

AUTOMATION REGRESSION SUITE CREATION FOR HEALTH CARE SOLUTION

Anjali Rawat¹ and Shahid Ali²

¹Department of Information Technology, AGI Institute, Auckland, New Zealand

²Department of Information Technology, AGI Institute, Auckland, New Zealand

ABSTRACT

Regression testing is very important for dynamic verification. It helps to simulate a suite of test cases periodically and after major changes in the design or its environment, in order to check that no new bugs were introduced. Evidences regarding benefit of implementing automation testing which includes saves of time and cost as it can re-run test scripts again and again and hence is much quicker than manual testing, providing more confidence in the quality of the product and increasing the ability to meet schedules and significantly reducing the effort that automation requires from testers are provided on the basis of survey of 115 software professionals. In addition to this, automated regression suite has an ability to explore the whole software every day without requiring much of manual effort. Also, bug identification is easier after the incorrect changes have been made. Occupational Health Management Solution (OHMS) is going through continuous development and requires testing again and again to check if new feature implementation has affected the existing functionality. In addition to this, The Company is facing issue in validation of the OHMS installation at client site since it requires availability of testers to check the critical functionality of the software manually. Company wants to create an automated regression suite for OHMS which can be executed at client site for checking the functionality of the software. In addition to this, this suite will also help the testing team to validate if the new features which have been added to the existing software are affecting the existing system or not. Visual studio, Selenium Webdriver, Visual SVN and Trello are the tools which have been used to achieve the creation of automation regression suite. The current research will provide guidelines to the future researchers on how to create an automated regression suite for any web application using open source tools.

KEYWORDS

Automation testing, Regression testing, Visual Studio, C#, Selenium Webdriver, Agile- Scrum

1. INTRODUCTION

The Company was founded in collaboration with the University of Auckland in 2002 which mainly focuses on creating health care solutions. The two main products of this Company are Company's Occupational Health (OH) which is a cloud-based occupational health platform designed to help organizations proactively manage employee health monitoring, pre-employment and wellbeing activities and Company's Framework (FW) which is comprised of a number of common features that are prominent across many health systems allowing organization to get the benefits of a custom solution, cost effectively. This Company is creating a cloud based Occupational Health Management Solution (OHMS) which manages and records

all Electronic Health Record (EHR) for client employees and customers. An EHR is a person-centric dataset which provides a 360 degree view of an individual's health and well-being. It must capture and manage information such as patient demographics, progress notes, medications, past medical history, immunizations and laboratory tests. OHMS is going through continuous development to meet the needs of diverse clients which in turn requires testing of software again and again to check if new feature implementation have affected the existing functionality of the software. In addition to this, Company is facing issue in validation of the OHMS installation at client site since it requires availability of testers to check the critical functionality of the software manually. Both these activities need lot of time and efforts of manual testing therefore, to save the time and effort which are getting used in this whole process Company wants to create an automated regression suite for OHMS which can be executed at client site for checking the functionality of the software. In addition to this, this suite will also help the testing team to validate if the new features which have been added to the existing software are affecting the existing system or not.

The scope of this project is to achieve the functional automation testing and later to create automated regression test suite. This regression test suite will help the Company to confirm if any new changes in the code have affected any existing feature of OHMS. Since this activity of validation is a repetitive process which includes execution of the same functional test case therefore Company have come with a solution of automating the test cases which not only will reduce the execution time but will also help in saving the cost and effort.

This research project report is organized as follow: Section 2 focuses on the literature review of various studies concentrating on automation regression. Section 3 is focused on the research methodology for this research project. Section 4 of this research is focused on project execution results. Discussion to results of this project is provided in section 5. Section 6 is dedicated towards the future work recommendations. Finally, in section 7 conclusions to the research is provided.

2. LITERATURE REVIEW

Different research paper has been referred and analyzed before implementing the methodology or using any tool and technique for this project. Regression testing is very important for dynamic verification. It helps to simulate a suite of test cases periodically and after major changes in the design or its environment, in order to check that no new bugs were introduced [1]. Evidences regarding benefit of implementing automation testing which includes saves of time and cost as it can re-run test scripts again and again and hence is much quicker than manual testing, providing more confidence in the quality of the product and increasing the ability to meet schedules and significantly reducing the effort that automation requires from testers are provided on the basis of survey of 115 software professionals [2]. In addition to this, automated regression suite has an ability to explore the whole software every day without requiring much of manual effort. Also, bug identification is easier after the incorrect changes have been made [3].

Selenium is the best choice of open source tool for web based testing being able to support all prominent browsers and operating systems. It supports automation of dynamic web pages where web elements changes without the page being reloaded [4]. The selection of selenium Webdriver can be further supported by highlighting the easiness of test suite maintenance using selenium Webdriver [9]. Creation of automation framework for web application with "Selenium Webdriver could reduce time for development, increase the return on investments (ROI) and minimize the risk" (Elior, 2017).

Visual studio is an open source full-featured integrated development environment (IDE) for Android, IOS, Windows, web, and cloud. It enables to work with the Git repository, including those in GitHub. The functionality of Visual Studio (VS) can be extended by developers through plug-ins. Plug-ins can add new or customized functionality by exploiting the extensibility features of VS components. Numerous extensions for Visual Studio can be found in the Visual Studio Gallery [6]. Trello is a free cloud-based project management solution to assist project leaders in managing large projects, planning programs, and coordinating teams [7].

Scrum methodology is ideal for rapidly changing, accumulating requirements, provides customer by optimizing turnaround time and responsiveness to requests and work estimates are much easier [8].

Though most of the research paper demonstrates the benefit of using selenium Webdriver, visual studio and achieving regression testing using automation but they failed to explain how the combination of selenium Webdriver with Visual Studio can help any organization to achieve regression automation.

In addition, most of these research papers are confined to the usage of JAVA as the programming language in their projects, whereas, a large portion of companies in New Zealand still uses C# as a programming language in their coding. This report has explained how automation can be achieved using C# with selenium Webdriver.

3. RESEARCH METHODOLOGY

Research methodology for OHMS Project has been discussed below.

3.1. Proposed Architecture for the project

The complete automation testing architecture for OHMS Project is shown in Figure 1. All the test scripts are divided into three categories namely the main class (which will initialize the scripts for execution), Base class (which has two abstract methods namely up and down methods for creating and deleting entries respectively) and the test scripts which contain all the web elements and the actions which are performed on them.

Main class calls “Browser start” class which will initialize the Webdriver and later return the initialized Webdriver to the main class. Test script’s classes inherit and implement the base class. Once these two actions are complete, main class start calling test script’s classes one by one, to perform the actions on the OHMS and at the same time displays the results in the console.

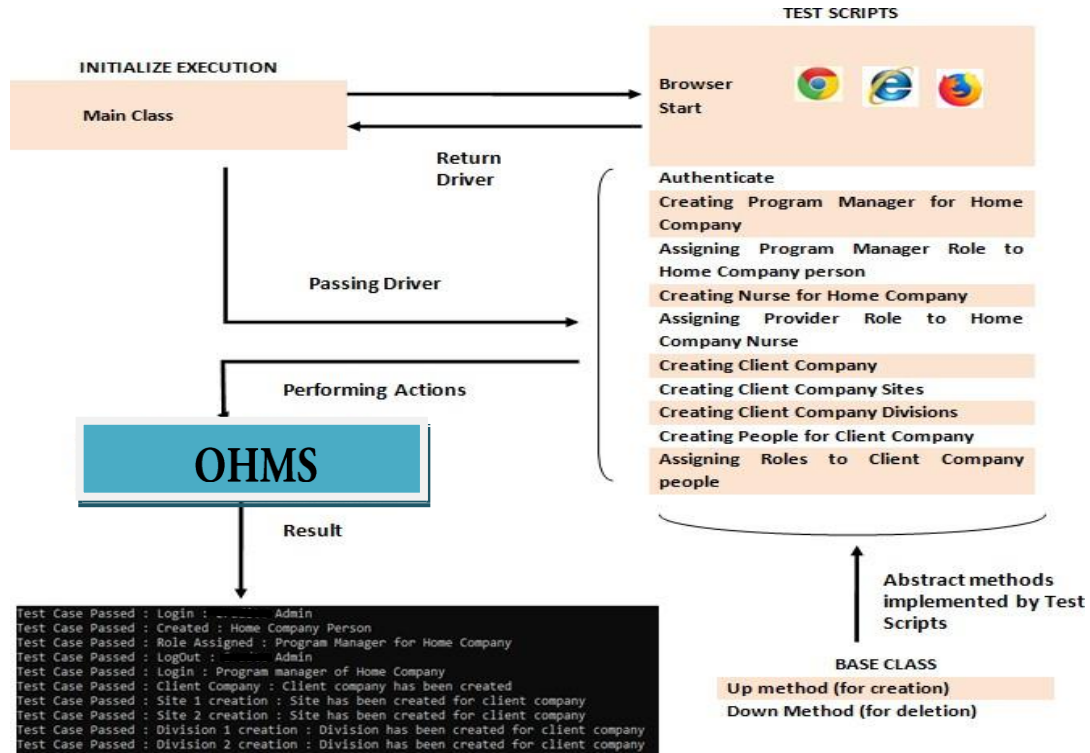


Figure 1. Proposed Architecture for OHMS Project

3.2. Project Execution

3.2.1. Testing Phases for Project

Automation testing phase which has been implemented by OHMS project is shown in figure 2. This explains the different activities which have been performed in each stage. The whole project is divided into 6 phases. First phases include requirement gathering and analysis of requirements. In this Company, Business Analyst (BA) along with the help of product owner (PO) accomplished this task. Second step consists test planning where PO will help the scrum team in identifying the user stories and will also prioritize the user stories which need to be covered in each sprint. Third phase is completed by setting up the testing environment. During this phase, Visual Studio has been installed in the local machine, Selenium Webdriver related jar files and libraries has been imported and Visual SVN is installed which helped the team to commit the updated code to the central repository. Once the setup was ready, team started writing test scripts while making sure that they are covering all the requirements. This was ensured with the help of traceability matrix where all requirements have been mapped with a number of test cases. In the execution phase, test scripts are executed and once they are passed they are committed to the central repository. The last phase was the test result analysis phase where the complete code and the output was reviewed by the scrum team, and team members would suggest areas of improvement.

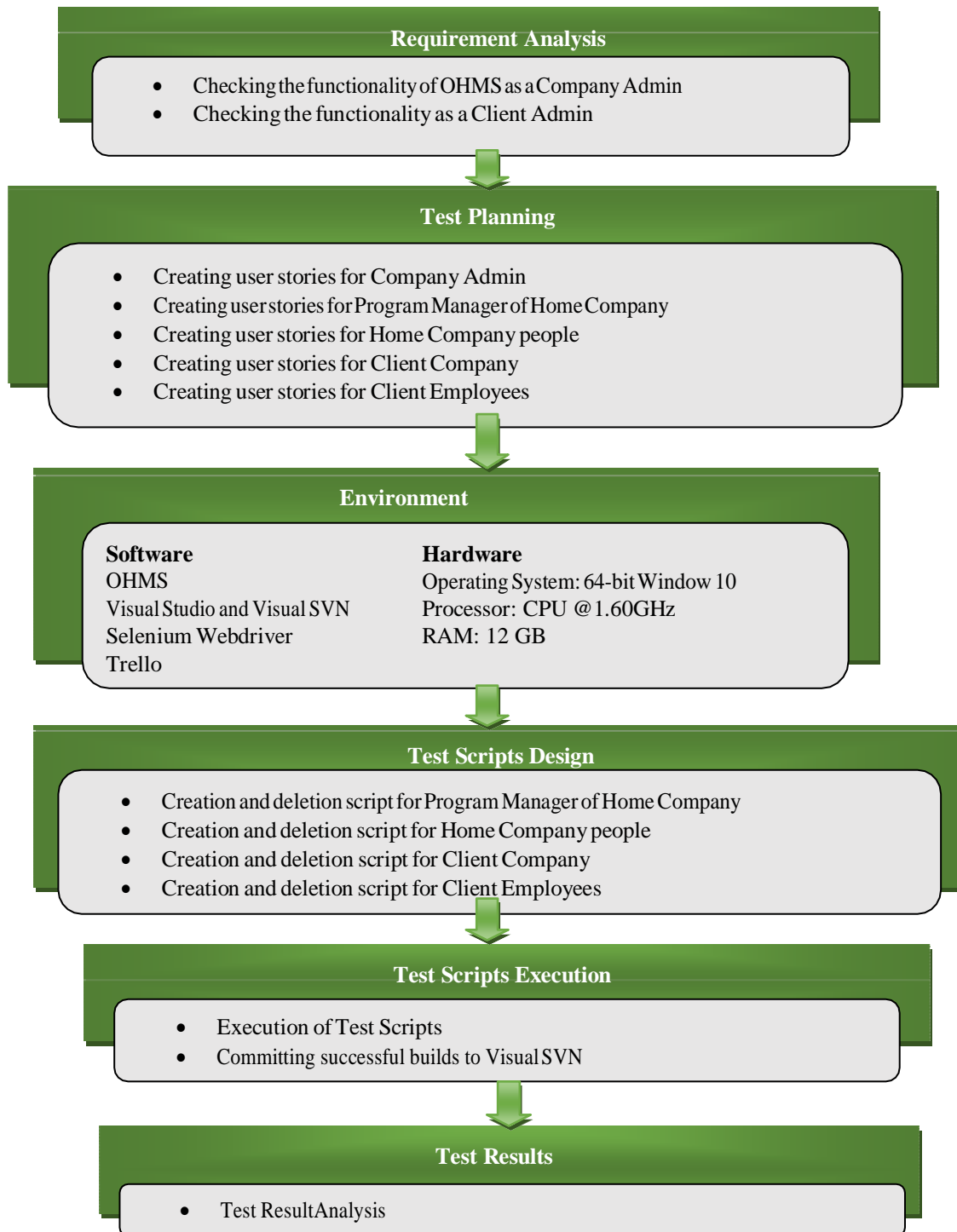


Figure 2. Testing Phases for OHMS project

3.2.2. Test Cases

Test cases are the set of steps and conditions which helps to determine whether application under test works correctly and meets the requirements. Some of the test cases which have been created for OHMS are shown in Table 1. This table explains what each test case is supposed to

do, pre-requisites which need to be fulfilled, detailed steps which need to be followed to achieve the expected result. Test cases are helpful in ensuring that all the requirements have been covered and creates a layout for the automation scripts.

Table 1. Test cases for OHMS

Test Case ID	Test Case Description	Pre- requisites	Steps	Expected Result	Actual Result	Result
TC_001	To verify Company's Admin is able to login to OHMS using valid credentials	URL : OHMS website Username : "Admin" Password : "XXXXXX"	1.Navigate to "OHMS website" 2.Enter username 3.Enter Password 4.Click on Login	User should be logged-In to OHMS successfully	User is successfully logged- in	PASS
TC_002	To verify user is able to create person for Program manager (PM) role	Logged-In as Company Admin	1. Navigate to "My Company" tab 2. Go to "People" tab 3.Click New 4. Fill details 5. Create	A person should get created for home company	Person is successfully created	PASS
TC_003	To verify user is able to assign PM role and valid credentials to newly created person	Logged in as Company Admin	1. Navigate to "Security" tab 2. Enter password 3.Click on "Program Manager" role 4. Create	"Program Manager" role should be assigned to newly created person successfully.	"Program Manager" role is assigned successfully	PASS
TC_004	To verify user is able to logout as Company's Admin	Logged in as Company Admin	1. Navigate to "Menu" 2. Click on "Sign-Out"	User should be logged-Out successfully	User is successfully logged-Out	PASS
TC_005	To verify PM is able to login to OHMS	URL : OHMS website Username : "PM HomeCompany" Password : "XXXXXX"	1.Navigate to "OHMS website" 2.Enter username 3.Enter Password 4.Click on Login	PM should be logged-In to OHMS successfully	PM is successfully logged in	PASS

TC_006	To verify PM is able to create Client company	Logged in as PM	<ol style="list-style-type: none"> 1. Click on “Client Company” tab 2. Click on “New” 3. Fill details 4. Click on “Create” 	Client Company should be created successfully	Client Company is created successfully	PASS
TC_007	To verify PM is able to create Client company Sites	Logged in as PM	<ol style="list-style-type: none"> 1. Click on newly created “Client Company” 2. Click on “New” 3. Click on “New Site” 4. Fill details 5. Click on “Create” 	Sites should be created successfully	Sites are created successfully	PASS
TC_008	To verify PM is able to create Client company Division	Logged in as PM	<ol style="list-style-type: none"> 1. Click on newly created “Client Company” 2. Click on “New” 3. Click on “New Division” 4. Fill details 5. Click on “Create” 	Divisions should be created successfully	Divisions are created successfully	PASS
TC_014	To verify PM is able to create Client admin for the client company	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to newly created “Client Company” 2. Go to “People” tab 3. Click New 4. Fill details 5. Create 	A person should get created for client company	User is successfully logged in	PASS
TC_015	To verify PM is able assign role and valid credentials to client admin	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “Security” tab 2. Enter password 3. Click on “Client Admin” role 4. Create 	Person should be assigned “Client Admin” role	“Client Admin” role is assigned successfully	PASS
TC_016	To verify PM is able to logout from OHMS	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “Menu” 2. Click on “Sign-Out” 	User should be logged-Out of OHMS successfully	User is successfully logged-Out	PASS

TC_017	To verify PM is able to revoke provider role from home company nurse	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “My Company” tab 2. Go to “People” tab. 3. Search for the nurse 4. Navigate to “Security” tab 5. Uncheck “Provider” role radio button 6. Save 	Provider role should be revoked from the nurse successfully	Provider role is revoked successfully	PASS
TC_018	To verify PM is able to delete Nurses from home company	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “My Company” tab 2. Go to “People” tab. 3. Search for the nurse 4. Click “delete” tab 5. Delete 	Nurse should be deleted successfully	Nurse is deleted successfully	PASS
TC_022	To verify PM is able to delete client admin from client company	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “client Company” tab 2. Search for the client company 3. Search for the client admin 4. Click “delete” tab 5. Delete 	client admin should be deleted successfully	client admin is deleted successfully	PASS
TC_023	To verify PM is able to delete employees of client company	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “client Company” tab 2. Search for the client company 3. Search for the Employee 4. Click “delete” tab 5. Delete 	Employee should be deleted successfully	Employee is deleted successfully	PASS
TC_024	To verify PM is able to delete client company Divisions	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “Client Company” tab 2. Search for the client company 3. Search for the Division 4. Click “delete” tab 5. Delete 	Division should be deleted successfully	Division is deleted successfully	PASS

TC_025	To verify PM is able to delete client company Sites	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “Client Company” tab 2. Search for the client company 3. Search for the Site 4. Click “delete” tab 5. Delete 	Site should be deleted successfully	Site is deleted successfully	PASS
TC_026	To verify user is able to revoke PM role	Logged in as Company Admin	<ol style="list-style-type: none"> 1. Navigate to “My Company” 2. Go to “People” tab. 3. Search for the PM 4. Navigate to “Security” tab 5. Uncheck “Program Manager” role radio button 6. Save 7. 	“Program Manager” role should be revoked successfully	“Program Manager” role is revoked successfully	PASS
TC_027	To verify user is able to delete newly created program manager	Logged in as PM	<ol style="list-style-type: none"> 1. Navigate to “My Company” tab 2. Go to “People” tab. 3. Search for the PM 4. Click “delete” tab 5. Delete 	PM should be deleted successfully	PM is deleted successfully	PASS

3.2.3. Traceability Matrix

Traceability matrix is one of the most important elements of testing. Traceability matrix shows the mapping of the requirements with the test cases and ensures that all the critical requirements of the application have been covered by the test cases. The matrix in Table 2. shows the number of test cases which has been created in order to give full test coverage to the particular requirement. Also, in case any of the requirement changes in future, it become easier for the tester to track the number of test cases which has been affected by that particular requirement change.

Requirement ID	Requirement Description	Test Case ID	Status
OHMS_RID_001	To check Login functionality of the OHMS	TC_001 TC_005	Pass
OHMS_RID_002	To check Company Admin rights which involves : Creating and deleting the person Assigning and Revoking role to/from the person	TC_002 TC_003 TC_026 TC_027	Pass Pass Pass Pass
OHMS_RID_003	To check Program Manager rights which involves : Creating Client company Creating and deleting Client company sites Creating and deleting Client company divisions Creating and deleting client people Assigning and revoking roles to/from the client company people	TC_006 TC_007 TC_008 TC_011 TC_012 TC_013 TC_014 TC_015 TC_019 TC_020 TC_021 TC_022 TC_023 TC_024 TC_025	Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass Pass
OHMS_RID_004	To check Program manager rights which involves : Creating and deleting Home company people Assigning and revoking roles to/from the Home company people	TC_009 TC_010 TC_017 TC_018	Pass Pass Pass Pass
OHMS_RID_005	To check Logout functionality of the OHMS	TC_004 TC_016	Pass Pass

3.2.4. Solution Structure for OHMS Project

Complete structure in which OHMS automation scripts have been managed inside Visual Studio is shown in Figure 3. It consists of three main folders namely: Initialize Execution, Task Manager and Test Scripts. There is a separate section named as “Dependencies” which consist of all the selenium libraries and packages which have been installed in visual studio for carrying out the automation.

Initialize Execution: It has the main class which helps to initialize the web driver.

Task Manager: It has the base class which needs to be implemented by all the classes of “Test Script” folder.

Test Scripts: It has the scripts, which contains selenium and C# commands for carrying out the execution.

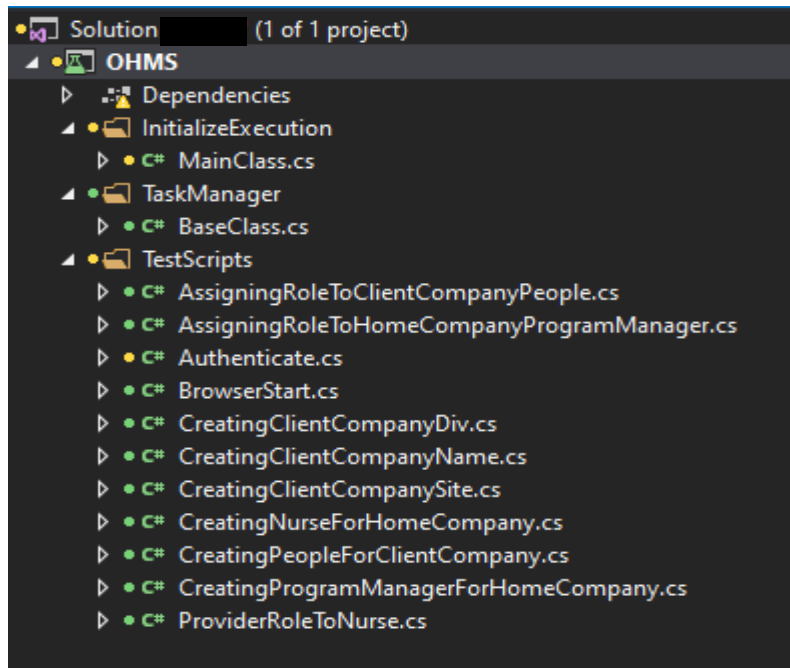


Figure 3. Solution structure for OHMS Project

3.2.4.1. Task Manager Folder

This folder contains the Base class which has two abstract methods namely Up and Down under it. This base class is inherited, and methods are implemented by the classes of “Test Scripts” folder. Up method is used to create new entries in the OHMS and down method is used to delete the same entries at later stage.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace OHMS.TaskManager
{
    public abstract class BaseClass
    {
        public abstract bool Up();
        public abstract bool Down();
    }
}
```

Figure 4. Task Manager

3.2.4.2. Initialize Execution folder

This folder contains the main class. This main class is the starting point of the execution and calls all other classes for performing different tasks. The first action performed by the main class is to pass driver name to the “browser start” class. Depending upon the passed parameter, “browser start” class will initialize the driver of the same browser and later will return the initialized driver back to the main class. Main class with the help of that driver will navigate to OHMS official site and later will call other test scripts to perform next actions. The code which

has been used to create main class is shown in Figure 5.

```
public static void Main(string[] args)
{
    var successfullyCompletedTasks = new List<testTask>();
    var driver = BrowserStart.StartBrowser("Chrome", "OHMS website");
    int WAIT_TIME = 3000;
    string CompanyName = "New Company";
    string Male = "///input[@value='M']";
    string Female = "///input[@value='F']";
    string Site1 = "Headquarter";
    string Site2 = "Production";
    string Div1 = "Division1";
    string Div2 = "Division2";
    string Password = "XXXXXX";

    //Login as Admin
    Authenticate authenticateobjAdmin = new Authenticate(driver, "Admin", "XXXXX");
    Authenticate authenticatePMobj = null;
    CreatingProgramManagerForHomeCompany CreatingProgramManagerForHomeCompanyObj = null;
    AssigningRoleToHomeCompanyProgramManager assigningRoleobject = null;
    CreatingClientCompanyName CreatingClientCompanyNameobj = null;
    CreatingClientCompanySite CreatingClientCompanySite1obj = null;
    CreatingClientCompanySite CreatingClientCompanySite2obj = null;
    CreatingClientCompanyDiv CreatingClientCompanyDiv1obj = null;
    CreatingClientCompanyDiv CreatingClientCompanyDiv2obj = null;
    CreatingNurseForHomeCompany Nurse1ForHomeCompanyobj = null;
    CreatingNurseForHomeCompany Nurse2ForHomeCompanyobj = null;
    ProviderRoleToNurse ProviderRoleToNurse1obj = null;
    ProviderRoleToNurse ProviderRoleToNurse2obj = null;
    CreatingPeopleForClientCompany MaleEmployeeAucklandDiv1Obj = null;
    CreatingPeopleForClientCompany FemaleEmployeeAucklandDiv2Obj = null;
    CreatingPeopleForClientCompany MaleEmployeeWaikatoDiv1Obj = null;
    CreatingPeopleForClientCompany FemaleEmployeeWaikatoDiv2Obj = null;
    CreatingPeopleForClientCompany ClientAdminCreationObj = null;
    AssigningRoleToClientCompanyPeople AssigningAdminRoletoClientAdminObj = null;
    CreatingPeopleForClientCompany CompanyManagerCreationObj = null;
    AssigningRoleToClientCompanyPeople AssigningCompanyManagerRoleObj = null;
}
```

```
if (authenticateobjAdmin.Up())
{
    //creating home company person
    string Title = driver.Title;
    string AdminName = driver.FindElement(By.Id("uname")).Text;

    if (Title.Equals("Administration") && AdminName.Equals("Admin"))
        Console.WriteLine("Test Case Passed : Login : Admin");
    else
        Console.WriteLine("Home page is not opened");
    CreatingProgramManagerForHomeCompanyObj = new CreatingProgramManagerForHomeCompany(driver, "PM", "HomeCompany",
    "PM@example.com");

    if (CreatingProgramManagerForHomeCompanyObj.Up())
    {
        successfullyCompletedTasks.Add(testTask.CreatingProgramManagerForHomeCompanyEnum);
        string PersonTitle = driver.Title;
        if (PersonTitle.Equals("Person"))
            Console.WriteLine("Test Case Passed : Created : Home Company Person");
        else
            Console.WriteLine("Person is not created");

        //Assigning PM role
        assigningRoleobject = new AssigningRoleToHomeCompanyProgramManager(driver, "PM", "XXXXXX",
        "///div[@class='\\editor-field rolerlist\\']//input[@value=5]");

        if (assigningRoleobject.Up())
        {
            string RoleTitle = driver.Title;
            if (RoleTitle.Equals("Details"))
                Console.WriteLine("Test Case Passed : Role Assigned : Program Manager for Home Company");
            else
                Console.WriteLine("PM Role is not assigned");
            successfullyCompletedTasks.Add(testTask.AssigningRolesEnum);
        }
    }
}
```

```

//Logout as Admin
Thread.Sleep(WAIT_TIME);
driver.FindElement(By.Id("menulinkspan")).Click();
Thread.Sleep(WAIT_TIME);
driver.FindElement(By.XPath("//a[contains(., 'Sign Out')]")).Click();
Thread.Sleep(WAIT_TIME);
Console.WriteLine("Test Case Passed : Logout : Admin");
authenticatePMobj = new Authenticate(driver, "PM", "XXXXX");

if (authenticatePMobj.Up())
{
    successfullyCompletedTasks.Add(testTask.AuthenticatePMEnum);
    string ProgramManagerName = driver.FindElement(By.Id("uname")).Text;
    if (ProgramManagerName.Equals("PM HomeCompany"))
        Console.WriteLine("Test Case Passed : Login : Program manager of Home Company");
    else
        Console.WriteLine("Home page is not opened");
    CreatingClientCompanyNameobj = new CreatingClientCompanyName(driver, CompanyName);

    if (CreatingClientCompanyNameobj.Up())
    {
        successfullyCompletedTasks.Add(testTask.CreatingClientCompanyNameEnum);
        string CompanyTitle = driver.Title;
        if (CompanyTitle.Equals("Details"))
            Console.WriteLine("Test Case Passed : Client Company : Client company has been created");
        else
            Console.WriteLine("Company is not created");
        CreatingClientCompanySiteobj = new CreatingClientCompanySite(driver, Site1, "New Zealand:Auckland Region");

        if (CreatingClientCompanySiteobj.Up())
        {
            successfullyCompletedTasks.Add(testTask.CreatingClientCompanySite1Enum);
            Console.WriteLine("Test Case Passed : Site 1 creation : Site has been created for client company");
            CreatingClientCompanySite2obj = new CreatingClientCompanySite(driver, Site2, "New Zealand:Waikato");
        }
    }
}

```

```

successfullyCompletedTasks.Reverse();
foreach (var task in successfullyCompletedTasks)
{
    switch (task)
    {
        case testTask.AuthenticatePMEnum:
            authenticatePMobj.Down();
            Console.WriteLine("Test Case Passed : Logout : Program Manager");
            break;

        case testTask.CreatingProgramManagerForHomeCompanyEnum:
            CreatingProgramManagerForHomeCompanyObj.Down();
            Console.WriteLine("Test Case Passed : Person deletion : Person of Home Company successfully removed");
            Console.WriteLine("Test Case Passed : Logout : Admin");
            break;

        case testTask.AssigningRolesEnum:
            assigningRoleobject.Down();
            Console.WriteLine("Test Case Passed : Role removed : Program Manager role for Home Company person is" +
                " successfully revoked");
            break;

        case testTask.CreatingClientCompanyNameEnum:
            CreatingClientCompanyNameobj.Down();

            break;

        case testTask.CreatingClientCompanySite1Enum:
            CreatingClientCompanySiteobj.Down();
            Console.WriteLine("Test Case Passed : Site 1 : Successfully removed");
            break;

        case testTask.CreatingClientCompanySite2Enum:
            CreatingClientCompanySite2obj.Down();
            Console.WriteLine("Test Case Passed : Site 2 : Successfully removed");
            break;
    }
}

```

Figure 5. Main class

3.2.4.3. Test Scripts Folder

This folder contains the scripts which consist of all the actions which are need to be performed by the script. Each class under “Test Scripts” folder is named as per the task they are performing. Below is the brief explanation about the functionality of each script:

- **Brower Start:** Script to initialize the browser and to navigate to OHMS website.
- **Authenticate:** Script which will help the user to get logged-In to the OHMS and later to get logged-Out.
- **Creating Program Manager for Home Company:** Script to create a person for the home company and later to delete this person from the system.
- **Assigning Program Manager Role to Home Company person:** Script to assign Program Manager Role to the newly created person and to generate credentials and later to revoke the assigned role from the same person.
- **Creating Nurse for Home Company:** Script to create a Nurse for the home company and later to delete the nurse from the system.
- **Assigning Provider Role to Home Company Nurse:** Script to assign Provider role to the newly created nurse and to generate credentials and later to revoke the assigned role from the same person.
- **Creating Client Company:** Script to create client company.
- **Creating Client Company Sites:** Script to create sites for client company and later to remove these sites from the client company.
- **Creating Client Company Divisions:** Script to create divisions for client company and later to remove these divisions from the client company.
- **Creating People for Client Company:** Script to create employees,client admin and company manager for the client company and later to remove these people from the client company.

- **Assigning Roles to Client Company people:** Script to assign admin role and company manager role to the concerned people and later to revoke the assigned role from the same people.

3.2.4.4. Selected Tool

Below are few of the code snippets which have been created for OHMS using C# and selenium Webdriver. Appendices can be referred for the code snippets of all scripts.

Start browser class: It has implemented the three major drivers namely: Chrome, Firefox and IE. Depending on the user requirement, user passes a driver name from the main class. Once the driver name gets passed, the driver gets automatically initialized in the “Start Browser”. After initializing the driver, “Start Browser” returns the driver to the main class and now main class use this driver to execute the actions on the OHMS. The complete code for this class can be seen in Figure 6.

```
public class BrowserStart
{
    static IWebDriver driver;
    public static IWebDriver StartBrowser(String browsername, String URL)
    {
        if (browsername.Equals("Chrome"))
        {
            driver = new ChromeDriver(@"C:\Users\Anjali\.nuget\packages\selenium.webdriver.chromedriver\75.0.3770.140\driver\win32");
            //driver = new ChromeDriver();
        }
        else if (browsername.Equals("Firefox"))
        {
            driver = new ChromeDriver(@"C:\Users\Anjali\.nuget\packages\selenium.webdriver.chromedriver\75.0.3770.140\driver\win32");
            // System.setProperty("webdriver.gecko.driver", "C:\\Temp\\geckodriver.exe");
            driver = new FirefoxDriver();
        }
        else if (browsername.Equals("IE"))
        {
            driver = new ChromeDriver(@"C:\Users\Anjali\.nuget\packages\selenium.webdriver.chromedriver\75.0.3770.140\driver\win32");
            // System.setProperty("webdriver.ie.driver", "C:\\Temp\\IEDriverServer64bit.exe");
            driver = new InternetExplorerDriver();
        }
        driver.Navigate().GoToUrl(URL);
        driver.Manage().Window.Maximize();
        return driver;
    }
}
```

Figure 6. Start Browser Class

Authenticate Class: The test scripts which is created for the login credentials and then for logout is shown in Figure 7. The main class calls this authenticates class and at the same time passes the driver and the login credentials to this class. This class receives the parameters and at the same time inherited Base class so that it can implement the up and down methods. In up methods, the script is trying to login to the OHMS using valid login credentials and once all the entries are made, it finally logout from the OHMS using down method.

```

class Authenticate : BaseClass
{
    private IWebDriver driver;
    private static int WAIT_TIME = 3000;
    private string uid;
    private string pass;

    public Authenticate(IWebDriver Driver, string Uid, string Pass) {
        this.driver = Driver;
        this.uid = Uid;
        this.pass = Pass;
    }

    public override bool Up()
    {
        driver.FindElement(By.Id("UserName")).SendKeys(uid);
        driver.FindElement(By.Id("Password")).SendKeys(pass);
        driver.FindElement(By.Id("LoginButton")).Click();
        Thread.Sleep(WAIT_TIME);
        return true;
    }

    public override bool Down()
    {
        Thread.Sleep(WAIT_TIME);
        driver.FindElement(By.Id("menulinkspan")).Click();
        Thread.Sleep(WAIT_TIME);
        driver.FindElement(By.XPath("//a[contains(., 'Sign Out')]")).Click();
        Thread.Sleep(WAIT_TIME);
        Authenticate authenticateobj = new Authenticate(driver, "XXXXX@gmail.com", "XXXXX");
        return true;
    }
}

```

Figure 7. Authenticate class

4. RESULTS

The results of the test scripts execution are shown in Figure 8. From the output it can be seen that all the test scenarios have been covered for OHMS. The first half of the results is more about login test cases, followed by creating company and employees for that company and then assigning different roles to these people according to their profile.

The second half of the result is more about revoking the assigned roles, deleting the created entries and finally getting logged-Out from the OHMS. Deletion of these entries was critical for this project as Company had limited database storage. And also, earlier these entries were getting deleted manually by the database admin which was again a time consuming activity. Hence by automating the deletion process, testing team has saved a lot of effort and time.


```

C:\Program Files\dotnet\dotnet.exe
Test Case Passed : Login : Admin
Test Case Passed : Created : Home Company Person
Test Case Passed : Role Assigned : Program Manager for Home Company
Test Case Passed : LogOut : Admin
Test Case Passed : Login : Program manager of Home Company
Test Case Passed : Client Company : Client company has been created
Test Case Passed : Site 1 creation : Site has been created for client company
Test Case Passed : Site 2 creation : Site has been created for client company
Test Case Passed : Division 1 creation : Division has been created for client company
Test Case Passed : Division 2 creation : Division has been created for client company
Test Case Passed : Nurse 1 creation : Nurse 1 has been created for home company
Test Case Passed : Provide role given : Nurse 1 has been given provider role for home company
Test Case Passed : Nurse 2 creation : Nurse 2 has been created for home company
Test Case Passed : Provide role given : Nurse 2 has been given provider role for home company
Test Case Passed : Male Employee Auckland site creation : Employee has been created for Div 1 in Auckland Site
Test Case Passed : Female Employee Auckland site creation : Employee has been created for Div 2 in Auckland Site
Test Case Passed : Male Employee Waikato site creation : Employee has been created for Div 1 in Waikato Site
Test Case Passed : Female Employee Waikato site creation : Employee has been created for Div 2 in Waikato Site
Test Case Passed : Person created for client Admin
Test Case Passed : Client Admin Role Assigned to client company person
Test Case Passed : Person created for company manager
Test Case Passed : Company manager Role Assigned to client company person
Test Case Passed : Company Manager role from the person of Home company has been successfully revoked
Test Case Passed : Person for the role of Company Manager is successfully removed
Test Case Passed : Person for the role of Client Admin is successfully removed
Test Case Passed : Client Admin role from the person of Home company has been successfully revoked
Test Case Passed : Waikato Female employee successfully removed
Test Case Passed : Waikato Male employee successfully removed
Test Case Passed : Auckland Female employee successfully removed
Test Case Passed : Auckland Male employee successfully removed
Test Case Passed : Provider role from Nurse 2 is successfully revoked
Test Case Passed : Nurse 2 is successfully removed
Test Case Passed : Provider role from Nurse 1 is successfully revoked
Test Case Passed : Nurse 1 is successfully removed
Test Case Passed : Division 2 : Successfully removed
Test Case Passed : Division 1 : Successfully removed
Test Case Passed : Site 2 : Successfully removed
Test Case Passed : Site 1 : Successfully removed
Test Case Passed : Company name cannot be deleted
Test Case Passed : LogOut : Program Manager
Test Case Passed : Login : Admin
Test Case Passed : Role removed : Program Manager role for Home Company person is successfully revoked
Test Case Passed : Person deletion : Person of Home Company successfully removed
Test Case Passed : LogOut : Admin

```

Figure 8. Test script execution result for OHMS

5. DISCUSSION

The whole purpose of this project was to create automation scripts for OHMS which will be used for regression testing later. Creation of automation regression suit is accomplished by using Visual studio and selenium Webdriver with C# programming language.

The major issue Company was facing was to test the whole application manually again and again, every time a change is introduced to the application and the need of physically going to client location to validate the correct installation of OHMS in client premises. Research papers and the requirements of this project helped to determine the need of going for regression automation testing. Moreover, budget constraint for OHMS suggested the team to go for the open source tools. These are not only free but also have a huge community support.

Based on the successful delivery of software by the Company in past years using Scrum

methodology, this time also they came up with the adoption of Scrum. It helps the team to keep an eye on the progress of the project as well as it is helpful when there are constant changes in client requirements. In addition to this there are many research papers which support the use of Scrum methodology for a successful delivery of a product.

There were few bottlenecks which were encountered during the automation of OHMS. One of them was having limited database for storing the values. Every time a new entry is made in the application it has to be deleted manually. Another issue was the data which was getting entered into the application was getting linked to the other database following primary-foreign key concept. Hence it was impossible to delete the primary keys without first deleting the linked foreign keys.

As a solution for this, team comes up with the idea of creating a deleting mechanism as well for the scripts. So, for every entry which has been made in the application there was a connected delete mechanism and hence the need of up and down method came into the picture.

One more issue was related to the web elements. There were few web elements available in the application that were not assigned unique id or name and were changing the values dynamically which made their detection quite difficult. Since the application was still in the stage of getting complete, having continuous changes and updates, therefore taking web elements Xpath for locating them was not a good approach. To overcome this problem, it was decided that testing team need to make a list of all these types of web elements and need to ask developers to assign unique ids. In places where giving unique ids were not possible, relative Xpath have been used.

6. RECOMMENDATIONS

Several recommendations are derived after completing this project, which are discussed below.

- OHMS is still in the process of getting completed and new features are still getting added to this application based on the feedback from the client after each sprint. Therefore, going for automation at the early age of development cycle is not advised. It is ideal to go for the automation once the critical functionalities or features of the application have been stable and frozen. Continuous change in these critical functionalities will require rework in the automation scripts, affecting the time and effort of the testers.
- A thorough manual testing of the application is required before it is moved to the automation. At time absence of manual testing or investing less time in manual testing adversely affect the time invested in automating the things. The same issue was faced while automating OHMS. Many a time while creating scripts, there were few critical bugs which were found. As a result, creating scripts for those particular functionalities was getting on hold until the issue is fixed hence impacting the time frame which has been assigned for automation.
- Currently testing team is not using any framework like TestNG for maintenance of test scripts and for reporting purpose. Simple console has been used to check the test cases status. So, it is recommended to use framework like TestNG which would allow execution of test cases in batch, parallel execution and giving a user-friendly reporting experience.
- Current project has not used any defect reporting and tracking tool. Though currently the team size is very small so it is still easy to keep track of the defects but once the application reaches the complexity and the team size also increase it will become difficult to keep track of all the defects which has been raised and are in open state, hold state, seeking further

clarification or which are needed to be closed on priority. Therefore, the organization need to understand the importance of using bug reporting and tracking tool and should start using the same as early as possible.

- For few of the field while entering invalid entries it was observed that the application is not throwing any error message which would alert the user where they went wrong. Taking an example, for the field where user need to enter the date of birth, there is a pre-defined pattern (DD/MM/YYYY) in which date need to be entered but in case user try to enter date in (MM/DD/YYYY) it is not getting entered and at the same time not displaying any error message which would help the user to know the pattern in which date need to be entered. So, the team needs to work little more on making the software more user-friendly.

7. CONCLUSION

This project helped to create regression automation suite for OHMS successfully using open source tools. It also helped the team to reduce the manual efforts and time which was required to run same test cases again and again whenever a new feature or any modification is done to the software and when new client need to install this software in their premises. By creating automation regression suite, now team can validate if the application is working fine with just one click .It also helped the team to ensure if any major or minor changes which has been made at later stages due to client requirement has impacted the critical functionality of the application. This report has also highlighted few issues which have been faced during the testing and the alternatives taken by the team to tackle these bottlenecks. For issues related to the database constraints, testing team came up with the idea of creating deleting mechanism, which would run automatically once all the entries have been created. The current research will provide guidelines to the future researchers on how to create an automated regression suite for any web application with the help of open source tools namely Visual Studio, Selenium Webdriver and Trello.

REFERENCES

- [1] Shai Fine, S. U. (2004). Probabilistic regression suites for functional verification. In Proceedings of the 41st annual Design Automation Conference (DAC '04) (pp. 49-54). New York, NY, USA: ACM.
- [2] Dudekula Mohammad Rafi, K. R. (2012). Benefits and limitations of automated software testing: systematic literature review and practitioner survey. In Proceedings of the 7th International Workshop on Automation of Software Test (AST '12) (pp. 36-42). USA: IEEE Press, Piscataway.
- [3] Stanislav Stresnjak, Z. H. (2011). Usage of Robot Framework in Automation of Functional Test Regression. The Sixth International Conference on Software Engineering Advances.
- [4] Prakash, K. S. (2017). A Comparative Study of Testing Framework with Special Emphasis on selenium for financial applications. International Journal of soft computing , 148-155.
- [5] Elior Vila, G. N. (2017). Automation Testing Framework for Web Applications with Selenium WebDriver: Opportunities and Threats. In Proceedings of the International Conference on Advances in Image Processing (ICAIP 2017) (pp. 144-150). New York, NY, USA: ACM.
- [6] Jamshaid G. Mohebzada, G. R. (2011). SRP-plugin: a strategic release planning plug-in for visual studio 2010. Proceeding of the 1st workshop on Developing tools as plug-ins - TOPI '11 (pp. 36-39). New York, NY, USA: ACM.
- [7] Johnson, H. A. (2017). "Trello.". Journal of the Medical Library Association : JMLA , 209–211.

- [8] M. Mahalakshmi, D. M. (2013). Traditional SDLC Vs Scrum Methodology – A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering* , 3(6).
- [9] Maurizio Leotta, D. C. (2013). Comparing the maintainability of selenium WebDriver test suites employing different locators: a case study. In *Proceedings of the 2013 International Workshop on Joining AcadeMiA and Industry Contributions to testing Automation (JAMAICA 2013)* (pp. 53-58). New York, NY, USA: ACM.

AUTHORS

I am **Anjali Rawat** and a QA Analyst. Have worked with Wipro Limited, India where I get an opportunity to work with clients like Vodafone, Cisco and Bharti Airtel. I had experience in database testing, regression testing, manual testing and automation testing. I have recently completed my graduate diploma in software testing from AGI Education Limited and now currently working as a QA lead in Born Digital, New Zealand. I am passionate about testing and love to deliver quality product to the customers.



Dr. Shahid Ali is a senior lecturer and IT program leader at AGI Education Limited, Auckland, New Zealand. He has published number of research papers on ensemble learning. His expertise and research interests include ensemble learning, machine learning, data mining and knowledge discovery.