A STUDY OF REGRESSION TESTING FOR TRADE ME WEBSITE

Kenil Manishkumar Patel and Shahid Ali

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

ABSTRACT

Regression testing plays a critical role to verify the functionality testing of a product. Trade Me is New Zealand based website. It is one of the major websites in New Zealand dealing in buying and selling online. The aim of this research is to find out the functionalities of Trade Me website after injecting new features. Automation regression suite is used to execute test scripts which helped the company to save time and cost compared to manual testing. Automation regression test suite also helped to prioritize test cases are designed in such a way that it can maximize the fault detection. For research analysis scrum methodology is used to meet the ultimate desires of software development companies and to increase the client satisfaction.

KEYWORDS

Regression testing, automation testing, scrum methodology, testNG, selenium

1. INTRODUCTION

Trade Me is New Zealand based website. It is one of the major internet sale websites working in New Zealand. Trade Me was founded by the New Zealand tycoon Sam Morgan in 1999. In 2015 according to Alexa Internet Trade Me was fifth most visited website in New Zealand. In a country having 5 million populations Trade Me website has 3.7 million active members and approximately eight hundred thousand visit websites on daily basis. Trade Me provides broad range of categories of new and used items. Trade Me has increased its scope over time and Trade Me now offers a wide range of listing possibilities general items, jobs, Motors, Rental property, pets and animals, Antiques & Collectables, Flatmates wanted.

The main scope and objectives of this research study is to create regression test suite and automating the test suite for testing the functionalities of Trade Me is working fine after the modification or not. This is designed for checking login, search, jobs, sort, add to watch list, remove from the watch list, logout functionalities. Trade Me have enormous number of features and is a fast-growing website approximately eight hundred thousand people are using this website on daily basis for buying and selling things. As Trade Me is a quick growing website new functionality are getting implemented frequently so to test the old functionalities are affected because of adding new functionalities we do automation regression suite.

Trade Me is rapid growing website having wide range of categories of new and used products for the vendors. As Trade Me is rapidly growing website, new functionalities are added frequently and test the old functionalities are not affected while addition of new functionalities, we do regression testing. The problem found in the Trade Me website is manual regression testing on Trade Me website is taking too long and it is monotonous job. Manual regression testing will take more time compared to automation regression suite. Because Trade Me includes much

David C. Wyld et al. (Eds): SPTM, IPPR, CSIT, BDAP - 2021
pp. 61-75, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110505

functionality, retesting all the functionalities after a change is time consuming. While in automation, creation of test scripts takes time, but we can run it efficiently in no time. So, my goal is to automate the regression suite for Trade Me website. As full automation of Trade Me website is not possible, the main functionalities are prioritized. This will not only help to decrease the testing time for every release, but it will also help to free up the manual test resources for quality and exploratory testing.

This research paper is organized as follow: Section 2 focuses on the literature review of the automation regression testing. Section 3 is focused on the research methodology for this research. Section 4 explains execution for this research and results are discussed in section 5. Section 6 provides the discussion on the results of this research. In section 7 recommendations for future research is provided. Finally, in section 8 conclusion to the research is provided.

2. LITERATURE REVIEW

Various research papers have been referred and analysed before implementing the methodology or using any tool and technique for this research. Test case prioritization was conducted which involves scheduling test cases in order to achieve effectiveness of performance goals [1]. In this study they mentioned that Test cases should be performed in such a manner that enhances the possibility of fault detection. In this study they build up validate requirement-based system level which is used for test case prioritization method to expose more harsh faults at the beginning stage.

Another study was conducted which showed regression testing is performed to give assurance that the alterations which are made in the existing software do not harm the live behaviour of the software [2]. In this study they mentioned that if the test suites tend to grow as software progresses, so it becomes too expensive to accomplish the entire test suites. In this study regression testing is also used for prioritizing the test cases in such a way that it maximized the fault detection in the earlier stage.

In a study it was determined that regression testing is to evaluate the adjustments made to software while adding new features have not been affected to the software [3]. In this study they revealed many techniques which have been reported on how to choose the regression tests so that quantity of test cases does not expand up too big as the software grows. In this study they projected hybrid technique combines minimization, modification, and prioritization-based variety using a list of source code changes and carrying out traces from test cases run on earlier versions.

Test case Prioritization techniques plan test cases study was conducted for implementation in order that attempt to make best use of some objective function. There are various objective functions are appropriate, one such function involves pace of fault detection a measure of how speedily faults are detected within the testing process [4]. In this study they stated about how improved pace of error detection during regression testing can give quicker feedback on a system under regression test and let debuggers start their work in advance. In this study they forecast several techniques for prioritizing test cases and report the efficiency of this techniques for improving pace of fault detection.

A study was conducted on test case prioritization for continuous regression testing is limited by fixed time constraints. To ensure time constraints and accomplish test cases, testing goals must be well planned in execution [5]. Prioritization techniques are usually used to command test cases to mirror their significance according to one or more criteria. High fault detection or reduced time test are vital part. In this study they proposed test prioritization approach Rocket to get better effectiveness of continuous regression testing of industrial video conferencing software. The outcome of the study shows that using Rocket it provides quicker fault detection, it also improves regression fault detection rate. It also reveals that "30% more faults for 20% of the test suite executed" comparing to manually prioritized test cases.

A further study was conducted on test case prioritization for web service regression testing is vital to confirm the worth of service-oriented business application in their evolutions [6]. In this study they mentioned how the test case prioritization technique plays vital role to boost the effectiveness of web service application regression testing. In this study they analyse the reliance relationship using control and data flow information in an orchestration language: WS-BPEL. They also prioritize the test cases according to wrap more alteration affected elements with the maximum weight.

A study was conducted to comprehend regression testing techniques which are most vital phase in the software development life cycle [7]. In the maintenance phase of software development life cycle the development team maintains the software and delivered to clients. In this study they mentioned about the software maintenance findings for the reasons like enhancement of capabilities, deletion of capabilities, error corrections and optimization. In this study they have presented a variety of regression testing techniques and their classifications offered by different researchers and elaborating selective and prioritizing test cases for regression testing in brief.

A research was conducted to test prioritization using system model's regression testing is important to check customized system is retested using existing test suite. The size of the test suite may be very huge, and testers are fascinated in defecting the faults in the earlier stage during the retesting process [8]. In this study they mentioned about the existing prioritization methods which are totally based on the code of the system. In this study they have introduced variety of methods of test case prioritization which are based on state-based models after changes to the model and the system. The model is implemented for the test suite and information about model implementation is used to prioritize tests.

A further research was conducted on test case prioritization approaches in regression testing, a systematic literature review states that quality of the software can be assured by going through software testing process [9]. In this study they talked about how regression testing phase is a costly process as it consumes a longer time. In this study they come up with the solution by scheduling test cases implementation with the help of prioritization approach.

Another research was conducted on prioritization for regression testing using ant colony optimization based on test factors, regression testing is believed to be one of the most expensive, time-taking, and vital activity which is executed in an environment with number of limitations for confirming the strength of a modified software [10]. In this study they talked about why it is necessary to select the sequence of test cases which is accurate, and it can traverse all the faults and take minimum execution time. In this study they come up with the solution that prioritization supports to accomplish performance requirements in which only the vital test cases are implemented. Previous studies have shown that prioritization approaches based on test factors like complexity, fault rate, importance, volatility, time, coverage have good outcomes and it also approaches based on intelligent practices like ant colony, genetic algorithm have been encouraging.

After going through various research on various aspects of regression testing and test prioritisation our aim for this research is to find whether the functionalities of the Trade Me website is working fine or not after inserting new features and for this we need to use certain tools which are discussed in next section.

3. RESEARCH METHODOLOGY

Selenium Web driver is used to build this automation framework. In this research automation framework for a web application has been designed and implemented. Using these framework testers can easily write and execute the test cases effectively and in less span of time. This framework is supportive to developer as well as tester to analyse their code and it produces the customized report to tester [11]. For browser testing we can use selenium because it is an open source tool [12]. Selenium web driver has been selected for this research and why it is selected it is shown below in table 1.

Table 1: Comparisons of Tools

Parameter	Selenium Webdriver	QTP	Test Complete
Programming Language	Scripts are normally designed using ruby, java, python, Perl, and PHP.	It only supports VBScript and JavaScript**.**	It Allows scripts to be designed in C#, C++, VBScript, JS Script.
Platform Supported	MAC, UNIX, Windows operating system.	It only supports windows.	It runs on windows Vista and 7.
Open source/ Paid Tool	Open Source	It is a paid tool, and it is based on the project.	It cost lesser than QTP and it is licensed tool.
Test Execution Report	It does not generate the report by itself we need plugin like TestNG to execute and generate reports in html format.	Using QTP we can generate test execution report and determine the execution status.	Execution results are generated as a separate file and displayed to the user [13].

For this research scrum methodology is selected because of its nonstop iteration for the development and testing throughout the life cycle of the software development process. The Sprint last for 2 weeks. The team member has to co-ordinate with the scrum master for the daily report via email or face to face meeting when needed. The scrum master reviews the work and provides suggestion. At the end of the Sprint there is retrospective meeting to see development and testing progress.

Scrum methodology has been opted for this research. While using traditional SDLC are not able to meet the market requirement and leading to customer dissatisfaction, as customer requirements are changing frequently and making it more complex for the testers. Scrum methodology has been introduced to meet the desires of the software development companies like Trade Me [14].

4. RESEARCH EXECUTION

4.1. Automation Test Plan

It is time consuming to write the automation scripts. To automate everything is not possible so spending money and time on automation is to develop a strategy that boost up the velocity for the short and long term.

Repetitive Test

- As we know login is repetitive test because we are calling login several times on the website.

High Risk

- Auto-updating of the web-browser is high risk.
- Using X-path sometimes creates a lot of error because it does not call the expected value.

Test Approach

- Selecting the appropriate automated testing tool for testing Trade Me website.
- Test cases should be given test priority, which test case has high priority should be automated first, test cases having low priority should be tested at the end.
- Test early and Test often.

Test automation phases

- Requirement analysis: - Functional requirements of Trade Me.
- Test Planning: - Functional Automation planning.
- Test case design and development: - Planning test cases and prioritizing the test cases according to the priority (High, Medium, Low).
- Test environment: - Installing Selenium, Xampp, TestNG for generating report and Eclipse.
- Test execution: - Scripts Implementation.

Roles and Responsibilities

Table 2: Roles and Responsibilities

Members	Task	Time
1.Automation Tester	For writing automation scripts.	6 hours
2.Manual Tester	For writing the test cases	5 Hours

Test Environment

- **Software**

1. Installing Selenium.
2. Installing Xampp.
3. TestNG for generating report.
4. Eclipse.
5. Database Server.
6. Test Data.

- **Hardware**

1. CPU 1.60 GHZ.
2. Operating System: 64 Bit Windows 10
3. Ram: 12 GB

Assumptions

- Testing environment stability.
- Test resources available.
- Stable application.

Risks

- Auto updating of the web-browser is high risk.
- Using X-path sometimes creates a lot of errors because it does not call the expected value.

4.2. Proposed Architecture for Automation Test Plan

The figure above justifies the architecture of automation test plan. There are 8-Page Factory Classes including login and logout which are connected to the test cases individually and that test cases are connected to the TestNG class. TestNG class is further divided in to two different parts. One TestNG class is connected with Selenium Web-Driver, Chrome Browser and test under execution. The second part is connected with the TestNG framework to generate reports and generate logs.

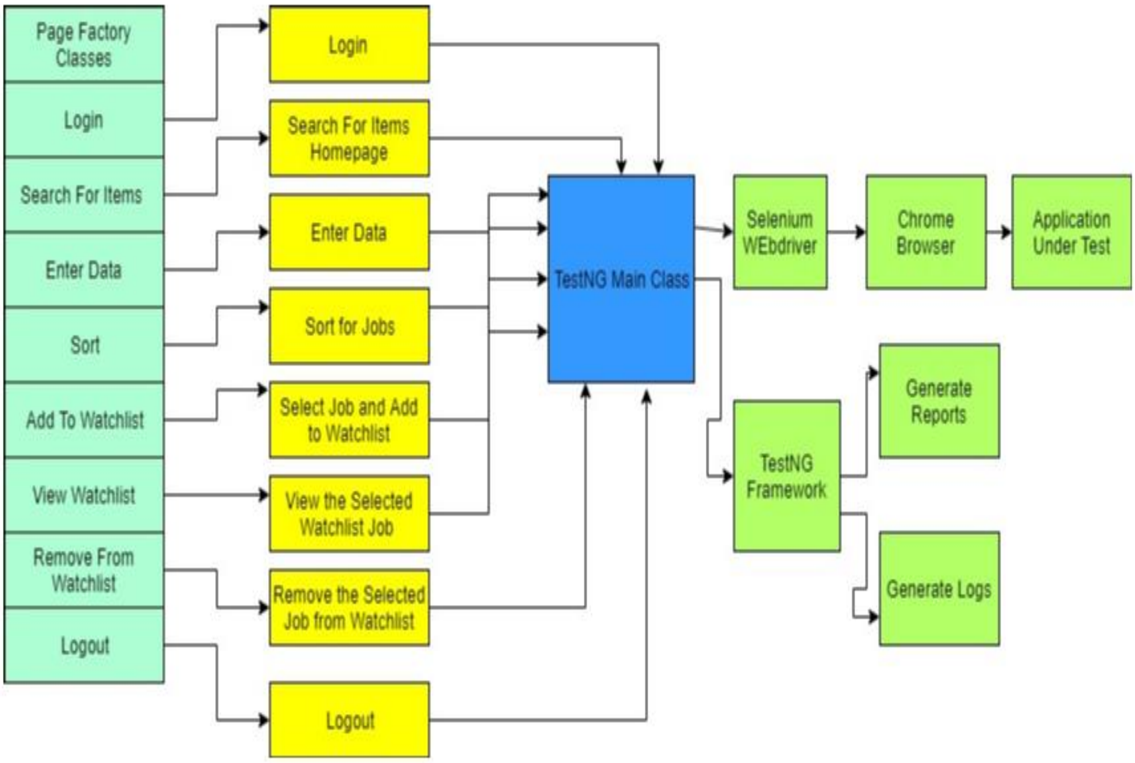


Figure 1: Architecture of Automation Test Plan

4.3. Functional Test Cases

The functional test cases are prioritized as per the requirement of research. Regression automation testing for the whole Trade Me website is not possible. So, the aim of the research is to prioritize the test cases which are listed below in table 3.

Table 3: Functional Automation Testing

Test Scenarios ID	Test Scenarios	Test Steps	Expected Results	Actual Results	Pass/ Fail
TC-01	Verify successful login functionality.	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx" 2. Enter valid "Username". 3. Enter valid "password". 4. Click on "Login Button".	User should be able to see the homepage.	User has been logged in successfully.	Pass
TC-02	Verify that user can search for the items.	1.Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button".	User should be able to see the "Trade Me Jobs HomePage".	User can see the Trade Me jobs homepage.	Pass
TC-03	Verify that user can enter Keywords sucessfully.	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs.	User should be able too enter the valid details.	User can enter the details.	Pass
TC-04	Verify that user can select "Sort".	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs. 12. Select "Featured Jobs".	User should be able to see the Featured jobs.	User can see the featured jobs.	Pass

TC-05	Verify that user can "Add to Watchlist".	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs. 12. Select "Featured Jobs". 13. Click on "Add to Watchlist".	User should be able to add jobs to the watchlist.	User can add the jobs to the watchlist.	Pass
TC-06	Verify that user can "View the Watchlist".	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs. 12. Select "Featured Jobs". 13. Click on "Add to Watchlist". 14. Click on "View Watchlist".	User should be able to view the watchlist.	User can view the watchlist.	Pass
TC-07	Verify that user can "Remove from the Watchlist".	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs. 12. Select "Featured Jobs". 13. Click on "Add to Watchlist". 14. Click on "View Watchlist". 15. Click on "Remove from Watchlist".	User should be able to Remove from the watchlist.	User can remove jobs from the watchlist.	Pass

TC-08	Verify that user can "Logout Sucessfully".	1. Open browser enter "https://www.trademe.co.nz/Members/Login.aspx". 2. Enter valid credentials for login. 3.Search for "Jobs". 4. Click on "Search Button". 5.Enter "Keywords". 6. Select "Job Type". 7. Select "Region". 8. Select "categories". 9. Select "Annual Income". 10. Select "Any Experience". 11.Search for the jobs. 12. Select "Featured Jobs". 13. Click on "Add to Watchlist". 14. Click on "View Watchlist". 15. Click on "Remove from Watchlist". 16. Click on "Logout".	Verify that user has logged out successfully.	User has been logged out successfully.	Pass

4.4. Gantt Chart for research

The Gantt chart used for this research is given in below Table 4. The Gantt chart in Table 4 explains brief about the Sprint 0 and Sprint 1. It explains activities from day 1 to day 25. The Gantt chart also have the start and the end date so we can get the clear idea about the Sprints and have the duration that which Sprint Activity took more hours to complete.

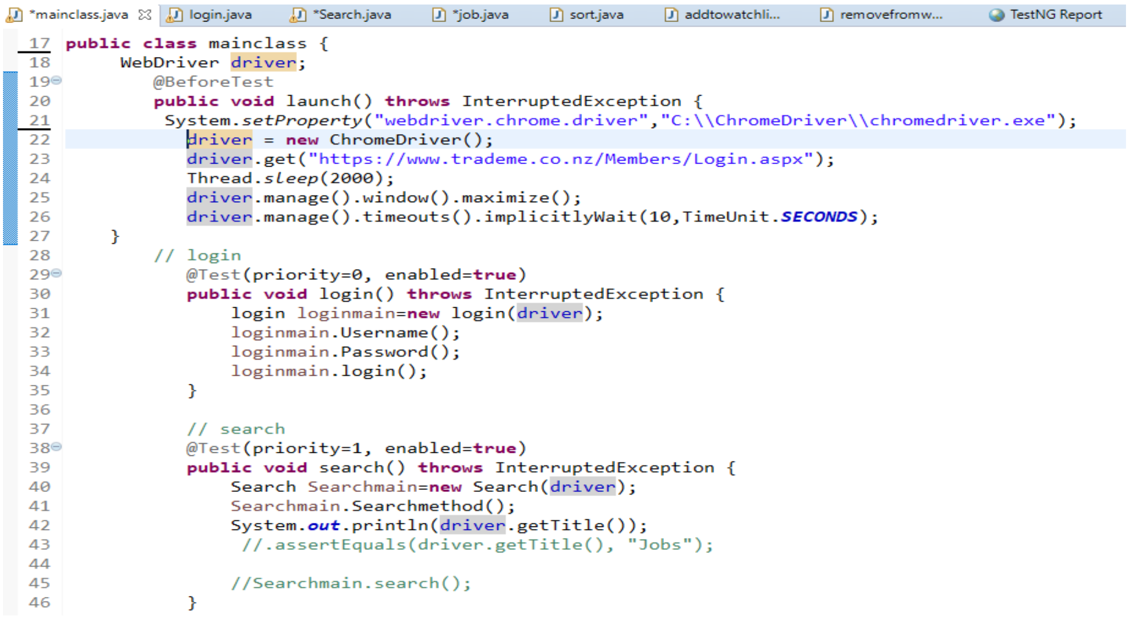
Table 4: Gantt chart for research

Sprint	Actions/Deliverables	Day	Start date	End Date	Durations
Sprint 0	Selecting and understanding the research.	1	19/08/19	19/08/19	7
	Understanding the plan of the proposal.	2	20/08/19	20/08/19	6
	Collecting the information related to proposal.	3	21/08/19	21/08/19	5
	Create plan for the proposal and check the literature for proposal.	4	22/08/19	22/08/19	6
	Meeting supervisor for the proposal feedback and changes made according to the guidance of the supervisor.	5	23/08/19	23/08/19	6
	Installing and setting up the environment.	6	26/08/19	26/08/19	6
	Analysing the application.	7	27/08/19	27/08/19	6
	Analyse the technical architecture and design.	8	28/08/19	28/08/19	6
	According to business requirements create a test plan.	9	29/08/19	29/08/19	5
	Creating scenarios for regression testing.	10	30/08/19	30/08/19	7
Sprint 1	Creating scripts for regression testing.	11	02/09/19	02/09/19	6
	Scripts execution	12	03/09/19	03/09/19	8
	Generate reports	13	04/09/19	04/09/19	8
	Meeting supervisor for the research report.	14	05/09/19	05/09/19	4
	Changes can be made according to the	15	06/09/19	06/09/19	4

	guidance of supervisor.				
Sprint 2	Gathering information for final report.	16	09/09/18	10/09/18	10
	Preparing the final report for the research.	18	11/09/18	16/08/19	26
	Preparing the final presentation.	22	17/09/19	17/09/19	8
	Research monitoring.	23	18/09/19	18/09/19	6
	Meeting supervisor for the feedback about the report and presentation.	24	19/09/19	19/09/19	4
	Final changes made as per the requirement of research supervisor.	25	20/09/19	20/09/19	4

4.5. Snippets of code

The code snippets for the main class is displayed below in figure 2. @Test (annotations) in the figure states that all the test scenarios are implemented in a chronological way. (Priority=0) states the test case priority that login test case should be executed first. In the Main TestNG class sub-methods are created.



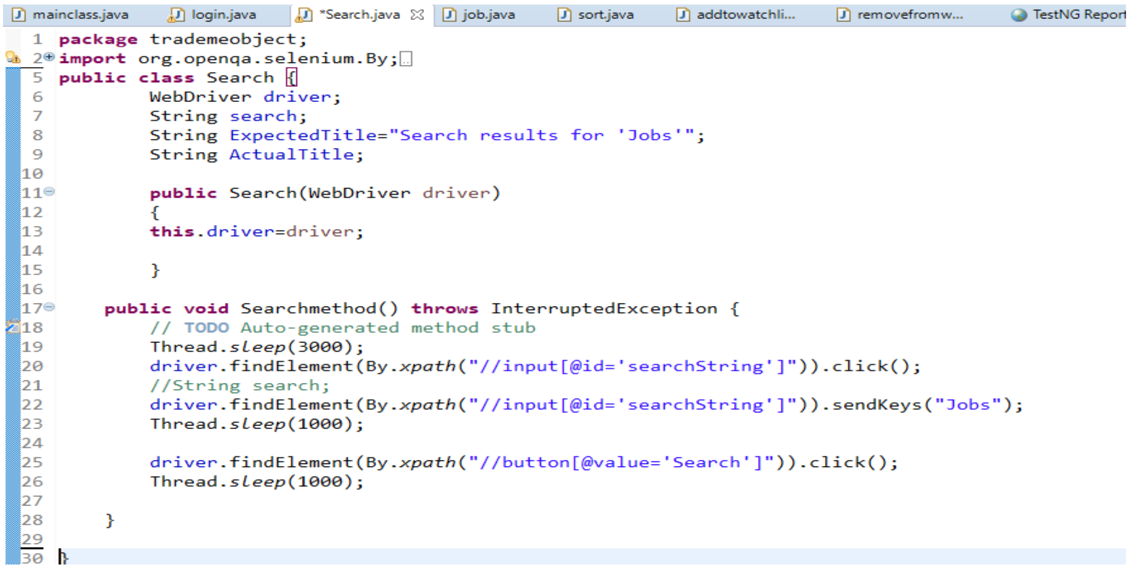
```java
public class mainclass {
     WebDriver driver;
        @BeforeTest
        public void launch() throws InterruptedException {
         System.setProperty("webdriver.chrome.driver","C:\\ChromeDriver\\chromedriver.exe");
           driver = new ChromeDriver();
           driver.get("https://www.trademe.co.nz/Members/Login.aspx");
           Thread.sleep(2000);
           driver.manage().window().maximize();
           driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
    }
        // login
           @Test(priority=0, enabled=true)
           public void login() throws InterruptedException {
               login loginmain=new login(driver);
               loginmain.Username();
               loginmain.Password();
               loginmain.login();
           }

           // search
           @Test(priority=1, enabled=true)
           public void search() throws InterruptedException {
               Search Searchmain=new Search(driver);
               Searchmain.Searchmethod();
               System.out.println(driver.getTitle());
                //.assertEquals(driver.getTitle(), "Jobs");

               //Searchmain.search();
           }
```

Figure 2: Snippets of Main class

The code snippets for search test suite are show in Figure 3. The Figure 3 below defines methods in each method and defined the actions for each method to be performed.



```
package trademeobject;
import org.openqa.selenium.By;
public class Search {
        WebDriver driver;
        String search;
        String ExpectedTitle="Search results for 'Jobs'";
        String ActualTitle;

        public Search(WebDriver driver)
        {
        this.driver=driver;

        }

    public void Searchmethod() throws InterruptedException {
        // TODO Auto-generated method stub
        Thread.sleep(3000);
        driver.findElement(By.xpath("//input[@id='searchString']")).click();
        //String search;
        driver.findElement(By.xpath("//input[@id='searchString']")).sendKeys("Jobs");
        Thread.sleep(1000);

        driver.findElement(By.xpath("//button[@value='Search']")).click();
        Thread.sleep(1000);

    }

}
```

Figure 3: Snippets of Search Class

5. RESULTS

Research results of this study are given below.

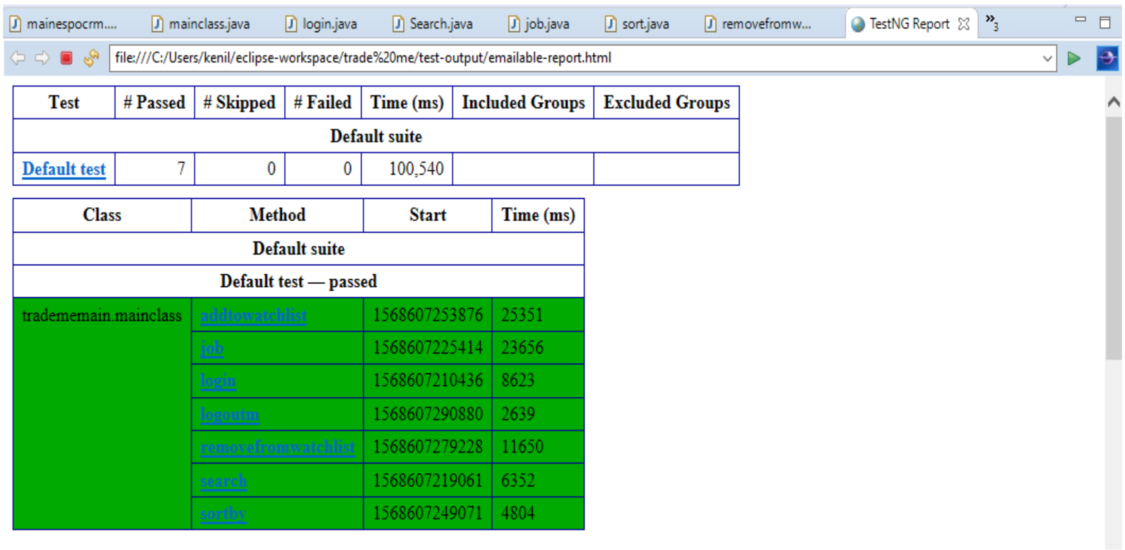
5.1. TestNG Reports

TestNG is used to create testing reports and tester can easily come to know how many test are passed, failed and skipped. TestNG is an automation Framework where NG stands for next generation. TestNG is motivated from Junit which uses the annotations (@). Using TestNG tester can also execute the failed test cases separately. Also, TestNG provides a separate option " testing-failed.xml file" in the test output folder. So that if tester wants to run the failed test cases it can run with the help of XML file.

TestNG is used to capture the metrics which is used to improve the effectiveness and efficiency of a software testing process. The below points show how to capture TestNG report.

- TestNG will generate default report.
- After executing the java project should be refreshed to get test-output folder.
- Right clicking on the emailable-report.html.
- Same output can be seen in the web browser.

The Figure 4 shows that how many test cases are passed, skipped, failed. It also shows time taken for one test to complete. The Figure 4 defines that there are 7 number of test cases have been executed successfully. The number of test cases failed and skipped are 0.



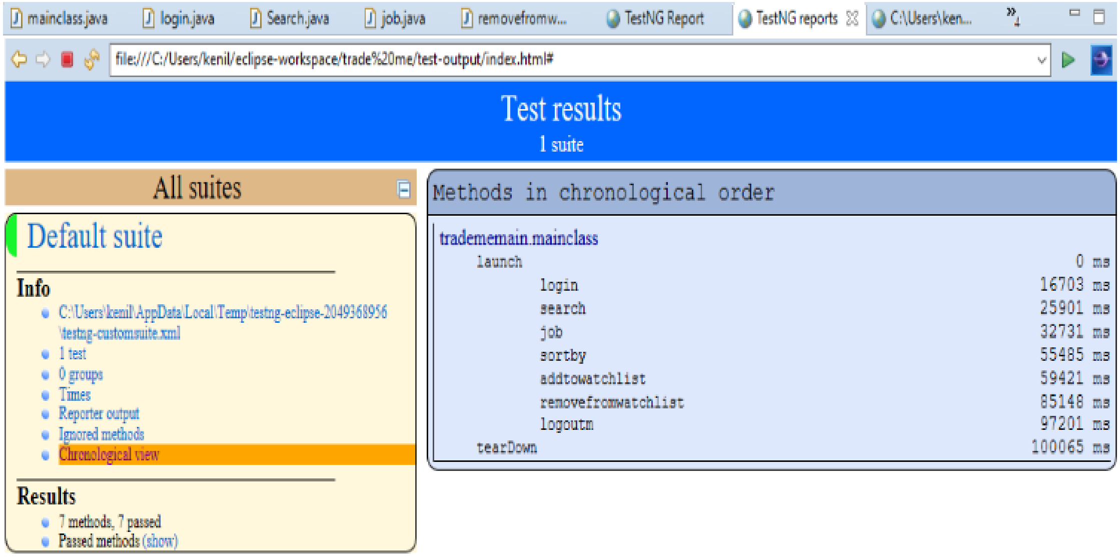
Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite						
Default test	7	0	0	100,540		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
tradememain.mainclass	addtowatchlist	1568607253876	25351
	job	1568607225414	23656
	login	1568607210436	8623
	logoutm	1568607290880	2639
	removefromwatchlist	1568607279228	11650
	search	1568607219061	6352
	sortby	1568607249071	4804

Figure 4: TestNG Reports

5.2. Chronological view

The chronological view of Trade Me main class is shown in the below Figure 5. The Figure 5 shows chronological view order of the test cases and what is the flow of test case. The flow of Trade Me main class is shown in the Figure 5.



Test results
1 suite

All suites

Default suite

Info
- C:\Users\kenil\AppData\Local\Temp\testng-eclipse-2049368956\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results
- 7 methods, 7 passed
- Passed methods (show)

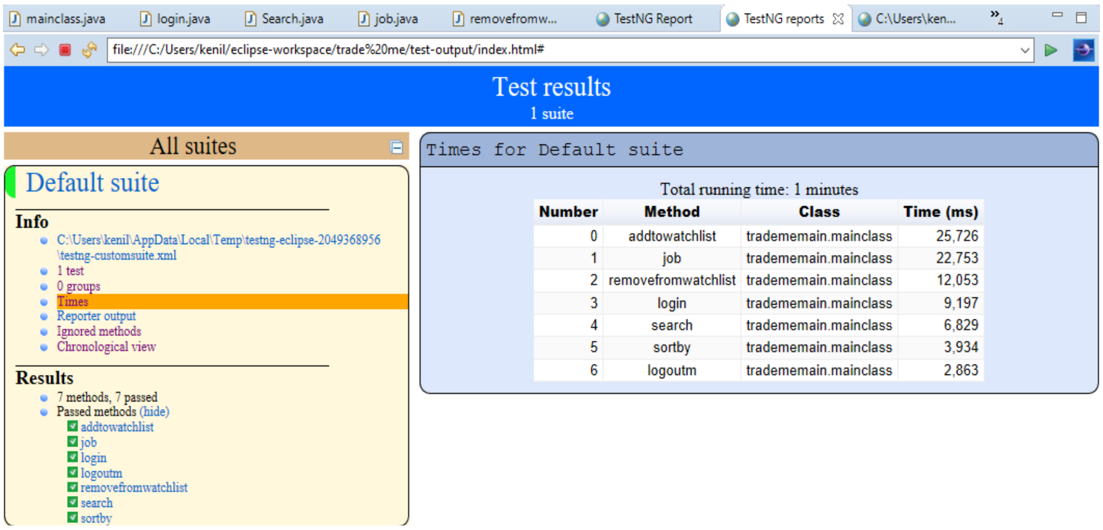
Methods in chronological order

```
tradememain.mainclass
    launch                                  0 ms
        login                           16703 ms
        search                          25901 ms
        job                             32731 ms
        sortby                          55485 ms
        addtowatchlist                  59421 ms
        removefromwatchlist             85148 ms
        logoutm                         97201 ms
    tearDown                           100065 ms
```

Figure 5: Chronological View

5.3. Times

The Figure 6 shows time taken for each test to passed. It shows how much time taken by one test to get executed. Add to watchlist has taken 25,726(MS) approximately which is highest to execute one test. And logout took approximately 2863(MS) which least time to execute the test case.



Test results

1 suite

All suites

Default suite

Info

- C:\Users\kenil\AppData\Local\Temp\testng-eclipse-2049368956 \testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 7 methods, 7 passed
- Passed methods (hide)
  - ☑ addtowatchlist
  - ☑ job
  - ☑ login
  - ☑ logoutm
  - ☑ removefromwatchlist
  - ☑ search
  - ☑ sortby

Times for Default suite

Total running time: 1 minutes

Number	Method	Class	Time (ms)
0	addtowatchlist	tradememain.mainclass	25,726
1	job	tradememain.mainclass	22,753
2	removefromwatchlist	tradememain.mainclass	12,053
3	login	tradememain.mainclass	9,197
4	search	tradememain.mainclass	6,829
5	sortby	tradememain.mainclass	3,934
6	logoutm	tradememain.mainclass	2,863

Figure 6: Times

6. Discussion

The result shown in Figure 4 defines that, all the test scenarios has been passed. But what happens when one of the test cases failed? Let us take test scenarios search for the job has been failed. So, it will not have allowed me to do further regression testing on different scenarios. To overcome this challenge, we need to fix the error and run the test successfully. Using TestNG tester can run the failed test cases individually. TestNG also provides separate option like " testing-failed.xml file" in the test output folder. So that tester can run the failed test cases rather than running whole scripts which will help to reduce time. Let's suppose what heppens if the test cases got skipped? Sometimes QA team needs to execute the last minute issue verification at that moment QA team does not have enough time to execute all the test cases so in this case QA team can skipped the test case by @Test(enabled=false).

The Chronological view of Trade Me website is shown in Figure 5. Chronological means in a specify order. The Figure shows the test suites in step by step order. What if it is not in chronological order? If the test suites are not in order the test cases will be failed. Test suites should be in a specify order to execute the test cases.

The time taken by each test to execute successfully is shown in the Figure 6. Add to watch list has taken 25,726 (MS) to execute the test which is highest and logout has taken least amount of time to execute the test which is 2,863 (MS). Now, what happens if the test case is taking too long or test cases get success? We can just simply add wait time between the element to get the accurate result and after the execution of all test cases we can simply decrease the wait time between the element so that we don't have to wait everytime.

7. Recommendations

In this research regression automation test suite has been used to overcome the problems. First problem faced was automating the whole Trade Me website is not possible because it has lot of scenarios. To overcome this problem came up with the solution of prioritizing test cases. Prioritizing test cases helped to reduce the time and it also helped in reducing the cost. Second challenge faced during the execution phase. As, Trade Me is dynamic website we cannot automate with the help of X-path because it changes after every few minutes. To overcome this problem ID and Name is used to identify the correct path. Regression automation test suite helped this research to reduce the time and it also helped to reduce the cost. It also helped to free up the manual test resources for quality and exploratory testing.

8. CONCLUSION

In this research we have covered the plan and the problem of the Trade Me website. For this research selenium Webdriver was selected and it helped to test the new features and processes of Trade Me website. I successfully implemented Regression testing which helped me to prioritize the test cases in such a way that it maximized the fault detection. Scrum methodology is used to meet the desires of the software development companies and increase the client satisfaction. The automation test plan, functional automation test cases, sprints, architecture of automation test plan explains the workflow in detail about the research. Regression testing is the best technique, but we cannot automate the "Trade Me" website fully. So, we prioritized the test cases and automate. This research will also help in future for similar kind of fast-growing websites.

REFERENCES

[1] Raju, S., & Uma, G. V. (2012). Factors oriented test case prioritization technique in regression testing using genetic algorithm. European Journal of Scientific Research, 74(3), 389-402.

[2] Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. Software Testing, Verification and Reliability, 22(2), 67-120.

[3] Wong, W. E., Horgan, J. R., London, S., & Agrawal, H. (1997, November). A study of effective regression testing in practice. In PROCEEDINGS The Eighth International Symposium On Software Reliability Engineering (pp. 264-274). IEEE.

[4] Rothermel, G., Untch, R. H., Chu, C., & Harrold, M. J. (1999, August). Test case prioritization: An empirical study. In Proceedings IEEE International Conference on Software Maintenance-1999 (ICSM'99).'Software Maintenance for Business Change'(Cat. No. 99CB36360) (pp. 179-188). IEEE.

[5] Marijan, D., Gotlieb, A., & Sen, S. (2013, September). Test case prioritization for continuous regression testing: An industrial case study. In 2013 IEEE International Conference on Software Maintenance (pp. 540-543). IEEE.

[6] Chen, L., Wang, Z., Xu, L., Lu, H., & Xu, B. (2010, June). Test case prioritization for web service regression testing. In 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (pp. 173-178). IEEE.

[7] Duggal, G., & Suri, B. (2008, March). Understanding regression testing techniques. In Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology.

[8] Korel, B., Tahat, L. H., & Harman, M. (2005, September). Test prioritization using system models. In 21st IEEE International Conference on Software Maintenance (ICSM'05) (pp. 559-568). IEEE.

[9] Khatibsyarbini, M., Isa, M. A., Jawawi, D. N., &Tumeng, R. (2018). Test case prioritization approaches in regression testing: A systematic literature review. Information and Software Technology, 93, 74-93.

[10] Ahmad, S. F., Singh, D. K., &Suman, P. (2018). Prioritization for regression testing using ant colony optimization based on test factors. In Intelligent Communication, Control and Devices (pp. 1353-1360). Springer, Singapore.

[11] Gojare, S., Joshi, R., &Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. Procedia Computer Science, 50, 341-346.

[12] Holmes, A., & Kellogg, M. (2006, July). Automating functional tests using selenium. In AGILE 2006 (AGILE'06) (pp. 6-pp). IEEE.

[13] Jain, C. R., &Kaluri, R. (2015). Design of automation scripts execution application for selenium webdriver and test NG framework. ARPN J EngApplSci, 10, 2440-2445.

[14] Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC Vs Scrum Methodology–A Comparative Study. International Journal of Emerging Technology and Advanced Engineering, 3(6), 192-196.

AUTHORS

Kenil is a graduate of BE computer science engineering in India. He moved to New Zealand in 2018 to pursue his career in Graduate Diploma in Software Testing at AGI Education limited. He started exploring about testing concepts and practices. He has an interest in Functional automation testing as well as performance Testing.

Dr. Shahid Ali is IT program leader and senior lecturer 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.