UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Riana Randoja

Lab Package: Automated Web-Application Testing

Bachelor's thesis (9 ECTS)

Supervisor: Dietmar Pfahl

Tartu 2018

Lab Package: Automated Web-Application Testing

Abstract:

The main goal of this bachelor's thesis is to create lab materials about Automated Web-Application Testing for the course „Software Testing (MTAT.03.159)" in University of Tartu. Materials produced for this lab will be introduced, feedback gathered from students is analysed and there are suggestions for future improvements.

Keywords:

Software testing, lab package, Selenium, web-application

CERCS: P170 Computer science, numerical analysis, systems, control

Praktikumimaterjal: Veebirakenduse testimise automatiseerimine

Lühikokkuvõte:

Antud bakalaureusetöö põhieesmärgiks on praktikumimaterjalide loomine veebirakenduse testimise automatiseerimise kohta ainesse "Tarkvara testimine(MTAT.03.159)", mida loetakse Tartu Ülikoolis. Kirjeldatakse loodud materjale, analüüsitakse tudengite tagasisidet ja antakse ettepanekuid edaspidiseks arenduseks.

Võtmesõnad:

Tarkvara testimine, praktikumimaterjal, Selenium, veebirakendus

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Table of Contents

1. Introduction

Web applications are a big part of today's software world. They provide many high value services in different fields like communication, health care and finance. Thus, it's essential to test them and verify that they are functioning correctly. In web applications there are usually many functionalities and their development cycle is very fast. To make finding bugs quicker and eliminate manual work, it makes sense to automate the tests. Automated tests can be run multiple times, ensuring that an application's old functionalities are working after integrating new ones.

The main goal of this bachelor's thesis is to create lab materials that convey to students basic knowledge about automation of functional tests of web applications. This lab package is part of "Software Testing" course (MTAT.03.159) in University of Tartu.

Motivation to include this lab in the "Software Testing" course came from insufficient coverage about web application testing throughout the Bachelors studies in University of Tartu.

For this lab a new web application was created. The intention was to have control over all functionalities, to keep the application small, and to focus on testing. For writing functional tests, it was decided to use Selenium Webdriver, as it's easy to use and is commonly used by modern web application development teams.

The first chapter of this thesis gives the background information about web applications, their testing categories and possibilities are given. The second Chapter gives an overview of the materials and exercises created for this lab. Furthermore, lab flow and requirements for the homework assignment are explained. In the third Chapter, the analysis of feedback gathered from students after completing this lab and proposals for future improvements are presented. In appendix there are lab materials composed for this lab.

2. Background

Software testing helps to verify that a software application meets its requirements, is working as expected, and to find flaws and errors in the application code [1].

2.1 Web Application Testing

Web applications contain a lot of different functionalities and are evolving quickly. They have to run on multiple browsers and operating systems, which makes testing their full functionality manually inefficient and practically impossible in time matter.

As per DevOps Report [2] most of the high performing IT companies use Continuous Integration (CI) to deploy their application at least weekly, more often daily. CI is used to automate builds and to verify, that the program is functioning correctly after every change from developers. For that an automated test suite is required to run after every commit. CI makes finding bugs easier and quicker.

Web application testing consists of functional, usability, interface, compatibility, performance and security testing [3]:

- Functional testing - Ensures that application is performing as expected, all links are working, forms are validated. End to end workflow is tested, including negative scenarios, that meaning appropriate error messages show up if validation fails.
- Usability testing - It's intuitive to navigate on the page, links are easily findable and grammar is correct.
- Interface testing - Checks whether application, web and database servers are working together successfully. If any errors are caught, they must be shown to admin, not end user.
- Compatibility testing - Testing functionalities on different browsers and operating systems.
- Performance testing - Tests application with different connection speeds and determines how does the system behave under peak and normal loads.
- Security testing - Verifies, that restricted pages are not accessible without proper authorisation.

In this lab the focus is on functional testing. There are many tools and frameworks for automating functional tests. For example, Selenium, Katalon Studio, Unified Functional Testing,

TestComplete and Watir [4]. They all have their advantages and disadvantages. For example, TestComplete supports multiple scripting languages like: JavaScript, Python, C++, but works only on Windows computers. Katalon Studio on the other end supports Windows, Linux and Mac, but the scripts must be written in Java. Selenium supports multiple environments (Windows, Linux, Mac), many browsers, scripts can be written in various languages and it's free of charge. Although Selenium requires knowledge to set it up and the framework lacks document upload possibilities, it's overall the most flexible and commonly used [5]. Based on the arguments above it was decided to use Selenium in context of this lab.

2.2 Selenium

Selenium is an open source testing framework that focuses on web-based applications [6]. The framework consists of multiple components, for example Selenium IDE and Selenium Webdriver.

Selenium IDE

Selenium IDE makes it possible to record actions in the web browser and replay them. It doesn't require any previous testing experience, but the tests are easily broken if web application interface changes. Selenium IDE is implemented as Mozilla Firefox plugin and is supported only until version 55 [7].

Selenium Webdriver

Selenium Webdriver executes scripted tests. Writing them requires previous programming experience, but script code can be reused. It's possible to use conditional statements, loops and parameters. Webdriver test scripts can be written using several programming languages, like Ruby, C++, Java, PHP and the same tests can be run against most modern browsers like Mozilla Firefox, Google Chrome, Safari, Internet Explorer and HtmlUnit [8]. Selenium provides different libraries, that provide useful commands like clicking a button, filling fields and waiting for the browser to load the required page. Selenium Webdriver based test scripts are more flexible and easier to modify.

Selenium has many other tools like for example Selenium Grid, used to run tests in multiple machines, but as they are not important in the context of this lab, no more detailed descriptions are provided.

3. Lab Description

3.1 Materials produced for the lab

Links to available materials can be found in Appendix I: Lab Materials.

Web Application

For the purpose of this lab, a new web application was created. To make testing more interesting and meaningful, some bugs were intentionally inserted. Reason for creating a new application was getting control over all of the functionalities and intended bugs within the system.

The application itself is an online store. It has restricted number of functionalities to stay easily manageable and keep focus on testing.

Functionalities on the admin side:

1. Registering an account
2. Login
3. Logout
4. Delete your account
5. Add products
6. Edit products
7. Delete products

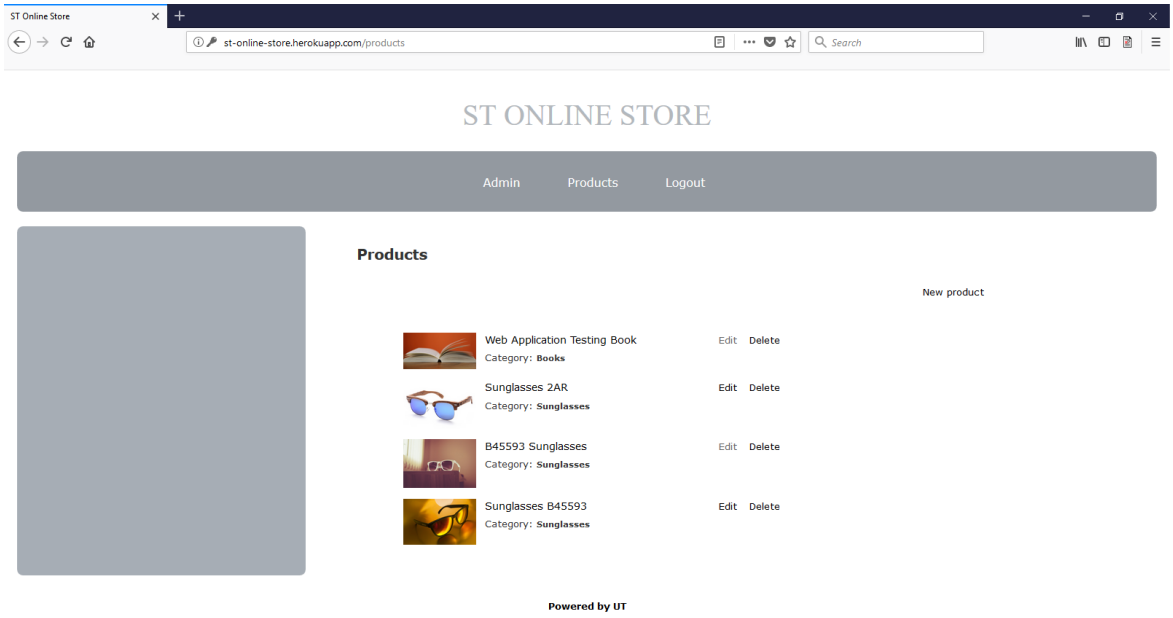


Figure 1. View from the admin side.

Figure 1 shows the page, where admin has a list of all the available products.

Functionalities on end-user side:

1. Add products to cart
2. Increase/decrease the quantity of a product in cart
3. Delete an item from the cart
4. Delete the entire cart
5. Search products by name and filter by categories
6. Purchase items

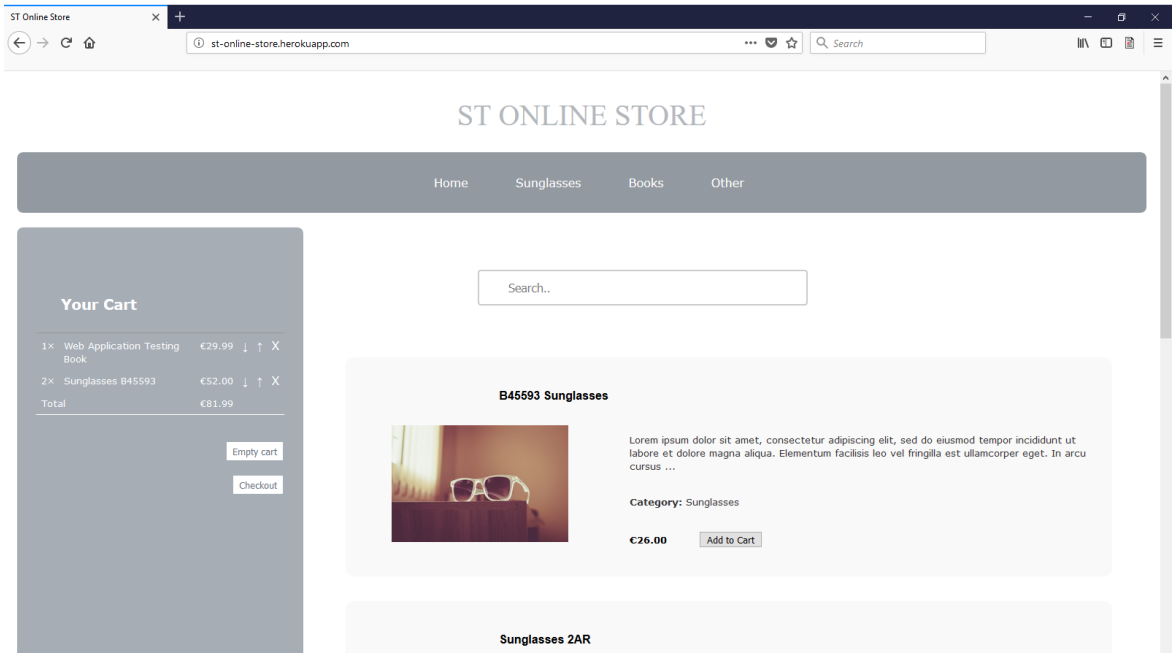


Figure 2. View from End-user side.

Figure 2 shows how listing of products on end user side appears. If there are products in cart, they appear on the left panel.

Student Instructions

Student Instructions introduce web application testing, Selenium, give instructions on how to set up the given web application and the Maven project used for testing. There are some useful tips about testing and Selenium overall, explanations for in-class and homework tasks, submission instructions and grading scheme.

Source Code for Testing

The source code is a Maven project, that already contains partially Selenium setup and some small examples, on how to structure tests. Its purpose is to give the students something to start with to make the using of Selenium easier.

```
@Before
public void setUp(){

    System.setProperty("webdriver.gecko.driver", "C:\\Users\\...\\geckodriver.exe");
    driver = new FirefoxDriver();

    driver.manage().timeouts().implicitlyWait( l: 10, TimeUnit.SECONDS);
    driver.get(baseUrl);

}

void goToPage(String page){
    WebElement elem = driver.findElement(By.linkText(page));
    elem.click();
    waitForElementById(page);
}
```

Figure 3. Selenium Webdriver usage example

```
void login(String username, String password){

    driver.get(baseUrlAdmin);

    driver.findElement(By.linkText("Login")).click();

    driver.findElement(By.id("name")).sendKeys(username);

    // ...

    By loginButtonXpath = By.xpath("//input[@value='Login']");
    // click on the button
    // ...
}
```

Figure 4. First in-class exercise

Figure 3. shows an example of a Selenium Webdriver setup method and a helper method for navigating to another page. On figure 4. is one part of the in-class exercise, that students have to fill in.

TA Instructions

TA Instructions contains everything that is included in Student Instructions. There are also listing of bugs inserted into the system, hints on what to point out in class and extended grading scheme.

Lab Assignment Solution

Example solutions for tests, that students should write during the practical session

Homework Solution

Example suite of test cases, that students should submit as their homework. More tests are included than it's required from/of students. Some tests have multiple possibilities and bug finding tests of all intentional bugs are provided.

Homework Test Results

HTML Test Results file lists names of all tests and shows, which of them passed and which not.

3.2 Lab Workflow

In this Chapter the flow of the Lab will be explained.

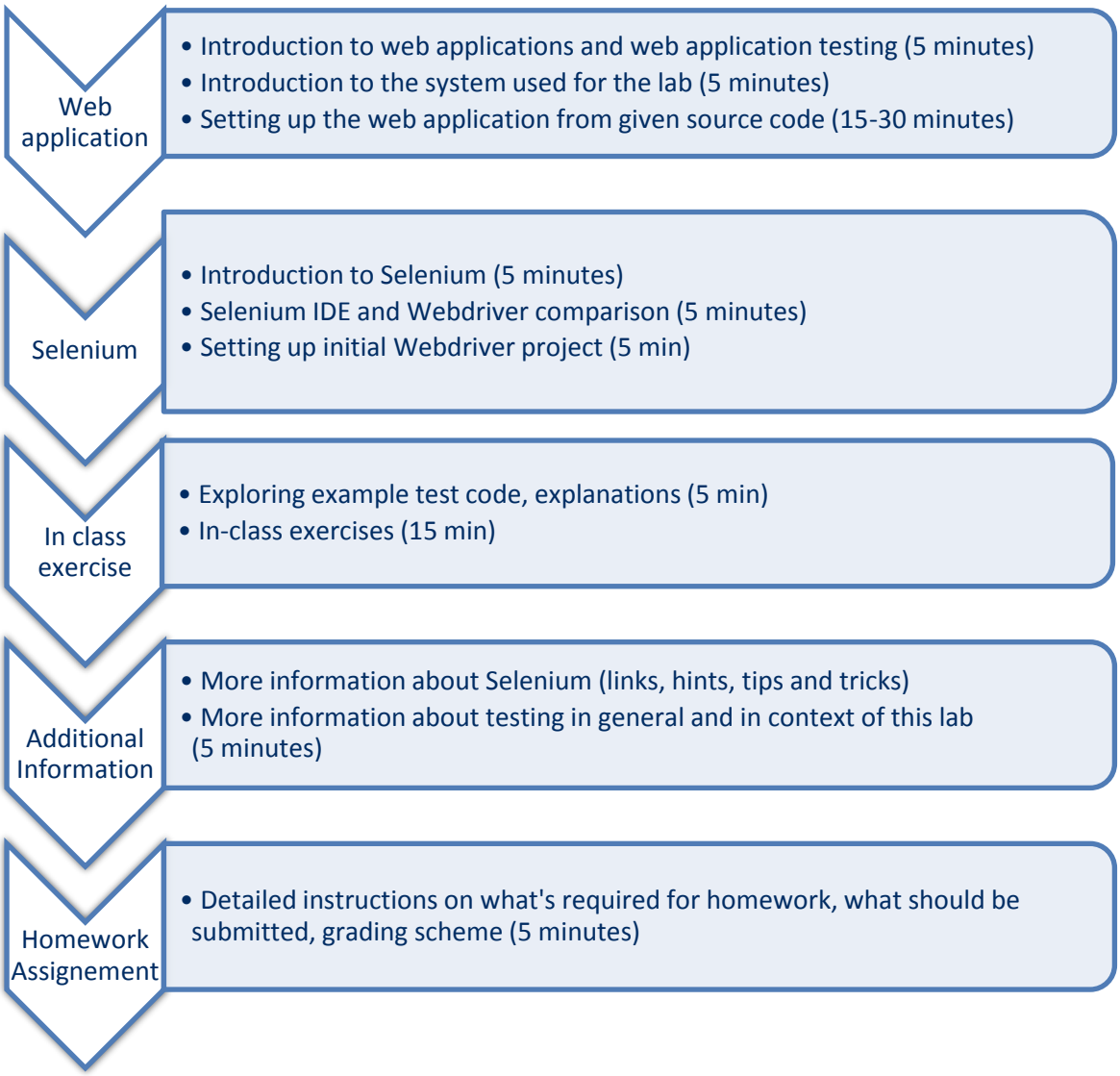


Figure 5. Lab workflow.

Figure 5 shows approximate workflow through the lab with time estimations.

Web application

At first students are given a short introduction to web applications, web application testing and the system they'll be using for this lab.

Students have access to the application source code on courses page, they have to deploy the application themselves. Detailed instructions are provided in the lab instructions.

Selenium

If students hear the word 'Selenium', they usually mean Selenium IDE, the click around plugin for Firefox. But in this lab, we want them to use Webdriver, and write proper test scripts. To make it clear for students, why they should use Selenium Webdriver over IDE, pros and cons of both are described.

Students are provided with initial code and installation instructions to setup Webdriver based test.

In class exercises

The example test code provides multiple examples, how to use Selenium for testing. The example test code will be explored and some examples will be explained to get the initial knowledge of structure and syntax.

First in-class exercise is to complete a test case, that checks if login, logout functionality works.

For the second exercise they have to write a negative test case of login. Under negative test case it's meant, that login fails in case, where it should fail.

Additional information

After the students have had basic introduction to Selenium and writing tests, they are provided with some useful links to documentation and hints on how to use selenium and make tests easily manageable.

Homework assignment

For the homework assignment the students have to write unit tests covering all of the functionalities. To get full marks it's required to write at least 20 test cases in total, at least 5 of them as negative tests and two failing tests, indicating that student has found at least two bugs.

Submission should contain self-written test suite and executed test report.

Maximum number of points in this lab is 10. Students will lose points if they don't submit all required files, write less test cases or don't find bugs.

4. Evaluation

4.1 Lab Execution

The given lab was executed in spring 2018. Students were beforehand familiar with web applications and had gotten a brief introduction to Selenium during lecture.

Overall the lab went smoothly. For over half of the students everything worked perfectly, the other students didn't have typical issues, that could be fixed beforehand. With the exception of one student, all problems could be fixed within the lab session time.

During web application setup students got errors if they didn't follow the directions in right order or installed not compatible, older versions of jruby and/or Heroku. In some computers it was also necessary to run the installation process in administrator privileges.

Some students had old versions of Firefox, that Selenium wasn't compatible with, installed and caused the Webdriver to crash with an unexpected error message. Selenium Webdriver officially works with Firefox versions 55 and greater, but for better performance it's always suggested to use the latest version [9].

4.2 Feedback

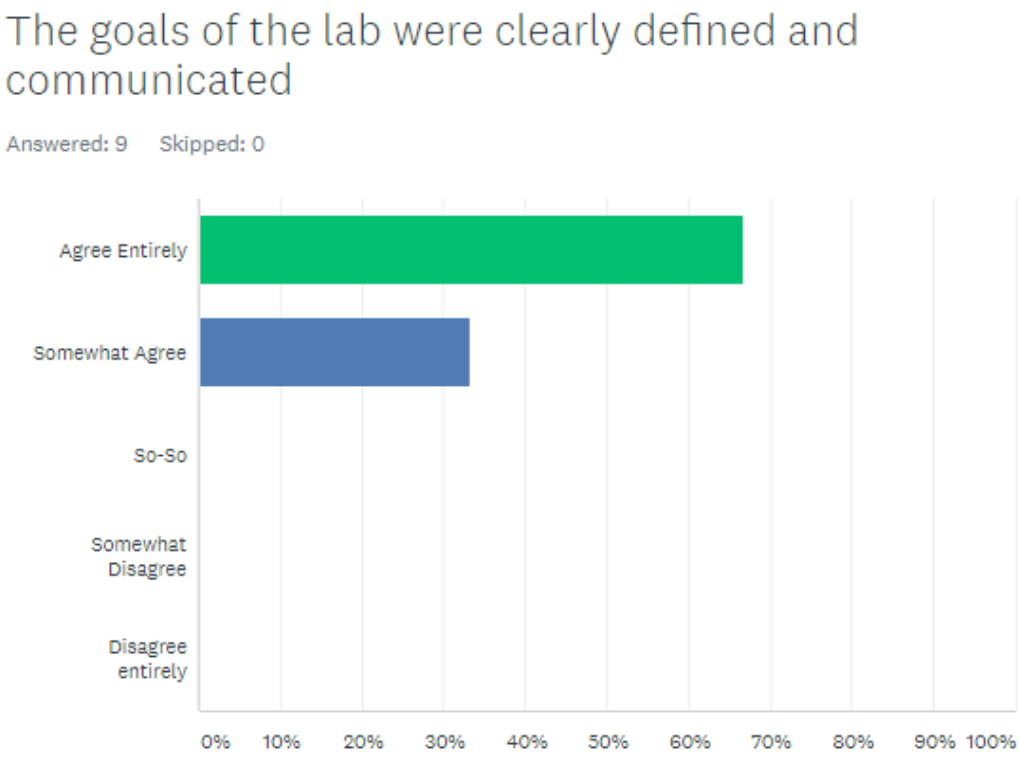
After students submitted their work, they were asked to fill in an online questionnaire using SurveyMonkey[1]. The feedback was anonymous, voluntary and didn't affect the grade received for the lab. 9 students out of 95 enrolled in the course filled in the questionnaire. The questionnaire was evaluated based on a Likert Scale [10], where students expressed, how much they agree with the statements. There were 8 statements in the questionnaire:

1. "The goals of the lab were clearly defined and communicated";
2. "The tasks of the lab were clearly defined and communicated";
3. "The materials of the lab were appropriate and useful";
4. "The Selenium framework was interesting and useful";
5. "The support received from the lab instructors was appropriate";
6. "The grading scheme was transparent and appropriate";
7. "Compared to the previous labs, the difficulty/complexity of the lab was";
8. "Overall the lab was useful in the context of this course".

[1] https://www.surveymonkey.com

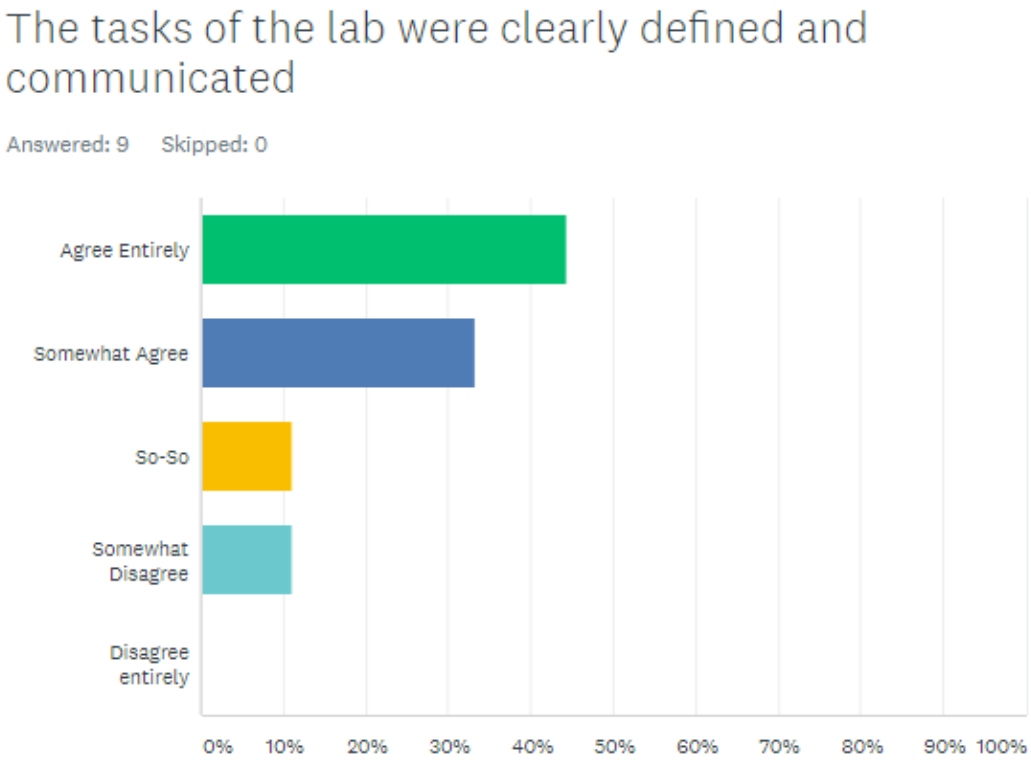
Statements 1-6 and 8 had possible answers “Agree Entirely”, “Somewhat Agree”, “So-So”, “Somewhat Disagree”, and “Disagree Entirely”. Statement 7 had options “Much Lower”, “Somewhat Lower”, “The Same”, “Somewhat Higher”, “Much Higher”. Finally, students had the possibility to give any additional feedback in open form.

Results from feedback.



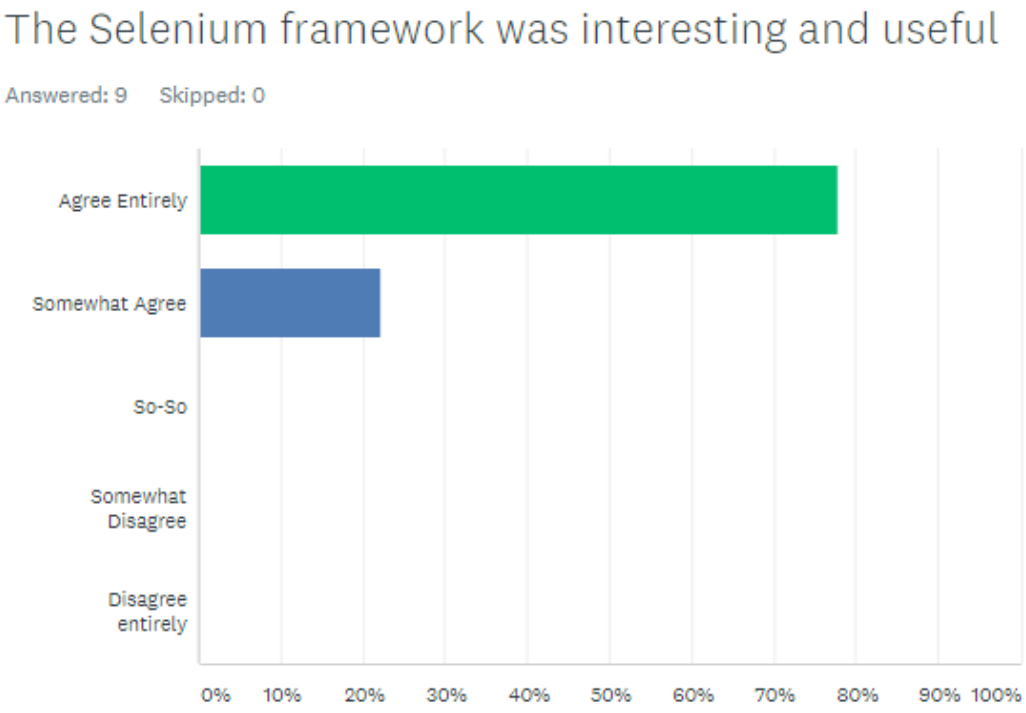
ANSWER CHOICES	RESPONSES	
Agree Entirely	66.67%	6
Somewhat Agree	33.33%	3
So-So	0.00%	0
Somewhat Disagree	0.00%	0
Disagree entirely	0.00%	0
TOTAL		9

Figure 6. Results for statement “The goals of the lab were clearly defined and communicated”



ANSWER CHOICES	RESPONSES	
Agree Entirely	44.44%	4
Somewhat Agree	33.33%	3
So-So	11.11%	1
Somewhat Disagree	11.11%	1
Disagree entirely	0.00%	0
TOTAL		9

Figure 7. Results for statement "The tasks of the lab were clearly defined and communicated"

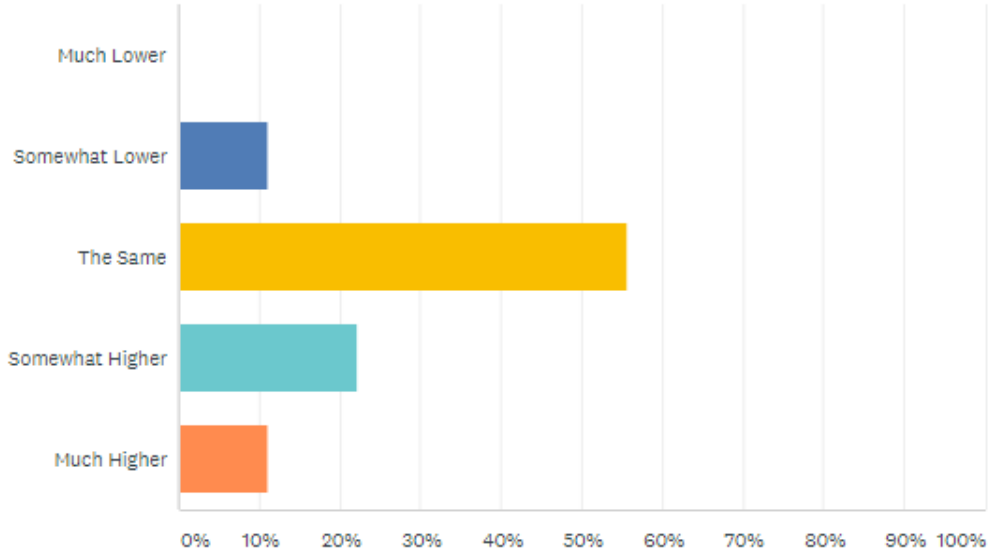


ANSWER CHOICES	RESPONSES	
Agree Entirely	77.78%	7
Somewhat Agree	22.22%	2
So-So	0.00%	0
Somewhat Disagree	0.00%	0
Disagree entirely	0.00%	0
TOTAL		9

Figure 8. Results for statement “The Selenium framework was interesting and useful”

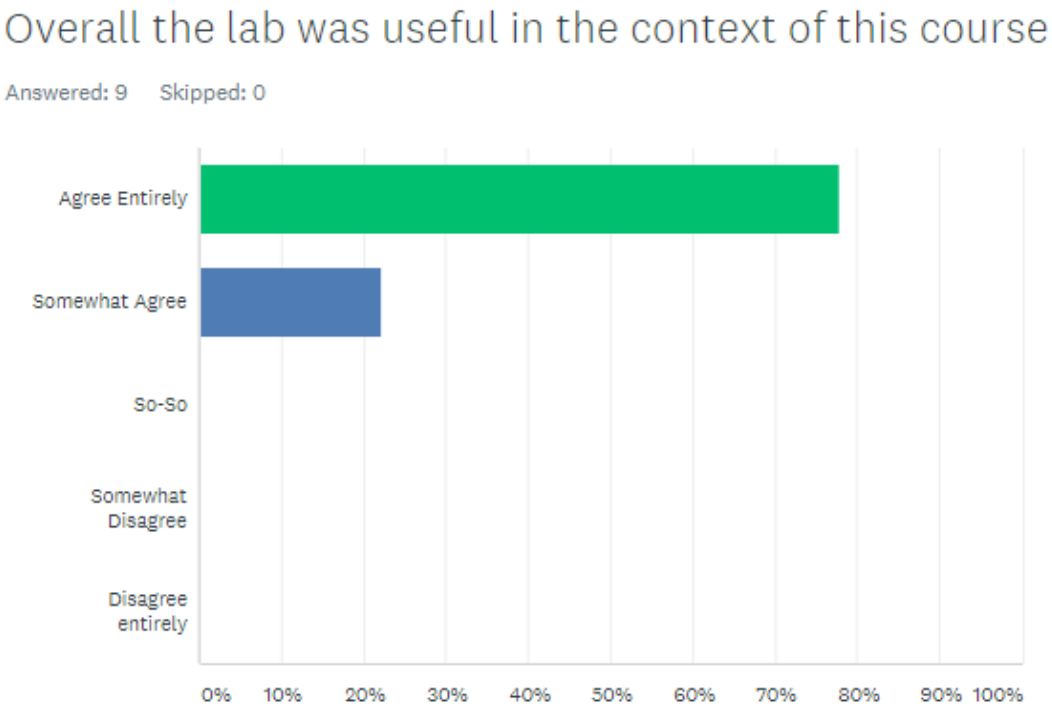
Compared to the previous labs, the difficulty/complexity of the lab was

Answered: 9 Skipped: 0



ANSWER CHOICES	RESPONSES	
Much Lower	0.00%	0
Somewhat Lower	11.11%	1
The Same	55.56%	5
Somewhat Higher	22.22%	2
Much Higher	11.11%	1
TOTAL		9

Figure 9. Results for statement "Compared to the previous labs, the difficulty/complexity of the lab was"



ANSWER CHOICES	RESPONSES	
Agree Entirely	77.78%	7
Somewhat Agree	22.22%	2
So-So	0.00%	0
Somewhat Disagree	0.00%	0
Disagree entirely	0.00%	0
TOTAL		9

Figure 10. Results for statement "Overall the lab was useful in the context of this course"

Positive aspects

Most of the feedback received was positive. All students, that answered the questionnaire, agreed that the goals of this lab were clearly defined and about 80% of them agreed that the tasks were understandable. All the students found the lab useful and Selenium framework interesting to learn. Majority of the students found the lab difficulty to be about the same level as other labs, so it's not required to add any extra tasks or to lower the effort required, to complete this lab.

Negative aspects

Compared to the previous labs, this lab required much more effort from 11% of the students. The main reasons were, that they had no prior knowledge about HTML or the tests failed,

because of slow internet connection and it took a while to figure out the reason. 55% of the students found the materials appropriate and useful, but some thought, they could be more informative. The students, that didn't find the materials so useful, had problems with finding "Export Test Results" button or had no prior knowledge on HTML. This course is targeted at second year students, who, if following the curriculum put together by the University of Tartu, should be at the same time taking the "Web Application Development" course, where HTML is covered in the beginning of the course. As the web application test automation lab was executed in the second half of the semester, it was assumed, that students are already familiar with HTML, but as of next year, the "Software Testing" course will be executed one semester after the "Web Application Development" course. Based on this and that it wasn't a problem for most of the students it's not required to provide any additional instructions or details on basic knowledge of HTML.

Ideas for future improvement.

It was requested to add into the instructions, where to find the button for exporting test results. This has already been made for next year and no bigger changes are currently requested. If in future a new lab would be added to the course, it could be possible to use this lab as build up and add for example performance testing of web applications as the next one.

5. Conclusions

In this thesis a new set of materials about automated web application testing was created and used in the "Software Testing" course. Students had the possibility to give feedback in an online survey. Overall the feedback collected was positive. Although some students found the lab to be more difficult, than the previous ones, all of them found the lab package to be interesting and useful to learn.

For future, no crucial changes are needed to be made for this lab package. As a suggestion, a new lab package on different type of web application testing, for example performance testing, could be built up based on this lab.

The lab package turned out to be successful and should be kept as part of "Software Testing" course.

6. References

[1] John E. Bentley (2005) Software Testing Fundamentals—Concepts, Roles, and Terminology. Proceedings of the Thirtieth Annual SAS® Users Group International Conference (SUGI 30), Philadelphia, Pennsylvania, April 10-13, 2005, paper 141-30. (27.02.2108)

[2] Puppet and DevOps Research and Assessments (DORA). "2016 State of DevOps Report", https://puppet.com/resources/whitepaper/2016-state-of-devops-report, 2016 (19.03.2018)

[3] "Web Application Testing", https://www.guru99.com/web-application-testing.html (19.03.2018)

[4] B. Anderson, "Best Automation Testing Tools for 2018", https://medium.com/@briananderson2209/best-automation-testing-tools-for-2018-top-10-reviews-8a4a19f664d2 (15.04.2018)

[5] Selenium Homepage, "Test Automation for Web Applications", https://www.seleniumhq.org/docs/01_introducing_selenium.jsp#test-automation-for-web-applications (27.02.2018)

[6] P. Ramya, V. Sindhura, P. V. Sagar (2017) Testing using Selenium Web Driver. Proceedings of the Second International Conference on Electrical, Computer and Communication Technologies (ICECCT 2017), IEEE, pp. 391-397. (27.02.2018)

[7] Official Selenium Blog, "Firefox 55 and Selenium IDE", https://seleniumhq.wordpress.com/2017/08/09/firefox-55-and-selenium-ide (30.03.2018)

[8] Selenium Homepage, "Selenium Webdriver", https://www.seleniumhq.org/docs/03_webdriver.jsp (30.03.2018)

[9] Geckodriver Wiki, https://github.com/mozilla/geckodriver (19.04.2018)

[10] S. McLeod, "Likert Scale", https://www.simplypsychology.org/likert-scale.html (19.04.2018)

Appendix

I. Lab Materials

Student Materials

- "Lab 5 Instructions" – https://courses.cs.ut.ee/2018/SWT2018/spring/uploads/Main/SWT2018-lab05-20180207.pdf
- "Lab 5 Test Code" - https://courses.cs.ut.ee/2018/SWT2018/spring/uploads/Main/SWT2018lab05-SourceCodev1.zip
- "Application Source Code" - https://courses.cs.ut.ee/2018/SWT2018/spring/uploads/Main/SWT2018-lab05-ApplicationSourceCode.zip

TA Materials

- "Lab 5 TA Material"
- "Lab 5 Lab Assignment Solution"
- "Lab 5 Homework Solution"
- "Lab 5 Homework Test Results"

TA Materials are not made available in the thesis, but are available upon request.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Riana Randoja**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

    1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

    1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis **Title, Lab Package: Automated Web-Application Testing**,

supervised by Dietmar Pfahl,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **08/05/2018**