

# Übungsblatt 02

---

Abgabe von: **Jonas Merkle** [jonas.merkle@uni-ulm.de]  
Abgabe bis: 05.11.2024 08:00  
Repository: <https://github.com/jonas-merkle/Softwarequalitaetssicherung>  
Dateien: PDF, ZIP, HTML

---

## Inhaltsverzeichnis

- Übungsblatt 02
  - Inhaltsverzeichnis
  - Aufgabe 1
    - \* Project Structure
    - \* Test Cases
    - \* Setup and Execution
    - \* Notes

## Aufgabe 1

The test project `SeleniumTestOfLSF` contains UI tests for the **LSF website** of Ulm University, implemented using **Xunit** and **Selenium WebDriver**. The tests verify the functionality of various UI components on the website, ensuring that elements such as the navigation menu, course search, and login fields work as expected.

### Project Structure

- **UiTests.cs**: Contains multiple test cases, each following the Arrange-Act-Assert pattern.
- **Dependencies**:
  - `Selenium.WebDriver`
  - `Selenium.WebDriver.ChromeDriver`
  - `FluentAssertions`
  - `Xunit`

### Test Cases

1. **Test\_TopMenuInGerman**: Verifies that the default top menu items are displayed in German.
2. **Test\_TopMenuInEnglish**: Checks that the top menu items switch to English after selecting the English language option.
3. **Test\_SearchForCourse**: Searches for a specific course by course number and confirms its presence in the results.
4. **Test\_CourseNameAndID**: Validates that the course title and ID are displayed correctly on the course details page.

5. **Test\_LoginInputFieldsPresence:** Confirms that the username, password, and login button fields are present on the login page.

### Setup and Execution

1. **Install Chrome:** Make sure Chrome is installed on the machine running the tests.
2. **Install Dependencies:**
  - Use NuGet Package Manager or run:

```
dotnet add package Selenium.WebDriver  
dotnet add package Selenium.WebDriver.ChromeDriver  
dotnet add package FluentAssertions  
dotnet add package xunit
```
3. **Run Tests:**
  - To execute tests, use the following command in the terminal:

```
dotnet test
```

### Notes

- The tests are configured to run in **headless mode** by default. You can remove the `--headless` argument in `UiTests.cs` to view the browser during tests.
- `ChromeDriver` is set to maximize the browser window to ensure all elements are visible for interaction.