

POST ASSIGNMENT QUESTIONS

After having completed the assignment you have also to fill-in this questionnaire

1) Implementing test methods manually with Selenium WebDriver is a simple task

(1) Strongly Agree **(2) Agree** (3) Unsure (4) Disagree (5) Strongly Disagree

2) Implementing test methods with ASSESSOR+ is a simple task

(1) Strongly Agree (2) Agree (3) Unsure **(4) Disagree** (5) Strongly Disagree

3) It is faster to produce test suites with ASSESSOR+ w.r.t manually with Selenium WebDriver

(1) Strongly Agree (2) Agree (3) Unsure **(4) Disagree** (5) Strongly Disagree

4) The recording phase with ASSESSOR+ is easy to complete

(1) Strongly Agree **(2) Agree** (3) Unsure (4) Disagree (5) Strongly Disagree

5) The code generated by ASSESSOR+ is easy to modify

(1) Strongly Agree (2) Agree (3) Unsure (4) Disagree **(5) Strongly Disagree**

6) I found the assignment useful

(1) Strongly Agree (2) Agree (3) Unsure (4) Disagree **(5) Strongly Disagree**

7) I found the assignment too difficult/complex

(1) Strongly Agree (2) Agree (3) Unsure **(4) Disagree** (5) Strongly Disagree

8) I had enough expertise to complete the assignment

(1) Strongly Agree **(2) Agree** (3) Unsure (4) Disagree (5) Strongly Disagree

COMMENTS AND CONSIDERATIONS (Optional):

Unfortunately happened that the tool missed some explicit ({ASSESSOR}) instructions, generating a code that don't work as it is. Moreover, happened that the implementation suggested waited for some elements to load that by gherkin definition they were deleted. The code after some fixes work, but still lacks in perfection.

Other problems I found with ASSESSOR+ are the followings:

- There are A LOT of unused imports, probably because the tool prepares a pool of imports used and copy and paste them in each script: this can be useful in certain cases, but reporting also all those imports in the page object script is not.
- There are a lot of variable creation for pages on which a single function call is implemented: the code length of the function can be reduced by a lot by just concatenating those calls: "new <PageObject>.functionToCall()".

README (before running tests)

NOTE#1: to run the test scripts I suggest following **strictly** the lexicographical order of the names, to allow the correct creation of the several elements under test. If this does not happen, the tests may fail.

NOTE#2: the tests are made to be run on a clean installation of the requested services. Since xpaths are used, if the user manually tries to change in any form the website (e.g. by adding elements) and then runs the script, some tests may fail.

NOTE#3: if for any reason some “http error” happen, please re-run the test, it shouldn’t be a problem with the script itself.

Results report

Now I will write some opinions about the results obtained.

I expected that a process guided by automation was more straightforward and efficient. Unfortunately, this is not the case. I spent about 4 hours for the first application (Bludit) to test, also influenced by the fact that there was the first time (after some self-made warming-up) that I used the combination of the tools.

After that, the second application (MantisBT) to be tested manually was very lightweight and efficient, even if it was done manually. Here I spent more or less 3 hours.

Then there is Prestashop app, on which I’ve done automatic testing. This time I was more prepared to, and I spent “only” 2 hours and half. For this specific case I think that if I had to do this application manually, I would have taken a lot more time, since the application is more complex.

Finally, there was one of the lightest applications to test: Claroline, on which I spent about 2 hours with a manual testing. In this case the application itself showed some simplicity and guidance since many fields were labelled with an ID and so they were easier to be called with the functions.

In the end I prefer a manual approach, since I have more control and familiarity over it. But I can’t say I dislike the record and repeat behaviour of Selenium WebDriver, since it allows less windows switching then testing that happen when testing manually (I need to click and then “report” manually what I see and so the behaviours of the scripts). The particularities I really don’t like of ASSESSOR+ are the strange implementations that it suggests (even for similar behaviours!) and the ambiguous name of the generated functions: no limiting problems found.