

A Cross Platform Framework for Functional Testing of Web Applications

B. A. Harshitha¹, G. Shobha²

¹M.Tech. Student, Dept. of Computer Science and Engineering, RV College of Engineering, Bengaluru, India

²Professor, Dept. of Computer Science and Engineering, RV College of Engineering, Bengaluru, India

Abstract: The main aim of this paper is to propose a design for test automation framework to test web applications. This framework is platform independent and uses Selenium to test the functionality of elements present in web applications. This framework is used to create test cases based on the parameter names, values and actions of the elements and not based on the keywords present in the web application. The framework generates test report with screenshot of all the test steps executed. The report contains details of all the test steps and their status of execution.

Keywords: Test automation, framework, Functional testing, and Web applications.

1. Introduction

Functional testing is very important for any software that is developed. It is better to test the functionality during the development so that the bugs can be rectified in earlier stages and the cost will be less. Functional testing of web applications deals with testing the functionality of each element present in the web application. All the elements must behave as it is designed to be.

Manual testing is a very tedious job. It requires lot of time and effort to test all elements of a web application manually by performing click events. Automation is the best thing that has happened to the testing field. Automation reduces the time, cost, and allows the user to test the application in multiple platforms at the same time, which is not an easy task to do manually. This paper proposes a test automation framework, which will reduce the task of manual testers and increases the efficiency of the test results. This framework can run multiple test cases continuously for hours which can increase the number of test cases being tested and the time in which the test will be completed will be very less compared to manual testing. This framework has its own UI which can be used to create test cases. The test case consists of parameter names, values, test data and actions of different web elements.

The organization of the paper is as follows: Section II describes the work done so far in automation testing frameworks. In section III, the proposed methodology is explained. Section IV discusses the implementation and results of the work and finally in section V, the conclusions and future implementation of this work is discussed.

2. Related work

The following related work was studied in order to analyze different automated testing methods or frameworks, which are available in the field of automated software testing.

A testing framework for testing the web applications was proposed [1]. The framework performs testing based on keywords and uses Selenium automation tool for this. Selenium is used to import the keywords into the framework from an excel sheet. This paper tests a well-known web application, www.Flipkart.com. Series of tests executed using this framework shows the online shopping on flipkart and the test results are displayed in a window. The results show whether the testing process passed or failed.

In order to test the web applications, an adaptive framework of keyword driven automation testing was proposed [2]. The language used to code the commands in this framework is XML. Their adaptive framework consists of three distinct layers, namely automation layer, test driver layer and test execution layer. As the test script is executed, the interaction among these layers are stored simultaneously in a test log. To verify the working of the framework they also presented a prototype.

The evolution of the automated testing methodologies over time is explained in [3]. Record and playback, functional decomposition, data driven approach and keyword driven approach are various test automation techniques described. They further explained that the methodology of keyword testing is the vital approach in the today's scenario. The authors describe the various advantages and disadvantages of the above-mentioned automated testing technique. By discussing the essential parts of the process, which are, keywords, object repository and keyword driven module, the authors presented an in depth study of the keyword testing approach.

3. Proposed methodology

This Web Functional testing automation framework has the ability to perform complex set of tasks repetitively with same amount of accuracy every time, which makes it both cost effective, and reliable solution for functional testing. This framework is platform independent. The framework consists of GUI which helps to create input file that is to be given to the

engine for executing test cases/ test suite. The backend engine helps to automate the process of testing and overcomes the cons of manual testing by executing the test scripts or a test suite and generating the reports accurately with details of their time of execution, screenshots of each test step and other test step specific details. Thus, efficient checking of functionality is done using this framework.

The steps involved are as follows:

- *Step-1:* Develop the GUI that allows the testers or executors or admin to login to the tool and an option to register for new users.
- *Step-2:* The GUI should have an option to create new project, new test case, group the test cases for sprints.
- *Step-3:* Testers should create functional test cases in the GUI that is provided as input to the backend process.
- *Step-4:* The backend consists of Java Selenium code that reads the URL and performs different actions as mentioned in the test cases provided as input and generate reports with screenshots and details of all the actions performed and their success or failure.
- *Step-5:* Repeat testing the website with different test cases.

A. Proposed framework architecture

The proposed framework is based on the idea of testing the web application using parameter names and values of the elements. These parameters have meaningful name and actions of the elements are used in the test cases to perform the required action. The proposed framework works with selenium web driver that is an open source tool. Hence, it is a cost efficient framework. Fig. 1 shows the architecture of the proposed framework.

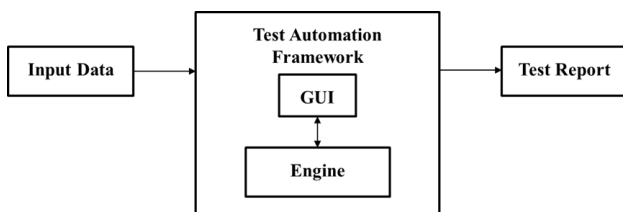


Fig. 1. System architecture of test automation framework

4. Implementation and results

A. Implementation

The UI of the framework is built using HTML, JavaScript, PHP and the backend engine is built using Selenium with Java in Eclipse IDE. First, the user needs to log into the framework or register if he is a first time user. There are three types of users namely, admin, tester and executor. The testers can create and execute test cases where as executors can only execute and not edit any test case. Fig. 2 shows the login page of the framework.

The user can either create a new project or open an existing project. If the user does not know any elements of the web

application, he wishes to test he might not be able to create a test case. Hence we have included an option to use Selenium IDE which will capture the clicks performed by the user in the web application to be tested and creates a test case based on those clicks. The created test case can then be executed using the framework. Fig. 3 shows the screenshot with options of creating new project or opening existing project.

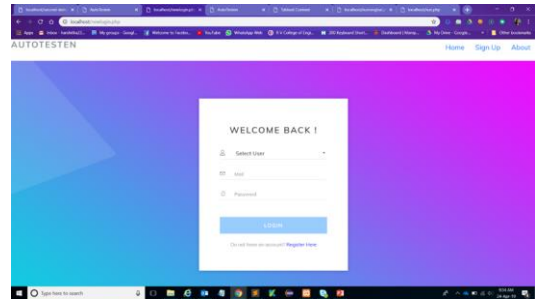


Fig. 2. Login page of the framework

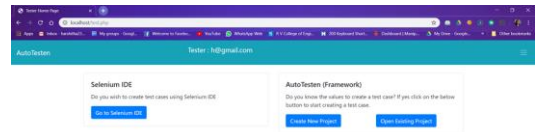


Fig. 3. Screenshot of options to create new project or open existing project

The user can create a test case by entering parameter names, values, test data and actions of different elements of the web application in the table provided by the framework. On saving the test case, a csv file will be created and saved in the test cases folder of the project. Fig. 4 shows the test case creation page.

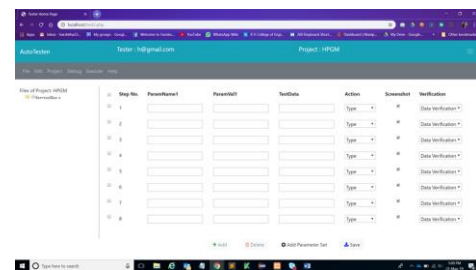


Fig. 4. Screenshot of test case creation page

The user can also import existing test case into the project test cases folder by clicking on import button in the GUI of the framework. The test case can be edited and saved again. The user can select desired test cases for execution and click on execute button to start the execution. The click of execute button will run the Selenium Java code that will read the configuration file which has information on the browser to be used, wait time and the application URL that needs to be tested.

The engine will launch the web browser with the help of Selenium driver and loads the URL to be tested. It then reads the csv files of the test cases and performs required actions based on the data in the test cases. It captures screenshots after the completion of each step and updates the status as passed or failed next to corresponding test step in the test report that is generated. Fig. 5 shows the sample test report generated.

EXECUTION DETAILS FOR: @@ScreenshotCheck

Test Case #	Start Time	Status	End Time	Screenshot	Test Step Detail	Exception
1	15/04/2019 14:21:34	Passed	15/04/2019 14:21:34	@@ScreenshotCheck_@@ScreenshotCheck_1	Function exec is invoked with Base Url is Call	
2	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_2	Function exec is invoked with TypeId Position Type is Call	
3	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_3	Function exec is invoked with TypeId Position Type is Call	
4	15/04/2019 14:21:40	Passed	15/04/2019 14:21:40	@@ScreenshotCheck_@@ScreenshotCheck_4	Function exec is invoked with TypeId Position Type is Call	
5	15/04/2019 14:21:40	Passed	15/04/2019 14:21:40	@@ScreenshotCheck_@@ScreenshotCheck_5	Function exec is invoked with TypeId Position Type is Call	
6	15/04/2019 14:21:40	Passed	15/04/2019 14:21:40	@@ScreenshotCheck_@@ScreenshotCheck_6	Function exec is invoked with TypeId Position Type is Call	
7	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_7	Button click event is invoked with parameter text with value PROCESSED	
8	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_8	Type Click event is invoked with parameter text with value Home	
9	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_9	Type Click event is invoked with parameter text with value Orders	
10	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_10	Function exec is invoked with TypeId Position Type is Call	
11	15/04/2019 14:21:37	Passed	15/04/2019 14:21:37	@@ScreenshotCheck_@@ScreenshotCheck_11	CSI click event is invoked with parameter text with value a default.com	
12	15/04/2019 14:21:38	Passed	15/04/2019 14:21:38	@@ScreenshotCheck_@@ScreenshotCheck_12	Function exec is invoked with TypeId Position Type is Call	
13	15/04/2019 14:21:38	Passed	15/04/2019 14:21:38	@@ScreenshotCheck_@@ScreenshotCheck_13	Function exec is invoked with TypeId Position Type is Call	
14	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_14	Function exec is invoked with TypeId Position Type is Call	
15	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_15	Function exec is invoked with TypeId Position Type is Call	
16	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_16	Function exec is invoked with TypeId Position Type is Call	
17	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_17	Function exec is invoked with TypeId Position Type is Call	
18	15/04/2019 14:21:39	Passed	15/04/2019 14:21:39	@@ScreenshotCheck_@@ScreenshotCheck_18	Function exec is invoked with TypeId Position Type is Call	

Fig. 5. Screenshot of test report generated

B. Experimental results

The user creates many test cases for different web applications. The engine will launch each application, perform the tests and generate a single report for all the tests performed in one execution cycle. The report consists of the execution summary that states the total number of test steps executed with the status of how many passed and how many failed. Fig. 6 shows one such execution summary.

EXECUTION SUMMARY

Total Test Step Count:	14	100%
Passed:	14	100.00%
Failed:	0	0.00%

Fig. 6. Execution summary

To test the normal working of the framework a few applications of various organizations were used such as:

- *SAP Hybris Commerce application:* In this application, the framework was used to check whether orders that are placed by the customers are reflected as placed or not.
- *Service Max:* Field Service Management Solution (Sales Force Platform Product): This application is first tested to see if it is compatible with the developed framework. In this application, a remote service work order functionality is being tested by creating a remote service work order and assigning a service provider

employee to that work order.

- *Philips USA and UK shopping websites:* The framework developed is used to test the Philips shopping websites of UK as well as USA. The tests performed are, comparing two products, ordering one with dummy card details, applying a discount coupon for a product, and then ordering the product as a guest.
- *Other public websites:* Like logging into a tour-booking website, exploring several pages in that website and booking a return flight with all the details, and then paying the amount by dummy card details.

5. Conclusion

The Framework is created for cross platform functional testing of web applications. It has features like multiple browser support (Chrome, Firefox and IE), converts the selenium IDE generated scripts to our engine acceptable format, all interactions occur only on visible elements, test stability and high repeatability, and allows manually implementing customized timings if needed. Flexible design, quick and easy set up an execution.

It allows reusing code in the tests. Supports data and keyword driven approach to simplify tests and make them readable. Easy setup process. Easy-to-use tabular test data format so all these features makes the implemented framework a very useful one. The test reports generated will have details of each step and the passed or failed status of the execution. Screenshots are captured after execution of each step.

References

- [1] K. Y. Arya, H. Verma, "Keyword Driven Automated Testing Framework for Web Application", in proceedings of 9th IEEE International Conference on Industrial and Information Systems (JCIIS), Gwalior 2014, pp. 1-6.
- [2] Tang, X. Cao, A. Ma, "Towards Adaptive Framework of Keyword Driven Automation Testing", in Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 200S, pp. 1631 – 1636.
- [3] R. Gupta, N. 8ajpai, " A Keyword Driven Framework for Testing Web Applications," in International Journal of Advanced Computer Science and Applications (IJACSA), Vol 3, No. 3, 2012.
- [4] Disha Garg; Abhishek Singhalt and Abhay Bansa, "A Framework for Testing Web Applications Using Action Word Based Testing", 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015.
- [5] K. V. Aiya; Hemdutt Verma, "Keyword Driven Automated Testing Framework for Web Application", 2014 9th International Conference on Industrial and Information Systems (ICIIS), 2014.
- [6] Fei Wang; Wencai Du, "A Test Automation Framework Based on WEB", 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, 2012.