

BLG 475E - Software Quality and Testing

2023-24 Spring Term

Homework #2

Due: May, 10 23:59

Introduction

In this assignment, you will learn how to implement and create UI functionality tests.

For any issues regarding the assignment, please contact Sultan Çoğay (cogay@itu.edu.tr).

Implementation Notes

The implementation details below are included in the grading:

1. Please follow a consistent coding style (indentation, variable names, etc.) with comments.
2. You should create test scenarios on the given website : <https://en.wikipedia.org/w/index.php?search>
3. Test scenarios must be implemented using Java programming language and test automation tool Selenium WebDriver version 4.19.1. You should perform your test on Chrome browser utilizing *ChromeDriver*.
4. Feel free to discover and use all properties and functionality of Selenium WebDriver. Documentation for Selenium : <https://www.selenium.dev/documentation/webdriver/>

Submission Notes

1. You should compress all your necessary s files into an archive file (.zip). Please write your name and ID on the top of each document that you will upload as in following format:

```
/* @Author  
* Student Name: <student_name>  
* Student ID: <student_id>  
*/
```
2. Submissions are made through Ninova system **only**, and have a strict deadline. Assignments submitted after the deadline will not be accepted.
3. This is not a group assignment and getting involved in any kind of cheating is subject to disciplinary actions. Your homework should not include any copy-paste material.
4. You can not publish your homework (on Github, social media, personal blogs etc.) before it has graded.

1 Tests

1.1 Scenario 1 (30 pts.)

Advanced Search: Implement **Scenario1.java** file under hw2 package and perform automated tests to assess the correctness of the first scenario by searching for these words **software testing** in which the page title contains **testing** and file types can be one of the three: **pdf, video, mpg**. The results should be sorted according to their relevance as shown in Figure 1a.

The screenshot shows the 'Advanced search:' section of a search engine. It includes several input fields and buttons:

- Search in page text:**
 - These words:** A text box containing 'software' and 'testing', each with a small 'X' icon to its right.
 - Exactly this text:** A text box containing '"cat loves goat"'.
 - Not these words:** An empty text box.
 - One of these words:** An empty text box.
- Structure:**
 - Page title contains:** A text box containing 'testing'.
 - Subpages of this page:** An empty text box.
 - Pages in these categories:** An empty text box.
 - Pages with these templates:** An empty text box.
- Files and images:**
 - File type:** A dropdown menu with 'pdf' selected.
- Sorting order:** A dropdown menu with 'Relevance' selected.

At the bottom, there is a 'Search in:' section with '(Article)' and 'File' selected, each with a small 'X' icon to its right.

(a) Scenario 1

The screenshot shows the 'Advanced search:' section of a search engine. It includes several input fields and buttons:

- Search in page text:**
 - These words:** A text box containing 'software' and 'quality', each with a small 'X' icon to its right.
 - Exactly this text:** A text box containing '"software quality"'.
 - Not these words:** A text box containing 'covid', with a small 'X' icon to its right.
 - One of these words:** An empty text box.
- Structure:**
 - Page title contains:** An empty text box.
 - Subpages of this page:** An empty text box.
 - Pages in these categories:** An empty text box.
 - Pages with these templates:** An empty text box.
- Files and images:**
 - File type:** A dropdown menu with 'Image' selected.
 - File width in pixels:** A dropdown menu with 'greater than' selected, followed by an empty text box and 'px'.
 - File height in pixels:** A dropdown menu with 'greater than' selected, followed by an empty text box and 'px'.
- Sorting order:** A dropdown menu with 'Creation date – current on top' selected.

At the bottom, there is a 'Search in:' section with '(Article)' and 'File' selected, each with a small 'X' icon to its right.

(b) Scenario 2

Figure 1: Test Scenarios

1.2 Scenario 2 (30 pts.)

Add more functionality to the first scenario and implement **Scenario2.java** file under hw2 package by **adding words that should not be seen in the found pages, such as covid, Africa, food, and by adding words that should exactly be seen in the found pages, such as software quality, automation**. The pages should be filtered to results having one of the two file formats: audio and image. Pick the file types from these formats yourself during your tests. Test as many formats as possible. Sort the pages according to creation date and edit date. An example scenario shown in Figure 1b

1.3 Scenario 3 (40 pts.)

For last step, implement **Scenario3.java** file under hw2 package and **create a scenario yourself** for a new keyword from the actions that can be done in the advanced search section. Use your creativity and try to explore more options.

Evaluation criteria:

Your tests will be evaluated based on the number and the variety of the test cases such as variety of tested formats and text boxes etc.