Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 341 – 346

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Analysis and Design of Selenium WebDriver Automation Testing Framework

Satish Gojare[a,*],Rahul Joshi[b],Dhanashree Gaigaware[c]

[a]*M.Tech student, Symbiosis International University, Gram-Lavale, Tal-Mulshi,Pune, 412115, India*
[b]Research Guide, Symbiosis International University, Gram-Lavale, Tal-Mulshi , Pune, 412115, India
[c]Research Scholar, Pune, India.

Abstract

Nowadays, number of software system has been implemented as web-based applications. These web applications are very complex. It is very difficult to test such complex web applications. Automation testing uses automation tools to reduce human intervention and repeatable tasks. In this paper we have designed and implemented automation testing framework for testing web applications. This new automation testing framework has been implemented using selenium WebDriver tool. Using this framework tester can easily write their test cases efficiently and in less time. Tester need not to study the selenium webdriver tool in detail. This framework is helpful to developer to analyze their code due to screen shot property of framework. This framework produces the customized test report to tester. It is very easy to maintain and repair the test suite for new release of the application using this framework.

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

*Keywords:*Web applications; Automation testing; selenium webdriver; Automation testing framework.

1. Introduction

The goal of software testing is to find defects in software as early as possible. Software testing consumes 30 to 60 percent of all life cycle cost, depending on product criticality and complexity [1].With the development of internet technologies, web applications became more popular. Nowadays large number of software systems has been

* SatishGojare. Tel.:+918087065620.
E-mail address:satish.gojare@sitpune.edu.in

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.038

implemented as web applications. The quality of these web applications is one of the important factors while deploying these web applications [3]. So to increase the quality of software, testing plays an vital role. Software development cycle becomes shorter and shorter; this makes the software testing more difficult. Manual testing is a time consuming process and it require human intervention. So to avoid these problems, automation testing came into picture.

Automation testing means to automate the testing process or activities including design and execution of test scripts and use effective software automation tools [4]. Automation testing improves the quality of software testing and minimizes the human intervention in software testing process. To support these tasks there are various commercial and open source tools available, such as Watir, JMeter, Selenium, QTP and many other. The Selenium automation tool considered most popular and open source tool for testing the web applications [7].In this paper we have proposed automation testing framework based on the selenium webdriver and TestNG tool.

2. Basics of Selenium and TestNG

Selenium is composed of multiple software automation tools such as, Selenium IDE, Selenium RC (selenium 1.0), and Selenium webdriver (selenium 2.0) [7]. Selenium IDE is an integrated development environment to build the test scripts. It is a Firefox plug-in allow you to record edit and debug the selenium test cases [4]. It records all actions performed by the end user and generate the test scripts. Selenium remote control (RC) was main selenium project for long time. Selenium RC is slower than the selenium webdriver because it uses the java script program called selenium core [7]. Selenium RC requires to start the server before executing the test scripts. It doesn't support the Ajax applications. To avoid the limitations of selenium RC, selenium webdriver has been invented by merging selenium and webdriver.

Selenium webdriver is also known as selenium 2.0 [7]. Selenium webdriver directly communicate with the browser, so selenium webdriver is faster than selenium RC. Selenium webdriver supports multiple web browsers and also support for Ajax applications. The main goal of the selenium webdriver is to improve support for modern web application testing problems. Selenium webdriver supports multiple languages to write the test scripts. Selenium webdriver's API is simpler than the selenium RC's [5]. However, despite all advantages of selenium web driver, it has some limitations when testing the web applications. Selenium webdriver does not have built in functionality to generate the screenshots for failure test cases. Selenium webdriver does not have built in capability to generate the test results. It depends on third party tools to generate the test reports. This limitation can be avoided by using TestNG framework.

TestNG is a testing framework designed to overcome the limitations of JUnit testing framework [8]. TestNG provides some new functionality that makes it more powerful than JUnit. TestNG covers all categories of tests such as unit, functional, integration testing. In the proposed framework we have integrated TestNG with eclipse to generate the test report and execute multiple test cases in parallel. This TestNG report contains all the passed and failed test cases. TestNG report is very tedious to understand, so it requires some modifications. Each organization has different requirements about the test report. In proposed framework we have customized TestNG report according to organization requirements. So, organization can get the test report as they want. This report also contains the link for failure test cases. By using this functionality, developer can easily find out defects in web application.

3. Design of Automation Framework

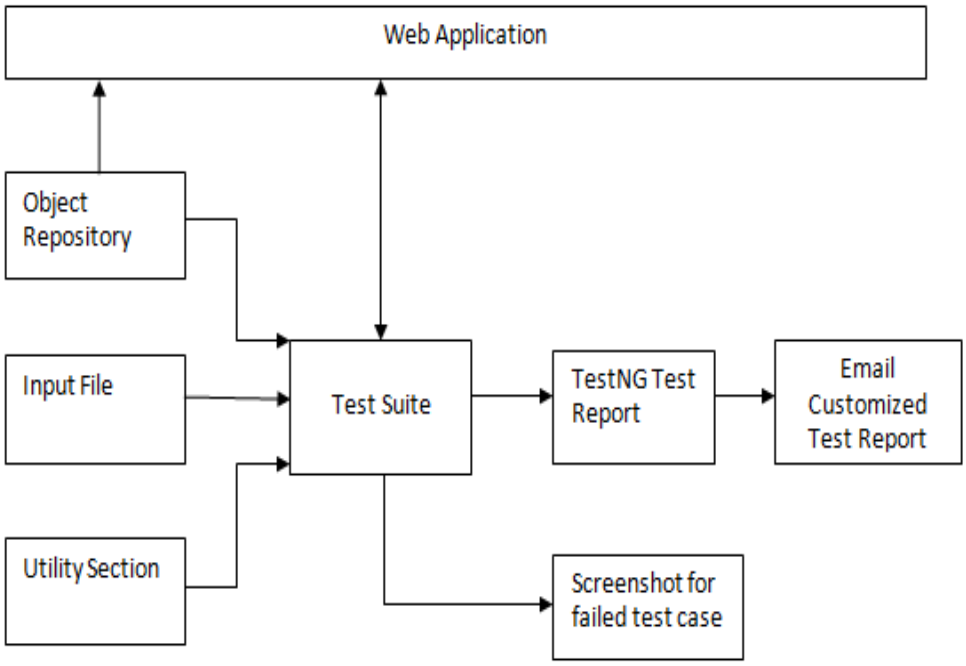


Fig.1. Architecture of Proposed Framework

4. Proposed Work

We have designed test automation framework based on selenium webdriver and TestNG testing framework (Fig. 1). The framework designed in this paper includes five components listed below.

- Object Repository
- Input File
- Utility Section
- Test Suite
- Customization Test Report

4.1 Object repository

Selenium webdriver supports various types of locator to locate the web page elements. Web page elements can be located by its id, link text, xpath or css locators. Object repository stores all the locators of web page elements. This will simplify the task of maintaining and repairing the test cases. For e.g. previous version of web application contains ‘Login’ button. In next version of web application ‘Login’ button changed to ‘Login Now’, so it is required

to change the all the test cases which contains the 'Login' button [2]. To avoid such kind of problems, we have implemented object repository which contains the id's, xpath and link text for all web page elements. Whenever tester writes the test case, tester will use the information to locate the web page element. This will reduce the maintenance cost of test cases. Whenever change occurs in web application elements, tester needs to change only object repository.

4.2 Input file

In Web application, end user needs to enter some information for e.g. Gmail login requires user name and password to login. Such kind of inputs stored in input file. Instead of entering same information in web application, tester may access these inputs from input file. In this file tester can store the input values required by web application.

4.3 Utility Section

Utility section contains two files which are described below.

1. **User Actions File:**Selenium webdriver doesn't support the direct functions to perform certain operations like clicking a button, selecting checkbox etc. This section contains the common functions like click button, select checkbox, click link etc. This will reduce the redundancy of code in test script. This file also contains the application specific functions for example if web application contains table and you need to verify particular column is sorted or not. To verify this kind of application specific functionality, separate functions can be created. These functions will be useful for writing different test cases.
2. **Utility file:** This file comprises the common functionality of web application like login and logout. In test suite tester need to login to web application to test internal functionality of application and log out after completion of suite it needs to logout. To avoid this kind of repetition, we have added login and logout functions in utility file.
3. **Screenshot Generation:** Selenium webdriver does not support the screenshot for failure test cases. We have implemented new function that will take the screenshot for failure test case only. Using this function tester can easily capture the error occurred in web application. This will also help to developer to analyze their failure. After execution of test suite, screenshots for failure test cases are stored in directory according to date wise folder.

   Steps to generate screenshot

   1. Create directory where you store the screenshot for failure test case.
   2. Capture the result from TestNG
   3. Verify the result is pass or fail
   4. If the result is fail then capture the screenshot for web page.
   5. Set the date and time for screenshot Image as title.
   6. Store the image file in defined directory.

4.4 Test suite

In Web application, end user needs to enter some information for e.g. Gmail login requires credentials to login. Such kind of inputs stored in input file. Instead of entering same information in web application, tester may access

these inputs from input file. In this file tester can store the input values required by web application

4.5 Customization of test report

Selenium webdriver doesn't support built in functionality to generate the report .Here we are using TestNG testing framework to generate the test report. TestNG generates the report in HTML format, which is tedious to understand. Some organization want specific format of report. So there is need to customize the TestNG report according to organization.

We have implemented one class in which we have captured the TestNG report. We have used iText library to generate the report in PDF format. IText is an open source library that allows you to create or manipulate PDF documents. This class implements ITestListener of TestNG framework. This report contains the link to the failure test case screenshot. So, one can easily verify the error page of web application.

4.6 Email customized report to respective person

After customization of test report, it needs to send to respective authority. Here we have used mail.jar for sending the mail to the respective person. Mail.jar file support multiple protocols like SMTP, POP3. After each test suite we have used the sendreport () method to send the test report.

5. Implementation Result

After use of implemented automation testing framework, the regression testing efficiency is highly improved. Tester can write the test cases twice than older approach of writing test cases. This minimizes the human resource required to test the web application. The maintenance cost of test cases also reduced due to centralized repository. As the version of web application changes, you need to change only object repository for newly added elements. Sometimes test cases are failed due to synchronization issues of selenium webdriver not due to web application defects. This framework reduces the error rate of failing the test cases due to synchronization issues. Ultimately passing rate will increases. This shows the accuracy of proposed framework over traditional approach of testing.

We have executed test suite of 250 test cases on student information system web application. After automation run we got the following results (Table 1.) in terms of overall pass rate, failure rate, execution time etc. Pass rate shows that test case gives the exact result as manual test case gives. Sometimes web application works correctly but test cases fails due to synchronization. Proposed framework synchronizes the test cases properly, so failure rate is reduced than traditional approach.

Table 1. Results of proposed framework and traditional approach.

Approach/ Parameter	Overall pass rate (%)	Overall failure rate (%)	No. of test cases per day	Execution time (hrs.)	Maintenance Cost
Proposed Framework	95.42	4.58	15	2.3	Low
Traditional Approach	71.7	29.3	8	3.5	High

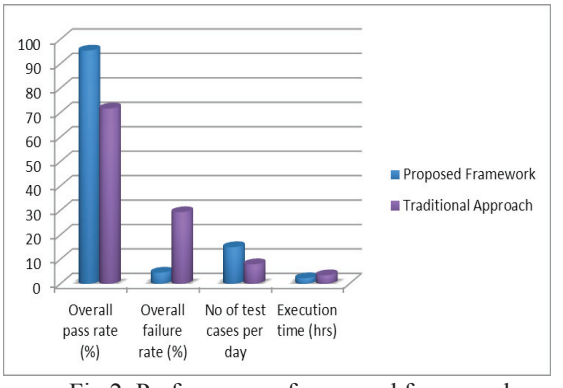
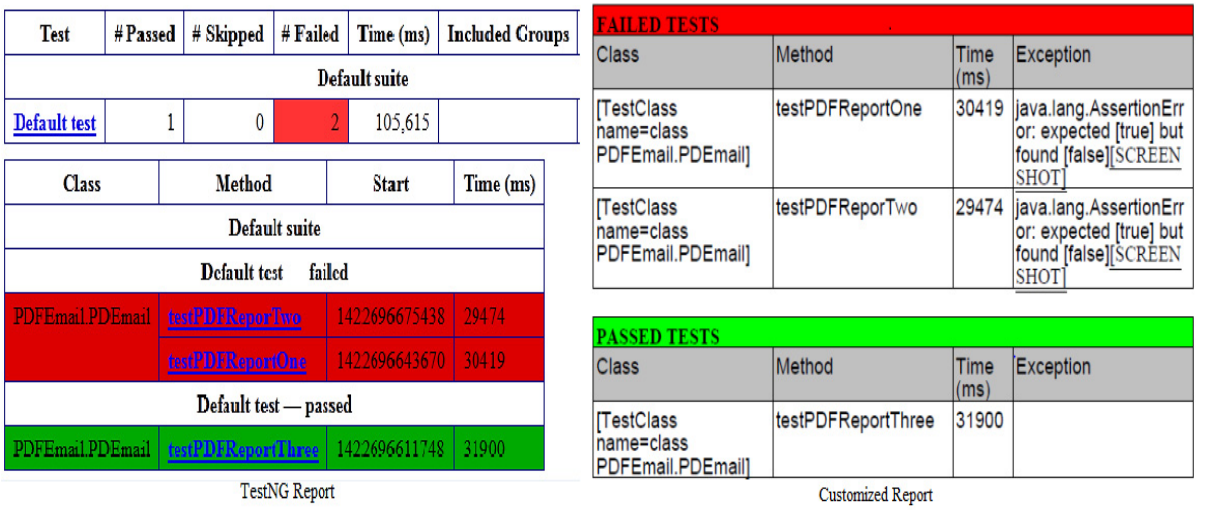


Fig.2. Performance of proposed framework

Test report is customized according to organization need; this improves the readability of test report (Fig.3). After automation run is completed customized reports are sent to stakeholders. These stakeholders can analyse details and failure if any. Screenshot property provides an efficient way to developer to fix their bugs by just clicking on screenshot link in test report.



Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups
Default suite					
Default test	1	0	2	105,615	

Class	Method	Start	Time (ms)
Default suite			
Default test failed			
PDFEmail.PDEmail	testPDFReporTwo	1422696675438	29474
	testPDFReportOne	1422696643670	30419
Default test — passed			
PDFEmail.PDEmail	testPDFReportThree	1422696611748	31900

TestNG Report

FAILED TESTS

Class	Method	Time (ms)	Exception
[TestClass name=class PDFEmail.PDEmail]	testPDFReportOne	30419	java.lang.AssertionError: expected [true] but found [false][SCREEN SHOT]
[TestClass name=class PDFEmail.PDEmail]	testPDFReporTwo	29474	java.lang.AssertionError: expected [true] but found [false][SCREEN SHOT]

PASSED TESTS

Class	Method	Time (ms)	Exception
[TestClass name=class PDFEmail.PDEmail]	testPDFReportThree	31900	

Customized Report

Fig.3. Result of the report customization

6. Conclusion

In this paper we have proposed new automation testing framework to test the web based applications based on selenium webdriver. In order to test the web application proposed automation framework surely reduces the time required to write the test cases and increase the pass percentage of test cases. It also reduces hectic workload of tester. By using this framework one can generate the customized test reports and also analyze the failures using screenshots of failed test cases. Tester can maintain the all data from central place. This framework is very useful for dynamically changing web applications. The automation test scripts are easy to understand using this framework. In this way automation framework helps organization to test web applications efficiently.

References

1. MacarioPolo,PedroReales,MarioPiattini. Computing Test Automation; IEEE Software, VOL. 30, NO. 1, January 2013.
2. Maurizio Leotta,DiegoClerissi,FilippoRicca,CristianoSpadoaro. Comparing the Maintainability of Selenium WebDriver Test Suites Employing Different Locators; ACM,2013.
3. AndrzaM,Giesel A. etl. Extension of Selenium RC Tool to Perform Automated Testing with Databases in Web Applications; Automation of Software Test (AST), 2013 8th International Workshop ,2013.125–131.
4. Sherry Singla, HarpreetKaur. Selenium Keyword Driven Automation testing Framework, International Journal of Advance Research in Computer Science and software Engineering, VOL. 4,Issue 6,2014
5. RigzinAngmo,Monika Sharma. Selenium Tool:A web based Automation testing Framework. International Journal of Emerging Technologies in Computational and Applied Science,2014.
6. Z. Wanadan,J. Ninkang,Z. Xubo. Design And Implementation Of A Web Application Automation Testing Framework; Ninth International Conference On Hybrid Intelligent Systems, 2009.
7. Selenium Documentation.[Online].(http://www.seleniumhq.org). (Accessed 15 DEC. 2014).
8. TestNG Documentation.[Online].(http://www.testng.org). (Accessed 25 DEC. 2014).
9. F. Wang., Du.A Test Automation Framework Based on WEB.IEEE/ACIS 11th International Conference on Computer and Information Science,2012, 683-687.