Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Selenium Keyword Driven Automation Testing Framework

Sherry Singla*
Department of Computer Engg,
UCOE, Punjabi University,
Patiala, India

Harpreet Kaur
Assistant Professor, Department of Computer Engg,
UCOE, Punjabi University,
Patiala, India

Abstract-- Software testing is the main technique to ensure quality and finding bugs. It is in general a difficult and time-consuming task. Web applications are becoming more and more complex. Testing is often poorly performed or skipped by practitioners. Web testing may be even more difficult, due to the peculiarities of such applications. Test automation can help to avoid this situation. Automation Testing is the use of testing tools and reduce the need of manual or human involvement, repetitive or redundant tasks. The Objective of this thesis is to perform Automation Testing for web applications using Software Testing Tool "Selenium Webdriver". With this web testing tool, I have developed keyword driven framework that means instead of writing multiple functions to automate driven website, we have abstracted those functions to excel files and in that excel file we are giving the steps and the program written is going to drive best on the data excel set. In this way with this web testing tool, test cases are automatically tested by using Keyword driven framework.

Keywords-- Software Testing, Automation testing, Selenium webdriver, Keyword driven, Eclispe.

I. INTRODUCTION

Automation Testing is the use of testing tools and reduce the need of manual or human involvement, repetitive or redundant tasks. Selenium is a tool designed to generate automated tests and enhance the testing performance. Automated testing is used by software developer to save time and resources .Selenium is an open source automation testing tool for web based applications. It runs directly on browser and supports almost all available browsers such as Firefox, chrome, IE, Opera, Safari etc. It runs on all platforms such as Windows, Linux and Macintosh. It's a very useful tool for System functional testing and browser compatibility testing. It is really strong as compare to other available automation tools and is very flexible and simple to use. There are many languages supported in Selenium but the language in which your program is built is independent of the language being used by the website or the web based application, that means, if your website is build in Java then you can use selenium with mentioned languages and create your test scripts in any of these languages. You only need to know one language in order to work with Selenium. Selenium is a browser automation tool, commonly used for writing end-to-end tests of web applications. A browser automation tool does exactly what you would expect: automate the control of a browser so that repetitive tasks can be automated. It sounds like a simple problem to solve, but as we will see, a lot has to happen behind the scenes to make it work. Before describing the architecture of Selenium it helps to understand how the various related pieces of the project fit together. At a very high level, Selenium is a suite of three tools. The first of these tools, Selenium IDE, is an extension for Firefox that allows users to record and playback tests. The record/playback paradigm can be limiting and isn't suitable for many users, so the second tool in the suite, Selenium WebDriver, provides APIs in a variety of languages to allow for more control and the application of standard software development practices. The final tool, Selenium Grid, makes it possible to use the Selenium APIs to control browser instances distributed over a grid of machines, allowing more tests to run in parallel. Within the project, they are referred to as "IDE", "WebDriver" and "Grid". Selenium is a browser automation tool which lets you automated operations like: type ,click ,and selection from a drop down of a web page. Selenium provides a rich set of testing functions specifically geared to the needs of testing of a web application. These operations are highly flexible, allowing many options for locating UI elements.One of Selenium's key features is the support for executing one's tests on multiple browser platforms .[10]

II. PROPOSED WORK

In this section we will explain about how we can design and use keyword driven automation framework in Selenium Webdriver with Java along with example.

In Keyword driven testing, each keyword corresponds to an individual testing action like a mouse click, selection of a menu item, keystrokes, opening or closing a window or other actions. A keyword-driven test is a sequence of operations, in a keyword format, that simulate user actions on the tested application. In keyword driven automation framework we create the methods in Java that are mapped to the functionality of the application.

With the use of keyword driven framework , we can automate the following test scenarios for Gmail as under :-

1. User should able to logging in its account, when we are entering correct userid and password.
2. User should not be able to login in its account, when any one of userid and password is incorrect.
3. User should able to delete any mail available in Inbox.
4. User should able to compose and send mail.
5. User should be able to view the inbox mails
6. User should able to view starred
7. User should able to save the composed mail
8. User should be able to discard the composed mail
9. User should be able to view sent mail
10. User should be able to View drafted mails
11. User should be able to View spam mails
12. User should able to search mails , through search functionality available in gmail.

Snapshot of 1st test case scenario:

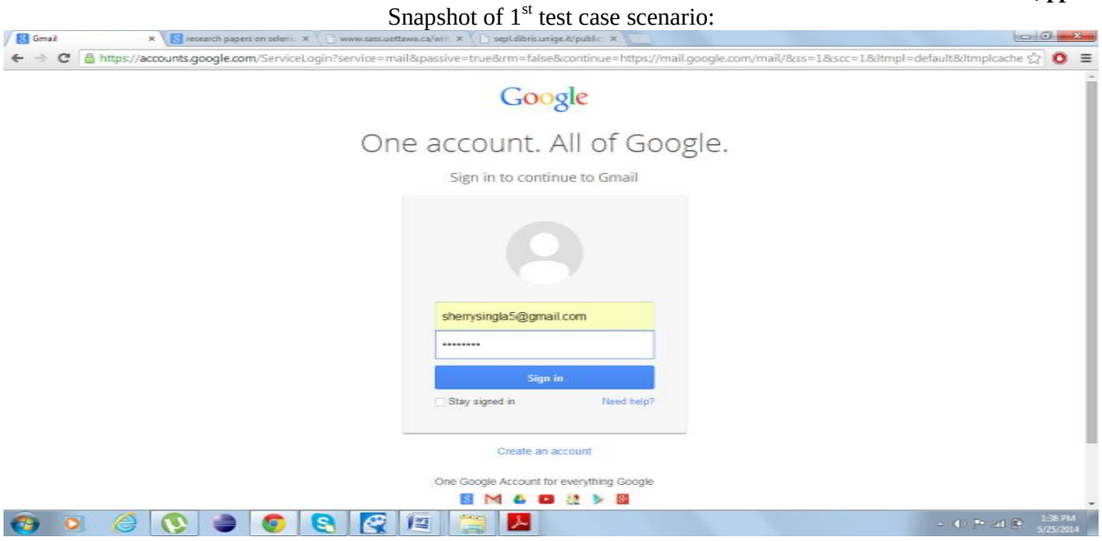


Fig 1 logging into gmail account

Snapshot For testing the other testscenario's for Gmail application.

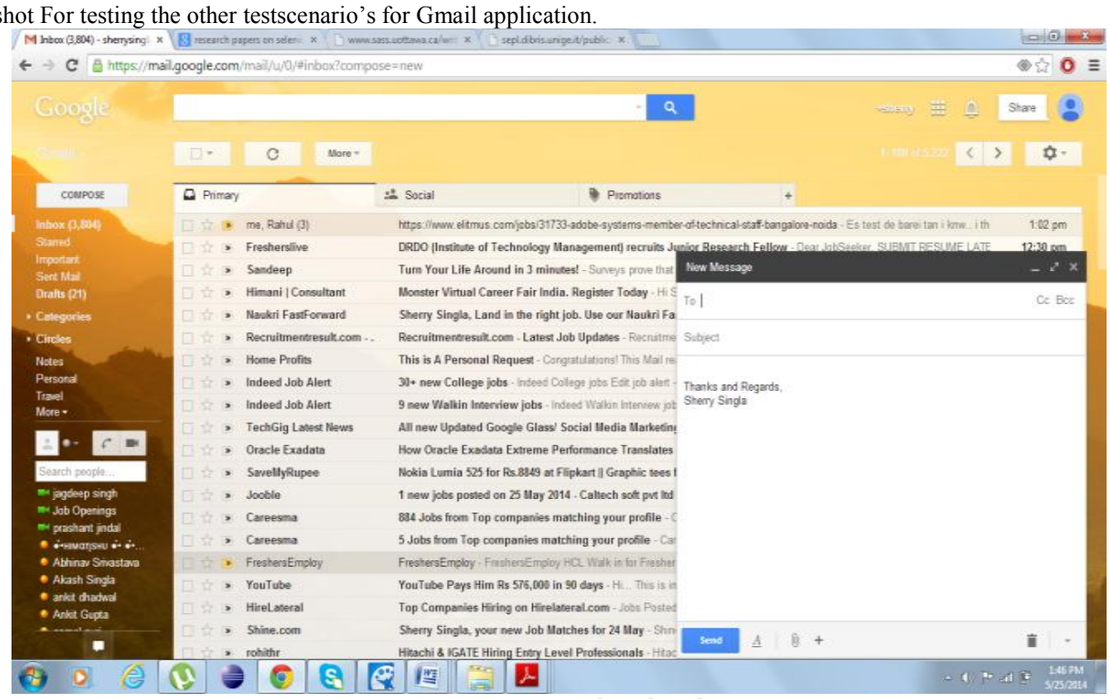


Fig 2 Compose and send mail

To automate the test cases for such web applications, We usually write the methods that perform specific task. For example we may write the Excel_IO method in Java which will call connection function for creating a connection string with the excel sheets and we may write fillscreen method to write something in textbox like username or password. Similarly we will create the methods for each functionality. The advantage of creating methods is that we can re-use these methods in multiple test cases with different input test data.This speeds up the automation process with increased productivity.

A. The Components of keyword driven automation framework in Selenium webdriver.

Each keyword driven automation framework has some common components as mentioned below[11].

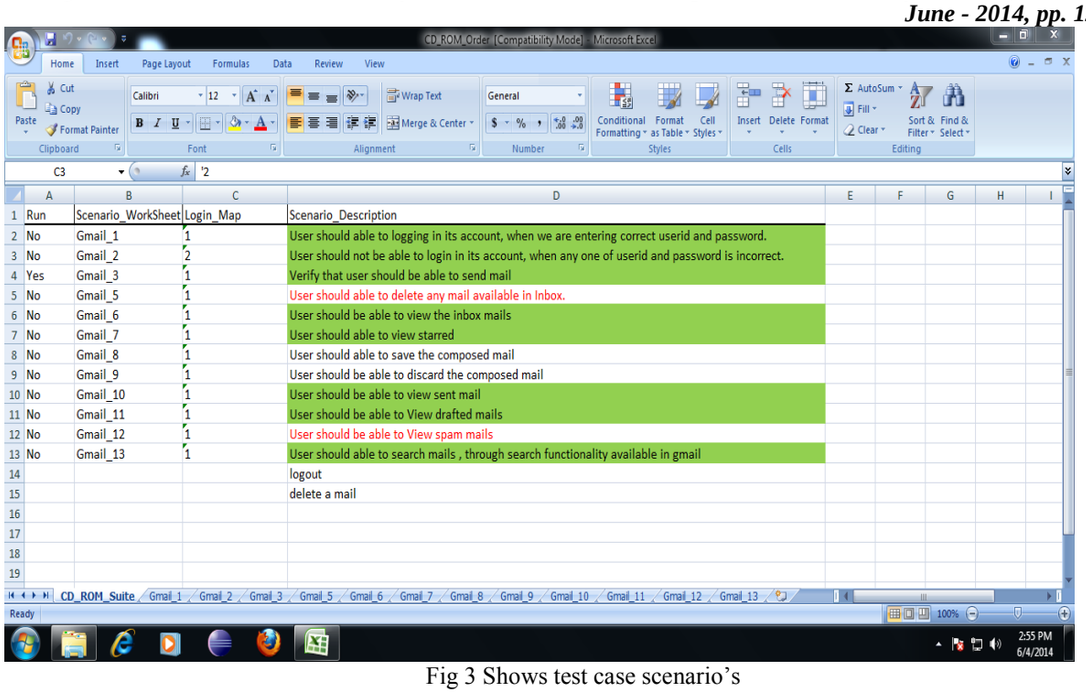
- Java Class Library with functionality specific methods.
- Test Data Sheet (generally in excel format)
- Selenium Webdriver with Java.
- Reports in HTML format
- Main Java driver script

1) Java Class Library with functionality specific methods: As explained earlier, we can create methods for each functionality in the application like select, fillscreen etc.

2) Test Data Sheet in Selenium Webdriver framework: As displayed in below figure, the data sheet contains below columns.

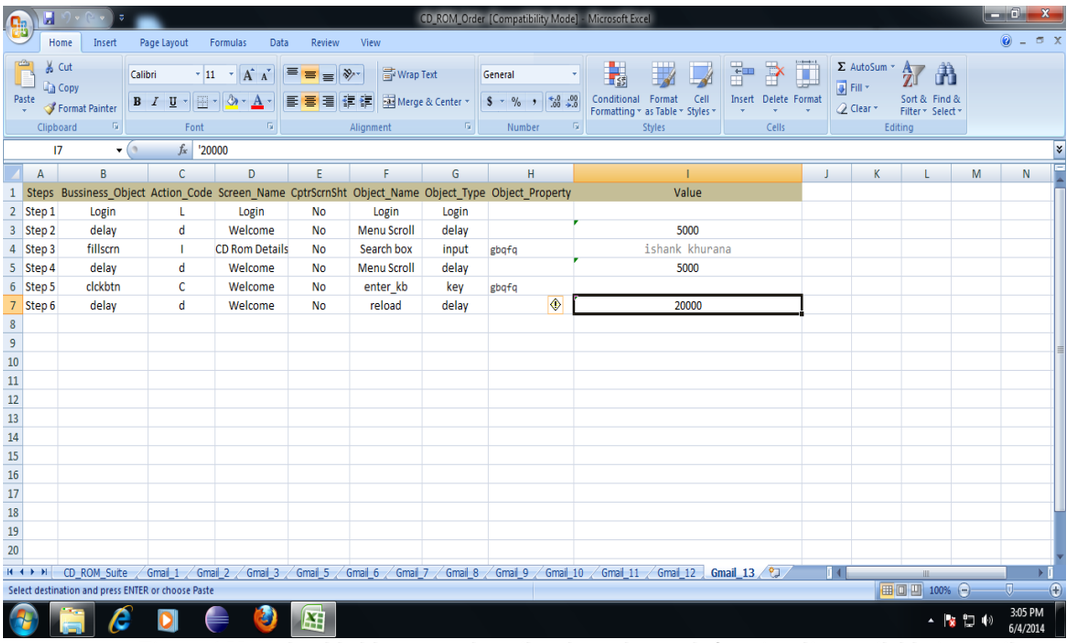
ID -Manual test case ID

1. Test_Case_Name - Name of the test case
2. Exec_Flag - Execution flag to tell if you want to execute this test case. Y means that test will be executed
3. Test_Step_Id - Step Id of the Test case
4. Test_Case_Steps - Step name of the test case
5. Keyword - The name of method in function library you want to call.
6. objectTypes - type of the web elements like webedit, weblist, webcheckbox etc
7. objectNames - Web element identification method and expression like xpath://td
8. objectValues - actual test data you want to enter in the webElement
9. parameter1 - extra parameter to drive the method control
10. parameter2 - one more parameter to drive the method control



	A	B	C	D
1	Run	Scenario_WorkSheet	Login_Map	Scenario_Description
2	No	Gmail_1	1	User should able to logging in its account, when we are entering correct userid and password.
3	No	Gmail_2	2	User should not be able to login in its account, when any one of userid and password is incorrect.
4	Yes	Gmail_3	1	Verify that user should be able to send mail
5	No	Gmail_5	1	User should able to delete any mail available in Inbox.
6	No	Gmail_6	1	User should be able to view the inbox mails
7	No	Gmail_7	1	User should able to view starred
8	No	Gmail_8	1	User should able to save the composed mail
9	No	Gmail_9	1	User should be able to discard the composed mail
10	No	Gmail_10	1	User should be able to view sent mail
11	No	Gmail_11	1	User should be able to View drafted mails
12	No	Gmail_12	1	User should be able to View spam mails
13	No	Gmail_13	1	User should able to search mails , through search functionality available in gmail
14				logout
15				delete a mail

Fig 3 Shows test case scenario's



	A	B	C	D	E	F	G	H	I
1	Steps	Bussiness_Object	Action_Code	Screen_Name	CptrScrnSht	Object_Name	Object_Type	Object_Property	Value
2	Step 1	Login	L	Login	No	Login	Login		
3	Step 2	delay	d	Welcome	No	Menu Scroll	delay		5000
4	Step 3	fillscrn	I	CD Rom Details	No	Search box	input	gbqfq	ishank khurana
5	Step 4	delay	d	Welcome	No	Menu Scroll	delay		5000
6	Step 5	clckbtn	C	Welcome	No	enter_kb	key	gbqfq	
7	Step 6	delay	d	Welcome	No	reload	delay		20000

Figure 4 User should able to search mails , through search functionality available in gmail

3) Selenium Webdriver in Java: This is the driver instance you will create to execute the test cases.Sample code to create a web driver instance is given below.

```
System.setProperty("webdriver.chrome.driver","F:\\selenium\\chromedriver.exe");
WebDriver driver =  new ChromeDriver();
```

4) Creating html Reports in Selenium Webdriver automation framework: You can create the html reports using file handling mechanism in Java.

Below code will delete the existing report file

```
if ((new File(c:\\reportfile.html)).exists() )
   (new File(c:\\reportfile.html)).delete();
```

Below code will append the data to the existing report file stored at given filePath

```
\public static void appendToFile(String filePath,String data) {
      //This function will be used to append text data to filepath
      try{
      File temp = new File(filePath);
     FileWriter  = new FileWriter(temp,true);
      fw.append(data);
      fw.close();
      }catch(Exception e){}
}
```

Here will need to import the classes in the package java.io to work with files.

5) Main driver script in Selenium webdriver automation framework: This is the main method and starting point for the framework code. The main responsibilities of this method are given below.

- Read the test case steps from the datasheet one row at a time
- Execute the method corresponding to the current step in the test case
- Log the verification points in the html report file.

B. Description about the reports are generated in selenium webdriver:

So far we had been doing Selenium tests without generating a proper format for the test results. From this point on, we shall tackle how to make these reports using a test framework called TestNG. The "NG" means "Next Generation".

1) Advantages of TestNG: There are three major advantages of TestNG :

- Annotations are easier to understand
- Test cases can be grouped more easily
- Parallel testing is possible

Annotations in TestNG are lines of code that can control how the method below them will be executed. They are always preceded by the @ symbol.

2) Multiple Test Cases: We can use multiple @Test annotations in a single TestNG file. By default, methods annotated by @Test are executed alphabetically.

TestNG is a testing framework that is capable of making Selenium tests easier to understand and of generate reports that are easy to understand.

III. RESULTS AND DISSCUSSIONS

First the various test scenario's for gmail application will be tested and the reports will be generated.

- TestNG can generate reports based on our Selenium test results.
- WebDriver has no native mechanism for generating reports.
- TestNG can generate the report in a readable format .
- TestNG simplifies the way the tests are coded.
- There is no more need for a static main method in our tests. The sequence of actions is regulated by easy to understand annotations that do not require methods to be static.
- Uncaught exceptions are automatically handled by TestNG without terminating the test prematurely. These exceptions are reported as failed steps in the report.

The Console window in Eclipse generates a text-based result while the TestNG window is more useful because it gives us a graphical output of the test result plus other meaningful details such as:

- Runtimes of each method.
- TestNG is capable of generating HTML-based reports.
- Annotations can use parameters just like the usual Java methods.

Screenshot of the report generated by selenium webdriver keyword driven framework:

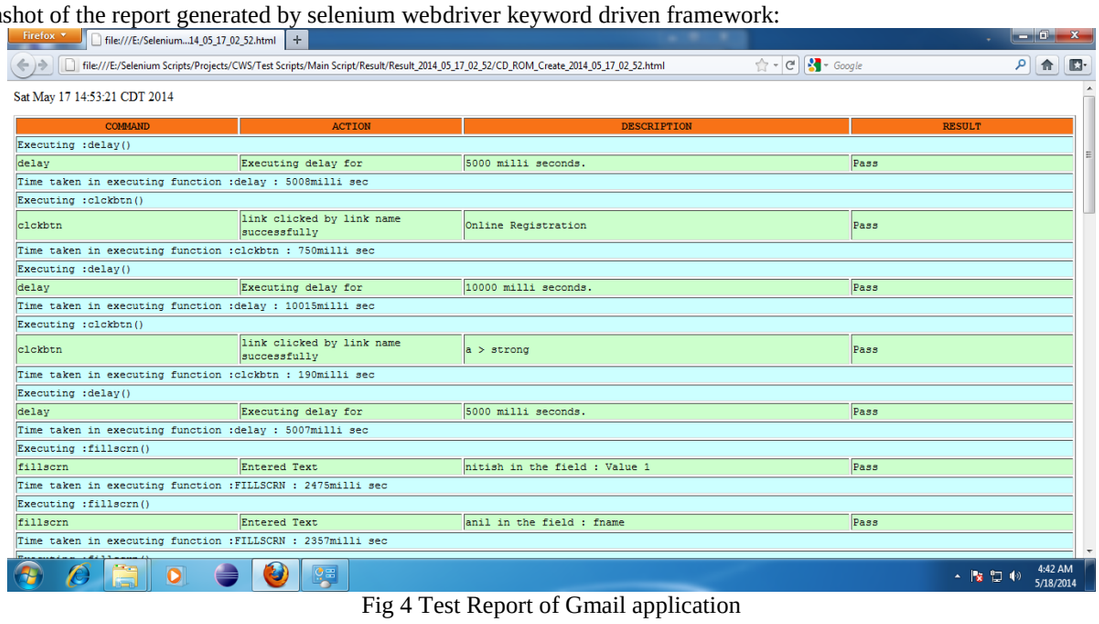


Fig 4 Test Report of Gmail application

- Command column: contains the name of the method that are executed .
- Action column: contains the actions performed by the method.
- Desciption column: contains the description about the method.
- Result column: etheir the test is passed or failed on that particular method.

These test reports also contains the execution time taken by particular method and also contains the screen shots of method. This provides report details for which test scripts have failed or passed while running a test suite. The summary report provides details of execution, duration, test start time and end time. The detailed reports describe exceptional cases handled, steps passed, and steps failed. This helps in performing effective analysis on the execution report.

IV. CONCLUSION

Selenium is a framework comprises of many tools used for testing web applications. In this paper, the Keyword Driven Framework has been created to perform Automation Testing for web applications using Software Testing Tool "Selenium Webdriver". By using Keyword Driven Framework, instead of writing multiple functions to automate driven website, we have abstracted those things to excel files and then in that excel file we are giving the steps and the program we have written is going to drive best on the data excel set and the entire functionality of our Application under Test (AUT) gets captured as step by step instructions for every test as well as in a table. In this way, test cases are automatically tested by using Keyword driven framework.

REFERENCES

[1] Boni Garcıa, " Automated Functional Testing based on the Navigation of Web Applications".
[2] Vishawjyoti and Sachin Sharma, " STUDY AND ANALYSIS OF AUTOMATION TESTING TECHNIQUES".
[3] Open 2 Test, "Test Automation Framework for Selenium Web Driver – Introduction".
[4] Nidhika uppal, "Enhancement in Selenium automation testing tool and functionalities.
[5] Harpreet Kaur, Dr.Gagan Gupta, "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete".
[6] Regina Ranstrom, "Automated Web Software Testing With Selenium".
[8]Anujajain,MS.Prabu,Swarnalatha,"Web-Based Automation testing Framework", International Journal of Computer Applications ,Vol 45,pp1-5,2012.
[9] P.Nirmaladevi,K.Rajeswar,"Effective Automating testing with -web application Using elenium",International Journal of Communications and Engineering,Vol-5,NO.5,pp 40-46 ,2012.
[10] Giuseppe A. Di Lucca & Anna Rita Fasolino(2006): Testing Web-based applications: The state of the art and future trends. Information & Software Technolgy.

