TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

🙥🕮🙧



BÁO CÁO BÀI TẬP LỚN

KIỂM THỬ PHẦN MỀM

NHÓM 3

Đề tài: Kiểm thử website quản lý khách sạn với phần mềm Selenium

**Giảng viên hướng dẫn: Nguyễn Thu Hường**

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | Mã sinh viên |
| 1 | Nguyễn Quỳnh Chi |  |
| 2 | Vũ Bảo Lâm | 211241205 |
| 3 | Đỗ Thị Hải | 211243882 |

**Lớp: Công nghệ thông tin 1 – Khoá 62**

**Hà Nội – 2024**

LỜI NÓI DẦU

Trong thời đại công nghệ phát triển nhanh chóng, nhu cầu quản lý hiệu quả và tối ưu hóa các quy trình kinh doanh trở nên vô cùng quan trọng đối với các doanh nghiệp trong nhiều lĩnh vực, đặc biệt là ngành dịch vụ khách sạn. Để đáp ứng nhu cầu đó, phần mềm quản lý khách sạn đóng vai trò quan trọng trong việc nâng cao chất lượng phục vụ, tối ưu hóa quy trình và cải thiện trải nghiệm khách hàng. Phần mềm quản lý khách sạn mà nhóm chúng em tự phát triển không chỉ nhằm giải quyết các vấn đề về quản lý phòng, đặt phòng và thanh toán mà còn chú trọng đến các tính năng bảo mật và hiệu năng, giúp hệ thống hoạt động ổn định và an toàn.

Kiểm thử phần mềm là một bước không thể thiếu để đảm bảo chất lượng của phần mềm trước khi đưa vào sử dụng. Trong đề tài này, nhóm chọn công cụ Selenium cho quá trình kiểm thử tự động, nhằm giúp tiết kiệm thời gian và gia tăng hiệu quả trong việc kiểm tra các chức năng của hệ thống quản lý khách sạn. Selenium, với khả năng tương thích với nhiều trình duyệt và ngôn ngữ lập trình, đã trở thành công cụ hữu ích để nhóm kiểm thử website một cách toàn diện, đảm bảo rằng phần mềm hoạt động chính xác, đáp ứng được kỳ vọng của người sử dụng.

Chúng em xin chân thành cảm ơn!

**Nhóm sinh viên thực hiện**

Nhóm 3MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU CHUNG 1](#_Toc181893512)

[1.1. Tổng quan đề tài: 1](#_Toc181893513)

[1.2. Lý do chọn đề tài: 1](#_Toc181893514)

[1.3. Mục tiêu và phạm vi: 2](#_Toc181893515)

[1.4. Phương pháp nghiên cứu 2](#_Toc181893516)

[CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ KHÁCH SẠN VÀ KIỂM THỬ PHẦN MỀM 3](#_Toc181893517)

[2.1. Tổng quan về hệ thống quản lý khách sạn: 3](#_Toc181893518)

[2.1.1. Giới thiệu hệ thống quản lý khách sạn 3](#_Toc181893519)

[2.1.2. Lợi ích và thách thức 3](#_Toc181893520)

[2.2. Đặc tả hệ thống quản lý khách sạn 3](#_Toc181893521)

[2.2.1. Yêu cầu chức năng 3](#_Toc181893522)

[2.2.2. Yêu cầu phi chức năng 4](#_Toc181893523)

[2.2.3. Sơ đồ tuần tự 4](#_Toc181893524)

[2.3. Kiểm thử phần mềm: 7](#_Toc181893525)

[2.3.1. Khái niệm và vai trò của kiểm thử phần mềm 7](#_Toc181893526)

[2.3.2. Quy trình kiểm thử phần mềm 7](#_Toc181893527)

[2.3.3. Phân loại kiểm thử phần mềm 8](#_Toc181893528)

[2.3.4. Kiểm thử tự động và vai trò của Selenium 10](#_Toc181893529)

[CHƯƠNG 3: TỔNG QUAN VỀ CÔNG NGHỆ KIỂM THỬ SELENIUM VÀ SELENIUM WEBDRIVER 12](#_Toc181893530)

[3.1. Khái quát về Selenium 12](#_Toc181893531)

[3.1.1. Giới thiệu chung 12](#_Toc181893532)

[3.1.2. Lịch sử hình thành và phát triển 12](#_Toc181893533)

[3.1.3. Đặc điểm của Selenium 12](#_Toc181893534)

[3.1.4. Các thành phần của Selenium 13](#_Toc181893535)

[3.2. Selenium Webdriver 14](#_Toc181893536)

[3.2.1. Đặc điểm của Selenium Webdriver 14](#_Toc181893537)

[3.2.2. Các tính năng nổi bật của Selenium WebDriver 15](#_Toc181893538)

[3.2.3. Ưu điểm và hạn chế của Selenium WebDriver 15](#_Toc181893539)

[CHƯƠNG 4: PHÂN TÍCH YÊU CẦU VÀ CHIẾN LƯỢC KIỂM THỬ 17](#_Toc181893540)

[4.1. Phân tích yêu cầu kiểm thử của hệ thống: 17](#_Toc181893541)

[4.2. Chiến lược kiểm thử: 17](#_Toc181893542)

[4.3. Thiết kế kịch bản kiểm thử (Test Case): 18](#_Toc181893543)

[CHƯƠNG 5: TRIỂN KHAI KIỂM THỬ 19](#_Toc181893544)

[5.1. Kiểm thử đơn vị 19](#_Toc181893545)

[5.1.1. Kiểm thử chức năng lọc và sắp xếp phòng 19](#_Toc181893546)

[5.2. Kiểm thử cơ sở dữ liệu 20](#_Toc181893547)

[5.2.1. Kiểm thử chức năng Login 20](#_Toc181893548)

[5.2.2. Kiểm thử chức năng thêm mới tài khoản nhân viên 23](#_Toc181893549)

[5.2.3. Kiểm thử chức năng thêm phòng 26](#_Toc181893550)

[5.3. Kiểm thử giao diện 30](#_Toc181893551)

[5.3.1. Kiểm thử chức năng checked in 30](#_Toc181893552)

[CHƯƠNG 6: ĐÁNH GIÁ VÀ SO SÁNH KIỂM THỬ THỦ CÔNG VÀ KIỂM THỬ TỰ ĐỘNG ………………………………………………………………………35](#_Toc181893553)

[6.1. So sánh kiểm thử thủ công và kiểm thử tự động: 35](#_Toc181893554)

[6.2. Đánh giá độ bao phủ kiểm thử: 35](#_Toc181893555)

[6.3. Hạn chế và bài học kinh nghiệm: 36](#_Toc181893556)

[KẾT LUẬN 37](#_Toc181893557)

[TÀI LIỆU THAM KHẢO 38](#_Toc181893558)

MỤC LỤC HÌNH ẢNH

[Hình 1. Sơ đồ tuần tự chức năng đăng nhập 4](#_Toc181873897)

[Hình 2. Sơ đồ tuần tự của chức năng sửa thông tin 4](#_Toc181873898)

[Hình 3. Vòng đời của quá trình kiểm thử 5](#_Toc181873899)

[Hình 4. Quy trình kiểm thử phần mềm 6](#_Toc181873900)

[Hình 5. Xác định ca kiểm thử với kiểm thử hộp trắng 7](#_Toc181873901)

[Hình 6. Minh hoạ kỹ thuật kiểm thử hộp đen 8](#_Toc181873902)

[Hình 7. Kiểm thử chức năng Login TC1 20](#_Toc181873903)

[Hình 8. Kiểm thử chức năng Login TC6 21](#_Toc181873904)

[Hình 9. Kiểm thử chức năng Thêm mới tài khoản nhân viên TC1 23](#_Toc181873905)

[Hình 10. Kiểm thử chức năng Thêm mới tài khoản nhân viên TC3 24](#_Toc181873906)

[Hình 11. Kiểm thử chức năng Thêm phòng TC5 28](#_Toc181873907)

[Hình 12. Kiểm thử chức năng Thêm phòng TC9 28](#_Toc181873908)

[Hình 13. Kiểm thử chức năng Checkin TC1 31](#_Toc181873909)

[Hình 14. Kiểm thử chức năng Checkin TC2 31](#_Toc181873910)

[Hình 15. Kiểm thử chức năng Checkin TC3 32](#_Toc181873911)

1. GIỚI THIỆU CHUNG
   1. ****Tổng quan đề tài****:

Trong kỷ nguyên số, các hệ thống quản lý khách sạn ngày càng trở nên phổ biến và đóng vai trò quan trọng trong việc tối ưu hóa quản lý dịch vụ khách hàng. Website quản lý khách sạn là công cụ hỗ trợ khách sạn quản lý các phòng, khách hàng, nhân viên, đồng thời cung cấp trải nghiệm trực tuyến tiện lợi cho người dùng.

Mục tiêu nghiên cứu: Nghiên cứu này nhằm đánh giá hiệu quả của việc kết hợp kiểm thử thủ công và kiểm thử tự động bằng Selenium trong quá trình kiểm thử một website quản lý khách sạn. Cụ thể, nghiên cứu tập trung vào đánh giá chất lượng của hệ thống thông qua các phương pháp kiểm thử khác nhau, từ đó đảm bảo rằng các chức năng chính của website như đặt phòng, các hoạt động check in, check out, và quản lý doanh thu của khách sạn hoạt động ổn định và chính xác. Nghiên cứu này sẽ cung cấp một cái nhìn tổng quan về quá trình kiểm thử một website quản lý khách sạn, từ đó giúp các nhà phát triển và người kiểm thử hiểu rõ hơn về tầm quan trọng của việc kết hợp kiểm thử thủ công và tự động. Đồng thời, nghiên cứu cũng sẽ đóng góp vào việc xây dựng một bộ phương pháp kiểm thử hiệu quả cho các hệ thống tương tự trong tương lai.

Phạm vi nghiên cứu: Phạm vi của nghiên cứu giới hạn ở các chức năng cốt lõi của hệ thống quản lý khách sạn, bao gồm đặt phòng, thanh toán và quản lý phòng. Nghiên cứu sẽ không đi sâu vào các yếu tố bảo mật hoặc tối ưu hóa hiệu suất hệ thống, mà chỉ tập trung vào các lỗi chức năng.

Đối tượng nghiên cứu: Đối tượng nghiên cứu là một hệ thống quản lý khách sạn do nhóm tự thiết kế với các chức năng chính được thiết kế để phục vụ quy trình quản lý phòng, doanh thu và nhân viên trong khách sạn.

* 1. ****Lý do chọn đề tài****:

Lựa chọn kiểm thử hệ thống quản lý khách sạn bằng Selenium kết hợp với kiểm thử thủ công xuất phát từ các lý do chính sau:

* Tính cấp thiết: Việc đảm bảo chất lượng của các hệ thống quản lý khách sạn là vô cùng quan trọng để mang lại trải nghiệm tốt nhất cho khách hàng và tăng tính cạnh tranh cho các doanh nghiệp. Bất kỳ lỗi nào trong hệ thống đều có thể dẫn đến sự cố trong quá trình đặt phòng, thanh toán hoặc quản lý phòng, gây ảnh hưởng trực tiếp đến doanh thu và uy tín của khách sạn. Do đó ứng dụng công nghệ tự động hóa trong kiểm thử phần mềm giúp tăng độ chính xác, giảm thời gian thực hiện và chi phí kiểm thử, đồng thời đảm bảo khả năng phát hiện lỗi kịp thời.
* Ưu thế của Selenium trong kiểm thử tự động: Selenium là công cụ kiểm thử tự động mạnh mẽ, được sử dụng rộng rãi trong các ứng dụng web. Khả năng hỗ trợ đa nền tảng và khả năng tích hợp linh hoạt của Selenium giúp đơn giản hóa quá trình kiểm thử cho các ứng dụng phức tạp như hệ thống quản lý khách sạn.
* Kết hợp phương pháp kiểm thử: Sự kết hợp giữa kiểm thử thủ công và kiểm thử tự động có thể mang lại hiệu quả cao hơn, giúp phát hiện lỗi một cách toàn diện. Trong trường hợp này, Selenium được sử dụng để kiểm thử tự động các thao tác lặp lại và tốn nhiều thời gian, trong khi kiểm thử thủ công được áp dụng cho những khía cạnh cần sự linh hoạt và sáng tạo hơn trong phát hiện lỗi.
  1. ****Mục tiêu và phạm vi****:
* Mục tiêu cụ thể: Nghiên cứu này hướng tới việc:

Xây dựng kịch bản kiểm thử chi tiết cho các chức năng chính của hệ thống quản lý khách sạn.

Thực hiện và đánh giá kết quả kiểm thử thông qua phương pháp thủ công và tự động.

So sánh hiệu quả giữa kiểm thử tự động và kiểm thử thủ công về các khía cạnh như độ bao phủ, thời gian thực hiện, và khả năng phát hiện lỗi.

* Phạm vi: Đề tài tập trung vào các kiểm thử chức năng chính của hệ thống như đặt phòng, thanh toán và quản lý phòng, không bao gồm các khía cạnh bảo mật hoặc hiệu năng.
  1. Phương pháp nghiên cứu

Để đạt được mục tiêu nghiên cứu, các phương pháp sau sẽ được áp dụng:

* Phương pháp thu thập dữ liệu: Tìm hiểu và phân tích tài liệu về hệ thống quản lý khách sạn, các chiến lược kiểm thử phần mềm, và công nghệ kiểm thử tự động Selenium.
* Phương pháp thực nghiệm: Triển khai các kịch bản kiểm thử trên hệ thống quản lý khách sạn cụ thể. Các kịch bản kiểm thử sẽ được thực hiện thông qua kiểm thử thủ công và tự động bằng Selenium WebDriver để thu thập dữ liệu thực tế.
* Phương pháp phân tích và so sánh: Phân tích kết quả kiểm thử để đánh giá hiệu quả của kiểm thử thủ công so với kiểm thử tự động, qua đó xác định ưu điểm và hạn chế của từng phương pháp trong việc kiểm thử hệ thống quản lý khách sạn.

1. TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ KHÁCH SẠN VÀ KIỂM THỬ PHẦN MỀM
   1. Tổng quan về hệ thống quản lý khách sạn:
      1. Giới thiệu hệ thống quản lý khách sạn

Hệ thống quản lý khách sạn là một giải pháp phần mềm giúp các khách sạn và khu nghỉ dưỡng tối ưu hóa các hoạt động quản lý, từ đặt phòng, lưu trữ thông tin khách hàng, đến quản lý phòng và thanh toán. Mục tiêu của hệ thống là cung cấp một công cụ giúp tự động hóa và hợp lý hóa các quy trình, nâng cao hiệu quả vận hành, và cung cấp trải nghiệm tốt hơn cho khách hàng.

* + 1. Lợi ích và thách thức

Lợi ích:

* Tăng cường hiệu quả quản lý: Giảm bớt các thao tác thủ công, tiết kiệm thời gian và nhân lực.
* Cải thiện trải nghiệm khách hàng: Cung cấp quy trình đặt phòng nhanh chóng và thuận tiện, cũng như quản lý yêu cầu khách hàng một cách chuyên nghiệp.
* Khả năng truy cập dữ liệu tức thời: Thông tin về phòng, khách hàng, và tình trạng đặt phòng luôn sẵn sàng, hỗ trợ việc ra quyết định nhanh chóng.

Thách thức:

* Đảm bảo an toàn dữ liệu: Hệ thống cần có các biện pháp bảo mật để bảo vệ dữ liệu khách hàng và ngăn chặn các truy cập trái phép.
* Đảm bảo khả năng mở rộng: Khả năng tích hợp với các hệ thống khác và xử lý tăng trưởng số lượng người dùng là yêu cầu quan trọng để đáp ứng nhu cầu kinh doanh mở rộng.
* Duy trì hiệu suất: Hệ thống cần duy trì hiệu suất cao khi xử lý nhiều yêu cầu đồng thời.
  1. Đặc tả hệ thống quản lý khách sạn
     1. Yêu cầu chức năng

Mô tả các chức năng chính mà hệ thống quản lý khách sạn cần thực hiện. Ví dụ:

Quản lý phòng: Thêm, xóa, chỉnh sửa thông tin phòng.

Quản lý khách hàng: Lưu trữ thông tin khách hàng, kiểm tra lịch sử đặt phòng.

Quản lý nhân viên: Theo dõi thông tin, ca làm việc và phân quyền.

Đặt phòng và thanh toán: Chức năng đặt phòng trực tuyến, quản lý thanh toán, và hủy đặt phòng.

* + 1. Yêu cầu phi chức năng

Các yêu cầu phi chức năng của hệ thống bao gồm:

* Hiệu suất: Hệ thống cần xử lý đồng thời một lượng lớn yêu cầu từ khách hàng và nhân viên mà không bị gián đoạn. Điều này đảm bảo thời gian phản hồi nhanh và khả năng xử lý đồng thời nhiều người dùng.
* Khả năng mở rộng: Hệ thống cần dễ dàng mở rộng để tích hợp thêm các dịch vụ mới hoặc tăng công suất khi lượng người dùng tăng. Điều này bao gồm khả năng mở rộng theo chiều ngang (thêm máy chủ) và chiều dọc (tăng công suất máy chủ hiện tại).
* Bảo mật: Hệ thống phải bảo vệ dữ liệu khách hàng và giao dịch tài chính, tuân thủ các tiêu chuẩn bảo mật như mã hóa dữ liệu và kiểm tra xác thực người dùng.
  + 1. Sơ đồ tuần tự

Sơ đồ tuần tự mô tả cách các thành phần trong hệ thống tương tác với nhau qua các thao tác tuần tự, giúp hình dung quá trình thực hiện các chức năng của hệ thống.

1. Sơ đồ tuần tự đăng nhập

Kịch bản: Người dùng muốn đăng nhập.

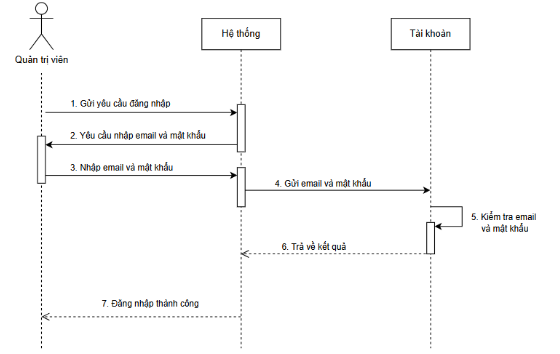
Người dùng nhập thông tin tài khoản vào hệ thống.

Hệ thống kiểm tra thông tin tài khoản với cơ sở dữ liệu

Cơ trở dữ liệu trả kết quả về hệ thống

Nếu tài khoản hợp lệ, hệ thống hiển thị thông báo lỗi. Nếu tài khoant hợp lệ chuyển đến trang chủ

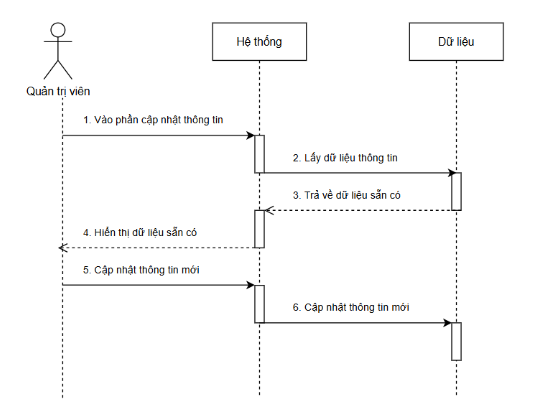
Sơ đồ tuần tự của chức năng đăng nhập:



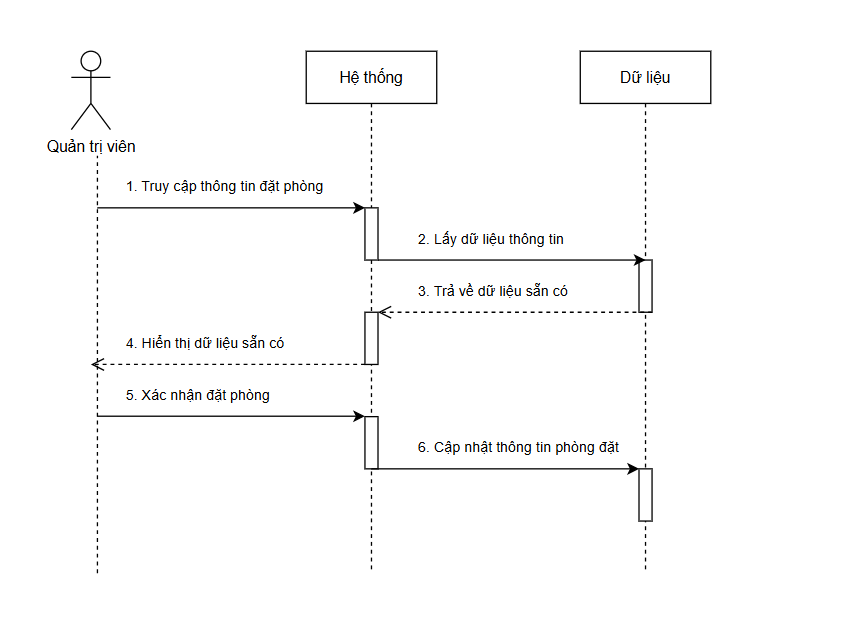
1. Sơ đồ tuần tự chức năng đăng nhập
2. Sơ đồ tuần tự của chức năng sửa thông tin:

Kịch bản: Người dùng muốn sửa tài khoản

* Người dùng yêu cầu sửa tài khoản
* Hệ thống gửi lại cho người dùng giao diện sửa, dữ liệu sẵn có của tài khoản
* Người dùng nhập thông tin sửa
* Hệ thống gửi yêu cầu xác nhận sửa
* Người dùng xác nhận sửa
* Hệ thống gửi thông tin tài khoản và thông tin sửa về cơ sở dữ liệu
* Cơ sở dữ liệu cập nhật



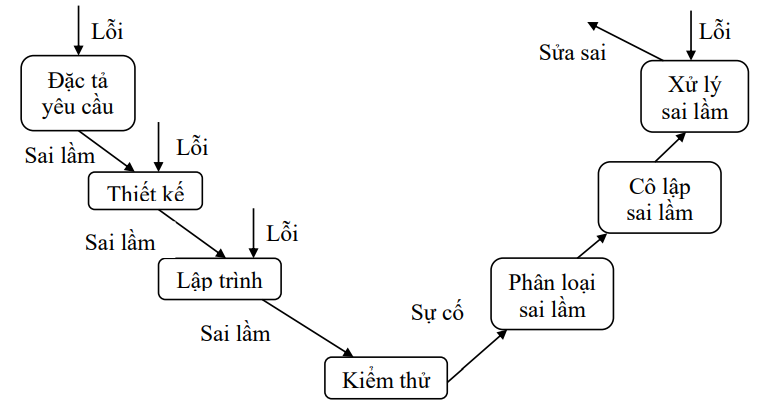
1. Sơ đồ tuần tự của chức năng sửa thông tin
2. Sơ đồ tuần tự của chức năng check in:



1. Sơ đồ tuần tự của chức năng check in
   1. Kiểm thử phần mềm:
      1. Khái niệm và vai trò của kiểm thử phần mềm

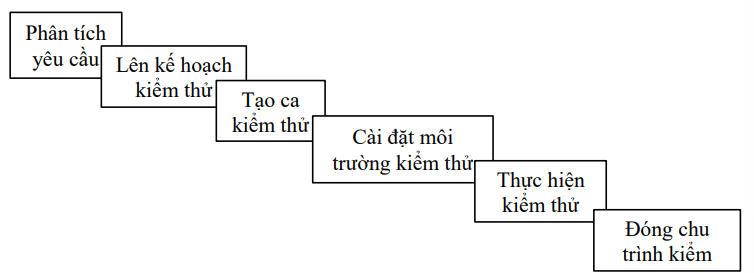
Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Hiểu theo cách đơn giản hơn, kiểm thử phần mềm là quá trình tìm thất bại hoặc chứng tỏ việc tiến hành của phần mềm là đúng đắn.

Kiểm thử phần mềm chiếm một vị trí quan trọng trong việc nâng cao chất lượng cũng như độ tin cậy của phần mềm trong quá trình phát triển. Hoàn thành vòng quay “đưa lỗi vào – tìm lỗi – khử lỗi đi” của quy trình kiểm thử phần mềm sẽ thu lại được những cải tiến đáng kể cho chất lượng sản phẩm phần mềm. Việc biết được sản phẩm phần mềm tốt tới mức nào trước khi đưa vào sử dụng sẽ hạn chế tối đa những rủi ro gặp phải trong quá trình phát triển phần mềm.



1. Vòng đời của quá trình kiểm thử
   * 1. Quy trình kiểm thử phần mềm

Kiểm thử phần mềm bao gồm nhiều giai đoạn với sự phối hợp của nhiều bên liên quan chứ không chỉ là một hoạt động đơn lẻ. Chính vì thế, cần có quy trình kiểm thử phần mềm để làm rõ các công đoạn, các bước kiểm thử, người chịu trách nhiệm và khi nào việc kiểm thử được tiến hành trong toàn bộ quy trình phát triển phần mềm. Nói cách khác, quy trình kiểm thử phần mềm chính là chuỗi các hoạt động được tiến hành để thực hiện việc kiểm thử. Các giai đoạn trong quy trình kiểm thử phần mềm được biểu diễn tổng quát bằng sơ đồ sau:



1. Quy trình kiểm thử phần mềm

Phân tích yêu cầu: Nhóm kiểm thử sẽ tương tác với các bên liên quan để hiểu rõ những yêu cầu cụ thể cần cho việc kiểm thử. Các yêu cầu có thể là chức năng (xác định phần mềm cần phải làm những gì) hoặc phi chức năng (hiệu năng, tính bảo mật hệ thống, màu sắc, v.v.)

Lên kế hoạch kiểm thử: Còn được gọi bằng tên khác là lên chiến lược thử nghiệm. Ở giai đoạn này, trưởng nhóm kiểm thử sẽ dự toán chi phí cho dự án cũng như chuẩn bị kế hoạch kiểm thử.

Tạo ca kiểm thử: Giai đoạn này cần phải tạo, xác minh, kiểm tra lại các ca kiểm thử. Dữ liệu kiểm thử cũng được tạo và xác định trong giai đoạn này.

Cài đặt môi trường kiểm thử: Môi trường kiểm thử quyết định bởi các điều kiện phần cứng và phần mềm trong từng dự án. Thiết lập môi trường kiểm thử có thể thực hiện song song với giai đoạn sinh ca kiểm thử và là một tiêu chí quan trọng trong quá trình kiểm thử. Tuy nhiên, nhóm kiểm thử có thể không cần tham gia vào giai đoạn này nếu đã có các bên liên quan khác hỗ trợ, nhiệm vụ của nhóm kiểm thử chỉ là yêu cầu môi trường kiểm thử cần thiết.

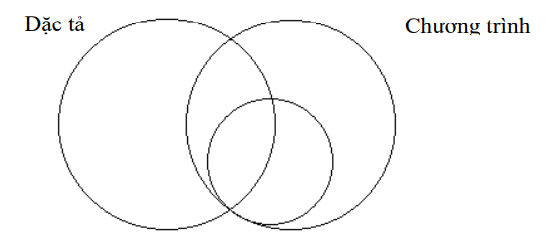
Thực hiện kiểm thử: Nhóm kiểm thử thực hiện kiểm thử theo kế hoạch và danh sách ca kiểm thử đã chuẩn bị từ giai đoạn trước. Các lỗi phát hiện ở giai đoạn này sẽ được thông báo lại cho nhóm phát triển phần mềm để chỉnh sửa và thực hiện kiểm thử lại.

Đóng chu trình kiểm thử: Nhóm kiểm thử sẽ họp, thảo luận và phân tích những bài học rút ra sau quá trình kiểm thử, đưa ra chiến lược cho những lần kiểm thử kế tiếp hoặc chia sẻ kinh nghiệm cho những dự án tương tự.

* + 1. Phân loại kiểm thử phần mềm

Có 2 cách cơ bản để xác định các ca kiểm thử là kiểm thử tĩnh và kiểm thử động.

* Kiểm thử tĩnh: là một hình thức của kiểm thử phần mềm mà không cần thực thi chương trình. Điều này ngược với thử nghiệm động. Công việc chủ yếu là kiểm tra tính đúng đắn của mã lệnh, thuật toán hay tài liệu. Đây là loại kiểm thử được thực hiện bởi lập trình viên. Lỗi được phát hiện bằng kiểm thử tĩnh ít tốn kém để sửa chữa hơn so với lỗi phát hiện bằng kiểm thử động sẽ được đề cập dưới đây. Các lập trình viên có thể trao đổi mã nguồn chéo nhau hoặc làm việc một cách độc lập để thực hiện kiểm thử tĩnh.
* Kiểm thử động: Liên quan đến việc thực thi chương trình để phát hiện các lỗi, thất bại có thể có của chương trình hay tìm ra các vấn đề về hiệu năng hệ thống. Việc thực thi chương trình trên tất cả các dữ liệu đầu vào là không thể nên ta chỉ có thể chọn một tập con các dữ liệu đầu vào để thực thi hay nói cách khác là sinh ra các ca kiểm thử. Trong kiểm thử động, người ta chia làm 2 kỹ thuật: kiểm thử hộp trắng (kiểm thử cấu trúc) và kiểm thử hộp đen (kiểm thử chức năng).
* Kiểm thử hộp trắng: là kỹ thuật kiểm thử dựa vào thuật toán, cấu trúc mã nguồn bên trong của chương trình với mục đích đảm bảo rằng tất cả các câu lệnh và điều kiện sẽ được thực hiện ít nhất một lần. Người kiểm thử truy cập vào mã nguồn chương trình và kiểm tra nó, lấy đó làm cơ sở để thực hiện việc kiểm thử. Kiểm thử hộp trắng bao gồm các công việc cơ bản: Kiểm thử đường dẫn, kiểm thử luồng điều khiển, kiểm thử nội bộ (xác nhận các tham số, vòng lặp), kiểm thử tính năng (kiểm tra thời gian xử lý, dữ liệu cụ thể). Tuy nhiên, việc kiểm thử hộp trắng tồn tại khá nhiều hạn chế như: không thể đảm bảo rằng chương trình đã tuân theo đặc tả, khó phát hiện được các lỗi do dữ liệu, thiếu đường dẫn, v.v. Như vậy, không thể chỉ sử dụng kiểm thử hộp trắng để kiểm thử chương trình.



1. Xác định ca kiểm thử với kiểm thử hộp trắng

* Kiểm thử hộp đen: Là kỹ thuật kiểm thử dựa trên đầu vào và đầu ra của chương trình mà không quan tâm tới mã nguồn bên trong được viết ra sao. Với kỹ thuật này, kiểm thử viên xem phần mềm như là một hộp đen. Để thực hiện, kiểm thử viên sẽ xây dựng các nhóm giá trị đầu vào sao cho chúng có thể thực hiện đầy đủ các chức năng cần có của chương trình. Kiểm thử hộp đen sử dụng các phương pháp: phân tích giá trị biên, kiểm thử tính bền vững, kiểm thử trường hợp xấu nhất, kiểm thử phân lớp tương đương miền dữ liệu đầu vào, đầu ra, kiểm thử giá trị đặc biệt, kiểm thử dựa trên bảng quyết định. Tất cả Đặc tả Chương trình Kiểm thử ứng dụng trên nền Web bằng công cụ Selenium Vũ Công Minh – CT1801 23 các phương pháp trên đều dựa trên thông tin xác định về các thành phần đang được kiểm thử.



1. Minh hoạ kỹ thuật kiểm thử hộp đen
   * 1. Kiểm thử tự động và vai trò của Selenium

Kiểm thử tự động là quá trình kiểm tra một hệ thống nào đó bằng các công cụ tự động hoá với dữ liệu đầu vào và đầu ra đã được xác định.

Công việc kiểm thử thường chiếm từ 11% - 40% chi phí cho quá trình phát triển phần mềm. Hơn nữa, các dự án phần mềm đều mong muốn giảm chi phí về thời gian, nhân lực mà vẫn đem lại hiệu quả cao, chất lượng tốt. Đó chính là lý do kiểm thử tự động được áp dụng rộng rãi trong các quy trình phát triển phần mềm ngày nay.

Kiểm thử tự động đặc biệt phát huy tác dụng trong các trường hợp kiểm thử lặp đi lặp lại, kiểm thử hồi quy hay các ca kiểm thử có giá trị dữ liệu đầu vào rất lớn khiến cho việc kiểm thử thủ công gặp nhiều khó khăn. Đối với các trường hợp kiểm thử lặp đi lặp lại, nếu thực hiện thủ công sẽ gây ra sự nhàm chán cho người kiểm thử, dẫn tới năng suất lao động kém. Đó là chưa kể tới việc lặp đi lặp lại quy trình một cách thủ công hoàn toàn có thể dẫn tới sai sót. Ngược lại, nếu thay bằng kiểm thử tự động, dù có lặp đi lặp lai bao nhiêu lần thì cũng cho ra thao tác và kết quả chính xác. Điều này giúp chúng ta tránh được những rủi ro không đáng có và giảm đáng kể thời gian cho việc kiểm thử.

Selenium là một trong những công cụ kiểm thử tự động phổ biến nhất hiện nay, đặc biệt trong kiểm thử các ứng dụng Web. Selenium - công cụ kiểm thử phần mềm mã nguồn mở hỗ trợ kiểm thử tự động cho các ứng dụng Web. Selenium cung cấp chức năng ghi tự động và phát lại, hỗ trợ hữu ích cho kiểm thử hồi quy. Điểm mạnh của Selenium là hỗ trợ nhiều nền tảng khác nhau, tích hợp vào các trình duyệt phổ biến hiện nay, có thể thực hiện nhiều ca kiểm thử cùng lúc, có khả năng lưu các ca kiểm thử để sử dụng lại khi cần và cho phép người dùng chèn chú thích ở giữa kịch bản kiểm thử để hiểu rõ hơn nội dung kiểm thử. Selenium cũng hỗ trợ một lượng lớn các ngôn ngữ lập trình Web phổ biến hiện nay như C#, Java, Perl, PHP, Python, Ruby, v.v. Selenium có thể kết hợp với một số công cụ khác như Bromien và Junit nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm các công cụ hỗ trợ. Selenium hiện nay đang được cộng đồng đánh giá là một trong những công cụ tốt nhất cho kiểm thử tự động các ứng dụng Web.

1. TỔNG QUAN VỀ CÔNG NGHỆ KIỂM THỬ SELENIUM VÀ SELENIUM WEBDRIVER
   1. Khái quát về Selenium
      1. Giới thiệu chung

Selenium là một công cụ hỗ trợ kiểm thử tự động cho các ứng dụng Web. Selenium hỗ trợ kiểm thử trên hầu hết các trình duyệt phổ biến hiện nay như Firefox, Internet Explorer, Safari, ... cũng như các hệ điều hành chủ yếu như Windows, Linux, Mac,... Selenium cũng hỗ trợ một số lớn các ngôn ngữ lập trình Web phổ biến hiện nay như C#, Java, Perl, PHP, Python, Ruby,... Công cụ này có thể kết hợp thêm với một số công cụ khác như Junit và TestNG nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm các công cụ bổ trợ.

* + 1. Lịch sử hình thành và phát triển

Selenium bắt đầu vào năm 2004 bởi Jason Huggins khi làm việc tại ThoughtWorks. Ban đầu, nó là một công cụ đơn giản để kiểm thử nội bộ cho các ứng dụng web của công ty, sau đó nhanh chóng phát triển nhờ vào tính linh hoạt và khả năng mở rộng. Một số cột mốc quan trọng trong quá trình phát triển của Selenium gồm:

* Selenium RC (Remote Control): Phiên bản đầu tiên của Selenium cho phép tự động hóa trình duyệt từ xa, giúp thực hiện các thao tác từ nhiều ngôn ngữ lập trình khác nhau.
* Selenium WebDriver: Ra mắt năm 2009, WebDriver cải tiến mạnh mẽ Selenium RC bằng cách tương tác trực tiếp với trình duyệt, đem lại hiệu suất cao và độ ổn định tốt hơn.
* Selenium 2.0: Đây là phiên bản hợp nhất Selenium RC và WebDriver, tạo nền tảng vững chắc cho Selenium như hiện nay.
* Selenium 3 và 4: Phiên bản 3 giới thiệu nhiều tính năng nâng cao và chuẩn hóa WebDriver. Phiên bản 4 (ra mắt vào năm 2021) nâng cấp thêm giao diện người dùng (Selenium IDE) và bổ sung hỗ trợ các tính năng mới như xử lý thao tác trên trình duyệt và cải tiến API.
  + 1. Đặc điểm của Selenium
* Mã nguồn mở: Đây là điểm mạnh nhất của Selenium khi so sánh với các test tool khác. Vì là mã nguồn mở nên chúng ta có thể sử dụng mà không phải lo lắng về phí bản quyền hay thời hạn sử dụng.
* Cộng đồng hỗ trợ: vì là mã nguồn mở nên Selenium có một cộng đồng hỗ trợ khá mạnh mẽ. Bên cạnh đó, Google là nơi phát triển Selenium nên chúng ta hoàn toàn có thể yên tâm về sự hỗ trợ miễn phí khi có vấn đề về Selenium. Tuy nhiên, đây cũng là một điểm yếu của Selenium. Vì công cụ này hoàn toàn miễn phí, cộng đồng lại đông nên một vấn đề có thể nhiều giải pháp, và có thể một số giải pháp là không hữu ích. Mặc khác, chúng ta không thể hối thúc hay ra deadline cho sự hỗ trợ.
* Selenium hỗ trợ nhiều ngôn ngữ lập trình.
* Selenium hỗ trợ chạy trên nhiều hệ điều hành khác nhau với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của người dùng.
* Chạy test case ở background. Khi chúng ta thực thi một test script, chúng ta hoàn toàn có thể làm việc khác trên cùng một máy tính. Điều này hỗ trợ chúng ta không cần tốn quá nhiều tài nguyên máy móc khi chạy test script.
* Không hỗ trợ Win app. Selenium thực sự chỉ hỗ trợ chúng ta tương tác với Browser mà không hỗ trợ chúng ta làm việc với các Win app, kể cả Win dialog như Download/Upload. Vậy nên, để xử lý các trường hợp cần tương tác với hệ thống hay một app thứ ba, chúng ta cần một hay nhiều thư viện khác như AutoIt hay Coded UI.

Là một công cụ hỗ trợ kiểm tra tính năng nên Selenium không có khả năng giả lập nhiều người dùng ảo cùng một lúc. Công việc của nó là chạy kiểm thử tự động dựa trên một kịch bản đã được thiết kế từ trước. Qua đó chúng ta có thể chắc chắn rằng đối tượng kiểm thử có hoạt động đúng như mong đợi hay không.

* + 1. Các thành phần của Selenium

Selenium là một bộ công cụ hỗ trợ kiểm thử tự động các tính năng của ứng dụng trên nền Web, bao gồm 4 thành phần: Selenium IDE, Selenium Grid, Selenium 1.0 ( hay Selenium Remote Control - Selenium RC) và Selenium 2.0 (hay Selenium WebDriver). Mỗi loại có một vai trò cụ thể trong việc hỗ trợ sự phát triển của tự động hóa kiểm thử ứng dụng web.

* Selenium IDE (Integrated Development Environment)

Selenium IDE là được phát triển dưới hình thức add-on của Firefox. Chúng ta chỉ có thể Record trên trình duyệt FireFox, nhưng bù lại, chúng ta có thể Playback trên các trình duyệt khác như là IE, Chrome....

Selenium có thể sinh code tự động hoặc nạp các đoạn mã viết tay. Công cụ này cung cấp chức năng “thu và chạy lại” – Record and Playback. Sau đó chạy lại các câu lệnh này để kiểm thử. Chức năng này rất hữu dụng giúp tiết kiệm thời gian viết kịch bản kiểm thử. Selenium IDE còn cho phép lưu kịch bản đã thu dưới nhiều loại ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby....

Selenium Core: Đã được tích hợp trong Selenium IDE. Selenium Core là một công cụ chạy các test script viết bằng Selenese. Thế mạnh của công cụ này là có thể chạy test script trên gần như tất cả các trình duyệt, nhưng lại yêu cầu được cài đặt trên máy chủ của website cần kiểm tra. Điều này là không thể khi Tester không có quyền truy cập đến máy chủ đó.

* Selenium RC (Remote Control)

Selenium Remote Control là một framework kiểm thử cho phép thực hiện nhiều hơn và tuyến tính các hành động trên trình duyệt. Nó cho phép cho phép các nhà phát triển tự động hóa kiểm thử sử dụng một ngôn ngữ lập trình cho tính linh hoạt tối đa và mở rộng trong việc phát triển logic thử nghiệm.

Công cụ này có thể nhận các test script được thu bởi Selenium IDE, cho phép chỉnh sửa, cải tiến linh động bằng nhiều ngôn ngữ lập trình khác nhau. Sau đó khởi động một trong các trình duyệt Web được chỉ định để thực thi kiểm thử trực tiếp trên trình duyệt đó. Selenium RC còn cung cấp khả năng lưu lại kết quả kiểm thử; cung cấp một API (Application Programming Interface) và thư viện cho mỗi ngôn ngữ được hỗ trợ: HTML, Java, C#, Perl, PHP, Python và Ruby. Khả năng sử dụng Selenium RC với một ngôn ngữ lập trình bậc cao để phát triển các trường hợp kiểm thử cũng cho phép kiểm thử tự động được tích hợp với một dự án xây dựng môi trường tự động.

* Selenium WebDriver

Selenium WebDriver là sự kế thừa từ Selenium Remote Control, làm việc trực tiếp với trình duyệt ở mức hệ điều hành, cho phép gửi lệnh trực tiếp đến trình duyệt và xuất ra kết quả

* Selenium-Grid

Selenium – Grid Là một hệ thống hỗ trợ người dùng thực thi test script trên nhiều trình duyệt một cách song song mà không cần phải chỉnh sửa test script.

Thực hiện phương pháp kiểm tra phân bổ, phối hợp nhiều Selenium RC để có thể thực thi trên nhiều trình duyệt Web khác nhau trong cùng một lúc nhằm giảm thiểu thời gian thực hiện.

* 1. Selenium Webdriver
     1. Đặc điểm của Selenium Webdriver

Kiến trúc hướng trực tiếp (Direct Browser Automation): Thay vì chạy qua máy chủ trung gian như Selenium RC, WebDriver tương tác trực tiếp với trình duyệt bằng cách sử dụng các API nội tại của trình duyệt. Điều này giúp WebDriver có tốc độ nhanh hơn và độ ổn định cao hơn.

Hỗ trợ nhiều trình duyệt: Selenium WebDriver tương thích với nhiều trình duyệt phổ biến như Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, và Opera. Điều này giúp kiểm thử trên các môi trường khác nhau và đảm bảo tính nhất quán cho ứng dụng web.

Đa nền tảng: WebDriver chạy trên nhiều hệ điều hành, bao gồm Windows, macOS, và Linux, giúp các đội phát triển dễ dàng thực hiện kiểm thử trên các môi trường khác nhau.

Đa ngôn ngữ lập trình: Selenium WebDriver hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, C#, Ruby, và JavaScript. Điều này cho phép các lập trình viên có thể viết kịch bản kiểm thử trong ngôn ngữ mà họ thành thạo.

* + 1. Các tính năng nổi bật của Selenium WebDriver

Hỗ trợ tương tác phong phú với trang web: WebDriver cho phép mô phỏng hầu hết các thao tác của người dùng như click chuột, nhập văn bản, chọn dropdown, di chuyển con trỏ chuột, kéo thả, và cuộn trang.

Quản lý và chờ đợi trạng thái phần tử (Waits): WebDriver có cơ chế chờ đợi cả Implicit Waits (chờ ngầm định) và Explicit Waits (chờ tường minh), giúp đảm bảo các phần tử được tải đầy đủ trước khi thực hiện các thao tác. Điều này rất hữu ích khi kiểm thử các ứng dụng web có nội dung động hoặc thời gian tải không đồng nhất.

Hỗ trợ đa tab và đa cửa sổ: WebDriver cho phép điều khiển các tab hoặc cửa sổ mới, giúp tester có thể kiểm thử các chức năng cần mở liên kết trong tab hoặc cửa sổ khác.

Xử lý JavaScript và Pop-up: WebDriver có thể tương tác với các đoạn mã JavaScript và xử lý các cửa sổ pop-up, thông báo, hoặc hộp thoại xác nhận (alert). Điều này giúp mô phỏng hoàn chỉnh hơn các thao tác người dùng thực tế.

Tích hợp với các framework kiểm thử: Selenium WebDriver có thể dễ dàng tích hợp với các framework kiểm thử như TestNG, JUnit (Java), hoặc PyTest (Python) để tổ chức các kịch bản kiểm thử, thực hiện kiểm thử song song, và tạo báo cáo kết quả kiểm thử.

* + 1. Ưu điểm và hạn chế của Selenium WebDriver

Ưu điểm của Selenium WebDriver:

* Hiệu suất cao và đáng tin cậy: Nhờ vào kiến trúc trực tiếp, WebDriver có khả năng thực thi nhanh và ổn định hơn so với Selenium RC.
* Mở rộng và linh hoạt: Có thể mở rộng dễ dàng bằng cách kết hợp với các công cụ khác và framework kiểm thử, hoặc tích hợp với CI/CD.
* Phù hợp với môi trường Agile và DevOps: Tốc độ và tính linh hoạt của Selenium WebDriver rất phù hợp để tích hợp vào quy trình Agile và DevOps, giúp cải thiện chất lượng phần mềm liên tục.

Hạn chế của Selenium WebDriver:

* Chỉ hỗ trợ ứng dụng web: WebDriver không thể kiểm thử ứng dụng desktop hoặc ứng dụng di động mà chỉ dành cho ứng dụng web.
* Khó khăn trong việc kiểm thử hình ảnh: WebDriver không hỗ trợ tốt kiểm thử hình ảnh hoặc xác minh giao diện.
* Headless Browser Testing: WebDriver hỗ trợ chế độ headless, tức là kiểm thử không cần mở giao diện trình duyệt. Điều này giúp giảm tải hệ thống và tăng tốc độ kiểm thử, đặc biệt hữu ích trong các môi trường CI/CD.

1. PHÂN TÍCH YÊU CẦU VÀ CHIẾN LƯỢC KIỂM THỬ
   1. ****Phân tích yêu cầu kiểm thử của hệ thống****:

Yêu cầu chức năng:

* Chức năng đăng nhập hệ thống: Kiểm tra khả năng xác thực người dùng bằng tên đăng nhập và mật khẩu. Mục tiêu là đảm bảo người dùng chỉ có thể truy cập hệ thống khi cung cấp thông tin chính xác.
* Chức năng thêm, sửa, xóa phòng: Kiểm thử việc quản lý thông tin phòng, bao gồm việc thêm mới, sửa đổi hoặc xóa thông tin phòng.
* Chức năng checked in: Kiểm tra khả năng nhận phòng của khách, bao gồm việc cập nhật trạng thái phòng và thông tin khách. Cần kiểm tra liệu hệ thống có lưu lại chính xác thông tin nhận phòng.
* Chức năng checked out: Kiểm tra chức năng trả phòng, bao gồm việc xử lý thanh toán và cập nhật trạng thái phòng. Đảm bảo rằng việc tính toán phí phòng và thanh toán diễn ra chính xác.
* Chức năng thêm, sửa tài khoản nhân viên: Kiểm thử việc quản lý tài khoản nhân viên, bao gồm việc thêm mới, sửa đổi tài khoản. Đảm bảo rằng quyền truy cập của từng loại tài khoản được thiết lập chính xác.

Yêu cầu phi chức năng:

* Tính bảo mật: Kiểm thử các biện pháp bảo mật của hệ thống, bảo vệ thông tin người dùng và các giao dịch tài chính, bao gồm kiểm tra mã hóa dữ liệu và xác thực người dùng.
* Hiệu suất: Đánh giá khả năng xử lý đồng thời nhiều yêu cầu từ khách hàng và nhân viên mà không gây gián đoạn dịch vụ.
* Khả năng sử dụng: Kiểm tra tính dễ sử dụng của giao diện người dùng, đảm bảo người dùng có thể thao tác nhanh chóng và hiệu quả.
  1. Chiến lược kiểm thử:

Phương pháp chọn lọc giữa kiểm thử thủ công và kiểm thử tự động cho từng chức năng:

* Kiểm thử thủ công: Được áp dụng cho các chức năng cần tương tác trực tiếp với người dùng, như kiểm thử giao diện người dùng, các trường hợp đặc biệt hoặc các tình huống mà tự động hóa không hiệu quả.
* Kiểm thử tự động: Sử dụng cho các chức năng lặp đi lặp lại như kiểm thử hồi quy, kiểm thử các ca kiểm thử có dữ liệu đầu vào lớn hoặc các chức năng không thay đổi thường xuyên.

Định hướng kiểm thử:

* Kiểm thử hồi quy: Đảm bảo rằng các tính năng cũ của hệ thống vẫn hoạt động chính xác sau khi thực hiện thay đổi hoặc cập nhật.
* Kiểm thử chức năng: Xác minh các chức năng hệ thống có thực hiện đúng như yêu cầu ban đầu.
* Kiểm thử giao diện: Đánh giá khả năng tương tác của người dùng với hệ thống, đảm bảo các giao diện người dùng hoạt động đúng và dễ sử dụng, các chức năng cần thiết có thể truy cập dễ dàng.
  1. Thiết kế kịch bản kiểm thử (Test Case):

Quy trình thiết kế test case:

Test case sẽ được thiết kế để đảm bảo kiểm thử đầy đủ các chức năng hệ thống. Quá trình thiết kế bao gồm:

* Xác định điều kiện kiểm thử: Các điều kiện, yêu cầu chức năng cần kiểm tra.
* Dữ liệu đầu vào: Xác định dữ liệu đầu vào cho từng tình huống kiểm thử.
* Kịch bản kiểm thử: Thiết lập các bước cần thực hiện để kiểm tra tính năng.
* Kết quả mong đợi: Xác định kết quả dự kiến từ hệ thống.

Kịch bản kiểm thử cho các chức năng:

* Đăng nhập: Kiểm tra với tài khoản hợp lệ, tài khoản không tồn tại, mật khẩu sai, khóa tài khoản, quên mật khẩu.
* Thêm, sửa, xóa phòng: Kiểm tra các trường bắt buộc, tính hợp lệ của dữ liệu, kiểm tra tính toàn vẹn dữ liệu sau khi thực hiện các thao tác, kiểm tra khả năng thêm phòng loại mới.
* Checked in/out: Kiểm tra quá trình nhận phòng, trả phòng, cập nhật trạng thái phòng, tính toán phí phòng, in hóa đơn, xuất hóa đơn điện tử.
* Thêm, sửa tài khoản nhân viên: Kiểm tra quyền truy cập của từng loại tài khoản, kiểm tra tính toàn vẹn dữ liệu, kiểm tra khả năng thay đổi mật khẩu.

1. TRIỂN KHAI KIỂM THỬ
   1. Kiểm thử đơn vị
      1. Kiểm thử chức năng lọc và sắp xếp phòng

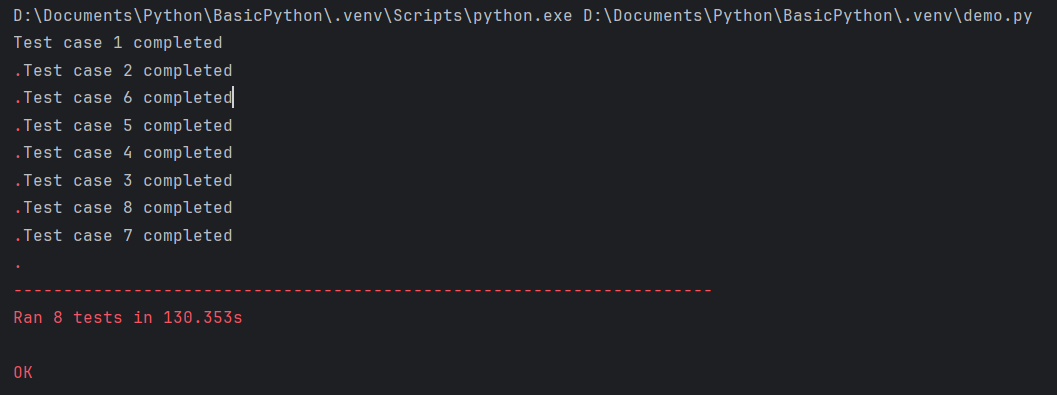
Mô tả: Tại trang Cabins, user có thể chọn lọc theo các trạng thái như: With discount, No discount và theo giá tiền phòng tăng dần và ngược lại.

* Đầu vào: Danh sách các bookings
* Đầu ra: Sắp xếp và lọc danh sách bookings theo trạng thái đã chọn

Bảng quyết định:

| Input |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| With discount | T | T | T | F | F | F | F | F |
| No discount | F | F | F | T | T | T | F | F |
| Increase price | F | T | F | F | T | F | T | F |
| Decrease price | F | F | T | F | F | T | F | T |
| Output | Chỉ hiển thị danh sách bookings với trạng thái With discount | Hiển thị danh sách bookings với trạng thái With discount và sắp xếp giá tăng dần | Hiển thị danh sách bookings với trạng thái With discount và sắp xếp giá giảm dần | Chỉ hiển thị danh sách bookings với trạng thái No discount | Hiển thị danh sách bookings với trạng thái No discount và sắp xếp giá tăng dần | Hiển thị danh sách bookings với trạng thái With discount và sắp xếp giá giảm dần | Hiển thị tất cả các sản phẩm và sắp xếp giá tăng dần | Hiển thị tất cả các sản phẩm và sắp xếp giá giảm dần |

Các ca kiểm thử:



|  |  |  |  |
| --- | --- | --- | --- |
| ID | Mô tả | Expected Output | Result Output |
| TC1 | Lọc sản phẩm No discount | Chỉ hiển thị sản phẩm không có giảm giá | PASS |
| TC2 | Lọc sản phẩm With discount | Chỉ hiển thị sản phẩm có giảm giá | PASS |
| TC3 | Sắp xếp sản phẩm theo giá tăng dần | Giá sản phẩm được sắp xếp từ thấp đến cao | PASS |
| TC4 | Sắp xếp sản phẩm theo giá giảm dần | Giá sản phẩm được sắp xếp từ cao đến thấp | PASS |
| TC5 | Lọc sản phẩm không có giảm giá và sắp xếp theo giá tăng dần | Chỉ hiển thị sản phẩm không có giảm giá và giá được sắp xếp từ thấp đến cao | PASS |
| TC6 | Lọc sản phẩm không có giảm giá và sắp xếp theo giá giảm dần | Chỉ hiển thị sản phẩm không có giảm giá và giá được sắp xếp từ cao đến thấp | PASS |
| TC7 | Lọc sản phẩm có giảm giá và sắp xếp theo giá giảm dần | Chỉ hiển thị sản phẩm có giảm giá và được sắp xếp từ cao đến thấp | PASS |
| TC8 | Lọc sản phẩm có giảm giá và sắp xếp theo giá tăng dần | Chỉ hiển thị sản phẩm có giảm giá và được sắp xếp từ thấp đến cao | PASS |

* 1. Kiểm thử cơ sở dữ liệu
     1. Kiểm thử chức năng Login

Mô tả: Tại màn hình Login có 2 field: “Email” và “Password”. Chỉ login thành công nếu nhập đúng cả 2 field “Email” và “Password”. Các trường hợp còn lại được hệ thống hiển thị thông báo “Provided email or password is incorrect”

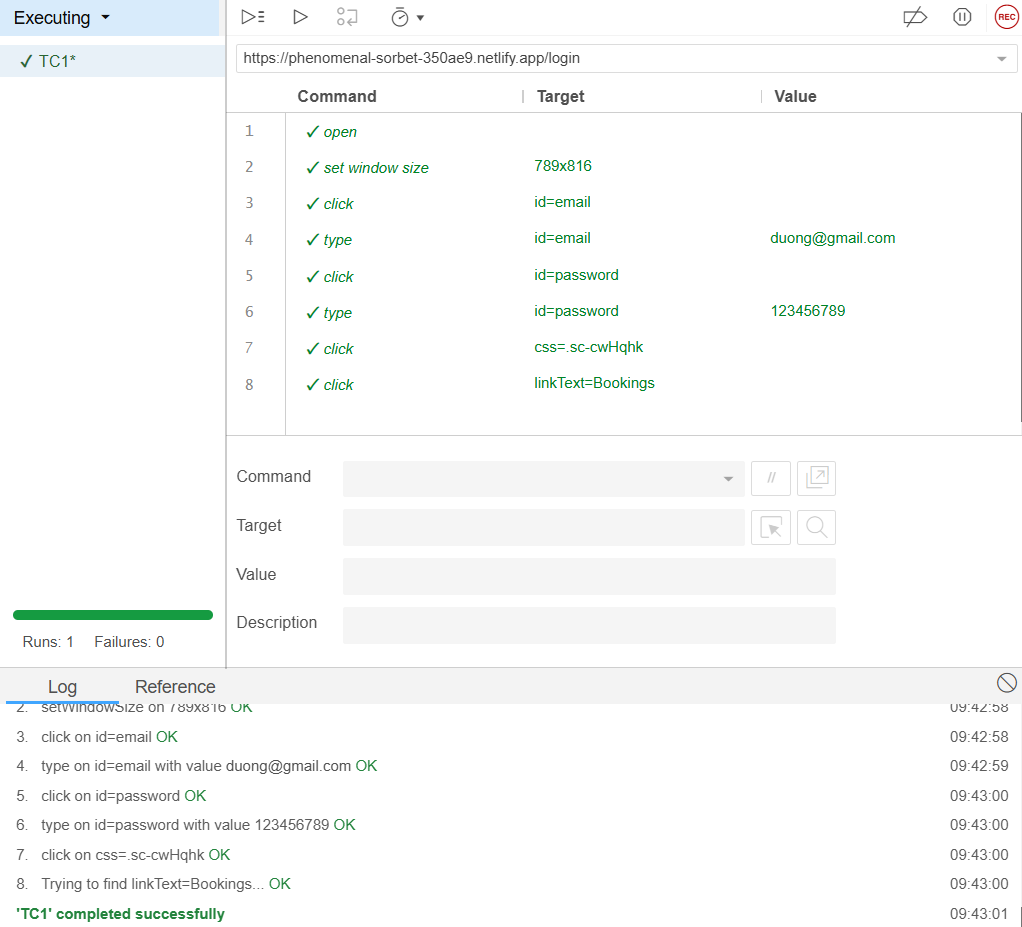
* Đầu vào: Email, Password (Mỗi giá trị đầu vào sẽ có 2 dạng: T (True –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Giá trị 1** | **Giá trị 2** | **Giá trị 3** | **Giá trị 4** |
| User Name | T | T | F | F |
| Password | T | F | T | F |
| Output | Login thành công | Login không thành công | Login không thành công | Login không thành công |

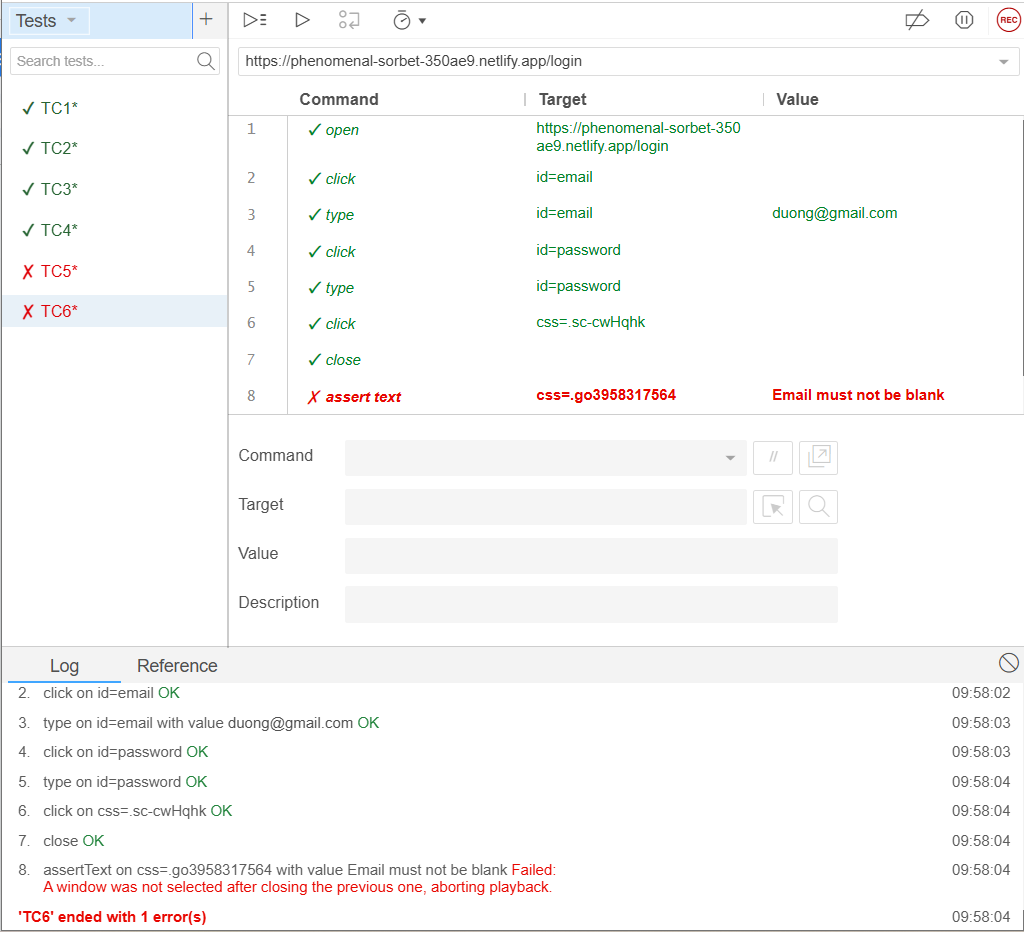
Các ca kiểm thử:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Email | Password | Expects Output | Result Outputs |
| TC1 | duong@gmail.com | 123456789 | Login successfully | PASS |
| TC2 | duong@gmail.com | 1111111111 | Login fail và in ra “Provided email or password is incorrect” | PASS |
| TC3 | duong@abc.com | 123456789 | Login fail và in ra “Provided email or password is incorrect” | PASS |
| TC4 | duong@abc.com | 1111111111 | Login fail và in ra “Provided email or password is incorrect” | PASS |
| TC5 |  | 123456789 | “Email must not be blank” | FAIL |
| TC6 | duong@gmail.com |  | “Password must not be blank” | FAIL |
| TC7 | nhập nguyên dấu cách | nhập nguyên dấu cách | “Password must not be blank” | FAIL |

Thực hiện kiểm thử trên Selenium cho các kết quả:



1. Kiểm thử chức năng Login TC1



1. Kiểm thử chức năng Login TC6
   * 1. Kiểm thử chức năng thêm mới tài khoản nhân viên

Các trường bắt buộc nhập không được bỏ trống

* Password: Tối thiểu 8 ký tự (>= 8 KT)
* passwordConfirm : Phải giống với password
* Email phải đúng định dạng xxx@xxx.xxx   không  chứa kí tự đặc biệt và là duy nhất

1. Phân vùng tương đương:

|  |  |  |
| --- | --- | --- |
|  | Valid | Invalid |
| Full Name |  | Trường để trống |
| Email | Đúng định dạng (xxx@xxx.xxx), không chứa ký tự đặc biệt, duy nhất. | Sai định dạng (thiếu ký tự "@", không có đuôi miền).  Chứa ký tự đặc biệt ngoài dấu ".".  Không duy nhất (đã tồn tại trong hệ thống). |
| Password | Độ dài tối thiểu 8 ký tự | Dưới 8 ký tự. |
| Password Confirm | Trùng với Password. | Không trùng với Password. |

1. Xác định các giá trị Biên cho kiểm thử

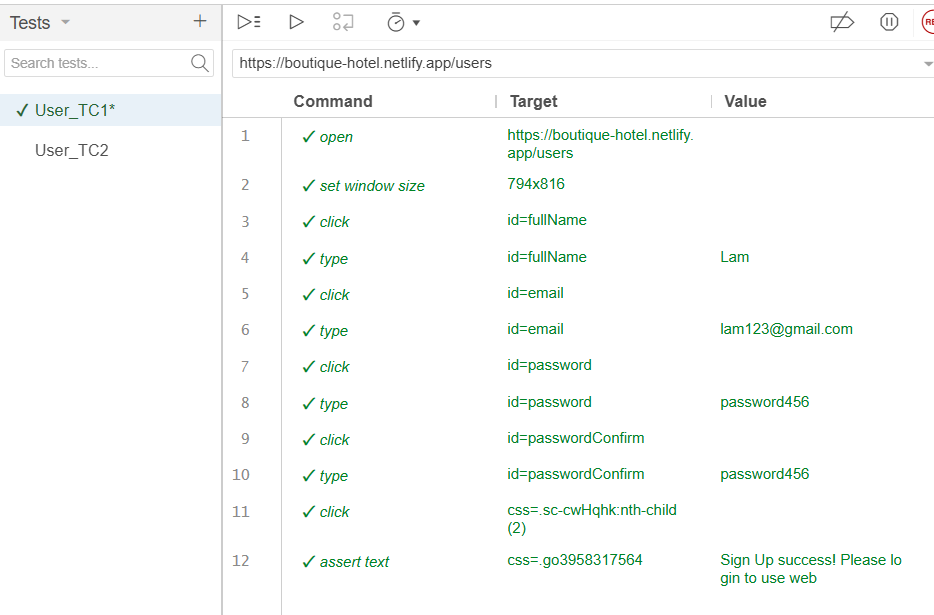
Với các biên của các trường:

* Password: Biên là 7 ký tự (không hợp lệ) và 8 ký tự (hợp lệ).
* Email: Kiểm thử các biên:
* Đúng định dạng (xxx@xxx.xxx) hoặc không đúng định dạng.
* Email đã tồn tại trong hệ thống hoặc chưa tồn tại.
* Password Confirm: Khác với Password hoặc trùng với Password.

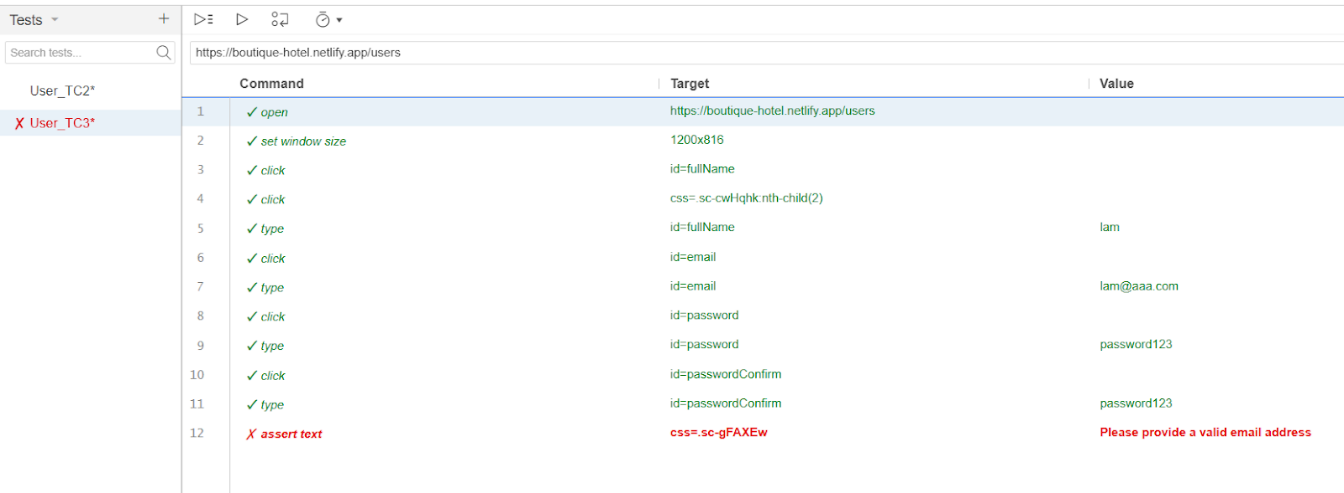
1. Các ca kiểm thử:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Username | Email | Password | Password Confirm | Mô tả | Kết quả mong đợi | Result Output |
| TC1 | Lam | lam2003@gmail.com | password456 | password456 | Tất cả các đầu vào hợp lệ. | Thêm tài khoản thành công | PASS |
| TC2 | lam | lam@.com | password123 | password123 | Email không đúng định dạng. | “Please provide a valid email address” | PASS |
| TC3 | lam | lam@aaa.com | password123 | password123 | Email không đúng định dạng | “Please provide a valid email address” | FAIL |
| TC4 | lam | lam@gmail.com | pass | pass | Password có độ dài dưới 8 ký tự. | “Password needs a minium of 8 characters” | PASS |
| TC5 | lam | lam@gmail.com | password123 | pass123 | Password Confirm không trùng với Password. | “Passwords need to match” | PASS |
| TC6 | lam | lam@.com | pass | pass12 | Email không hợp lệ và Password có độ dài dưới 8 ký tự và Password Confirm không trùng | Thêm tài khoản thất bại | PASS |
| TC7 |  |  |  |  | Để trống tất cả các trường | Hiện “This field is required” | PASS |
| TC8 |  |  |  |  | Nhập dấu cách vào các trường  (Blank space) | Hiện “This field is required” | FAIL |
| TC9 | lam | lam2003@gmail.com | password456 | password456 | Tất cả đầu vào hợp lệ nhưng Email đã tồn tại | Hiện “This email already exists” | FAIL |

Thực hiện kiểm thử trên Selenium cho các kết quả:



1. Kiểm thử chức năng Thêm mới tài khoản nhân viên TC1



1. Kiểm thử chức năng Thêm mới tài khoản nhân viên TC3
   * 1. Kiểm thử chức năng thêm phòng

Các trường bắt buộc nhập không được bỏ trống

* maxCapacity: Tối thiểu là 1 (>=1)
* regularPrice: Tối thiểu là 50$ (>= 50$)
* discount: >= 0, discount < regularPrice

Đầu vào: Cabin Name, maxCapacity, regularPrice, discount (Mỗi giá trị đầu vào sẽ có 2 dạng là T-True hoặc F-False)

Đầu ra: Thêm phòng mới thành công hoặc thất bại

1. Bảng quyết định

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Giá trị 1** | **Giá trị 2** | **Giá trị 3** | **Giá trị  4** | **Giá trị 5** | **Giá trị 6** | **Giá trị 7** | **Giá trị 8** |
| maxCapacity | T | T | T | T | F | F | F | F |
| regularPrice | T | T | F | F | T | F | T | F |
| discount | T | F | T | F | T | T | F | F |
| Output | Thêm phòng thành công | Thêm phòng không thành công | Thêm phòng không thành công | Thêm phòng không thành công | Thêm phòng không thành công | Thêm phòng không thành công | Thêm phòng không thành công | Thêm phòng không thành công |

1. Phân vùng tương đương

|  |  |  |
| --- | --- | --- |
|  | Valid | Invalid |
| maxCapactity: | Là số và >=1. | Không phải là số(ví dụ: ký tự chữ hoặc ký tự đặc biệt).  Nếu là số < 1 |
| regularPrice | Là số và >= 50. | Không phải là số  Nếu là số < 50 |
| discount | Phải là số, lớn hơn hoặc bằng 0, và nhỏ hơn hoặc bằng regularPrice. | Không phải là số.  Là số nhỏ hơn 0 hoặc bằng hoặc lớn hơn regularPrice. |

1. Xác định các giá trị Biên cho kiểm thử

**maxCapacity**

* Điều kiện: maxCapacity phải là số và ≥ 1.
* Giá trị biên:
* Biên dưới: 0 (giá trị không hợp lệ, nhỏ hơn 1).
* Biên hợp lệ: 1 (giá trị nhỏ nhất hợp lệ).
* Giá trị lớn hơn biên hợp lệ: 2 (để kiểm tra giá trị lớn hơn biên hợp lệ).
* Ký tự: Giá trị không hợp lệ (không phải là số).
* Các giá trị kiểm thử: 0, 1, 2,e.

**regularPrice**

* Điều kiện: regularPrice phải là số và ≥ 50.
* Giá trị biên:
* Biên dưới: 49 (giá trị không hợp lệ, nhỏ hơn 50).
* Biên hợp lệ: 50 (giá trị nhỏ nhất hợp lệ).
* Giá trị lớn hơn biên hợp lệ: 51 (để kiểm tra giá trị lớn hơn biên hợp lệ).
* Ký tự: Giá trị không hợp lệ (không phải là số).
* Các giá trị kiểm thử: 49, 50, 51,e

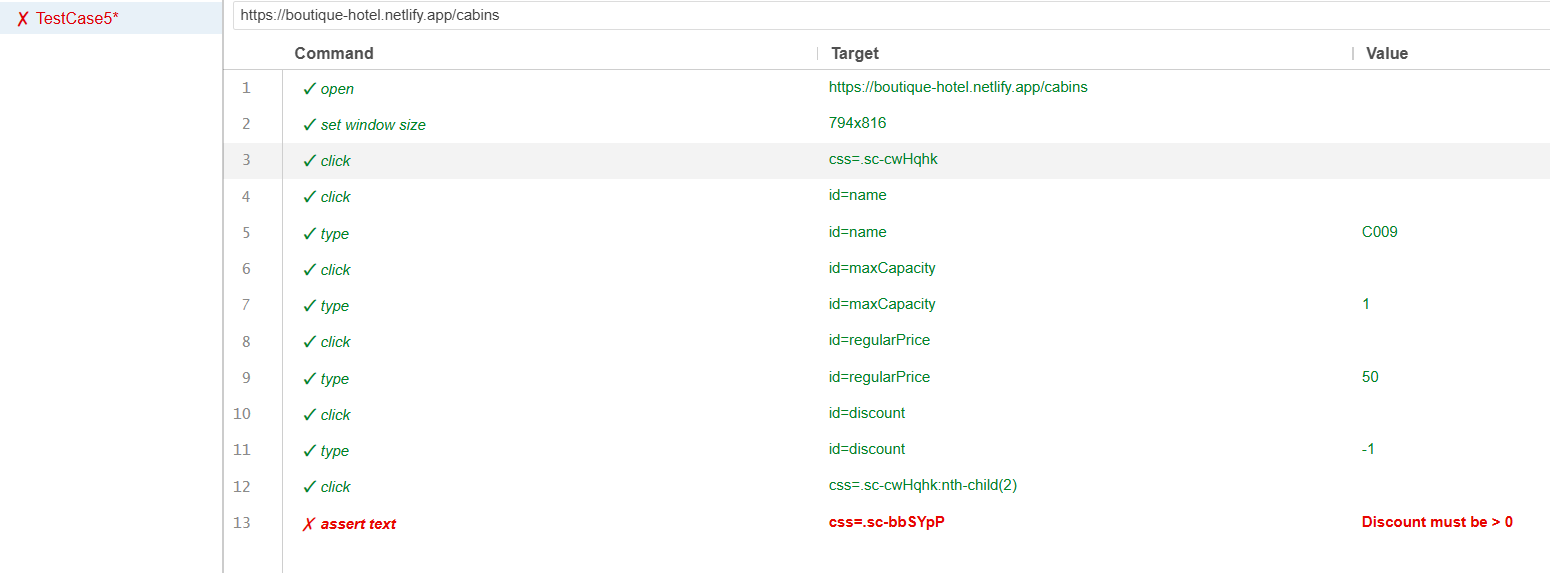
**Discount**

* Điều kiện: discount phải là số, ≥ 0, và <= regularPrice.
* Giá trị biên:
* Biên dưới: -1 (giá trị không hợp lệ, nhỏ hơn 0).
* Biên hợp lệ thấp nhất: 0 (giá trị hợp lệ thấp nhất).
* Biên trên phụ thuộc vào regularPrice: Nếu regularPrice = 50, giá trị biên trên của discount là 49 (giá trị lớn nhất hợp lệ, nhỏ hơn regularPrice).
* Giá trị lớn hơn biên trên: 50 (giá trị hợp lệ, bằng regularPrice).
* Ký tự: Giá trị không hợp lệ (không phải là số).
* Các giá trị cần kiểm thử: -1, 0, 49, 50,e.

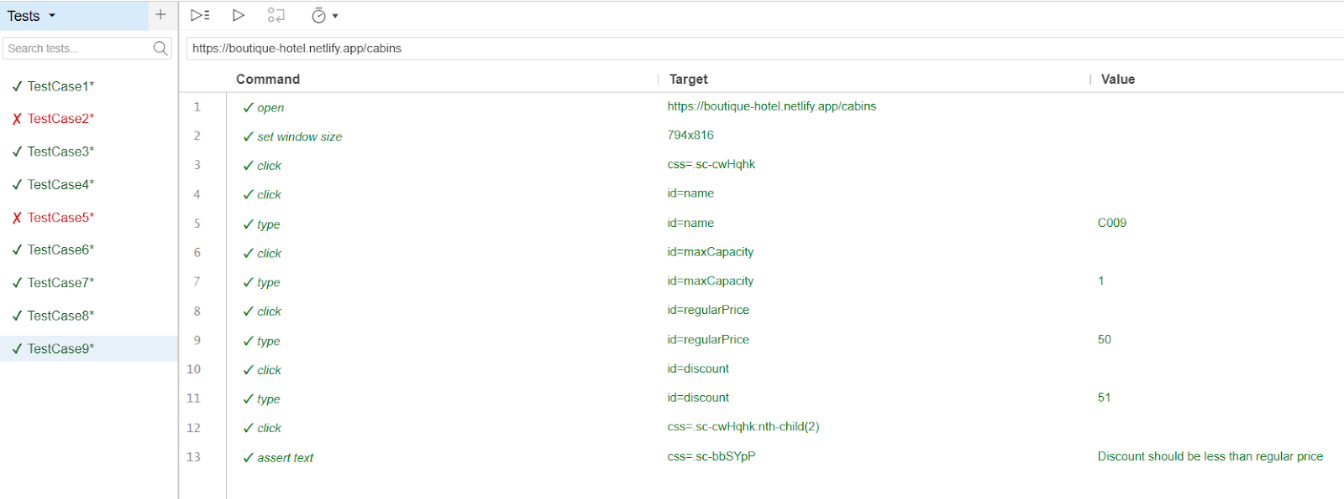
1. Các ca kiểm thử

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Cabin Name | Capacity | Regular Price | Discount | Mô tả | Kết quả mong đợi | Result Output |
| TC1 | C009 | 1 | 50 | 10 | Tất cả các đầu vào hợp lệ. | "New cabin successfully created" | PASS |
| TC2 | C009 | 1 | 350 | 50 | Tất cả các đầu vào hợp lệ. | Thêm phòng thành công | FAIL  (In ra “Discount should be less than regular price” |
| TC3 | C009 | 0 | 50 | 10 | maxCapacity < 1 | “Capacity should be 1” | PASS |
| TC4 | C009 | e | 50 | 10 | maxCapacity không phải là số | “Vui lòng nhập một số” | PASS |
| TC5 | C009 | 1 | 49 | 10 | regularPrice < 50 | “Capacity should be 50” | PASS |
| TC6 | C009 | 1 | 50 | -1 | discount < 0 | “Discount must be > 0” | FAIL |
| TC7 | C009 | 1 | 50 | 50 | discount = regularPrice | Thêm phòng thành công | PASS |
| TC8 | C009 | 2 | e | 49 | Regular Price không phải là số | “Vui lòng nhập một số” | PASS |
| TC9 | C009 | 1 | 51 | 0 | regularPrice > 50 và discount = 0 | Thêm phòng thành công | PASS |
| TC10 | C009 | 1 | 50 | 51 | discount > regularPrice | “Discount should be less than regular price” | PASS |
| TC11 |  |  |  |  | Các trường để trống | “This field is required” and “ Can't be empty, make it at least 0” | PASS |
| TC12 |  |  |  |  | Blank space | “This field is required” and “ Can't be empty, make it at least 0” | FAIL |
| TC13 | C009 | 2 | 50 | 5 | Discount không phải là số | “Vui lòng nhập một số” | PASS |
| TC14 | C009 | 2 | 50 | 5 | Cabin name đã tồn tại | “This field is already exists” | FAIL |

1. Kết quả kiểm thử tự động

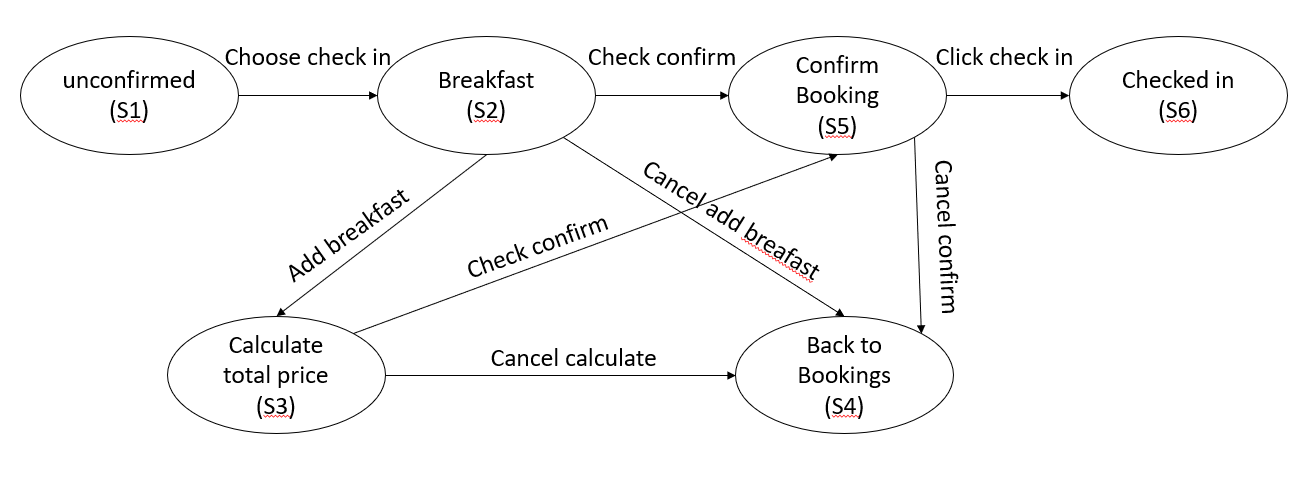


1. Kiểm thử chức năng Thêm phòng TC5



1. Kiểm thử chức năng Thêm phòng TC9
   1. Kiểm thử giao diện
      1. Kiểm thử chức năng checked in

Mô tả: Nếu khách hàng đang ở trạng thái unconfirmed, khi click vào menu sau đó chọn check in thì sẽ được chuyển sang trang check in. Ở đây nếu người dùng chọn Add Breakfast thì sẽ cộng tiền bữa sáng vào total price, và sau khi confirm booking của khách hàng thì bấm check in và in ra thông báo check in thành công



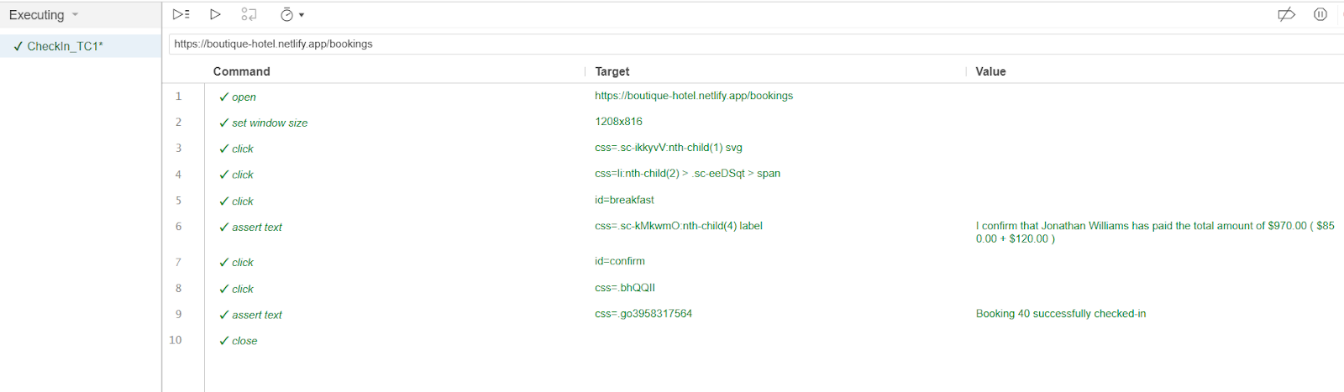
1. Bảng chuyển trạng thái

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| State | Choose check in | Add breakfast | Check confirm | Cancel add breakfast | Cancel Calculate | Cancel confirm | Click check in |
| S1 | S2 | - | - | - | - | - | - |
| S2 | - | S3 | S5 | S4 | - | - | - |
| S3 | - | - | S5 | - | S4 | - | - |
| S4 | - | - | - | - | - | - | - |
| S5 | - | - | - | - | - | S4 | S6 |
| S6 | - | - | - | - | - | - | - |

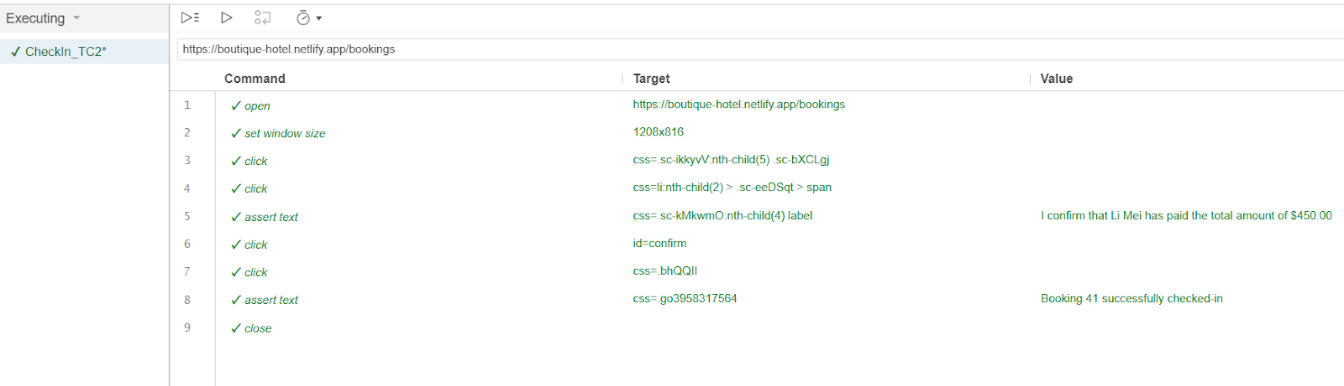
1. Các ca kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Category | Steps | Expects Output | Result Outputs |
| TC1 | Khách hàng checkin có đặt bữa sáng | 1. Tại giao diện Bookings, user chọn check in  2. Khách chọn đặt bữa sáng  3. Confirm booking  4. Click check in | - Chuyển đến giao diện check in để chọn Add breakfast  - Tính toán lại tổng chi phí cần phải trả  - Hiển thị thông báo check in thành công | PASS |
| TC2 | Khách hàng checkin không đặt bữa sáng | 1. Tại giao diện Bookings, user chọn check in  2. Khách không đặt bữa sáng  3. Confirm booking  4. Click check in | - Chuyển đến giao diện check in để chọn Add breakfast  - Tổng chi phí giữ nguyên như lúc đầu  - Hiển thị thông báo check in thành công | PASS |
| TC3 | Chọn check in nhưng bấm Back để quay lại trang Bookings | 1. Tại giao diện Bookings, user chọn check in  2. Bấm Back | - Chuyển đến giao diện check in để chọn Add breakfast  - Quay lại trang Bookings và trạng thái của khách vẫn là unconfirmed | PASS |
| TC4 | Huỷ check in khi đang chọn bữa sáng | 1. Tại giao diện Bookings, user chọn check in  2. Khách chọn đặt bữa sáng  3. Confirm booking  4. Bấm back | - Chuyển đến giao diện check in để chọn Add breakfast  - Tính toán lại tổng chi phí cần phải trả  - Quay lại trang Bookings và trạng thái của khách vẫn là unconfirmed | PASS |
| TC5 | Huỷ check in khi không chọn bữa sáng | 1. Tại giao diện Bookings, user chọn check in  2. Khách không đặt bữa sáng  3. Confirm booking  4. Bấm back | - Chuyển đến giao diện check in để chọn Add breakfast  - Tổng chi phí giữ nguyên như lúc đầu  - Quay lại trang Bookings và trạng thái của khách vẫn là unconfirmed | PASS |
| TC6 | Check in, chọn bữa sáng nhưng không chọn confirm | 1. Tại giao diện Bookings, user chọn check in  2. Khách đặt bữa sáng  3. Không chọn Confirm booking | - Chuyển đến giao diện check in để chọn Add breakfast  - Tính toán lại tổng chi phí cần phải trả  - Nút check in booking phải disable | PASS |
| TC7 | Check in, Không chọn bữa sáng và không chọn confirm | 1. Tại giao diện Bookings, user chọn check in  2. Khách không đặt bữa sáng  3. Không chọn Confirm booking | - Chuyển đến giao diện check in để chọn Add breakfast  - Tổng chi phí giữ nguyên như lúc đầu  - Nút check in booking phải disable | PASS |

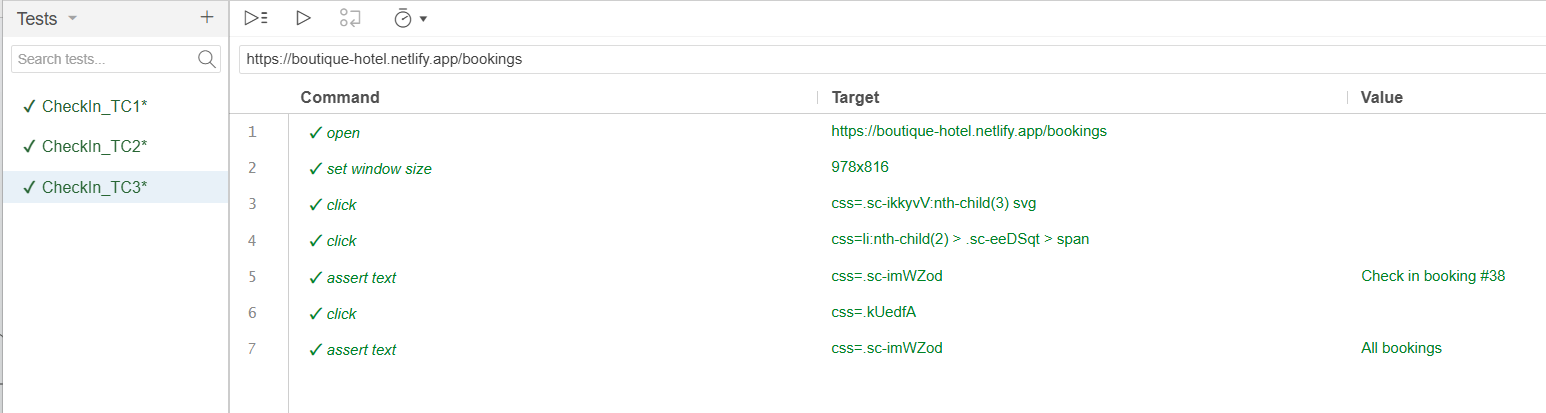
1. Kết quả kiểm thử tự động bằng Selenium



1. Kiểm thử chức năng Checkin TC1



1. Kiểm thử chức năng Checkin TC2



1. Kiểm thử chức năng Checkin TC3
2. ĐÁNH GIÁ VÀ SO SÁNH KIỂM THỬ THỦ CÔNG VÀ KIỂM THỬ TỰ ĐỘNG
   1. So sánh kiểm thử thủ công và kiểm thử tự động:

Phân tích điểm mạnh, điểm yếu của từng phương pháp trong phát hiện lỗi.

Đánh giá khả năng ứng dụng và hiệu quả của từng phương pháp.

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Kiểm thử thủ công | Kiểm thử tự động |
| Ưu điểm | Linh hoạt, phát hiện lỗi trực quan, phù hợp với các trường hợp đặc biệt | Tốc độ cao, độ chính xác cao, lặp lại được, phù hợp với kiểm thử hồi quy |
| Nhược điểm | Tốn thời gian, dễ xảy ra sai sót, khó mở rộng | Chi phí đầu tư ban đầu cao, cần chuyên môn kỹ thuật, khó kiểm thử các trường hợp đặc biệt |
| Khả năng ứng dụng | Kiểm thử UI/UX, kiểm thử khám phá, kiểm thử khả dụng | Kiểm thử hồi quy, kiểm thử hiệu năng, kiểm thử API |
| Chi phí | Thấp (chủ yếu là nhân lực) | Cao (công cụ, kịch bản, bảo trì) |

Đánh giá khả năng ứng dụng và hiệu quả:

* Kết hợp cả hai: Kiểm thử thủ công tập trung vào các trường hợp đặc biệt và trải nghiệm người dùng, trong khi kiểm thử tự động tăng cường độ phủ và độ tin cậy cho các trường hợp lặp đi lặp lại.
* Ưu tiên tự động hóa: Nên tự động hóa các quy trình kiểm thử có tính lặp đi lặp lại, dễ xảy ra sai sót hoặc yêu cầu nhiều thời gian.
* Linh hoạt: Điều chỉnh chiến lược kiểm thử dựa trên từng giai đoạn phát triển và yêu cầu cụ thể của dự án.
  1. Đánh giá độ bao phủ kiểm thử:

Độ bao phủ của các test case:

* Đánh giá mức độ bao phủ của các test case đã thiết kế, bao gồm số lượng và phạm vi các chức năng được kiểm thử. Đối với các chức năng quan trọng và dễ bị lỗi, kiểm thử tự động giúp tăng cường độ bao phủ, đặc biệt khi thực hiện hồi quy.

Những loại lỗi dễ phát hiện hơn:

* Kiểm thử thủ công: Dễ phát hiện các lỗi liên quan đến giao diện người dùng, tương tác, và các lỗi đặc thù cần sự quan sát trực quan.
* Kiểm thử tự động: Thường phát hiện các lỗi liên quan đến logic, hồi quy, và hiệu suất, đặc biệt khi thực hiện các ca kiểm thử có dữ liệu đầu vào lớn hoặc phức tạp.
  1. Hạn chế và bài học kinh nghiệm:

Hạn chế của dự án kiểm thử:

* Các hạn chế về tài nguyên, thời gian, và chi phí đã ảnh hưởng đến độ bao phủ và tính hiệu quả của kiểm thử, đặc biệt là kiểm thử thủ công. Khả năng mở rộng kiểm thử tự động cũng bị giới hạn trong một số tình huống yêu cầu sự can thiệp và quan sát của con người.

Bài học kinh nghiệm khi thực hiện kiểm thử kết hợp:

* Sử dụng kết hợp kiểm thử thủ công và tự động giúp tận dụng điểm mạnh của cả hai phương pháp. Việc xác định rõ ràng các tình huống nên kiểm thử tự động và các tình huống nên kiểm thử thủ công đã giúp tối ưu hóa hiệu quả và độ bao phủ.
* Cần có kế hoạch duy trì và cập nhật liên tục các ca kiểm thử tự động để đảm bảo tính chính xác và độ tin cậy khi có thay đổi trong hệ thống.

KẾT LUẬN

Sau quá trình nghiên cứu và triển khai đề tài, việc kiểm thử hệ thống quản lý khách sạn bằng công cụ Selenium đã giúp đạt được nhiều kết quả khả quan. Việc sử dụng Selenium không chỉ tiết kiệm thời gian kiểm thử mà còn đảm bảo độ chính xác cao, phát hiện được nhiều lỗi tiềm ẩn trong hệ thống. Thông qua các kịch bản kiểm thử được thiết kế và thực hiện, hiệu quả kiểm thử tự động được chứng minh trong việc kiểm tra các chức năng quan trọng như đặt phòng, thanh toán, quản lý thông tin khách hàng và nhân viên.

Đề tài đã giúp em hiểu rõ hơn về lợi ích cũng như các thách thức của việc áp dụng kiểm thử tự động trong thực tế. Những bài học và kinh nghiệm thu được trong quá trình thực hiện sẽ là nền tảng quý giá cho các dự án kiểm thử sau này. Em mong rằng đề tài này sẽ không chỉ là một tài liệu tham khảo hữu ích cho các bạn có chung mối quan tâm đến kiểm thử tự động mà còn có thể đóng góp một phần vào sự phát triển của lĩnh vực kiểm thử phần mềm tại Việt Nam.

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.http://www.testingvn.com/viewforum.php?f=33

4. http://en.wikipedia.org/wiki/Software\_testing

5. Một số trang web về kiểm thử phần mềm khác