình sang dạng ngôn ngữ máy sao cho thiết bị thực thi có thể hiểu được.
  + Nền tảng ứng dụng: như ASP.NET – nền tảng ứng dụng web của Microsoft, cái này hỗ trợ việc tạo ra các ứng dụng web, dịch vụ web (web service).
* Theo khả năng hay quyền hạn can thiệp vào mã nguồn thì được chia thành các loại như sau:
  + Phần mềm mã nguồn đóng (closed source software): Là phần mềm mà mã nguồn của nó không được công bố. Để sử dụng phần mềm nguồn đóng phải được cấp bản quyền (mua, tặng là tùy).
  + Phần mềm mã nguồn mở (open source software): Là phần mềm mà mã nguồn của nó được công bố rộng rãi, công khai và cho phép mọi người tiếp tục phát triển phần mềm đó. Thường thì loại phần mềm này miễn phí.

### Quy trình phát triển phần mềm

Quy trình phát triển phần mềm là một cấu trúc bao gồm tập hợp các thao tác và các kết quả tương quan sử dụng trong việc phát triển để sản xuất ra một sản phẩm phần mềm. Nhìn chung, một quy trình phát triển phần mềm bao gồm 6 giai đoạn chính như sau:

* Analysis : Thu thập yêu cầu
  + Nhiệm vụ: Thực hiện khảo sát chi tiết yêu cầu của khách hàng để xác định và hiểu rõ nhu cầu và mong đợi của khách hàng. Thu thập thông tin từ người sử dụng cuối và các bên liên quan khác. Từ đó tổng hợp vào tài liệu giải pháp. Tài liệu này phải mô tả đầy đủ các yêu cầu về chức năng, phi chức năng và giao diện: Thông số kỹ thuật, giao diện mẫu…
  + Kết quả: Đầu ra của giai đoạn này là tài liệu đặc tả yêu cầu ….
* Design: Thiết kế
  + Nhiệm vụ : Thực hiện thiết kế và tổng hợp vào tài liệu thiết kế. Xây dựng kiến trúc và thiết kế hệ thống dựa trên yêu cầu. Phát triển các bản thiết kế chi tiết cho các thành phần phần mềm.Thiết kế giao diện, kiến trúc ( backend, frontend, database ….)
* Development: Phát triển
  + Nhiệm vụ:
    - Lập trình viên thực hiện lập trình và triển khai các tính năng của phần mềm dựa trên tài liệu Giải pháp và Thiết kế đã được phê duyệt. Kiểm thử đơn vị để đảm bảo tính chính xác và hoạt động đúng đắn của từng phần.
    - Code, test sơ bộ …
  + Kết quả: Source code.
* Testing: Kiểm thử
  + Nhiệm vụ: Tester tạo kịch bản kiểm thử (test case) theo tài liệu đặc tả yêu cầu, thực hiện kiểm thử và cập nhật kết quả vào kịch bản kiểm thử, log lỗi trên các tool quản lý lỗi.
  + Kết quả: Test case , lỗi trên hệ thống quản lý lỗi.
* Deployment: Triển khai
  + Nhiệm vụ: Đưa phần mềm vào môi trường sản xuất. Huấn luyện người dùng cuối và hỗ trợ triển khai sản phẩm cho khách hàng.
  + Kết quả: Biên bản triển khai với khách hàng.
* Maintenance and Support : Bảo trì và Hỗ trợ
  + Nhiệm vụ : Cung cấp bảo trì, nâng cấp và hỗ trợ kỹ thuật sau khi phần mềm đã đi vào hoạt động. Đáp ứng các yêu cầu cải tiến và sửa lỗi của người dùng.
  + Kết quả : Duy trì và vá lỗi được sản phẩm.

### Một số mô hình phát triển phần mềm

#### Mô hình chữ V

Hình 1. 1 Mô Hình chữ V

Giới thiệu về mô hình chữ V:

Trong kiểm thử phần mềm (V Model) còn được gọi là mô hình xác thực hay mô hình xác minh. Nó là mô hình có tính kỷ luật cao, trong đó, giai đoạn kiểm thử chạy song song với mỗi giai đoạn phát triển phần mềm.

Đặc điểm điểm của mô hình:

* Hoạt động tốt với các dự án có quy mô vừa và nhỏ.
* Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.
* Toàn bộ quy trình được chia thành 2 nhóm giai đoạn tương ứng nhau là phát triển và kiểm thử. Mỗi giai đoạn phát triển sẽ tiến hành song song với một giai đoạn kiểm thử tương ứng. Do đó, các lỗi được phát hiện sớm ngay từ đầu.

Ưu điểm nhược điểm của mô hình chữ V:

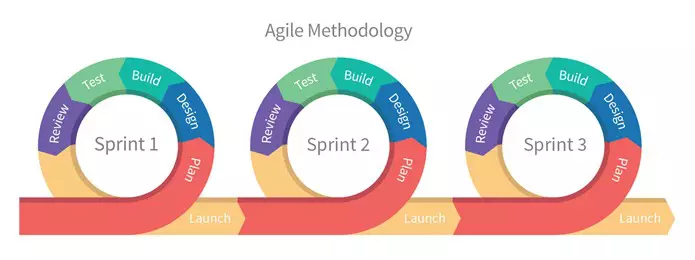
* Ưu điểm:

Ngay từ lúc nhận được tài liệu đặc tả yêu cầu, tester sẽ tham gia vào review tài liệu đặc tả yêu cầu sau đó lên kế hoạch và thực hiện viết test case. Lỗi được phát hiện từ giai đoạn này sẽ ít tốn thời gian và chi phí hơn các giai đoạn sau.

* Nhược điểm:

Trong mô hình chữ V, các yêu cầu vẫn được đưa vào thực hiện cùng 1 lúc mà rủi ro về thay đổi yêu cầu từ phía khách hàng là rất lớn. Do đó, mô hình này vẫn có thể gặp rắc rối khi khách hàng thường xuyên thay đổi yêu cầu.

#### Mô hình Agile (Quy trình Scrum)



Hình 1. 2 Mô Hình Agile

Giới thiệu về mô hình Agile (quy trình Scrum): Scrum trong Agile là một quy trình cho phép các đội dự án phát triển phần mềm tập trung vào việc cung cấp các giá trị kinh doanh trong thời gian ngắn nhất ( theo sprint) bằng cách test nhanh chóng và liên tục phần mềm hoạt động thực tế.

* **Đặc điểm của mô hình Agile:** là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt.
* **Đặc trưng của mô hình Agile :** Tính lặp (Interactive), tính tiệm tiến và tiến hóa, tính thích ứng
* Bao gồm 4 cuộc họp như sau:

Sprint Planning (Họp Kế hoạch Sprint)

Nhóm phát triển họp với Product Owner để lên kế hoạch làm việc cho một Sprint. Công việc lập kế hoạch bao gồm việc chọn lựa các yêu cầu cần phải phát triển, phân tích và nhận biết các công việc phải làm kèm theo các ước lượng thời gian cần thiết để hoàn tất các tác vụ. Scrum sử dụng cách thức lập kế hoạch từng phần và tăng dần theo thời gian, theo đó, việc lập kế hoạch không diễn ra duy nhất một lần trong vòng đời của dự án mà được lặp đi lặp lại, có sự thích nghi với các tình hình thực tiễn trong tiến trình đi đến sản phẩm.

Daily Scrum (Họp Scrum hằng ngày)

Scrum Master tổ chức cho Đội sản xuất họp hằng ngày trong khoảng 15 phút để Nhóm Phát triển chia sẻ tiến độ công việc Trong cuộc họp này, từng người trong nhóm phát triển lần lượt trình bày để trả lời 3 câu hỏi sau:

* Hôm qua đã làm gì?
* Hôm nay sẽ làm gì?
* Có khó khăn trở ngại gì không?

Sprint Review (Họp Sơ kết Sprint)

Cuối Sprint, nhóm phát triển cùng với Product Owner sẽ rà soát lại các công việc đã hoàn tất (DONE) trong Sprint vừa qua và đề xuất các chỉnh sửa hoặc thay đổi cần thiết cho sản phẩm.

Sprint Retrospective (Họp Cải tiến Sprint)

Dưới sự trợ giúp của Scrum Master, nhóm phát triển sẽ rà soát lại toàn diện Sprint vừa kết thúc và tìm cách cải tiến quy trình làm việc cũng như bản thân sản phẩm.

Ưu điểm:

Phù hợp với các yêu cầu / nghiệp vụ hay thay đổi, hoặc hệ thống nghiên cứu do làm theo từng giai đoạn ngắn ngày, có thể nhìn thấy những rủi ro hay những điểm chưa phù hợp để thay đổi.

Nhược điểm:

* Thiếu sự nhấn mạnh về thiết kế và tài liệu cần thiết
* Quy mô nhân lực thường giới hạn , sẽ có trở ngại lớn nếu nguồn nhân lực yêu cầu vượt quá con số này ví dụ trong các cuộc họp trao đổi.
* Yêu cầu nguồn nhân lực phải có kiến thức và am hiểu về Agile.

### Lỗi phần mềm

Lỗi phần mềm là một lỗi hay hỏng hóc trong chương trình hoặc hệ thống máy tính khiến nó tạo ra kết quả không chính xác hoặc không mong muốn hoặc hành xử theo những cách không lường trước được. Quá trình tìm và sửa lỗi được gọi là "gỡ lỗi" và thường sử dụng các kỹ thuật hoặc công cụ chính thức để xác định lỗi và từ những năm 1950, một số hệ thống máy tính đã được thiết kế để ngăn chặn, phát hiện hoặc tự động sửa các lỗi máy tính khác nhau trong quá trình hoạt động.

Lỗi phần mềm có thể là kết quả của nhiều yếu tố khác nhau, bao gồm:

* Lỗi lập trình: Các lỗi này xuất phát từ việc viết mã nguồn không chính xác hoặc không hoàn chỉnh. Điều này có thể bao gồm việc sử dụng cú pháp sai, logic lập trình không đúng, hoặc quy trình xử lý dữ liệu không chính xác.
* Lỗi thiết kế: Các lỗi này liên quan đến việc thiết kế không chính xác của phần mềm. Điều này có thể bao gồm thiết kế không đáp ứng được yêu cầu của khách hàng hoặc không tối ưu hóa được hiệu suất hoặc bảo mật của phần mềm.
* Lỗi hệ thống: Các lỗi này xuất phát từ các thành phần hệ thống không tương thích hoặc không hoạt động đúng đắn khi kết hợp với nhau. Điều này có thể bao gồm các vấn đề liên quan đến mạng, cơ sở dữ liệu, hoặc các dịch vụ bên ngoài.
* Lỗi môi trường: Các lỗi này liên quan đến các yếu tố môi trường như hệ điều hành, phần cứng, hoặc phần mềm bên thứ ba. Điều này có thể bao gồm các vấn đề liên quan đến sự không tương thích giữa các phiên bản phần mềm, hoặc môi trường triển khai không đúng.
* Lỗi do người dùng: Các lỗi này xuất phát từ việc sử dụng phần mềm không đúng cách hoặc không đáp ứng được nhu cầu của người dùng. Điều này có thể bao gồm việc nhập dữ liệu không chính xác, hoặc không hiểu rõ về các tính năng và chức năng của phần mềm

## Khái niệm về kiểm thử phần mềm

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là quá trình khảo sát một hệ thống hay thành phần dưới những điều kiện xác định, quan sát và ghi lại các kết quả, và đánh giá một khía cạnh nào đó của hệ thống hay thành phần đó. (Theo Bảng chú giải thuật ngữ chuẩn IEEE của Thuật ngữ kỹ nghệ phần mềm- IEEE Standard Glossary of Software Engineering Terminology).

Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi. (Theo “The Art of Software Testing” – Nghệ thuật kiểm thử phần mềm).

Kiểm thử phần mềm là hoạt động khảo sát thực tiễn sản phẩm hay dịch vụ phần mềm trong đúng môi trường chúng dự định sẽ được triển khai nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm ấy. Mục đích của kiểm thử phần mềm là tìm ra các lỗi hay khiếm khuyết phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau. (Theo Bách khoa toàn thư mở Wikipedia).

Có thể định nghĩa một cách dễ hiểu như sau: Kiểm thử phần mềm là một tiến trình hay một tập hợp các tiến trình được thiết kế để đảm bảo một ứng dụng hoặc hệ thống phần mềm hoạt động đúng đắn, theo đúng các yêu cầu và mong đợi của người sử dụng. Mục tiêu chính của kiểm thử phần mềm là phát hiện các lỗi, sai sót và vấn đề tiềm ẩn trước khi phần mềm được triển khai vào môi trường sản xuất.

Quá trình kiểm thử phần mềm bao gồm việc thiết kế các ca kiểm thử, thực hiện các ca kiểm thử này, đánh giá kết quả, và báo cáo về chất lượng của phần mềm. Nó là một quy trình lặp đi lặp lại, với mỗi vòng lặp cải thiện chất lượng của phần mềm bằng cách phát hiện và khắc phục các lỗi.

Các loại kiểm thử phần mềm có thể bao gồm:

* Kiểm thử đơn vị (Unit Testing): Kiểm thử từng đơn vị code nhỏ nhất của phần mềm để đảm bảo chúng hoạt động như mong đợi.
* Kiểm thử tích hợp (Integration Testing): Kiểm thử các thành phần phần mềm đã được kết hợp lại với nhau để đảm bảo chúng tương tác đúng cách.
* Kiểm thử hệ thống (System Testing): Kiểm thử toàn bộ hệ thống phần mềm như một thực thể độc lập để đảm bảo nó hoạt động đúng đắn và đáp ứng yêu cầu.
* Kiểm thử chấp nhận người dùng (User Acceptance Testing - UAT): Kiểm thử bởi người dùng cuối để đảm bảo rằng phần mềm đáp ứng được nhu cầu và mong đợi của họ.
* Kiểm thử hiệu suất (Performance Testing): Kiểm thử hiệu suất và khả năng chịu tải của phần mềm dưới điều kiện tải tốc độ và sức tải khác nhau.
* Kiểm thử bảo mật (Security Testing): Kiểm thử để xác định các lỗ hổng bảo mật và đảm bảo rằng phần mềm không bị tấn công hoặc xâm nhập.

### Các phương pháp kiểm thử

Có 2 phương pháp kiểm thử chính là: Kiểm thử tĩnh và Kiểm thử động.

#### Kiểm thử tĩnh – Static testing

Là phương pháp thử phần mềm đòi hỏi phải duyệt lại các yêu cầu và các đặc tả bằng tay, thông qua việc sử dụng giấy, bút để kiểm tra logic, lần từng chi tiết mà không cần chạy chương trình. Kiểu kiểm thử này thường được sử dụng bởi chuyên viên thiết kế người mà viết mã lệnh một mình.

Kiểm thử tĩnh cũng có thể được tự động hóa. Nó sẽ thực hiện kiểm tra toàn bộ bao gồm các chương trình được phân tích bởi một trình thông dịch hoặc biên dịch mà xác nhận tính hợp lệ về cú pháp của chương trình.

#### Kiểm thử động – Dynamic testing

Là phương pháp thử phần mềm thông qua việc dùng máy chạy chương trình để điều tra trạng thái tác động của chương trình. Đó là kiểm thử dựa trên các ca kiểm thử xác định bằng sự thực hiện của đối tượng kiểm thử hay chạy các chương trình. Kiểm thử động kiểm tra cách thức hoạt động động của mã lệnh, tức là kiểm tra sự phản ứng vật lý từ hệ thống tới các biến luôn thay đổi theo thời gian. Trong kiểm thử động, phần mềm phải thực sự được biên dịch và chạy. Kiểm thử động thực sự bao gồm làm việc với phần mềm, nhập các giá trị đầu vào và kiểm tra xem liệu đầu ra có như mong muốn hay không. Các phương pháp kiểm thử động gồm có kiểm thử đơn vị – Unit Tests, Kiểm thử tích hợp – Intergration Tests, Kiểm thử hệ thống – System Tests, và Kiểm thử chấp nhận sản phẩm – Acceptance Tests.

### Các chiến lược kiểm thử

Ba trong những chiến lược kiểm thử thông dụng nhất gồm: kiểm thử hộp đen, kiểm thử hộp trắng, kiểm thử hộp xám.

#### Kiểm thử hộp đen – Black box testing

Kiểm tra hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.

Các phương pháp kiểm thử:

* Phân tích giá trị biên – boundary value analysis.
* Phân lớp tương đương – equivalence partitioning.
* Kiểm thử mọi cặp – all-pairs testing.
* Kiểm thử fuzz – fuzz testing.
* Kiểm thử dựa trên mô hình – model-based testing.
* Ma trận dấu vết – traceability matrix.
* Kiểm thử thăm dò – exploratory testing.
* Kiểm thử dựa trên đặc tả – specification-base testing.

Kiểm thử dựa trên đặc tả tập trung vào kiểm tra tính thiết thực của phần mềm theo những yêu cầu thích hợp. Do đó, kiểm thử viên nhập dữ liệu vào, và chỉ thấy dữ liệu ra từ đối tượng kiểm thử. mức kiểm thử này thường yêu cầu các ca kiểm thử triệt để được cung cấp cho kiểm thử viên mà khi đó có thể xác minh là đối với dữ liệu đầu vào đã cho, giá trị đầu ra (hay cách thức hoạt động) có giống với giá trị mong muốn đã được xác định trong ca kiểm thử đó hay không. kiểm thử dựa trên đặc tả là cần thiết, nhưng không đủ để để ngăn chặn những rủi ro chắc chắn.

Ưu và nhược điểm:

Kiểm thử hộp đen không có mối liên quan nào tới mã lệnh, và kiểm thử viên chỉ rất đơn giản tâm niệm là: một mã lệnh phải có lỗi. sử dụng nguyên tắc “hãy đòi hỏi và bạn sẽ được nhận”, những kiểm thử viên hộp đen tìm ra lỗi mà những lập trình viên đã không tìm ra. nhưng, mặt khác, người ta cũng nói kiểm thử hộp đen “giống như là đi trong bóng tối mà không có đèn vậy”, bởi vì kiểm thử viên không biết các phần mềm được kiểm tra thực sự được xây dựng như thế nào. đó là lý do mà có nhiều trường hợp mà một kiểm thử viên hộp đen viết rất nhiều ca kiểm thử để kiểm tra một thứ gì đó mà đáng lẽ có thể chỉ cần kiểm tra bằng 1 ca kiểm thử duy nhất, và/hoặc một số phần của chương trình không được kiểm tra chút nào.

Do vậy, kiểm thử hộp đen có ưu điểm của “một sự đánh giá khách quan”, mặt khác nó lại có nhược điểm của “thăm dò mù”.

#### Kiểm thử hộp trắng – White box testing

Là một chiến lược kiểm thử khác, trái ngược hoàn toàn với kiểm thử hộp đen, kiểm thử hộp trắng hay kiểm thử hướng logic cho phép bạn khảo sát cấu trúc bên trong của chương trình. Chiến lược này xuất phát từ dữ liệu kiểm thử bằng sự kiểm thử tính logic của chương trình. Kiểm thử viên sẽ truy cập vào cấu trúc dữ liệu và giải thuật bên trong chương trình (và cả mã lệnh thực hiện chúng).

Các phương pháp kiểm thử hộp trắng

* Kiểm thử giao diện lập trình ứng dụng - API testing (application programming interface): là phương pháp kiểm thử của ứng dụng sử dụng các API công khai và riêng tư.
* Bao phủ mã lệnh – Code coverage: tạo các kiểm tra để đáp ứng một số tiêu chuẩn về bao phủ mã lệnh.
* Các phương pháp gán lỗi – Fault injection.
* Các phương pháp kiểm thử hoán chuyển – Mutation testing methods.
* Kiểm thử tĩnh – Static testing: kiểm thử hộp trắng bao gồm mọi kiểm thử tĩnh.

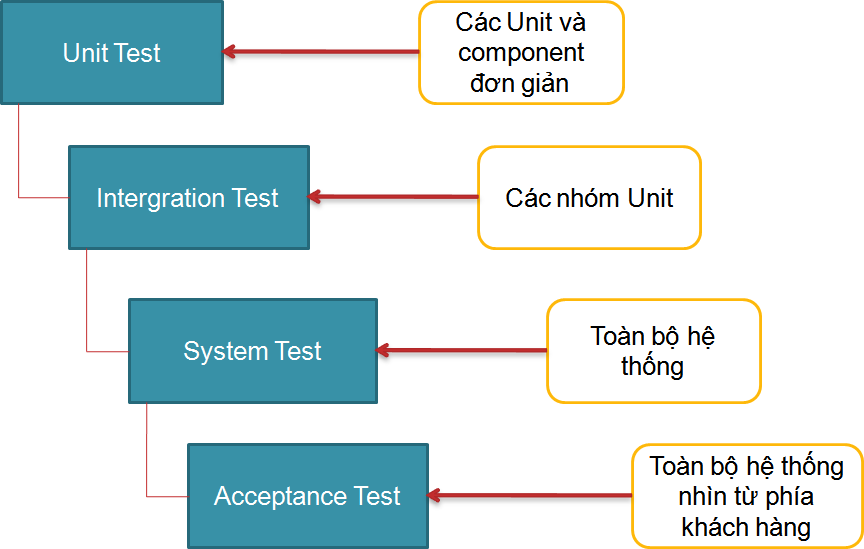
Phương pháp kiểm thử hộp trắng cũng có thể được sử dụng để đánh giá sự hoàn thành của một bộ kiểm thử mà được tạo cùng với các phương pháp kiểm thử hộp đen. Điều này cho phép các nhóm phần mềm khảo sát các phần của 1 hệ thống ít khi được kiểm tra và đảm bảo rằng những điểm chức năng quan trọng nhất đã được kiểm tra.

#### Kiểm thử hộp xám – Gray box testing

Kiểm thử hộp xám đòi hỏi phải có sự truy cập tới cấu trúc dữ liệu và giải thuật bên trong cho những mục đích thiết kế các ca kiểm thử, nhưng là kiểm thử ở mức người sử dụng hay mức hộp đen. Việc thao tác tới dữ liệu đầu vào và định dạng dữ liệu đầu ra là không rõ ràng, giống như một chiếc “hộp xám”, bởi vì đầu vào và đầu ra rõ ràng là ở bên ngoài “hộp đen” mà chúng ta vẫn gọi về hệ thống được kiểm tra. Sự khác biệt này đặc biệt quan trọng khi quản lý kiểm thử tích hợp – Intergartion testing giữa 2 modun mã lệnh được viết bởi hai chuyên viên thiết kế khác nhau, trong đó chỉ giao diện là được đưa ra để kiểm thử. Kiểm thử hộp xám có thể cũng bao gồm cả thiết kế đối chiếu để quyết định, ví dụ, giá trị biên hay thông báo lỗi.

### Các cấp độ kiểm thử phần mềm

Kiểm thử phần mềm gồm có các cấp độ: Kiểm thử đơn vị, Kiểm thử tích hợp, Kiểm thử hệ thống và Kiểm thử chấp nhận sản phẩm.



Hình 1.3 Sơ đồ các cấp độ kiểm thử

#### Kiểm thử đơn vị – Unit test

Một đơn vị là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm thử được. Ví dụ, các hàm (Function), thủ tục (Procedure), lớp (Class) hay phương thức (Method) đều có thể được xem là Unit.

Vì Unit được chọn để kiểm tra thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng ta không khó khăn gì trong việc tổ chức kiểm thử, ghi nhận và phân tích kết quả kiểm thử. Nếu phát hiện lỗi, việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn thể Unit đang kiểm tra. Một nguyên lý đúc

kết từ thực tiễn: thời gian tốn cho Unit Test sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức kiểm thử sau đó.

Unit Test thường do lập trình viên thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chu kỳ phát triển phần mềm. Thông thường, Unit Test đòi hỏi kiểm thử viên có kiến thức về thiết kế và code của chương trình. Mục đích của Unit Test là bảo đảm thông tin được xử lý và xuất (khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của Unit.

Điều này thường đòi hỏi tất cả các nhánh bên trong Unit đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một Unit. Ví dụ: chuỗi các lệnh sau điều kiện If và nằm giữa then ... else là một nhánh. Thực tế việc chọn lựa các nhánh để đơn giản hóa việc kiểm thử và quét hết Unit đòi hỏi phải có kỹ thuật, đôi khi phải dùng thuật toán để chọn lựa.

Cùng với các mục kiểm thử khác, Unit Test cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử (Test case) hoặc kịch bản kiểm thử (Test script), trong đó chỉ định rõ dữ liệu đầu vào, các bước thực hiện và dữ liệu đầu ra mong muốn. Các Test case và Test script này nên được giữ lại để tái sử dụng.

#### Kiểm thử tích hợp – Intergration Test

Integration test kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi Unit Test kiểm tra các thành phần và Unit riêng lẻ thì Intgration Test kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Hai mục tiêu chính của Integration Test:

* Phát hiện lỗi giao tiếp xảy ra giữa các Unit.
* Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ (Subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh (System) chuẩn bị cho kiểm thử ở mức hệ thống (System Test).

Trong Unit Test, lập trình viên cố gắng phát hiện lỗi liên quan đến chức năng và cấu trúc nội tại của Unit. Có một số phép kiểm thử đơn giản trên giao tiếp giữa Unit với các thành phần liên quan khác, tuy nhiên mọi giao tiếp liên quan đến Unit chỉ thật sự được kiểm tra đầy đủ khi các Unit tích hợp với nhau trong khi thực hiện Integration Test.

Trừ một số ít ngoại lệ, Integration Test chỉ nên thực hiện trên những Unit đã được kiểm tra cẩn thận trước đó bằng Unit Test, và tất cả các lỗi mức Unit đã được sửa chữa. Một số người hiểu sai rằng Unit một khi đã qua giai đoạn Unit Test với các giao tiếp giả lập thì không cần phải thực hiện Integration Test nữa. Thực tế việc tích hợp giữa các Unit dẫn đến những tình huống hoàn toàn khác.

Một chiến lược cần quan tâm trong Integration Test là nên tích hợp dần từng Unit. Một Unit tại một thời điểm được tích hợp vào một nhóm các Unit khác đã tích hợp trước đó và đã hoàn tất các đợt Integration Test trước đó. Lúc này, ta chỉ cần kiểm thử giao tiếp của Unit mới thêm vào với hệ thống các Unit đã tích hợp trước đó, điều này sẽ làm cho số lượng can kiểm thử giảm đi rất nhiều, và sai sót sẽ giảm đáng kể.

Có 4 loại kiểm thử trong Integration Test:

* Kiểm thử cấu trúc (Structure Test): Tương tự White Box Test, kiểm thử cấu trúc nhằm bảo đảm các thành phần bên trong của một chương trình chạy đúng và chú trọng đến hoạt động của các thành phần cấu trúc nội tại của chương trình chẳng hạn các câu lệnh và nhánh bên trong.
* Kiểm thử chức năng (Functional Test): Tương tự Black Box Test, kiểm thử chức năng chỉ chú trọng đến chức năng của chương trình, mà không quan tâm đến cấu trúc bên trong, chỉ khảo sát chức năng của chương trình theo yêu cầu kỹ thuật.
* Kiểm thử hiệu năng (Performance Test): Kiểm thử việc vận hành của hệ thống.
* Kiểm thử khả năng chịu tải (Stress Test): Kiểm thử các giới hạn của hệ thống.

#### Kiểm thử hệ thống – System Test

Mục đích System Test là kiểm thử thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không.

System Test bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công. Thông thường loại kiểm thử này tốn rất nhiều công sức và thời gian. Trong nhiều trường hợp, việc kiểm thử đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc thù, đặc biệt là các ứng dụng thời gian thực, hệ thống phân bố, hoặc hệ thống nhúng. Ở mức độ hệ thống, người kiểm thử cũng tìm kiếm các lỗi, nhưng trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và các yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Điểm khác nhau then chốt giữa Integration Test và System Test là System Test chú trọng các hành vi và lỗi trên toàn hệ thống, còn Integration Test chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau. Thông thường ta phải thực hiện Unit Test và Integration Test để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện System Test.

Sau khi hoàn thành Integration Test, một hệ thống phần mềm đã được hình thành cùng với các thành phần đã được kiểm tra đầy đủ. Tại thời điểm này, lập trình viên hoặc kiểm thử viên bắt đầu kiểm thử phần mềm như một hệ thống hoàn chỉnh. Việc lập kế hoạch cho System Test nên bắt đầu từ giai đoạn hình thành và phân tích các yêu cầu.

System Test kiểm thử cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi "tắc nghẽn" (deadlock) hoặc chiếm dụng bộ nhớ. Sau giai đoạn System Test, phần mềm thường đã sẵn sàng cho khách hàng hoặc người dùng cuối cùng kiểm thử chấp nhận sản phẩm (Acceptance Test) hoặc dùng thử (Alpha/Beta Test).

Đòi hỏi nhiều công sức, thời gian và tính chính xác, khách quan, System Test thường được thực hiện bởi một nhóm kiểm thử viên hoàn toàn độc lập với nhóm phát triển dự án. Bản thân System Test lại gồm nhiều loại kiểm thử khác nhau, phổ biến nhất gồm:

* Kiểm thử chức năng (Functional Test): Bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.
* Kiểm thử hiệu năng (Performance Test): Bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (ví dụ bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn…
* Kiểm thử khả năng chịu tải (Stress Test hay Load Test): Bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường như đang giao dịch thì ngắt kết nối (xuất hiện nhiều trong kiểm tra thiết bị như POS, ATM...)...
* Kiểm thử cấu hình (Configuration Test).
* Kiểm thử bảo mật (Security Test): Bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.
* Kiểm thử khả năng phục hồi (Recovery Test): Bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu; đặc biệt quan trọng đối với các hệ thống giao dịch như ngân hàng trực tuyến...

Nhìn từ quan điểm người dùng, các cấp độ kiểm thử trên rất quan trọng: Chúng bảo đảm hệ thống đủ khả năng làm việc trong môi trường thực.

Lưu ý là không nhất thiết phải thực hiện tất cả các loại kiểm thử nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tuỳ khả năng và thời gian cho phép của dự án, khi lập kế hoạch, người Quản lý dự án sẽ quyết định áp dụng những loại kiểm thử nào.

#### Kiểm thử chấp nhận sản phẩm – Acceptance Test

Thông thường, sau giai đoạn System Test là Acceptance Test, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của Acceptance Test là để chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm (và trả tiền thanh toán hợp đồng).

Acceptance Test có ý nghĩa hết sức quan trọng, mặc dù trong hầu hết mọi trường hợp, các phép kiểm thử của System Test và Acceptance Test gần như tương tự, nhưng bản chất và cách thức thực hiện lại rất khác biệt.

Đối với những sản phẩm dành bán rộng rãi trên thị trường cho nhiều người sử dụng, thông thường sẽ thông qua hai loại kiểm thử gọi là kiểm thử Alpha – Alpha Test và kiểm thử Beta – Beta Test. Với Alpha Test, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa. Với Beta Test, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa.

Thực tế cho thấy, nếu khách hàng không quan tâm và không tham gia vào quá trình phát triển phần mềm thì kết quả Acceptance Test sẽ sai lệch rất lớn, mặc dù phần mềm đã trải qua tất cả các kiểm thử trước đó. Sự sai lệch này liên quan đến việc hiểu sai yêu cầu cũng như sự mong chờ của khách hàng. Ví dụ đôi khi một phần mềm xuất sắc vượt qua các phép kiểm thử về chức năng thực hiện bởi nhóm thực hiện dự án, nhưng khách hàng khi kiểm thử sau cùng vẫn thất vọng vì bố cục màn hình nghèo nàn, thao tác không tự nhiên, không theo tập quán sử dụng của khách hàng v.v...

Gắn liền với giai đoạn Acceptance Test thường là một nhóm những dịch vụ và tài liệu đi kèm, phổ biến như hướng dẫn cài đặt, sử dụng v.v... Tất cả tài liệu đi kèm phải được cập nhật và kiểm thử chặt chẽ.

### Kiểm thử tự động

#### Khái niệm về kiểm thử tự động

Kiểm thử tự động: Là xử lý một cách tự động các bước thực hiện các testcase, kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử. Là một kỹ thuật tự động trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Nó về cơ bản là một quá trình tự động hóa của một quy trình kiểm thử thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm thử ứng dụng theo quan điểm tải, hiệu năng và ứng suất.

Ưu điểm của kiểm thử tự động

* Các bài kiểm tra có thể được thực hiện một cách nhanh chóng.
* Có thể phát hiện sớm các lỗi, thực hiện kiểm tra một cách chính xác.
* Loại bỏ lỗi của con người khi thực hiện các bài kiểm tra, cho phép các bài kiểm tra chính xác hơn.
* Có thể thực hiện kể cả thiếu nguồn nhân lực.

Nhược điểm của kiểm thử tự động

* Có thể bỏ sót lỗi do con người.
* Không thể tự động hóa tất cả các bài kiểm thử, cần phải chọn lọc cẩn thận.
* Yêu cầu hoạt động bảo trì định kỳ.
* Chỉ kiểm tra được những gì đã được thiết kế để kiểm thử.
* Kiểm thử tự động không có hiệu quả ngay lập tức.
* Kiểm tra tự động chỉ có thể kiểm tra những gì chúng ta thiết kế.
* Chi phí cao

#### Quy trình kiểm thử tự động

Tester sử dụng các kịch bản tự động (automation scripts) và thực thi các script để chạy ứng dụng với sự giúp sức của các automation tool. Một khi script đã sẵn sàng thì việc thực thi kiểm thử có thể diễn ra nhanh chóng và hiệu quả.

Các hoạt động của kiểm thử tự động:

* Phân tích yêu cầu/Xác định môi trường/công cụ
* Xác định tiêu chí đầu ra
* Lên kế hoạch và kiểm soát
* Thiết lập môi trường kiểm thử
* Triển khai thiết kế kiểm thử
* Thực thi kiểm thử
* Phân tích, báo cáo

#### Mục đích của việc sử dụng kiểm thử tự động

* Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử
* Tăng độ tin cậy.
* Giảm sự nhàm chán cho con người
* Rèn luyện kỹ năng lập trình cho kiểm thử viên
* Giảm chi phí cho tổng quá trình kiểm thử

#### Sử dụng kiểm thử tự động vào khi nào ?

* Không đủ tài nguyên: Khi số lượng testcase quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể.
* Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian.
* Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt (Đo tốc độ trung bình xử lý một yêu cầu của Web server, xác định cấu hình máy thấp nhất mà phần mềm vẫn có thể hoạt động tốt).

#### Các bước triển khai kiểm thử tự động

* Xác định phạm vi kiểm thử: Xác định các tính năng, chức năng và yêu cầu của phần mềm cần được kiểm thử.
* Lập kế hoạch kiểm thử: Xác định chiến lược và kế hoạch kiểm thử, bao gồm lựa chọn các phương pháp kiểm thử, tài nguyên cần thiết và lịch trình.
* Thiết kế các ca kiểm thử (Test Case Design): Xây dựng các ca kiểm thử để kiểm tra từng tính năng hoặc chức năng của phần mềm. Các ca kiểm thử này phải được thiết kế sao cho có thể phát hiện được các lỗi và sai sót có thể xảy ra.
* Thực hiện kiểm thử (Test Execution): Thực hiện các ca kiểm thử đã thiết kế và ghi nhận kết quả của từng ca kiểm thử.
* Phân tích kết quả (Test Result Analysis): Phân tích kết quả kiểm thử để đánh giá hiệu suất và chất lượng của phần mềm. Các lỗi và vấn đề phát hiện được trong quá trình kiểm thử cần được ghi nhận và báo cáo.
* Báo cáo và kiểm tra lại (Reporting and Retesting): Lập báo cáo về kết quả kiểm thử và các vấn đề phát hiện được. Tiến hành kiểm tra lại các lỗi đã được sửa chữa để đảm bảo rằng phần mềm đã được cải thiện.
* Bảo trì kịch bản kiểm thử: Cập nhật và bảo trì các kịch bản kiểm thử để đảm bảo chúng luôn phản ánh đúng các yêu cầu và thay đổi của ứng dụng.

#### Một số công cu ̣kiểm thử tự động

* Visual Studio Test Professional: Hỗ trợ kiểm thử tự động và thủ công, tích hợp chặt chẽ với Azure DevOps.
* TestComplete: Đa nền tảng, hỗ trợ kiểm thử script và không cần script, khả năng nhận diện đối tượng mạnh mẽ.
* Testing Anywhere: Tập trung vào kiểm thử không cần mã, dễ sử dụng với giao diện kéo và thả.
* SilkTest: Hỗ trợ kiểm thử các ứng dụng doanh nghiệp phức tạp, đa nền tảng và trình duyệt.
* Selenium: Công cụ kiểm thử tự động cho các ứng dụng web, hỗ trợ nhiều ngôn ngữ lập trình như Java, C#, Python.
* JUnit/TestNG: Công cụ kiểm thử đơn vị cho Java.
* pytest: Công cụ kiểm thử đơn vị cho Python.
* Appium: Công cụ kiểm thử tự động cho các ứng dụng di động.
* Jenkins: Công cụ tích hợp liên tục, thường được sử dụng để tự động hóa các kịch bản kiểm thử.

### Nguyên tắc kiểm thử phần mềm

Kiểm thử phần mềm đóng vai trò rất quan trọng :

* Kiểm thử phần mềm là hoạt động đảm bảo chất lượng phần mềm và mang tính sống còn trong các dự án sản xuất phần mềm. Vì vậy nó đã trở thành quy trình bắt buộc trong các dự án phần mềm hiện nay.
* Kiểm thử phần mềm để tránh những rủi ro, lỗi phát sinh trong suốt quá trình tạo ra sản phẩm.
* Lỗi càng phát hiện ra sớm càng giúp tránh được rủi ro và chi phí.

Mục đích của kiểm thử phần mềm:

* Kiểm thử phần mềm để đánh giá phần mềm có đạt yêu cầu mong đợi hay có sai sót nào không?
* Phần mềm có làm việc như mong muốn không?
* Phần mềm có giải quyết được yêu cầu của khách hàng không?Nó làm được gì mà người dùng mong đợi?
* Người dùng có thích nó không?
* Nó có tương thích với các hệ thống khác của chúng ta hay không?

Để kiểm thử đạt hiệu quả thì khi tiến hành kiểm thử phần mềm cần phải tuân thủ một số quy tắc sau:

* Quy tắc 1: Một phần quan trọng của 1 ca kiểm thử là định nghĩa của đầu ra hay kết quả mong muốn.
* Quy tắc 2: Lập trình viên nên tránh tự kiểm tra chương trình của mình.
* Quy tắc 3: Nhóm lập trình không nên kiểm thử chương trình của chính họ.
* Quy tắc 4: Kiểm tra thấu đáo mọi kết quả của mỗi kiểm tra.
* Quy tắc 5: Các ca kiểm thử phải được viết cho các trạng thái đầu vào không hợp lệ và không mong muốn, cũng như cho các đầu vào hợp lệ và mong muốn.
* Quy tắc 6: Khảo sát 1 chương trình để xem liệu chương trình có thực hiện cái mà nó cần thực hiện chỉ là 1 phần, phần còn lại là xem liệu chương trình có thực hiện cái mà nó không cần phải thực hiện hay không.
* Quy tắc 7: Tránh các các kiểm thử bâng quơ trừ khi chương trình thực sự là 1 chương trình bâng quơ.
* Quy tắc 8: Không dự kiến kết quả của kiểm thử theo giả thiết ngầm là không tìm thấy lỗi.
* Quy tắc 9: Xác suất tồn tại lỗi trong 1 đoạn chương trình là tương ứng với số lỗi đã tìm thấy trong đoạn đó.
* Quy tắc 10: Kiểm thử là 1 nhiệm vụ cực kỳ sáng tạo và có tính thử thách trí tuệ.

1. ĐẶC TẢ YÊU CẦU PHẦN MỀM (SRS)

## Giới thiệu về dự án

* Dự án quản lý nhân viên không chỉ là một hệ thống thông tin đơn thuần, mà là trái tim của mỗi tổ chức, một bộ não đầy sức mạnh để tạo ra sự phát triển và thành công bền vững. Trên nền tảng của môn Kiểm thử và Bảo đảm Chất lượng Phần mềm, dự án này không chỉ đảm bảo tính đúng đắn của thông tin mà còn là một công cụ linh hoạt và mạnh mẽ, giúp tối ưu hóa quản lý nhân sự từ những khía cạnh như tuyển dụng, đào tạo, phát triển, đánh giá và thăng tiến.
* Với sự kết hợp tinh tế giữa công nghệ và quản lý, dự án này không chỉ giúp tối ưu hóa hiệu suất lao động mà còn tạo ra một môi trường làm việc tích cực và tràn đầy năng lượng. Tích hợp các công cụ kiểm thử tiên tiến, dự án đảm bảo rằng mỗi chức năng, từ việc quản lý chấm công đến việc đánh giá hiệu suất, đều hoạt động một cách mượt mà và hiệu quả.
* Điều quan trọng hơn, dự án này không chỉ nhằm mục tiêu tối ưu hóa quy trình làm việc mà còn tạo ra sự kết nối giữa con người và công nghệ, tạo ra một cộng đồng nhân viên đoàn kết và phát triển. Bằng cách tận dụng sức mạnh của dữ liệu và phân tích thông minh, dự án giúp lãnh đạo hiểu rõ hơn về nhân viên của họ, từ đó đưa ra những quyết định chiến lược và phát triển bền vững cho tổ chức.
* Với tầm nhìn toàn diện và cam kết vững chắc với chất lượng, dự án quản lý nhân viên không chỉ là một phần mềm, mà là một đối tác đáng tin cậy và không thể thiếu trong hành trình phát triển của mọi tổ chức.

## Các yêu cầu chức năng

### Các tác nhân

- Hệ thống có tác nhân là Quản lý.

- Quản lý có thể đăng ký, đăng nhập và được cung cấp toàn bộ chức năng trong hệ thống.

### Các chức năng của hệ thống

1. Đăng nhập:

- Cho phép người quản lý đã đăng ký tài khoản truy cập vào hệ thống bằng cách nhập username và password(8-12 kí tự).

- Kiểm tra tính hợp lệ của thông tin đăng nhập và cấp quyền truy cập tương ứng.

1. Đăng ký:

Để truy cập sử dụng hệ thống thì người quản lý trước hết cần đăng ký tài khoản:

- Cho phép người quản lý mới tài khoản bằng cách cung cấp thông tin cá nhân như: Username, email, password, retype password.

- Kiểm tra tính hợp lệ của thông tin và lưu trữ tài khoản mới vào cơ sở dữ liệu.

1. Chức năng quản lý nhân viên:

\* Hiển thị danh sách nhân viên:

- Hiển thị danh sách các nhân viên có sẵn trong hệ thống, bao gồm thông tin chi tiết như: Mã nhân viên, Họ, Tên, Giới tính, Ngày sinh, Địa chỉ.

\* Thêm nhân viên:

- Cho phép người quản lý thêm các nhân viên mới vào hệ thống bằng cách cung cấp thông tin chi tiết về nhân viên.

\* Sửa nhân viên:

- Cho phép người quản lý sửa đổi thông tin của một nhân viên đã tồn tại.

\* Xóa nhân viên:

- Cho phép người quản lý xóa thông tin của một nhân viên đã tồn tại.

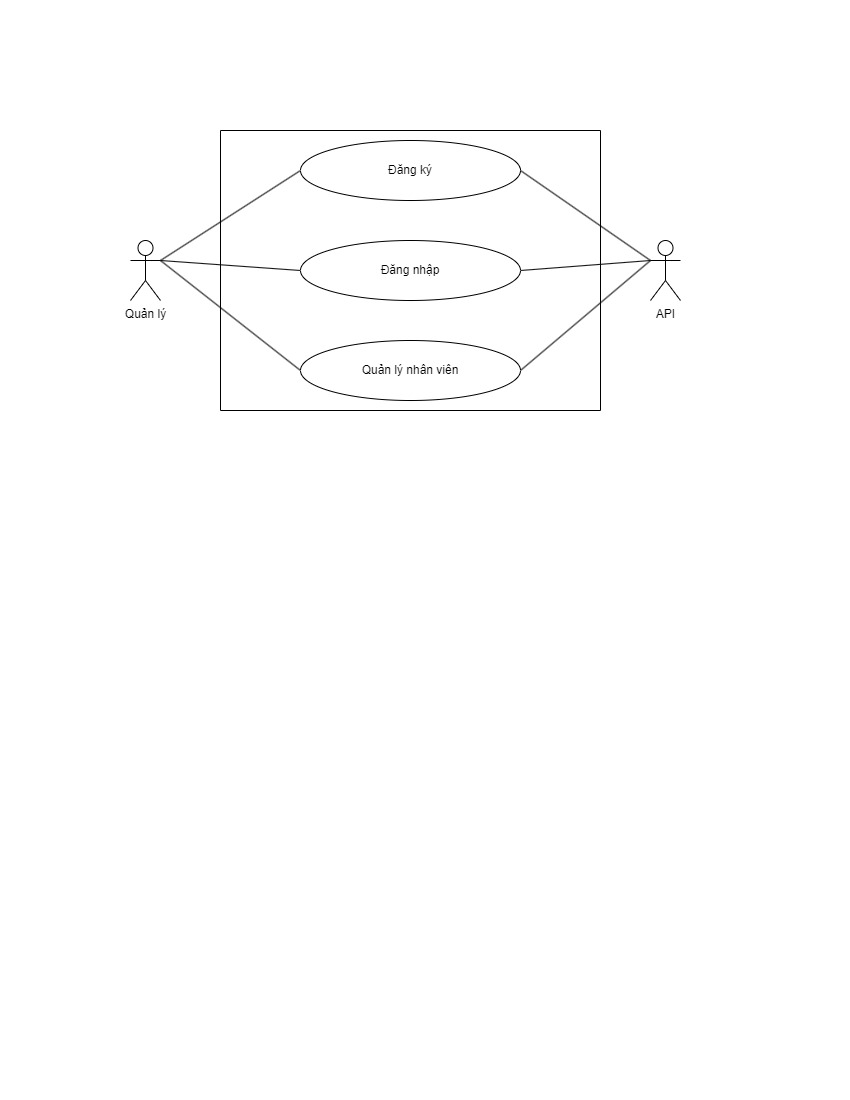
\* Lọc nhân viên:

- Cho phép người quản lý lọc nhân viên với thông tin tương ứng.

\* Tìm kiếm nhân viên:

- Cho phép người quản lý tìm kiếm nhân viên với thông tương ứng.

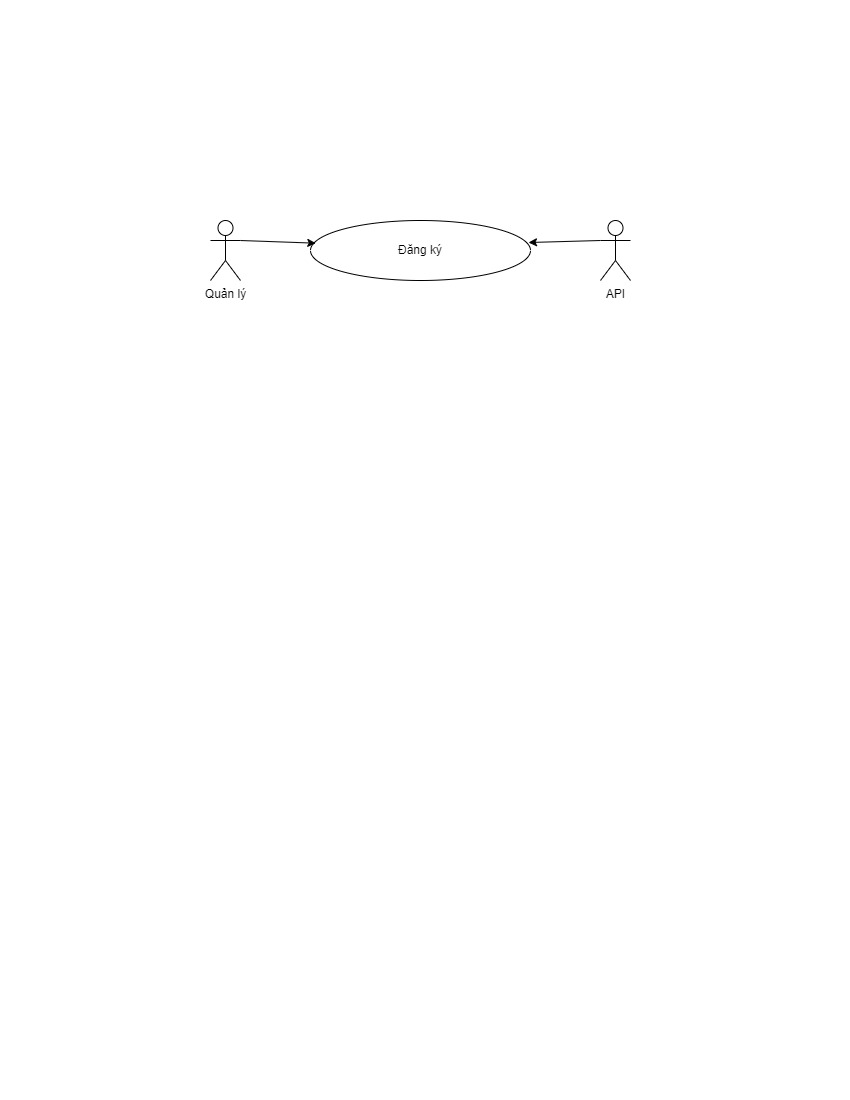
### Biểu đồ use case tổng quát.



Hình 2.1: Biểu đồ use case tổng quát

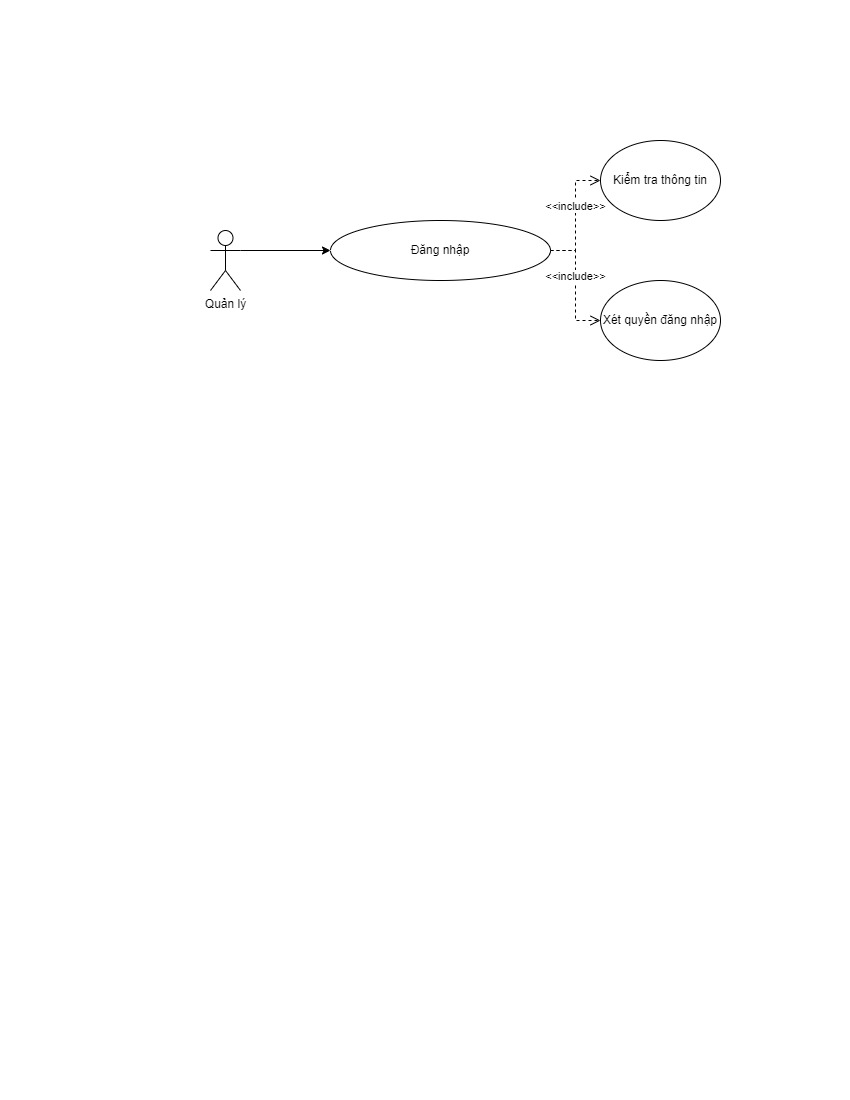
### Biểu đồ use case phân rã từng hoạt động.

#### Biểu đồ usecase Đăng ký



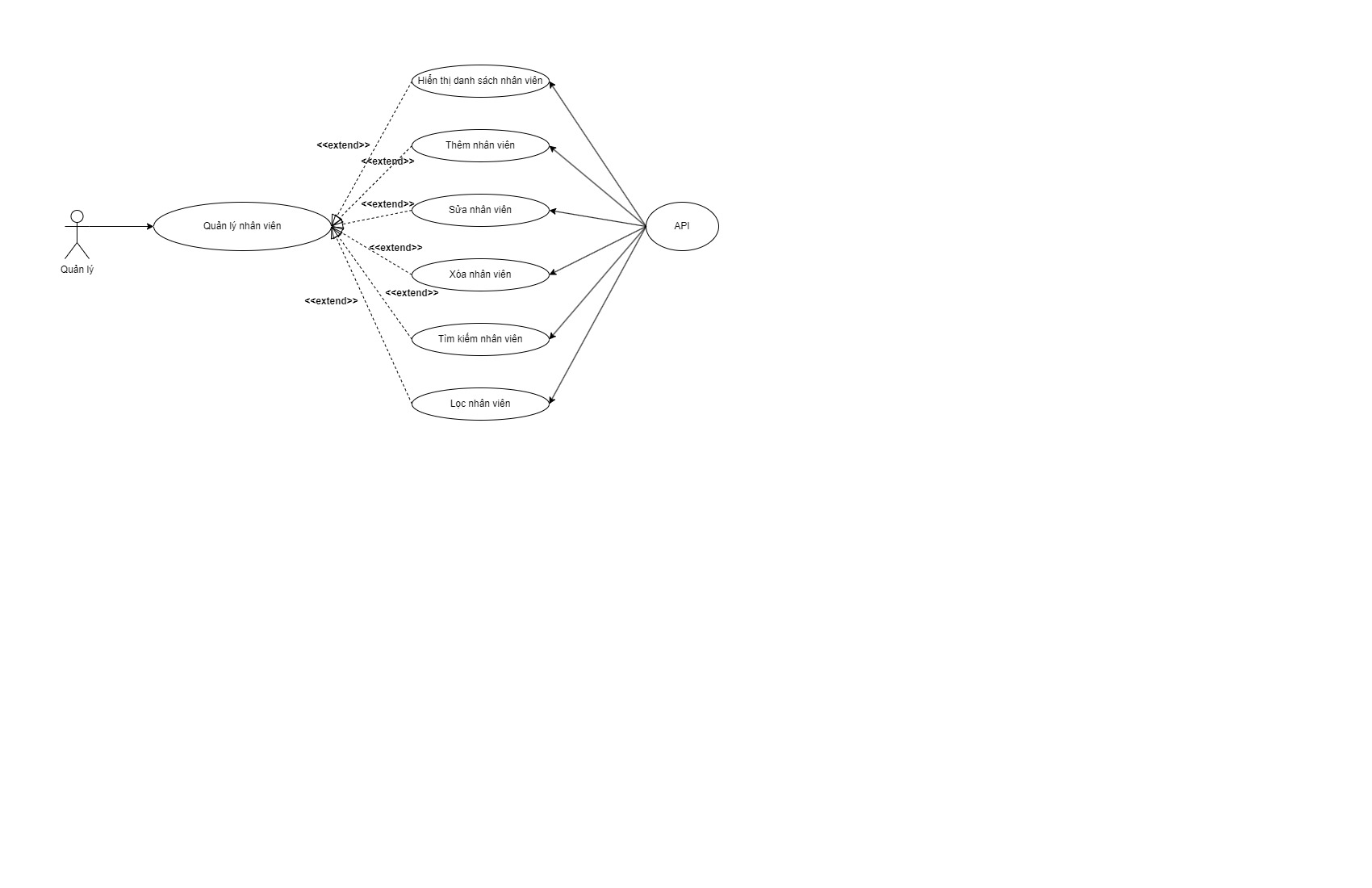
Hình 2.2: Biểu đồ use case Đăng ký

#### Biểu đồ usecase Đăng nhập



Hình 2.3: Biểu đồ use case Đăng nhập

#### Biểu đồ usecase Quản lý nhân viên



Hình 2.4: Biểu đồ use case Quản lý nhân viên

### Đặc tả các use case

#### Đăng nhập

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mã Use case | UC001 | | | | Tên Use case | Đăng nhập | |
| Tác nhân | Quản lý | | | | | | |
| Mô tả | Quản lý đăng nhập vào hệ thống để sử dụng các chức năng hệ thống | | | | | | |
| Sự kiện kích hoạt | Click vào nút đăng nhập trên giao diện website | | | | | | |
| Tiền điều kiện | Quản lý đã có tài khoản trên hệ thống | | | | | | |
| Luồng sự kiện chính (Thành công) |  | STT | Thực hiện bởi | Hành động | | |  |
| 1. | Quản lý | Chọn chức năng Đăng nhập | | |
| 2. | Hệ thống | Hiển thị giao diện đăng nhập | | |
| 3. | Quản lý | Nhập username và password(mô tả phía dưới \*) | | |
| 4. | Quản lý | Yêu cầu đăng nhập | | |
| 5. | Hệ thống | Kiểm tra xem quản lý đã nhập các trường bắt buộc nhập hay chưa | | |
| 6. | Hệ thống | Kiểm tra username và password có hợp lệ do quản lý nhập trong hệ thống hay không | | |
| 7. | Hệ thống | Hiển thị chức năng tương ứng đối với quản lý | | |
| Luồng sự kiện thay thế |  | STT | Thực hiện bởi | Hành động | | |  |
| 6a. | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu quản lý nhập thiếu | | |
| 7a. | Hệ thống | Thông báo lỗi: Username và/hoặc password chưa đúng nếu không tìm thấy username và password trong hệ thống | | |
| Hậu điều kiện | Quản lý đăng nhập được vào hệ thống | | | | | | |

Bảng 2.1: Đặc tả chức năng “Đăng nhập”

* + - * Dữ liệu đầu vào gồm các trường dữ liệu sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc? | Điều kiện hợp lệ | Ví dụ |
| 1. | Username | Input username field | Có | Username hợp lệ (không có ký tự đặc biệt, không có khoảng trắng, không được để trống, không quá 255 ký tự và chưa tồn tại trong hệ thống) | aloalo |
| 2. | Password | Password field | Có | Tối thiểu 8 ký tự và tối đa 12 ký tự | aloalo123 |

Bảng 2.2: Dữ liệu chức năng “Đăng nhập”

#### Đăng ký

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã Use case | UC002 | | | Tên Use case | Đăng ký | |
| Tác nhân | Quản lý | | | | | |
| Mô tả | Quản lý đăng ký tài khoản để sử dụng chức năng của hệ thống | | | | | |
| Sự kiện kích hoạt | Click vào nút đăng ký trên thanh tiêu đề | | | | | |
| Tiền điều kiện | Không | | | | | |
| Luồng sự kiện chính (Thành công) |  | STT | Thực hiện bởi | Hành động | |  |
| 1. | Quản lý | Chọn chức năng Đăng ký | |  |
| 2. | Hệ thống | Hiển thị giao diện đăng ký | |  |
| 3. | Quản lý | Nhập các thông tin tài khoản (mô tả phía dưới \*) | |  |
| 4. | Quản lý | Yêu cầu đăng ký | |  |
| 5. | Hệ thống | Kiểm tra xem quản lý đã nhập các trường bắt buộc nhập hay chưa | |  |
|  |  | 6 | Hệ thống | Kiểm tra xem Username nhập vào đã đúng yêu cầu hay chưa | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 7 | Hệ thống | Kiểm tra địa chỉ email của quản lý có hợp lệ không |  |
| 8. | Hệ thống | Kiểm tra mật khẩu nhập lại và mật khẩu có trùng nhau hay không |  |
| 9. | Hệ thống | Kiểm tra mật khẩu có đủ mức độ an toàn hay không |  |
| 10. | Hệ thống | Lưu thông tin tài khoản và thông báo đăng ký thành công |  |
| Luồng sự kiện thay thế |  | STT | Thực hiện bởi | Hành động |  |
| 6a. | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu quản lý nhập thiếu |  |
| 7a. | Hệ thống | Thông báo lỗi: User name nhập vào chưa đúng yêu cầu |  |
| 8a. | Hệ thống | Thông báo lỗi: Địa chỉ email không hợp lệ nếu địa chỉ email không hợp lệ |  |
| 9a. | Hệ thống | Thông báo lỗi: Mật khẩu xác nhận không khớp nếu hai mật khẩu không trùng nhau |  |
| 10a. | Hệ thống | Thông báo lỗi: Mật khẩu chưa đủ an toàn nếu mật khẩu nhập vào không đảm bảo yêu cầu của hệ thống (tối thiểu 8 ký tự và tối đa 12 ký tự) |  |
| Hậu điều kiện | Tài khoản được tạo và lưu trữ vào hệ thống | | | | |

Bảng 2.3: Đặc tả chức năng “Đăng ký”

* + - * Dữ liệu đầu vào của chức năng Đăng ký gồm các trường dữ liệu sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc? | Điều kiện hợp lệ | Ví dụ |
| 1. | Username | Input username field | Có | Username hợp lệ (không có ký tự đặc biệt, không có khoảng trắng, không được để trống, không quá 255 ký tự và chưa tồn tại trong hệ thống) | aloalo |
| 1. | Email | Input email field | Có | Địa chỉ email hợp lệ | aloalo123[@gmail.com](mailto:qndev@gmail.com) |
| 2. | Password | Password field | Có | Tối thiểu 8 ký tự, tối đa 12 kí tự | aloalo123 |
| 3. | Retype Password | Password field | Có | Trùng với Mật khẩu | aloalo123 |

Bảng 2.4: Dữ liệu chức năng “Đăng ký”

#### Quản lý nhân viên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mã Use case | | UC003 | | Tên use case | Quản lý nhân viên |  | |  |
| Tác nhân | | Quản lý | | | |  | |  |
| Mô tả | | Thực hiện các nhiệm vụ như: Hiển thị danh sách nhân viên, Thêm sửa xóa nhân viên, Lọc nhân viên, Tìm kiếm nhân viên. | | | |  | |  |
| Sự kiện kích hoạt | | Click nút "Thêm", "Sửa", "Xóa", "Lọc", "Tìm kiếm" tương ứng với thêm, sửa, xóa, lọc tìm kiếm thông tin nhân viên. | | | |  | |  |
| Tiền điều kiện | | Đăng nhập thành công | | | |  | |  |
|  | Thêm : | | | | |  | |  |
|  | |  |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động | |  | |  |
| 1 | Quản lý | Yêu cầu chức năng thêm nhân viên | |  | |  |
| 2 | Hệ thống | Hiển thị chức năng thêm nhân viên | |  | |  |
| 3 | Quản lý | Nhập thông tin nhân viên (\*mô tả phía dưới) và yêu cầu thêm mới | |  | |  |
| 4 | Hệ thống | Kiểm tra các trường nhập dữ liệu | |  | |  |
| 5 | Hệ thống | Thêm mới thành công | |  | |  |
| Luồng sự kiện thay thế | 4.a | Hệ thống | Thông báo lỗi nếu dữ liệu nhập vào không đúng định dạng | |  | |  |
| 5.a | Hệ thống | Thông báo lỗi nếu thêm mới không thành công | |  | |  |
| Sửa : | | | | |  | |  |
|  | |  |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động | |  | |  |
| 1 | Quản lý | Chọn nhân viên cần sửa và yêu cầu sửa | |  | |  |
| 2 | Hệ thống | Lấy thông tin nhân viên và hiển thị lên chức năng sửa | |  | |  |
| 3 | Quản lý | Chỉnh sửa các thông tin nhân viên(\*mô tả phía dưới) và yêu cầu sửa | |  | |  |
| 4 | Hệ thống | Kiểm tra các trường nhập dữ liệu | |  | |  |
| 5 | Hệ thống | Cập nhật thông tin và thông báo thành công | |  | |  |
| Luồng sự kiện thay thế | 4.a | Hệ thống | Thông báo lỗi nếu dữ liệu nhập vào không đúng định dạng | |  | |  |
| 5.a | Hệ thống | Thông báo lỗi nếu cập nhật không thành công | |  | |  |
| Xóa : | | | | |  | |  |
|  | |  |
| Luồng sự kiện chính | STT | Thực hiện bởi | Hành động | |  | |  |
| 1 | Quản lý | Chọn nhân viên cần xóa và yêu cầu xóa | |  | |  |
| 2 | Hệ thống | Hiển thị thông báo yêu cầu quản lý xác nhận xóa | |  | |  |
| 3 | Quản lý | Xác nhận xóa | |  | |  |
| 4 | Hệ thống | Xóa và thông báo thành công | |  | |  |
| Luồng sự kiện thay thế | 4.a | Hệ thống | Thông báo lỗi nếu xóa không thành công | |  | |  |
|  | Lọc: |  |  |  | |  | |  |
|  | Luồng sự kiện chính | STT | Thực hiện bởi | Hành động | |  | |  |
|  |  | 1 | Quản lý | Click vào ô lọc và nhập thông tin giới tính rồi click button lọc | |  | |  |
|  |  | 2 | Hệ thống | Lấy thông tin nhân viên có giới tính nam hoặc nữ gửi lại theo yêu cầu của quản lý | |  | |  |
|  |  | 3 | Hệ thống | Check trong cơ sở dữ liệu và hiển thị danh sách nhân viên theo yêu cầu lọc của quản lý | |  | |  |
|  | Luồng sự kiện thay thế | 3.a | Hệ thống | Thông báo lỗi nếu xóa nhập sai giới tính nhân viên không có sẵn | |  | |  |
|  | Tìm kiếm: |  |  |  | |  | |  |
|  | Luồng sự kiện chính | STT | Thực hiện bởi | Hành động | |  | |  |
|  |  | 1 | Quản lý | Nhập thông tin nhân viên cần tìm kiếm | |  | |  |
|  |  | 2 | Hệ thống | Xem trong cơ sở dữ liệu và đưa ra thông tin nhân viên tương ứng | |  | |  |
|  | Luồng sự kiện thay thế | 2.a | Hệ thống | Thông báo lỗi nếu nhập thông tin sai hoặc để trống | |  | |  |
| Hậu điều kiện | | Thêm thành nhân viên và được lưu trữ vào cơ sở dữ liệu; Cập nhật thành công thông tin mới sẽ được lưu vào cơ sở dữ liệu; Xóa thành công nhân viên; Lọc được nhân viên và Tìm kiếm được nhân viên. | | | |  |  |  |

Bảng 2.5: Đặc tả chức năng “Quản lý nhân viên”

## Các yêu cầu phi chức năng

1. Bảo mật:

- Yêu cầu hệ thống áp dụng các biện pháp bảo mật mạnh mẽ như mã hóa dữ liệu, xác thực hai yếu tố và kiểm tra phân quyền để đảm bảo rằng chỉ người dùng có quyền mới có thể truy cập thông tin nhân viên.

- Hệ thống phải lưu trữ mật khẩu của người dùng dưới dạng băm để đảm bảo an toàn thông tin đăng nhập.

2. Hiệu suất:

- Yêu cầu hệ thống có thời gian phản hồi dưới 1 giây cho các yêu cầu thông thường như hiển thị danh sách nhân viên.

- Hệ thống phải được thử nghiệm hiệu suất để đảm bảo khả năng xử lý hàng nghìn hoặc hàng trăm nghìn bản ghi nhân viên mà không gây ra độ trễ đáng kể.

3. Khả năng mở rộng:

- Yêu cầu cấu trúc của dữ liệu phải linh hoạt và dễ mở rộng để hỗ trợ việc thêm các chức năng mới như quản lý phòng ban, quản lý dự án, hoặc tính năng báo cáo.

- Hệ thống phải được thiết kế theo kiến trúc lớp để cho phép phát triển mở rộng dễ dàng và không gây ra tình trạng phức tạp.

4. Tương thích:

- Yêu cầu giao diện người dùng phải tương thích trên các trình duyệt phổ biến như Google Chrome, Firefox, và Microsoft Edge trên các nền tảng Windows, macOS và Linux.

- Hệ thống phải được thiết kế sử dụng các công nghệ web tiêu chuẩn để đảm bảo tính tương thích trên nhiều thiết bị và màn hình khác nhau.

5. Sẵn sàng và độ tin cậy:

- Yêu cầu hệ thống phải có sẵn để sử dụng 24/7 và có khả năng phục hồi nhanh chóng trong trường hợp có sự cố.

- Hệ thống phải được sao lưu định kỳ và có kế hoạch dự phòng để đảm bảo rằng không mất dữ liệu quan trọng.

6. Dễ sử dụng:

- Yêu cầu giao diện người dùng phải được thiết kế đơn giản và trực quan để người dùng có thể sử dụng một cách dễ dàng mà không cần đào sâu vào tài liệu hướng dẫn.

- Hệ thống phải cung cấp các hướng dẫn sử dụng và gợi ý dễ hiểu khi người dùng thực hiện các tác vụ.

7. Tính di động:

- Yêu cầu hệ thống phải có khả năng hoạt động trên các thiết bị di động như điện thoại thông minh và máy tính bảng.

- Giao diện người dùng trên thiết bị di động phải được tối ưu hóa để đảm bảo trải nghiệm người dùng tốt nhất trên các kích thước màn hình nhỏ.

8. Bảo trì và hỗ trợ:

- Yêu cầu hệ thống phải có khả năng cập nhật dễ dàng để triển khai các bản vá lỗi và cải tiến mới.

- Hệ thống phải cung cấp tài liệu hướng dẫn chi tiết và hỗ trợ kỹ thuật cho người quản trị hệ thống và người dùng cuối.

1. KẾ HOẠCH TEST

## Quy trình thực hiện test

### Thiết kế test plan cho dự án

#### Bảng kế hoạch kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| *Nhiệm vụ* | *Thời gian* | *Ngày bắt đầu* | *Ngày kết thúc* |
| *Tìm kiếm và nghiên cứu đề tài* | *4 ngày* | *2/02/2024* | *05/02/2024* |
| *Lập kế hoạch kiểm thử* | *2 ngày* | *04/02/2024* | *05/02/2024* |
| *Phân tích thiết kế hệ thống* | *7 ngày* | *06/02/2024* | *12/02/2024* |
| *Nghiên cứu tìm hiểu và cài đặt tool test phù hợp* | *2 ngày* | *13/02/2024* | *14/02/2024* |
| *Thiết kế Test case* | *21 ngày* | *15/02/2024* | *05/03/2024* |
| *Thực thi Test case* | *14 ngày* | *06/03/2024* | *13/03/2024* |
| *Kiểm thử tự động* | *21 ngày* | *14/03/2024* | *04/04/2024* |
| *Kiểm tra và sửa lỗi* | *10 ngày* | *05/04/2024* | *14/04/2024* |
| *Báo Cáo Kiểm Thử* | *1 ngày* | *15/04/2024* | *16/04/2024* |

#### Kế hoạch kiểm thử thủ công (kiểm thử bằng tay)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Công việc | Tổng thời gian | Thời gian bắt đầu | Thời gian kết thúc | Người thực hiện |
| Thiết kế test case | 11 ngày | 08/03/2024 | 18/03/2024 | Nguyễn Văn Hải |
| 1.Đăng ký | 2 Ngày | 08/03/2024 | 09/03/2024 | Nguyễn Văn Hải |
| 2.Đăng nhập | 2 Ngày | 10/03/2024 | 11/03/2024 | Nguyễn Văn Hải |
| 3.Thêm nhân viên | 3 Ngày | 12/03/2024 | 14/03/2024 | Nguyễn Văn Hải |
| 4.Sửa nhân viên | 2 Ngày | 15/03/2024 | 16/03/2024 | Nguyễn Văn Hải |
| 5.Xóa nhân viên | 1 Ngày | 17/03/2024 | 17/03/2024 | Nguyễn Văn Hải |
| 6.Tìm kiếm ,lọc nhân viên | 1 Ngày | 18/03/2024 | 18/03/2024 | Nguyễn Văn Hải |

#### Kế hoạch kiểm thử tự động

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Công việc | Tổng thời gian | Thời gian bắt đầu | Thời gian kết thúc | Người thực hiện |
| Kiểm thử tự động | **11 ngày** | 21/03/2024 | 18/03/2024 | Nguyễn Văn Hải |
| Đăng ký | 3 Ngày | 21/03/2024 | 23/03/2024 | Nguyễn Văn Hải |
| Đăng nhập | 2 Ngày | 24/03/2024 | 25/03/2024 | Nguyễn Văn Hải |
| Thêm nhân viên | 2 Ngày | 26/03/2024 | 27/03/2024 | Nguyễn Văn Hải |
| Sửa nhân viên | 2 Ngày | 28/03/2024 | 29/03/2024 | Nguyễn Văn Hải |
| Xóa nhân viên | 1 Ngày | 30/03/2024 | 30/03/2024 | Nguyễn Văn Hải |
| Tìm kiếm,lọc | 1 Ngày | 31/03/2024 | 31/03/2024 | Nguyễn Văn Hải |

### Test enviroment

*Là môi trường về phần cứng, phần mềm, cơ sở hạ tầng,… phục vụ cho việc test trong dự án:*

* + - * PC Windows (Version mới nhất)

### Test Items/ Features to be Tested

*Điền các chức năng sẽ thực hiện trong phase hiện tại (tốt nhất là theo thứ tự ưu tiên)*

**Front End**

Thiết kế giao diện:

- Xây dựng giao diện người dùng dựa trên thiết kế đã được xác định trước.

- Đảm bảo giao diện thân thiện, dễ sử dụng và tương thích với các thiết bị và trình duyệt khác nhau.

Đăng ký:

- Hiển thị giao diện đăng ký cho người dùng.

- Xác minh tính hợp lệ của thông tin đăng ký nhập từ người dùng.

- Xử lý việc lưu thông tin đăng ký mới vào cơ sở dữ liệu.

Đăng nhập:

- Hiển thị giao diện đăng nhập cho người dùng.

- Xác thực thông tin đăng nhập và cấp quyền truy cập tương ứng.

- Điều hướng người dùng sau khi đăng nhập thành công.

Hiển thị danh sách nhân viên:

- Hiển thị danh sách nhân viên có sẵn trong hệ thống cho người dùng.

Thêm nhân viên mới:

- Cung cấp giao diện và chức năng để thêm mới một nhân viên vào hệ thống.

- Xác nhận và lưu trữ thông tin của nhân viên mới vào cơ sở dữ liệu.

Sửa thông tin bài viết:

- Cho phép người quản lý chỉnh sửa thông tin của một nhân viên đã tồn tại trong hệ thống.

- Xác nhận và cập nhật thông tin đã được chỉnh sửa vào cơ sở dữ liệu.

Xóa nhân viên:

- Xác nhận và xử lý việc xóa một bài viết khỏi hệ thống.

- Cập nhật cơ sở dữ liệu để loại bỏ bài viết đã được xóa.

**Backend:**

Code:

- Phát triển mã backend để xử lý các yêu cầu từ phía người dùng và tương tác với cơ sở dữ liệu.

- Đảm bảo tính hợp lệ và an toàn của dữ liệu được trao đổi giữa frontend và backend.

Xác thực người dùng:

- Triển khai cơ chế xác thực để đảm bảo chỉ người dùng có quyền mới có thể thực hiện các thao tác quản lý bài viết.

Quản lý cơ sở dữ liệu:

- Thiết kế và triển khai cơ sở dữ liệu để lưu trữ thông tin về các bài viết và người dùng.

- Đảm bảo tính toàn vẹn và an toàn của dữ liệu

### Features Not to Be Tested:

Danh sách các tính năng/ sản phẩm không cần test:

Hiển thị nhân viên

Lý do không test có thể là:

Tính năng tự động hiển thị không được thực hiện bởi người dùng

### Test strategy

Liệt kê các kiểu test, level test

Có thể thêm các chiến lược test (như phân vùng tương đương, phân tích giá trị biên, bảng quyết định….)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test type** | **Test level** | | | |
| **Unit** | **Integration** | **System** | **Acceptance** |
| **GUI** |  | X |  |  |
| **Function** |  | X | X |  |
| **Regression** |  | X | X |  |
| **Performance test** |  |  | X |  |
| **Load test** |  |  | X |  |
| **Stress test** |  |  | X |  |

X: loại test/level test cần thực hiện trong dự án

### Milestones

Điền thông tin các mốc bàn giao, sản phẩm bàn giao ở các mốc vào bảng dưới

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Milestones | Tài liệu cần bàn giao | Người chịu TN | Ngày bàn giao | Ghi chú |
| 1 | 1 | Test plan | Lê Đức Anh,Hoàng Văn Quý | 26/01/2024 |  |
| 2 | 1 | File quan điểm test | Lê Đức Anh,Hoàng Văn Quý | 14/02/2024 |  |
| 3 | 1 | Test case | Nguyễn Văn Hải,Đinh Tuấn Duy | 05/03/2024 |  |
| 4 | 1 | Test result | Nguyễn Văn Hải,Đinh Tuấn Duy | 13/03/2024 |  |
| 5 | 1 | Test report | Nguyễn Văn Hải,Đinh Tuấn Duy | 16/04/2024 |  |
| 6 | 1 | Test evidence | Nguyễn Văn Hải,Đinh Tuấn Duy | 22/04/2024 |  |
| 7 | 1 | Interface design and programming | Nhữ Công Việt Hoàng | 18/03/2024 |  |

### Công số và schedule

Tham khảo sheet "Schedule"

### Thành viên tham gia

Điền thông tin danh sách nhân sự tester tham gia, vai trò, trách nhiệm trong dự án

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***No.*** | ***Họ và tên*** | ***Vai trò*** | ***Tỷ lệ*** | ***Ngày bắt đầu*** | ***Ngày kết thúc*** | ***Nhiệm vụ*** |
| *1* | *Nhữ Công Việt Hoàng* | *Leader* | *100%* | *02/01/2024* | *02/05/2024* | *Dev, Phân tích* |
| *2* | *Lê Đức Anh* | *Member* | *100%* | *02/01/2024* | *02/05/2024* | *Phân tích, Test* |
| *3* | *Nguyễn Văn Hải* | *Member* | *100%* | *02/01/2024* | *02/05/2024* | *Phân tích, Test* |
| *4* | *Đinh Tuấn Duy* | *Member* | *100%* | *02/01/2024* | *02/05/2024* | *Phân tích, Test* |
| *5* | *Hoàng Văn Quý* | *Member* | *100%* | *02/01/2024* | *02/05/2024* | *Phân tích, Test* |

### Tiêu chuẩn kết thúc

Tất cả Testcases Đều Đã Pass:

- Tất cả các testcases đã được thiết kế và triển khai đã đạt kết quả Pass, không có trường hợp nào bị Fail.

Yêu Cầu của Khách Hàng Đã Được Đáp Ứng:

- Các chức năng và tính năng của trang web quản lý bài viết đã được kiểm tra và đảm bảo rằng chúng đáp ứng đúng các yêu cầu và mong muốn của khách hàng.

Không Còn Lỗi Có Độ Nghiêm Trọng Cao:

- Tất cả các lỗi có độ nghiêm trọng cao đã được xử lý và kiểm tra lại để đảm bảo rằng không còn tồn tại trong sản phẩm cuối cùng.

Hết Chi Phí Kiểm Thử:

- Tất cả các hoạt động kiểm thử đã hoàn thành và không còn yêu cầu chi phí bổ sung cho việc kiểm thử dự án.

Đạt Đến Lịch Phát Hành:

- Quá trình kiểm thử đã hoàn thành đúng theo kế hoạch và đảm bảo rằng sản phẩm có thể được phát hành theo đúng lịch trình đã xác định trước.

### Tiêu chí tạm ngừng và tiêu chí bắt đầu lại(sau khi tạm ngừng)

**Tiêu chí tạm ngừng**

Chức năng/Màn hình Có Nhiều Lỗi Nghiêm Trọng:

- Nếu một chức năng hoặc màn hình của trang web quản lý bài viết gặp phải nhiều lỗi nghiêm trọng và không thể tiếp tục kiểm thử một cách hiệu quả.

Vấn Đề Về Phần Cứng/Phần Mềm:

- Trong trường hợp có vấn đề liên quan đến phần cứng hoặc phần mềm của hệ thống kiểm thử, gây ảnh hưởng đến quá trình kiểm thử của dự án.

Resource Không Sẵn Có:

- Khi không có đủ tài nguyên như con người, thiết bị hoặc công cụ cần thiết để tiếp tục thực hiện các hoạt động kiểm thử một cách hiệu quả.

**Tiêu chí bắt đầu lại(sau khi tạm ngừng)**

Các Tiêu Chí Tạm Ngừng Đã Được Giải Quyết:

- Tất cả các vấn đề và tiêu chí tạm ngừng đã được xử lý và giải quyết để đảm bảo rằng quá trình kiểm thử có thể tiếp tục một cách suôn sẻ và hiệu quả.

Resource Đã Sẵn Có:

- Các tài nguyên cần thiết như nhân lực, thiết bị và công cụ đã được chuẩn bị và sẵn sàng để tiếp tục quá trình kiểm thử một cách chất lượng và hiệu quả.

### Risks:

*Liệt kê các rủi ro có thể có ảnh hưởng tới quá trình test*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên rủi ro | Tần suất | Mức độ ảnh hưởng | Phương án ứng phó |
| 1 | Tài liệu không đầy đủ, rõ ràng | 80% | Cao | Q&A, yêu cầu giai đoạn tìm hiểu phải có output |
| 2 | Chức năng khó | 10% | Cao | Nhờ người có kinh nghiệm, tham khảo dự án khác |
| 3 | Nhân sự yếu | 30% | Thấp | Training |
| 4 | Nhân sự nghỉ | 20% | Thấp | Phương án thay thế |
| 5 | Thời gian phản hồiQ&A chậm | 80% | Cao | Đưa ra rule về thời gian phản hồi Q&A |

### Công cụ

*Liệt kê công cụ sử dụng trong quá trình test dự án*

|  |  |  |
| --- | --- | --- |
| No | Process | Tool |
| 1 | Requirement study | Xmind, mindmap |
| 2 | Test case management | Excel |
| 3 | Test case execute | Manual, Jmeter |
| 4 | Evidences capture | snipping tool |
| 5 | Defect tracking | Jira |
| 6 | Test report | Excel |

1. TRIỂN KHAI KIỂM THỬ

## Mô tả giao diện

### Đăng nhập

A screenshot of a computer

Description automatically generated

**Hình 3. 1 From đăng nhập**

Sau khi người dùng đăng ký xong sẽ nhập email và mật khẩu vừa đăng ký điền vào ô và sau đó bấm button đăng nhập.

Sau khi click button đăng nhập sẽ đưa người dùng về trang home với trạng thái đã đăng nhập.

Xác định phân vùng tương đương theo yêu cầu sau đây:

- Email:

+ Không được để trống và đinh dạng [abv123@gmail.com](mailto:abv123@gmail.com)

+ Không có khoảng trắng (space)

- Mật khẩu:

+ Có độ dài ít nhất là 6 ký tự số hoặc chữ thường (không dấu).

+ Không được để trống

+ Phải có kí tự đặc biệt

+ Ít nhất có từ viết hoa

- Button đăng nhập:

+ Kiểm tra và thông báo khi chưa nhập thông tin gì mà nhấn đăng nhập

+ Email hoặc mật khẩu không được để trống

+ Không được để trống một trong 2 trường thông tin

+ Nhập thông tin email và mật khẩu đúng yêu cầu sẽ chuyển trang qua màn hình chính của trang web

#### Email

|  |  |  |
| --- | --- | --- |
| Test case | Valid input | Invalid input |
| Email | Nhập đúng email đã đăng kí | Nhập email không tồn tại |

Bảng 3. 1 Test case Email kiểm thử hộp đen

* Case 1: Đăng nhập thành công khi đúng định dạng của gmail [abc12@gamil.com](mailto:abc12@gamil.com)
* Case 2: Đăng nhập không thành công do email không phù hợp với định dạng-> Hiện lên thông báo : The Email field is not a valid e-mail address.
* Case 3: Đăng nhập không thành công do email không hợp lệ vì có khoảng trắng (space)-> Hiện lên thông báo: The Email field is not a valid e-mail address.
* Case 4: Đăng nhập không thành công do email không hợp lệ vì có không đúng định dạng- > Hiện lên thông báo: The Email field is not a valid e-mail address

Case 5: Đăng nhập không thành công do Email để trống-> Hiện lên thông báo: The Email field is not a valid e-mail address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | 1 | 2 | 3 | 4 |
| Kết quả | Đúng yêu cầu | Thất bại | Thất bại | Thất bại |
| Lớp tương  đương | O | X | X | X |
| Đánh dấu | L1 | L2 | L3 | L4 |

Bảng 3. 2 Bảng phân vùng Email

#### Chú thích:

* X: không hợp lệ
* O: hợp lệ

Phân vùng tương đương hợp lệ: {L1}

Phân vùng tương đương không hợp lệ : {L2,L3,L4,L5}

#### Giá trị biên:

|  |  |  |
| --- | --- | --- |
| Test Case | Mô tả | Kết quả |
| 1 | Đúng định dạng của gmail [abc12@gamil.com](mailto:abc12@gamil.com) | Thành công |
| 2 | Email không phù hợp với  định dạng | The Email field is not  a valid e-mail address |
| 3 | Email có khoảng trắng | The Email field is not a valid e-mail address |
| 4 | Email không hợp lệ vì có  <6 | The Email field is not  a valid e-mail address |
| 5 | Email để trống | The Email field is not a valid e-mail address |

Bảng 3. 3 Giá trị biên Email

#### Mật khẩu:

|  |  |  |
| --- | --- | --- |
| Test case | Valid input | Invalid input |
|  | Nhập ít nhất 6 kí tự | Nhập bé hơn 6 kí tự |

57

|  |  |  |
| --- | --- | --- |
| Mật khẩu |  |  |
|  | Có ít nhất chứ viết hoa | Không có chữ viết hoa |
|  | Có kí tự đặc biệt | Không có kí tự đặc biệt |
|  | Có số | Không có số |

Bảng 3. 4 Test case Mật khẩu

Case 1: đăng nhập thành công khi mật khẩu hợp lí khi có ít nhất 6 kí tự và không để trống

-> chuyển đến trang chủ web.

Case 2: đăng nhập không thành công khi mật khẩu bé hơn 6 kí tự-> Hiện lên thông báo: Invalid login attempt

Case 3: đăng nhập không thành công khi mật khẩu để trống-> Hiện lên thông báo: The password field is required

Case 4: đăng nhập không thành công khi không có kí tự viết hoa-> Hiện lên thông báo: Invalid login attempt

Case 5: đăng nhập không thành công khi mật khẩu không có kí tự đặc biệt-> Hiện lên thông báo: Invalid login attempt

Case 6: đăng nhập không thành công khi mật khẩu không có chữ số-> Hiện lên thông báo: Invalid login attempt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case | 1 | 2 | 3 | 4 | 5 | 6 |
| Kết quả | Đúng  yêu cầu | Nhập sai | Nhập sai | Nhập sai | Nhập sai | Nhập sai |
| Lớp tương đương | O | X | X | X | X | X |
| Đánh dấu | L1 | L2 | L3 | L4 | L5 | L6 |

Bảng 3. 5 Phân vùng Mật khẩu

#### Chú thích:

* X: không hợp lệ
* O: hợp lệ

Phân vùng tương đương hợp lệ: {L1}

Phân vùng tương đương không hợp lệ : {L2,L3,L4,L5,L6}

#### Giá trị biên:

|  |  |  |
| --- | --- | --- |
| Test Case | Mô tả | Kết quả |
| 1 | Điền ít nhất 6 kí tự và  không bõ trống | Thành công |
| 2 | Mật khẩu có kí tư <6 | Invalid login attempt |
| 3 | Để trống | The password field is required |
| 4 | Không có kí tự viết hoa | Invalid login attempt |
| 5 | Không có kí tự đặc biệt | Invalid login attempt |
| 6 | Không có chữ số | Invalid login attempt |

Bảng 3. 6 Giá trị biên Mật khẩu

#### Nút đăng nhập:

Case 1: Email hoặc mật khẩu để trống -> hiện thông báo bắt buộc điền đầy đủ thông tin 2 trường đó

Case 2: Độ dài và mật khẩu <6 -> hiện thông báo Invalid login attempt

Case 3: mật khẩu không có kí tự đặc biệt -> hiện thông báo Invalid login attempt Case 4: mật khẩu không có kí tự viết hoa -> hiện thông báo Invalid login attempt Case 5: mật khẩu không có kí tự số-> hiện thông báo Invalid login attempt

Case 6: Email không đúng định dạng -> hiện thông báo tên The Email field is not a valid e-mail address

Case 7: Nhập cả email và mật khẩu hợp lệ thì bấm button đăng nhập sẽ chuyển sang trang chính của User

Case 8: Login với FaceBook Case 9: Login với gmail

Mô tả chức năng đăng kýA screenshot of a computer

Description automatically generated

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Điều kiện | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 |
| Email | T | T | T | F | T | F | F |
| Tên  người dùng | T | F | T | T | T | T | F |
| Mật  khẩu | T | T | F | T | T | T | F |
| Nhập lại mật khẩu | T | T | F | T | F | F | F |
| Hành  động | H | E | E | E | E1 | E | E |

Bảng 3. 7 Test case Đăng Ký

## Kiểm thử hộp đen

### Phân vùng tương đương và biên

#### Chức năng đăng ký

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Tài khoản* | *[a-zA-Z0-9]* | *1* | *Để trống,Tài khoản đã tồn tại* | *2* |
| *Email* | *Đúng định dạng email* | *3* | *Để trống,Sai định dạng* | *4* |
| *Mật khẩu* | *[a-zA-Z0-9]*  {8,12} | *5* | *Để trống,<8 ký tự,>12 ký tự* | *6* |
| *Xác nhận mật khẩu* | *Trùng mật khẩu* | *7* | *không trùng mật khẩu* | *8* |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Tài khoản=””*  *Email=””*  *Mật khẩu =””*  *Xác nhận mật khẩu=””* | *Các trường không được trống* | *2,4,6,8* | *Pass* |
| *2* | *Tài khoản=”user”*  *Email=”user@”*  *Mật khẩu =”12345678”*  *Xác nhận mật khẩu=”12345678”* | *Email không đúng định dạng* | *1,3,4,5* | *Pass* |
| *3* | *Tài khoản=”user”*  *Email=”user@gmail.com”*  *Mật khẩu =”123”*  *Xác nhận mật khẩu=”123”* | *Mật khẩu phải từ 8 ký tự-12 ký tự* | *1,3,6,7* | *Pass* |
| *4* | *Tài khoản=”user”*  *Email=”user@gmail.com”*  *Mật khẩu =”1234567890123”*  *Xác nhận mật khẩu=”1234567890123”* | *Mật khẩu phải từ 8 ký tự-12 ký tự* | *1,3,6,7* | *Pass* |
| *5* | *Tài khoản=”user”*  *Email=”user@gmail.com”*  *Mật khẩu =”123456783”*  *Xác nhận mật khẩu=”1234567”* | *Mật khẩu và xác nhận mật khẩu không khớp.* | *1,3,5,8* | *Pass* |
| *6* | *Tài khoản=”user”*  *Email=”user@gmail.com”*  *Mật khẩu =”123456783”*  *Xác nhận mật khẩu=”12345678”* | *Đăng nhập thành công* | *1,3,5,7* | *Pass* |

#### Chức năng đăng nhập

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Tài khoản* | *[a-zA-Z0-9]*  *,Có trong cơ sở dữ liệu* | *1* | *Để trống,*  *Không có trong cơ sở dữ liệu* | *2* |
| *Mật khẩu* | *[a-zA-Z0-9]*  {8,12}, *Có trong cơ sở dữ liệu* | *3* | *Để trống,<8 ký tự,>12 ký tự, Không có trong cơ sở dữ liệu* | *4* |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Tài khoản=””*  *Mật khẩu =””* | *Các trường không được trống* | *2,4* | *Pass* |
| *2* | *Tài khoản=”user”*  *Mật khẩu =”123”* | *Mật khẩu phải từ 8 ký tự-12 ký tự* | *1,4* | *Pass* |
| *4* | *Tài khoản=”user”*  *Mật khẩu =”1234567890123”* | *Mật khẩu phải từ 8 ký tự-12 ký tự* | *1,4* | *Pass* |
| *5* | *Tài khoản=”user”*  *Mật khẩu =”12345678”* | *Đăng nhập thành công* | *1,3* | *Pass* |

#### Chức năng thêm nhân viên

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Mã nhân viên* | *[a-zA-Z0-9],Chưa tồn tại trong cơ sở dữ liệu* | *1* | *Để trống,Mã nhân viên đã tồn tại* | *2* |
| *Họ* | *[a-zA-Z0-9]* | *3* | *Để trống* | *4* |
| *Tên* | *[a-zA-Z0-9]* | *5* | *Để trống* | *6* |
| *Ngày sinh* | *mm/dd/yyyy* | *7* | *Để trống* | *8* |
| *Địa chỉ* | *[a-zA-Z0-9]* | *9* | *Để trống* | *10* |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Mã nhân viên =””*  *Họ=””*  *Tên =””*  *Ngày sinh =””*  *Địa chỉ=””* | *Các trường không được trống* | *2,4,6,8,10* | *Pass* |
| *2* | *Mã nhân viên =”mnv1”*  *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Mã nhân viên đã tồn tại* | *1,2,5,7,9* | *Pass* |
| *3* | *Mã nhân viên =”mnv1”*  *Họ=””*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Họ không được để trống* | *1,4,5,7,9* | *Pass* |
| *4* | *Mã nhân viên =”mnv1”*  *Họ=”Trần Thị”*  *Tên =””*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Tên không được để trống* | *1,3,6,7,9* | *Pass* |
| *5* | *Mã nhân viên =”mnv1”*  *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”mm/đd/yyyy”*  *Địa chỉ=”Hà Nội”* | *Ngày sinh không được để trống.* | *1,3,5,8,9* | *Pass* |
| *6* | *Mã nhân viên =”mnv1”*  *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=””* | *Địa chỉ không được để trống* | *1,3,5,7,10* | *Pass* |
| *7* | *Mã nhân viên =”mnv20”*  *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Thêm nhân viên thành công* | *1,3,5,7,9* | *Pass* |

#### Chức năng sửa nhân viên

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Họ* | *[a-zA-Z0-9]* | *1* | *Để trống* | *2* |
| *Tên* | *[a-zA-Z0-9]* | *3* | *Để trống* | *4* |
| *Ngày sinh* | *mm/dd/yyyy* | *5* | *Để trống* | *6* |
| *Địa chỉ* | *[a-zA-Z0-9]* | *7* | *Để trống* | *8* |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Mã nhân viên =””*  *Họ=””*  *Tên =””*  *Ngày sinh =””*  *Địa chỉ=””* | *Các trường không được trống* | *2,4,6,8* | *Pass* |
| *3* | *Họ=””*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Họ không được để trống* | *2,3,5,7* | *Pass* |
| *4* | *Họ=”Trần Thị”*  *Tên =””*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=”Hà Nội”* | *Tên không được để trống* | *1,4,5,7* | *Pass* |
| *5* | *Mã nhân viên =”mnv1”*  *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”mm/đd/yyyy”*  *Địa chỉ=”Hà Nội”* | *Ngày sinh không được để trống.* | *1,3,6,7* | *Pass* |
| *6* | *Họ=”Trần Thị”*  *Tên =”C”*  *Ngày sinh =”12/09/2002”*  *Địa chỉ=””* | *Địa chỉ không được để trống* | *1,3,5,8* | *Pass* |
| *7* | *Họ=”Trần Thị”*  *Tên =”Thu”*  *Ngày sinh =”12/09/2008”*  *Địa chỉ=”Nghệ an”* | *Sửa nhân viên thành công* | *1,3,5,7* | *Pass* |

#### Chức năng xoá nhân viên

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Mã nhân viên* | *[a-zA-Z0-9],Tồn tại trong cơ sở dữ liệu* | *1* | *Không tồn tại* | *2* |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Mã nhân viên =””* | *Không tồn tại* | *2* | *Pass* |
| *2* | *Mã nhân viên =”mnv1”* | *Xoá thành công* | *1* | *Pass* |

#### Chức năng tìm kiếm,lọc nhân viên

* Mô tả điều kiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Điều kiện đầu vào*** | ***Lớp tương đương*** | | | |
| ***Vùng hợp lệ*** | ***Ký hiệu đánh dấu*** | ***Vùng không hợp lệ*** | ***Ký hiệu đánh dấu*** |
| *Tìm kiếm* | *[a-zA-Z0-9]* | *1* |  |  |

* Xây dựng testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TC*** | ***Đầu vào(Input)*** | ***Đầu ra mong đợi(Output)*** | ***Độ bao phủ(Cover)*** | ***Kết quả(Result)*** |
| *1* | *Tìm kiếm=”mnv1”* | *Trả về danh sách nhân viên có mã mnv1* | *1* | *Pass* |

Mô tả chức năng đăng nhập

Bảng quyết định

Chú thích:

TAG:

- Gom nhóm test case có chung mục đích, tương đồng kết quả xử lí, logic xử lý

- Thông thường các test case có cùng output thì gom chung 1 TAG

INPUT:

- Các đầu vào/ô nhập liệu của màn hình/form/page/ của chức năng cần test với giá trị nhập liệu tương ứng

\* Giá trị của các ô nhập trên màn hình có thể là:

- Đúng/Sai (True/False) ứng với 2 trạng thái có thể có của dữ liệu: hợp lệ và không hợp lệ

- Không/chưa xác định, khi người dùng bỏ trống không nhập (N/A NOT AVAILABLE)

\* CONDITION là các tổ hợp điều kiện đầu vào (input) và nó sẽ quyết định DECISION đầu ra (output) là gì

#### Chức năng đăng nhập

* Bảng quyết định

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Condition | Tag | | | | |
| A | B | C | D | E |
| Input | | | | | |
| 1 | Tài khoản | T | T | F | F | N/A |
| 2 | Mật khẩu | T | F | T | F | N/A |
| output | | | | | | |
| 1 | login successfully | X |  |  |  |  |
| 2 | login unsuccessfully |  | X | X | X | X |
| 3 | reset password? |  | X |  |  |  |
| 4 | sign-up account? |  |  | X | X |  |
| 5 | remember account? | X |  |  |  |  |
| 6 | fill in required fields |  |  |  |  | X |
| 7 | accumulate failed attempts | X | X | X | X | X |

* Xây dựng test case

|  |  |  |  |
| --- | --- | --- | --- |
| Test-case No | Test case description | Expected result | TAG |
| 1 | Check Login form with valid account | login successfully | A |
| 2 | Check Login form with invalid password | login unsuccessfully & suggest resetting password & accumulate failed attempts | B |
| 3 | Check Login form with invalid username | login unsuccessfully & suggest sign-up account & accumulate failed attempts | C |
| 4 | Check Login form with invalid username and password | login unsuccessfully & suggest sign-up account & accumulate failed attempts | D |
| 5 | Check Login form with empty fields | login unsuccessfully & alert empty fields & accumulate failed attempts | E |

#### Chức năng đăng ký

* Bảng quyết định

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Condition | Tag | | | | | |
| A | B | C | D | E | F |
| Input | | | | | |
| 1 | Tài khoản | T | F | T | T | T | N/A |
| 2 | Email | T | \_ | F | T | T | N/A |
| 3 | Mật khẩu | T | \_ | \_ | F | T | N/A |
| 4 | Nhập lại mật khẩu | T | \_ | \_ | \_ | F | N/A |
|  | Output | | | | | | |
| 1 | register successfully | X |  |  |  |  |  |
| 2 | register unsuccessfully |  | X | X | X | X |  |
| 3 | create account | X |  |  |  |  |  |
| 4 | fill in required fields |  |  |  |  |  | X |
| 5 | username exist or invalid |  | X |  |  |  |  |
| 6 | email invalid character |  |  | X |  |  |  |
| 7 | password invalid |  |  |  | X |  |  |
| 8 | Cofirmpassword invalid |  |  |  |  | X |  |

* Xây dựng testcase

|  |  |  |  |
| --- | --- | --- | --- |
| Test-case No | Test case description | Expected result | TAG |
| 1 | Check register form with valid info | register successfully | A |
| 2 | Check register form with username exist or invalid | register unsuccessfully & suggest username exist | B |
| 3 | Check register with invalid email | register unsuccessfully & suggest invalid email | C |
| 4 | Check register with invalid password | register unsuccessfully & suggest invalid confirmpassword | D |
| 5 | Check register with invalid confirmpassword | login unsuccessfully & alert empty fields | E |
| 6 | Check Login form with empty fields |  | F |

#### Chức năng thêm nhân viên

* Bảng quyết định

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Condition |  | Tag | | | | | |
| A | B | C | D | E | F | G |
| Input | | | | | | |
| 1 | Mã nhân viên | T | F | T | T | T | T | N/A |
| 2 | Họ | T | \_ | F | T | T | T | N/A |
| 3 | Tên | T | \_ | \_ | F | T | T | N/A |
| 4 | Ngày sinh | T | \_ | \_ | \_ | F | T | N/A |
| 5 | Địa chỉ | T | \_ | \_ | \_ | \_ | F | N/A |
|  | Output | | | | | | | |
| 1 | insert successfully | X |  |  |  |  |  |  |
| 2 | insert unsuccessfully |  | X | X | X | X | X | X |
| 3 | fill in required fields |  |  |  |  |  |  | X |
| 4 | Mã nhân viên existed |  | X |  |  |  |  | X |
| 5 | Họ invalid |  |  | X |  |  |  |  |
| 6 | Tên invalid |  |  |  | X |  |  |  |
| 7 | Ngày sinh invalid |  |  |  |  | X |  |  |
| 8 | Địa chỉ invalid |  |  |  |  |  | X |  |

* Xây dựng testcase

|  |  |  |  |
| --- | --- | --- | --- |
| Test-case No | Test case description | Expected result | TAG |
| 1 | Check insert with valid info | insert successfully | A |
| 2 | Check insert form with mnv exist | insert unsuccessfully & suggest manv exist&change manv | B |
| 3 | Check insert form with invalid ho | insert unsuccessfully & suggest change ho | C |
| 4 | Check insert form with invalid ten | insert unsuccessfully & suggest change ten | D |
| 5 | Check insert form with invalid ngaysinh | insert unsuccessfully & suggest change ngaysinh | E |
| 6 | Check insert form with invalid diachi | insert unsuccessfully & suggest fix diachi | F |
| 7 | Check insert form with empty fields | insert unsuccessfully & alert empty fields |  |

#### Chức năng sửa nhân viên

* Bảng quyết định

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Condition | Tag | | | | | |
| A | B | C | D | E | F |
| Input | | | | | |
| 1 | Họ | T | F | T | T | T | N/A |
| 2 | Tên | T | \_ | F | T | T | N/A |
| 3 | Ngày sinh | T | \_ | \_ | F | T | N/A |
| 4 | Địa chỉ | T | \_ | \_ | \_ | F | N/A |
|  | Output | | | | | | |
| 1 | update successfully | X |  |  |  |  |  |
| 2 | update unsuccessfully |  | X | X | X | X |  |
| 3 | fill in required fields |  |  |  |  |  | X |
| 4 | Họ invalid |  | X |  |  |  |  |
| 5 | Tên invalid |  |  | X |  |  |  |
| 6 | Ngày sinh |  |  |  | X |  |  |
| 7 | Địa chỉ invalid |  |  |  |  | X |  |

* Xây dựng testcase

|  |  |  |  |
| --- | --- | --- | --- |
| Test-case No | Test case description | Expected result | TAG |
| 1 | Check update with valid info | update successfully | A |
| 2 | Check update form with invalid ho | update unsuccessfully & suggest change ho | B |
| 3 | Check update form with invalid ten | update unsuccessfully & suggest change ten | C |
| 4 | Check update form with invalid Ngày sinh | update unsuccessfully & suggest change ngày sinh | D |
| 5 | Check update form with invalid diachi | update unsuccessfully & suggest fix diachi | E |
| 6 | Check update form with empty fields | update unsuccessfully & alert empty fields | F |

#### Chức năng xoá nhân viên

* Bảng quyết định

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Condition | Tag | | |
| A | B | C |
| Input | | |
| 1 | Mã nhân viên | T | F | N/A |
|  |  | Output | | |
| 1 | register successfully | X |  |  |
| 2 | register unsuccessfully |  | X | X |

* Xây dựng testcase

|  |  |  |  |
| --- | --- | --- | --- |
| Test-case No | Test case description | Expected result | TAG |
| 1 | Check delete NhanVien form with valid ma nhan vien | delete successfully | A |
| 2 | Check delete nhanvien with not exist ma nhan vien | delete unsuccessfully & suggest ma nhan vien not exist | B |
| 3 | Check nhan vien with empty fields | delete unsuccessfully & alert empty fields | C |

Kiểm thử hệ thống(system test)

#### Chức năng đăng ký

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng đăng ký khi trống | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Tích chọn đăng ký khi chưa nhập dữ liệu vào form | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Hiển thị thông báo nội dung "Thông tin bị trống" |  | Pass | 25/04/2024 | Basic flow |
| 2 | Kiểm tra tính năng đăng ký khi email sai định dạng | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: email:user@ Tích chọn đăng ký | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Hiển thị thông báo nội dung "Vui lòng nhập email theo định dạng user@gmail.com" |  | Pass | 25/04/2024 | Basic flow |
| 3 | Kiểm tra tính năng đăng ký khi mật khẩu quá ngắn hoặc dài | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: password:1234 Tích chọn đăng ký  4.Nhập dữ liệu: password:1234 Tích chọn đăng ký | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Hiển thị thông báo nội dung "Mật khẩu yêu cầu từ 8-12 ký tự" 4.Hiển thị thông báo nội dung "Mật khẩu yêu cầu từ 8-12 ký tự" |  | Pass | 25/04/2024 | Basic flow |
| 4 | Kiểm tra tính năng đăng ký khi tài khoản đã tồn tại | 1. Cài ide eclipse cho web,Mysql WorkBench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: Tài khoản:user Tích chọn đăng ký | 3.Hiển thị thông báo nội dung "Mật khẩu yêu cầu từ 8-12 ký tự" |  | Pass | 25/04/2024 | Basic flow |
| 5 | Kiểm tra tính năng đăng ký khi mật khẩu nhập lại không trùng | 1. Cài ide eclipse cho web,Mysql WorkBench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: Mật khẩu:1234678 Xác nhận mật khẩu:123456 Tích chọn đăng ký | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Hiển thị thông báo nội dung "Mật khẩu và xác nhận mật khẩu không khớp." |  | Pass | 25/04/2024 | Basic flow |
| 6 | Kiểm tra tính năng đăng ký khi tài khoản đã tồn tại | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: Tài khoản:user Tích chọn đăng ký | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Hiển thị thông báo nội dung "Tài khoản đã tồn tại vui lòng thử lại." |  | Pass | 25/04/2024 | Basic flow |
| 7 | Kiểm tra tính năng đăng ký khi tài khoản nhập đúng yêu cầu | 1. Cài ide eclipse cho web,MySQL Workbench  2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/register | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng ký 3.Nhập dữ liệu: Tài khoản:user1 Email:user@gmail.com Mật khẩu:12345678 Xác nhận mật: khẩu:12345678 Tích chọn đăng ký | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang đăng ký và hiển thị form cho đăng ký 3.Chuyển hướng đến giao diện đăng nhập |  | Pass | 25/04/2024 | Basic flow |

#### Chức năng đăng nhập

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng nhập khi trống | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/login | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng nhập 3.Tích chọn đăng nhập khi chưa nhập dữ liệu vào form | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang nhập và hiển thị form cho nhập 3.Hiển thị thông báo nội dung "Vui lòng nhập tên người dùng và mật khẩu." |  | Pass | 25/04/2024 | Basic flow |
| 2 | Kiểm tra tính năng đăng ký khi mật khẩu quá ngắn hoặc dài | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/login | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng nhập 3.Nhập dữ liệu: Mật khẩu:123 Tích chọn đăng nhập  4.Nhập dữ liệu: Mật khẩu:123 Tích chọn đăng nhập | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang nhập và hiển thị form cho nhập 3.Hiển thị thông báo nội dung "Mật khẩu phải có từ 8 đến 12 ký tự." 4.Hiển thị thông báo nội dung "Mật khẩu phải có từ 8 đến 12 ký tự." |  | Pass | 25/04/2024 | Basic flow |
| 3 | Kiểm tra tính năng đăng ký khi tài khoản không đúng | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/login | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng nhập 3.Nhập dữ liệu: Tài khoản:us Mật khẩu:12345678 Tích chọn đăng nhập | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang nhập và hiển thị form cho nhập 3.Hiển thị thông báo nội dung "Tên người dùng hoặc mật khẩu không chính xác." 4.Hiển thị thông báo nội dung "Mật khẩu phải có từ 8 đến 12 ký tự." |  | Pass | 25/04/2024 | Basic flow |
| 4 | Kiểm tra tính năng đăng ký khi tài khoản hợp lệ đã đăng ký | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/login | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn đăng nhập 3.Nhập dữ liệu: Tài khoản:user Mật khẩu:12345678 Tích chọn đăng nhập | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang nhập và hiển thị form cho nhập 3.Chuyển hướng về trang chủ,lưu cookie tài khoản đăng nhập |  | Pass | 25/04/2024 | Basic flow |

#### Thêm nhân viên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng thêm nhân viên khi không nhập | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/new | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn thêm nhân viên mới 3.Tích chọn lưu khi chưa nhập dữ liệu vào form | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang thêm và hiển thị form cho thêm nhân viên 3.Hiển thị thông báo nội dung "Yêu cầu điền thông tin" |  | Pass | 25/04/2024 | Basic flow |
| 2 | Kiểm tra tính năng mã nhân viên đã tồn tại | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/new | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn thêm nhân viên mới 3.Nhập dữ liệu: Mã nhân viên:mnv1 Tích chọn lưu | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang thêm và hiển thị form cho thêm nhân viên 3.Hiển thị thông báo nội dung "Mã nhân viên đã tồn tại" |  | Pass | 25/04/2024 | Basic flow |
| 3 | Kiểm tra tính năng thêm nhân viên khi nhập hợp lệ | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/new | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn thêm nhân viên mới 3.Nhập dữ liệu: Mã nhân viên:mnv02 Họ:Nguyen Van Tên:E Giới tính:nam Ngày sinh:25/03/2002 Nơi sinh:Hà Nội Tích chọn lưu | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang thêm và hiển thị form cho thêm nhân viên 3.Chuyển hướng trang chủ và thông báo "thêm thành công" |  | Pass | 25/04/2024 | Basic flow |

#### Chức năng sửa nhân viên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng sửa nhân viên khi để trống | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/edit?maNV=mnv1 | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn sửa cho nhân viên 3.Các trường dữ liệu trống Tích chọn lưu | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang thêm và hiển thị form cho thêm nhân viên 3.Hiển thị thông báo nội dung "Yêu cầu điền thông tin" |  | Pass | 25/04/2024 | Basic flow |
| 2 | Kiểm tra tính năng sửa nhân viên khi hợp lệ | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/edit?maNV=mnv1 | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn sửa cho nhân viên 3.Các trường dữ liệu không trống Tích chọn lưu | 1. Hiển thị giao diện danh sách nhân viên 2. Chuyển hướng đến trang thêm và hiển thị form cho thêm nhân viên 3.Chuyển về trang chủ Hiển thị thông báo nội dung "Sửa thành công" |  | Pass | 25/04/2024 | Basic flow |

#### Xóa nhân viên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng xoá nhân viên | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/list | 1.Từ trang chủ chọn nút bắt đầu 2.Chọn xoá cho nhân viên cần xoá 3.Xác nhận xoá | 1. Hiển thị giao diện danh sách nhân viên 2. Hiển thị thông báo xác nhận 3.Hiển thị thông báo nội dung "Xoá thành công" |  | Pass | 25/04/2024 | Basic flow |

#### Tìm kiếm,lọc nhân viên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Pre -Condition (Data/Environment)** | **Test Case Procedure/Steps** | **Expected Output** | Bug# | **Status** | **Test date** | **Note** |
| 1 | Kiểm tra tính năng tìm kiếm nhân viên | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/list | 1.Từ trang chủ chọn nút bắt đầu 2.Nhập tìm kiếm theo tên B | 1. Hiển thị giao diện danh sách nhân viên 2. Hiển thị danh sách nhân viên có ký tự B |  | Pass | 25/04/2024 | Basic flow |
| 1 | Kiểm tra tính năng lọc nhân viên | 1. Cài ide eclipse cho web,MySQL Workbench 2. tạo cơ sở dữ liệu nhanviendb 3. Thêm thư viện SRE,Junit5,tomcat 9 4. Chạy dự án CRUD\_Servlet 5.Truy cập url:http://localhost:8083/CRUD\_Servlet/list | 1.Từ trang chủ chọn nút bắt đầu 2.Lọc nhân viên theo giới tính nam | 1. Hiển thị giao diện danh sách nhân viên 2. Hiển thị danh sách nhân viên có giới tính nam |  | Pass | 25/04/2024 | Basic flow |

#### Chú thích:

* F: email/mật khẩu/ Tên người dùng /nhập lại mật khẩu không hợp lệ
* T: email/ Tên người dùng /mật khẩu/nhập lại mật khẩu hợp lệ
* E: hiển thị lỗi email, tên người dùng hoặc mật khẩu chưa hợp lệ
* E1: hiển thị lỗi nhập lại mật khẩu chưa khớp với mật khẩu
* H: chuyển đến màn hình trang chủ

#### Diễn giải:

Trường hợp 1: email, mật khẩu và confirm mật khẩu đúng, người dùng sẽ được chuyển hướng đến trang chủ.

Trường hợp 2: email đúng, tên người dùng sai, mật khẩu đúng, nhập lại mật khẩu đúng => nhận thông báo lỗi E.

Trường hợp 3: email đúng , tên người dùng đúng, mật khẩu sai, nhập lại mật khẩu sai => người dùng sẽ nhận được thông báo lỗi E1.

Trường hợp 4: email sai, tên người dùng và mật khẩu đúng, nhập lại mật khẩu đúng => người dùng sẽ nhận được thông báo lỗi E.

Trường hợp 5: email đúng, tên người dùng, mật khẩu đúng, nhập lại mật khẩu sai

=> người dùng sẽ nhận được thông báo lỗi E1.

Trường hợp 6: email sai, tên người dùng đúng,mật khẩu đúng, nhập lại mật khẩu sai => người dùng sẽ nhận được thông báo lỗi E.

Trường hợp 7: email sai, tên người dùng sai,mật khẩu sai, nhập lại mật khẩu sai

=> người dùng sẽ nhận được thông báo .

Kiểm thử dịch chuyển trạng thái

A diagram of a diagram

Description automatically generated

*Hình 4. 2 Dịch chuyển trạng thái*

|  |  |  |
| --- | --- | --- |
| Trạng thái | Sự kiện (Event- Action) | Number of Items |
| S1 | Thêm nhân viên | 1 |
| S2 | Cancel | 0 |
| S1 | Thêm nhân viên | 1 |
| S2 | Thêm nhân viên | 2 |
| S2 | Thêm nhân viên | 3 |
| S2 | Bỏ thêm nhân viên | 2 |
| S2 | Sửa | 2 |
| S3 | Cancel | 2 |

|  |  |  |
| --- | --- | --- |
| S2 | Bỏ nhân viên | 1 |
| S2 | Xoá | 1 |
| S3 | Xác nhận | 1 |
| S4 | Xác nhận | 1 |

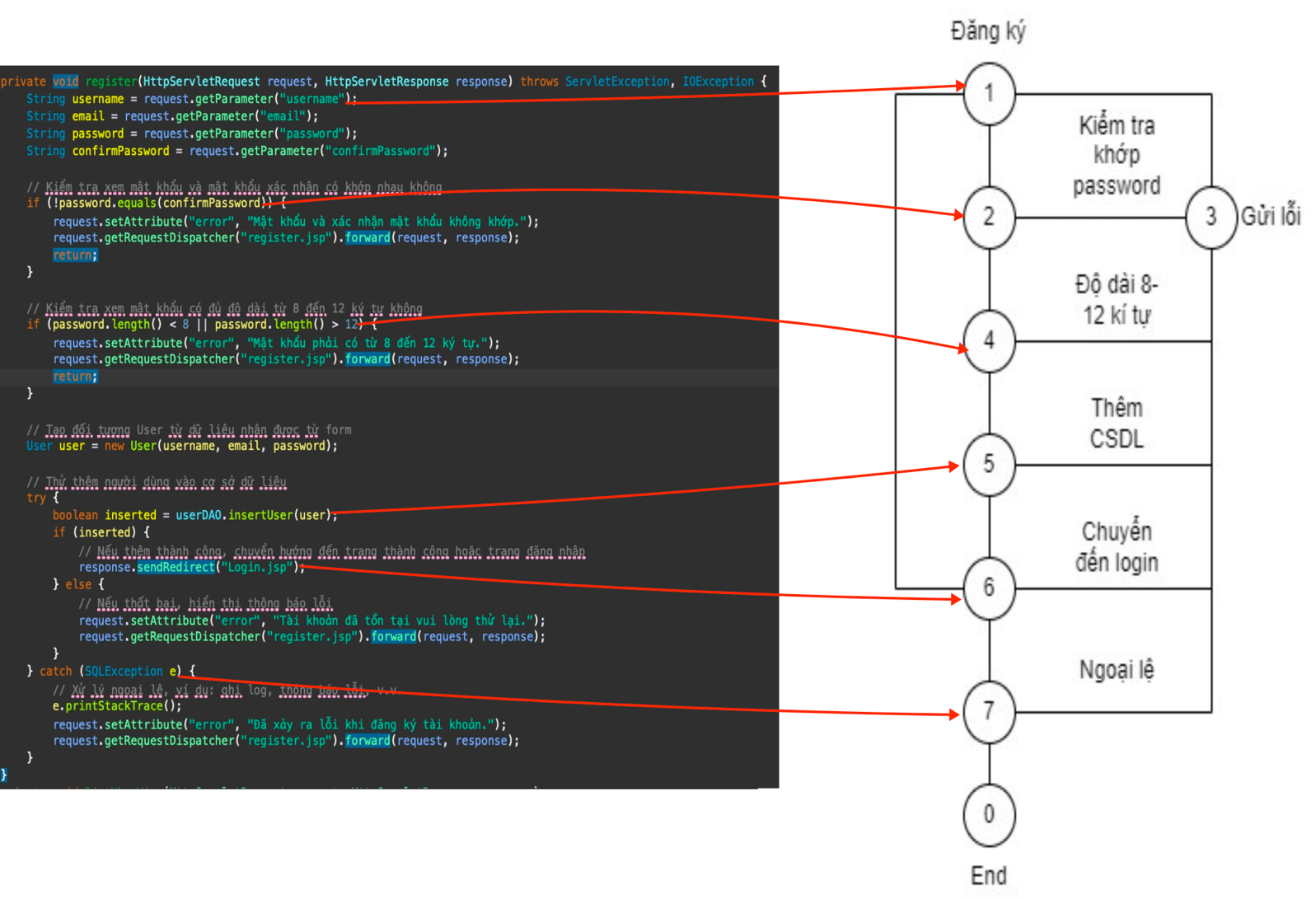
**Bảng 3. 8 Giá trị biên**

### Kiểm thử hộp trắng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Đường dẫn |  | Đầu vào | | | Đầu ra mong muốn |
| User name | Password | Email | Model |
| a, b, d, h, k | Sai | Sai | Sai | Có giá trị | Trở về trang Register |
| a, b, d, g | Đúng | Đúng | Tồn  tại | Có giá trị | Thông báo email đã  tồn tại |
| a, b, d, h, e,i | Đúng | Đúng | Đúng | Có giá trị | Thông báo đăng ký thành công |
| a, b, d, h, e,f, l | Sai | Sai | Đúng | Có giá trị | Trở về trang Register |
| a, c |  |  |  | Không có giá trị | Trở về trang Register |

Bảng 3.3. 1 Kiểm thử hộp trắng

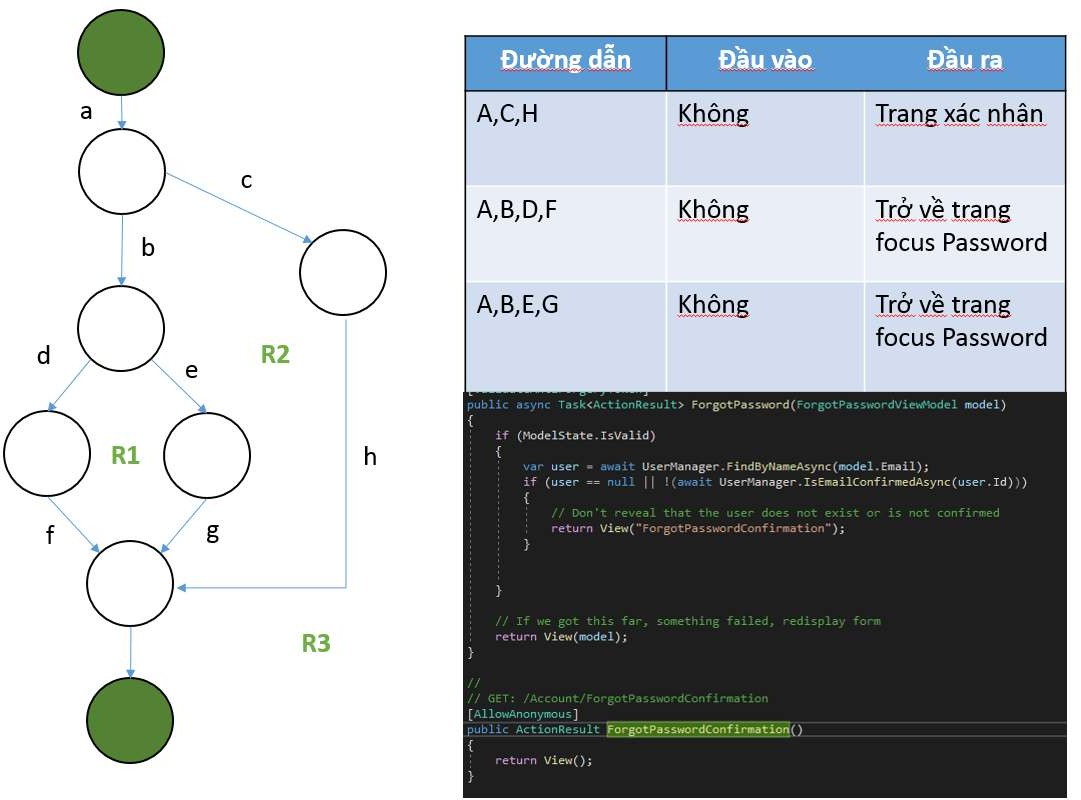
#### Đường dẫn cơ sở



Hình 3. 3 Đường cơ sở

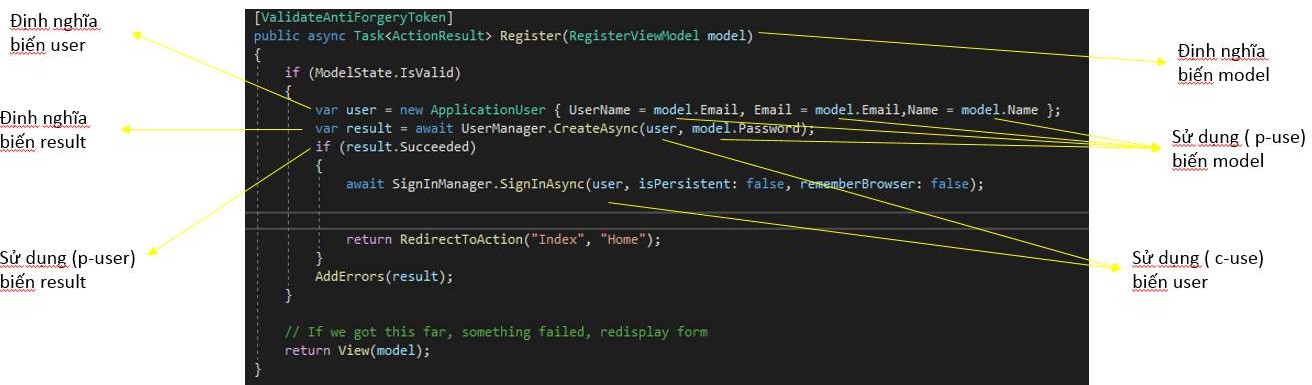
Bảng 3.9 Bảng

#### Phủ nhánh



Hình 3. 4 Phủ nhánh

#### Luồng dữ liệu



Hình 3. 5 Luồng dữ liệu

1. KIỂM THỬ TỰ ĐỘNG

## Công cụ kiểm thử Selenium

### Giới thiệu

Giới thiệu tóm tắt Selenium IDE Selenium Integrated Development Environment (IDE) là framework đơn giản nhất trong bộ Selenium và là cách đơn giản nhất để học. Đây là một plugin của Firefox mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì tính đơn giản của nó, Selenium IDE chỉ nên được sử dụng như là một công cụ tạo mẫu . Nếu bạn muốn tạo các trường hợp kiểm thử nâng cao hơn, bạn sẽ cần phải sử dụng Selenium RC hoặc WebDriver.

* + - * Selenium là công cụ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau.
      * Selenium tập trung vào việc tự động hoá các ứng dụng dựa trên web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium.
      * Selenium không chỉ là một công cụ mà còn là một bộ phần mềm, mỗi bộ đều đáp ứng nhu cầu kiểm tra khác nhau của một tổ chức.
      * Nó gồm bốn thành phần bên dưới:

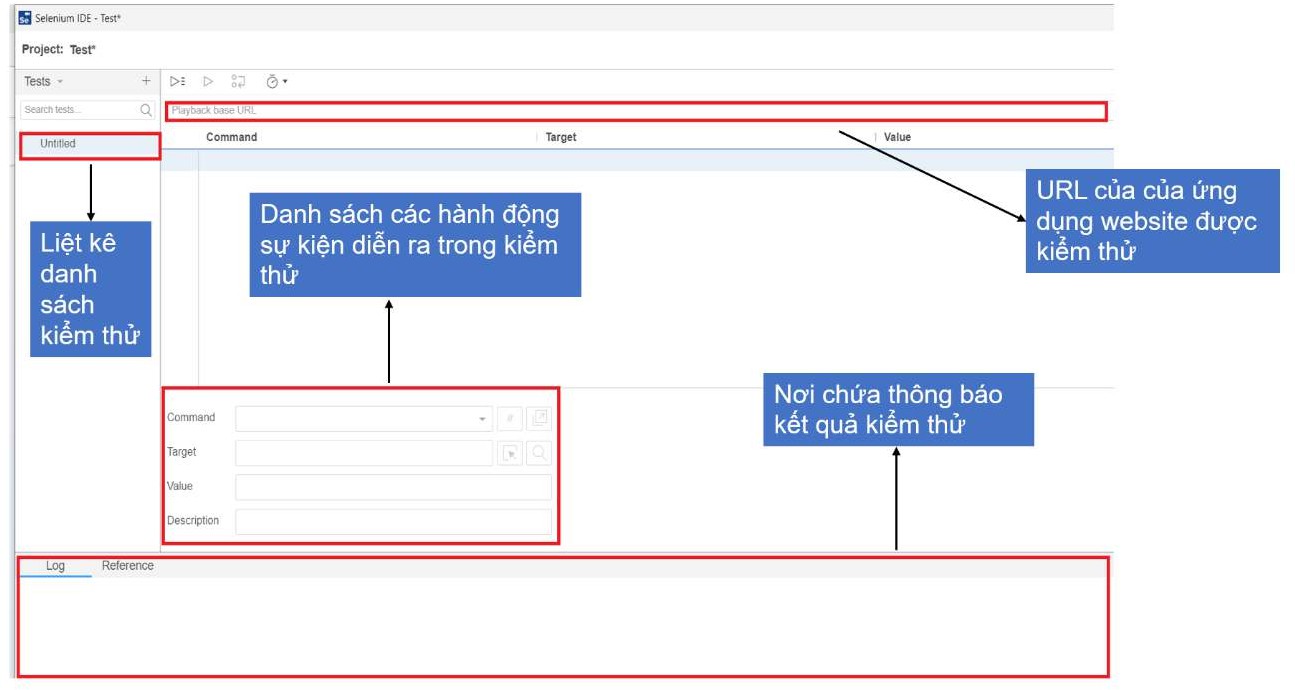
+ Selenium Integrated Development Environment (IDE)

+ Selenium Remote Control (RC)

+ WebDriver

+ Selenium Grid

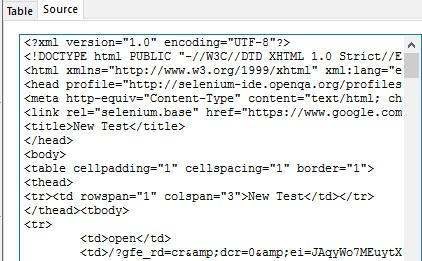
### Chức năng



*Hình 4. 1* Chi tiết về giao diện công cụ Selenium

* Base URL: Chứa địa chỉ URL của ứng dụng Web đang được kiểm thử.
* Thanh trượt: Điều chỉnh tốc độ thực thi ca kiểm thử tự động.
* Chạy bộ kiểm thử: Chạy tất cả các ca kiểm thử có trong bộ kiểm thử (test suite).
* Chạy một ca kiểm thử: Chạy ca kiểm thử hiện tại được chọn.
* Tạm dừng: Tạm dừng ca kiểm thử đang thực thi.
* Tiếp tục: Tiếp tục chạy ca kiểm thử vừa tạm dừng.
* Bỏ qua: Bỏ qua một bước ở vị trí nó đang dừng.
* Ghi (Record): Ghi lại các thao tác trong ca kiểm thử. Mặc định khi khởi động Selenium IDE, chức năng ghi thao tác tự động được kích hoạt.
* Textbox Command: Lưu trữ các hành động (action) trong ca kiểm thử.
* Textbox Target: Đích đến của thao tác.
* Textbox Value: Giá trị đầu vào cho thao tác trong ca kiểm thử.

Tab Source là nơi hiển thị kịch bản ca kiểm thử dưới dạng HTML. Đây cũng chính là source của file HTML sau khi tiến hành lưu lại ca kiểm thử cho mục đích sử dụng lại.



Hình 4. 3 Kịch bản kiểm thử được Selenium IDE lưu trữ dưới dạng HTML.

Ngoài ra, các thông báo cho thao tác vừa thực hiện, thông báo lỗi được thể hiện ở khung ghi Log.

Trong Selenium IDE, các ca kiểm thử luôn có điểm bắt đầu. Điều này tương ứng với cách thiết kế tiền điều kiện (Pre-condition) trong ca kiểm thử. Điển hình nhất cho điểm bắt đầu của một ca kiểm thử với Selenium IDE là việc truy cập một ứng dụng Web để bắt đầu luồng công việc.

### Áp dụng kiểm thử

Nhóm em sẽ áp dụng kiểm thử chức năng login của trang website Rick Story.

Bằng cách thực hiện kiểm thử tự động với đường dẫn cơ sở:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Đường dẫn |  | Đầu vào | | | Đầu ra mong muốn |
| User name | Password | Email | Model |
| a, b, d, h, k | Sai | Sai | Sai | Có giá trị | Trở về trang Register |
| a, b, d, g | Đúng | Đúng | Tồn  tại | Có giá trị | Thông báo email đã  tồn tại |
| a, b, d, h, e,i | Đúng | Đúng | Đúng | Có giá trị | Thông báo đăng ký thành công |
| a, b, d, h, e,  f, l | Sai | Sai | Đúng | Có giá trị | Trở về trang Register |
| a, c |  |  |  | Không có giá trị | Trở về trang Register |

Bảng 4. 1 Mô tả đường dẫn

A screenshot of a computer

Description automatically generated

*Hình 5. 1 Kết quả đăng ký trống*

*A screenshot of a computer

Description automatically generated*

*Hình 5. 2 Kết quả đăng ký mật khẩu không hợp lệ*

A screenshot of a computer

Description automatically generated

*Hình 5. 3 Kết quả đăng ký email không hợp lệ*

A screenshot of a computer

Description automatically generated

*Hình 5. 4 Kết quả đăng ký mật khẩu xác nhận không hợp lệ*

A screenshot of a computer

Description automatically generated

*Hình 5. 5 Kết quả đăng ký thành công*

*A screenshot of a computer

Description automatically generated*

*Hình 5. 6 Kết quả đăng nhập trống*

A screenshot of a computer

Description automatically generated

*Hình 5. 7 Kết quả đăng nhập không hợp lệ*

*A screenshot of a computer

Description automatically generated*

*Hình 5. 8 Kết quả đăng nhập thành công*

A screenshot of a computer

Description automatically generated

*Hình 5. 9 Kết quả thêm nhân viên không thành công*

*A screenshot of a computer

Description automatically generated*

*Hình 5. 10 Kết quả thêm nhân viên đã tồn tại*

*A screenshot of a computer

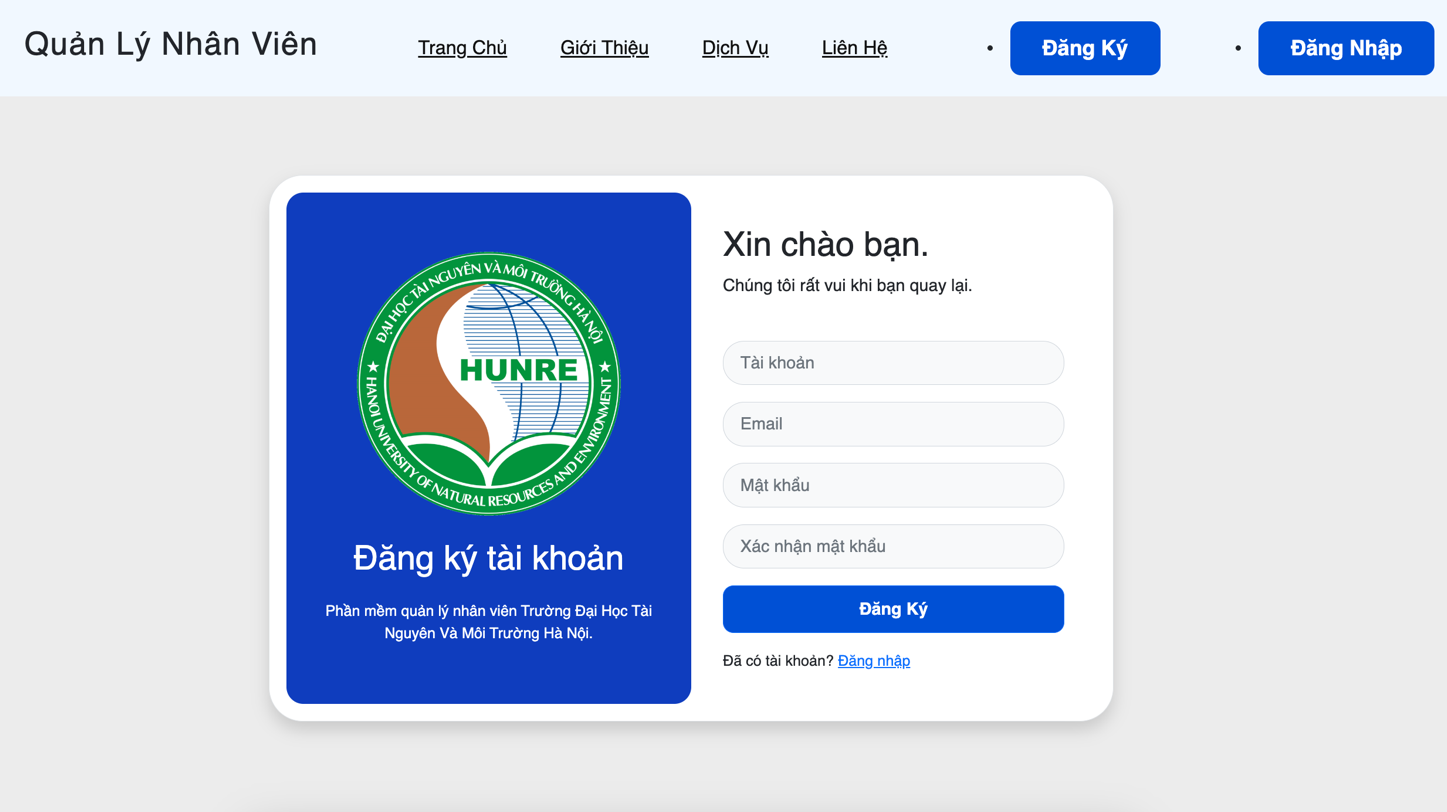
Description automatically generated*

*Hình 5. 11 Kết quả thêm nhân viên thành công*

*A screenshot of a computer

Description automatically generated*

*Hình 5. 12 Kết quả sửa nhân viên thành công*



Hình 4. 5 Thực thi Selenium

**Kết luận:** Quá trình kiểm thử tự động Selenium đã thành công với đường dẫn cơ đưa ra là trả về trang Register

## Công cụ kiểm thử Jmeter

### Giới thiệu

Apache JMeter là một phần mềm mã nguồn mở được dùng để kiểm tra độ chịu tải (load test) và hiệu suất (measure performance).

Apache JMeter được viết 100% bằng Java, lúc đầu nó được thiết kế để kiểm thử các ứng dụng web nhưng sau đó được mở rộng để kiểm thử các chức năng khác.

### Các tính năng

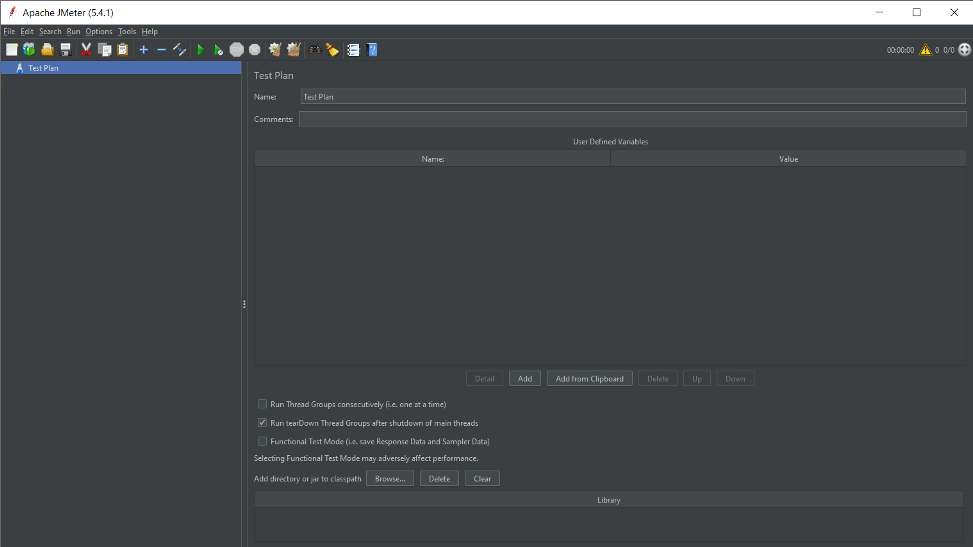
Các tính năng chính của JMeter bao gồm:

* Giấy phép (License)
* Giao diện đồ hoạ người dùng (Graphical User Interface – GUI)
* Hỗ trợ Giao thức/ Máy chủ (Server/Protocol Support)
* Nền tảng (Platform)
* Mô phỏng (Simulation)
* Hỗ trợ Phân phối tải (Supports Distributed Testing)
* Hiển thị kết quả (Test Result Visualation)
* Báo cáo (Reporting)
* Các loại kiểm thử (Testing Types)
* Record and Playback
* Framework
* Cài đặt (Installation)
* Kiến thức (Knowledge)

### Cách thức hoạt động

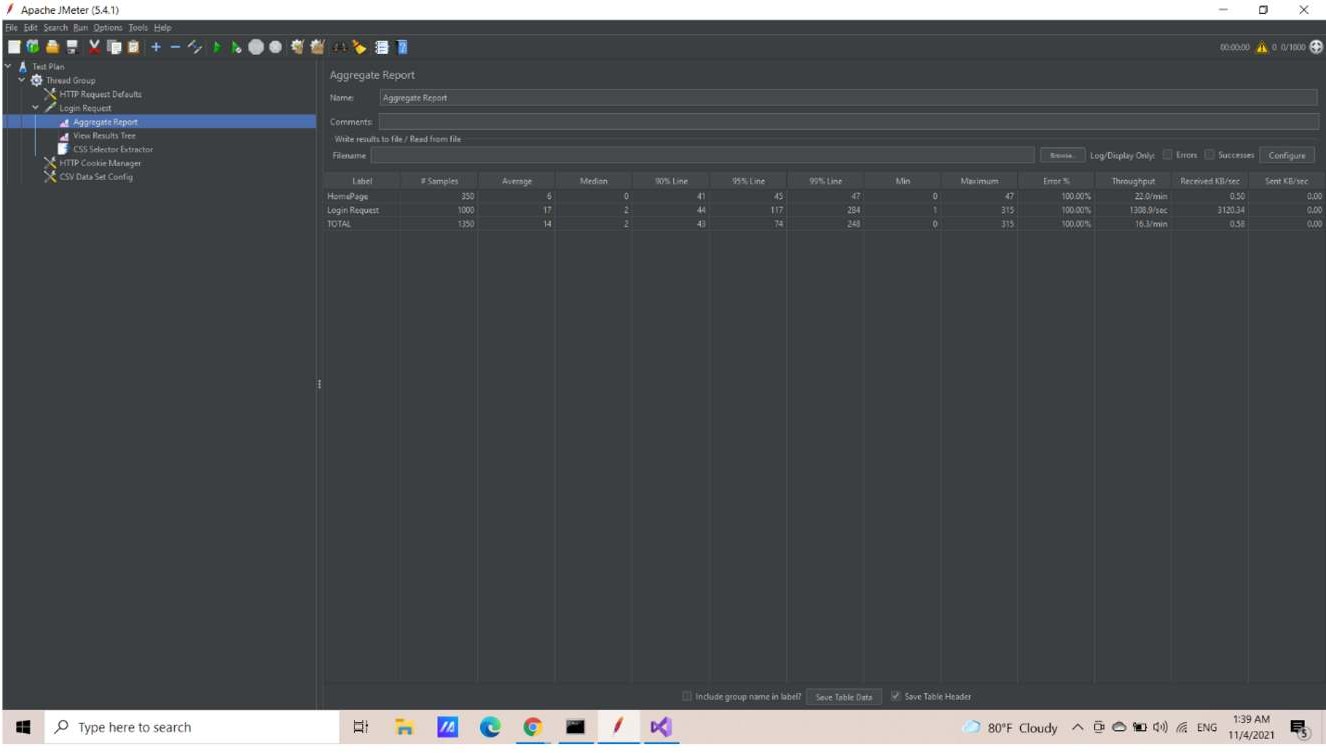
* Thời gian phản hồi: check xem ứng dụng phản hồi nhanh hay chậm
* Khả năng mở rộng: xác định tải người dùng tối đa mà ứng dụng phần mềm có thể xử lý. Tính ổn định: Xác định xem ứng dụng có ổn định dưới các tải khác nhau hay không.

### Giao diện



Hình 4. 8 Giao diện Jmeter

### Áp dụng kiểm thử



Hình 4. 9 Áp dụng kiểm thử Jmeter

## Công cụ kiểm thử LoadStom

### Giới thiệu

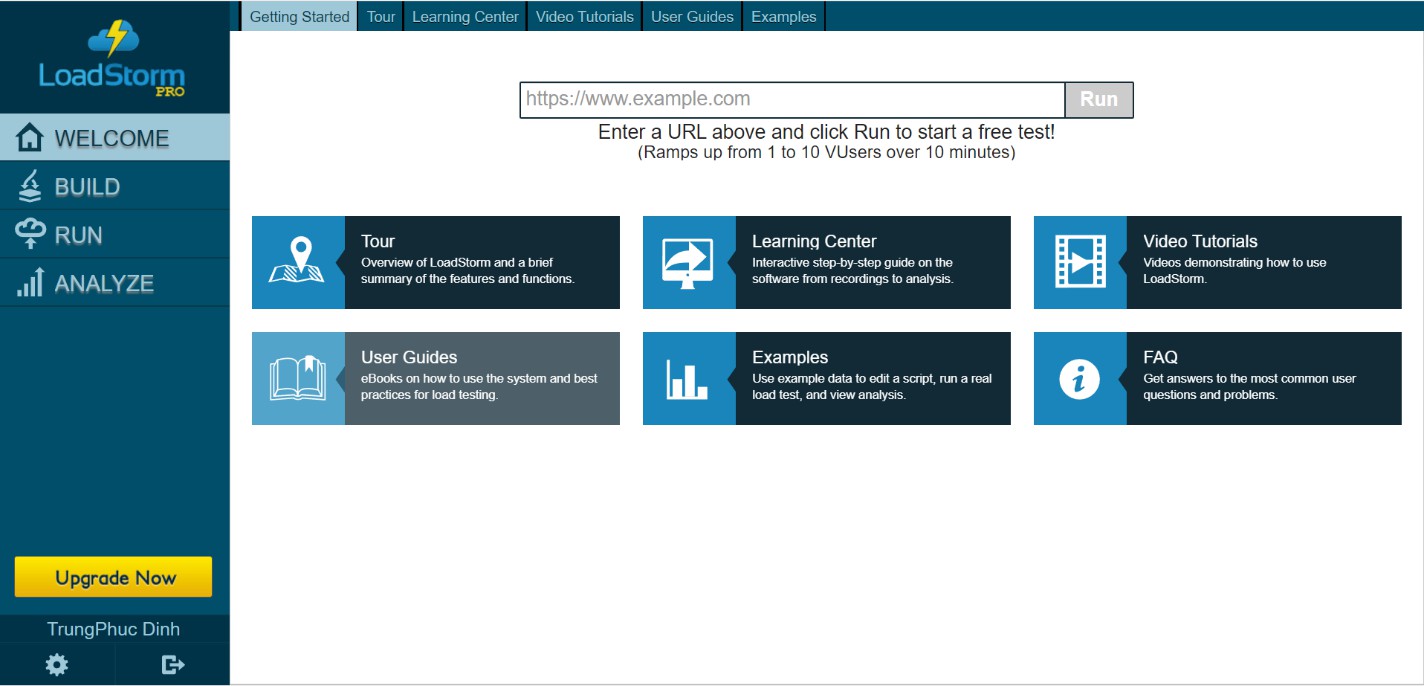
LoadStorm là công cụ có khả năng chịu tải rất tốt. Điểm đặc biệt ở công cụ này là nó có thể thiết lập hàng trăm nghìn , hàng triệu user để khai thác lỗ hổng trong ứng dụng.

LoadStorm tạo tải HTTP từ đám mây dựa trên kế hoạch thử nghiệm và kịch bản người dùng . Loadstorm được sử dụng như một thử nghiệm tải, kiểm tra căng thẳng hoặc công cụ kiểm tra độ bền

### Các tính năng

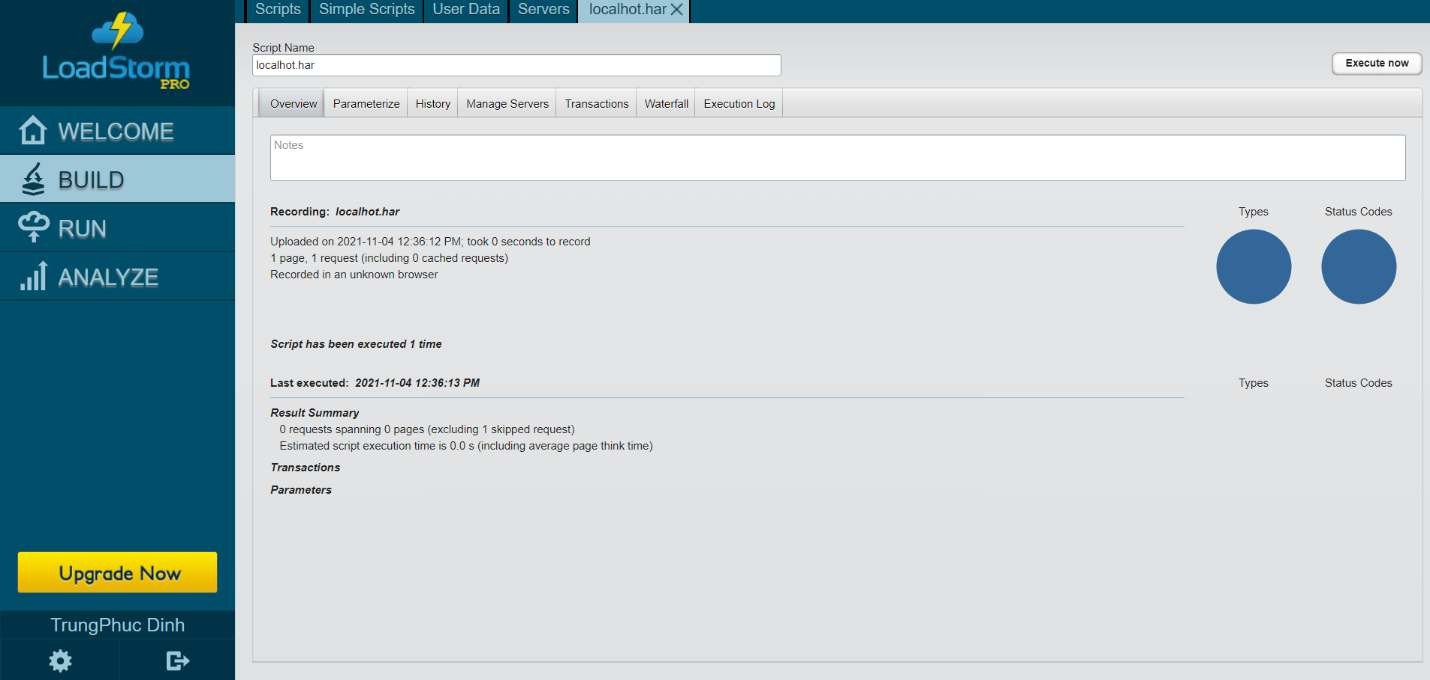
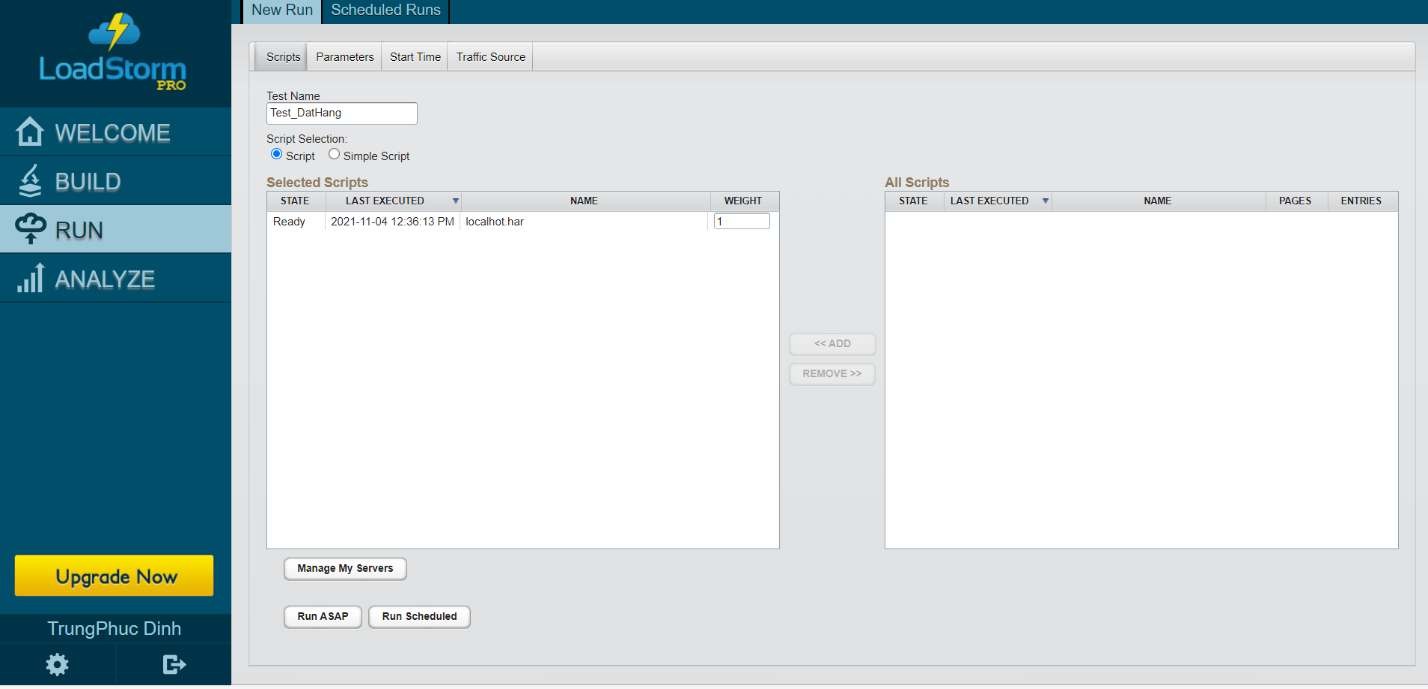
* + - * Nền tảng dựa trên đám mây
      * Dễ dàng ghi lại kịch bản
      * Điều khiển kịch bản tinh vi
      * Phân tích chuyên sâu
      * Tư vấn

### Áp dụng kiểm thử



Hình 4. 16 Giao diện LoadStom

88



Hình 4. 18 Giao diện viết kịch bản test công cụ LoadStorm



Hình 4. 19 Biểu đồ thể hiện số lượng user đăng nhập LoadStorm

KẾT LUẬN

Kiểm thử phần mềm, một hướng đi không còn mới mẻ trên thế giới, nhưng lại là một hướng đi rất mới ở Việt Nam. Nó hứa hẹn một tương lai mới cho các học sinh, sinh viên ngành CNTT. Qua tìm hiểu và thực hiện đồ án này, nhóm thấy đã đạt được một ưu điểm cũng như vẫn còn một số tồn tại

* Những điểm đạt được:
* Nắm được tổng quan về kiểm thử phần mềm: Các khái niệm cơ bản, các phương pháp kiểm thử phần mềm, và các vấn đề liên quan …
* Tìm hiểu và nắm được các phương pháp và chiến lược thiết kế test – case trong kiểm thử phần mềm, và áp dụng được các phương pháp đã tìm hiểu để xây dựng các test – case cho 1 bài toán cụ thể - Website cụ thể Rich Story
* Bổ sung kiến thức và rèn luyện thêm kỹ năng sử dụng các công cụ kiểm thử tự động được sử dụng trong đồ án
* Nâng cao khả năng đọc hiểu tài liệu Tiếng Anh Những điểm chưa đạt:
* Sự áp dụng những kiến thức tìm hiểu được mới chỉ dừng lại ở Website còn nhỏ chưa có sự logic như các website ngoài thực tế
* Từ những điểm đạt và chưa đạt ở trên, nhóm em hi vọng sẽ nhận được sự góp ý chân thành của các thầy để nhóm hoàn thiện hơn giúp thực hiện các đồ án quan trọng trong năm học cuối!!!

TÀI LIỆU THAM KHẢO

1. The Art of Software Testing, Glenford J. Myers, Second Edition, John Wiley and Sons, Inc.
2. Software Engineering - A Practitioner’s Approach, Roger S.Pressman, Sixth Edition, Ph.D, McGraw-Hill, Inc.
3. A Practitioner's Guide to Software Test Design, Lee Copeland, First Edition, Artech House Publishers Boston, London.
4. Effective methods for Software Testing, William E. Perry, 3rd Edition, Wiley Publishing, Indian.
5. Software Testing, Ron Patton, Second Edition, Sam Publishing.
6. <http://www.vietnamesetestingboard.org/>
7. <http://en.wikipedia.org/wiki/Software_testing>
8. Một số trang web về kiểm thử phần mềm khác.

# 