International Journal of
Science and Engineering Investigations vol. 7, issue 80, September 2018

ISSN: 2251-8843

Social Media Testing with Selenium

Festim Halili[1], Lirie Koraqi[2]
[1]Ph.D. in Computer Science
[2]Student in Master Degree
([1]Festim.halili@unite.edu.mk, [2]Liriekoraqi@gmail.com)

Abstract- Nowadays, software applications are being more complex and plaited with a great number of different platforms, that's the reason why it needs to be tested, without the testing process, there is no security that application will behave as it should, and the public results of not applying testing techniques couldn't be very harmful, leading to the fail of the application. Therefore, it is really important to have a specific test methodology to secure that the development software products to be assure according to the specific demands, this paper aim to present Facebook test with selenium, for reducing the errors (bugs) and the maintenance of software products.

Keywords: *Selenium, Bugs, Quality Assurance, UAT, IDE*

I. INTRODUCTION

Since Facebook is a very popular social network which has millions of users and every day is adding new utilities and functionalities to the platform, including the mobile versions enabling users with faster communication, better-quality posts, and creates to them a great space, where they can explore while they are logged in to the app. While adding new images to the web application, the need to test these images is necessary in order to be more acceptable to the user. The less errors and defects, the easier it will be the application in use. For this reason, when there is a good test team that perform code and automate a web or mobile app, it will definitely be more reliable, but of course knowing that these changes are executed by the code developers, that are able and have possibilities to implement applications. This paper work will show how tests are created to automate a web application, which in the case of study we have received Facebook's web application. For this case the study, as the main tool that enables automated testing, we have taken Selenium. For automated testing, we need to have Visual Studio installed (in our case we worked with Visual Studio 2015 Enterprise, but there are other versions of it).

II. WHY SELENIUM?

Selenium is an open source and a transmitted tool for automated software testing that enables testing of web applications. This tool can operate in different browsers and in different operating systems. [https://www.edureka.co/blog/what-is-selenium/]

There are many different tools that enable testing web applications, but some of them can't handle the complexity of a page that contains JavaScript. There are programs that just stimulate Firefox and Internet Explorer browsers to run JavaScript and this implies that they do not work as well as in real browsers. Selenium is the tool that covers these problems because it is based on JavaScript and runs tests directly in the browser. Selenium is executed within browser in JavaScript and controls it by giving commands. Tools that are open-source are designed so that programmers can share their code with the community and are usually ready for use without need to have their license and can be downloaded without cost. Nowadays, open-source tools are becoming more popular because of their integration and interaction with a debugging and rapid development. There are many tools used to perform functional testing, regression, and performance like Selenium, SOAP, UI, Open STA, Robotium, WebDriver, WebInject, Arbiter, Jmeter, Junit, NUnit etc.

From these tools, Selenium is considered to be the most used tool and is called a software testing port and its one of the best tools for use and the cheapest price. Tests are written in HTML or encrypted in different programming languages and can be executed directly in more modern browsers. Selenium is also used for UAC (User Acceptance Test). Among the many tools offered for different testing, Selenium has enabled market companies to have as much as possible functional products and to maintain them. It enables automated testing for applications that are developed for web, mobile, and desktop. The good of Selenium is that it is open-source, helps a tester to develop skills in learning a new programming language, and can know how to find the elements of a page. Selenium opens the test methods by relying on the browser you downloaded and calling it in the code part. However, new tools that are being developed for automated testing are enabling us to test the test methods in many browsers, which this Selenium does not support.

III. WHAT ARE THE REQUIREMENTS FOR USING SELENIUM?

Since Selenium is the most widely used tool so far, it has become one of the most searched tools online, as one of the software based on browsers and clay development. During regression testing, Selenium scripts can be used again. However, if the app continues to change, many of the selenium scripts can be destroyed. This can lead to confusion of test results that the QA team can't be sure if the cause was from an application bug or bug in the test script. For frequently-changing applications, script maintenance becomes a burden. In this case, since Selenium is only focused on web applications, testers can add to their testing with any other tool to perform end-to-end testing.

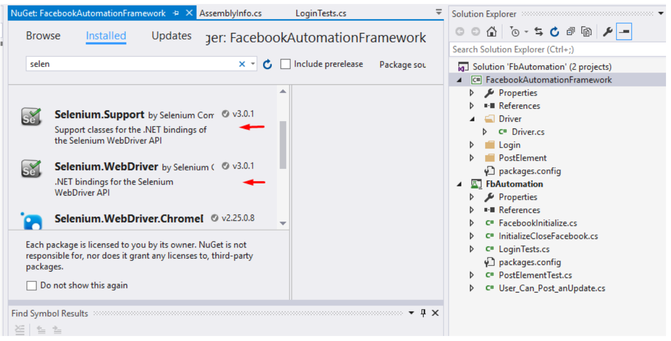
There has been a major improvement in Selenium features which is focused in simple to use, with a skilled GUI and can meet 90% or more of the testing application needs. Selenium ranks in the top of the software market with the following features:

- Simple and powerful documented which is oriented in modeling (DOM – Dosument object model), can be used for continuous integration with clay projects.
- It has extensive and flexibility along with its browser integration, unrivaled with other available tools.
- Supports browsers like: Internet Explorer, Firefox, Safari, Opera for Windows, Mac OS X and Linux
- Use programming languages oriented in objects such as Java, .Net, Ruby, Perl, PHP, etc.
- Provides options that support IDE (Integrated Development Environment) such as Eclipse, Netbeans, and Visual Studio etc. FRF, but always based on the language choice in which it is developed.

You can cover all types of testing using Selenium.

IV. How to write a class to automate a certain part of the Facebook app?

For automated testing, we need to have Visual Studio installed, a new project is launched to start automated testing. After we click the new project option, we open a dialog where we select the programming language with which we will write the code part and in our case we will work with the C # programming language. In the middle part of the dialog we choose the Class Library option; then, in order to divide the part of the code in two parts, we have: *Facebook Automation Framework*: Where is the whole code with the elements of the framework as well as the steps that has the test which is part of the test suite. Contains project classes that when we click on them, they perform their framework functionality. *FbAutomation:* Here is a test-suite for various parts of the Facebook page. Suites for different parts have their own name, and there are a few different classes that contain browser initialization, login and browser closure.



Selenium.Support by Selenium Com v3.0.1
Support classes for the .NET bindings of the Selenium WebDriver API

Selenium.WebDriver by Selenium (v3.0.1
.NET bindings for the Selenium WebDriver API

Selenium.WebDriver.ChromeI v2.25.0.8

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

Figure 1. Work area of a project

On the right side of the working window, at *Facebook Automation Framework* project click on with the right key and choose the "Add" option and then choose "Class". Type the name of the class and click on the "Add" button. After that, opens the working window for the class we added. Consider the creation of the "Driver" class. There is no rule on how to name classes, it is important to be understandable when calling to another class. The "Driver" class is written as follows:

```
using OpenQA.Selenium;
using OpenQA.Selenium.Firefox;
using System;
using System.Threading;
using System.Windows.Forms;
using FacebookAutomationFramework;
using OpenQA.Selenium.Chrome;
namespace FacebookAutomationFramework
{
public class Driver
{
public static IWebDriver Instance { get; set; }

public static void Initialize()
{ //WebDriver alows to open window of the browser
when we want to automatize in different browsers.
Instance = new FirefoxDriver(); // Alows to open
FireFox
//Instance = new ChromeDriver();
Instance.Manage().Window.Maximize(); //Maximize
the browser window.
}
//The method that alows us every time  we want to click the
enter button, we call this method
public static void PressEnter(int n)
{
for (int i = 1; i < n; i++)
{
SendKeys.SendWait(@"{ENTER}");
}
}
// Exit from browser window
public static void Close()
{
```

```
Instance.Dispose();
}
// It intends to wait until you execute the specified step
public static void Wait(int n)
{
Instance.Manage().Timeouts().ImplicitlyWait(TimeSpan.From
Seconds(n));
}
// Expect explicitly until you execute the specified step.
public static void ExplicitWait(int n)
{
Thread.Sleep(1000 * n);
}
// The basic address we will test - Facebook's address.
public static string BaseAddress
{
get { return "https://www.facebook.com"; }
}
// Method when uploading photos.
public static void UploadPhotoFile()
{
SendKeys.SendWait(@"C:\Users\QA
Lirie\Desktop\Foto per testim\Laptop.jpg");
Driver.Wait(2);
}
// The method when it comes to uploading videos
public static void UploadVideo()
{
SendKeys.SendWait(@"C:\AutomationContentFiles\minions.
mp4");
Driver.Wait(2);
}
// Method created to click the Enter button.
public static void PressEnter()
{
SendKeys.SendWait(@"{ENTER}");
Wait(2);
SendKeys.SendWait(@"{RIGHT}");
Wait(1);
SendKeys.SendWait(@"{ENTER}");
Wait(2);
}
}
}
```

In the rest of the project, in the test section, we create the *Facebook Initialize* class, where through the Driver class and the Initialize method, makes it possible to open the browser, send the basic address - Facebook address and at the moment its page opens, it sents data for user email, password as well as click on the Login button.

```
using FacebookAutomationFramework;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using OpenQA.Selenium.Chrome;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using OpenQA.Selenium;
namespace FacebookAutomationTests
{
public class FacebookInitialize
{
[TestInitialize]
public void Init()
{
Driver.Initialize();
Driver.Instance.Navigate().GoToUrl(Driver.BaseAddress);
Driver.Wait(5);
// Finds the position of the element - the position where the
user's email is marked and clicks on it
Driver.Instance.FindElement(By.Id("email")).Click();
// Sends data to user email
Driver.Instance.SwitchTo().ActiveElement().SendKeys("mary
.dailey@outlook.com");
Driver.Wait(5);
// Finds the position of the element - the position where the
user's password is marked and clicks on
Driver.Instance.FindElement(By.Id("pass")).Click();
// Sends data to the user's password.
Driver.Instance.SwitchTo().ActiveElement().SendKeys("solab
orate123");
Driver.Wait(5);
// Click on the Login button
var loginButton =
Driver.Instance.FindElement(By.Id("u_0_n"));
loginButton.Click();
}
[TestCleanup]
public void CleanUp()
{
Driver.Close();
}
}
}
```

Write the "LoginPage" class and from this class the browser opens, the Facebook url is marked, the user's email and pasword are sent, then the user is loged in to its account.

```
using FacebookAutomationFramework;
using System;
using System.Linq;
using OpenQA.Selenium;
namespace FacebookAutomation
{
public class LoginPage
{
public static void GoTo()
{
Driver.Instance.Navigate().GoToUrl(Driver.BaseAddress +
"login/");
}
public static void GoToFacebook()
{
Driver.Instance.Navigate().GoToUrl(Driver.BaseAddress);
}
public static LoginCommand LoginAs(string
username)
{
return new LoginCommand(username);
}
}
public class LoginCommand
{
private readonly string username;
private string password;
public LoginCommand(string username)
{
this.username = username;
}
public LoginCommand WithPassword(string
password)
{
this.password = password;
return this;
}
public void Login()
{
var loginInput =
Driver.Instance.FindElement(By.Id("username"));
loginInput.SendKeys(username);
var passwordInput =
Driver.Instance.FindElement(By.Id("password"));
passwordInput.SendKeys(password);
var loginButton =
Driver.Instance.FindElement(By.Id("u_0_n"));
loginButton.Click();
}
}
}
```

V. WHAT IS IMPORTANT WHEN WRITING A TEST METHOD?

It is that the assertion part should be created: to check whether you are on the right way or the steps that you have written are sending you to a certain position. For this reason, it is good to create assertions for each class you create so that you can check the results of the test methods. Asserting for the LoginPage class is this:

```
using FacebookAutomationFramework;
using OpenQA.Selenium;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace FacebookAutomation
{
public class LoginAssertion
{
public static bool UserLoggedIn
{
get
{
// The assertion looks for the position of the text in which
the page of the person I // is linked to his / her account, the
opposite, gives an error; that you are not on the right page
var UserInformation =
Driver.Instance.FindElement(By.XPath("/html/body/div[1]/di
v[2]/div[1]/div/div[2]/div[1]/div/div/div[1]/ul/li[1]/div/div/a"))
;
string UserInfoText = UserInformation.Text;
return UserInfoText.Equals("Kevin Smith");
}
}
}
}
```

One of the spaces in a page that can trigger a bug is the Post element (the item being posted) or the space where a user wants to post different types of posts. Since Facebook is an application widely used by users all over the world, if they are in the same time as you're also posting any simple post you encounter problems during posting your posts, then the performance of this, very well explained by Perfomance Load test. While these two types of test look at the performance and stability of an application when multiple users are logged in the same time space, then different bugs can be introduced, and it is easier if different tests are created for this. Whenever the code developer has made changes to the code and this change will go into production, the tester is obliged to make a smoke test (see for critical issue) and this can do it by executing the test methods. Testing methods for posting different posts is written belowe:

```
namespace FacebookAutomationTests
{
[TestClass]
public class PostElementTest : FacebookInitialize
{
[TestMethod]
public void User_Can_Post_Status_Update()
{
//assigns string values to the methods to use later
NewPostPage.CreatePost("This is the test post title")
.WithBody("Let's post something!")
.NewPostUpdate();
Driver.ExplicitWait(5);
//Checks if the post has been posted or not
var firstPostInNewsFeed =
Driver.Instance.FindElement(By.ClassName("_3ccb"));
var postTitle =
firstPostInNewsFeed.FindElement(By.ClassName("_58jë")).Text;
Assert.IsTrue(postTitle.Equals("Let's post something!"),
"Post has not been added");
}
[TestMethod]
public void User_Can_Tag_Connections()
{
//assigns string values to the methods to use later
NewPostPage.WriteTitle("Having fun!")
.ChooseTag("Kevin Smith")
PublishTag();
}
// Execute Checkin Method
[TestMethod]
public void User_Can_Post_CheckIn()
{
//assigns string values to the methods to use later
NewPostPage.CreateCheckIn("Hanging out")
.WithLocation("Los Angeles")
.PublishCheckIn();
}
// Execute Photo Method
[TestMethod]
public void User_Can_Post_Photo()
{
//assigns string values to the methods to use later
NewPostPage.ChoosePhoto("Hard working as
hardware!")
.PublishPhoto();
}
// Execute Video Method
[TestMethod]
public void User_Can_Post_Video()
{
//assigns string values to the methods to use later
NewPostPage.ChooseVideo("Video Superiority")
.PublishVideo();
}
}
}
```

Some of the methods on a worksheet are:

Clear: Delete something, for example: when we use any entry or any text space, we can delete it. It should be known in what context it calls this method.

Click: Of course is used, when you click on any button or anything else that can be clicked and causes events in JavaScript, then you should know when to use it because you can't call this method for things that are not buttons or links. Example: Class "Chat" which enables opening a chat window as well as sending a message to that chat. Here is the above method:

```
public class Chat
{
public static void ChatWithSomeone()
{
Driver.Instance.FindElement(By.Id("fbDockChatBuddylistNu
b")).Click();
Driver.Wait(2);
Driver.Instance.SwitchTo().ActiveElement()
.SendKeys("Lirie Koraqi");
Driver.Instance.FindElement(By.CssSelecto
r("._5l38._42ef")).Click();
Driver.Wait(2);
Driver.Instance.SwitchTo().ActiveElement().SendKeys("");
Driver.Wait(2);
Driver.PressEnter();
}
}
```

FindElement: Finds the right element.
FindElements: Finds the right elements that you can then find through this desired element.
GetAttribute: Gets the requested attribute in an element.
GetCssValue: Finds the value of Css that is defined for that element.
SendKeys: Sends basic data for that element.For example:

```
Driver.Instance.FindElement(By.CssSelector("._5l38._42ef
")).Click();
```

VI. CONCLUSION

While testing software is a work done entirely, automatic testing is the right choice for that. All the testing that can be made to an app, each one has its own benefits and disadvantages. Automated testing though sometimes spends time on writing test methods, it also has its positive side. During writing test methods, you do manual testing at the same time, but at the same time learn a programming language. The key to all of this is to know what benefits and running costs really are, and then make a decision about what is the right way. Today's companies want to have software that is tested adequately, but it works fast and is completely accomplished. To accomplish this goal, these companies are largely using automated testing. For a product to be properly implemented, behind it, it requires a general testing by the CAB team that when the customer uses that product, do not be disappointed in the non-functionalization of the different parts. Automated testing through Selenium has enabled the client to remain satisfied with the final product when delivered. Selenium as a tool that is open-source, do not need any plugin on the computer because you can download it free of charge from the internet, this has enabled testers to have knowledge in both manual and automatic testing and be able to learn other programming languages, which then requires it in their further development.

Creating a working framework to create classes that enable testing of certain parts of web applications helps testers collaborate and interact with each other. The way it controls a page to find the right elements to include as part of the code enables the tester to create a wider space with regard to programming languages. In the end of this, we conclude that Selenium and automated testing have enabled each customer to have a quality, fast and without errors product as well as the most efficient operation of the product intended to be used by users.

REFERENCES

[1] Ranstrom.R, "Automated Web Software Testing with Selenium", Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556.

[2] Grater T.M, "Benefits of using automated software testing tools to achieve software quality assurance", Intermediate Methods Analyst, United Parcel Service, June 2015.

[3] Chinmay Sathe & Amit Prabhu, “Automation using Selenium”, Cybage Software Pvt. Ltd.

[4] Linda G. Hayes, “The Automated Testing Handbook”.

[5] Navneesh Garg, “Test Automation using Selenium WebDriver with Java: Step by Step Guide”.

[6] Mark Garzone, “Software Testing: A Guide to Testing Mobile Apps, Websites, and Games”.

[7] https://www.solaborate.com/

Festim Halili currently works at the Department of Computer Sciences, State University of Tetovo. Festim does research in Service Oriented Computing, Information Systems (Business Informatics), Software Engineering, Database Systems and Artificial Intelligence. Their current project is 'Book: Software Engineering - A perspective on Service Oriented Computing.