

ĐẠI HỌC QUỐC GIA HÀ NỘI
ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO BÀI TẬP LỚN MÔN
Kiểm thử và đảm bảo chất lượng phần mềm

CÔNG CỤ KIỂM THỬ TỰ ĐỘNG SELENIUM

Giảng viên hướng dẫn: Nguyễn Thu Trang

Triệu Vũ Hải
Lại Tuấn Anh
Nguyễn Tuấn Anh

Hà Nội, 2021

Mục lục

| | | |
|----------|--|-----------|
| 1 | Mở đầu | 2 |
| 2 | Giới thiệu về Selenium | 3 |
| 2.1 | Selenium là gì? | 3 |
| 2.2 | Các công cụ của Selenium | 4 |
| 2.3 | Selenium IDE | 5 |
| 2.3.1 | Tính năng của Selenium IDE | 6 |
| 2.3.2 | Selenese là gì? | 7 |
| 2.3.3 | Phân loại các lệnh Selenium | 7 |
| 2.3.4 | Ưu điểm | 7 |
| 2.3.5 | Nhược điểm | 8 |
| 2.4 | Selenium WebDriver | 8 |
| 2.4.1 | Ưu điểm | 9 |
| 2.4.2 | Nhược điểm | 10 |
| 2.5 | Selenium Grid | 10 |
| 2.5.1 | Ưu điểm | 11 |
| 2.5.2 | Nhược điểm | 12 |
| 2.6 | So sánh với các bộ công cụ khác | 13 |
| 3 | Hướng dẫn tải và cài đặt | 14 |
| 3.1 | Selenium IDE | 14 |
| 3.2 | Selenium WebDriver | 15 |
| 3.3 | Selenium Grid | 15 |
| 4 | Áp dụng Selenium vào kiểm thử ứng dụng Web | 16 |
| 4.1 | Hệ thống học trực tuyến Remind Clone | 16 |
| 4.2 | Phân tích vấn đề | 17 |
| 4.3 | Viết ca kiểm thử | 17 |
| 4.3.1 | Ca kiểm thử cho màn hình đăng nhập | 18 |
| 4.3.2 | Ca kiểm thử cho những chức năng khác | 20 |
| 4.3.3 | Kiểm thử trên nhiều nền tảng với Selenium Grid | 22 |
| 4.4 | Thực thi ca kiểm thử | 22 |
| 4.4.1 | Kết quả kiểm thử | 23 |
| 5 | Kết luận | 24 |

1 Mở đầu

Kiểm thử phần mềm được coi là bước quan trọng nhất trong vòng đời phát triển phần mềm. Mục tiêu chính của kiểm thử phần mềm là so sánh kết quả thu được với kết quả mà người dùng cuối mong đợi. Theo dòng thời gian, các hệ thống phần mềm ngày càng lớn và phức tạp hơn. Nếu như trước đây, việc kiểm thử phần mềm hầu hết đều làm thủ công thì so với các ứng dụng hiện nay, kiểm thử bằng tay như vậy hoàn toàn không hiệu quả và tốn công sức. Việc kiểm thử các ứng dụng lớn, đặc biệt là các ứng dụng web trở nên rất khó khăn và tốn nhiều chi phí. Làm sao để giảm được những tác vụ lặp lại, sự can thiệp của con người cũng như giảm chi phí kiểm thử, bảo trì là điều tất yếu. Vì vậy, các công cụ kiểm thử tự động đã ra đời. Một trong số đó không thể không kể đến Selenium - bộ công cụ kiểm thử được cộng đồng lập trình yêu thích và sử dụng nhiều nhất hiện nay.

2 Giới thiệu về Selenium

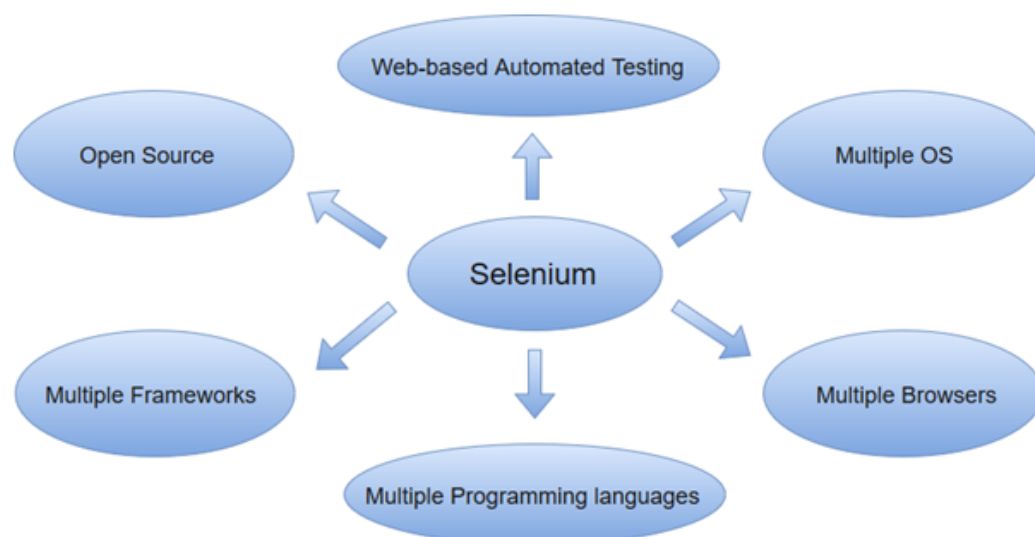
Phần này giới thiệu về bộ công cụ kiểm thử tự động hóa Selenium.

2.1 Selenium là gì?

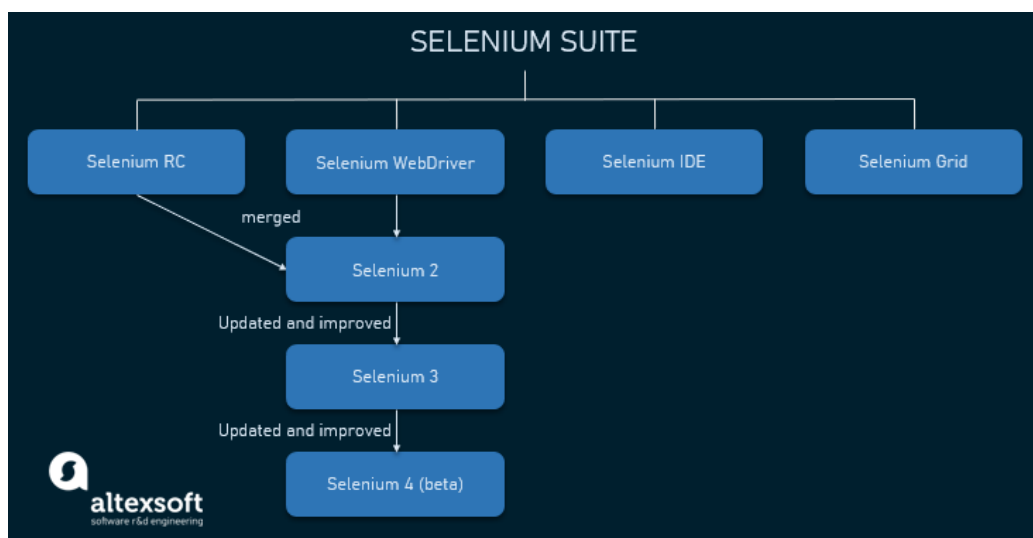
Selenium là một trong những bộ công cụ kiểm thử tự động giao diện người dùng (User Interface) có mã nguồn mở được sử dụng rộng rãi nhất. Ban đầu nó được phát triển bởi Jason Huggins vào năm 2004 như một công cụ nội bộ tại Thought Works.

Selenium không thể kiểm thử tự động các ứng dụng trên desktop vì nó chỉ có thể được sử dụng trong các trình duyệt. Selenium được coi là một trong những bộ công cụ được yêu thích nhất để kiểm thử tự động các ứng dụng web vì nó hỗ trợ hầu hết các trình duyệt web phổ biến như Google Chrome 12+, Internet Explorer 7, 8, 9, 10, Safari 5.1+, Opera 11.5, Firefox 3+ và hệ điều hành như Windows, Mac, Linux/Unix. Hơn nữa, nó còn hỗ trợ các hệ điều hành cho ứng dụng di động như iOS và Android.

Selenium cũng cung cấp khả năng tương thích với các ngôn ngữ lập trình khác nhau như Java, C#, Perl, JavaScript, Ruby, Python, PHP,... phổ biến nhất vẫn là Java và C#. Người kiểm thử có thể chọn bất cứ ngôn ngữ nào mà Selenium hỗ trợ để thiết kế các ca kiểm thử. Nếu ứng dụng đang được kiểm thử được viết bằng PHP thì người kiểm thử không nhất thiết phải viết mã Selenium bằng PHP, có thể thay bằng Java, Ruby,... Do đó sử dụng Selenium rất thuận lợi bởi tính linh hoạt của nó.



Hình 1: Framework Selenium



Hình 2: Bộ công cụ Selenium

Selenium có thể được sử dụng để tự động hóa các ca kiểm thử chức năng và tích hợp với các công cụ kiểm thử tự động như Maven, Jenkins và Docker để đạt được khả năng kiểm thử liên tục. Nó cũng có thể được tích hợp với các công cụ như TestNG và JUnit để quản lý các ca kiểm thử và tạo báo cáo.

2.2 Các công cụ của Selenium

Sản phẩm đầu tiên trong bộ kiểm thử Selenium là Selenium Remote Control (hay còn gọi là Selenium 1). Do tồn tại một số hạn chế và sự hợp nhất sau với WebDriver (dẫn đến Selenium 2), nó đã sớm không còn được sử dụng và hỗ trợ nữa. Vào năm 2016, Selenium 3 đã được phát hành, loại bỏ Selenium RC nhưng mở rộng danh sách các trình duyệt được hỗ trợ và khả năng kiểm thử di động. Vào tháng 2 năm 2021, bản phát hành beta đầu tiên của Selenium 4 đã được công bố.

Selenium IDE (Môi trường phát triển tích hợp) chủ yếu là một công cụ record and playback. Nó là một Add-on hay Extension có sẵn cho cả Firefox và Chrome, tạo ra ca kiểm thử nhanh chóng thông qua chức năng record and playback. Không cần phải học bất kỳ một ngôn ngữ kịch bản kiểm thử nào nhưng vẫn có thể tạo ra các ca kiểm thử.

Trong trường hợp làm việc với Selenium Remote Control (Điều khiển từ xa), người kiểm thử phải có kiến thức tốt về ít nhất một ngôn ngữ lập trình. Công cụ này cho phép tạo ra các ca kiểm thử bằng ngôn ngữ kịch bản đã chọn. Server và Client là hai thành phần chính của Selenium RC. Kiến trúc của nó rất phức tạp và tồn tại nhiều hạn chế.

Selenium WebDriver là phiên bản nâng cấp của Selenium RC. Nó ra đời để khắc phục những hạn chế gặp phải trong Selenium RC. Mặc dù nó là một phiên bản nâng cấp từ RC, nhưng kiến trúc của nó hoàn toàn khác với RC. Giống như Selenium RC, Selenium WebDriver cũng hỗ trợ nhiều nền tảng lập trình để mang lại sự linh hoạt và yêu cầu người kiểm thử phải biết ít nhất một ngôn ngữ lập trình nào đó.

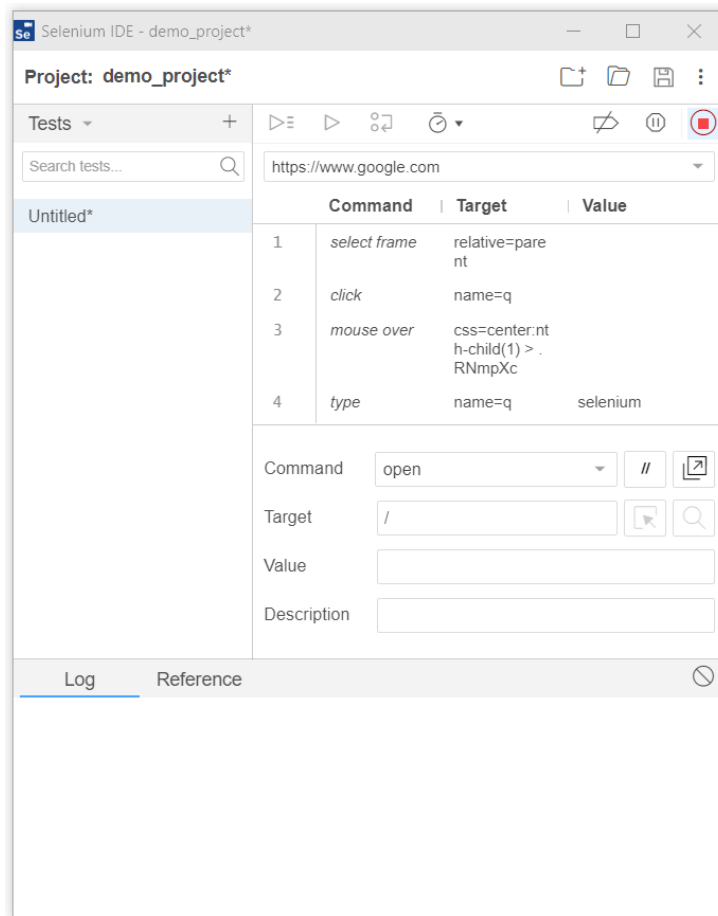
Selenium Grid là một công cụ được sử dụng để thực thi đồng thời các ca kiểm thử trên các trình duyệt và hệ điều hành khác nhau cùng một lúc. Công cụ này giúp kiểm tra khả năng tương thích giữa các trình duyệt rất dễ dàng. Đến nay, Selenium Grid đã ra mắt phiên bản 4, tuy nhiên phiên bản này vẫn đang trong quá trình thử nghiệm. Để tránh lỗi phát sinh trong quá trình kiểm thử, nên chọn phiên bản ổn định mới nhất là 3.14.

2.3 Selenium IDE

Selenium IDE (Môi trường phát triển tích hợp) là công cụ record and playback mà bất cứ ai cũng có thể sử dụng từ tester đến lập trình viên, thậm chí là những người không có chuyên môn về lập trình bởi đây là một công cụ dễ sử dụng, không yêu cầu bất cứ kiến thức về lập trình, chỉ cần thêm tiện ích mở rộng Selenium IDE từ cửa hàng vào trình duyệt.

Selenium IDE cung cấp một GUI (Giao diện người dùng đồ họa) để dễ dàng ghi lại các tương tác của người dùng với trang web, cho phép người dùng hoặc lập trình viên tạo ra các ca kiểm thử, bộ kiểm thử và chỉnh sửa nó theo yêu cầu của họ.

Môi trường phát triển cũng cung cấp khả năng chuyển đổi các ca kiểm thử sang các ngôn ngữ lập trình khác nhau. Trước đó, Selenium chỉ có sẵn cho người dùng Firefox. Nhưng giờ đây, từ phiên bản nâng cấp 3.12.0 vào ngày 16 tháng 7 năm 2019, cộng đồng Selenium đã giới thiệu Selenium IDE cho Chrome. Sau cùng, người dùng Chrome cũng được cung cấp đầy đủ các tính năng của Selenium IDE như trên Firefox. Ảnh chụp màn hình bên dưới cho biết cách Selenium IDE trên Chrome xuất hiện sau khi nhập tên dự án và URL cơ sở.



Hình 3: Giao diện Selenium IDE

Sau khi nhập URL của trang web, quá trình record bắt đầu và tất cả các tương tác với trang web được ghi lại và phân loại thành ba loại chính là:

- Command: câu lệnh
- Target: vị trí của phần tử
- Value: giá trị kiểm thử so với kết quả thực tế

Tham số target và value có thể có hoặc không và chúng phụ thuộc vào command.

2.3.1 Tính năng của Selenium IDE

Các tính năng mà Selenium IDE cung cấp đều nằm trên thanh công cụ giúp người dùng có thể dễ dàng kiểm soát quá trình thực thi các ca kiểm thử:

- Speed Control - Kiểm soát tốc độ của các ca kiểm thử
- Run All - Thực thi toàn bộ các ca kiểm thử
- Run - Thực thi ca kiểm thử hiện tại
- Pause/Resume - Tạm dừng và tiếp tục một ca kiểm thử cụ thể
- Step - Bước vào từng lệnh cụ thể trong test script Rollup - Nhóm tất cả các lệnh Selenese lại với nhau và yêu cầu chúng thực thi một thao tác duy nhất

2.3.2 Selenese là gì?

Selenese là ngôn ngữ được sử dụng để viết các lệnh trong Selenium. Các lệnh Selenium này được sử dụng để kiểm tra các ứng dụng web. Dựa trên các thẻ HTML từ giao diện người dùng, người kiểm thử có thể kiểm tra sự tồn tại của chúng. Các lệnh giúp Selenium hiểu những hành động hoặc thao tác cần thực hiện.

2.3.3 Phân loại các lệnh Selenium

Các lệnh Selenium được phân loại thành ba loại:

- Actions - Gồm các lệnh thao tác trực tiếp với các phần tử : click, type,... Ví dụ: Nhấp chuột vào một số liên kết hoặc tùy chọn trên trang.
- Accessors - Gồm các lệnh để lưu giá trị vào một biến: store, storeTitle,... Ví dụ: Lệnh “storeTextPresent” - nếu văn bản được tìm thấy trên trang thì nó sẽ lưu trữ “true”, ngược lại lưu trữ “false”.
- Assertions - So sánh kết quả mong đợi và thực tế. Chia thành 3 chế độ: Assert, Verify và WaitFor. Chúng hoạt động giống như một bảng checkpoint, nếu cả hai giá trị đều bằng nhau thì ca kiểm thử đó trả về “pass”, ngược lại trả về “false”. Do đó, Assertions giúp xác minh trạng thái của ứng dụng sau khi thực hiện test case có phù hợp với trạng thái mong muốn hay không Ví dụ: VerifyText, waitForPageToLoad.

2.3.4 Ưu điểm

- Tự động ghi lại các ca kiểm thử dựa trên các tương tác với trình duyệt
- Mang lại sự linh hoạt trong việc thực thi các ca kiểm thử. Người kiểm thử có thể chạy toàn bộ các ca kiểm thử hoặc thực thi một ca kiểm thử duy nhất

- Hoạt động dựa trên tập hợp các câu lệnh Selenese phong phú, giúp IDE hiểu được những gì cần phải thực hiện
- Cho phép người kiểm thử đặt các breakpoint để gỡ lỗi các ca kiểm thử cụ thể
- Tái sử dụng các ca kiểm thử bằng lệnh run.
- Sử dụng nhiều bộ định vị cho từng phần tử trong IDE đảm bảo thực thi thành công

2.3.5 Nhược điểm

- Không thích hợp để kiểm thử với dữ liệu lớn
- Không thể kiểm tra kết nối với cơ sở dữ liệu
- Không thể xử lý phần động của các ứng dụng web
- Không hỗ trợ chụp ảnh màn hình khi kiểm thử không thành công
- Không có sẵn tính năng để tạo báo cáo kết quả

2.4 Selenium WebDriver

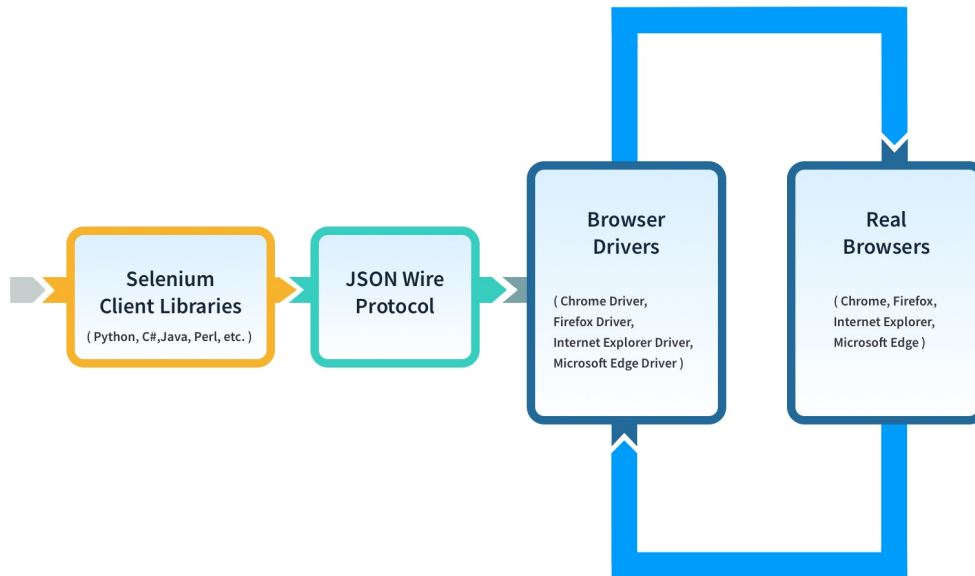
Selenium WebDriver là một web framework cho phép thực thi các ca kiểm thử trên nhiều trình duyệt. Công cụ này được sử dụng để kiểm thử tự động các ứng dụng web và xác minh rằng chúng hoạt động như ta như mong đợi.

Selenium WebDriver cho phép lựa chọn ngôn ngữ lập trình để tạo ra các kịch bản kiểm thử. Đây là phiên bản nâng cấp so với Selenium Remote Control để khắc phục một số hạn chế tồn tại trước đó. Selenium WebDriver không có khả năng xử lý các thành phần của sổ, nhưng nhược điểm này có thể được khắc phục bằng cách sử dụng các công cụ như Sikuli, Auto IT,...

Kiến trúc WebDriver được tạo thành từ bốn thành phần chính: Thư viện Selenium Client, giao thức JSON qua HTTP, trình điều khiển trình duyệt và các trình duyệt.

Thư viện Selenium Client Selenium cung cấp khả năng tương thích cho nhiều ngôn ngữ lập trình như Ruby, Python, Java, C#,... đều có trên trang web chính thức của Selenium.

JSON Wire Protocol JSON là từ viết tắt của JavaScript Object Notation. JSON là một tiêu chuẩn mở để trao đổi dữ liệu giữa client và server.



Hình 4: Thành phần chính của WebDriver

Trình điều khiển trình duyệt Selenium cung cấp các trình điều khiển cụ thể cho từng trình duyệt.

Các trình duyệt Selenium WebDriver hỗ trợ cho nhiều trình duyệt như Chrome, Firefox, Safari, Internet Explorer,...

2.4.1 Ưu điểm

- Là một trong những công cụ kiểm thử mã nguồn mở phổ biến nhất và rất dễ bắt đầu để kiểm thử các ứng dụng web, cho phép thực hiện kiểm thử tự động
- Hỗ trợ nhiều hệ điều hành như Windows, Mac, Linux, Unix,...
- Hỗ trợ cho các trình duyệt hiện đại như Microsoft Edge, Google Chrome, Firefox, Opera, Safari,...
- Cung cấp khả năng tương thích với một số ngôn ngữ lập trình như Python, Java, Perl, Ruby,...
- Tốc độ của Selenium WebDriver nhanh hơn một số công cụ kiểm thử tự động khác do giao tiếp trực tiếp với trình duyệt

- API ngắn gọn hơn so với Selenium Remote Control
- Cung cấp khả năng tương thích với iPhoneDriver, AndroidDriver và HtmlUnitDriver

2.4.2 Nhược điểm

- Không có lệnh tích hợp để tạo kết quả kiểm thử tự động
- Cần có hiểu biết cơ bản về ngôn ngữ lập trình
- Không thể kiểm thử tự động Audio và Video
- Cài đặt phức tạp, cần có công cụ để viết mã

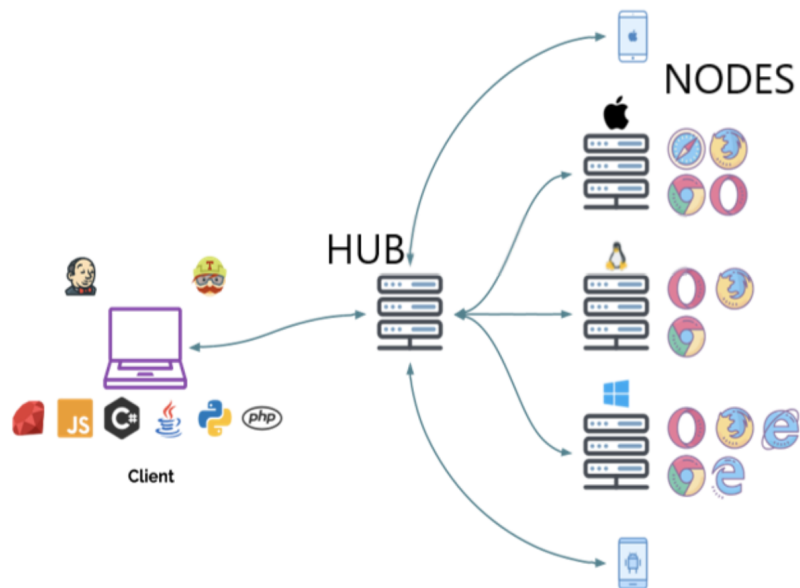
2.5 Selenium Grid

Selenium Grid là một proxy server thông minh giúp người dùng dễ dàng thực thi các bộ kiểm thử song song trên nhiều thiết bị và trình duyệt khác nhau. Việc kiểm thử được thực hiện bằng cách định tuyến các lệnh đến các trình duyệt web từ xa. Các thiết bị được sử dụng để kiểm thử không nhất thiết phải lưu trữ hard code do Selenium Grid sử dụng chế độ phân tán, chỉ với 1 code base duy nhất là có thể hoạt động được.

Selenium Grid được tạo thành từ hai thành phần chính: hub và node.

Hub máy chủ server, chứa hard code và là nơi nhận các yêu cầu truy cập từ WebDriver client, định tuyến các lệnh kiểm thử đến các Nodes. Hub chỉ có thể được cài đặt duy nhất trên một máy tính.

Node những thiết bị bao gồm một hệ điều hành gốc và một WebDriver từ xa được kết nối vào Hub để thực thi các kịch bản kiểm thử. Có thể có nhiều Nodes trong một mô hình Grid.



Hình 5: Kiến trúc Selenium Grid

Người kiểm thử nên sử dụng Selenium Grid trong các trường hợp:

- Chạy kiểm thử trên nhiều trình duyệt, thiết bị và hệ điều hành khác nhau và các phiên bản của chúng.
- Giảm thời gian mà một bộ kiểm thử cần hoàn thành.

Selenium Grid cải thiện chu kỳ thời gian của các kết quả thử nghiệm. Sự khác biệt này xảy ra là đáng kể, đặc biệt là khi bộ kiểm thử lớn và cần nhiều thời gian hơn để chạy. Nó cung cấp tính linh hoạt và mở rộng phạm vi thử nghiệm trong một thời gian giới hạn. Kể từ khi đưa cơ sở hạ tầng ảo vào sử dụng, việc bảo trì trở nên dễ dàng hơn rất nhiều.

2.5.1 Ưu điểm

- Mã nguồn mở, cho phép thực hiện kiểm thử tự động, tương thích với nhiều trình duyệt, hệ điều hành, hỗ trợ nhiều ngôn ngữ lập trình
- Tăng thời gian hoàn thành bộ kiểm thử vì khả năng chạy kiểm thử song song
- Thực thi trên đám mây, tăng tính khả dụng, độ tin cậy, tiết kiệm chi phí bảo trì phần cứng và phần mềm

2.5.2 Nhược điểm

- Gia tăng chi phí cho dự án do yêu cầu các thiết bị kiểm thử làm Nodes
- Khả năng mở rộng kém, gần như không thể tăng hoặc giảm quy mô theo yêu cầu
- Phải cấu hình lại khi có thay đổi do máy chủ được cấu hình trước với một tập hợp các phiên bản trình duyệt bắt buộc có sẵn
- Bảo mật kém, yêu cầu người kiểm thử có tay nghề cao thì mới có thể tạo ra và bảo trì nó

2.6 So sánh với các bộ công cụ khác

| | Selenium | Katalon Studio | UFT | TestComplete |
|-----------------------------------|---|---|---|--|
| Hệ điều hành | Cross-platform | Cross-platform | Windows | Windows |
| Ứng dụng kiểm thử | Web, Mobile (với Appium) | Web, Mobile, Desktop | Web, Mobile, Desktop | Web, Mobile, Desktop |
| Ngôn ngữ kịch bản | Java, C#, Perl, Python, JavaScript, Ruby, PHP | Java, Groovy | VBScript | JavaScript, Python, VBScript, JScript, Delphi, C++, C# |
| Kỹ năng lập trình | Có yêu cầu | Không yêu cầu | Không yêu cầu | Không yêu cầu |
| Cài đặt và sử dụng | Yêu cầu cài đặt một số công cụ | Dễ | Dễ | Dễ |
| Thời gian tạo script | Chậm | Nhanh | Nhanh | Nhanh |
| Lưu trữ và bảo trì | XPath, UI Maps | Built-in object repository, XPath, object re-identification | Built-in object repository, smart object detection and correction | Built-in object repository, detecting common objects |
| Kiểm thử dựa trên hình ảnh | Yêu cầu thư viện ngoài | Hỗ trợ tích hợp | Hỗ trợ tích hợp | Hỗ trợ tích hợp |
| Tích hợp DevOps/ALM | Yêu cầu thư viện ngoài | Có | Có | Có |
| Tích hợp liên tục | Jenkins, Cruise Control | Jenkins, Teamcity | Jenkins, HP Quality Center | Jenkins, HP Quality Center |
| Phân tích kiểm thử | Không | Có | Có | Có |
| Giấy phép | Mã nguồn mở | Độc quyền | Độc quyền | Độc quyền |
| Chi phí | Miễn phí | Có phí | Có phí | Có phí |



Selenium IDE

Do seleniumhq.org cung cấp

★★★★★ 217 | Công cụ dành cho nhà phát triển | 500.000+ người dùng

Thêm vào Chrome

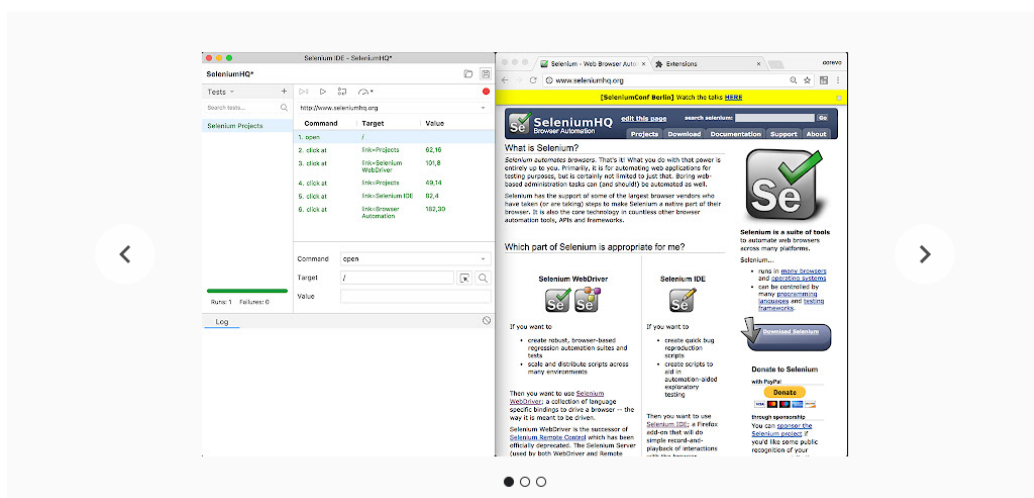
Tổng quan

Phương thức bảo vệ quyền riêng tư

Bài đánh giá

Hỗ trợ

Có liên quan



Hình 6: Trang của Selenium IDE trên Chrome Web Store

3 Hướng dẫn tải và cài đặt

Trong phần này, nhóm sẽ giới thiệu cách cài đặt và sử dụng 3 công cụ của Selenium.

3.1 Selenium IDE

Vì Selenium IDE là add-on trên Mozilla Firefox hoặc extension trên Google Chrome, Microsoft Edge nên việc cài đặt Selenium IDE vô cùng đơn giản. Yêu cầu bắt buộc để cài đặt Selenium IDE là máy tính có cài đặt một trong các trình duyệt trên. Nhóm sẽ hướng dẫn các bước cài đặt Selenium IDE trên trình duyệt Google Chrome.

1. Tìm kiếm Selenium IDE trên cửa hàng extension của Google Chrome hoặc Firefox (hình 6)
2. Nhấn “Thêm vào Chrome” để tiến hành cài đặt

3. Sau khi cài đặt xong, Selenium IDE sẽ hiển thị trên thanh công cụ. Nhấn vào biểu tượng Selenium IDE để chạy ứng dụng

3.2 Selenium WebDriver

Trước khi sử dụng được công cụ WebDriver, trình điều khiển cho các trình duyệt như Chrome, Firefox phải được cài đặt. Các trình điều khiển này có thể được tải về trên trang chủ của Selenium. Sau khi tải về, người dùng cần đưa các trình điều khiển vào PATH trên máy tính.

Khi hoàn thành cài đặt trình điều khiển, người dùng có thể bắt đầu cài đặt WebDriver. Công cụ này hỗ trợ nhiều ngôn ngữ lập trình phổ biến, nhưng tài liệu này sẽ chỉ hướng dẫn cài đặt WebDriver cho ngôn ngữ Java. Người dùng có thể cài đặt công cụ này thông qua Maven bằng việc thêm dependency vào file pom.xml trong project Java.

```
<!--Add this to pom.xml-->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.X</version>
</dependency>
```

Listing 1: pom.xml

3.3 Selenium Grid

Để cài đặt Selenium Grid, người dùng cần tải xuống file jar cho chương trình Selenium Server trên trang Downloads của Selenium. Sau đó, người dùng có thể chạy file jar này để cài đặt máy tính của mình với tư cách là một hub hoặc một node.

```
java -jar selenium-server-standalone.jar -role hub
```

Listing 2: Lệnh để chạy Selenium Server

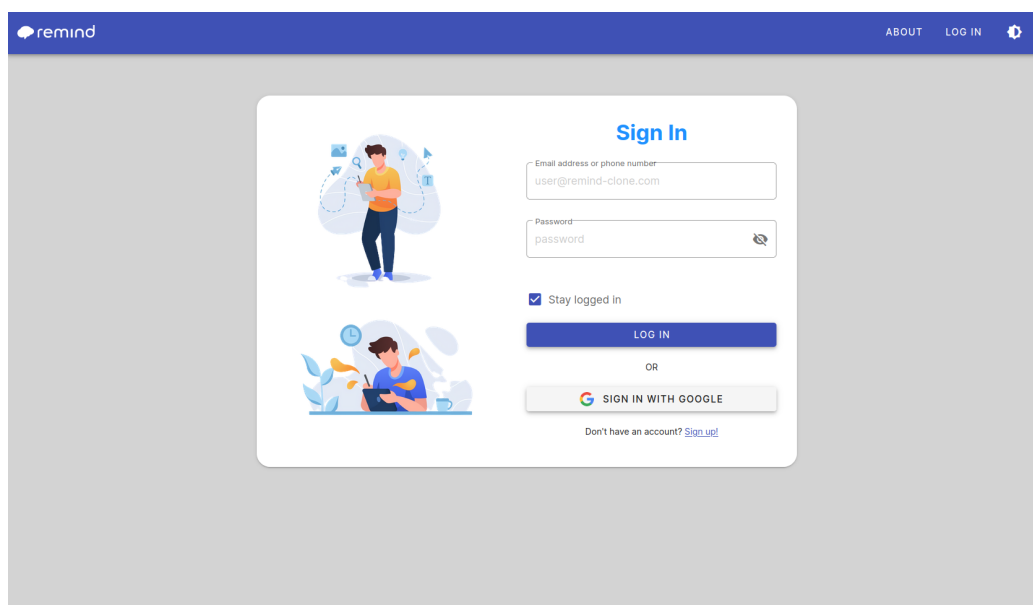
4 Áp dụng Selenium vào kiểm thử ứng dụng Web

Trong phần này, nhóm sẽ giới thiệu một ứng dụng mẫu và các bước để thực hiện ca kiểm thử Selenium trên ứng dụng đó.

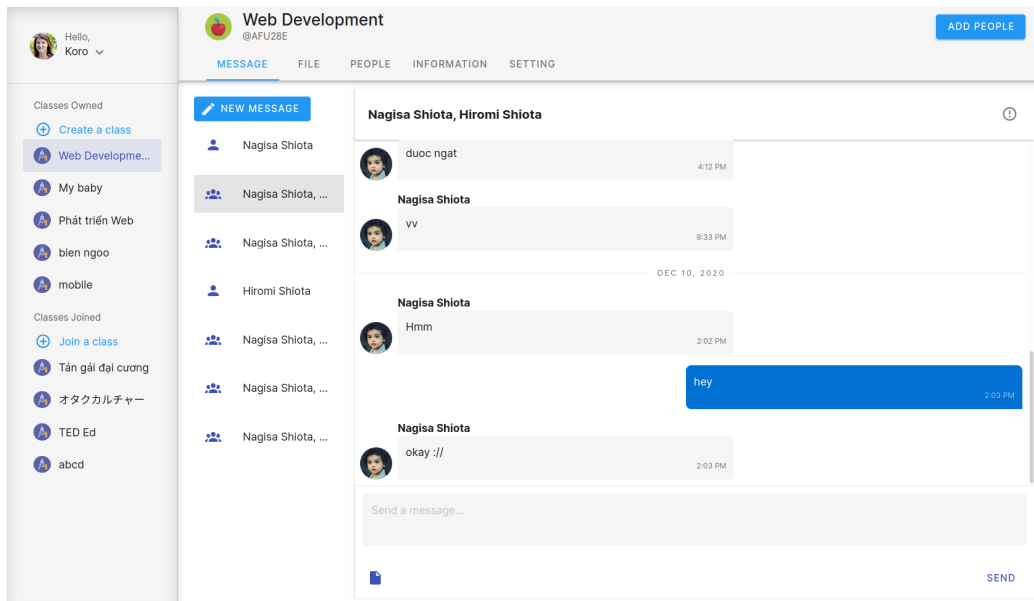
4.1 Hệ thống học trực tuyến Remind Clone

Ứng dụng được nhóm lựa chọn để thực hiện kiểm thử là hệ thống học trực tuyến Remind Clone. Đây là phần mềm hỗ trợ giáo viên, học sinh, phụ huynh trong việc giảng dạy, học tập và những công việc liên quan đến giáo dục.

Giáo viên sẽ sử dụng nền tảng này để tạo lớp học và thêm học sinh vào lớp học của mình. Tại đây, giáo viên sẽ có thể nhắn tin, đăng thông báo và chia sẻ tài liệu cho học sinh trong lớp. Ngược lại, học sinh có thể trò chuyện với nhau và với giáo viên của mình. Bằng việc chia người dùng vào từng lớp học riêng biệt, mỗi giáo viên có thể quản lý lớp học của mình hiệu quả hơn. Họ cũng có thể đưa ra một số cài đặt đặc biệt cho lớp mình đang dạy để giúp việc quản lý trở nên dễ dàng hơn.



Hình 7: Màn hình đăng nhập



Hình 8: Màn hình chính

4.2 Phân tích vấn đề

Trong bước này, nhóm thực hiện phân tích yêu cầu của chương trình. Hệ thống học trực tuyến Remind Clone có những chức năng chính sau:

- Đăng nhập và đăng ký
- Tạo và gia nhập lớp học
- Xem và gửi tin nhắn
- Chia sẻ tài liệu

Khi đã xác định được danh sách chức năng, nhóm sẽ xác định các giao diện và thành phần tương ứng để thực hiện những chức năng đó. Sau khi hoàn thành bước phân tích vấn đề, nhóm sẽ chuyển sang bước tiếp theo, đó là viết ca kiểm thử.

4.3 Viết ca kiểm thử

Một ca kiểm thử có nhiệm vụ thực hiện một chức năng cụ thể thông qua việc tương tác với các thành phần giao diện. Thông qua đó, ca kiểm thử có thể xác định tính đúng đắn của hệ thống. Để viết ra các ca kiểm thử cụ thể, Selenium cung cấp cho người dùng hai công cụ là Selenium IDE và Selenium

WebDriver. Hai công cụ này phục vụ mục đích tương tự nhau nhưng trong đó, WebDriver đem lại hiệu năng cao hơn cũng như cho phép tùy chỉnh ca kiểm thử dễ dàng hơn. Hơn nữa, WebDriver hỗ trợ nhiều ngôn ngữ lập trình phổ biến, khiến việc viết ca kiểm thử trở nên đơn giản hơn. Vì vậy, nhóm chọn WebDriver để thực hiện việc kiểm thử. Ngoài ra, nhóm cũng sử dụng framework JUnit của Java để so sánh giá trị trả về và sinh báo cáo kết quả kiểm thử.

Ca kiểm thử sẽ được viết theo mẫu thiết kế Page Object Models. Theo mẫu thiết kế này, mỗi trang giao diện cần kiểm thử sẽ được đóng gói thành một class. Các ca kiểm thử thay vì tương tác trực tiếp với giao diện qua WebDriver thì sẽ dựa vào các class này để thực hiện thao tác và lấy các giá trị cần thiết ra để so sánh. Bằng cách này, mã nguồn sẽ có thể tái sử dụng và bảo trì dễ dàng hơn do toàn bộ thao tác liên quan đến giao diện đều được đóng gói. Ngoài ra, các ca kiểm thử sẽ đều kế thừa lớp AbstractTest. Lớp này sẽ khởi tạo đối tượng WebDriver và thực hiện đóng trình duyệt khi ca kiểm thử kết thúc. Cấu trúc file nhóm sử dụng có dạng như sau:

```
test/
├── AbstractTest.java
├── FileSharingTest.java
├── LoginTest.java
├── page/
│   ├── FileListPage.java
│   ├── HomePage.java
│   ├── LoginPage.java
│   ├── RegisterPage.java
│   └── SendMessagePage.java
├── PropertyReader.java
├── RegisterTest.java
└── SendMessageTest.java
```

4.3.1 Ca kiểm thử cho màn hình đăng nhập

Màn hình đăng nhập có 3 thành phần chính: input cho email, input cho password và nút đăng nhập. Ca kiểm thử sử dụng một tài khoản có sẵn trong hệ thống để kiểm tra xem người dùng có được đưa tới trang chủ sau khi đăng nhập thành công hay không.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
```

```

public class LoginPage {
    protected WebDriver driver;

    private By emailBy = By.id("txtEmail");
    private By passwordBy = By.id("txtPassword");
    private By loginBtnBy = By.id("btnSignIn");
    private By formTitleBy = By.className("titleSignin");

    public LoginPage(WebDriver driver) {
        this.driver = driver;

        new WebDriverWait(driver, 2).until(ExpectedConditions
.presenceOfElementLocated(formTitleBy));
    }

    public HomePage attemptLogin(String email, String
password) {
        WebElement loginInput = driver.findElement(emailBy);
        loginInput.sendKeys(email);
        WebElement passwordInput = driver.findElement(
passwordBy);
        passwordInput.sendKeys(password);
        WebElement loginBtn = driver.findElement(loginBtnBy);
        loginBtn.click();

        return new HomePage(driver);
    }
}

```

Listing 3: LoginPage.java

```

@Test
public void loginWithCorrectInfo() {
    LoginPage loginPage = new LoginPage(driver);
    loginPage.attemptLogin(EMAIL, PASSWORD);
}

```

Listing 4: Ca kiểm thử đúng email và mật khẩu

Ngoài ra, trường hợp email hoặc mật khẩu sai cũng được kiểm thử để đảm bảo tính bảo mật của hệ thống.

```

@Test
public void loginWithIncorrectUsername() {
    LoginPage loginPage = new LoginPage(driver);
    assertThrows("User failed to login", Exception.class,
() -> loginPage.attemptLogin(WRONG_EMAIL, PASSWORD));
}

```

```

@Test
public void loginWithIncorrectPassword() {
    LoginPage loginPage = new LoginPage(driver);
    assertThrows("User failed to login", Exception.class,
    () -> loginPage.attemptLogin(EMAIL, WRONG_PASSWORD));
}

```

Listing 5: Ca kiểm thử sai thông tin đăng nhập

4.3.2 Ca kiểm thử cho những chức năng khác

Việc kiểm thử cũng được thực hiện với chức năng đăng ký, nhóm đã kiểm thử trường hợp người dùng điền đúng thông tin đăng ký và trường hợp email bị trùng nhau.

```

@Test
public void pageLoadCorrectly() {
    String registerTitle = new RegisterPage(driver).
    getFormTitle();
    assertTrue("Correct Title", registerTitle.equals("Sign
Up"));
}

@Test
public void registerCorrectly() {
    RegisterPage registerPage = new RegisterPage(driver);
    Random rand = new Random();

    int randomId = rand.nextInt(1000);

    String newUser = "testuser" + randomId;
    String newEmail = "testuser" + randomId + "@email.com";
    String newPassword = "password";

    LoginPage loginPage = registerPage.attemptRegister(
newUser, newEmail, newPassword, newPassword);
    loginPage.attemptLogin(newEmail, newPassword);
}

@Test
public void registerWithExistingEmail() {
    RegisterPage registerPage = new RegisterPage(driver);
    assertThrows(Exception.class,
    () -> registerPage.attemptRegister("newUser",
EXISTING_EMAIL, "password", "password"));
}

```

Listing 6: Ca kiểm thử chức năng đăng ký

Về chức năng chia sẻ file, nhóm thực hiện kiểm thử tính năng upload file. Một file rỗng có tên ngẫu nhiên sẽ được tạo ra và upload lên server. Sau đó, chương trình sẽ kiểm tra file vừa được upload đã thực sự được upload hay chưa. Bằng cách kiểm tra tên của file đó có tồn tại trong danh sách các file trên server, nếu tồn tại thì ca kiểm thử đó sẽ pass.

```
@Test
public void pageLoadCorrectly() {
    new HomePage(driver).goToFileListPage();
}

@Test
public void uploadFile() {
    FileListPage flPage = new HomePage(driver).
goToFileListPage();
    File randomFile = createRandomlyNamedFile();
    flPage.uploadFile(randomFile.getAbsolutePath(), "Test
File For You And Me");

    // check if the file has been successfully uploaded
    flPage.getFileByName(randomFile.getName(), true);
    randomFile.delete();
}
```

Listing 7: Ca kiểm thử gửi file

Với tính năng trò chuyện, nhóm sẽ kiểm thử việc gửi và hiển thị tin nhắn. Ở đầu ca kiểm thử, chương trình sẽ thực hiện đăng nhập và truy cập vào cuộc trò chuyện tại lớp học đầu tiên. Với chức năng hiển thị, chương trình sẽ kiểm tra xem có tồn tại tin nhắn trong cuộc trò chuyện này hay không. Mặt khác, với chức năng gửi tin nhắn, chương trình sẽ gửi một tin nhắn ngẫu nhiên, đợi một khoảng thời gian nhất định và kiểm tra xem tin nhắn gần đây nhất có giống với tin vừa được gửi đi hay không.

```
private static void goToFirstClassConvo() {
    new WebDriverWait(driver, 3).until(driver -> driver.
findElement(By.className("itemClass")));
    new HomePage(driver).goToFirstClassConvo();
    new WebDriverWait(driver, 3).until(driver -> driver.
findElement(By.className("itemMessage")));
}

@Test
public void pageLoadCorrectly() {
```

```

    new HomePage(driver);
}

@Test
public void showMessage() throws Exception {
    goToFirstClassConvo();
}

@Test
public void sendMessage() throws Exception {
    goToFirstClassConvo();
    SendMessagePage smPage = new SendMessagePage(driver);

    Random rand = new Random();
    int randomId = rand.nextInt(10000);
    String newMessage = "Test Message " + randomId;
    smPage.sendMessage(newMessage);

    Thread.sleep(1500);

    String lastMessage = smPage.getLastMessage();
    assertTrue(lastMessage.equals(newMessage));
}

```

Listing 8: Ca kiểm thử gửi tin nhắn

4.3.3 Kiểm thử trên nhiều nền tảng với Selenium Grid

Để có thể kiểm thử trên nhiều trình duyệt ở các phiên bản khác nhau, cũng như kiểm thử ứng dụng web trên các hệ điều hành khác, nhóm sử dụng công cụ Selenium Grid. Grid cho phép chúng ta thực thi ca kiểm thử trên nhiều môi trường khác nhau, giúp giảm thiểu rủi ro phần mềm gặp lỗi trên hệ điều hành hay trình duyệt khác.

4.4 Thực thi ca kiểm thử

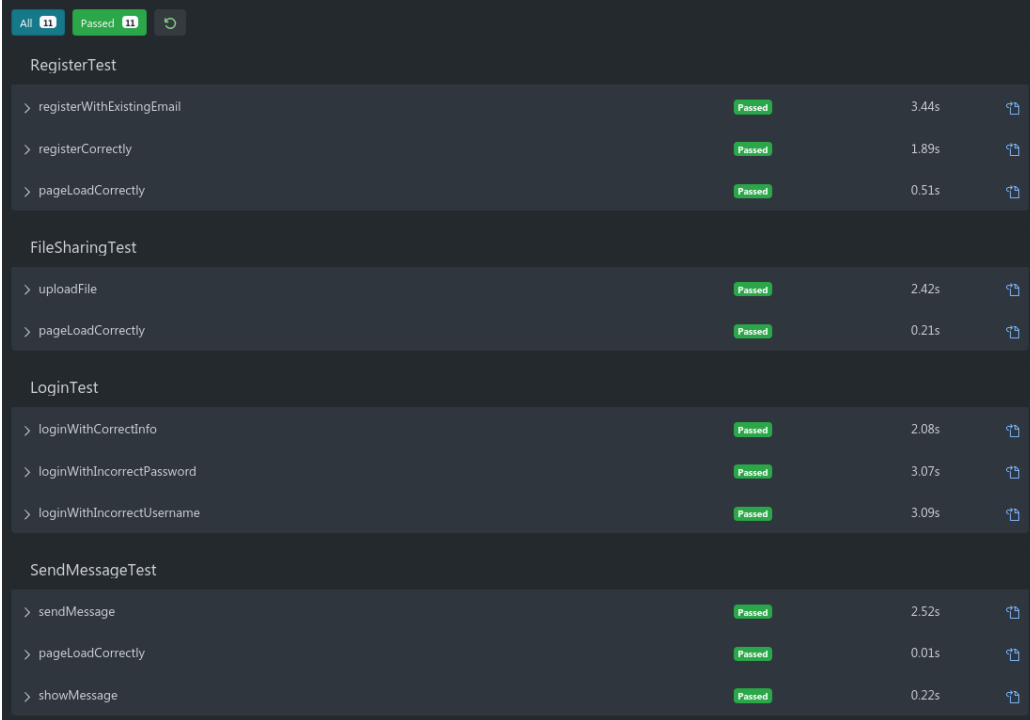
Để thực thi toàn bộ ca kiểm thử với Maven, người dùng chỉ cần chạy câu lệnh sau trong terminal.

```
mvn test
```

Listing 9: Chạy toàn bộ ca kiểm thử

4.4.1 Kết quả kiểm thử

Dưới đây là kết quả kiểm thử với 13 ca kiểm thử trong chương trình.



| | | |
|------------------------------|-----------|-------|
| All 11 | Passed 11 | ↺ |
| RegisterTest | | |
| > registerWithExistingEmail | Passed | 3.44s |
| > registerCorrectly | Passed | 1.89s |
| > pageLoadCorrectly | Passed | 0.51s |
| FileSharingTest | | |
| > uploadFile | Passed | 2.42s |
| > pageLoadCorrectly | Passed | 0.21s |
| LoginTest | | |
| > loginWithCorrectInfo | Passed | 2.08s |
| > loginWithIncorrectPassword | Passed | 3.07s |
| > loginWithIncorrectUsername | Passed | 3.09s |
| SendMessageTest | | |
| > sendMessage | Passed | 2.52s |
| > pageLoadCorrectly | Passed | 0.01s |
| > showMessage | Passed | 0.22s |

Hình 9: Kết quả kiểm thử

5 Kết luận

Như vậy, ta có thể thấy rằng Selenium là một framework mạnh mẽ cho kiểm thử tự động. Qua bộ công cụ của Selenium, người dùng có thể thực hiện kiểm thử nhanh chóng, hiệu quả và đem lại độ chính xác cao. Hơn nữa, Selenium là một framework có mã nguồn mở và luôn được cộng đồng hỗ trợ và phát triển. Tuy nhiên, Selenium có nhược điểm là không phù hợp với người không có kinh nghiệm về lập trình. Nếu chỉ sử dụng được Selenium IDE thì hiệu quả của việc kiểm thử sẽ không cao. Do đó, người kiểm thử bắt buộc phải biết ít nhất một ngôn ngữ lập trình mà Selenium hỗ trợ để có thể viết được các ca kiểm thử một cách bài bản, chuyên nghiệp.