

Lab 4 Report

Name: 陳宇宏

Student ID: 110598067

Date: 2021/06/05

1 Test Plan

1.1 Scope

對 KeystoneJS 網站的功能進行測試，並通過基於網站功能所設計出的測試案例。

測試工具::

- Google Chrome 瀏覽器
- Selenium with Python,
- Pycharm
- Python Unittest

1.2 Features to be tested

根據分析 KeystoneJS 上的功能，列出了以下 8 個 Features 來涵蓋所有相關 Features 以進行測試。

TC-001	TestCreateAPost : Create a post on the Admin UI page
TC-002	TestEditAPost : Edit a post on the Admin UI page
TC-003	testDeleteAPost : Delete a post on the Admin UI page
TC-004	testCreateAComment : Create a comment on Admin UI page
TC-005	testEditAComment : Edit a comment on Admin UI page
TC-006	testDeleteAComment : Delete a post on the Admin UI page
TC-007	TestCreateACategory : Create a category on Admin UI page
TC-008	testCreateAEnquiryonContactPage : Create an enquiry on the "Contact" page

1.3 Success criteria of completing the test

- (1) 測試須達到 100% 的通過率。

1.4 Test approaches

根據網站的功能，使用Each choice Criterion和Basis Path Coverage設計測試案例。

1.5 Testing tasks

No.	Activity Name	Plan hours	Schedule Date
1	測試環境架設	2	6/3
2	熟悉 KeystoneJS 功能,	1	6/4
3	設計測試案例	5	6/4
4	測試案例實作	10	6/5
5	撰寫測試報告	3	6/5
6	完成 Lab4 報告	3	6/5

2 Test Design

Test Design如同Excel

3 Test Implementation

因程式碼過長，只擷取部分程式碼，全部的測試實作程式碼放置於
Geoproject/LabReport/Lab4/testCode 內

No.	Test method	Source test code
1	testCreateANewPost	<pre>def test_create_a_post_by_ECC_TC001_T2(self): posts_page_button = self.driver.find_element_by_xpath("//a[normalize-space()='Posts']") posts_page_button.click() create_post_button = self.driver.find_element_by_xpath("//button[@data-e2e-list-create- button='no-results']") create_post_button.click() input_post_title = self.driver.find_element_by_xpath("//input[@class='css-foh633']") input_post_title.send_keys("Wakanda forever") create_button = self.driver.find_element_by_xpath("//button[normalize- space()='Create']") create_button.click() self.driver.switch_to.frame(self.driver.find_element_by_xpath("//*[co ntains(@id,'keystone-html-0_ifr')]")) content_brief = self.driver.find_element_by_xpath("//*[contains(@class,'mce-content- body')]") content_brief.send_keys("Spider man") self.driver.switch_to.default_content() save_post_button = self.driver.find_element_by_xpath("//button[@data-button='update']") save_post_button.click() success_prompt = self.driver.find_element_by_xpath("//div[@data- alert-type='success']") self.assertTrue("Your changes have been saved successfully" in success_prompt.get_attribute('innerHTML').strip()) post_list_page_button = self.driver.find_element_by_xpath("//a[contains(@class,'css- dmf4a8')]") post_list_page_button.click() new_post_link = self.driver.find_element_by_xpath("//a[contains(@class,'ItemList__val ue')]") self.assertEqual("Wakanda forever", new_post_link.get_attribute('innerHTML').strip())</pre>

		<pre>delete_post_button = self.driver.find_element_by_xpath("//button[contains(@class,'ItemList __control--delete')]") delete_post_button.click() confirm_delete_post_button = self.driver.find_element_by_xpath('//*[@data-button-type="confirm"]') confirm_delete_post_button.click()</pre>
--	--	--

2	testCreateAComment	<pre> def test_create_a_comment_by_BPC_TC004_P1(self): AUTHOR = None POST = "*-/+8dfdfgfdE6" STATE = "Draft" CONTENT = "*-/+8#\$^&fgfdE6asdfsfwefwef\nsdf\ndfewf" posts_page_button = self.driver.find_element_by_xpath("//a[normalize-space()='Posts']") posts_page_button.click() create_post_button = self.driver.find_element_by_xpath("//button[@data-e2e-list-create- button='no-results']") create_post_button.click() input_post_title = self.driver.find_element_by_xpath("//input[@class='css-foh633']") input_post_title.send_keys(POST) create_button = self.driver.find_element_by_xpath("//button[normalize- space()='Create']") create_button.click() self.go_to_comment_list_page() create_comment_button = self.driver.find_element_by_xpath('//*[@class="css-h629qq"]') create_comment_button.click() if AUTHOR is not None: author_option = self.driver.find_element_by_xpath("//*[@for='author']//*[contains(@cl ass,'Select--single')]") author_option.click() selected_author_option = self.driver.find_element_by_xpath("//*[@for='author']//*[contains(@cl ass,'Select-option') and normalize-space()='%s']" % AUTHOR) selected_author_option.click() post_option = self.driver.find_element_by_xpath("//*[@for='post']//*[contains(@clas s,'Select--single')]") post_option.click() selected_post_option = self.driver.find_element_by_xpath("//*[@for='post']//*[contains(@clas s,'Select-option') and normalize-space()='%s']" % POST) selected_post_option.click() confrim_create_comment_button = self.driver.find_element_by_xpath("//*[contains(@type,'submit')]") confrim_create_comment_button.click() </pre>

```

time.sleep(1)
self.driver.refresh()

state =
self.driver.find_element_by_xpath('//*[@for="commentState"]//div[contains(@class,"has-value")]')
state.click()
state_option =
self.driver.find_element_by_xpath('//*[@for="commentState"]//div[contains(@class,"has-value")]//*[contains(@class,"Select-option") and normalize-space()="%s"]' % STATE)
state_option.click()

PUBLISHED_DATE =
self.driver.find_element_by_xpath('//*[@for='publishedOn']//*[contains(@class,'FormField__inner')]").text

self.driver.switch_to.frame(self.driver.find_element_by_xpath("//iframe[contains(@id,'keystone-html')]"))
content =
self.driver.find_element_by_xpath("//*[contains(@class,'mce-content-body')]")
content.send_keys(CONTENT)
self.driver.switch_to.default_content()

save_button = self.driver.find_element_by_xpath("//*[@data-button='update']")
save_button.click()

success_prompt = self.driver.find_element_by_xpath("//div[@data-alert-type='success']")
self.assertTrue("Your changes have been saved successfully" in success_prompt.get_attribute('innerHTML').strip())

self.go_to_comment_list_page()

comment_column_item_list =
self.driver.find_elements_by_xpath("//*[contains(@class,'ItemList__col')]")
self.assertEqual(" " if AUTHOR is None else AUTHOR,comment_column_item_list[2].text)
self.assertEqual(POST,comment_column_item_list[3].text)

for element in PUBLISHED_DATE.split(' '):
    self.assertTrue(element in comment_column_item_list[4].text)
self.assertEqual(STATE,comment_column_item_list[5].text)

comment_column_item_list[1].click()
select_item_list_in_comment_edit_page =
self.driver.find_elements_by_xpath("//*[contains(@class,'Select--single')]//*[contains(@class,'Select-multi-value-wrapper')]")
self.assertEqual("Select..." if AUTHOR is None else AUTHOR,select_item_list_in_comment_edit_page[0].text)

self.assertEqual(POST,select_item_list_in_comment_edit_page[1].text)
self.assertEqual(STATE,select_item_list_in_comment_edit_page[2].text)

self.driver.switch_to.frame(self.driver.find_element_by_xpath("//iframe

```

		<pre>me[contains(@id, 'keystone-html')]")) content = self.driver.find_element_by_xpath("//*[contains(@class, 'mce-content- body')]") self.assertEqual(CONTENT, content.text) self.driver.switch_to.default_content()</pre>
--	--	--

3	testEditAComment	<pre> def test_edit_a_comment_by_ECC_TC005_T3(self): AUTHOR = None POST = "extenDssssssssssssdsfdsafdfwefwrrrrrStrangerThingsjfAaaaaaaaaaaaaawef aeddsvawedsvwefdwfaaaaaaaaaaqeqqq" STATE = "Archived" CONTENT = "extenDssssssssssssdsfdsafdfwefwrrrrrStrangerThingsjfAaaaaaaaaaaaaawef aeddsvawedsvwefdwfaaaaaaaaaaqeaaa" posts_page_button = self.driver.find_element_by_xpath("//a[normalize-space()='Posts']") posts_page_button.click() create_post_button = self.driver.find_element_by_xpath("//button[@data-e2e-list-create- button='no-results']") create_post_button.click() input_post_title = self.driver.find_element_by_xpath("//input[@class='css-foh633']") input_post_title.send_keys(POST) create_button = self.driver.find_element_by_xpath("//button[normalize- space()='Create']") create_button.click() comments_list_page_button = self.driver.find_element_by_xpath('//*[@data-list-path="post- comments"]') comments_list_page_button.click() create_comment_button = self.driver.find_element_by_xpath('//*[@class="css-h629qq"]') create_comment_button.click() </pre>


```

        if AUTHOR is not None:
            author_option =
self.driver.find_element_by_xpath("//*[@for='author']//*[contains(@class,'Select--single')]")
            author_option.click()
            selected_author_option =
self.driver.find_element_by_xpath("//*[@for='author']//*[contains(@class,'Select-option') and normalize-space()='s']" % AUTHOR)
            selected_author_option.click()

        post_option =
self.driver.find_element_by_xpath("//*[@for='post']//*[contains(@class,'Select--single')]")
        post_option.click()
        selected_post_option =
self.driver.find_element_by_xpath("//*[@for='post']//*[contains(@class,'Select-option') and normalize-space()='s']" % POST)
        selected_post_option.click()

        confrim_create_comment_button =
self.driver.find_element_by_xpath("//*[@type='submit']")
        confrim_create_comment_button.click()

        time.sleep(1)
        self.driver.refresh()

        self.go_to_comments_list_page()

        comment_column_item_list =
self.driver.find_elements_by_xpath("//*[@contains(@class,'ItemList__col')]")
        comment_column_item_list[1].click()

        time.sleep(1)
        self.driver.refresh()

        state =
self.driver.find_element_by_xpath("//*[@for='commentState']//div[contains(@class,'has-value')]")
        state.click()
        state_option =
self.driver.find_element_by_xpath("//*[@for='commentState']//div[contains(@class,'has-value')]//*[contains(@class,'Select-option') and normalize-space()='s']" % STATE)
        state_option.click()

        PUBLISHED_DATE =
self.driver.find_element_by_xpath("//*[@for='publishedOn']//*[contains(@class,'FormField__inner')]").text

self.driver.switch_to.frame(self.driver.find_element_by_xpath("//iframe[contains(@id,'keystone-html')]"))
        content =
self.driver.find_element_by_xpath("//*[@contains(@class,'mce-content-body')]")
        content.send_keys(CONTENT)
        self.driver.switch_to.default_content()

        save_button = self.driver.find_element_by_xpath("//*[@data-

```

```

button='update']")
    save_button.click()

    success_prompt = self.driver.find_element_by_xpath("//div[@data-
alert-type='success']")
    self.assertTrue("Your changes have been saved successfully" in
success_prompt.get_attribute('innerHTML').strip())

    self.go_to_comments_list_page()

    comment_column_item_list =
self.driver.find_elements_by_xpath("//*[contains(@class,'ItemList__co
l')]")
    self.assertEqual(""" if AUTHOR is None else
AUTHOR,comment_column_item_list[2].text)
    self.assertEqual(POST,comment_column_item_list[3].text)

    for element in PUBLISHED_DATE.split(' '):
        self.assertTrue(element in comment_column_item_list[4].text)
    self.assertEqual(STATE,comment_column_item_list[5].text)

    comment_column_item_list[1].click()
    select_item_list_in_comment_edit_page =
self.driver.find_elements_by_xpath("//*[contains(@class,'Select--
single')]//*[contains(@class,'Select-multi-value-wrapper')]")
    self.assertEqual("Select..." if AUTHOR is None else
AUTHOR,select_item_list_in_comment_edit_page[0].text)

    self.assertEqual(POST,select_item_list_in_comment_edit_page[1].text)
    self.assertEqual(STATE,select_item_list_in_comment_edit_page[2].text
)

    self.driver.switch_to.frame(self.driver.find_element_by_xpath("//ifra
me[contains(@id,'keystone-html')]"))
    content =
self.driver.find_element_by_xpath("//*[contains(@class,'mce-content-
body')]")
    self.assertEqual(CONTENT,content.text)
    self.driver.switch to.default content()

```

4 Test Results

TC-001TestCreateAPost

```
===== test session starts =====
collecting ... collected 6 items

TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_BPC_TC001_alternative_pa
TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_BPC_TC001_happy_path_cho
TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_ECC_TC001_T1
TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_ECC_TC001_T2
TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_ECC_TC001_T3
TestCreateAPost.py::TestCreateAPost::test_create_a_post_by_ECC_TC001_T4

===== 6 passed, 6 warnings in 41.55s =====

Process finished with exit code 0
PASSED [ 16%]PASSED [ 33%]PASSED [ 50%]PASSED [ 66%]PASSED [ 83%]PASSED [100%]
```

TC-002TestEditAPost

```
===== test session starts =====
collecting ... collected 7 items

TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_P1
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_P2
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_P3
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_TC002_T1
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_TC002_T2
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_TC002_T3
TestEditAPost.py::TestEditAPost::test_edit_a_post_by_ECC_TC002_T4

===== 7 passed, 8 warnings in 46.52s =====

Process finished with exit code 0
PASSED [ 14%]PASSED [ 28%]2022-06-05
PASSED [ 42%]PASSED [ 57%]PASSED [ 71%]PASSED [ 85%]PASSED [100%]
```

TC-003testDeleteAPost

```
>> ✓ Tests passed: 5 of 5 tests – 1 min 48 sec
sec D:\anaconda\python.exe "E:\PyCharm 2022.1.1\plugins\python\helpers\pycharm\_jb_pytest
Testing started at 下午 11:46 ...
Launching pytest with arguments TestDeleteAPost.py::TestDeleteAPost --no-header --no-

===== test session starts =====
collecting ... collected 5 items

TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_BPC_TC003_P1
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_BPC_TC003_P2
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T1
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T2
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T3

===== 5 passed in 108.42s (0:01:48) =====

Process finished with exit code 0
PASSED [ 20%]PASSED [ 40%]PASSED [ 60%]PASSED [ 80%]PASSED [100%]
```

TC-004testCreateAComment

```
===== test session starts =====
collecting ... collected 6 items

TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_BPC_TC004_P1
TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_BPC_TC004_P2
TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_ECC_TC004_T1
TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_ECC_TC004_T2
TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_ECC_TC004_T3
TestCreateAComment.py::TestCreateAComment::test_create_a_comment_by_ECC_TC004_T4

===== 6 passed, 36 warnings in 70.05s (0:01:10) =====

Process finished with exit code 0
PASSED [ 16%]PASSED [ 33%]PASSED [ 50%]PASSED [ 66%]PASSED [ 83%]PASSED [100%]
```

TC-005testEditAComment

✓ Tests passed: 6 of 6 tests – 1 min 4 sec

D:\anaconda\python.exe "E:\PyCharm 2022.1.1\plugins\python\helpers\pycharm_jb_pytest_

Testing started at 下午 11:36 ...

Launching pytest with arguments TestEditAComment.py::TestEditACommment --no-header --n

```
===== test session starts =====  
collecting ... collected 6 items
```

```
TestEditAComment.py::TestEditACommment::test_edit_a_comment_by_BPC_TC005_P1  
TestEditAComment.py::TestEditACommment::test_edit_a_comment_by_BPC_TC005_P2  
TestEditAComment.py::TestEditACommment::test_edit_a_comment_by_ECC_TC005_T1  
TestEditAComment.py::TestEditACommment::test_edit_a_comment_by_ECC_TC005_T2  
TestEditAComment.py::TestEditACommment::test_edit_a_comment_by_ECC_TC005_T4  
TestEditAComment.py::TestEditACommment::test_edt_a_comment_by_ECC_TC005_T3
```

```
===== 6 passed, 40 warnings in 64.02s (0:01:04) =====
```

Process finished with exit code 0

PASSED [16%]PASSED [33%]PASSED [50%]PASSED [66%]PASSED [83%]PASSED [100%]

TC-006testDeleteAComment

```
===== test session starts =====  
collecting ... collected 5 items
```

```
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_BPC_TC003_P1  
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_BPC_TC003_P2  
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T1  
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T2  
TestDeleteAPost.py::TestDeleteAPost::test_delete_a_post_by_ECC_TC003_T3
```

```
===== 5 passed in 111.75s (0:01:51) =====
```

Process finished with exit code 0

PASSED [20%]PASSED [40%]PASSED [60%]PASSED [80%]PASSED [100%]

TC-007TestCreateACategory

```
===== test session starts =====
collecting ... collected 6 items

TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_BPC_TC007_P1
TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_BPC_TC007_P2
TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_ECC_TC007_T1
TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_ECC_TC007_T2
TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_ECC_TC007_T3
TestCreateACategory.py::TestCreateACategory::test_create_a_category_by_ECC_TC007_T4

===== 6 passed in 65.12s (0:01:05) =====

Process finished with exit code 0
PASSED [ 16%]PASSED [ 33%]PASSED [ 50%]PASSED [ 66%]PASSED [ 83%]PASSED [100%]
```

TC-008testCreateAEnquiryonContactPage

```
===== test session starts =====
collecting ... collected 6 items

TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_happypath_Expect_create_new_enquirie_success
TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_with_T1_by_ECC_Expect_get_warning
TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_with_T2_by_ECC_Expect_get_warning
TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_with_T3_by_ECC_Expect_get_warning
TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_with_T4_by_ECC_Expect_create_new_enquirie_success
TestCreateAnENquiryOnContactPage.py::CreateANewEnquirieonContactPage::testCase_with_T5_by_ECC_Expect_create_new_enquirie_success

===== 6 passed in 77.31s (0:01:17) =====
```

5 Summary

在Lab4中，每個Testcase都使用了Each choice criterion和Basic Path Coverage來設計測試。且執行並通過全部的測試Selenium程式碼。這次的Lab4花比較多時間在於熟習Selenium以及環境安裝，並透過實作程式碼更熟悉ISP和Graph的測試方法。