

Hybrid Test Automation Framework for Web Application

Bhagyashree Bhondokar¹, Pooja Ranawade², Snehal Jadhav³, Mayuri Vibhute⁴

Dept. of Computer Engineering
MAEER'S MIT College of Engineering, Pune
Savitribai Phule Pune University

Abstract—Testing is the main technique to ensure quality and find out bugs in the software development. Hence, it becomes necessary to have knowledge about testing. This paper gives an idea of test automation. The main focus of our paper is to provide information about Hybrid Test Automation Framework for web application. It will be useful to run the test cases given in the input file (an excel file containing keywords and data) and generate report automatically with minimal human intervention. Organization can significantly increase the accuracy and speed of the testing process by implementing the appropriate automation framework.

Keywords—Automation Framework, Test script, Selenium, Automation Testing, Hybrid framework

I. INTRODUCTION

Software testing is one of the fields that are growing rapidly in the past ten years. Till now, many approaches have been put forth but still there is a room for improvement. Software testing is defined as “an investigation conducted to provide stakeholders with information about the quality of the product or service under test” [4]. Therefore, the main goals of this activity are to detect and prevent defects as well as to insure the intended behavior of the tested software [5], [6].

A. Test Automation

One of the most interesting parts of testing is test Automation. It is a part of both testing and development team which requires programming concepts as well as testing strategies.

Need of test automation:

1. Saves time and money
2. Increases depth and scope of tests
3. Optimizes speed, efficiency and quality of tests

It all started with record and playback approach and further moving to data driven approach of automation and moving towards keyword hybrid automation.

B. Automation Framework

An automated test framework may be defined as a set of procedures, concepts and environment in which tests will be created and implemented [3]. There are a various types of test automation framework but the most common are:

- Data driven Framework
- Keyword Driven Framework
- Hybrid Framework.

The test data is stored in external files or database in case of data driven framework. Its biggest limitation is the

fact that every time similar kind of tests is executed. This can be used in application which requires testing with larger set of data over similar functionalities. A keyword driven framework continues the idea of data driven frameworks and along with data action set is also provided in external files [10]. In case of a hybrid framework, the basic concepts of data driven and keyword driven are combined.

In this paper, we presented information about Hybrid Framework.

II. TECHNOLOGY OVERVIEW

A. Java

It is an object oriented programming language and platform independent i.e. compiled code can run on any machine supporting java. It follows WORA way i.e. “Write Once, Run Anywhere”. Java was released in 1995 as a core part of Sun Microsystems' Java platform, developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation).

B. Selenium

Selenium [1] is an open source software testing tool used as interface while testing web applications. Providing a well-designed object-oriented API that gives improved support for testing problems of advanced web applications is a main aim of Web Driver [1]. By using each browser's native support for automation, it creates direct calls to particular browser. It supports different browsers like Mozilla, Firefox, Safari, IE etc. It is platform independent.

C. Eclipse Java EE IDE

It is an open source IDE. It helps programmers /testers write code much faster and easier way than using a text editor.

III. FRAMEWORK DESIGN

The framework contains some main components, they are as follows:

1. Reusable Library
2. Library Files
3. Test Data
4. Test Report
5. Test Screens AUT
6. Test Scripts

1. Reusable Library:

It consists of common functions that can be used again and again. Application independent part of framework is situated in this folder. We are trying to create our framework more generic so that 70% of part should be kept

in this folder. Special folder for application independent code makes our framework more reliable and useful for the various web applications. This part of framework will consist of various utilities which are common in many of the web application.

2. Library Files:

This consists of all the files required for test automation. It consists of various jar files supported by selenium. For ex: selenium-server.jar, selenium-java-client-driver.jar.

3. Test Data:

Test data is provided in the excel sheet as an input to our framework.

4. Test Report:

The framework will generate report for executed test cases. Log files will be generated with time stamp of execution of test cases and also their result Pass/Fail. It will also create email ready HTML report.

5. Test Screens:

This is nothing but AUT i.e. Application under Test, application on which our framework is working.

6. Test Scripts:

These are nothing but .java files which represents test cases of application under test.

The "Fig.1" shows the integration of various parts of Framework Structure.

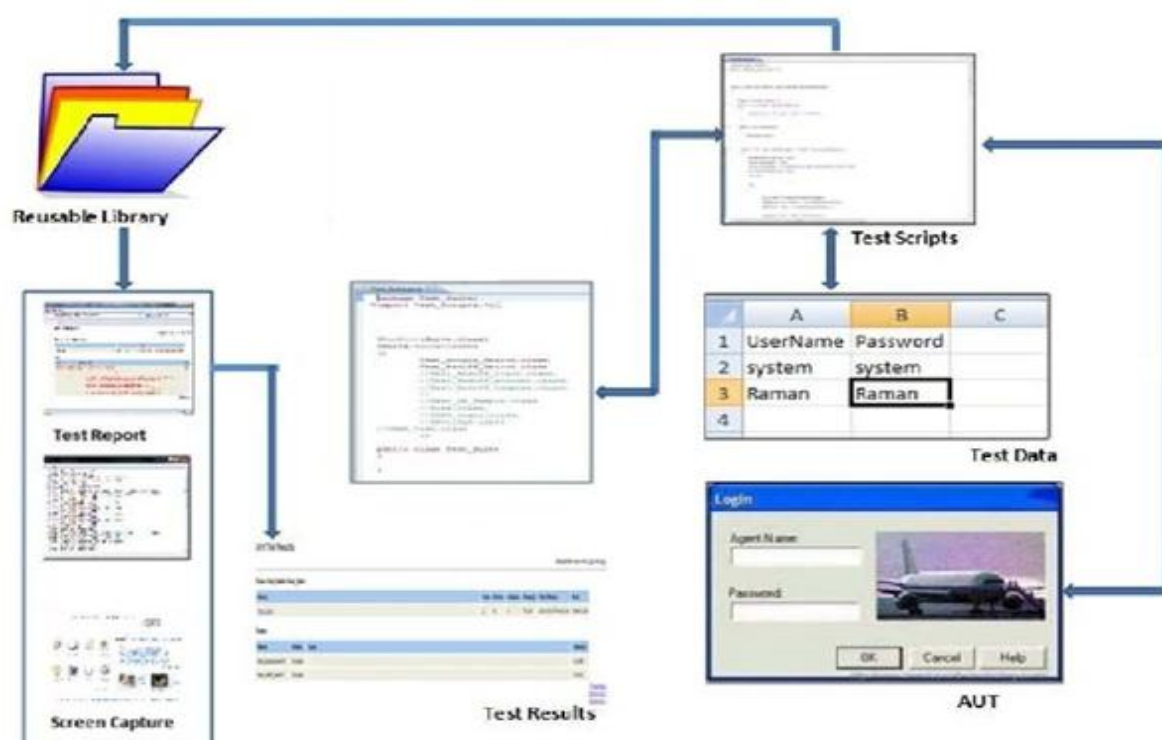


Fig.1. A Framework Component Structure

IV. FLOW OF FRAMEWORK

The step by step approach as shown in “Fig.2” can be understood as follows:

1. Execution of the test scripts start from the excel sheet input. Once you click on execute button, the Excel Reader file will go through the excel sheet to find out the actions and data set provided to form test cases that needs to be executed.
2. Once it finds the test cases, it loads the test case into TestExecution and asks to execute those test cases.

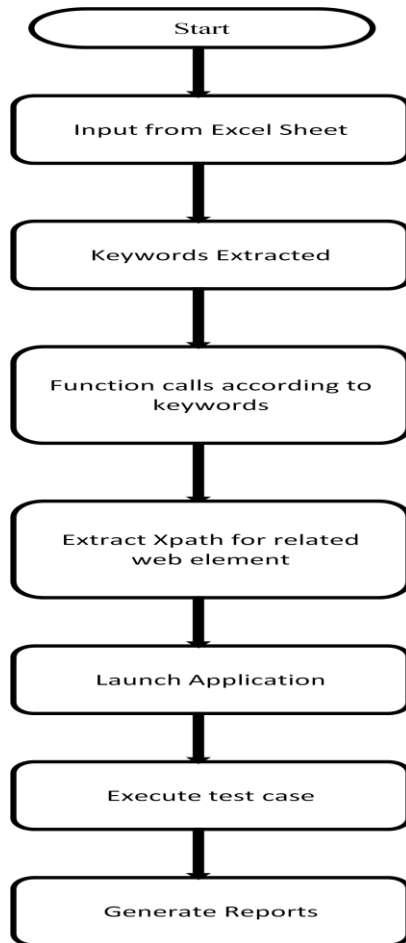


Fig.2. General Flow of Framework

3. TestExecution (with the help of function libraries, object repository and reusable libraries) executes the test case.
4. As soon as test cases are executed, control moves towards ReportGeneration Unit. Various screen shots are taken while executing test cases and report is generated.
5. Once all the test cases have been executed, it will create the test results and will store it into its corresponding folders.

V. IMPLEMENTATION

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

In Hybrid testing, we write keyword that corresponds to an individual action like a click, selection of a menu item or other actions. In Hybrid automation framework, we create the different methods in Java that are mapped to the functionality of the application under test.

Consider following example, we will automate it with Hybrid automation framework:

Test case 1: Launch Web Application and verify URL

Test case 2: To verify user is able to navigate to Login page.

To automate the test cases, we will write functions in java such as Read_Sheet.java to read keywords from sheet and Write_Sheet.java to write results back to sheet. Similarly we can create different methods for different functionality. Object repository will be created to store Xpaths of web elements. Two different sheets will be maintained as

Test case sheet (refer “Fig. 3”) and Data sheet (refer “Fig. 4”). These files will be provided as input to framework.

TestCase ID	Test Description	Execute	Browser	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5
TC_01	To verify that when correct URL is entered , application should be invoked	yes	Chrome	OpenBrowser	close			
TC_02	To verify that user is able to navigate to Login page	yes	Chrome	OpenBrowser	navigate_MyAccounts	Login_page	verify_URL	close

Fig.3. TestDriver.xlsx

Test case ID	Application URL	Login portal
TC_01	http://jahagir.getwebcare.in/	
TC_02	http://jahagir.getwebcare.in/	http://jahagir.getwebcare.in/index.php?route=account/login

Fig.4. Data.xlsx

After taking input, driver instance will be created using selenium Webdriver. This will help to create interface between framework and application under test.

Further main functions will be called based on the keywords used in TestDriver.xls. From those keywords, corresponding web elements are considered and their path

is extracted from object repository. After that common function containing appropriate actions for web element is called to execute test case.

At the end, report generation code is executed. Report in the form of log is generated shown in "Fig. 5". Also HTML report will be generated as shown in "Fig. 6".

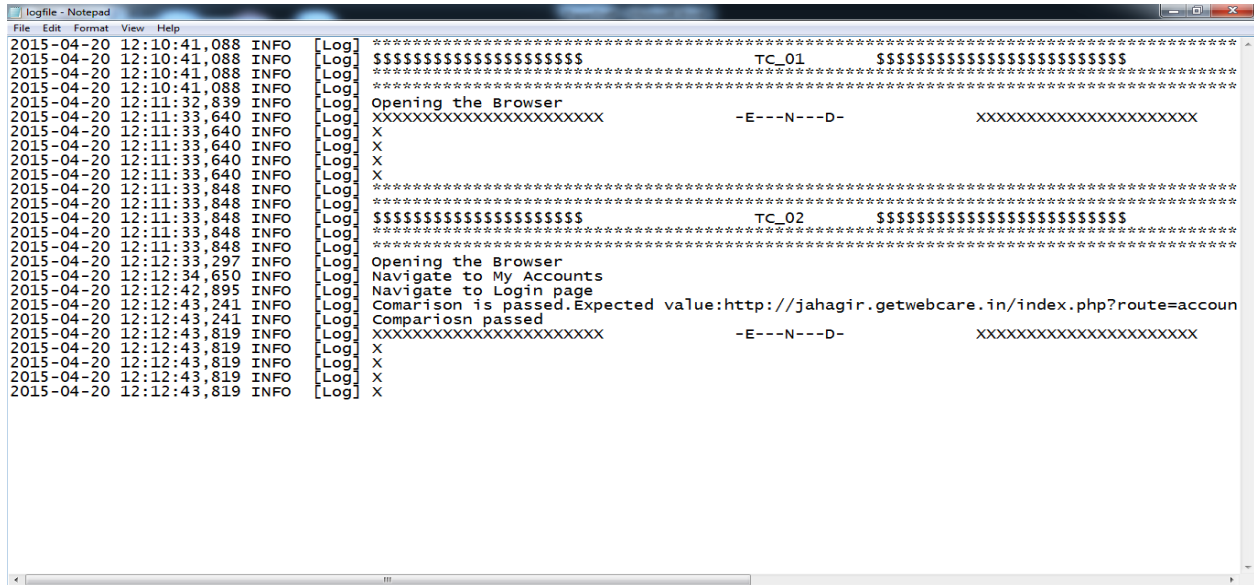


Fig. 5. logfile

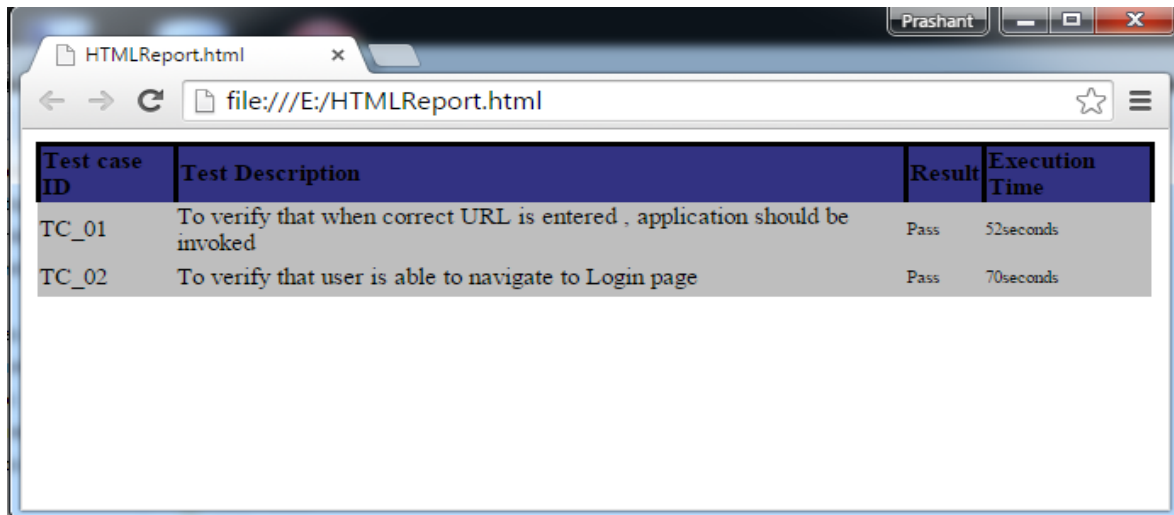


Fig.6. HTML Report

VI. FEATURES

1. It increases the importance and accuracy of testing and gives large test coverage.
2. It provides greater reliability as when developer writes test case and adds them in test suite there is no necessity to remember that test cases.
3. Compatible to different types of browsers.
4. Multiple test cases can run continuously producing corresponding results.
5. Operating System compatible can be used on any operating system e.g. Linux and Windows.
6. Requires less time to execute test cases as compared to manual testing [11].
7. Detailed log file is generated to show status of execution.
8. Easy to use even if technology (no need of particular language specific knowledge) is not known to user.
9. Although the initial effort for building the framework is high, in the long run, the return on investment will

be high because of the reusability and optimum utilization of the tool.

VII. CONCLUSION

The process of creating Automation framework requires detailed planning and extreme efforts. To achieve desirable features, one should design framework accurately. Such type of framework can then be used for any kind of application with minimum changes in any type of organization who requires automation testing. In this paper, we have presented idea of hybrid framework and its components. By using this Framework, instead of writing multiple functions separately, we have abstracted those things to excel sheet by giving keywords and test data as input which will help to test the entire functionality of Application under Test. This type of framework can be used widely in any type of web application for automation testing.

ACKNOWLEDGMENT

We would like to thank Prof. Mrs Reena Pagare (Department of Computer Engineering), MIT College of Engineering, Kothrud, Pune for her consistent and valuable guidance.

REFERENCES

- (1) "Selenium" "Selenium Web Driver" [Online] Available: <http://seleniumhq.org>
- (2) [Online] Available: www.softwaretestinghelp.com
- (3) [Online] Available: <http://www.automatedtestinginstitute.com>
- (4) A. M. Memon, M. E. Pollack, and M. L. Soffa, Using a goal-driven approach to generate test cases for GUIs. In ICSE '99: Proceedings of the 21st international conference on Software engineering, pages 257–266. IEEE Computer Society Press, 1999.
- (5) I. M. Iacob, R. Constantinescu, Testing: First step towards software quality - http://jaqm.ro/issues/volume-3,issue-3/pdfs/iacob_constantinescu.pdf
- (6) C. Kaner, J. Bach, and B. Pettichord, Les-sons Learned in Software Testing, John Wiley & Sons, 2002.
- (7) I. Ivan, C. Boja, Practica optimizarii aplicatiilor informatice, Editura ASE, Bucuresti 2007, 483 pg, ISBN 978-973-594-932-7
- (8) Suresh Thummalapenta, Saurabh Sinha, NimitSinghanian, and Satish Chandra "Automating Test Automation" † IBM Research – India , ICSE 2012, Zurich, Switzerland
- (9) FeiWang , Wencai Du, "A Test Automation Frame work Based on WEB" 2012 IEEE/ACIS 11th International Conference on Computer and Information Science
- (10) Jingfan Tang "Towards adaptive framework of keyword driven automation testing" Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008
- (11) Prof.(Dr.)V. N.Maurya, Er.Rajender Kumar "Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model" International Journal of Electronics and Electrical Engineering ISSN 2277-7040 Volume 2 Issue 1(January 2012)