

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: NGHIÊN CỨU THƯ VIỆN HỖ TRỢ KIỂM TỰ ĐỘNG
SELENIUM VÀ ỨNG DỤNG VÀO XÂY DỰNG PHẦN MỀM
KIỂM THỬ TỰ ĐỘNG CHO WEB ATALINK

Giảng viên hướng dẫn: Th.S NGUYỄN LÊ MINH

Sinh viên thực hiện: TÔ VÕ NHƯ QUỲNH

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa: 59

Tp. Hồ Chí Minh, ngày 18 tháng 06 năm 2022

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: NGHIÊN CỨU THƯ VIỆN HỖ TRỢ KIỂM TỰ ĐỘNG
SELENIUM VÀ ỨNG DỤNG VÀO XÂY DỰNG PHẦN MỀM
KIỂM THỬ TỰ ĐỘNG CHO WEB ATALINK

Giảng viên hướng dẫn: Th.S NGUYỄN LÊ MINH

Sinh viên thực hiện: TÔ VÕ NHƯ QUỲNH

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa: 59

Tp. Hồ Chí Minh, ngày 18 tháng 06 năm 2022

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP
BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 5951071089

Họ tên SV: Tô Võ Như Quỳnh

Khóa: 59

Lớp: Công nghệ thông tin

1. **Tên đề tài:** Nghiên cứu thư viện hỗ trợ kiểm thử tự động Selenium và ứng dụng vào xây dựng phần mềm kiểm thử tự động cho web Atalink.
2. **Mục đích, yêu cầu**
 - a. **Mục đích:** Nâng cao hiệu suất phát hiện lỗi, tái sử dụng test case, tiết kiệm nguồn lực.
 - b. **Yêu cầu:**
 - **Yêu cầu chức năng:**
 - Test được những chức năng về thao tác người dùng, luồng hoạt động, kiểm tra input của người dùng, kiểm tra thông báo, ...
 - Hiện thị case fail và lý do cụ thể, tính tỉ lệ pass/ fail và đánh giá bộ test case...
Mở rộng: Import file test case, test được nhiều web khác nhau, chạy trên nhiều browser(không chạy cùng 1 lúc được)...
 - **Yêu cầu phi chức năng:** Giao diện hợp lý, dễ sử dụng.
3. **Nội dung thực hiện:** Tìm hiểu về selenium và xây dựng hệ thống automation test cho web Quản trị chuỗi cung ứng ATALINK.
4. **Công nghệ, công cụ và ngôn ngữ lập trình:**
Selenium, Java, TestNG framework, maven project, cấu trúc design pattern: page object model.
5. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng:**
Xây dựng được hệ thống automation test giúp tester tiết kiệm được thời gian, tài nguyên, tái sử dụng test case.
Hiểu và áp dụng được selenium, quản lý test report, quản lý quy trình test, lập lịch test.

6. Kế hoạch thực hiện:

- Tuần 1: Tham khảo và chọn đề tài
- Tuần 2, 3, 4, 5: Tìm hiểu về selenium
- Tuần 5, 6, 7, 8: Xây dựng bộ data test case
- Tuần 9, 10, 11, 12, 13: Xây dựng hệ thống
- Tuần 14, 15: Viết báo cáo

7. Giáo viên và cán bộ hướng dẫn

Họ tên: ThS. NGUYỄN LÊ MINH

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại:

Email: nlminh@utc2.edu.vn

Ngày tháng năm 2021
Trưởng BM Công nghệ Thông tin

Đã giao nhiệm vụ TKTN
Giảng viên hướng dẫn

Nguyễn Lê Minh

LỜI CẢM ƠN

Lời đầu tiên, em xin cảm ơn tập thể giảng viên bộ môn Công nghệ thông tin trường Đại học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh đã giảng dạy và giúp đỡ em trong suốt quá trình học tập tại trường. Thầy cô đã giúp em trang bị các kiến thức chuyên môn và kỹ năng nghề nghiệp để có thể áp dụng vào thực tiễn.

Em xin chân thành cảm ơn sự giúp đỡ tận tình của thầy Nguyễn Lê Minh, người thầy cho em những chỉ dạy, những lời góp ý chân thành về kiến thức chuyên môn để em có thể hoàn thành tốt nhất đề tài của mình.

Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn là nguồn động viên to lớn giúp em vượt qua những khó khăn trong suốt quá trình học tập.

Mặc dù đã cố gắng hoàn thiện đề tài với tất cả nỗ lực của bản thân, nhưng chắc sẽ không tránh khỏi những thiếu sót. Kính mong thầy cô có thể đóng góp ý kiến để em có thể hoàn thiện kiến thức của bản thân.

Cuối cùng, em xin kính chúc quý thầy cô luôn mạnh khỏe, thành công trong sự nghiệp. Em xin chân thành cảm ơn.

Trân trọng!

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm

Giáo viên hướng dẫn

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	i
LỜI CẢM ƠN.....	iii
NHẬN XÉT CỦA GIÁO VIÊN	iv
DANH MỤC CHỮ VIẾT TẮT	vii
BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ	viii
CHƯƠNG 1. MỞ ĐẦU	1
1.1 Tổng quan về đề tài:	1
1.2 Mục tiêu nghiên cứu.....	1
1.3 Phạm vi thực hiện.....	1
1.4 Cấu trúc báo cáo thực tập tốt nghiệp	2
1.4.1 Chương 1: Mở đầu	2
1.4.2 Chương 2: Nghiên cứu thư viện hỗ trợ kiểm thử tự động Selenium và các công nghệ sử dụng trong chương trình	2
1.4.3 Chương 3: Xây dựng phần mềm kiểm thử tự động cho web Atalink.....	2
1.4.4 Chương 4: Kết luận	2
CHƯƠNG 2. NGHIÊN CỨU THƯ VIỆN HỖ TRỢ KIỂM THỬ TỰ ĐỘNG SELENIUM VÀ CÁC CÔNG NGHỆ SỬ DỤNG TRONG CHƯƠNG TRÌNH	3
2.1 Tổng quan về kiểm thử phần mềm.....	3
2.1.1 Tổng quan.....	3
2.1.2 Quy trình chung.....	3
2.2 Tổng quan về thư viện hỗ trợ kiểm thử tự động Selenium	3
2.2.1 Thư viện mã nguồn mở Selenium	3
2.2.2 Selenium WebDriver.....	7
2.2.3 Ngôn ngữ Java.....	11
2.3 Tổng quan về Page Object Model (POM).....	13
2.4 Tổng quan về TestNG	14

CHƯƠNG 3. XÂY DỰNG PHẦN MỀM KIỂM THỬ TỰ ĐỘNG CHO WEB ATALINK	16
3.1 Giới thiệu sơ lược về web quản trị chuỗi cung ứng Atalink	16
3.2 Hệ thống automation test cho web Atalink	17
3.2.1 Bài toán hiện nay.....	17
3.2.2 Thiết kế hệ thống.....	17
3.3.3 Các chức năng của phần mềm.....	18
CHƯƠNG 4. KẾT LUẬN.....	33
4.1 Kết quả đạt được.....	33
4.2 Tính năng phát triển trong tương lai.....	34

DANH MỤC CHỮ VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	POM	Page Object Model	
2	QTP	Quick Test Pro	
3	IDE	Integrated Development Environment	
4	RC	Remote Control	
5	UI	User Interface	
6	JRE	Java Runtime Environment	
7	JDK	Java Development Kit	
8	NG	Next Generation	
9	Regression test	Test lại toàn bộ test case	
10	Smoke test	Test ngẫu nhiên 1 số case	
11	CPU	Central Processing Unit	
12	B2B	Business to Business	
13	Test Case	Trường hợp kiểm thử	
14	Test Report	Báo cáo kết quả kiểm thử	
15	Locator	Vị trí của các thành phần trong web	
16	Manual Testing	Kiểm thử thủ công	
17	ScreenShot	Chụp ảnh màn hình	
18	Record video	Ghi hình	

BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ

Hình 1: Sơ đồ các thành phần của thư viện Selenium.....	4
Hình 2: Giao diện của công cụ Selenium IDE	4
Hình 3: Thành phần của Selenium RC	5
Hình 4: Thành phần của Selenium WebDriver	6
Hình 5: Thành phần của Selenium Grid.....	6
Hình 6: Locator ID	7
Hình 7: Locator Name	7
Hình 8: Locator LinkText.....	8
Hình 9: Locator TagName.....	8
Hình 10: Locator ClassName	8
Hình 11: Locator CSS Selector	9
Hình 12: Locator Xpath.....	9
Hình 13: Thành phần của Page Object Model	13
Hình 14 Sơ đồ hoạt động TestNG	14
Hình 15: Giới thiệu về ATALINK	16
Hình 16: User Case Diagram.....	18
Hình 17: Cấu trúc Test Case.....	19
Hình 18: Test case kiểm tra notification đã chuyển thành code.....	19
Hình 19: Test case kiểm tra đăng nhập chuyển sang code	19
Hình 20: Cấu trúc POM trong chương trình.....	20
Hình 21: Hàm dừng lại trong chương trình	21
Hình 22: Cấu trúc của file dữ liệu test.....	22
Hình 23: Giao diện quản lý test case	23
Hình 24: Tính năng chạy toàn bộ test case.....	23
Hình 25: Tính năng chạy một số test case được chọn	24
Hình 26: Tính năng chỉ chạy test case fail	24
Hình 27: Tính năng sắp xếp test case	24
Hình 28: Tính năng hiển thị lịch sử chạy test case.....	25
Hình 29: Tính năng Import, Export kết quả test.....	25

Hình 30: Cài đặt trong quản lý test case.....	25
Hình 31: Test trên Google Chrome	26
Hình 32: Test trên Microsoft Edge	26
Hình 33: Cấu trúc file Log.....	27
Hình 34: Cấu trúc Log console.....	27
Hình 35: Lưu trữ video ghi được của test case fail	28
Hình 36: Mở video bằng VLC media player.....	29
Hình 37: Giao diện test report	29
Hình 38: Lọc test case Pass và Fail	30
Hình 39: Chi tiết test case.....	30
Hình 40: Ảnh chụp màn hình	31
Hình 41: Giao diện test report ở tab biểu đồ	32

CHƯƠNG 1. MỞ ĐẦU

1.1 Tổng quan về đề tài:

Đề tài của em xoay quanh việc nghiên cứu thư viện hỗ trợ kiểm thử tự động Selenium. Bên cạnh đó còn sử dụng thêm ngôn ngữ lập trình Java, cấu trúc Design Pattern (POM), thư viện TestNG và HTML cơ bản để ứng dụng vào xây dựng phần mềm kiểm thử tự động cho web Atalink.

1.2 Mục tiêu nghiên cứu

- Hiểu và nắm rõ được cách hoạt động của 1 quy trình kiểm thử tự động.
- Biết cách sử dụng các cấu trúc, framework, hàm,... trong thư viện hỗ trợ kiểm thử tự động Selenium.
- Xây dựng phần mềm kiểm thử tự động cho web Atalink
 - Xây dựng được bộ test case và dữ liệu test cho web Atalink
 - Hiểu và sử dụng được Design Pattern: Page Object Model (POM)
 - Tạo hàm dùng lại cho test case
 - Import dữ liệu từ file Excel
 - Kiểm thử trên nhiều trình duyệt: Google Chrome, Microsoft Edge, Firefox
 - Hiện thị test case pass / fail, những test case fail sẽ được screenshot và record video
 - Tạo file log quản lý thời gian kiểm thử
 - Quản lý được web test report

1.3 Phạm vi thực hiện

- Nghiên cứu về thư viện hỗ trợ kiểm thử tự động Selenium
 - Tổng quan về thư viện hỗ trợ kiểm thử tự động Selenium: Khái niệm, nguồn gốc ra đời, các thành phần,...
 - Giới thiệu về thành phần được sử dụng được chọn sử dụng trong phần mềm, ưu nhược điểm,...
- Tìm hiểu domain web Atalink để tạo bộ test case và xây dựng phần mềm test phù hợp cho web

1.4 Cấu trúc báo cáo thực tập tốt nghiệp

1.4.1 Chương 1: Mở đầu

1.4.2 Chương 2: Nghiên cứu thư viện hỗ trợ kiểm thử tự động Selenium và các công nghệ sử dụng trong chương trình

1.4.3 Chương 3: Xây dựng phần mềm kiểm thử tự động cho web Atalink

1.4.4 Chương 4: Kết luận

CHƯƠNG 2. NGHIÊN CỨU THƯ VIỆN HỖ TRỢ KIỂM THỬ TỰ ĐỘNG SELENIUM VÀ CÁC CÔNG NGHỆ SỬ DỤNG TRONG CHƯƠNG TRÌNH

2.1 Tổng quan về kiểm thử phần mềm

2.1.1 Tổng quan

- Kiểm thử phần mềm là một giai đoạn trong quy trình phát triển phần mềm để đảm bảo độ tin cậy và chất lượng của phần mềm.
- Các mục tiêu chính của kiểm thử phần mềm:
 - Phát hiện càng nhiều lỗi càng tốt trong thời gian kiểm thử.
 - Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả yêu cầu của nó.
 - Xác thực chất lượng kiểm thử phần mềm tiết kiệm chi phí và nguồn lực.
 - Tạo các kịch bản kiểm thử (test case) chất lượng cao, thực hiện kiểm thử hiệu quả và tạo ra các báo cáo vấn đề đúng và hữu dụng

2.1.2 Quy trình chung

Các hoạt động cơ bản của quy trình phát triển phần mềm:

- Đặc tả phần mềm: Định nghĩa được các chức năng, điều kiện hoạt động của phần mềm.
- Phát triển phần mềm: Là quá trình xây dựng các đặc tả.
- Đánh giá phần mềm: Phần mềm phải được đánh giá để chắc chắn rằng ít nhất có thể thực hiện những gì mà tài liệu đặc tả yêu cầu.
- Tiến hóa phần mềm: Đây là quá trình hoàn thiện các chức năng cũng như giao diện để ngày càng hoàn thiện phần mềm cũng như các yêu cầu đưa ra từ phía khách hàng.

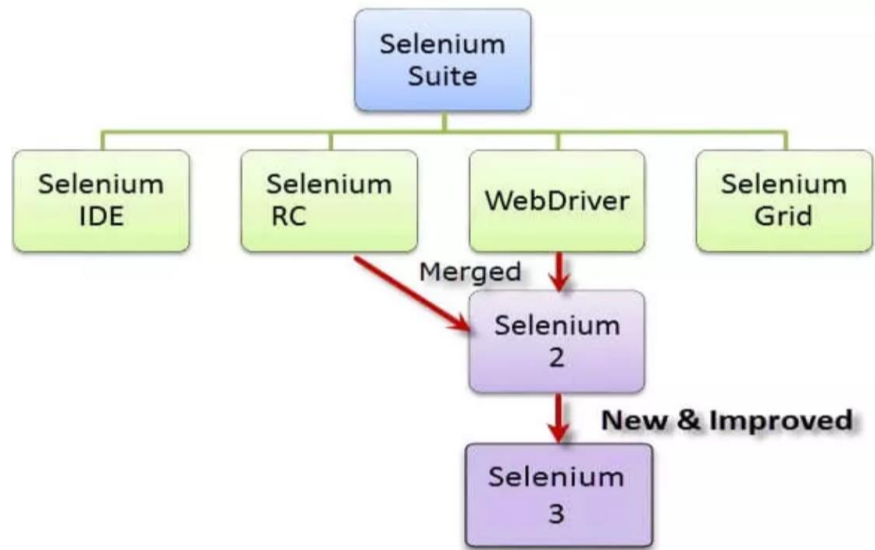
2.2 Tổng quan về thư viện hỗ trợ kiểm thử tự động Selenium

2.2.1 Thư viện mã nguồn mở Selenium

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được

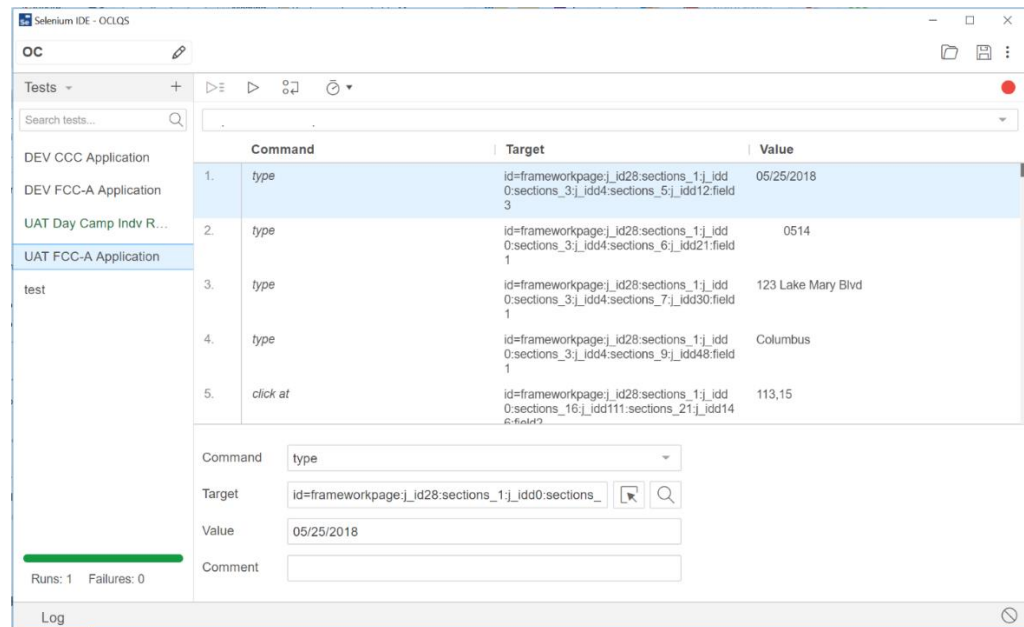
thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức.

Selenium có 4 thành phần:



Hình 1: Sơ đồ các thành phần của thư viện Selenium

- Selenium Integrated Development Environment (IDE)

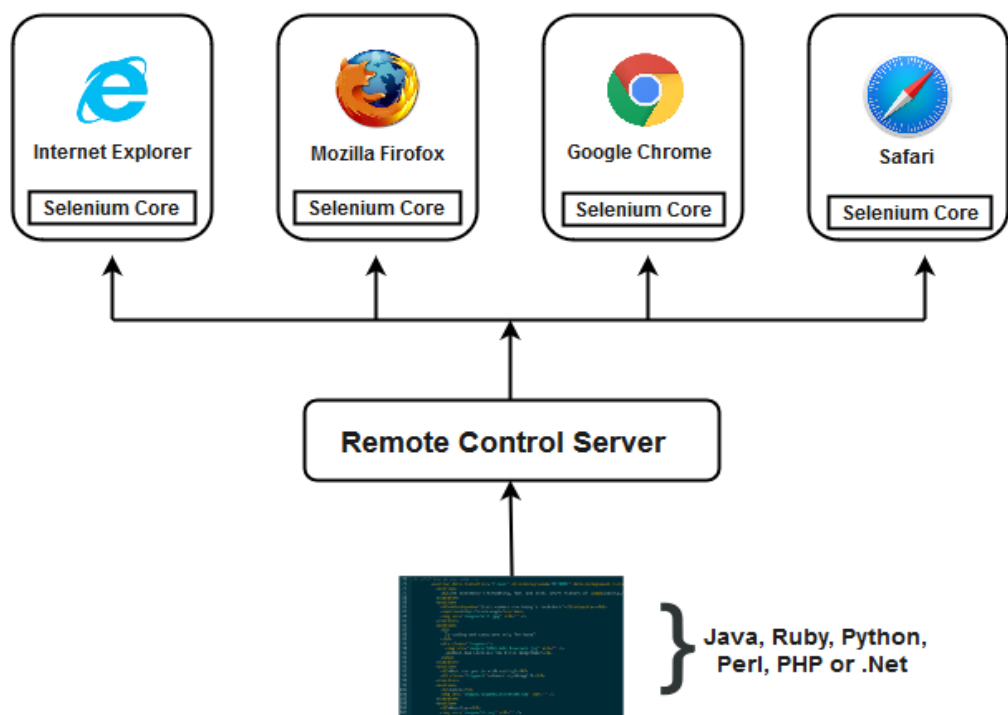


Hình 2: Giao diện của công cụ Selenium IDE

Là framework đơn giản nhất trong bộ Selenium và nó là phần dễ nhất để học. Nó là một Firefox plugin mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì sự đơn giản của nó, nên chỉ sử dụng công cụ IDE làm công cụ tạo mẫu. Nếu bạn muốn các trường hợp kiểm thử nâng cao thì bạn có thể sử dụng hoặc là Selenium RC hoặc là WebDriver.

- Selenium Remote Control (RC)

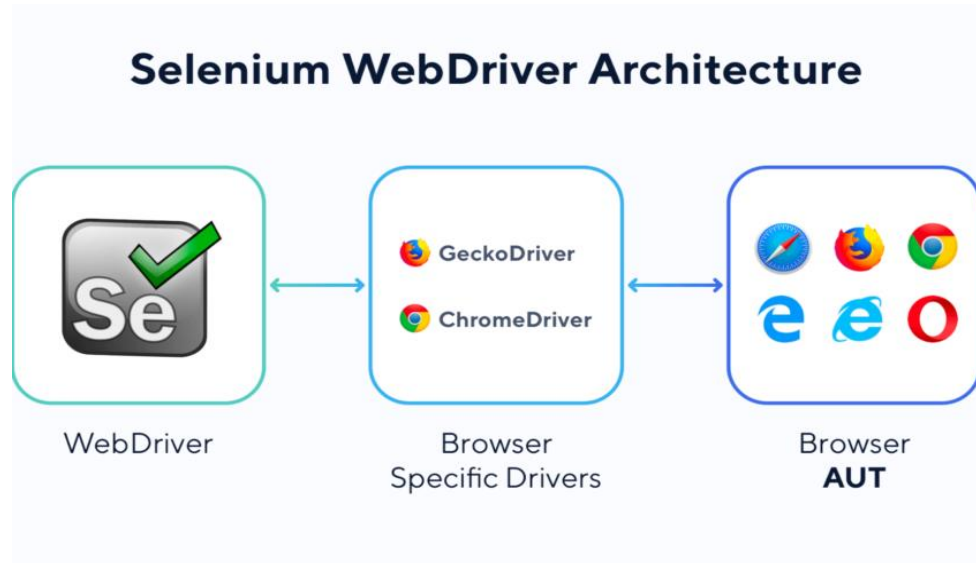
Selenium RC Architecture



Hình 3: Thành phần của Selenium RC

Là framework kiểm thử hàng đầu của dự án Selenium trong một thời gian dài. Đây là công cụ kiểm tra web tự động đầu tiên cho phép người dùng sử dụng ngôn ngữ lập trình mà họ thích. Kể từ phiên bản 2.25.0, RC có thể hỗ trợ các ngôn ngữ lập trình sau: Java, C#, PHP, Python, Perl, Ruby.

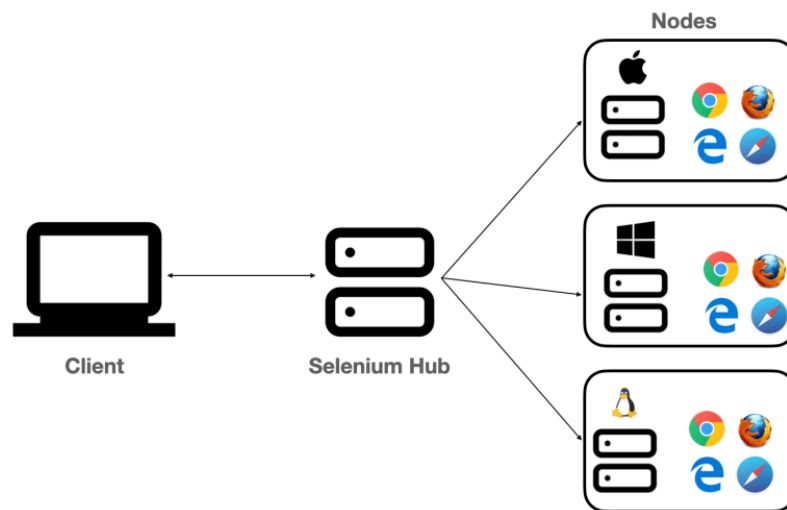
- Selenium WebDriver



Hình 4: Thành phần của Selenium WebDriver

Chúng minh rằng nó tốt hơn cả Selenium IDE và Selenium RC ở nhiều khía cạnh. Nó thực hiện một cách tiếp cận hiện đại và ổn định hơn trong việc tự động hóa các hành động của trình duyệt. WebDriver, không giống như Selenium RC, không dựa vào JavaScript cho tự động hóa. Nó kiểm soát trình duyệt bằng cách giao tiếp trực tiếp với nó. Các ngôn ngữ được hỗ trợ giống như ngôn ngữ trong Selenium RC là Java, C#, PHP, Python, Perl, Ruby.

- Selenium Grid



Hình 5: Thành phần của Selenium Grid

Là một công cụ được sử dụng cùng với Selenium RC để chạy thử nghiệm song song trên các máy khác nhau và các trình duyệt khác nhau cùng một lúc. Thực thi song song có nghĩa là chạy nhiều thử nghiệm cùng một lúc.

2.2.2 Selenium WebDriver

a. Tổng quan về đối tượng UI (Locator)

- Trong Manual Testing ta có thể tự biết được nhập dữ liệu vào đâu, click vào button, checkbox nào, nhận biết được item trước và sau thay đổi như thế nào,...Nhưng đối với automation cách để máy có thể hiểu được sẽ thao tác vào link, button, checkbox... thì sẽ dựa vào Locator.
- Mỗi phần tử HTML trong trang web đều được xác định bằng Locator. Locator giúp phân biệt được các đối tượng UI trên phần mềm.
- Các cách lấy Locator:
 - Xác định phần tử web theo ID

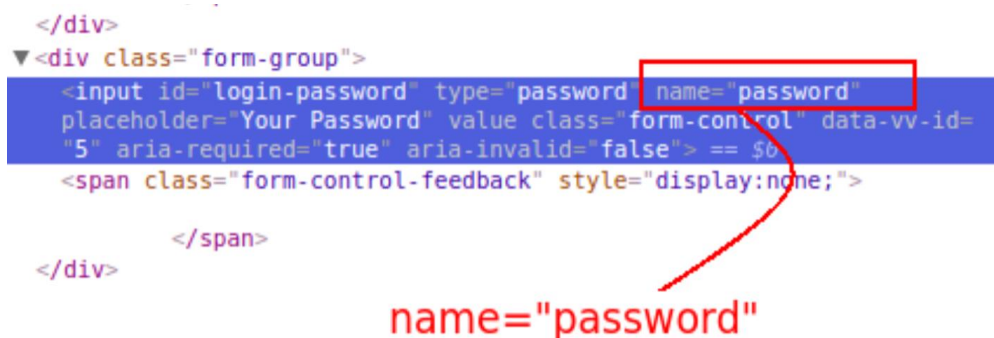


```
▼ <div class="form-group has-danger">
  <input id="login-username" name="username" placeholder="Your Name (E-mail Address or Username)" value class="form-control" data-vv-id="4" aria-required="true" aria-invalid="true"> == $0
  <span class="form-control-feedback" style>
    The username field is required.
  </span>
```

id="login-username"

Hình 6: Locator ID

- Xác định phần tử web theo Name



```
</div>
▼ <div class="form-group">
  <input id="login-password" type="password" name="password" placeholder="Your Password" value class="form-control" data-vv-id="5" aria-required="true" aria-invalid="false"> == $0
  <span class="form-control-feedback" style="display:none;">
  </span>
</div>
```

name="password"

Hình 7: Locator Name

- Xác định phần tử web theo LinkText (bộ định vị liên kết văn bản)



Hình 8: Locator LinkText

- Xác định phần tử web theo TagName



Hình 9: Locator TagName

- Xác định phần tử web theo ClassName



Hình 10: Locator ClassName

- Xác định phần tử web theo CSS Selector

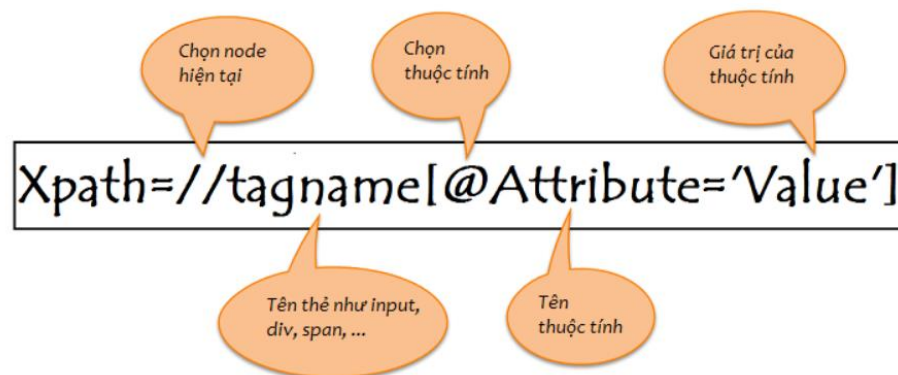
▶ box-sizing	border-box
▶ color	rgb(255, 255, 255)
▶ cursor	pointer
▶ display	inline-block
▶ fill	rgb(255, 255, 255)
▶ font-family	Roboto, Arial, "sans-serif"
▶ font-feature-settings	"tnum", "tnum"
▶ font-size	24px
▶ font-style	normal
▶ font-variant-caps	normal
▶ font-variant-east-asian	normal
▶ font-variant-ligatures	normal

Hình 11: Locator CSS Selector

- Xác định phần tử web theo Xpath

Trong Selenium nếu không thể tìm kiếm các elements bởi các định vị như id, class, name... thì hoàn toàn có thể nghĩ tới xpath để tìm các elements đó trên web, xpath cũng là phương thức được đông đảo người dùng sử dụng để tìm kiếm các thành phần trên trang web.

Xpath được định nghĩa như một XML path. Nó là cú pháp hay ngôn ngữ để tìm kiếm bất kỳ element nào trên trang web sử dụng XML path expression. Cú pháp cơ bản của Xpath:



Hình 12: Locator Xpath

Có 2 loại Xpath:

- Xpath tuyệt đối: Xpath tuyệt đối bắt đầu bằng dấu gạch chéo đơn `/`, cho phép xác định một đường dẫn tuyệt đối đến đối tượng UI.

- Xpath tương đối: Xpath tương đối bắt đầu bằng 2 dấu gạch chéo "/", cho phép xác định một đối tượng UI ở bất kỳ đâu trên trang web, không cần bắt đầu bởi thẻ html trong đường dẫn.

b. Các thư viện cần thiết để chạy Selenium WebDriver

- Selenium – java
- Selenium – server
- Selenium hỗ trợ các trình duyệt (ChromeDriver, FireFoxDriver, Edge Driver,...)
- Selenium – ie – driver

Để cài được những thư viện này thì phải dùng project Apache Maven. Apache maven là một chương trình quản lý dự án cho phép các developers có thể quản lý về version, các dependencies (các thư viện sử dụng trong dự án), quản lý build, tự động download javadoc & source, ...

c. Ưu điểm và nhược điểm của Selenium WebDriver:

- Ưu điểm:

- Mã nguồn mở không tốn chi phí khi sử dụng.
- Tương thích với nhiều trình duyệt như là: Opera, Yandex, Firefox, Internet Explorer, Chrome.
- Hỗ trợ đa ngôn ngữ lập trình phổ biến hiện nay (C#, Java, Python, Ruby, Perl,...).
- Thân thiện, dễ sử dụng, câu lệnh dễ hiểu, cung cấp cho người dùng nhiều tài liệu, cộng đồng sử dụng đông đảo.
- Tốc độ thực thi nhanh nhờ vào mỗi trình duyệt web đều có một công cụ hỗ trợ khác nhau dành cho Selenium WebDriver. Google Chrome thì có ChromeDriver, Firefox thì có FireFoxDriver,...

- Nhược điểm:

- Chỉ hỗ trợ ứng dụng web
- Đòi hỏi kinh nghiệm lập trình

- Khó bảo trì và mở rộng
- Thiếu tính năng tự báo cáo

2.2.3 Ngôn ngữ Java

- Ưu điểm:

- Dễ học hơn c ++.
- Dễ đọc mã hơn c ++.
- Phát triển nhanh hơn và có khả năng ít lỗi hơn (do quản lý bộ nhớ ít hơn) so với c ++.
- Hầu hết các nền tảng chéo kể từ khi nó được thiết kế cho nó và Sun tạo ra JRE và JDK cho hầu hết các nền tảng.
- Tất cả mọi thứ được truyền bằng tham chiếu ngoại trừ các loại giá trị, theo mặc định.
- Công cụ thu gom rác dọn dẹp đồ vật một khi chúng không còn được sử dụng, vì vậy bạn không phải tự mình theo dõi mọi thứ.
- Chương trình được biên dịch và tối ưu hóa cho nền tảng.
- Dễ dàng tạo nhiều luồng và bạn có thể khai báo các chức năng được đồng bộ hóa để chỉ một luồng có thể có trong một luồng (IIRC).
- Bạn có thể sử dụng Wait () và notifyAll () và thông báo () để tạm dừng các luồng để chờ tín hiệu và đánh thức chúng lại.
- Các IDE miễn phí khác nhau có sẵn trên tất cả các nền tảng, chẳng hạn như Eclipse.
- Nếu chương trình gặp sự cố, nó sẽ báo cho người dùng biết mã trong đó ở đâu và tại sao, hiển thị dấu vết ngăn xếp bao gồm số dòng (nếu biểu tượng gỡ lỗi được bật hoặc một cái gì đó tương tự) hoặc có thể được đổ vào tệp nhật ký lỗi.
- Các chương trình có thể được đóng gói thành các tệp jar với các tệp dữ liệu khác nhau trong các tệp jar khác nhau và được người dùng truy cập bằng Java WebStart, sẽ tự động cập nhật chương trình và mọi tệp jar cần thiết khác lên phiên bản mới nhất, chỉ tải xuống các tệp jar đã thay đổi.

- Mạng dễ dàng hơn / nhiều nền tảng hơn c ++.
- Các tệp được đọc và ghi dưới dạng big endian trên tất cả các nền tảng (là thứ tự byte mạng) đảm bảo rằng giao tiếp giữa các chương trình java và tải tệp không bao giờ yêu cầu hoán đổi thứ tự byte (trừ khi nói chuyện với chương trình không phải java hoặc sử dụng tệp được viết bởi một).

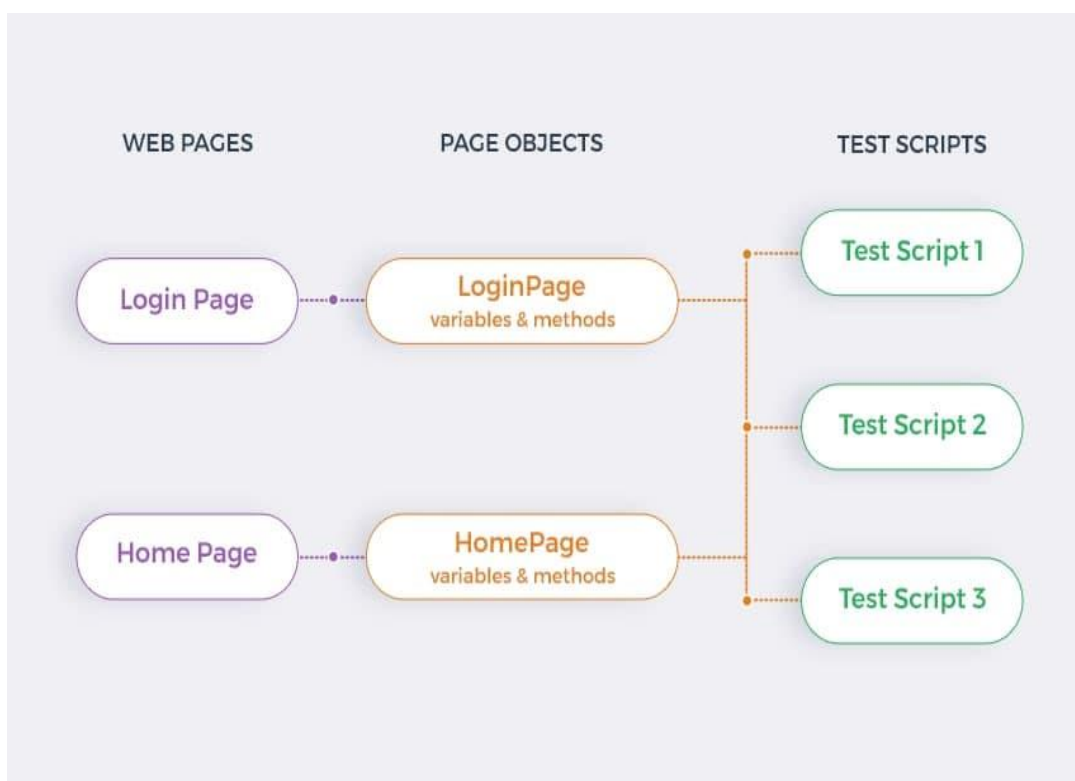
- Nhược điểm:

- Sử dụng nhiều bộ nhớ hơn c ++ hoặc c #.
- JNI thật đáng sợ, nhưng nếu chúng tôi là nền tảng chéo, có lẽ chúng tôi sẽ không sử dụng nó, (ngoại trừ công cụ 3d sẽ?).
- Trình thu gom rác sử dụng chu kỳ và bộ nhớ CPU - (nhưng hầu như không có, ít nhất là thời gian CPU).
- Không có số nguyên không dấu.
- Không có con trỏ, không có tham số ref hoặc out (thay thế con trỏ trong c #). Để trả về nhiều thứ từ một hàm java, bạn có xu hướng phải trả về một đối tượng [] đang giữ từng thứ hoặc tạo một lớp chứa từng thứ và trả về một đối tượng của lớp đó.
- Một số thứ đã được đổi tên - const là cuối cùng trong java.
- Java không cho phép người lập chỉ mục hoặc toán tử trên các đối tượng, ví dụ: nếu bạn sử dụng danh sách có thể thay đổi kích thước, v.v., bạn phải thực hiện list.get (index) thay vì list [index] và bạn không thể thực hiện someVector + someOtherVector.
- Java không cho phép bạn quyết định xem một loại nên là một giá trị hay một loại tham chiếu (xác định xem nó nên được truyền theo giá trị hay chuyển qua tham chiếu), không giống như c #.
- Các tệp Jar được sử dụng bởi java webstart phải được ký bằng chứng chỉ. Chứng chỉ ký mã có giá hàng trăm đô la mỗi năm hoặc bạn có thể tự tạo một ký tự và sử dụng nó, điều này sẽ khiến Java Webstart bật lên hộp thoại "Chúng nhận này không thể được xác minh" khi người dùng cố chạy chương trình. (Cách khác, không sử dụng Java

Webstart và chỉ phân phối các tệp jar trực tiếp - yêu cầu người dùng tự kiểm tra các bản cập nhật.)

- Yêu cầu JRE được cài đặt để chạy chương trình, tải xuống vài giờ khi quay số.
- Phải sử dụng java 5 cho generic, 1.4 không có chúng. Java 6 mới nhất và đã được vài năm tuổi, nhưng - Apple phát hành phiên bản java của riêng họ cho người dùng Mac và vẫn chưa phát hành phiên bản java 6 "ổn định" cho người dùng.

2.3 Tổng quan về Page Object Model (POM)



Hình 13: Thành phần của Page Object Model

- POM là một mẫu thiết kế (Design Pattern) có các đặc điểm sau:

- Tạo Kho đối tượng (object repository) cho các phần tử giao diện của web (UI elements).
- Dưới mô hình này, mọi trang web trong ứng dụng đang viết có thể tương ứng với một lớp.

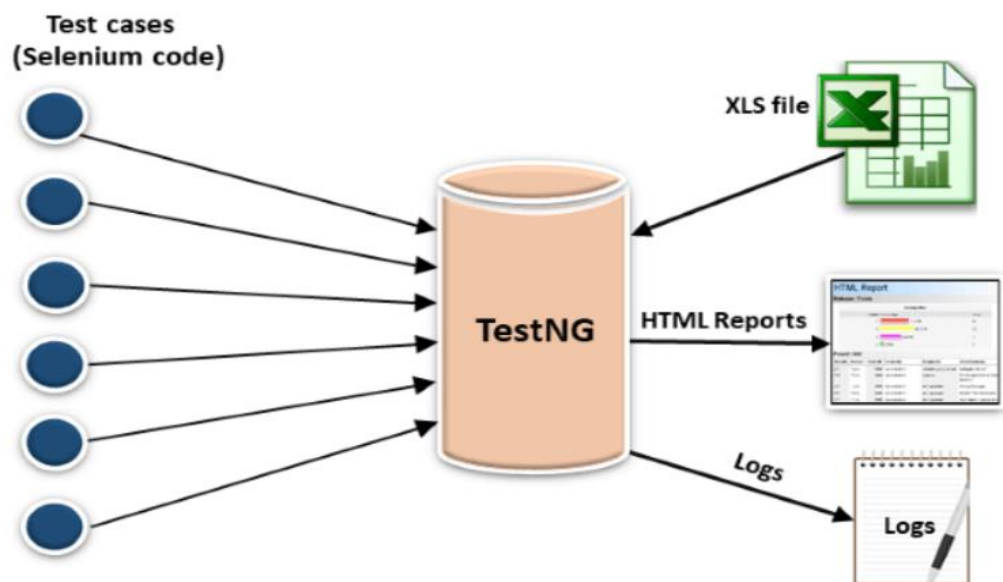
- Lớp này sẽ tìm kiếm tất cả phần tử của web (WebElements) và chỉ chứa các phương thức để thực hiện thao tác trên các phần tử của trang web đó.
- Các phương thức này nên được đặt tên như là nhiệm vụ (task) mà chúng sẽ thực hiện. Nó rất dễ dàng ánh xạ với các hành động (actions) xảy ra trong giao diện với người dùng. Ví dụ, Nếu muốn nhập user cho textbox user trên một trang, tên phương thức đó có thể là “setUserName”.

- Lợi ích của Page Object Model (POM)

- Làm cho code trở nên rõ ràng và dễ hiểu hơn, nó tránh sự lặp lại nhiều lần trong code. Do đó, Mô hình này sẽ trở nên dễ dàng hơn trong việc bảo trì và tái sử dụng.
- Các kho lưu trữ độc lập với các kịch bản test. Vì vậy, chúng ta có thể sử dụng các kho lưu trữ giống nhau cho các mục đích khác nhau với những tool khác nhau. Ví dụ, chúng ta có thể tích hợp POM với TestNG cho việc test chức năng hay Cucumber cho Acceptance Test (Kiểm thử chấp nhận).

2.4 Tổng quan về TestNG

- TestNG là 1 công cụ kiểm thử tự động được viết bằng Java, có chức năng quản lý việc tạo test case, thử tự chạy test case và report sau khi test.



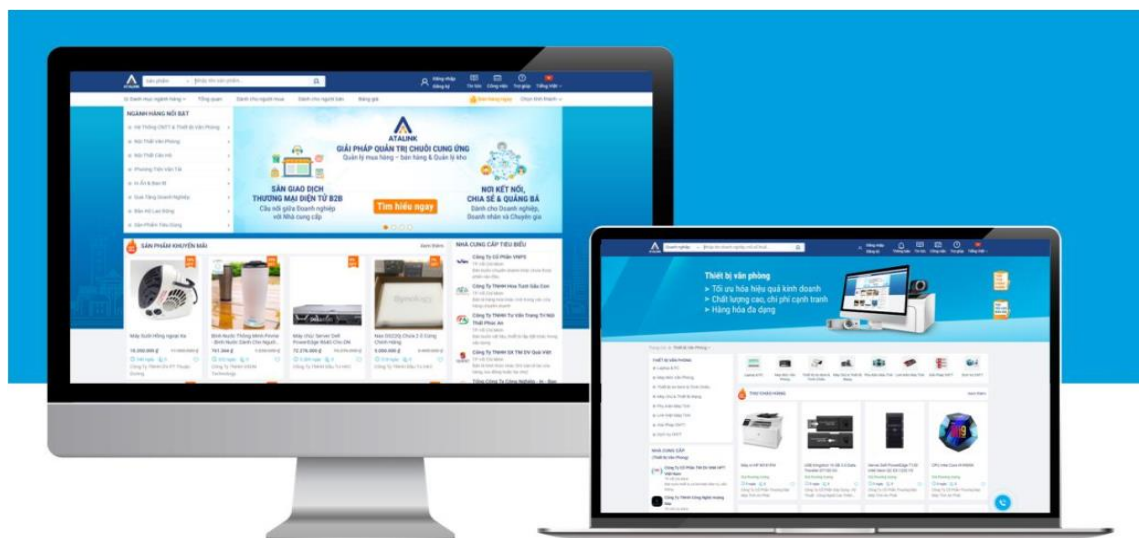
Hình 14 Sơ đồ hoạt động TestNG

- TestNG được xây dựng từ cảm hứng của 2 Framework là JUnit (Java) và NUnit (C#). NG là viết tắt của từ Next Generation.
- Các tính năng nổi bật của TestNG:
 - Mô tả các thiết lập khác nhau khi kiểm thử phần mềm bằng file XML
 - Phân nhóm kiểm thử
 - Tạo mối quan hệ ràng buộc lẫn nhau giữa các module
 - Xác lập thời điểm cho các xử lý trước và sau
 - Cung cấp các chỉ dẫn Annotation – based để nhận diện phương thức test
 - TestNG không yêu cầu có phương thức main ()
 - Các phương thức không cần tĩnh (static)
- Một số ưu điểm của TestNG
 - Nó cho phép tạo ra các bản báo cáo HTML của tiến trình thực thi.
 - Các chú thích giúp việc kiểm thử dễ dàng hơn.
 - Các trường hợp kiểm thử có thể được nhóm lại và được ưu tiên dễ dàng hơn.
 - TestNG có các annotation logic hơn và dễ hiểu hơn.
 - Có thể kiểm thử song song, tạo ra các log bug, tạo data Parameterized action.
 - TestNG cho phép bạn xác định các test case phụ thuộc.

CHƯƠNG 3. XÂY DỰNG PHẦN MỀM KIỂM THỬ TỰ ĐỘNG CHO WEB ATALINK

3.1 Giới thiệu sơ lược về web quản trị chuỗi cung ứng Atalink

- ATALINK - nền tảng thương mại điện tử B2B dành cho cộng đồng Doanh nghiệp vừa và nhỏ muốn bán hàng ở kênh mới; giúp các Doanh nghiệp chủ động sử dụng công nghệ số để mở rộng mạng lưới kinh doanh, nắm bắt cơ hội bán hàng.
- Sàn giao dịch thương mại điện tử B2B ATALINK là nơi kết nối, chia sẻ và quảng bá của các Doanh nghiệp, Doanh nhân và Chuyên gia, giúp Doanh nghiệp nâng tầm năng lực kinh doanh trong thời đại số.



Hình 15: Giới thiệu về ATALINK

- Các màn hình sẽ kiểm thử tự động:
 - Main Menu Bar: Thanh công cụ chính của web
 - Marketplace HomePage: Trang chủ
 - Category level 1: Trang ngành hàng chính
 - Category level 2: Trang ngành hàng con
 - Sign In: Trang đăng nhập tài khoản
 - Sign Up: Trang đăng ký tài khoản
 - List Buying Request: Trang danh sách yêu cầu báo giá của khách hàng
 - List Sales Offer: Trang danh sách thư chào hàng từ nhà cung cấp

- List Product: Trang danh sách sản phẩm được đăng từ các nhà cung cấp
- Detail Buying Request: Trang chi tiết của yêu cầu báo giá
- Detail Sales Offer: Trang chi tiết của thư chào hàng
- Detail Product: Trang chi tiết sản phẩm

3.2 Hệ thống automation test cho web Atalink

3.2.1 Bài toán hiện nay:

- Ngày nay, với sự phát triển mạnh mẽ của công nghệ 4.0 cũng như sự phát triển của nhiều ngành nghề, đặc biệt là các ứng dụng của ngành công nghệ thông tin.
- Ngày càng có nhiều sản phẩm về phần mềm, vì vậy lĩnh vực kiểm thử phần mềm cũng đóng vai trò quan trọng giúp sản phẩm trở nên tốt hơn.
- Ngoài việc kiểm tra thủ công bằng tay, hiện nay kiểm thử tự động hóa còn góp phần nâng cao năng suất kiểm thử, giảm sự nhầm lẫn khi kiểm thử bằng tay trong thời gian dài, và lặp lại nhiều lần. Từ đó, em nảy ra ý tưởng phát triển một phần mềm hỗ trợ kiểm thử tự động để giúp người kiểm thử có thể dễ dàng thực hiện quá trình kiểm tra.

3.2.2 Thiết kế hệ thống

a. Mô tả hệ thống:

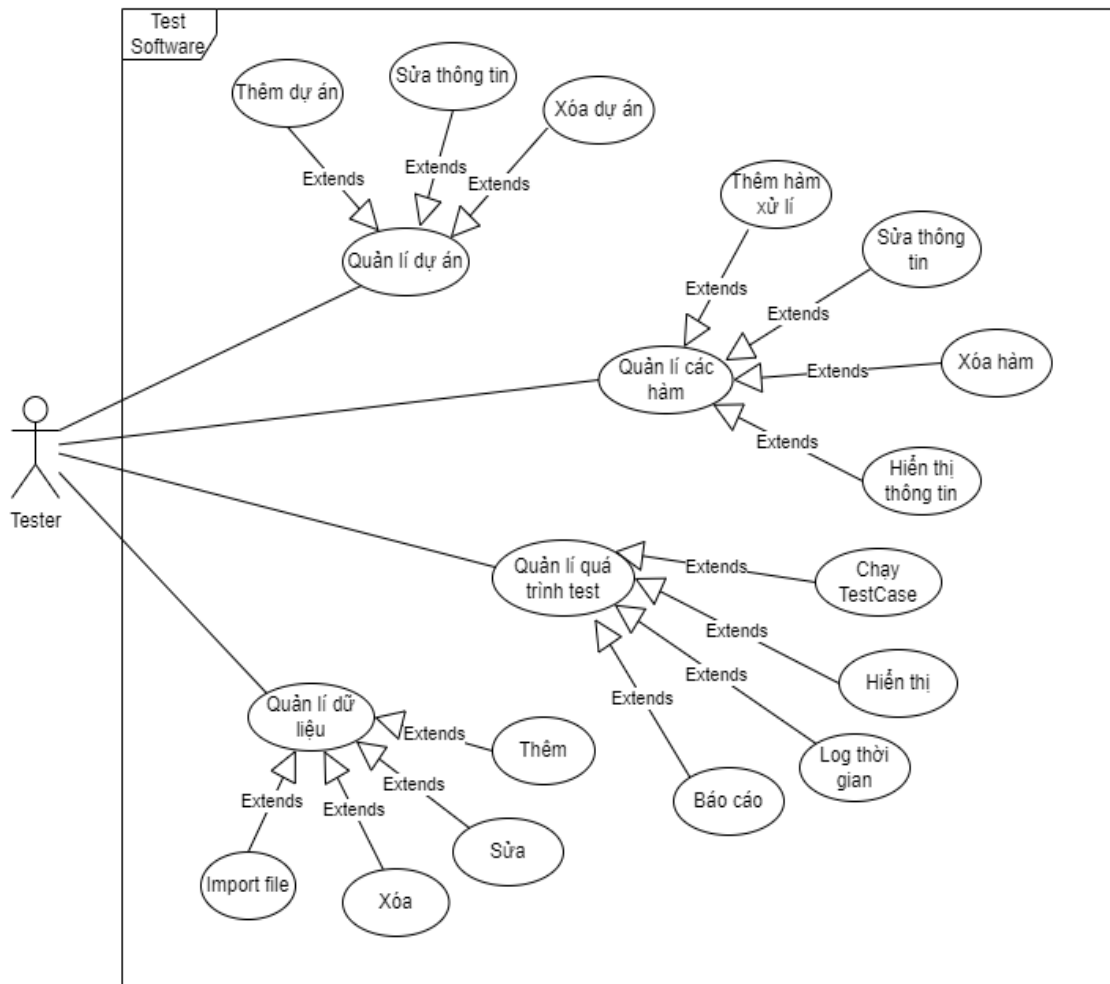
Trong hệ thống, tester sẽ có thể thực hiện các chức năng là quản lý dự án, quản lý các hàm trong dự án, quản lý dữ liệu test của các hàm, quản lý quy trình test. Ở chức năng quản lý dự án sẽ thực thi những hoạt động: Thêm, sửa, xóa project.

Ở chức năng quản lý các hàm trong dự án sẽ thực thi những hoạt động: Thêm, sửa, xóa, hiển thị hàm test.

Ở chức năng quản lý dữ liệu test của các hàm sẽ thực thi những hoạt động: Thêm, sửa, xóa, import file dữ liệu test.

Ở chức năng quản lý quy trình test sẽ thực thi những hoạt động: Chạy test case, hiển thị màn hình chạy test case, quản lý thời gian chạy test case, xuất báo cáo sau khi hoàn thành quá trình test.

b. User Case Diagram:



Hình 16: User Case Diagram

3.3.3 Các chức năng của phần mềm

a. Test case kịch bản kiểm thử:

- Test case là test - kiểm tra các case - tình huống có thể xảy ra giúp Tester xác định một ứng dụng, hệ thống phần mềm hay một chứng năng ứng dụng có hoạt động đúng hay không. test case mô tả dữ liệu đầu vào (input), hành động (action) hoặc sự kiện (event) và một kết quả mong đợi (expected response).

Đây là test case trước khi được chuyển sang code autoTest :

ID	Description	Step	Result	Data
1	Verify in case click on button 'Notification'	1. Obserwe title of 'Main' Page	1. Title is "ATALINK - Quản trị chuỗi cung ứng"	
		2. Click on button 'Notification'	2. Redirect to 'Sign in' page	
		3. Obserwe title of 'Sign in' Page	3. Title is "ATALINK - Đăng nhập"	
2	Verify behavior in case sign in successfully	1. Obserwe title of 'Sign in' Page	1. Title is "ATALINK - Đăng nhập"	
		2. Type value in text box "UserName"	2. Show username in text box	quynhtvn11
		3. Type value in text box "UserNme"	3. show password in text box	123QWEasd
		4. Click on button "Sign in"	4. Sign in successfully, redirect to "My task" screen	

Hình 17: Cấu trúc Test Case

Test đã chuyển sang code autoTest:

- Verify in case click on button 'Notification'

Description

```

@Test(description = "Verify in case click on button 'Notification'")
public void C07_Verify_ClickOn_BtnNoti() throws Exception{
    Step 1: SetUp_MainMKP();
    Step 2: driver.findElement(By.xpath("//span[contains(text(),'Thông báo')]")).click();
            Thread.sleep( millis: 2000);
            String title = driver.getTitle();
    Step3: Assert.assertEquals(title, expected: "ATALINK - Đăng nhập");
}
  
```

Hình 18: Test case kiểm tra notification đã chuyển thành code

- Verify behavior in case sign in successfully

Description

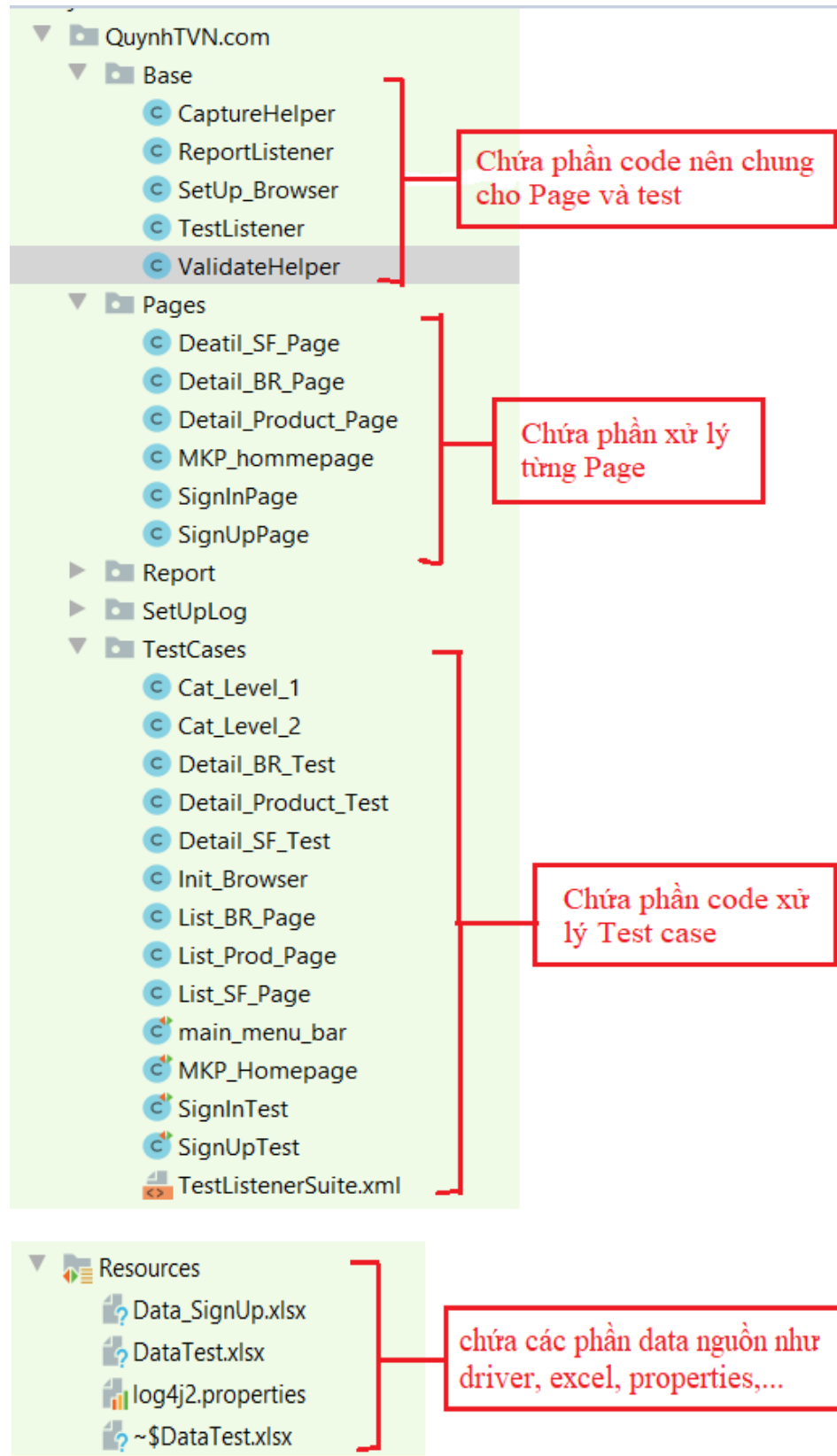
```

@Test(description = " Verify behavior in case sign in successfully")
public void C01_SignInLagre() throws Exception {
    driver.get("https://qc.atalink.com.vn/sign-in");
    Step 1 driver.findElement(By.linkText("Đăng nhập")).click();
            //Read data from file excel
    Step 2, 3 signInPage.SignIn(excel.getCellData("UserName", i), excel.getCellData("PassWord", i));

            Thread.sleep( millis: 2000);
            //Write data to file excel
    Step 4 if (driver.getTitle().equals("ATALINK - Đăng nhập")) {
            excel.setCellData( text: "False", i, colnum: 2);
        } else {
            excel.setCellData( text: "Pass", i, colnum: 2);
        }
}
  
```

Hình 19: Test case kiểm tra đăng nhập chuyển sang code

b. Cấu trúc code theo Page Object Model (POM)



Hình 20: Cấu trúc POM trong chương trình

c. Hàm dùng lại cho test case

Trong một màn hình ta sẽ test cho những locator cố định và có thể lặp lại trong nhiều test case, thay vì cứ phải viết lại các hàm cho cho những hành động lặp đi lặp lại với locator đó thì ta sẽ viết hàm để dùng lại.

Những hành động thường xuyên được sử dụng:

- Click: Ấn vào (Nút, hình ảnh, thẻ, nội dung, tiêu đề,...)
- Hover: rê chuột lên (Nút, hình ảnh, thẻ, nội dung, tiêu đề,...)
- Type: Nhập dữ liệu và textbox.
- Submit: Gửi dữ liệu.
- Wait for page load: Đợi tải xong trang web.
- IsDisplayed: Kiểm tra phần tử có được hiển thị.
- isEnabled: Kiểm tra phần tử đó có đang được bật.
- getText: Lấy ra chuỗi ký tự được hiển thị.

Ví dụ minh họa về 2 hàm dùng lại:

Dưới đây là 2 hàm đã được viết sẵn, việc còn lại là gọi và truyền locator vào.

```
public void SetText(By element, String Text){
    //Sendkey cho giá trị được truyền vào
    wait.until(ExpectedConditions.elementToBeClickable(element));
    driver.findElement(element).clear();
    driver.findElement(element).sendKeys(Text);
}

public void ClickElement(By element){
    wait.until(ExpectedConditions.elementToBeClickable(element));
    driver.findElement(element).click();
}

public class SignInPage {

    private WebDriver driver;
    private ValidateHelper validateHelper;

    private By E_UserName = By.id("SignIn_username");
    private By E_PassWord = By.id("SignIn_password");
    private By btn_Submit = By.xpath("//*[@id='SignIn']/div[6]/button");

    public void SignIn(String UserName, String Password){
        validateHelper.WaitForPageLoading();
        validateHelper.SetText(E_UserName, UserName);
        validateHelper.SetText(E_PassWord, Password);
        validateHelper.ClickElement(btn_Submit);
    }
}
```

Khởi tạo biến toàn phần

Test case được tinh gọn

Hình 21: Hàm dùng lại trong chương trình

Trong màn hình đăng nhập, khởi tạo biến toàn cục cho tên tài khoản, mật khẩu và nút gửi. Sau đó viết test case cho trường hợp đăng nhập.

Có thể thấy test case đã được tinh gọn đi rất nhiều nhờ chỉ cần gọi tên của hành động mà không phải làm các bước khởi tạo kiểu và locator. Các hàm thực hiện hành đó còn có thể được dùng lại nhiều lần trong nhiều test case, tiết kiệm nguồn lực và thời gian viết test case.

d. Import, export dữ liệu từ file dữ liệu (file Excel)

Tùy vào từng chức năng đòi hỏi kiểu dữ liệu như thế nào cần nhập vào bao nhiêu tiêu chí, số dữ liệu cần dùng, mà ta có thể thiết kế ra file dữ liệu phù hợp với chức năng đó.

Ví dụ: Dữ liệu test cho chức năng đăng nhập (nhiều người dùng cùng đăng nhập trong 1 khoảng thời gian với các thông tin có thể đúng hoặc sai)

Input: Dữ liệu trong file Excel (UserName, PassWord)

Output: Trả về kết quả UserName, PassWord (Pass/Fail) vào file Excel và kết quả Pass/Fail của test case xuất report.

File dữ liệu:

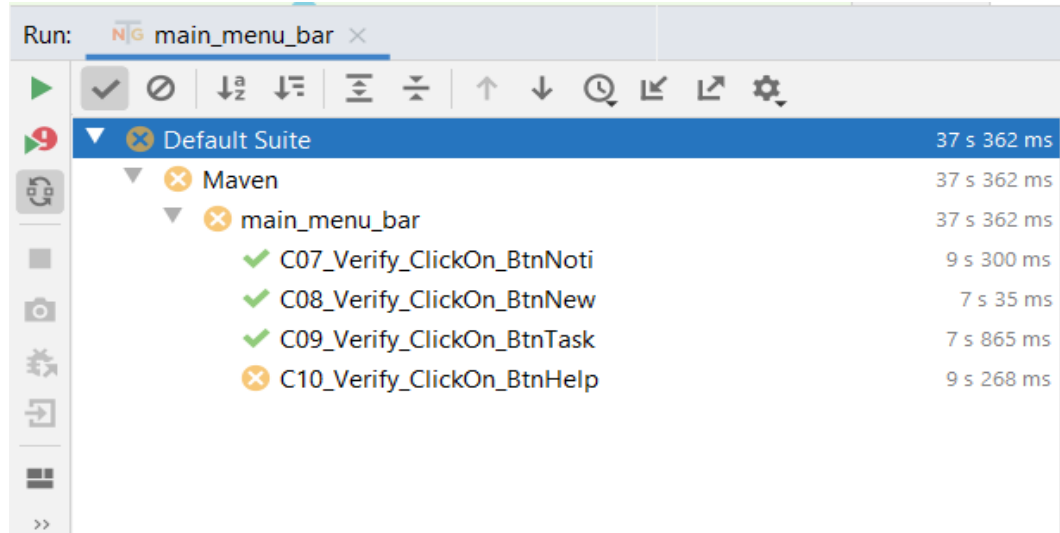
UserName	PassWord	Result
quynhtvn11+1@gmail.com	Thaytoo02	False
quynhtvn11+2@gmail.com	Thaytoo02	Pass
quynhtvn11+3@gmail.com	Thaytoo02	
quynhtvn11+4@gmail.com	Thaytoo02	
quynhtvn11+5@gmail.com	Thaytoo02	
quynhtvn11+6@gmail.com	Thaytoo01	
quynhtvn11+7@gmail.com	Thaytoo03	
quynhtvn11+8@gmail.com	Thaytoo02	
quynhtvn11+9@gmail.com	Thaytoo02	
nhungbth2912+1@gmail.com	123QWEasd	
nhungbth2912+2@gmail.com	123QWEas5	
nhungbth2912+3@gmail.com	123QWEasd	
nhungbth2912+4@gmail.com	123QWEasd	
nhungbth2912+5@gmail.com	123QWEasd	

Hình 22: Cấu trúc của file dữ liệu test

e. Quản lý test case

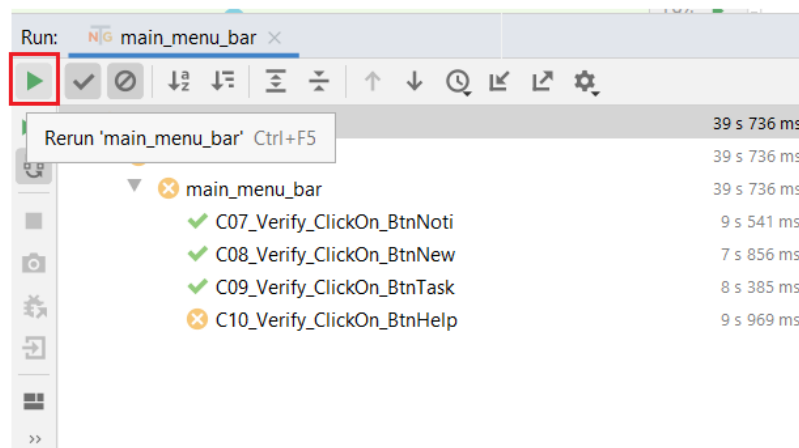
Test case được quản lý nhờ sự hỗ trợ của thư viện TestNG, sẽ có các chức năng:

Giao diện: Đang hiển thị test case pass(v) và fail(x)



Hình 23: Giao diện quản lý test case

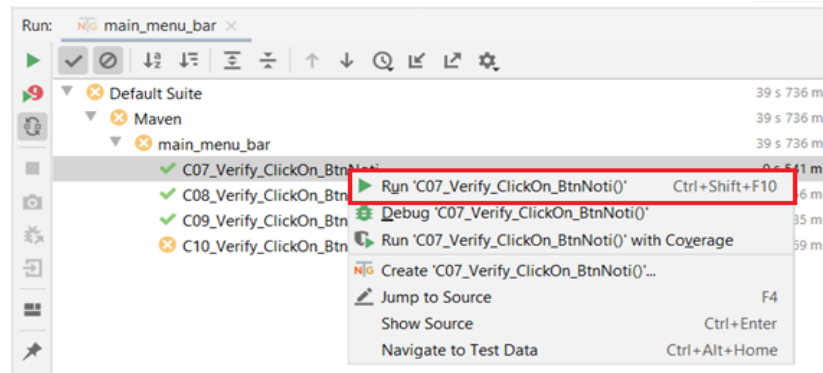
- Chạy 1 hoặc nhiều Test case trong 1 lần test
- Chạy toàn bộ test case



Hình 24: Tính năng chạy toàn bộ test case

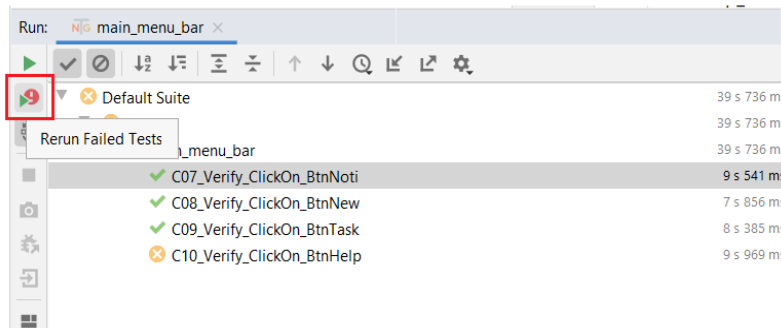
main_menu_bar là hàm tổng chứa test case kiểm tra thanh menu chính, chạy hàm này có nghĩa là chạy tất cả các test case có trong nó.

Chạy 1 test case nhất định: Nhấn chuột phải vào test case đó sẽ hiển thị danh sách chức năng có thể dùng cho test case đó



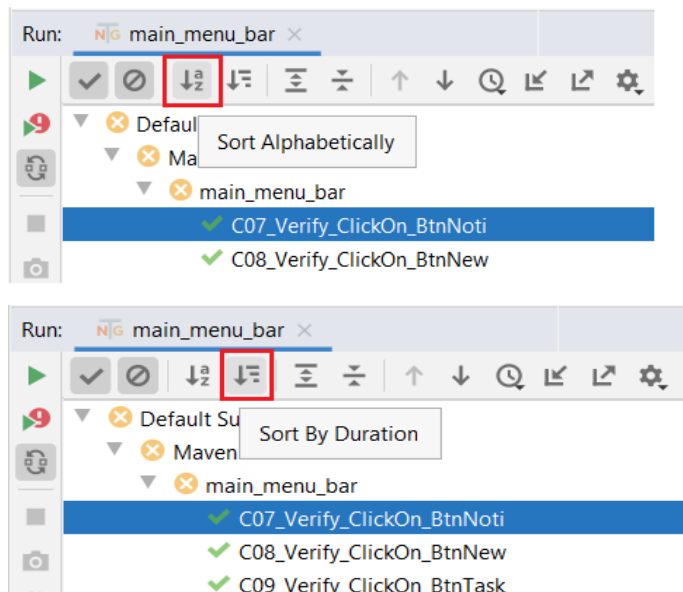
Hình 25: Tính năng chạy một số test case được chọn

- Chỉ chạy lại test case fail



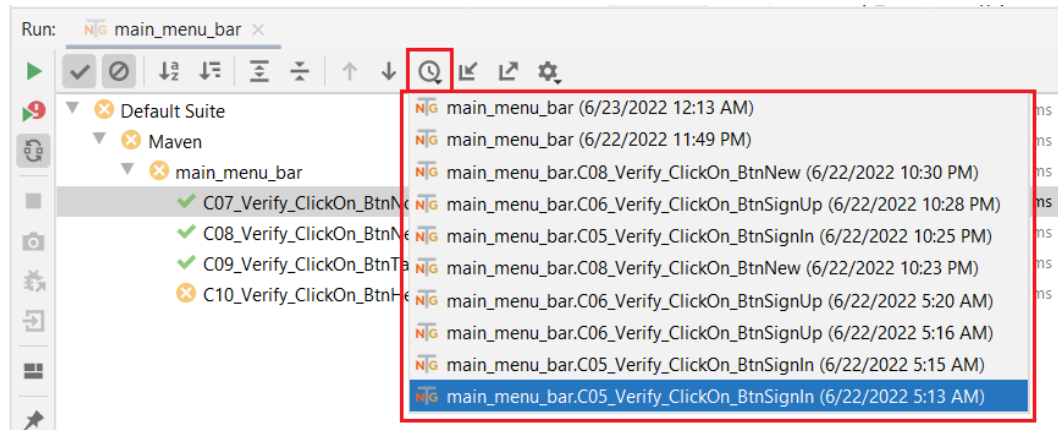
Hình 26: Tính năng chỉ chạy test case fail

- Sắp xếp test case từ A->Z, thời gian chạy



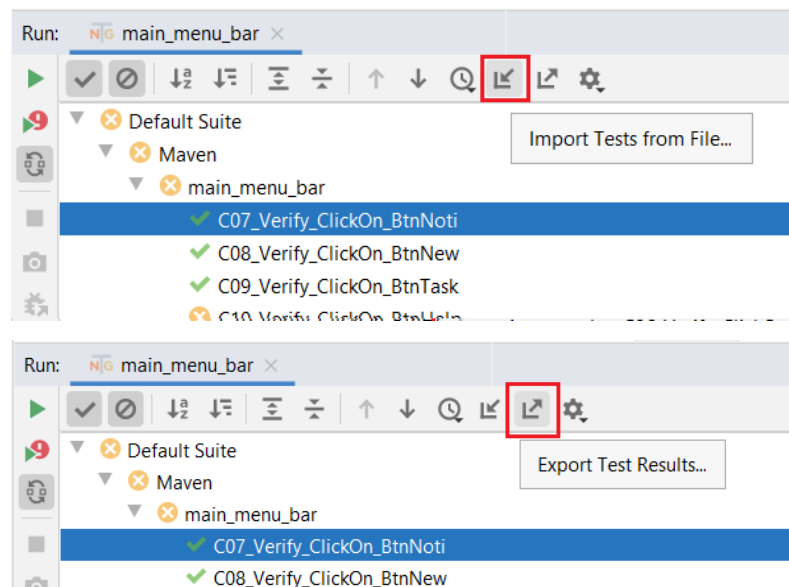
Hình 27: Tính năng sắp xếp test case

- Hiện thị lịch sử test



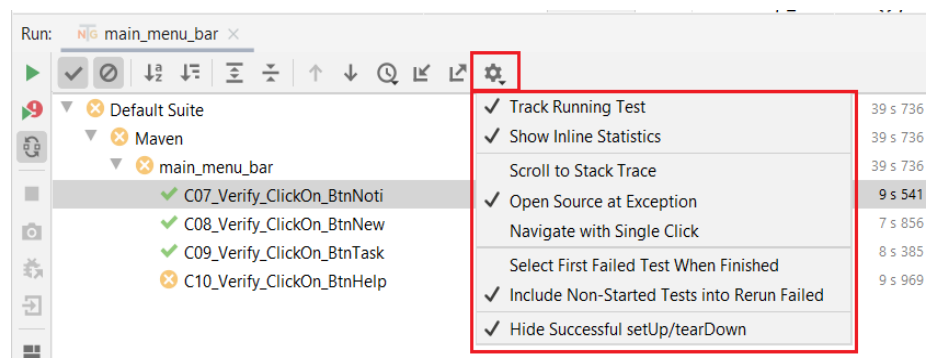
Hình 28: Tính năng hiện thị lịch sử chạy test case

- Import test từ file, export kết quả



Hình 29: Tính năng Import, Export kết quả test

- Một số cài đặt

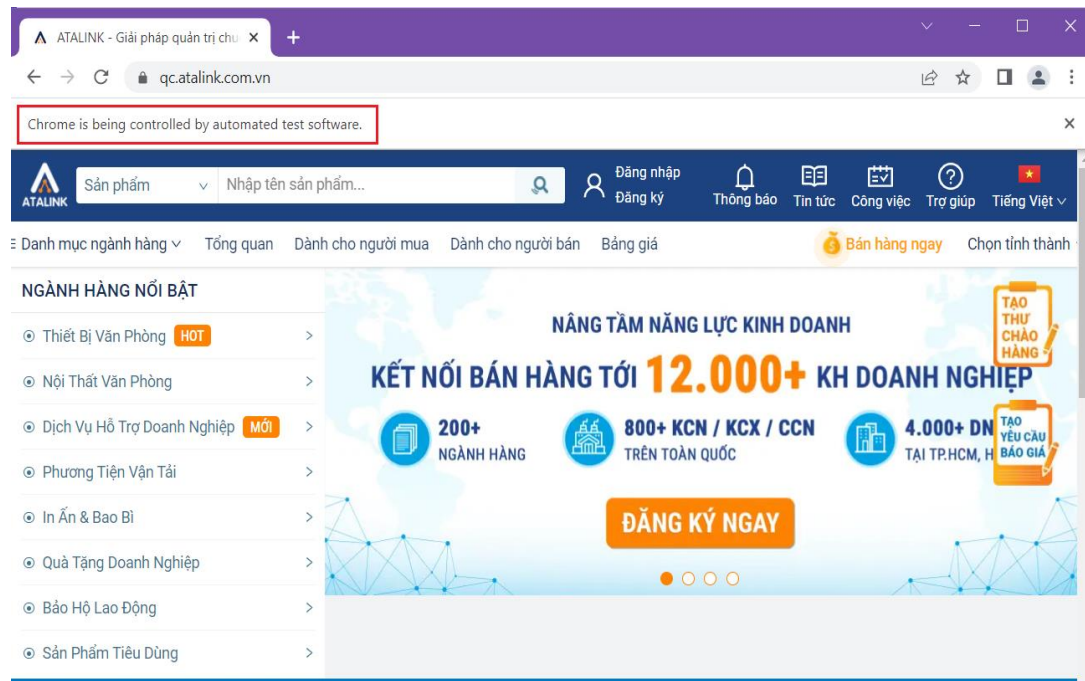


Hình 30: Cài đặt trong quản lý test case

f. Multi Browser

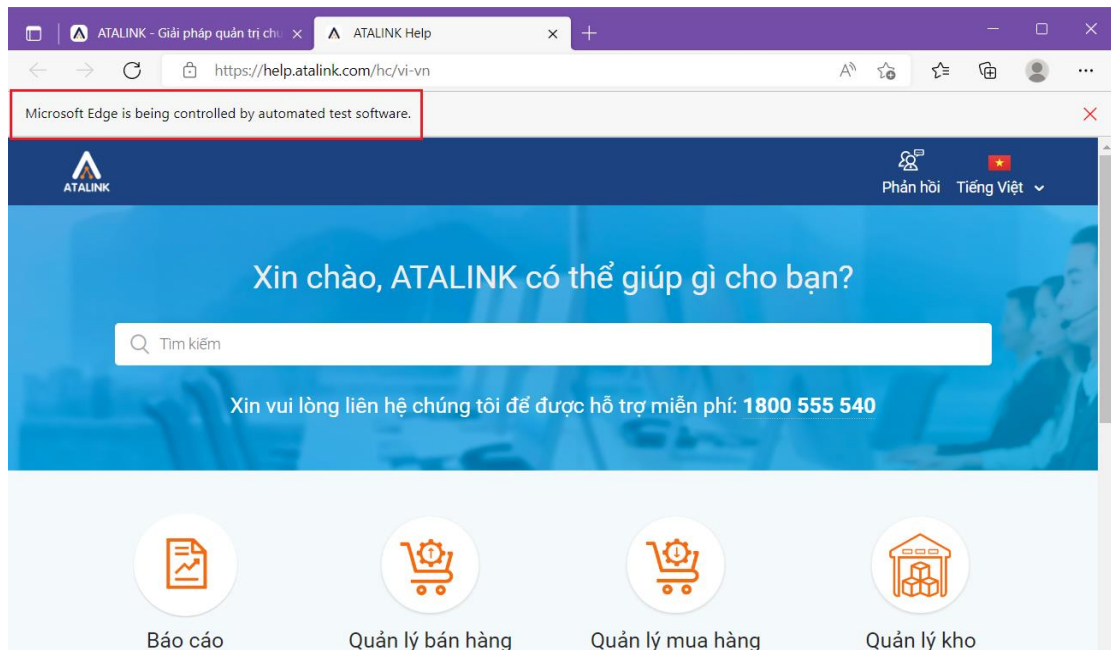
Phần mềm có thể test trên 3 Browser là: Google Chrome, Ms Edge

- Google Chrome



Hình 31: Test trên Google Chrome

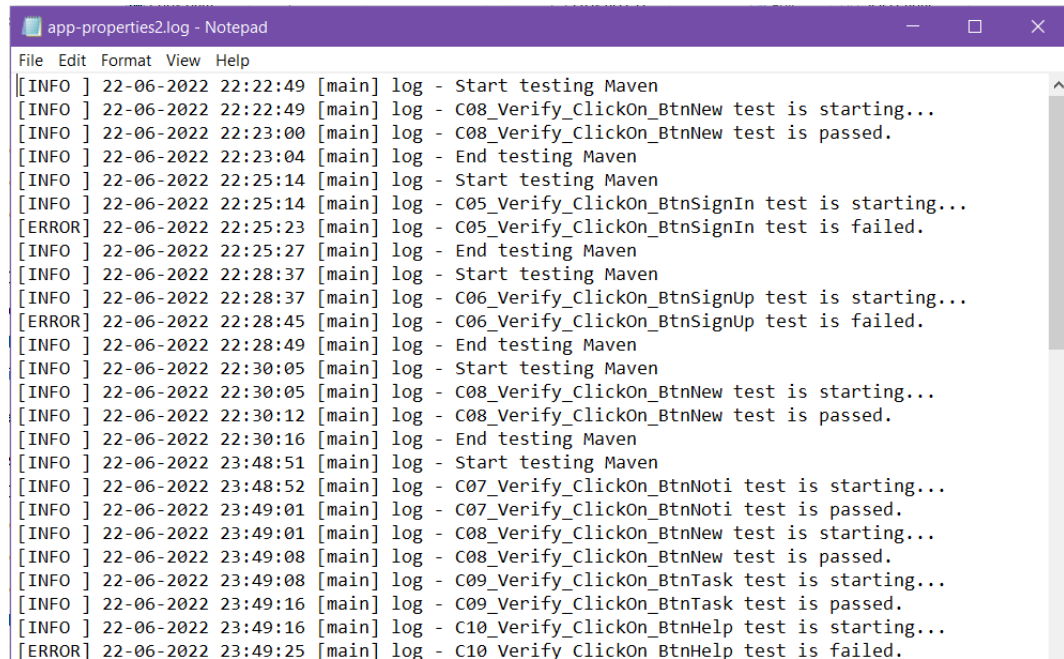
- Microsoft Edge



Hình 32: Test trên Microsoft Edge

g. Quản lý thời gian test, tạo file log và log console

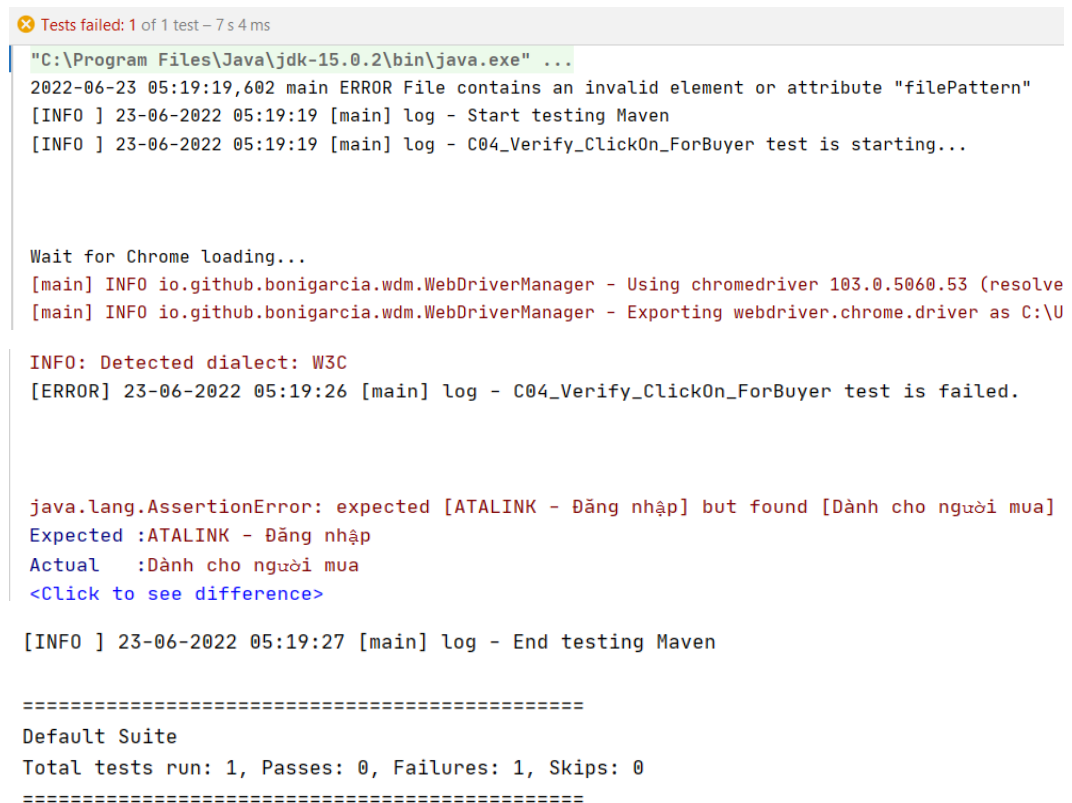
File log:



```
File Edit Format View Help
[INFO ] 22-06-2022 22:22:49 [main] log - Start testing Maven
[INFO ] 22-06-2022 22:22:49 [main] log - C08_Verify_ClickOn_BtnNew test is starting...
[INFO ] 22-06-2022 22:23:00 [main] log - C08_Verify_ClickOn_BtnNew test is passed.
[INFO ] 22-06-2022 22:23:04 [main] log - End testing Maven
[INFO ] 22-06-2022 22:25:14 [main] log - Start testing Maven
[INFO ] 22-06-2022 22:25:14 [main] log - C05_Verify_ClickOn_BtnSignIn test is starting...
[ERROR] 22-06-2022 22:25:23 [main] log - C05_Verify_ClickOn_BtnSignIn test is failed.
[INFO ] 22-06-2022 22:25:27 [main] log - End testing Maven
[INFO ] 22-06-2022 22:28:37 [main] log - Start testing Maven
[INFO ] 22-06-2022 22:28:37 [main] log - C06_Verify_ClickOn_BtnSignUp test is starting...
[ERROR] 22-06-2022 22:28:45 [main] log - C06_Verify_ClickOn_BtnSignUp test is failed.
[INFO ] 22-06-2022 22:28:49 [main] log - End testing Maven
[INFO ] 22-06-2022 22:30:05 [main] log - Start testing Maven
[INFO ] 22-06-2022 22:30:05 [main] log - C08_Verify_ClickOn_BtnNew test is starting...
[INFO ] 22-06-2022 22:30:12 [main] log - C08_Verify_ClickOn_BtnNew test is passed.
[INFO ] 22-06-2022 22:30:16 [main] log - End testing Maven
[INFO ] 22-06-2022 23:48:51 [main] log - Start testing Maven
[INFO ] 22-06-2022 23:48:52 [main] log - C07_Verify_ClickOn_BtnNoti test is starting...
[INFO ] 22-06-2022 23:49:01 [main] log - C07_Verify_ClickOn_BtnNoti test is passed.
[INFO ] 22-06-2022 23:49:01 [main] log - C08_Verify_ClickOn_BtnNew test is starting...
[INFO ] 22-06-2022 23:49:08 [main] log - C08_Verify_ClickOn_BtnNew test is passed.
[INFO ] 22-06-2022 23:49:08 [main] log - C09_Verify_ClickOn_BtnTask test is starting...
[INFO ] 22-06-2022 23:49:16 [main] log - C09_Verify_ClickOn_BtnTask test is passed.
[INFO ] 22-06-2022 23:49:16 [main] log - C10_Verify_ClickOn_BtnHelp test is starting...
[ERROR] 22-06-2022 23:49:25 [main] log - C10_Verify_ClickOn_BtnHelp test is failed.
```

Hình 33: Cấu trúc file Log

Log console:



```
Tests failed: 1 of 1 test - 7 s 4 ms
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" ...
2022-06-23 05:19:19,602 main ERROR File contains an invalid element or attribute "filePattern"
[INFO ] 23-06-2022 05:19:19 [main] log - Start testing Maven
[INFO ] 23-06-2022 05:19:19 [main] log - C04_Verify_ClickOn_ForBuyer test is starting...

Wait for Chrome loading...
[main] INFO io.github.bonigarcia.wdm.WebDriverManager - Using chromedriver 103.0.5060.53 (resolve
[main] INFO io.github.bonigarcia.wdm.WebDriverManager - Exporting webdriver.chrome.driver as C:\U

INFO: Detected dialect: W3C
[ERROR] 23-06-2022 05:19:26 [main] log - C04_Verify_ClickOn_ForBuyer test is failed.

java.lang.AssertionError: expected [ATALINK - Đăng nhập] but found [Dành cho người mua]
Expected :ATALINK - Đăng nhập
Actual   :Dành cho người mua
<Click to see difference>

[INFO ] 23-06-2022 05:19:27 [main] log - End testing Maven

=====
Default Suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```

Hình 34: Cấu trúc Log console

File log hiển thị:

- Thông tin về thời gian thực hiện của test case.
- Các hành động như khởi tạo mở trình duyệt, tên trường hợp thử nghiệm, các thông tin khác, đóng trình duyệt.
- Khi toàn bộ test case được thực thi, nó sẽ tóm tắt tổng số các trường hợp thử nghiệm bao gồm số lượng test case đã pass và số lượng test case fail.

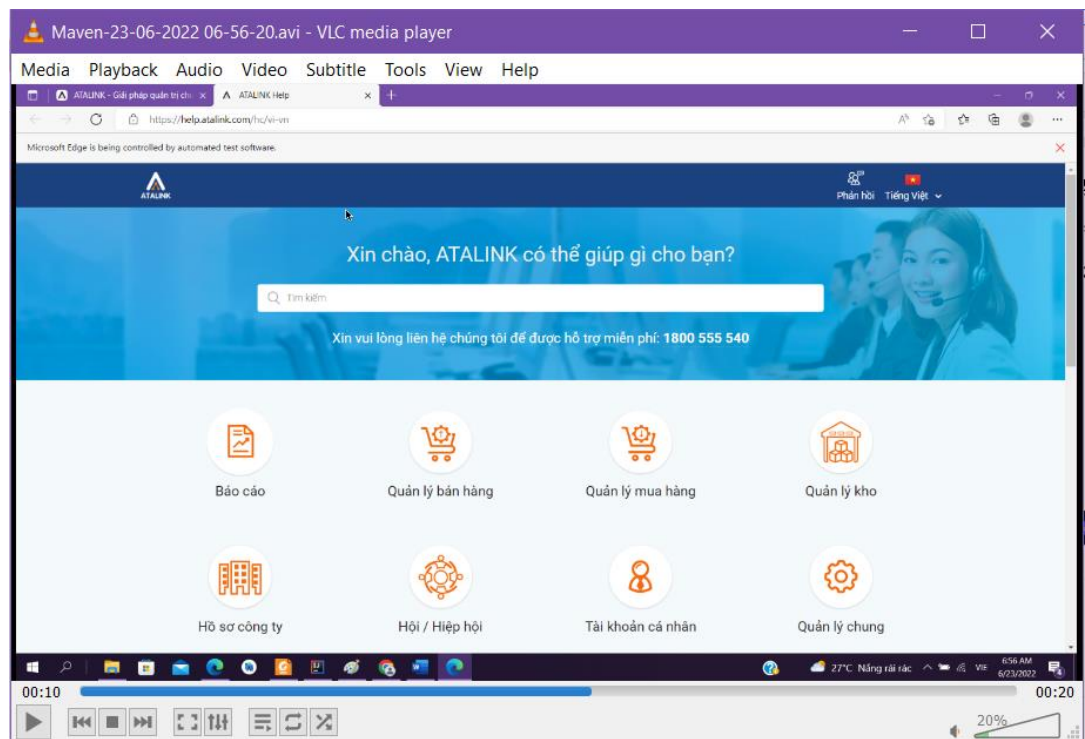
h. Ghi hình quá trình chạy của test case bị fail (chưa đẩy được lên Report)

Đối với những test case bị Fail thì sẽ được ghi hình lại, hiện tại thì đang được lưu vào thư mục Screenshot >> Video.



Hình 35: Lưu trữ video ghi được của test case fail

Khi nhấn vào tên của file, video sẽ được mở bằng ứng dụng VLC media player vì đang xuất video có đuôi là .avi



Hình 36: Mở video bằng VLC media player

i. Đẩy Report lên web

Giao diện của web Report kết quả chạy của bộ test case:

Sẽ có 2 tab trong web Report:

View List:

Tests		C02_Verify_Dropdown_searchBar	
C01_Verify_Default_Box_SearchFor	10:35:12 PM / 00:00:16:971	Fail	
C02_Verify_Dropdown_searchBar	10:35:29 PM / 00:00:08:598	Fail	
C03_Verify_TextinBox_clickOn_aOption	10:35:37 PM / 00:00:11:346	Pass	
C04_Verify_placeHolder_of_FieldSearch	10:35:49 PM / 00:00:13:249	Pass	
C05_Verify_ClickOn_BtnSignIn	10:36:02 PM / 00:00:14:381	Fail	
C06_Verify_ClickOn_BtnSignUp	10:36:16 PM / 00:00:14:294	Fail	
C07_Verify_ClickOn_BtnNoti	10:36:30 PM / 00:00:16:374	Pass	
C08_Verify_ClickOn_BtnNew	10:36:47 PM / 00:00:16:201	Pass	
C09_Verify_ClickOn_BtnTask	10:37:03 PM / 00:00:17:506	Pass	
C10_Verify_ClickOn_BtnHelp	10:37:21 PM / 00:00:24:682	Fail	

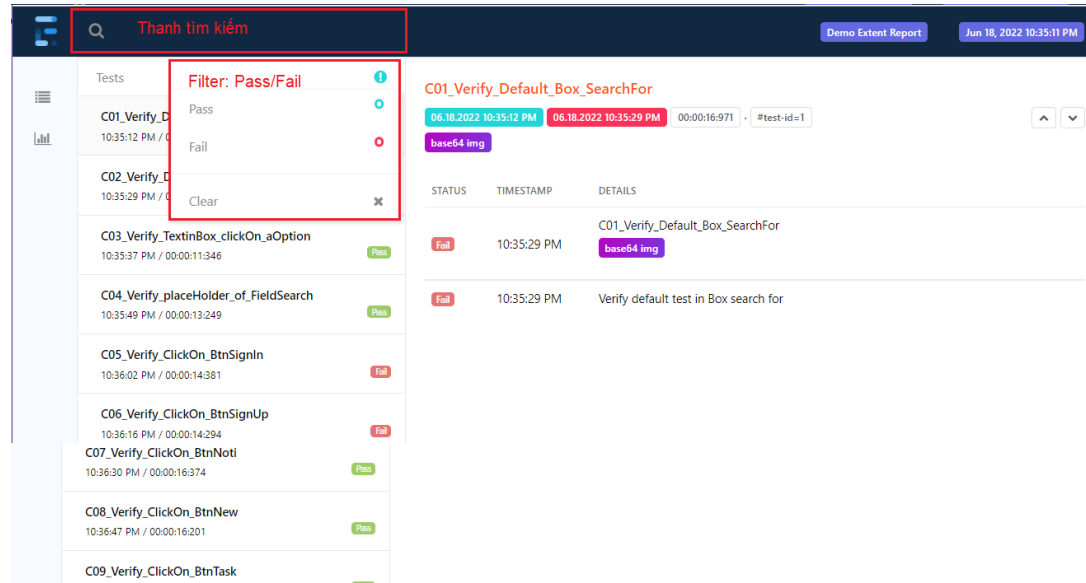
C02_Verify_Dropdown_searchBar		
06.18.2022 10:35:29 PM	06.18.2022 10:35:37 PM	00:00:08:598
#test-id=2		
base64 img		
STATUS	TIMESTAMP	DETAILS
Fail	10:35:37 PM	C02_Verify_Dropdown_searchBar
		base64 img
Fail	10:35:37 PM	Verify select in drop down list of search bar

Hình 37: Giao diện test report

Hiển thị thời gian chạy của bộ test case.

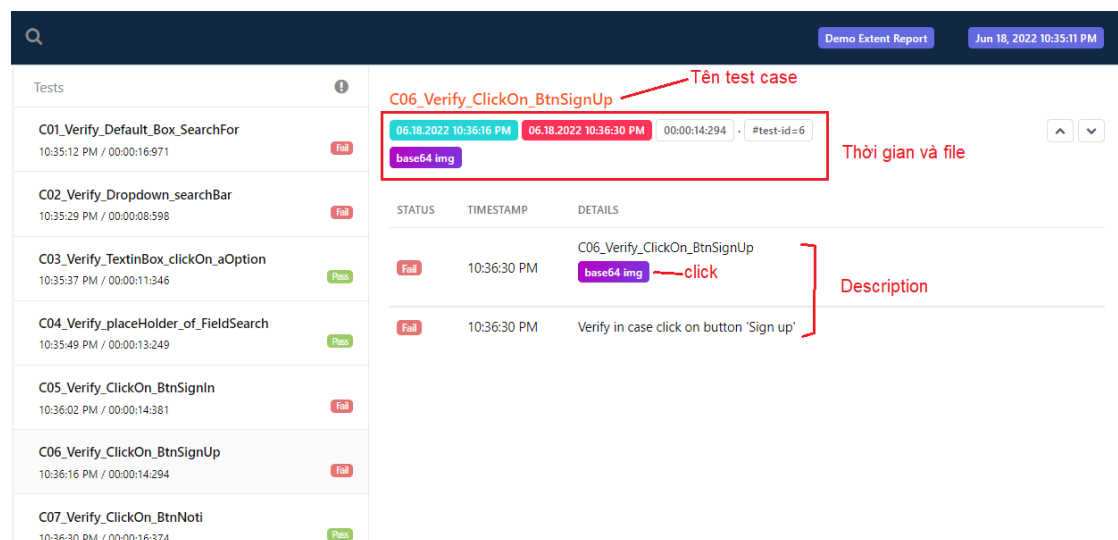
Hiển thị danh sách test case đã được chạy trong 1 lần.

Có các chức năng: Tìm kiếm test case bằng cách nhập tên của test case và lọc test case theo 2 tiêu chí Pass hoặc Fail.



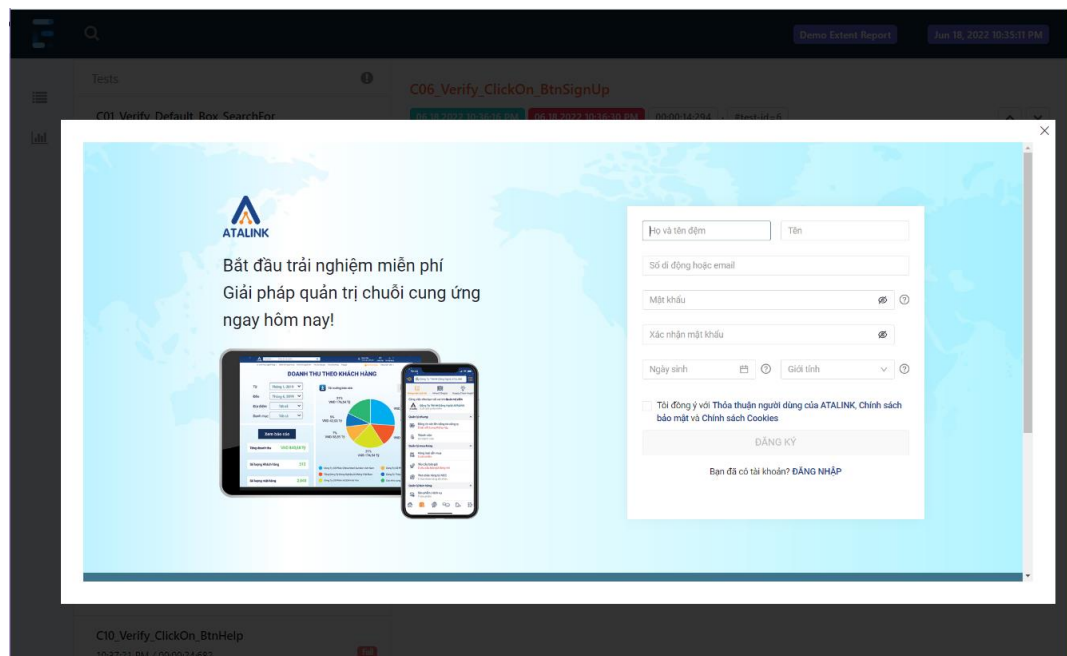
Hình 38: Lọc test case Pass và Fail

Nhấn chọn 1 test case sẽ hiển thị chi tiết của test case đó bên góc phải bao gồm các thông tin:



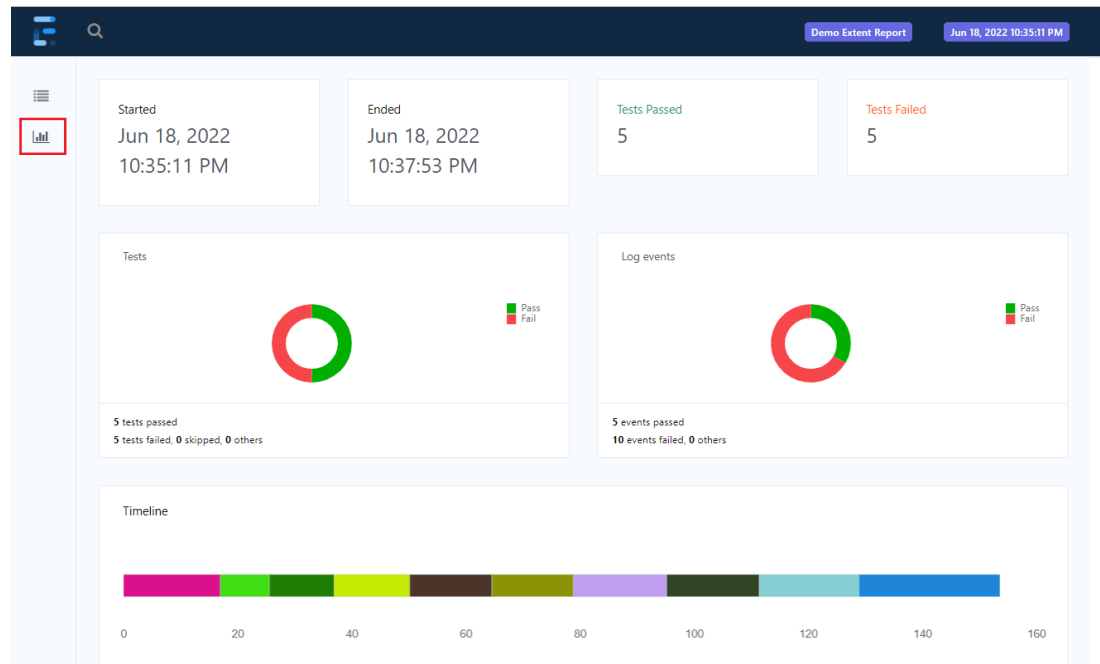
Hình 39: Chi tiết test case

- Tên của case được chọn
- Thời gian bắt đầu chạy
- Thời gian kết thúc chạy
- Test case chạy trong bao lâu
- ID tự động của test case
- Hình ảnh báo lỗi nếu test case bị fail
- Bảng thông tin chạy của test case gồm các cột: Trạng thái, thời gian và thông tin chi tiết.
- Đối với những test case bị fail sẽ được chụp ảnh lại tại màn hình bị fail và có thể xem trong web Report ở phần thông tin chi tiết của test case đó:



Hình 40: Ảnh chụp màn hình

View Chart:



Hình 41: Giao diện test report ở tab biểu đồ

Hiển thị thời gian test.

Số lượng test case Pass và Fail.

Hiển thị biểu đồ thể hiện tỉ lệ giữa test case Pass và Fail.

Hiển thị timeline cho từng test case, mỗi test case là 1 màu khác nhau.

CHƯƠNG 4. KẾT LUẬN

4.1 Kết quả đạt được

- Hiểu và nắm rõ được cách hoạt động của 1 quy trình kiểm thử tự động.
- Biết cách sử dụng các cấu trúc, framework, hàm,... trong thư viện hỗ trợ kiểm thử tự động Selenium.
- Xây dựng phần mềm kiểm thử tự động cho web Atalink
 - Xây dựng được bộ test case và dữ liệu test cho web Atalink
 - Hiểu và sử dụng được Design Pattern: Page Object Model (POM)
 - Tạo hàm dùng lại cho test case
 - Import dữ liệu từ file Excel
 - Kiểm thử trên nhiều trình duyệt: Google Chrome, Microsoft Edge
 - Hiện thị test case pass/fail, những test case fail sẽ được screenshot và record video
 - Tạo file log quản lý thời gian kiểm thử
 - Quản lý được web test report
- Ưu điểm:
 - Thực hiện kiểm thử dễ dàng, tiết kiệm thời gian và công sức.
 - Nhiều từ khóa có thể giúp thực hiện các tác vụ tự động hóa của người dùng.
 - Nhiều trường hợp thử nghiệm ngẫu nhiên hơn.
 - Tập báo cáo hiển thị thông tin rõ ràng, người dùng có thể lọc kết quả đạt, kết quả không đạt và hiển thị đạt / không đạt tỷ lệ phần trăm.
 - Người dùng có thể thấy trường hợp thử nghiệm thất bại bằng hình ảnh trong tập kết quả báo cáo.
 - Hỗ trợ tốt cho regression test và smoke test.
- Nhược điểm:
 - Người dùng phải có kiến thức lập trình cơ bản và có kiến thức chuyên môn về hệ thống.

- Còn nhiều trường hợp mà kiểm thử tự động chưa hỗ trợ được như:
Tooltip, links, images, tables.
- Khó bảo trì và mở rộng.

4.2 Tính năng phát triển trong tương lai

- Lập lịch kiểm thử: là các bộ test case sẽ được cài đặt tự động chạy ngầm sau một khoảng thời gian, để kiểm tra mức độ ổn định của web được kiểm thử.
- Gửi mail report sau khi kiểm thử theo lịch để nắm được tình hình kiểm thử.
- Đẩy được video vào file report để dễ dàng biết cụ thể lỗi đang ở phần nào.

TÀI LIỆU THAM KHẢO

- [1]. Trang chủ ATALINK: <https://vietnam.atalink.com/>
- [2]. TestNG Tutorial: [TestNG Tutorial - Javatpoint](#)
- [3]. Selenium Tutorial for Beginners: WebDriver & Testing
[Selenium Tutorial for Beginners: Learn WebDriver & Testing \(guru99.com\)](#)
- [4]. Trang chủ Selenium Tutorial với java
[Selenium Tutorial - javatpoint](#)
- [5]. Tìm hiểu quy trình kiểm thử phần mềm
[Quy trình kiểm thử phần mềm \(viblo.asia\)](#)
- [6]. Tìm hiểu các giai đoạn trong kiểm thử phần mềm
[Các giai đoạn trong quy trình kiểm thử phần mềm | Anh Tester](#)
- [7]. Các thư viện hỗ trợ Selenium trong Maven project:
[Maven Repository: Search/Browse/Explore \(mvnrepository.com\)](#)
- [8]. Sử dụng POP trong Selenium
[Sử dụng Page Object Model trong selenium như thế nào \(viblo.asia\)](#)
- [9]. Mô hình hướng đối tượng Page Object Model trong Selenium
[Mô hình đối tượng POM và Page Factory trong Selenium WebDriver - Blog Tiền Điện Tử \(blogtiendientu.vn\)](#)
- [10]. Trang chủ Maven
[Maven – Maven in 5 Minutes \(apache.org\)](#)
- [11]. Hướng dẫn dùng và cài đặt Maven project
[Maven Tutorial \(tutorialspoint.com\)](#)
- [12]. Automation test tutorial:
[Automation Testing Tutorial: What is Automation Testing? \(guru99.com\)](#)
- [13]. How to use TestNG:
[Create TestNG Project In Eclipse & Run Selenium Test Script \(lambdatest.com\)](#)
- [14]. TestNG with IntelliJ IDEA
[TestNG | IntelliJ IDEA \(jetbrains.com\)](#)
- [15]. Selenium with IntelliJ IDEA
[Selenium | IntelliJ IDEA \(jetbrains.com\)](#)