

Available online at <http://www.mecs-press.net/ijeme>

A Case Study of Web Based Application by Analyzing Performance of a Testing Tool

Hafsah Mahmood ^{a*}, Mehreen Sirshar ^b

^a Fatima Jinnah Women University, The Mall, Rawalpindi, 46000, Pakistan

^b Fatima Jinnah Women University, The Mall, Rawalpindi, 46000, Pakistan

Abstract

For the better business reason and to spare time and cash, automation testing is required. Software automated testing tools are being used by many companies for the sake of judging the performance, efficiency, and effectiveness of the software. There are assortments of apparatuses, which are accessible in the market. One of the best known instruments is selenium suite which is a blend of various automated testing apparatus. The paper presents a comparison made between selenium and other tools to find the best tool and on the basis of the results; a case study is presented to explore the performance of the selenium.

Index Terms: Software testing, Web-based application testing, Open-source testing software, selenium.

© 2017 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

1. Introduction

Testing a web application manually is a cumbersome task to do so nowadays, testers have moved on from manual testing to automated testing in order to save the time and cost as well. Test automation contains numerous focal points. There are varieties of tools that are available for test automation. Some of them are open source and can easily be used by anyone. Selenium is one of the most widely used open source test automation tool.

Selenium is an arrangement of various programming instruments each with a diverse way to deal with supporting test mechanization. Selenium IDE is a coordinated improvement environment for Selenium scripts which is executed as a Firefox module. It permits tests to be recorded, altered and repaired and by and large it is the most proficient approach to create test cases. It is a perfect environment for making Selenium tests with the autocomplete bolster and the capacity to move orders around quickly.

The next section of the paper presents the Software pattern followed by automatic generation of test environment in Section III. Furthermore a comparison of the Selenium with other automated testing tool is

* Hafsah Mahmood. Tel.: 03235138410

E-mail address: hafuumahmood@gmail.com

made in Section IV and V which is followed by the overall testing activities for web application in Section VI.

Section VII contains the optimal testing approach that is employed for the case study. Section VIII and IX contains the performance testing of the case study and analysis and results present in it. The last section is presenting the summary and conclusion.

2. Software Test Pattern

The key purpose of introducing a pattern in the software testing to eliminate the problems and issues the tester has to face and to advance the effectiveness and value of Software testing. Pattern is a nothing but an instruction that has a relationship with three elements. Those three elements are a particular environment, a problem and a solution for that specific problem. The software test pattern must contain some details in order to avoid any kind of confusion [1]. These details are name of the test case, type, scenario, problem, solution, implementation, effect, example and lastly but not the least related pattern. This will provide enough information about the test cases and the interaction of these test case components with each other. For testing a scenario, one has to make a choice among the right pattern that will be used for testing, then instantiate that pattern, use it and at the end evaluate it.

3. Automatic Generation of Test Environments

The automatic generation of the testing environment contains numerous steps. The first step is to analyze the target web application followed by the selection of those critical sections that need to be tested. As most of the testing is done on 20% critical section of the application so after the identification of those crucial sections, information regarding the test cases is dig out and constraints are premeditated. Once the data is obtained then the test cases are organized for execution [2]. These are containing the steps including test data generation, test drivers, along with deployment. All of the steps are automatic. Using the specified properties, a prototype system is generated that tests the quality of the web application.

4. Selenium

Automation testing devices pushes the tester to effortlessly computerize the entire testing process. This testing enhances the precision furthermore spare time of the tester as contrasted with the manual testing

Those things that are missed during the manual testing because of human error are given due consideration in Selenium test automation tool. It also provides a brief critique to the manual testing as well. One of the limitations of the Selenium suite is that it supports testing of only web-application [3].

Java Scripts infusion is used by the selenium to cooperate with the UI (user interface) components. For this purpose a server is used which amalgamate the java scripts files to the html file for the further analysis. Selenium RC manages the html source files that are available [4]. It records and replays the test very efficiently. On the off chance that one needs to make capable, program based backslide computerization and scale and disperse scripts transversely over various circumstances, then one have to use Selenium Web Driver; an amassing of lingo specific binds to drive a program.

Selenium is an easy tool and one can take full advantage of it if he is capable of writing good tests and keeping them self-sufficient and for finding a way to ensure the acceptance of the users towards the application that is generally known as acceptance testing. It provides the tester with the red green signals while testing that could give an instant feedback to the tester. This technique is used for the acceptance tests as well as the regression tests [5].

5. Alternative Open Source Automated Testing Software

There are many automated software testing tools. Some of them are discussed below in comparison with

Selenium.

- **SAHI**

SAHI is usually used by the non-programmers and professional testers. SAHI chips away at all programs and can specifically summon anything uncovered by Java script. SAHI is harder to begin since it needs establishment [6].

A portion of the elements of SAHI are In-program controls, savvy recorder, content based scripts, and AJAX bolsters rich inbuilt reports. SAHI keeps running as an intermediary server which catches movement from the web program and records the web perusing activities. Those recorded activities can be played by SAHI through infusing JavaScript into the program with the goal that components can be gotten to in the website page.

- **WATIR**

Watir is an open-source (BSD) library for computerizing web programs. It permits composing tests that easy to peruse and keep up in a straightforward and adaptable way [7]. In this one need to learn Ruby first for Watir and each program requires an alternate library for running experiments. Frames and pop-ups are open utilizing API as a part of WATIR.

- **QTP**

QTP is a business Tool presented by HP. It performs relapse and utilitarian testing through a UI of utilization. QTP underpins electronic application and additionally window based application. While assessing distinctive test computerization apparatuses the testing group ought to watch out who will create keep up and bolster test cases. QTP could fit one's group, in the event that one have cash to spend, QA analyzer will create and keep up experiments and testing will be outside of the web program.

- **Other Tools**

Along with that there are other tools as well including QuestionMark Perception, Fasttest Web, and Prometric IBT. All of these tools support the paper and pencil delivery options. These types of delivery options are traditional ways of testing. Weblock delivery option is provided in all tools under consideration in which the browsers have no copy paste access as they are locked. LMS or third party delivery is only supported by QuestionMark. Prometric IBT supports an additional delivery option that is called gated test delivery [8]. Similarly Moodle XML is an option that is compatible with all the three type of tools.

HP UFT is best where information security is required even while testing. Be that as it may, a noteworthy detriment of HP UFT is its cost. As HP UFT is a testing instrument, we have to pay a higher cost for this device. Soap UI is a utilitarian API testing instrument as well as gives us a chance to perform non-useful testing, for example, execution and security test [9]. There is no issue of high cost. Soap UI is likewise accessible as Soap UI Pro, which incorporates a few timesaving highlights went for making the testing speedier and simpler. Soap UI is open source while we can utilize its propelled rendition Soap UI Pro with lesser cost than TC and HP UFT [10].

The table below shows the comparison of all these testing tools on the basis of some parameters.

Table 1. Comparison of Automated Testing Tools

Parameters	Selenium	SAHI	WATIR	QTP
Language Support	C#, Groovy, Java, Perl, PHP, Python, Ruby, JavaScript	JavaScript	BSD Library	VBScript
Ease of Use	Requires a little programming skills to start testing	Requires programming skills to customize	WATIR is easy to use. It empowers you to connect to databases, trade XML records, read documents and spreadsheets, and arranges your code as reusable libraries.	QTP is flexible to use and it can easily be used for functional and regression testing.
Customization	Highly customizable and customized defects reporting	SAHI is highly customizable and one can customize reports easily.	WATIR can customize HTTP requests to servers.	QTP can customize the controls by using its custom proxy server documentation
Hardware Requirement	4x Dual-Core AMD Opteron	1.5 GB RAM required	Linux with latest version of Xcode 4.4 at least	3 GHz processor required
Cost	Free of cost	Free of Cost	Free of Cost	Trial version free for 30 days

The choice of the automated testing tool for this case study is Selenium. It will be used with amalgamation of WebDriver and WebLOAD with the purpose to create virtual users for execution of performance testing.

6. Testing Activities for Web Applications

The functional requirement testing can be connected to the conventional testing strategies in software testing. Then again, the non-functional requirement testing is extraordinary in testing an electronic application because of its dynamic surroundings that the web application collaborates with. In this manner, our study is to concentrate more on the non-functional testing [11]. Table 2 abridges seven distinctive testing exercises that are generally utilized as a part of non-functional requirement testing:

Table 2. Types of Non-Functional Testing

Testing Type	Description
Performance Testing	Performance testing is the way toward deciding the speed or viability of an application.
Load Testing	Load testing is the way toward putting request on a product framework or processing gadget and measuring its reaction. Load testing is performed to decide a framework's conduct under both ordinary and peak conditions.
Stress Testing	This test is utilized to assess web application conduct when it is pushed past typical and pinnacle load conditions. Not at all like performance testing and load testing, the objective of this testing is uncovered deformities under substantial load conditions, for example, memory holes and synchronization issues
Compatibility Testing	This testing is a kind of programming testing used to guarantee compatibility of the framework/application/website built
Usability Testing	This testing is a technique by which clients of an item are requested that play out specific assignments with an end goal to gauge the item's usability, undertaking time, and the client's view of the experience.
Accessibility Testing	Accessibility testing is a testing where in the clients under thought are individuals with all capacities and handicaps.
Security Testing	Security testing is a procedure planned to uncover imperfections in the security components of a data framework that ensure information and keep up usefulness as proposed.

The performance testing and load testing is given due consideration in web based application testing [12]. In reality, both performance testing and load testing could recognize potential issue territories bringing on poor execution when a web application keeps running at the typical load conditions. Our main focus will be on performance testing.

7. Generating Optimal testing Procedures based on Case Study

The initial step to create ideal testing techniques is to comprehend the online application framework to be tried and the goals of testing. In this case the application is PhotoSnack. It is a web application that helps the non-professional photographers to create a tempting slideshow to make it attractive and impressive. It is particularly used by those industrialists who want to do marketing of their products in a unique manner and are having shortage of money. So this application can prove to help them in this regard. As to adequately accomplish the goals of the testing, in the first place, we recognized the most widely familiar and critical practical strides that are relied upon to be performed on the PhotoSnack application. We then arranged these center useful strides into a testing system. Performance testing will be conducted on the application which included 17 test cases. The quantities of experiments utilized as a part of test system depend on the most widely recognized and general capacities that the PhotoSnack should deliberately and strategically accommodate.

All the testing will be done on Selenium. WebLOAD's coordination with Selenium, the open source testing instrument for web program applications is used that gives one a chance to incorporate genuine web programs that perform genuine action into performance test situations and precisely distinguish issues [13]. Utilizing the combination, one can stretch his framework utilizing WebLOAD's virtual clients to produce the heap, and at the same time run Selenium web programs that measure genuine customer conduct. WebLOAD gathers all test measurements from both the Selenium WebDriver and the backend servers, to give one an entire test examination report.

8. Performance Testing

Table 3. Results of Performance Testing

Test Identifier	Test Case Description	Average Mean Testing Time		
		1 users	50 users	100 users
Test 1.1	Query about upload photos	4.943	9.336	11.780
Test 1.2	Query rate records	10.931	14.925	12.928
Test 1.3	Query about sign up	3.125	8.564	7.123
Test 1.4	Make new Slide Show	5.967	9.802	8.087
Test 1.5	Import Pictures from Facebook	13.0097	24.4221	31.0985
Test 1.6	Import Pictures from Instagram	11.469	20.245	30.432
Test 1.7	Import Pictures from Flickr	3.7498	5.7356	7.5828
Test 1.8	Save slideshow	8.657	9.0023	29.765
Test 1.9	PhotoSnack Subscription query	3.3539	5.0705	7.5828
Test 1.10	Snack tools VIP Subscription query	6.0070	9.4221	11.0885
Test 1.11	Snack tools Business Subscription query	3.6577	6.4866	8.2078
Test 1.12	Query for selecting templates	5.9556	9.6926	8.0769
Test 1.13	Query for uploading videos	14.9091	22.9227	32.7347
Test 1.14	Query for downloading templates	5.3945	7.6137	6.7617
Test 1.15	Query for sharing slideshow on Facebook	3.7019	8.6563	9.4772
Test 1.16	Query for sharing slideshow on Instagram	4.9433	9.3366	11.7802
Test 1.17	Query for sharing slideshow on Flickr	5.423	8.4543	10.234

Each of 17 functional tests performed each one in turn with three distinctive client levels, 1 client, 50 clients and 100 clients, by doing likewise steps all the while. The average testing time demonstrates the time to pursue every test running the test different times. By and large, the average mean testing time fundamentally expanded

when it was tried with 1 client contrasting when test with 50 clients and 100 clients. Conversely, the testing time modestly expanded when tried with 50 clients looking at when tried 100 clients. The three experiments, Test 1.2, Test 1.3, and Test 1.14, are the main experiments the server prepared quicker with 100 clients than with 50 clients. A portion of the experiments experienced more normal testing time than the others. This demonstrates a few cases require fundamentally more CPU use than the others.

9. Analysis and Results

The case study on the PhotoSnack web based application is presented by conducting performance testing on the mentioned application by using Selenium. As it can be seen from the testing that the application was efficient when a single user was accessing it and performing a number of tasks on it. The average time is minimum in case of 1 user (see figure 1). Whereas as soon as the number of users are increased, the average mean time is increased as well. It is moderate in case of 50 users for all the test cases where as it is even much amplified when the users are double to 100. Therefore it concludes that by the increase in the number of simultaneous users, the performance of the application is reduced and the response time is magnified. As far as the testing mechanism is concerned, the Selenium tools proofs out to be one of the best tool for testing the performance of the web application under peak conditions. It can also be seen that the Selenium tool effectively find out all the errors and performance issue that was present in the web application under study. The amalgamation of Selenium with external tools leads it to outstanding performance. Hence it is a great, efficient and effective tool which could be used for creating virtual users and conducting performance testing on many web applications.

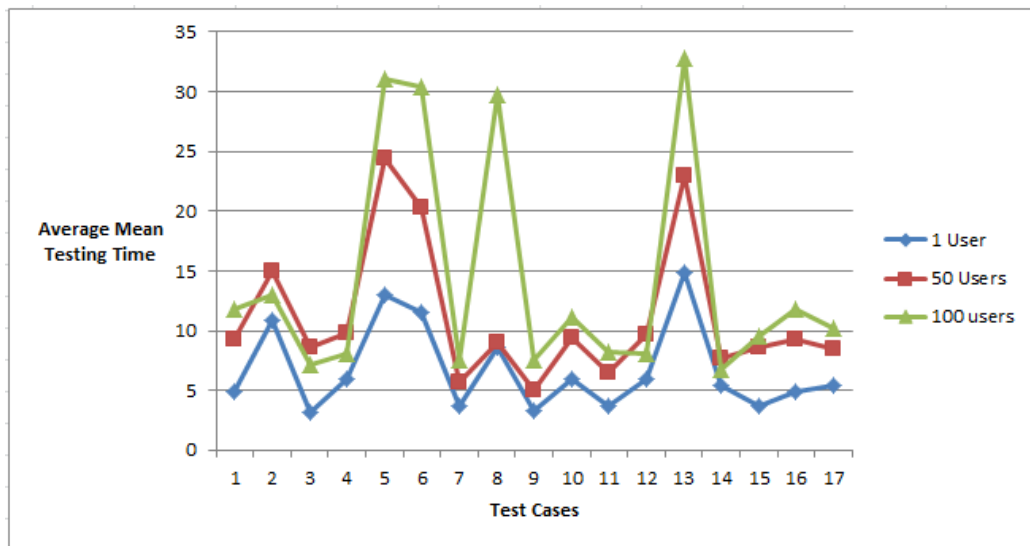


Fig.1. Graph showing average Mean Time w.r.t 1, 50 and 100 users

10. Summary and Conclusion

We utilized a contextual analysis from a photo slideshow making web application to show how to effectively test a mind boggling online application to convey the application with the top notch electronic application notwithstanding when just constrained IT asset is accessible to the testing group. The outcomes demonstrate that it can be a practical system to utilize an open source web source application testing apparatus to better

oversee and consult with partners by as yet testing an online application and figuring out how to convey it with the high caliber, particularly when there is a requirement of tight or restricted spending plan.

The advantages of utilizing open-source testing software incorporate ease, programming customization and code reusability. The open source testing instrument utilized for this study is Selenium, which encourages quick sending as far as simplicity of portion and straightforwardness in running experiments, and gives load test independency to analyzers. These two components appear to be extremely pivotal to test online applications. Regularly, numerous business web testing softwares, for example, SAHI, WATIR and QTP require broad comprehension of how to run and introduce the product, and are more costly to convey in an online application testing environment. For this situation examine, Selenium ends up being extremely practical and helpful to recognize the potential execution issues before real generation or usage.

In future it is expected to extend the work by analyzing the web based application on other tools as well and make a comparison.

Acknowledgements

This research was supported by IEEE gathering distributions, ACM and springer's productions. We thank our associates from Fatima Jinnah Women University, the mall road, Rawalpindi who gave understanding and expertise that fundamentally strengthened the examination, however they may not settle with the greater part of the clarifications of this paper

References

- [1] N. A. B. Thanuja Janarthana Naidu, "SAHI vs. Selenium: A Comparative Analysis," IEEE, 2014.
- [2] F. R. F. Rosnisa Abdull Razak, "Agile Testing with Selenium," IEEE, 2011.
- [3] A. H. a. M. Kellogg, "Automating Functional Tests Using Selenium," IEEE, 2006.
- [4] M. M. S. Ms. Rigzin Angmo, "Performance Evaluation of Web Based Automation Testing Tools," IEEE, 2014.
- [5] G. A. M. E. F. C. A. C. D.-N. Andreza M. F. V. de Castro, "Extension of Selenium RC Tool to Perform Automated Testing with Databases in Web Applications," IEEE, 2013.
- [6] J. A. D. F. u. R. M. R. Insha Altaf, "SURVEY ON SELENIUM TOOL IN SOFTWARE TESTING," IEEE, 2015.
- [7] W. Y.-k. Wang Yi-chen, "The Research on Software Test Pattern," IEEE, 2011.
- [8] H. U. Saleh M. Alsaleem, "Analysis and Evaluation of Web-Based Testing Tools," IEEE, 2015.
- [9] S. H. K. a. H. S. Kim, "Automatic Generation of Testing Environments for Web Applications," IEEE, 2008.
- [10] N. U. A. S. Himanshi, "Automation Testing: An Introduction to Selenium," IEEE, 2015.
- [11] H. V. K. V. Arya, "Keyword Driven Automated Testing Framework for Web Application," ACM, 2009.
- [12] W. X. B. K. B. W. E. W. Dianxiang Xu, "Mining Executable Specifications of Web Applications from Selenium IDE Tests," IEEE, 2012.
- [13] A. K. Preeti Yadav, "AN AUTOMATION TESTING USING SELENIUM TOOL," Springer, 2015.
- [14] R. J. D. G. Satish Gojarea, "Analysis and Design of Selenium WebDriver Automation Testing Framework," Science Direct, 2015.
- [15] M. E. A. A. S. Akram Hedayati, "Investigating into Automated Test Patterns in Erratic Tests by Considering Complex Objects," MECS, 2015.

Authors' Profiles

Hafsa Mahood (born October 1, 1995) is a Software Engineer by profession and an article writer by hobby. She is a technology, SEO, and Content writer. She is presently working with Discreet Solutions, Content Verge and OVC technologies as a writer. In addition, she is also a member of Character Building Society, Youth Burst, Inventive Leads and Firefox Mozilla. She is a Chairperson at IEEE Women and Engineering Chapter.



Mehreen Sirshar is a Lecturer at Fatima Jinnah Women University. She has done her BS from Fatima Jinnah Women University. She has done MS in software engineering from NUST and now enrolled in PHD in Computer Software Engineering. She has 43 international publications in various science and technology journal.

How to cite this paper: Hafsa Mahmood, Mehreen Sirshar, "A Case Study of Web Based Application by Analyzing Performance of a Testing Tool", International Journal of Education and Management Engineering(IJEME), Vol.7, No.4, pp. 51-58, 2017.DOI: 10.5815/ijeme.2017.04.06