Lab 4 Report

Name 張家齊
Student ID 110598109
Date 2022/6/1

1 Test Plan

1.1 Test requirements

本次測試之網站分為 Post Features、Comment Features、Category Features、Enquirie Feature、User Feature，在此 5 個 feature 內部，Lab 4 須滿足 (1) select one or more test cases for each of the following features (2) design at least 8 test cases (3) must do the verifications or assertions for each test scenario.

1.2 Test Strategy

To satisfy the test requirements listed in Section 1, a proposed strategy is to:

(1) design the test cases for those selected methods by using the **graph coverage** testing technique.

1.3 Test activities

To implement the proposed strategy, the following activities are planned to perform.

No.	Activity Name	Plan hours	Schedule Date
1	Study Selenium	2	5/23
2	Study Selenium IDE	1	5/23
3	Try use keystone.js	1	5/24
4	Design test cases for the selected methods	3	5/25
5	Implement test cases	12	5/28-5/30
6	Complete Lab4 report	2	5/31

1.4 Design Approach

以 activity diagram 先行分析各項 feature，並且在設計測試案例上，需要滿足 edge-coverage。

1.5 Success criteria

90% of all test cases must pass.

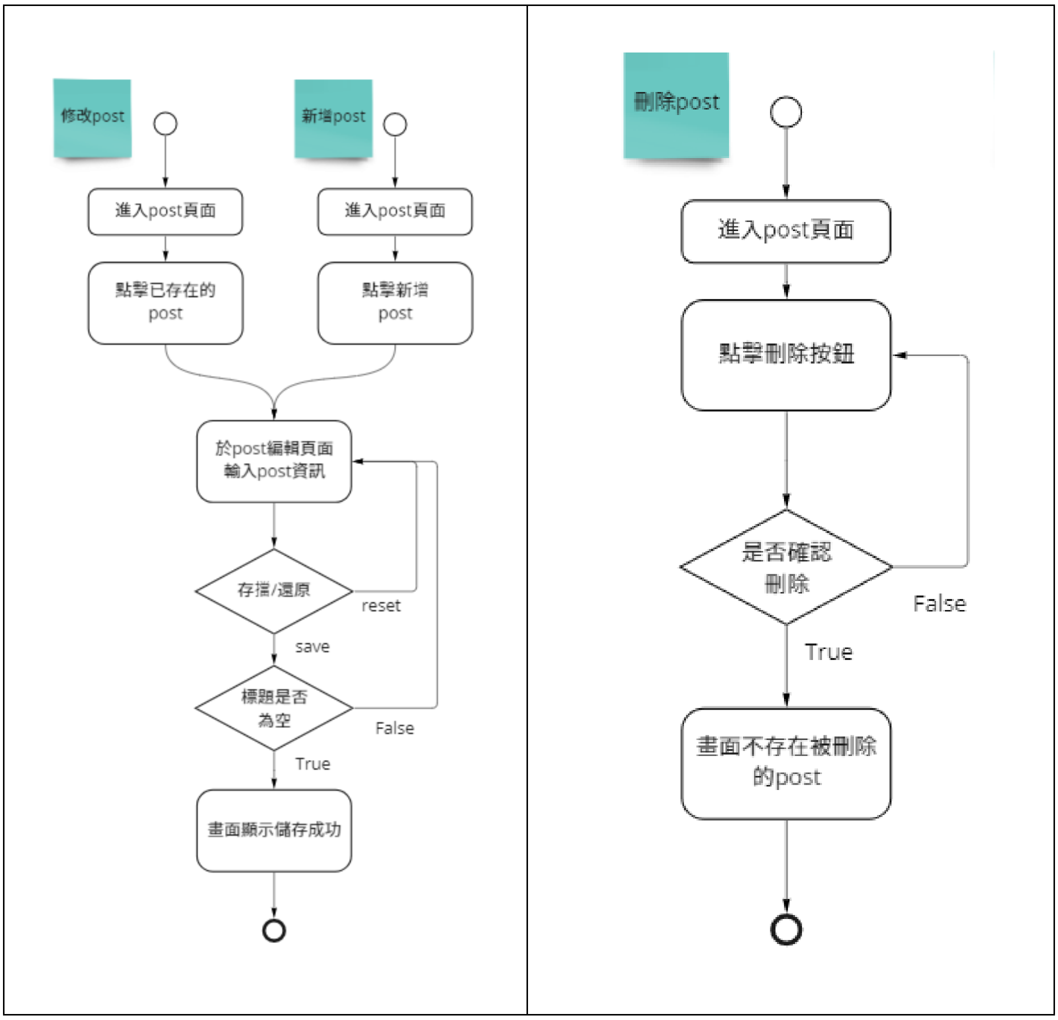
2 Test Design

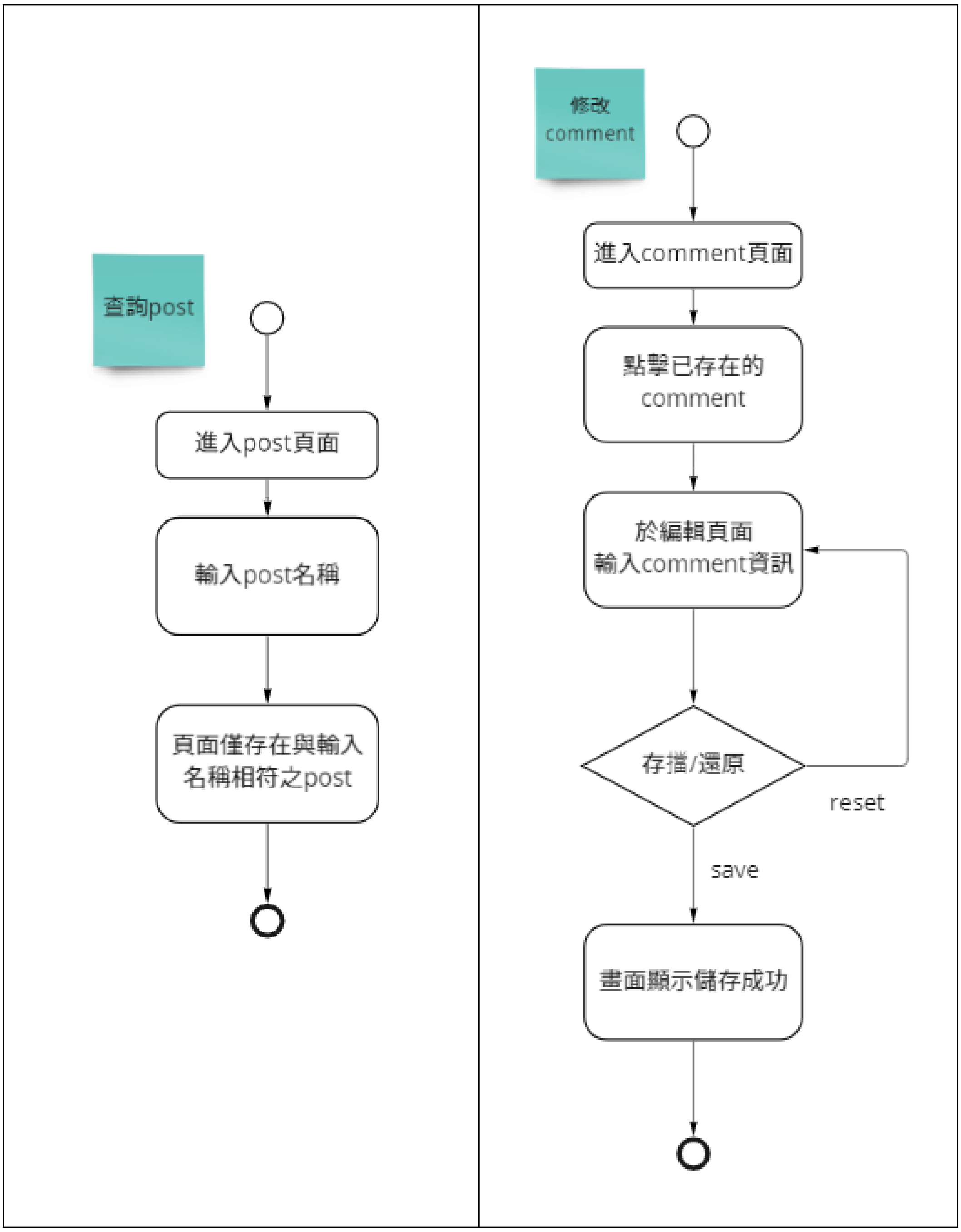
To fulfill the test requirements listed in section 1.1, the following methods are selected and corresponding test cases are designed.

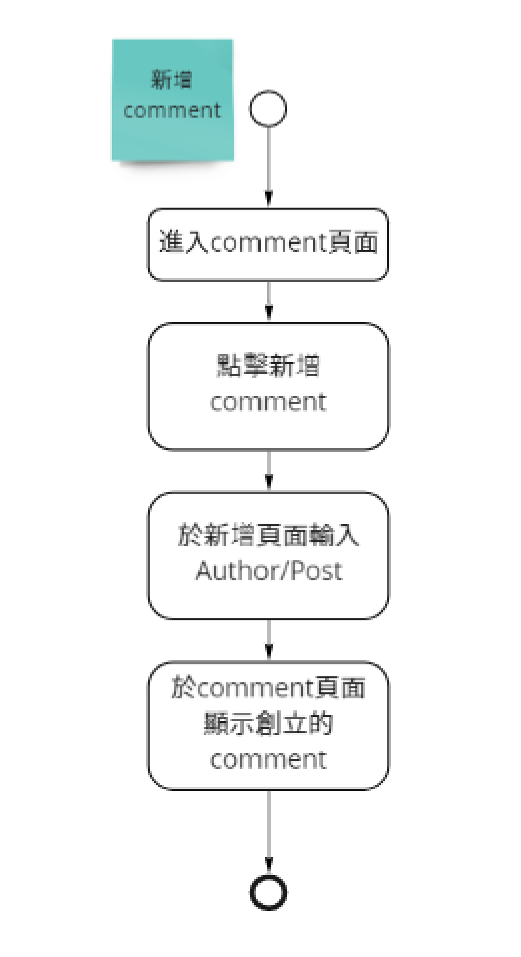
No.	Feature	requirement
1	Create a post on the Admin UI page	edge-coverage
2	Edit a post on the Admin UI page	edge-coverage
3	Delete a post on the Admin UI page	edge-coverage
4	Search posts by keyword on the Admin UI page	edge-coverage
5	Create a comment on Admin UI page	edge-coverage
6	Edit a comment on Admin UI page	edge-coverage
7	Create a category on Admin UI page	edge-coverage
8	Create a new user on Admin UI page	Happy path

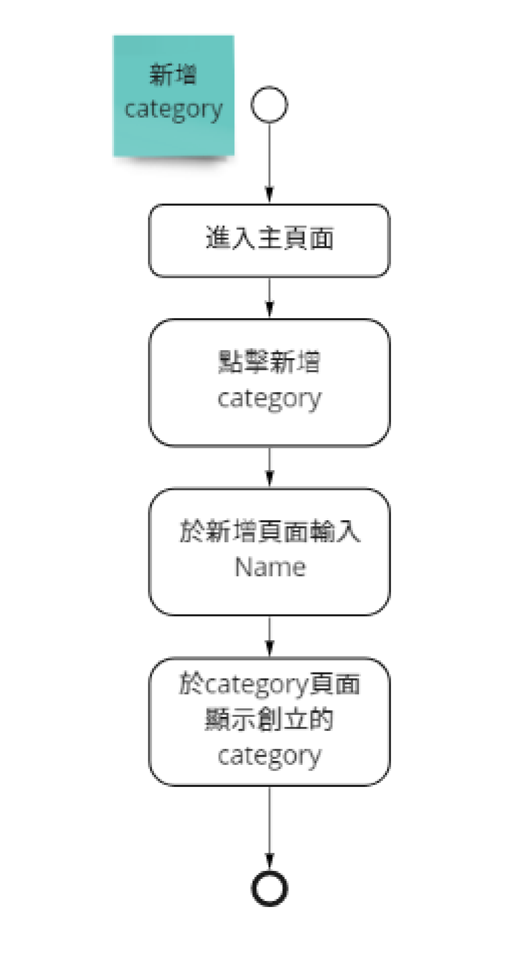
(1) Part1 : Design by edge-coverage

此處選擇以 activity diagram 先行分析，並且在設計測試上須滿足 edge-coverage，所繪出的 diagram 如下表，8 個 feature 需要 11 個 test case。




查詢post
進入post頁面
輸入post名稱
頁面僅存在與輸入名稱相符之post
修改comment
進入comment頁面
點擊已存在的comment
於編輯頁面輸入comment資訊
存擋/還原
reset
save
畫面顯示儲存成功





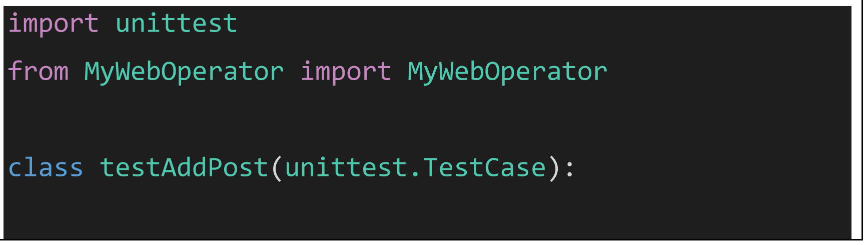
(2) Part2 : Design by happy path

在「Create a new user on Admin UI page」部分，因為牽涉可能的 diagram 分支過多，考量時間成本，最終在此部份選擇以 happy path 處理。

3 Test Implementation

The design of test cases specified in Section 2 was implemented using selenium. The test scripts of 3 selected test cases are given below. 其餘程式碼一併附於 lab report 內，此外設計之測試案例表也於同資料夾內。

No.	Test Case ID	Source test code

1	TC-POST-ADD-01	
		import unittest
		from MyWebOperator import MyWebOperator
		
		class testAddPost(unittest.TestCase):

<table>
<tr>
<td></td>
<td></td>
<td>

```
    webOperator = MyWebOperator()

    def setUp(self):
        self.webOperator.login("demo@keystonejs.com", "demo")
        self.webOperator.add_post("TC-POST-ADD-01")

    def test(self):
        self.webOperator.post_exist("TC-POST-ADD-01")

    def tearDown(self):
        self.webOperator.delete_post("TC-POST-ADD-01")
        self.webOperator.sign_out()
        self.webOperator.close_browser()

if __name__ == "__main__":
    unittest.main()
```

</td>
</tr>
<tr>
<td>3</td>
<td>TC-POST-EDIT-01</td>
<td>

```
import unittest
from MyWebOperator import MyWebOperator
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.common.by import By
from time import sleep


class testAddPost(unittest.TestCase):

    webOperator = MyWebOperator()
    driver = webOperator.driver
    wait = webOperator.wait

    def setUp(self):
        self.webOperator.login("demo@keystonejs.com", "demo")
        self.webOperator.add_post("TC-POST-EDIT-01")

    def test(self):
```

</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td><pre><code>   self.webOperator.into_edit_post_page("TC-POST-EDIT-01")
        postName_text_box = self.webOperator.driver.find_element(By.XPATH, "//*[@name='name']")
        postName_text_box.clear()
        postName_text_box.send_keys("TC-POST-EDITED")
        self.webOperator.click_save_post_button()
        self.webOperator.from_post_edit_page_to_post_page()
        self.webOperator.post_exist("TC-POST-EDITED")

    def tearDown(self):
        self.webOperator.delete_post("TC-POST-EDITED")
        self.webOperator.sign_out()
        self.webOperator.close_browser()

if __name__ == "__main__":
    unittest.main()
</code></pre></td></tr>
<tr><td>10</td><td>TC-USER-ADD-01</td><td><pre><code>import unittest
from MyWebOperator import MyWebOperator

class testAddPost(unittest.TestCase):

    webOperator = MyWebOperator()

    def setUp(self):
        self.webOperator.login("demo@keystonejs.com", "demo")

    def test(self):
        self.webOperator.add_user("FN", "LN", "userEmail@gmail.com", "userpassword123", "userpassword123")
        self.webOperator.user_exist("FN", "LN")

    def tearDown(self):
</code></pre></td></tr>
</table>

<table>
<tr><td></td><td></td><td><pre>
        # issue on delete user(can't delete user by UI)
        self.webOperator.sign_out()
        self.webOperator.close_browser()

if __name__ == "__main__":
    unittest.main()
</pre></td></tr>
</table>

4 Test Results

4.1 Test result

在 11 個 test case 中，成功的有 9 個 test case；失敗的兩個 case 分別為 TC-COMMENT-EDIT-01 與 TC-USER-ADD-01。

TC-COMMENT-EDIT-01 失敗原因在於無法再 comment 正確修改 post，drop down 選項 invisible 當嘗試 select element，這部分嘗試使用 selector hub 等外部工具，但也無法找出適當解決做法。

至於 TC-USER-ADD-01，經過使用認為這部分有 issue 存在，於 4.2 段落說明。

4.2 Find issue

User Feature 部分，使用者可以正確創立與正確顯示所創立的使用者，但是在刪除的操作上，即使經過刪除流程仍無法正確刪除使用者。關於此段落的驗證有確實進行創立與驗證使用者的實作，但是 tear down 部分先行省略，避免批次執行錯誤。

5 Summary

In Lab 4, **11 test cases have been designed and implemented using selenium and the path coverage technique(through activity diagram)**. In these cases, **9 test cases have passed, 1 can't implement and find 1 issue**. Thus, the test requirements described in Section 1 are satisfied.