Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 1, 226-232 (2020)

www.astesj.com

Special Issue on Multidisciplinary Sciences and Engineering

ASTESJ

ISSN: 2415-6698

Framework for Automation of Cloud-Application Testing using Selenium (FACTS)

Md Nurul Islam*, Syed Mohammad Khurshaid Quadri

Department of Computer Science, Jamia Millia Islamia (A Central University), 110025, India

ARTICLE INFO

Article history:
Received: 10 December, 2019
Accepted: 08 January, 2020
Online: 22 January, 2020

Keywords:
Test
Automation
Framework
Selenium

ABSTRACT

A framework is an amalgamation of integrated tools, libraries, utilities and its coordination to interact with the other automation components. The motive of designing a test automation framework is to implement uniform standards towards automation throughout the organization to achieve the desired outcomes. Test automation framework is required to maintain the standardization of the activities performed to manipulate the operations (addition, modification and deletion) on the test scripts and functions easily, to provide the scalability and reliability. Tools, process and testing team are the three key player of test automation. The proposed framework is based on the consideration of test environment (includes language binding, IDE, automation tools and testing framework). Selenium WebDriver is used as web application automation framework to execute test across multiple browsers that supports multiple operating system with variant programming language. Selection of WebDriver automation tool for this framework is due to their internal architecture to directly communicate with browser for fast execution. The combination of test automation tools (i.e. Java, Eclipse, Selenium and TestNG) makes JEST for designing test automation framework. Both Positive and negative test must be performed to verify functionalities of applications to handle unusual exceptions.

1. Introduction

Software testing is an evaluating as well as mitigating process, as it evaluates the functionalities of application to meet out the specific requirements of software product, at the same time it also identify the defects in product to mitigate the risk of software failure by find bugs and errors in the program.

"Framework for Automation of Cloud-application Testing using Selenium (FACTS)" is almost self-explanatory, though this paper may also explore the terminology explicitly. This paper may discuss about, why do we require test automation framework, how the software testing framework may organize the test automation, how do we select best appropriate software testing framework, what are the challenges being faced by the software automation team to design and deploy the test automation framework, what are the potential benefits of using software testing framework.

Why selenium for automation? Selenium is powerful open source tool for test automation framework independent of language binding & development platform that support functional testing across the browsers.

A framework is a combination of various forms of principles, guidelines, processes, concepts and tools etc. followed by testers to take advantage of it, to automate the application. Test automation framework decides the success or failure of any software automation project. Test automation is testing process of automating test with the help of automation tools to find out defects in the application. Test automation reduces test execution time and testing cost along with the less human interaction with the system. Test script has been used to execute test multiple times, when required. Test script must consider, what to automate or what not to be, like functionalities that are going to be change in near future or not, whether script is applicable for regression testing or not, whether test cases takes more time to execute/execute with errors or operated over huge amount of input data etc.

2. Test Automation Framework – An Understanding

Test automation framework is set of guidelines/rules followed while writing test script to generate test cases as per the

* Md. Nurul Islam, Department of Computer Science, Jamia Millia Islamia, New Delhi, India 110025, 9818933615 & mnislam@jmi.ac.in

requirement of test scenarios. A framework is an amalgamation of integrated tools, libraries, utilities and its coordination to interact with the other automation components to execute the test scripts and to capture the outcomes in systematic and beneficial manner to test a specific product. If a testing activity automated without having a test automation framework, it is obvious to raise the question that "how reliable the method would be? And how quickly they can produce the same output?"

The motive of designing a test automation framework is to implement uniform standards/policies towards automation throughout the organization to achieve the desired result. A test automation framework can be considered effective if the result of test automation is easily understandable to end user, otherwise executing high volume of test cases are of little or no use for end user. The automated test generated report must include the point where the application gets failure along with test data being used for automation.

2.1. Why do we require test automation framework?

Test automation framework is required to maintain the standardization of the activities performed to manipulate the operations (addition, modification and deletion) on the test scripts and functions easily, with the aim to provide the scalability and reliability. If a tester automate testing activities without considering a framework, it is obvious to ask how reliable the method would be. Can the tester re-generate the same result? An automation framework sort out all such problems by providing a standard guideline to leverage reusable components to attain the desired & reliable results rapidly. There are various factors for which an automation framework required [1].

- The Ease of use (Reusability of test script) to maintain the test script.
- To maintain reliability and consistency follow specific guidelines.
- To maintain scalability with changing requirement.
- To improve the efficiency of testing activities
- To provide easily understandable reports of automated results.

2.2. How do we select best appropriate test automation tool?

Selecting best appropriate tool for automation is one of the basic tasks for organization. The impact of selecting best automation tool may greatly affect the automation process which may provide sophistication in automation. While selecting tools for automation, certain conditions must keep in mind which makes sense for automation, like how frequently the repetitive test may occur, how often the regression testing is required etc. There are certain criteria to evaluate and select the automation tools [2] as mentioned below.

- Whether the tools fulfill the testing requirements. Is it compatible with project environment & technology that support objects in application?
- Which testing level and types does the tools support like (system, functional, performance) testing. Select tools which support maximum testing type with the ability to automate complex requirements.
- Whether the tools support all major browsers/OS or portable devices in which the application runs?
- Whether the estimated budget for purchasing automation tool affordable for organization?
- Whether the organization has skilled resources that can understand the language the tools support?
- Does the tool have powerful mechanism for graphical reporting?

While selecting a tool comprehensive understanding of its integration, reporting and ease of use with other automation tool must be executed. The standard automation testing framework should be application & tool independent & loosely coupled, it means that if the organization decides to migrate from one automation tool to other, it need not required to create a new test automation framework from scratch.

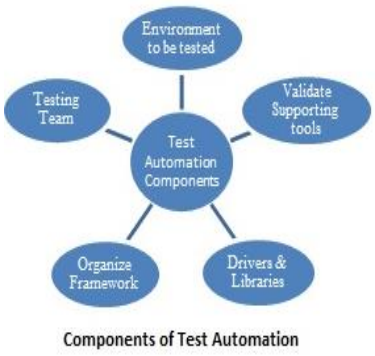

Figure 1: Components of Test Automation Framework

2.3. Components of test automation framework [3]

1) Environment to be tested: While designing a test automation framework all the test related tools, equipment, scripts and procedures must be verified and validated to ensure the system is stable to perform the desired task. This would be the first step towards the environment setup for the test automation. Until the test environment is not supposed to work accordingly the test automation may not be cost effective. Thus all the coordinating tools that support the environment must be stable and work together smoothly as a whole prior to start the test automation.

2) Validate supporting tools: While writing test scripts various supporting tools are required to organize the test. All such type of tools are properly investigated and independently monitored for expected contributory result as a unit, like test automation tool, bandwidth monitoring tools used for traffic management, other supporting tools such as requirement analysis, configuration management tools, test factory, defect tracking. A traffic monitoring and controlling activity of network is important to improve performance, efficiency and security of network resources. Network accommodates different kind of data traffic, identification of such traffic or congestion may lead the network administrator to optimize the network. Various types of network traffics are [4].

Table 1: Different Forms of Network Traffic

Traffic Type	Problem	Solution	Example
Busty Traffic	Consume high bandwidth starve applications	Fix constraint to limited bandwidth access	Download of FTP, Video Content
Latency Sensitive Traffic	Susceptible to compete for bandwidth results in poor response time	Fix min. & max. bandwidth range based on priority	Streaming application, VOIP, Video Conferencing
Interactive Traffic	Susceptible to compete for bandwidth results in poor response time	Prioritize over less essential traffic	SSL transactions, IM, Telnet Session
Non-Real Time Traffic	Consumes bandwidth during business hours	Schedule bandwidth during non-business hours	Email, batch processing applications

Configuration management tools are used to implement the changes (add, remove and update) occur during the test automation framework design to all associated system by implementing and enforcing to adopt such possible change. Test factory is an extensive use of test automation that minimizes the cost of testing and error rate along with supporting rapid scalability. Defect tracking is tracking the logged defects from beginning to end by testing, inspection or from customer feedback in a product with the aim to provide latest version of the product to fix the defect.

3) Drivers & Library: Different browsers require their own driver to communicate and control with the WebDriver. Some of the browsers are OS dependent but Firefox and Chrome are easily available across the platform. Download and unpack the drivers and add the location to you system PATH variable. The browser driver interacts with the respective browser by establishing a secure connection. While a test script is executed with WebDriver environment various activities performed in the background like:

- For every selenium command an HTTP request is generated to communicate with the browser driver.
- Drivers receive the HTTP request trough HTTP Server.
- HTTP Server decides the execution of instruction on the browser.
- As HTTP Server receives the execution status, it automates the script.

Browser automation library has been used to interact with other task required for automation. Library works as basic building block to establish a successful test environment to execute the test script by providing the required jar files, libraries, drivers and other supportive files.

Steps to add the Selenium Library to the project:

- Download & extract current release of Java Selenium library.
- Open the Eclipse IDE, go to the "Project→Properties"
- New dialog box open, Click on " Java Build Path" in the left pan, the to the register "Libraries"
- Click on "Add External JARs" in the right pan.
- Move to selenium library downloaded folder to select all .jar files by clicking "Open" button.
- Repeat this for all .jar files in the sub folder libs as well.

Once the library is configured with the project it can be reusable with the different tests, such utilities can perform different tasks for different test cases.

4) Organize Framework: Selenium is automated testing framework for web applications. To organize the framework following major steps can be taken into consideration [5].

- Select Programming Language: Select appropriate programming language as per the need of application under test and the developer/tester going to use the framework to write tests?
- Select Unit Test Framework: To design a framework a unit test framework is required. TestNG Compatibility with JAVA is highly recommended as it eliminates the limitation of earlier framework with the ability to write more flexible and more powerful tests. Other features TestNG are: annotation, grouping, sequencing and parameterization etc.
- Design the Framework Architecture: The architecture of framework is designed in such a way that it can be easily maintainable, scalable and sustainable in nature.
- Build the SeleniumCore component: SeleniumCore is designed to communicate between the web elements and browsers instances, it is used to create and destroy WebDriver objects, since it is not being taken into consideration by test writers.
- Build SeleniumTest Component: SeleniumTest components are all those test cases going use the classes provided by SeleniumCore. The design pattern used is Page Object Model (POM). It means for every page in the web application their corresponding a Page Object for them to organize the UI element in the pages.
- Select a Reporting Mechanism: Unless we get useful insights from the test result to take meaningful corrective action, test automation is of no use. A good report must include details about number of pass or failed test cases, passing rate, execution time and reason why test cases failed.

5) Testing Team: Successful test automation depends on efficient testing tools, standard testing process and skilled testing team to execute assigned responsibilities. Tools, process and testing team are the three key player of test automation. Software testing team is mainly engaged for test automation to utilize dedicated automation testing resources. The test team ensures that

the success of testing and automation effort based on the roles, duties and skill they provide.

- ✓ Manual Tester – Design and develop manual scenarios from requirement analysis.
- ✓ Automation Tester – Design and develop automation script.
- ✓ Lead Automator- Design and develop project automation architecture. Provide training and guidelines to testing team and collaborate to achieve the testing goal.
- ✓ Automation Environment Expert- Design, configure and update test environment and automation software.

Apart from the above role and responsibility testing team manages test platform, test tools, test cases and also provides tutorial to the users.

2.4. Characteristics of test automation framework

- If test automation framework is independent of Operating System, Programming language, tools and browsers, it would be considered as standard framework.
- The framework must be easily understandable without having programming skills.
- The framework must be easily maintainable and extendable.
- The Framework must execute test cases automatically, which includes test scenarios, test data, report generation, error handling and reporting final result.
- The framework must be able to execute tests without human intervention.
- Every test executed, must assign status of either Pass or Fail, failed test cases must include the failure data, the point of failure and other short description for failure.
- The framework should have meaningful logging and reporting structure.

2.5. Types of automation framework [6]

1) Linear Scripting Framework - This framework is also known as 'Record and Playback' framework as it generates the test scripts based on simulated record and playback method. It is not useful to run with multiple data set i.e. lack of reusability. This framework is suitable for small size applications. It works for individual test cases because it records the test script for particular scenarios.

2) Modular Testing Framework - In this framework, the complete application is divided into number of smaller modules and test script is generated individually. Finally the results of individual modules are compiled together by the master script to achieve the required target.

3) Keyword Driven Testing Framework – This framework is based on the keywords or actions for executing functions or methods. The test script is totally keyword/action dependent stored in tabular format in excel sheet, so it is also known as table-driven testing. The main class maintains the keywords and their corresponding action in the external table.

4) Data Driven Framework – Data Driven Test Automation framework is based on bifurcating the test data from test script. Test data is stored in separate files/format, test scripts connect and execute with different sets of test data. Due to having variant combination of test data set in separate table, this framework covers more test cases with flexibility in execution of test by changing input test data. This framework requires the programming skill to develop the test script.

5) Hybrid Driven Testing Framework – This framework is combination of two or more framework to leverage the strengths and benefits of each of the framework individually and to produce a better test environment overall.

6) Behavior Driven Development Testing Framework – The Purpose of using this framework is to share the process and tools and to communicate amongst stakeholders, with the aim to deliver quality product. BDD test cases are being written in simple language that can be easily understandable, which contains the details of how application behaves? BDD provides the platform for both technical and non-technical team to work together. A test case written in BDD follows the format 'Given-When-Then'. Given a certain scenarios, When an action takes place, Then this should be the outcome.

3. Related Works

Selenium WebDriver is powerful tool for test automation that directly interacts with web browsers, it follows the page object pattern to access the web page to reuse the test script to write the test cases. The page object pattern is used to reduce the coupling between web pages and test cases, which improves test suite maintainability [7]. UI locators are used to locate the web elements (Text box, button, checkbox, etc.) on the web page. Different identifiers are used to locate the web elements like (ID, link, XPath, link Test etc.), ID Locators have better result than XPath Locator, Combination of ID locator or XPath Locator with Link Text have improved the result but still not conclusive [7].

WebDriver tool can be used to write test cases easily and efficiently, it can helpful to develop and analyze the test cases using screenshot property, at the same time it is easily maintainable, reusable and extendable with the new requirement. Since the selenium WebDriver does not support any built in mechanism to generate test report, for that purpose TestNG is used as a testing framework. TestNG is compatible to perform unit, functional and integration testing. The author has generated a customized test report to analyze the failure test cases using screenshot [8].

WebDriver is cross platform testing framework that provides dynamic characteristics of web page, it means page reload is not required, if page elements gets changed. Selenium WebDriver execute faster as it directly interacts, run and control browsers using own control engine [9]. Selenium WebDriver is not equipped with reporting mechanism, TestNG generate report on behalf of selenium result. Such report contains execution time, method and detailed status of test script of running test suite "Passed or Failed" [10].

Table 2: Comparisons of Automation tools

Products & Parameters	Selenium	Katalon Studio	Test Complete	QTP/ UFT	Soap UI
Ease of Configuration & use	Advance skill required for configure & execution	Easy to configure & execute	Easy to configure. Training required for tool operation	Expertise require for configuration. Training required for tool operation	Easy to configure & execute
Licences required	Open Source & free to use, Free of Charge	No licence required. Not open source but free	Commercial tool, Licence required	Commercial, It is HP licenced product avail as single/Concurrent user	SoapUI is an open source. SoapUI Pro is the commercial.
Service support	Expert or group professional support	User friendly interface	Community Volunteer support	HP professional support	SoapUI pro Customer get support from forum or wen form
Operating System Support	Ms Windows, Apple OS X, Linux, Solaris	Ms Windows7/8/10, Apple OS X, Linux	MS Windows10/8.1/8/ 7/ vista, Server16/12/8	Only Windows Windows7/8/8.1/ XP/Vista/ 2003/2008	Ms Windows, Apple OS X, Linux
Scripting Language	Java, C#, Perl, PHP, Python, R, Ruby, JavaScript	Groovy, Java	JavaScript, C#, Python, VBScript, JScript	VB Script, Java Script, Window Shell Script	Groovy
Browser Support	Firefox, IE, Chrome, Safari, Opera, Phantoms, HtmlUnit, Android, iOS	IE9/10/11,Firefox, Chrome, Opera, Safari, MS Edge, Remote, Headless Browser	MS Edge,IE10/11, Chrome, Firefox	IE(V6-V11), Firefox(V3-V31),Chrome(V12-35),Safari6/6.1/7	SoapUI Default Browser
Device/Technology Support	Web apps	Web/API/Mobile/ Desktop apps	Desktop, Web & Mobile apps	Web/API/Mobile/ Desktop apps	API/Web apps
Testing Framework	RSpec (Ruby), Test::Unit (Ruby), Unittest (Python 2), JUnit 4 (Java), TestNG (Java), NUnit (C#)	Katalium Framework with TestNG	Application's UI	Build-in Feature for framework, like Key-word driven, Data Driven, Hybrid	Application's GUI
Type of test Performed	Functional Testing	Cross Browser Testing	Unit Testing, Functional Testing	Functional, Regression & Service Testing	Functional. Load & compliance Testing
Initially launched	2004	2015	1999	1998	2005

Comparisons of automation tools [11]

4. Framework Design for Automation of Cloud-Application Testing Using Selenium

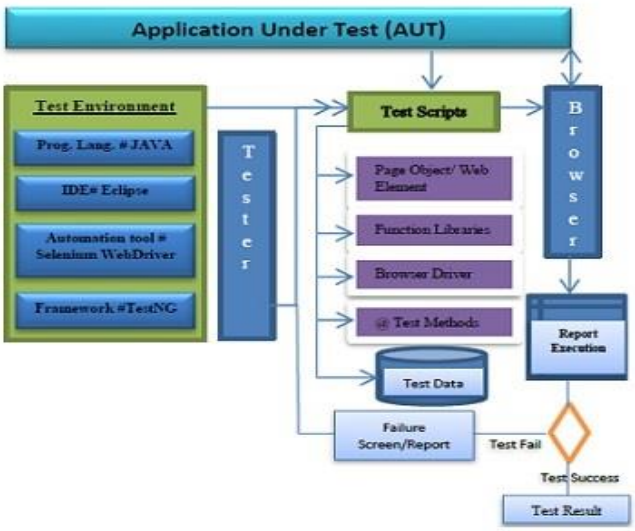

Figure 2: Architecture of Proposed Framework (FACTS)

5. Proposed Work

The framework (FACTS), we have designed is based on the consideration of test environment (includes language binding, IDE, automation tools and testing framework), types of Application under Test (AUT) with the compatibility of web browsers. The architecture of the proposed framework (Figure 2) includes following tools, functions, method and libraries etc.

- Programming Language # Java
- IDE #Eclipse
- Automation Tool # Selenium WebDriver
- Framework # TestNG
- Test Script
- Page Object /Web Element
- Function Libraries
- Browser Driver
- @Test Methods
- Test Data

The proposed framework may utilize the following tools, functions, method and libraries for designing automation framework:

- *Programming Language #Java*: Why Java is selected for this test automation framework? Java is a programming language which supports multiple operating systems. It is platform independent language, support Object Oriented features. Other important characteristics are simple, robust, Secure, Multithreading, distributed, portable to use and free of cost availability, open source and licence free.
- *IDE #Eclipse*: Eclipse is an Integrated Development Environment (IDE) tool most widely used to develop applications in Java and other programming languages

like C, C++, Perl, PHP, Python, Groovy, Ruby, R etc. Eclipse development environment is customizable as per the workspace and plug-in required. It is free to download and install.

- *Automation Tool #Selenium WebDriver*: Selenium is widely accepted open source test automation tool to test web-applications. Selenium WebDriver API is easy to explore and understand which support tester to easily read and maintain the test. Selenium WebDriver is web application automation framework that execute test across multiple browser, supports multiple operating system, support different programming language also. WebDriver executes faster due to their internal architecture to directly communicate with browser. When we execute a selenium command, an HTTP request created to communicate with particular browser driver, the browser driver then uses HTTP services to get HTTP requests, to establish the selenium command. Since WebDriver does not provide any build in facility to automatically generate Test result, we have incorporated TestNG as test reporting tool.
- *Framework #TestNG*: TestNG evolved as a testing framework to overcome the shortcomings of Junit and NUnit. It is an open source used for test automation framework. Where NG stand for Next Generation. According to some survey 76% tester uses Java programming along with TestNG framework. TestNG also provides an option to execute failed test cases only, it creates a file named testing-failed.xml for failed test cases in output folder. If the tester is interested to see only failed test cases they can execute this .xml file. There are various features due to which TestNG is in demand:
    - TestNG covers a broad range of test at different levels started from unit testing to system testing.
    - It supports annotations to create test cases easily.
    - It introduces 'test groups' to group test cases with prioritization, if required.
    - It supports parameterization and parallel test execution, it also execute multiple programs/classes using .xml.

The combination of test automation tools used for proposed framework has gained immense popularity among test automation tools as language, IDE and framework. The combination of the tools (i.e. **J**ava, **E**clipse, **S**elenium and **T**estNG) makes JEST for design and develop test automation framework.

- *Test Script*: Test script is set of instructions execute to test the system functions on the application under test for expected results. Test script mainly includes detailed descriptions of actions with required data to perform test. Test script follows the sequence to explore how to use the program and the order in which it is being utilized/executed, by pressing which button to carry out particular action in the program along with the stepwise expected results to observer the change in UI. Test script has a drawback with project which changes their requirement frequently, so the testers have to continuously update the script to fulfill the requirements. Since the test script is designed to test a particular task frequently, if bugs occur in the code the result would be same every time until the tester change the conditions.
  Page Object /WebElement: A Page object is an object oriented class through which AUT interface the web page. It plays a key role to identify the element on the web page, at the same time it reduces the code duplication by enhancing the mechanism of test maintenance. Whenever a test needs to communicate with UI of the page they use only methods of page object class. In case the UI of the page gets changed, only test script within the page object needs to be change. All possible user interactions can be used as a method on the class, ex. *clickLoginButton ();*
  setCredentials (user_name, Password);
  Selenium encapsulates elements on the page as an object of WebElement. WebDriver uses various techniques to identify the element on the webpage based on the its properties like ID, Name, Class, XPath, TagName, CSS Selectors, Link Text etc. WebDriver uses method like findElement() to find single web element and returns WebElement object. The method findElement () uses various locators like By.id (), By.name (), By.XPath () etc. to locate the element on the page by its properties.
  Ex. WebElementemail_id=driver.findElement (By.id ("User_Name")); // to get the web element for email address text field.
  Selenium WebDriver uses *findElement(By.Locator())* method to find the element on the page. Here findElement method uses locator/object as 'By' identifier.
- *Function Libraries*: One of the major purposes of designing a framework is its reusability, if any functionality is used frequently, such function can be written at one place and called multiple times for other test case. It is known as generic function. User defined functions are created based on their own requirement used with help of calling the appropriate package file. Function libraries are used to maintain the logic which is being used to write automation test scripts and called with their name. Methods are also known as functions.
- *Browser Driver*: Selenium WebDriver is web Automation tools that help to run the test across multiple browsers like FireFox, IE or Chrome. Corresponding browser driver is required to run the test in particular browser. Selenium launches corresponding browser as

per the script to execute test steps. To initialize WebDriver we can write code as :

WebDriver driver = new FireFoxDriver(); // where WebDriver works as interface and driver is reference variable for interface. Object can be associated with the instance of class FireFoxDriver//

When we execute a selenium command (driver.getTitle();) an HTTP request will be created to communicate with particular browser driver, the browser driver then uses HTTP service to get HTTP requests to establish the selenium command. All the built-in methods of corresponding drivers can only be accessible after configuring the driver.

- *@Test Methods*: Why do we use @ annotation before writing test method. Annotations in selenium are used to bind the associated method going to execute. Test annotation @ always used before every method, if this test annotation '@' is not prefixed before the test method, that method would not be considered as part of test code to be executed. @Test indicates that the annotated method is a part of test case serves as the main method [12]. Some other annotations used with TestNG are:

  @BeforeSuite: This method will execute once before all tests in this suite have run.

  @BeforeTest: This method will execute before any test method belonging to the classes inside the <test> tag is run.

  @BeforeClass: This method will execute only once before the first test method in the current class is invoked.

  @BeforeMethod: This method will execute before each test method.

  @AfterMethod: This method will execute after each test method.

  @AfterClass: This method will execute only once after all the test methods in the current class have executed.

  @AfterTest: This method will execute after all the test methods belonging to the classes inside the <test> tag have executed.

  @AfterSuite: This method will execute only once after all tests in this suite have executed.

- *Test Data*: Test data is actual input used to test the application. Test data would be used for positive or negative testing. Positive data is used to verify the functionality of application to meet the expected results. Negative data is used to test the ability of program to handle unusual, exceptional or unexpected input with its corresponding outcome [13].

6. Conclusion

In-depth planning, process and efforts are expected while design and develop a test automation framework. Selecting right automation tools plays a pivotal role to accomplish the automation task. Selenium WebDriver has been used to test web application end-to-end. WebDriver is flexible to integrate with variant development environment, tools (test, reporting and automation), OS, browsers and methods to design a test automation framework. Various types of test automation frameworks are available to design and develop test automation. The proposed framework (FACTS) is designed to work with Data Driven Framework as well as Hybrid Driven Testing Framework. A Standard automation tool is expected to execute its integration with other automation tools and reporting mechanism. Positive and negative test must be performed to verify the functionalities of application and to check the ability of program to handle unusual exceptions respectively.

References

[1] "A Guide To Automation Frameworks |Smartsheet" . Smartsheet, 2019, https://www.Swmartsheet.com/test-automation -frameworks-software.

[2] K. Kunal, and Mohammad Saad. "Test Automation Tool Selection Criteria And Comparison Matrix (A Complete Guide)". Softwaretestinghelp.Com, 2019, https://www. Softwaretestinghelp.com/automation-testing-tutorial-4/.

[3] Genius, Software. "Six Major Components of A Test Automation Framework - Software Testing Genius". Software Testing Genius, 2019, https://www.softwaretestinggenius. com/six-major-components-of-a-test-automation-framework/.

[4] "Network Traffic Management | Ipv6.Com". Ipv6.Com, 2019, https://www.ipv6.com/applications/network-traffic-management/.

[5] Logigear.Com, 2019, https://www.logigear.com/blog/test-automation/ building-a-selenium-framework-from-a-to-z/.

[6] "Types Of Test Automation Frameworks | Software Testing Material". Software Testing Material, 2019, https://www. softwaretestingmaterial.com/types-test-automation-frameworks/.

[7] Leotta, Maurizio et al. "Comparing The Maintainability Of Selenium Webdriver Test Suites Employing Different Locators: A Case Study". ACM, 2013, doi:http://dx.doi.org/10.1145/2489280.2489284.

[8] Gojare, Satish et al. "Analysis And Design Of Selenium Webdriver Automation Testing Framework". Procedia Computer Science, Vol 50, 2015, pp. 341-346. Elsevier BV, doi:10.1016/j.procs.2015.04.038.

[9] K, Revathi, and Janani V. "Selenium Test Automation Framework in On-Line Based Application -IJARSE". International Journal of Advance Research In Science And Engineering, Vol. No.4, no. Special Issue (02) ISSN-2319-8354(E), 2015, pp. 55-63.

[10] Singla, Sherry, and Harpreet Kaur. "Selenium Keyword Driven Automation Testing Framework ". International Journal Of Advanced Research In Computer Science And Software Engineering, Volume 4, Issue 6, 2014, pp. 125-129.

[11] Nurul, Md., and S.M.K. Quadri. "Software Testing Approach for Cloud Applications (STACA) – Methodology, Techniques & Tools." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2019, 10.1109/confluence.2019.8776915.

[12] "Selenium WebDriver Tutorial | TestNG For Test Case Management And Report Generation | Edureka". Edureka, 2019, https://www.edureka.co/ blog/selenium-webdriver-tutorial.

[13] "Test Data Generation: What Is, How To, Example, Tools". Guru99.Com,2019,https://www.guru99.com/ software -testing-test-data.htm