Test assignment selenium

- **Discuss Pros and Cons with manual versus automated tests**

  Pros:
  - Manual testing is easy to carry out
  - It is cheap to maintain, and a single tester can perform the processes
  - It gives more credibility, since real users act different from a simulation. Testers act as end-users and gives an overview of issues of what is needed to be resolved
  - Ideas arise doing the software testing and this can help to coverage areas that have not been considered to touch

  Cons:
  - Human errors is one of the biggest issues and it can't be prevented
  - Because it is most likely to be performed by a human, the same errors will theoretically be present each testing
  - Each time something is changed, a resource is need to perform the testing. It is time consuming and therefore automated testing can help out with this.

- **Explain about the Test Pyramid and whether this exercise supports the ideas in the Test Pyramid**

  The Test Pyramid gives an indication of how much testing you should do in each of its layer: UI Testing, Service Testing, and Unit Testing.
  This exercise gave an idea of testing the UI and whether it behave as expected, but it didn't really tell anything about each Units of the program itself. This was more a matter of automating the UI testing and getting your hands dirty in that field. You could argue that each of the different things that were testing is called an unit test, but I would state that it should be more "lower" level to be a unit test.

- **Discuss some of the problems with automated GUI tests and what makes such tests "vulnerable"**

  I think that one problem with automated GUI testing is that it neglects the human component. It misses the actual interaction with the system. Humans behave differently. Even though your system might work, it doesn't mean it's friendly to the user. UI testing is at its most about communication. Automating testing can't tell whether the login sequence was OK to the user. Maybe the human has a hard time to tell whether they were logged in or not – even though the automated test said it was OK?

- **Demonstrate details in how to create a Selenium Test using the code for the exercise**

This has been demonstrated with the Java Project uploaded on Github

- **Explain shortly about the DOM, and how you have read/manipulated DOM-elements in your test**

  When you load a web page, the browser creates what is called a Document Object Model, which is a standard to access documents in the HTML code. It is like a programable interface that gives you the rights to select the HTML objects, properties, methods, and events. With events you can even interact with them. In short: a way to get, change, add, or delete HTML elements.

  With these elements with can piggy bag on the DOM from the document with loaded on the page. We can extract the different tags and their values and check whether they match with what was expected.

- **Explain how (and why it was necessary) you have solved "waiting" problems in your test**

  I have solved waiting problems in my test by sleeping the thread. This means that the code that needs to be executed to check the different scenarios that I want to accomplish, they have to wait to make sure that the browser can follow. As an example: if a table had to load 1000 elements, but you have a small idea that the database might be slow on this; then you sleep the thread for 5-10 seconds to make sure you get the correct values, as you expect.

  This was necessary to avoid the code being faster to check than the browser could give the data. This will avoid the test failing even though it was doing the correct things.