

Chapter 4

The Test Case Set

Introduction

In chapter 4, we developed all 25 test cases for the user interface for Wheelmap. We also made improvements based on the feedback we received on deliverable 3, particularly about making our output report more verbose.

Updates to Report

During deliverable 3, our test case output looked like Figure 1.

Unittest Results

Start Time: 2020-11-05 09:43:35

Duration: 61.68 s

Summary: Total: 6, Pass: 6

__main__.TestMap	Status
test_1	Pass
test_2	Pass
test_3	Pass
test_4	Pass
test_5	Pass
test_function	Pass

Total: 6, Pass: 6 -- Duration: 61.68 s

Figure 1: test case output from deliverable 3

Dr. Bowring gave us the feedback to make this report more verbose, meaning to add all parts of the test case (component, requirement, output) to the output. We are using a package called HTMLTestRunner to configure the output to our unittest test cases; initially we were just using the default template. To update the columns of the output table, we had to create our own template to pipe to HTMLTestRunner. It required a new template engine none of us had ever worked in called Jinja2.

```
{%- for test_case in tests_results %}
{%- if not test_case.subtests is defined %}
<tr class='{{ status_tags[test_case.outcome] }}'>
  {%- if not test_case.test_id.split(".")[-1] == "test_function" %}
  <td class="col-xs-5">test_{{ testCase_list[loop.index0].id }}</td>
  <td> {{ testCase_list[loop.index0].component }}</td>
  <td> {{ testCase_list[loop.index0].req }}</td>
  <td> {{ testCase_list[loop.index0].output }}</td>
  {%- endif -%}

```

Figure 2: template.html

In short, we started with the template provided by the HTMLTestRunner github. Then, we added new columns and as we looped through creating each row, used the loopindex to grab the correct

requirement, component, or output to add to the column-row combination.

Start Time: 2020-11-17 11:49:09

Duration: 253.16 s

Summary: Total: 26, Pass: 26

TestMap for WheelMap	Component	Requirement	Output	Status
test_11	CONTACT-nav-link	Link redirect	PASS	Pass
test_7	Claim	text rendered	PASS	Pass
test_15	ADDPLACE-nav-link	Link redirect	PASS	Pass
test_16	ImproveThisMap-nav-link	Link redirect	PASS	Pass
test_21	OpenStreetMap	SubComponent click	PASS	Pass
test_20	MapBox	SubComponent click	PASS	Pass
test_17	sozialhelden-logo	Link redirect	PASS	Pass
test_9	NEWS-nav-link	Link redirect	PASS	Pass
test_22	User-ZoomOut	ZoomOut	PASS	Pass
test_23	ac-marker-yes	elementRendered	PASS	Pass
test_4	leaflet-interactive	userLocationUpdate	PASS	Pass
test_5	search-input	searchBar	PASS	Pass
test_25	ac-marker-no	elementRendered	PASS	Pass
test_24	ac-marker-limited	elementRendered	PASS	Pass
test_10	PRESS-nav-link	Link redirect	PASS	Pass
test_12	IMPRINT-nav-link	Link redirect	PASS	Pass
test_1	CookieButton	elementRendered	PASS	Pass
test_6	logo	elementRendered		Pass
test_14	EVENTS-nav-link	Link redirect	PASS	Pass
test_2	NoCookieButton	elementRendered	PASS	Pass
test_13	FAQ-nav-link	Link redirect	PASS	Pass
test_18	User-ZoomIn	ZoomIn	PASS	Pass
test_3	leaflet-interactive	userLocationButtonRendered	PASS	Pass
test_8	GetInvolved-nav-link	Link redirect	PASS	Pass
test_19	MapBox-wordmark	SubComponent click	PASS	Pass
Pass				

Total: 26, Pass: 26 -- Duration: 253.16 s

Figure 3: test case output from deliverable 4

TestMap Component Test

Start Time: 2020-11-17 12:01:19

Duration: 21.89 s

Summary: Total: 3, Pass: 1, Fail: 1, Error: 1

TestMap for WheelMap

Component	Requirement	Output	Status
test_1	CookieButton	elementRendered	PASS

AssertionError:

Traceback (most recent call last): File "./scripts/testmap.py", line 17, in test_exec(input) in globals(), locals() File "", line 2, in AssertionError

test_2	NoCookieButton	elementRendered	PASS	Error	Hide
--------	----------------	-----------------	------	-------	------

SyntaxError: invalid syntax (, line 2)

Traceback (most recent call last): File "./scripts/testmap.py", line 17, in test_exec(input) in globals(), locals() File "", line 2 assertt elem.text == "Continue without cookies" ^ SyntaxError: invalid syntax

Pass

Total: 3, Pass: 1, Fail: 1, Error: 1 -- Duration: 21.89 s

Figure 5: test case output Failure and Error modes

```

24 if __name__ == "__main__":
25
26     testsmap = Parser.parse()
27
28     for test in testsmap:
29         test_func = TestMap.test_function(test.input)
30         setattr(TestMap, 'test_{0}'.format(test.id), test_func)
31
32     template_args = {
33         "testCase_list": testsmap
34     }
35
36     unittest.main(testRunner=HtmlTestRunner.HTMLTestRunner(template='./scripts/template.html',
37     template_args=template_args, output='../reports', report_name='testReport',
38     open_in_browser=True, report_title='TestMap component test'))

```

Figure 6: testmap.py (driver) updated passing in testsmap list into HtmlTestRunner

Reflection

This deliverable, we got together as a team a few times to focus on the task at hand. Learning about Jinja2 to reformat the report was an unexpected but valuable experience of working with a new template engine. We also had an interesting new software engineering experience. After spending a few hours working on our modified Jinja2 template, we finally got the report outputting the way we wanted. When we met the next day to put the finishing touches on this deliverable, we realized that the template had not saved! Fortunately, we were able to put the template back together in a fraction of the time it first took us.