**CSCE 5430-001 SOFTWARE ENGINEERING SUMMER 2022**

**GROUP 5 TESTING REPORT**

**Diagram

Description automatically generated**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Issue | Description | Author |
| June 30, 2022 | Project3 Report | Introduction | Mounika, Chandralekha |
| June 30, 2022 | Project3 Report | Testing Interface | Pavan |
| July 1, 2022 | Project3 Report | Testing Levels & Methods | Venkat |
| July 1, 2022 | Project3 Report | Testing Plugins | Priyanka |
| July 2, 2022 | Project3 Report | Testing Overall | Moulya |
| July 2, 2022 | Project3 Report | Testing Reports | Sai Dev Prakash |

**INDEX:**

1. Introduction

1.1. SeleniumHQ…………………………………………………………………………….…………...3

1.2. SeleniumWebdriver……………………………………………………………….…………….….3

1.3. SeleniumWebdriver Framework Architecture…………………………….………………………...3

1.4. Selenium Client Libraries/Language Binding……………………………….……………………...3

1.5. Selenium IDE………………………………………………………………………………………..4

1.6. Features of Selenium IDE………………………………………………….………………………..4

1.7. SeleniumHQ Architecture………………………………………………...…….………..………….4

1.8. SeleniumHQ Working……………………………………………………………..….……………..6

1.9. History……………………………………………………………………...…………….………….7

2. Testing Interface…………………………………………………….……………………………….….….8

3. Testing Levels & Methods

3.1. End-to-End Testing………………………………………………………...………..………………9

3.2. System Testing…………………………………………………………………………..…….…….9

4. Testing Plugins

4.1. Java……………………………………………………………………..…………………………..10

4.2. Web Driver……………………………………………………………………………………….....10

4.3. Chrome Driver………………………………………………………………………………………10

4.4. Adobe Flash……………………………………………………………………………………..….10

5. Testing Overall

5.1. Implementation………………………………………………………………...……………...…….11

5.2. Advantages of SeleniumHQ……………………………………………………….………………..11

5.3. Disadvantages of SeleniumHQ……………………………………..……………………………….11

6. Testing Reports

6.1. Signup Testcase…………………………………………………………………….……….……….12

6.2. Login Testcase………………………………………………………………………..……...………13

6.3. Place Order Testcase………………………………………………………………..………..……...14

6.4. Feedback Testcase……………………………………………………………………….………….15

6.5. Add Product Testcase……………………………………………………………...………………..16

6.6. Update Status Testcase…………………………………………………………….………………..17

**1. Introduction:**

**1.1. SeleniumHQ:**

Selenium HQ is the #4 solution in the top Functional Testing Tools category and the #3 solution in the top Regression Testing Tools category. Selenium HQ receives an average score of 8 out of 10 from Peer Spot users. Selenium HQ versus Eggplant Digital Automation Intelligence is the comparison that Selenium HQ is most frequently made. Seventy percent of individuals looking for information on Selenium HQ on Peer Spot are from the large enterprise market. Professionals from a computer software company are the leading industry looking into this solution, making about 27% of all views.

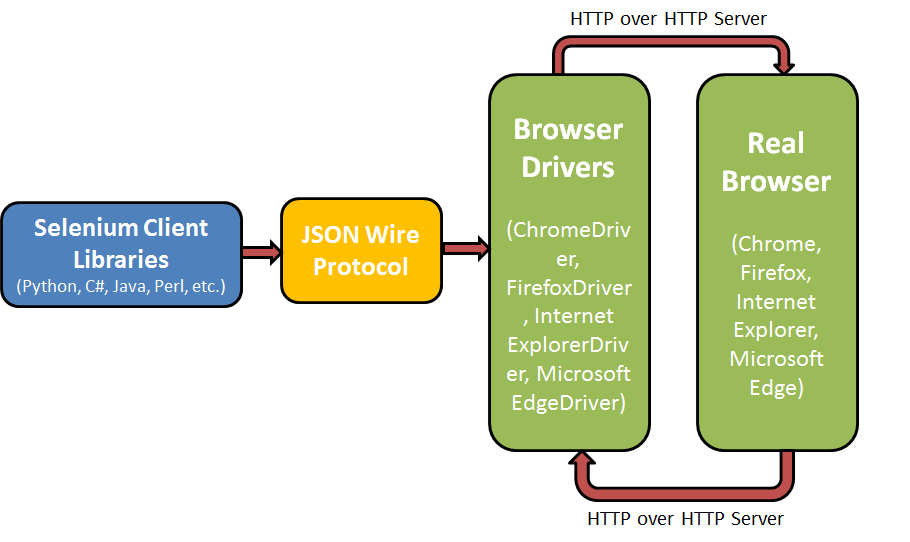
A collection of technologies called Selenium HQ can automate web browsers on several platforms. Selenium is compatible with a wide range of operating systems, browsers, and computer languages, and it may be managed by numerous testing frameworks. There are two varieties of selenium:

**1.2. Selenium WebDriver.**

A web framework called Selenium WebDriver enables you to run cross-browser testing. This program is used to automate the testing of web-based applications to ensure that they function as planned. You can select a programming language to use while creating test scripts using Selenium WebDriver. As was previously said, it improves upon Selenium RC to get around a few drawbacks. Although Selenium WebDriver cannot handle window components, this limitation can be addressed by utilizing programs like Sikuli, Auto IT, etc.

**1.3. Selenium WebDriver Framework Architecture:**

* WebDriver Architecture is made up of four major components:
* Selenium Client library
* JSON wire protocol over HTTP
* Browser Drivers



**1.4. Selenium Client Libraries/Language Bindings:**

Selenium offers help to various libraries like Ruby, Python, Java, and so on as language ties have been created by Selenium designers to give similarity to numerous dialects. For example, if you need to involve the program driver in Python, utilize the Python Bindings. You can download all your preferred upheld language ties from the authority webpage of Selenium

* **JSON Wire Protocol:**

JSON is an abbreviation for JavaScript Object Notation. An open standard gives a vehicle instrument to moving information among client and server on the web. It offers help for different information structures like clusters and articles which makes it simpler to peruse and compose information from JSON.JSON fills in as a REST (Representational State Transfer) API that trades data between HTTP servers. Get more familiar with REST API for getting to Selenium.

* **Browser Drivers:**

Selenium gives drivers intended for every program and without uncovering the inner rationale of program usefulness, the program driver communicates with the particular program by laying out a protected association. These program drivers are additionally intended for the language which is utilized for experiment mechanization like C#, Python, Java, and so on. You can download your preferred program driver according to your language prerequisites. For instance, you can arrange a Selenium Web driver for Python on BrowserStack.

* At the point when a test script is executed with the assistance of WebDriver, the accompanying undertakings are acted behind the scenes:
* A HTTP demand is produced and it is conveyed to the program driver for each Selenium Command
* The HTTP demand is gotten by the driver through an HTTP server
* Every one of the means/directions to be executed on the program is chosen by an HTTP server
* The HTTP server then, at that point, gets the execution status and thus sends it back to the robotization scripts
* As examined before, Selenium offers help for different Browsers like Chrome, Firefox, Safari, Internet Explorer and so on.

**1.5. Selenium IDE:**

Selenium IDE (Integrated Development Environment) is essentially a record/run instrument that an experiment designer utilizations to foster Selenium Test cases. Selenium IDE is a simple to-utilize instrument from the Selenium Test Suite and might in fact be utilized by other people to create computerized experiments for their web applications. One requires no extraordinary arrangement, to begin with, Selenium IDE. You simply have to add the expansion of your particular program. Selenium IDE furnishes you with a GUI (Graphical User Interface) for effectively recording your cooperation with the site.

Selenium IDE permits a client or an experiment engineer to make the experiments and test suites and alter them later according to their necessities. The improvement climate likewise gives the capacity of changing over experiments to various programming dialects, which makes it simpler for the client and doesn't command the requirement of knowing a particular programming language.

**1.6. Features of Selenium IDE**

There are several features provided in the IDE under the toolbar, using which one can control the execution of test cases:

* **Speed Control** – Helps control the speed of test cases
* **Run All** – Allows execution of the entire Test Suite
* **Run** – Runs the currently selected test
* **Pause/Resume** – Allows a user to pause and resume a particular test case
* **Step** – Helps step into each specific command in the test script
* **Rollup** – Helps group all the Selenium Commands together and make them execute as a single operation.

The features keep on getting eliminated or added depending on the usage of different versions of Selenium IDE extensions.

**1.7. Selenium HQ Architecture:**

Selenium WebDriver API enables interaction between browsers and browser drivers. This architecture consists of four layers namely the Selenium Client Libraries, JSON Wire Protocol, Browser Drivers, and Real Browsers.

* Selenium Client Library consists of languages like Python, Ruby, Java, Java Script, C# and so on. When the test cases are triggered, the entire Selenium code is converted to JSON format.
* JSON means JavaScript Object Notation. It performs the task of transferring information from the server to the client. JSON Wire Protocol is primarily responsible for transfer of data between HTTP servers. Generated JSON is made available to browser drivers through http Protocol.
* There is a unique browser driver for every browser. Browser drivers communicate with their individual browsers and conduct commands by deciphering JSON that the browser sends them. Any instructions received by the browser driver are immediately executed by the browser. The response is then returned as an HTTP response.

A picture containing diagram

Description automatically generated

Consider the following block of code −

WebDriver driver = new FirefoxDriver();

driver.get ([“https://www.iclothing.index.html“](https://www.tutorialspoint.com/index.htm));

The following are the steps:

* Once we run this block of code, the entire code will be converted with the help of JSON Wire Protocol over HTTP as a URL. The converted URL will be fed to the Firefox Driver.
* The HTTP server is used by the browser driver to obtain the HTTP request. As soon as the browser driver receives the URL, it sends the request over HTTP to the browser. It will cause the browser's Selenium commands to be executed.
* Now if the request is that of POST, it will trigger an action on the browser. If it’s a GET request, then the response will be produced at the browser end. Finally, it will be passed over HTTP to the browser driver. The browser driver will in turn send it to the User interface via JSON Wire Protocol. This is how Selenium WebDriver Architecture is explained.

Now let us see the Uses of Web Drivers in various testing scenarios.

* Automation Testing - By employing commands from the Selenium programming language, it automates web application testing. By managing mouse and keyboard events, it mimics actual actions. Using the native OS functionality, Web Driver communicates with the OS directly. It communicates using the browser driver.
* Performance Testing – Selenium testing is used to test the servers and website with the maximum loads that can handle. It is also used florentines Load testing and Stress testing.
* Regression testing – This testing is the most important and necessary testing phase. Regression test execution can be done with Selenium. Since selenium needs little work from humans. By running the entire test bed more quickly, it speeds up regression testing. There is less room for errors and defects as a result.
* Mobile testing – As the universe is revolving around mobiles, we should be able to test in mobiles also. Selenium also supports testing mobile-based applications. But we didn’t find many applications.

**1.8. SELENIUMHQ WORKING:**

* Selenium is a free and open-source tool that automates web browsers. It provides a unified interface for writing test scripts in a variety of programming languages, including Php, java, JavaScript, Ruby, Python, etc.
* The HUB acts as an interface between testing scripts and web browsers such as Chrome, Firefox, And Windows edge in windows and Safari browser in macOS etc.
* Selenium works perfectly on any device that has browsers such as Mobile devices, Tabs, and Laptops.
* The client-server architecture is supported by Selenium. A client system and a server system, which can connect to different computers or via a network, make up the client-server architecture model of software architecture.

Graphical user interface

Description automatically generated

* **Selenium WebDriver**:

The automation test created by the tester is triggered by Web Driver, which operates directly on the browser and employs functionality built into the browser. Another benefit of Web driver is that it can be used with HTML Unit browsers, which are headless browsers that are invisible to the user and lack a graphical user interface. This makes testing on HTML Unit browsers faster because it eliminates the need to wait for the page elements to load. Less time is required to execute test cases as a result. Web Driver is faster than Selenium RC since it manages the browser at the OS level.

* **Selenium Grid:**

We utilize Selenium Grid to simultaneously run several test cases on various remote workstations. This increases the execution's speed. Consider a test suite that includes both sophisticated and straightforward test cases. In this circumstance, we can separate the test cases based on their complexity and execute each one individually. The key benefit of Selenium Grid is the quicker and concurrent execution of test cases. It can perform a variety of test cases in a variety of situations on a variety of remote machines with the help of Selenium RC.

* **Selenium RC:**

A remote control called RC uses the remote from the browser to inject custom scripts that were built as automation code to be tested. Using Selenium RC Server, Selenium RC communicates with browsers. When a website loads, JavaScript is injected into browsers. Since the Web Driver does not offer an automated HTML file of test results, RC has an edge over it. For each of the supported languages, including Java, C#, Perl, Ruby, PHP, and Python, Selenium RC offers an API and library, making it the first automated web testing tool that enables users to develop in the language of their choice. It functions across platforms and browsers and is easily adaptable to new browser.

**1.9. History:**

**Jason Huggins** was the pioneer of the Selenium automation industry. As early as 2000 Mercury interactive was popular and a competitor to Thought works. Jason cracked a joke in an email sent to his team at Thought Work’s, where he mocked their competitor “Mercury” by specifying that selenium is the antidote for Mercury poisoning! His team took the name, that was how the team approved the name Selenium for their framework.

**Brief History of Selenium:**

**Diagram

Description automatically generated**

The Selenium is an assortment of various devices and has commitments from various prominent individuals. The long history of the selenium project has various stages with key people contributing enormously to the development at various stages. Selenium was first evolved by Jason Huggins in 2004 while he was functioning as an Engineer in ThoughtsWork on a web application that requires regular testing. He made the program utilizing JavaScript, subsequent to utilizing it he understood the inadequacies of manual testing and the need to check dullness. He initially named the program JavaScriptTestRunner yet after understanding the capability of the program, he made it an open-source program which he re-named as Selenium Core.

Graphical user interface, text

Description automatically generated

Anyway, there were issues. Because of the "Same Origin Policy" which precludes JavaScript from being utilized from an alternate Domain name from which it was sent off, analyzers needed to go through the pressure of introducing Selenium Core and Web servers containing web applications to be tried so they can have a place with a similar space. Paul Hammant another ThoughtWork Engineer offered an answer for this issue by making Selenium Remote control (Selenium RC) or Selenium 1.

Selenium Grid was created by Patrick Lightbody for equal testing purposes which addresses the need to lessening the time spent on test execution to negligible. Selenium mechanization test was quicker as different tests can be completed at the same time. Shinya Kasatani of Japan contributed by completing an undertaking on Selenium IDE in the year 2006. Selenium IDE mechanizes the program through the record and playback highlights. Simon Stewart made Web Driver Circa in 2006. This instrument helps the product analyzers to perform tests through automatic methodologies utilizing any programming language that is upheld in the year 2008, the entire Selenium robotization testing pioneer group chose to combine the web driver and Selenium RC to frame an extremely helpful device called Selenium 2. Throughout the long term, a lot of changes and upgrades have occurred the latest stable delivery was Selenium 3.14.0 in August 2018.

**2. TESTING INTERFACE:**

* Web driver in selenium is an interface and chrome browser, Firefox browser, Safari browser etc. are the classes that implement that interface.
* For each selenium command, Selenium Script generates an HTTP Request and sends it to the browser driver. The server is then contacted with an HTTP request by means of Browser Driver. The HTTP server is used to conduct the steps. The HTTP server receives the execution status, which the automation script then records.

Diagram

Description automatically generated

* WebDriver is a remote-control interface that enables introspection and control of user agents (browsers). The methods in this interface fall into three categories:
* Control of browser itself
* Selection of Web elements
* Debugging aids
* Selenium can run the same tests on several browsers and browser versions, making it ideal for UI testing and cross-browser testing.

**3. Testing Level and Methods:**

**3.1. End-to-End Testing:**

End-to-end testing (E2E testing) refers to a software testing method that involves testing an application's workflow from beginning to end. This method basically aims to replicate real user scenarios so that the system can be validated for integration and data integrity.

Essentially, the test goes through every operation the application can perform to test how the application communicates with hardware, network connectivity, external dependencies, databases, and other applications. Usually, E2E testing is executed after functional and system testing is complete.

Steps for End-to-End Testing are:

* The steps below are required to initiate and complete any E2E test.
* Analyze requirements. Have a clear idea of how the app is supposed to work in every aspect
* Set up a test environment in alignment with all the requirements
* Analyze software and hardware requirements
* List down how every system needs to respond
* List down testing methods required to test these responses. Include clear descriptions of standards (language, tools, etc.) to be followed in each test.
* Design test cases
* Run tests, study, and save results

**3.2 System Testing:**

System testing also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black-box testing, focuses on the functionality of an application.

System testing, for example, might check that every kind of user input produces the intended output across the application. With system testing, a QA team gauges if an application meets all of its requirements, which include technical, business, and functional requirements. To accomplish this, the QA team might utilize a variety of test types, including performance, usability, load testing, and functional tests.

**4. Testing Plugins:**

**4.1. Java:**

Selenium supports Java. So, testers can leverage the active community of contributors and detailed documentation to write test cases. Programs written in Java are faster than in other popular languages like Python.

**4.2. WebDriver:**

Webdriver drives a browser natively, as a user would, either locally or on a remote machine using the Selenium server, marks a leap forward in terms of browser automation. Selenium WebDriver refers to both the language bindings and the implementations of the individual browser controlling code.

This is commonly referred to as just WebDriver.WebDriver is designed as a simple and more concise programming interface.WebDriver is a compact object-oriented API. It drives the browser effectively.

**4.3. Chromedriver:**

ChromeDriver is a separate executable that Selenium WebDriver uses to control Chrome. It is maintained by the Chromium team with help from WebDriver contributors.

WebDriver finds the downloaded ChromeDriver execute any of these steps should do the trick: include the ChromeDriver location in your PATH environment variable and specify its location via the webdriver.chrome.driver system property the ChromeDriver class starts the ChromeDriver server process at creation and terminates it when quit is called.

This can waste a significant amount of time for large test suites where a ChromeDriver instance is created per test.

**4.4. Adobe Flash:**

To import the Flash library, you need to download the Flash Selenium jar file. You can download the Flash.jar file for Selenium WebDriver.There are some pre-defined methods available for testing youtube flash applications.

* playVideo
* pauseVideo
* mute
* setVolume
* seekTo

**5.Testing Overall:**

**5.1. Implementation:**

We used a wide range of technologies for implementing iClothing websites. To implement front-end design, we used HTML as a mark-up language and CSS to add styles to the website; we used PHP as a bridge between the front-end and back-end. MySQL is used for database establishment. Selenium-HQ tool is used in the testing phase.

**5.2. Advantages of SeleniumHQ:**

* It provides us with a ton of opportunities to code anything, there is no limitation on the kind of functionality you can do. We are using version3.2.1. we can use it on different platforms, like Windows, Linux, or macOS
* Selenium is the quickest apparatus contrasted with different contenders. It can run on any language, like Java, Python, C++, and. NET. Along these lines, we can test any application on Selenium, whether it's versatile or work area. It's simple for new individuals to get prepared for this arrangement. It has a more extensive local area. On the off chance that there are issues, you can glance around on the web and track down a decent arrangement. Selenium is in a nonstop turn of events. They discharge entirely stable forms. Those are the central issues that assisted us with picking Selenium over different tools.
* SeleniumHQ’s greatest benefit is that it is entirely adaptable. It saves time and enables us to execute our smoke test and regression tests quickly.
* The execution is great, the most significant highlights are Expected Conditions, activities, affirmations, checks, adaptable rates, and outsider combinations. The grids, as well as the selectors, are the most valuable features. The ease of use for the entire tool.
* This testing arrangement delivers the best web applications.

**5.3. Disadvantages of SeleniumHQ:**

* One downside to Selenium is that there is nothing like an article archive, like that found in QTP, particularly taking into account ceaseless joining rehearses that have become normal these days. On the off chance that I can contrast Selenium and another arrangement, for example, SoapUI Pro, there's a major distinction with regards to coding your own answers. With SoapUI Pro, if you need to get a little information, you don't have to compose any code or anything, and for somebody with no coding experience, this makes it a lot simpler to have the option to successfully utilize.
* The most recent adaptations are frequently unsteady. The parallel testing in Selenium is lacking. There's no UI in Selenium. Novices who need to begin utilizing it should know about Java, Python, or any programming language. Assuming you are utilizing a Chrome expansion, that is a BlazeMeter augmentation, you can record and make the content, nonetheless, that isn't exceptionally useful for fledglings. The level of stability depends on the version you are referencing. For example, Version 3.1.46 is very stable, however, Version 4.0.0, the latest version, is not.
* At the point when an article is changed or something is changed in the UI, then, at that point, we should refactor the code. Whenever an object is changed or something is changed in the UI, then we must refactor the code. This means that we must spend a lot of time on maintenance. For this, we were using another set of tools for quite some time before we switched to Selenium HQ, like Appium.
* Selenium utilizes a layer-based approach that is to some degree slower about executing code. Selenium utilizes a layer-based approach that is slower than Eggplant in executing code. The exhibition is needing improvement. There is no immediate choice for picture approval, and this would be a helpful expansion later. In our application, we have a graphical portrayal that should be approved, and it tends to be finished by approving the pictures. Nonetheless, we can't do that with Selenium, so we don't cover these situations in our experiments.

**6. Test Case Report:**

Below are the test cases we evaluated: -

* + Signup test case.
  + Login test case.
  + Place Order test case.
  + Feedback test case.
  + AddProduct test case.
  + UpdateStatus test case.

Below is the code folder for the above test cases: -



Since we are using SeleniumHQ for our testing we have downloaded the software by the below command.

**“*pip install selenium*”**

The above file contains the python file for each test case and since we are using a chrome browser, we downloaded the chrome latest version driver and kept it in the above folder.

**6.1 SignUp Test case:**

This test case focuses on signing up/registering a new user to iClothing website. We are testing with 3 cases one with new user details, another with already existing user details, and the last one with mismatched password details. The results should be registered successfully with the new user details, another showing existing customers, and third with password mismatch.

Below is the test case result screenshot

Graphical user interface, text, application, Word

Description automatically generated

**6.2 Login Test case:**

This test case focuses on logining up iClothing website using user credentials. We are testing with 2 cases by giving correct credentials and another by giving wrong credentials. If the credentials are correct, we should be able to log in and get logged out and if the credentials are wrong if should say user not found.

Below is the test case result screenshot.

Graphical user interface, text, application

Description automatically generated

**6.3 PlaceOrder Test case:**

This test case focuses on automating placing an order iClothing website. We are testing with 1 case by placing “Grey Balzar” by adding it to the cart and giving user details during checkout and selecting payment details and clicking placing an order.

Below is the test case result screenshot.

Graphical user interface, text, application

Description automatically generated

**6.4 Feedback Test case:**

This test case focuses on automating giving feedback on iClothing website. We are testing with 1 case by login into an existing user and clicking it to feedback and giving feedback and the user’s email and submitting it.

Below is the test case result screenshot.

Graphical user interface, text, application

Description automatically generated

**6.5. AddProduct Test case:**

This test case is done on the admin side by focusing on automating adding a product to iClothing website. We are testing with 1 case by adding a “Men Denkin Blue Jeans” product by login into the admin website clicking add product and giving relevant details and submitting it.

Below is the test case result screenshot.

Graphical user interface, text, application

Description automatically generated

**6.6. UpdateStatus Test case:**

This test case is done on the admin side of iClothing website. We are testing with 1 case by updating order of a customer from “Awaiting Confirmation” to “Confirmed”. This involves by log in to admin page and clicking an order and updating it status.

Below is the test case result screenshot.

Graphical user interface, text, application

Description automatically generated