Lab4 Report

Name:吳俊青
Student ID:108598014
Date:2020/06/08

1. Test Plan

1.1 Summary/Scope

Lab4 要撰寫測試腳本滿足以下五種 feature 以確保網站行為無誤

(使用 ISP 方法進行測試案例設計以達到盡可能的覆蓋)

- Post Features
- Comment Features
- Category Features
- Enquirie Feature
- User Feature

測試案例中每個操作皆需要有 assertion or wait component visible

以確保網站正確性。

1.2 Features to be tested

- Post Features
  - Create post on the Admin UI page
  - Edit post on the Admin UI page
  - Delete post on the Admin UI page
  - Search posts by keyword on the Admin UI page
- Comment Features
  - Create comment on Admin UI page
  - Edit comment on Admin UI page
  - Delete comment on Admin UI page
- Category Features
  - Create category on Admin UI page
  - Show posts of the specific category by pressing category name on the "Blog" page
- Enquirie Feature
  - Create enquiry on the "Contact" page
  - Delete enquiry on Admin UI page
  - User Feature
  - Create a new user on Admin UI page (Name, Email, Phone, and Password must be set when creating the new user)

1.3 Test environment and infrastructure

1. 安裝 VSCODE
2. 安裝 python3.6
3. 安裝 selenium library
4. 安裝 chrome driver
5. 架設 keystoneJS
6. Run testcase(python TestSuite.py)

1.4 Success criteria

每項操作後之 assertion or wait component visible 皆通過。

12 個測試案例全數 pass

1.5 Test approaches

使用 ISP 設計 Test Case 需先辨別輸入欄位或下拉選單可能的 partitions 和邊界值， 針對 input 設計條件，並設計可能組合以及邊界值，最終根據這些組合，去設計測試資料並實作出來，並且將所有可能組合 cover 住，確保 testcase 測試完整性。

我在這次的 lab 會先設想欄位可以輸入什麼值或下拉選單可以選擇那些選項最後去進行 assertion or wait component visible 確保操作符合預期以及網頁運作正確。

1.6 Test activities

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

No.	Activity Name	Plan hours	Schedule Date
1	Study Selenium library	2	2020/05/30
2	Study python unittest	1	2020/05/31
3	Study KeystoneJS	2	2020/05/31
4	Design test cases for the selected features	14	2020/05/31~2020/06/03
5	Implement test cases	12	2020/06/04~2020/06/06
6	Perform tests	1	2020/06/04~2020/06/06
7	Complete Lab4 report	8	2020/06/07~2020/06/08

2. Test Design

Use Case Section	Post
Use Case Name	Create post on the Admin UI page
Preconditions	Login and go to Posts page
Success Guarantee	創建的 post 皆出現在 Post 頁面上
Main Success Scenario	1. 選擇創建 post 2. 輸入欲創建之 post name 3. 創建並儲存 4. 回到 post 頁面 5. 確認創建之 post 皆存在於頁面上
Extensions	2a. 可以輸入不同類型 post name 1. web 正常回應 2. 繼續進行接下來操作 2b. 輸入不同類型 post name 造成error 1. web不正常顯示 2. 繼續進行接下來操作 5a. 創建之 post 皆存在於頁面 1.通過 post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3. 關閉瀏覽器 5b. 創建之 post 不存在於頁面 1. post assertion or wait component visible失敗 2. 關閉瀏覽器 3. 顯現 Fail reason
Input Field And Value (ISP)	對 post name 欄位做 ISP 設計 C1:輸入英文字母 valid: “[A-Z]*” invalid:非”[A-Z]*” C2:輸入數字 valid: “[0-9]*” invalid:非”[0-9]*” C3:只輸入純符號 valid: any character invalid:不是純符號構成 ---------- Test

	Requirements: {TFF, FTF, FFT, TTF } Infeasible TRs: { TFT, FTT, TTT, FFF} ---------- T1: “THIS IS GENE” covers {TFF} T2: “34567” covers {FTF} T3: “TEST1” covers {TTF} T4: “!@#$%” covers {FFT}

Use Case Section	Post
Use Case Name	Edit post on the Admin UI page
Preconditions	已存在一個或以上之 Post
Success Guarantee	Edit 的欄位值確實被儲存並顯示
Main Success Scenario	1. 選擇欲編輯post 2. 進入post detail 3. 編輯State欄位(可選之選項全測) 4. 編輯Author欄位(可選之選項全測) 5. 編輯Published Date欄位(該月日期 1-30 隨機挑選) 6. 編輯Content Brief欄位 7. 編輯Content Extended欄位 8. 儲存對post之修改 9. 確認各欄位顯示符合預期結果
Extensions	1a. 選擇post成功 1. web正常回應 2. 繼續進行接下來操作 1b. 選擇post失敗 1. 關閉瀏覽器 2. 顯現Fail reason 3a. 編輯State欄位成功 1. web正常顯示State欄位值 2. 繼續進行接下來操作 3b. 編輯State欄位失敗 1. web不正常顯示State欄位 2.繼續進行接下來操作 4a. 編輯Author欄位成功 1. web正常顯示Author 欄位值 2.繼續進行接下來操作 4b. 編輯Author欄位失敗

	1. web不正常顯示Author欄位值 2.繼續進行接下來操作 5a. 編輯Published Date該月日期在1-30之間random挑 1a.日期選到30號(該月存在30號) 1.web正常顯示 2.繼續進行接下來操作 1b. 日期選到30號(該月不存在30號) 1. 關閉瀏覽器 2. 顯現 Fail reason 6a. 編輯Content Brief欄位成功 1. web正常顯示Content Brief欄位值 2.繼續進行接下來操作 6b.編輯 Content Brief 欄位失敗 1. web不正常顯示Content Brief欄位值 2. 繼續進行接下來操作 7a. 編輯Content Extended欄位成功 1. web正常顯示Content Extended欄位值 2.繼續進行接下來操作 7b. 編輯Content Extended欄位失敗 1. web不正常顯示Content Extended欄位值 2.繼續進行接下來操作 9a. Edit 過之 post field value 皆在頁面上顯示 1.通過 Edit post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 9b. Edit過之後post field value未在頁面上顯示 1. Edit post assertion or wait component visible失敗 2.關閉瀏覽器 3.顯現 Fail reason
Input Field And Value (ISP)	對 post 各欄位做 ISP 設計

	C1:選擇之 State 欄位值存在 valid: select value in dropdown invalid: select value not in dropdown C2:輸入之 Author 欄位值存在 valid: select value in dropdown invalid: select value not in dropdown C3:選擇之 Published Date 欄位值存在 valid:該月有該日期 (e.g. 2020-05-30) invalid:該月無該日期 (e.g. 2020-02-30) ---------- Test Requirements: {TTF, TTT, TFT, TFF } Infeasible TRs: { FTT, FTF, FFT,FFF} ---------- T1: State: “Published” Author: “DEMO User” Published Date: 2020-05-30 covers {TTT} T2: State: “Published” Author: “DEMO User” Published Date: 2020-02-30(not exist) covers {TTF} T3: State: “Published” Author: “TEST”(not exist) Published Date: 2020-05-30 covers {TFT} T4: State: “Published” Author: “TEST”(not exist) Published Date: 2020-02-30(not exist) covers {TFF}

Use Case Section	Post
Use Case Name	Deletee post on the Admin UI page
Preconditions	Login and go to Posts page
Success Guarantee	刪除的 post 確實消失在 Posts 頁面上
Main Success Scenario	1. 選擇欲刪除之 post 2. 刪除該 post 3. 確認刪除之 post 確實消失於 Posts 頁面上
Extensions	2a. 可以依序刪除各post 1. web正常回應 2.繼續進行接下來操作 2b.不可以依序刪除各post 1. web不正常顯示 2.繼續進行接下來操作

	3a. 刪除之post確實消失於Posts頁面 1.通過post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3. 關閉瀏覽器 3b. 刪除之 post 依然存在於 Posts 頁面 1. post assertion or wait component visible 失敗 2.關閉瀏覽器 3.顯現 Fail reason
Input Field And Value (ISP)	此 testcase 無明顯可設計輸入之欄位故只有確認所創建的 4 個 Post 1. THIS IS GENE 2. TEST1 3. 34567 4. !@#$% 有依序且確實被刪除

Use Case Section	Post
Use Case Name	Search posts by keyword on the Admin UI page
Preconditions	Login and go to Posts page
Success Guarantee	搜尋的 post 確實出現在頁面上
Main Success Scenario	1. 點選搜尋欄位 2. 輸入欲搜尋之 post name (有用all of name and part of name 去搜尋e.g name= “TEST” 也有用 name=”TE”去測試) 3. 確認欲搜尋之 post 出現於頁面上
Extensions	2a.可以輸入不同類型 post name 1. web正常回應 2.繼續進行接下來操作 2b.輸入不同類型 post name 造成 error 1. web不正常顯示 2.繼續進行接下來操作 3a.創建之post皆存在於頁面 1.通過post assertion or wait component visible

	2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 3b.創建之post不存在於頁面 1. post assertion or wait component visible失敗 2. 關閉瀏覽器 3. 顯現 Fail reason
Input Field And Value (ISP)	對搜尋 post 欄位做 ISP 設計 C1:輸入英文字母 valid: ”[A-Z]*” invalid:非”[A-Z]*” C2:輸入數字 valid: ” [0-9]*” invalid: 非” [0-9]* C3:只輸入純符號 valid: any character invalid:不是純符號構成 ---------- ------ Test Requirements: {TFF, FTF, FFT, TTF } Infeasible TRs: { TFT, FTT, TTT, FFF} ---------- T1: {“THIS IS GENE” , “THIS”, “THIS IS” } covers {TFF} T2: {“34567”, “345”, “3” } covers {FTF} T3: “TEST1” covers {TTF} T4: {“!@#$%”, “!@#”, “!” } covers {FFT}

Use Case Section	Comment
Use Case Name	Create comment on Admin UI page
Preconditions	至少存在一個或以上的 Post 並已至 Comments 頁面
Success Guarantee	創建的 comment 出現在 Comments 頁面上
Main Success Scenario	1. 選擇創建 comment 2. 選擇 Author 欄位 3. 選擇要回覆之 Post 4. 按下創建 comment 5. 回到 Comments 頁面 6. 確認創建之 comment 存在於頁面上

Extensions	3a. 輸入要回覆之 post name 存在 1. web 正常回應 2.繼續進行接下來操作 3b.輸入要回覆之 post name 不存在造成 error 1.關閉瀏覽器 2.顯現 Fail reason 6a.創建之 comment 存在於頁面 1.通過 post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 6b.創建之 comment 不存在於頁面 1. post assertion or wait component visible 失敗 2.自動重選一個存在的 Post
Input Field And Value (ISP)	對 post name 欄位做 ISP 設計 C1: 輸入之 Post 是否存在 valid: ”THIS IS GENE” invalid: “TEST” ---------- Test Requirements: {T,F } ---------- T1: “THIS IS GENE” covers {T} T2: “TEST” covers {F}

Use Case Section	Comment
Use Case Name	Edit comment on Admin UI page
Preconditions	已存在一個或以上之 comment
Success Guarantee	Edit 的欄位值確實被儲存並顯示
Main Success Scenario	1. 選擇欲編輯 comment 2. 進入comment detail 3. 編輯State欄位(可選之選項全測) 4. 編輯Author欄位(可選之選項全測) 5. 編輯Content欄位 6. 儲存對post之修改 7. 確認各欄位顯示符合預期結果
Extensions	1a. 選擇 comment 成功 1. web正常回應

	2.繼續進行接下來操作 1b.選擇comment失敗 1.關閉瀏覽器 2.顯現Fail reason 3a.編輯State欄位成功 1. web 正常顯示State欄位值 2.繼續進行接下來操作 3b.編輯 State 欄位失敗 1. web不正常顯示State欄位 2.繼續進行接下來操作 4a. 編輯Author欄位成功 1. web 正常顯示Author欄位值 2.繼續進行接下來操作 4b. 編輯Author欄位失敗 1. web 不正常顯示Author欄位值 2.繼續進行接下來操作 5a. 編輯Content欄位成功 1. web正常顯示Content Brief欄位值 2.繼續進行接下來操作 5b. 編輯 Content Brief 欄位失敗 1. web 不正常顯示 Content Brief 欄位值 2. 繼續進行接下來操作 7a. Edit 過之 post field value 皆在頁面上顯示 1.通過 Edit post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3. 關閉瀏覽器 7b. Edit 過之 post field value 未在頁面上顯示 1. Edit post assertion or wait component visible 失敗 2. 關閉瀏覽器 3. 顯現 Fail reason
Input Field And Value (ISP)	對 comment 各欄位做 ISP 設計

	C1:選擇之 State 欄位值存在 valid: select value in dropdown invalid: select value not in dropdown C2:輸入之 Author 欄位值存在 valid: select value in dropdown invalid: select value not in dropdown C3:輸入 Content 欄位值不為英文字母 valid:輸入數字或符號 invalid: ” [A-Z]*” ---------- Test Requirements: {TTF, TTT, TFT, TFF } Infeasible TRs: { FTT, FTF, FFT,FFF} ---------- T1: State: “Published” Author: “DEMO User” Content: “12345” covers {TTT} T2: State: “Published” Author: “DEMO User” Content: “THIS IS GENE”(不為數字或符號) covers {TTF} T3: State: “Published” Author: “TEST”(not exist) Content: “!@#$%” covers {TFT} T4: State: “Published” Author: “TEST”(not exist) Content: “THIS IS GENE”(不為數字或符號) covers {TFF}

Use Case Section	Comment
Use Case Name	Deletee comment on the Admin UI page
Preconditions	Login and go to Comments page
Success Guarantee	刪除的 post 確實消失在 Posts 頁面上
Main Success Scenario	1. 選擇欲刪除之 comment 2. 刪除該 comment 3. 確認刪除之 comment 確實消失於Comments頁面上
Extensions	2a.可以刪除comment 1. web正常回應 2.繼續進行接下來操作

	2b.不可以刪除comment 1. web 不正常顯示 2.繼續進行接下來操作 3a. 刪除之comment確實消失於Comments頁面 1.通過 post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 3b.刪除之comment依然存在於Comments頁面 1. post assertion or wait component visible 失敗 2.關閉瀏覽器 3.顯現Fail reason
Input Field And Value (ISP)	此 testcase 無明顯可設計輸入之欄位故只有確認所創建之 comment 有確實被刪除

Use Case Section	Category
Use Case Name	Create category on Admin UI page
Preconditions	Login and go to Categories page
Success Guarantee	創建的 Category 皆出現在 Categories 頁面上
Main Success Scenario	1. 選擇創建Category 2. 輸入欲創建之Category name 3. 創建並儲存 4. 回到Categories頁面 5. 確認創建之Category皆存在於頁面上
Extensions	2a.可以輸入不同類型 Category name 1. web正常回應 2.繼續進行接下來操作 2b.輸入不同類型Category name造成error 1. web不正常顯示 2.繼續進行接下來操作 5a.創建之 Category 皆存在於頁面 1.通過 post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器

	5b.創建之 Category 不存在於頁面 1. post assertion or wait component visible 失敗 2.關閉瀏覽器 3.顯現Fail reason
Input Field And Value (ISP)	對 Category name 欄位做 ISP 設計 C1:輸入英文字母 valid: ” [A-Z]*” invalid:非” [A-Z]* C2:輸入數字 valid: ”[0-9]*” invalid:非” [0-9]*” C3:只輸入純符號 valid: any character invalid:不是純符號構成 ---------- Test Requirements: {TFF, FTF, FFT, TTF } Infeasible TRs: { TFT, FTT, TTT, FFF} ---------- T1: “Test” covers {TFF} T2: “34567” covers {FTF} T3: “Test1”covers {TTF} T4: “!@#$%” covers {FFT}

Use Case Section	Category
Use Case Name	Show posts of the specific category by pressing category name on the "Blog" page
Preconditions	Login and go to Posts page
Success Guarantee	創建的 Post 出現在該 Category 頁面下
Main Success Scenario	1. 選擇創建post 2. 輸入Post Name 3. 創建post 4. 選擇state 5. 選擇category 6. 進入blog tab 下 7. 點選該post所屬之category 8. 確認該post正確顯現
Extensions	2a.輸入要創建之post name合乎規則 1. web正常回應 2.繼續進行接下來操作

	2b.輸入要創建之post name不合乎規則造成error 1.關閉瀏覽器 2.顯現Fail reason 4a.選擇State欄位成功 1. web 正常顯示State欄位值 2.繼續進行接下來操作 4b.選擇State欄位失敗 1. web不正常顯示State欄位 2.繼續進行接下來操作 5a. 選擇Category欄位成功 1. web 正常顯示Author欄位值 2.繼續進行接下來操作 5b.選擇 Category 欄位失敗 1. web 不正常顯示Author欄位值 2.繼續進行接下來操作 8a.創建之comment存在於頁面 1.通過post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 8b.創建之comment不存在於頁面 1. post assertion or wait component visible失敗 2.自動重選一個存在的Post
Input Field And Value (ISP)	對 post name 欄位做 ISP 設計 C1:輸入之 State 可顯現在 blog valid: ” Published” (選擇 published 才可顯現出來) invalid: {“Drafted”,“Archived”} C2:輸入之 Category 存在 valid: “TEST”, ---------- Test Requirements: {TT, TF, FT, FF } ---------- T1: State: ” Published” Category: “Test” covers {TT}

	T2: State: ” Published” Category: “GENE”(Not Exist) covers {TF} T3: State: ” Draft” Category: “Test” covers {FT} T3: State: ” Draft” Category: “GENE”(Not Exist) covers {FF}

Use Case Section	Enquirie
Use Case Name	Create enquiry on the "Contact" page
Preconditions	go to Contact page
Success Guarantee	成功創建 Enquirie
Main Success Scenario	1. 輸入Name欄位 2. 輸入Email欄位 3. 輸入Phone欄位 4. 選擇Regarding 5. 輸入Message 6. 提交Enquirie 7. 確認提交成功
Extensions	1a.輸入要創建之Name成功 1. web正常回應 2.繼續進行接下來操作 1b.輸入要創建之Name失敗 1. web不正常顯示Name欄位 2.繼續進行接下來操作 2a.輸入Email欄位成功 1. web正常顯示Email欄位值 2.繼續進行接下來操作 2b.輸入Email欄位失敗 1. web不正常顯示Email欄位 2.繼續進行接下來操作 3a. 輸入Phone欄位成功 1. web正常顯示Phone欄位值 2.繼續進行接下來操作

	3b.輸入Phone欄位失敗 1. web不正常顯示Phone欄位 2.繼續進行接下來操作 4a.選擇Regarding欄位成功 1. web正常顯示State欄位值 2.繼續進行接下來操作 4b.選擇Regarding欄位失敗 1.關閉瀏覽器 2.顯現Fail reason 5a.輸入之Message合乎規則 1. web 正常回應 2.繼續進行接下來操作 5b.輸入之Message不合乎規則造成error 1.關閉瀏覽器 2.顯現 Fail reason 7a.確認Enquirie提交成功 1.通過post assertion or wait component visible 2.頁面暫停幾秒 (通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 7b.發現 Enquirie 提交失敗 1. post assertion or wait component visible 失敗 2.關閉瀏覽器 3.顯現 Fail reason
Input Field And Value (ISP)	對 Category name 欄位做 ISP 設計 C1: phone 欄位非手機號碼 valid: ”0426352552”(市內電話) invalid: ”09123456789”(手機) C2: Regarding 都可選 valid: { ” Just leaving a message”, “I've got a question” “Something else...”} Invalid: someone can not be selected C3: Message 純英文字 Valid: “Test” invalid: {“12345”, ”Test1 ”, “!@#$%”}

	---------- Test Requirements: {TTT, FTF, FTT, TTF } Infeasible TRs: { TFT, FFT, TFF, FFF} (目前還未發現不可點選情形) ---------- T1: phone: 0426352552 Regarding: ” Just leaving a message” Message: “EDIT TEST” covers {TTT} T2: phone: 09123456789(手機) Regarding: ” Just leaving a message” Message: “12345” covers {FTF} T3: phone: 09123456789 (手機) Regarding: ” Just leaving a message” Message: “EDIT TEST” covers {FTT} T4: phone: 0426352552 Regarding: ” Just leaving a message” Message: “12345” covers {TTF}

Use Case Section	Enquirie
Use Case Name	Delete enquiry on Admin UI page
Preconditions	Login and go to Enquiries page
Success Guarantee	刪除的 Enquirie 確實消失在 Enquiries 頁面上
Main Success Scenario	1. 選擇欲刪除之Enquirie 2. 刪除該Enquirie 3. 確認刪除之Enquirie確實消失於Enquiries頁面上
Extensions	2a. 可以刪除Enquirie 1. web正常回應 2.繼續進行接下來操作 2b. 不可以刪除Enquirie 1. web不正常顯示 2.繼續進行接下來操作 3a. 刪除之Enquirie確實消失於Enquiries頁面 1.通過 post assertion or wait component visible 2.頁面暫停幾秒

	(通過 assertion 才會暫停方便人工查看) 3.關閉瀏覽器 3b. 刪除之Enquirie依然存在於Enquiries頁面 1. post assertion or wait component visible失敗 2.關閉瀏覽器 3.顯現 Fail reason
Input Field And Value (ISP)	此 testcase 無明顯可設計輸入之欄位故只有確認所創建之 Enquirie 有確實被刪除

Use Case Section	User
Use Case Name	Create a new user on Admin UI page (Name, Email, Phone, and Password must be set when creating the new user)
Preconditions	go to Users page
Success Guarantee	成功創建 User
Main Success Scenario	1. 輸入First Name and Last Name 欄位 2. 輸入Email欄位 3. 輸入Password欄位 4. 輸入Confirm Password欄位 5. 創建User 6. 進入User detail 7. 輸入Phone欄位 8. 儲存變更 9. 回到Users頁面 10.確認User已創建
Extensions	1a. 輸入要創建之First Name and Last Name成功 1. web正常回應 2.繼續進行接下來操作 1b. 輸入要創建之First Name and Last Name失敗 1. web不正常顯示Name 欄位 2.繼續進行接下來操作 2a. 輸入Email欄位成功 1. web正常顯示Email欄位值 2.繼續進行接下來操作 2b. 輸入Email欄位失敗 1. web不正常顯示Email欄位 2.繼續進行接下來操作

	3a. 輸入Password欄位成功 1. web正常顯示Password欄位值 2.繼續進行接下來操作 3b. 輸入Password欄位失敗 1. web不正常顯示Password欄位 2.繼續進行接下來操作 4a. 輸入Confirm Password欄位和Password一致 1. web正常顯示State欄位值 2.繼續進行接下來操作 4b. 輸入Confirm Password欄位和Password一致 1.關閉瀏覽器 2.顯現 Fail reason 7a. 輸入之Phone合乎規則 1. web正常回應 2.繼續進行接下來操作 7b. 輸入之Phone不合乎規則造成error 1.關閉瀏覽器 2.顯現 Fail reason
Input Field And Value (ISP)	對 User 各欄位做 ISP 設計 C1: phone 欄位非手機號碼 valid: ”0426352552” invalid: ”09123456789”(手機) C2: Email 不重複 valid: “qq55555520qq@gmail.com” invalid: Email already exist C3: password 只包含英文或數字 Valid: {“12345678”, “geneisNumber1”} invalid: “!@#$%” ---------- Test Requirements: {TTT, FTF, FTT, TTF, TFT, FFT, TFF, FFF } ---------- T1: phone: 0426352552 Email: ”qq55555520qq@gmail.com” password: “geneisNumber1” covers {TTT} T2: phone: 09123456789(手機) Email: ” qq55555520qq@gmail.com” password: “!@#$%”

	covers {FTF} T3: phone: 09123456789(手機) Email: ” qq55555520qq@gmail.com” password: “geneisNumber1” covers {FTT} T4: phone: 0426352552 Email: ” qq55555520qq@gmail.com” password: “!@#$%” covers {TTF} T5: phone: 0426352552 Email: qq55555520qq@gmail.com(已存在) password: “geneisNumber1” covers {TFT} T6: phone: 09123456789(手機) Email: qq55555520qq@gmail.com(已存在) password: “geneisNumber1” covers { FFT } T7: phone: 0426352552 Email: qq55555520qq@gmail.com(已存在) password: “!@#$%” covers {TFF} T8: phone: 09123456789(手機) Email: qq55555520qq@gmail.com(已存在) password: “!@#$%” covers {TFF}

3. Test Implementation

The design of test cases specified in Section 2 was implemented using Python and Selenium Library. The test scripts of 3 selected test cases are given below.

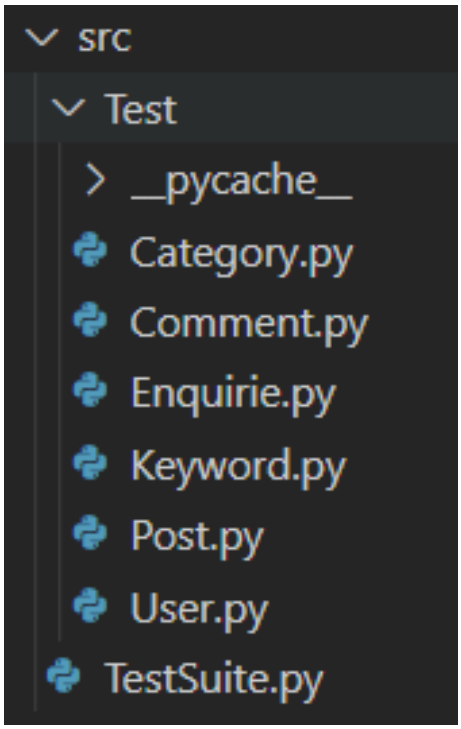
The rest of the test script implementations can be found in the link.

Link: https://stv.csie.ntut.edu.tw/108598014/GeoProject/tree/master/LabReport/Lab4/src

No.	Test feature	Source test code
1	Post	Post.py
2	Comment	Comment.py
3	Enquirie	Enquirie.py

4. Test Results

4.1 Selenium test scripts



4.2 execution results

```
C:\Users\11041\Desktop\Lab4\src>python ./Testsuite.py
test_create_post (Test.Post.test_post) ...
DevTools listening on ws://127.0.0.1:54771/devtools/browser/a3c7a48f-f74c-437b-a079-e83ee231ad0d
ok
test_edit_post (Test.Post.test_post) ...
DevTools listening on ws://127.0.0.1:54805/devtools/browser/2b52e252-0957-428f-b14f-2d88b0e57f04
ok
test_search_post (Test.Post.test_post) ...
DevTools listening on ws://127.0.0.1:54834/devtools/browser/21cc31f8-29d4-49c5-94ec-1d0fc87260c6
ok
test_delete_post (Test.Post.test_post) ...
DevTools listening on ws://127.0.0.1:54866/devtools/browser/178e08ae-ed35-4903-bcd8-6a1a74aa0b87
ok
test_create_comment (Test.Comment.test_comment) ...
DevTools listening on ws://127.0.0.1:54899/devtools/browser/4a46480f-2d7c-4210-a4c1-542e54d184c1
ok
test_edit_comment (Test.Comment.test_comment) ...
DevTools listening on ws://127.0.0.1:54930/devtools/browser/6babecad-6417-4205-8101-d2933d386b59
ok
test_delete_comment (Test.Comment.test_comment) ...
DevTools listening on ws://127.0.0.1:54959/devtools/browser/c2f7642c-a95f-47c0-828a-1aaa8e3d6cfb
ok
test_create_category (Test.Category.test_category) ...
DevTools listening on ws://127.0.0.1:54988/devtools/browser/c9762232-b673-407f-a852-8760a1bd731e
ok
test_show_post (Test.Category.test_category) ...
DevTools listening on ws://127.0.0.1:55017/devtools/browser/8651715f-6820-403d-b3a5-bb802bcc0a9a
ok
test_create_enquiry (Test.Enquirie.test_enquire) ...
DevTools listening on ws://127.0.0.1:55049/devtools/browser/05d035e0-afb2-4e00-a31f-80e52d22acbc
ok
test_delete_enquiry (Test.Enquirie.test_enquire) ...
DevTools listening on ws://127.0.0.1:55082/devtools/browser/c7adc30f-5f5a-4cb0-b643-5d5bc363e7d8
ok
test_create_user (Test.User.test_user) ...
DevTools listening on ws://127.0.0.1:55112/devtools/browser/c6fa0dcb-78ca-47f8-97bf-c3513f794bc6
ok

----------------------------------------------------------------------
Ran 12 tests in 211.270s

OK
```

5. Summary

在這次的 Lab 中使用到 selenium library 撰寫 test scripts，相對於我們實驗室中自己所使用到的 robot framework 有著不一樣的體驗，也讓我更加了解測試工具是非常多元化的，雖然直接利用底層寫 python code 不比已經可以直接做使用的 robot framework 框架穩定，但是運作的效率會相對較快速。

此外我也更了解如何抓取 xpath 以及如何寫出高重用性的 Function，提高使用率不必遇到新的功能就寫新的 Function 產生過多相似 code，造成維護困難，整體來說讓我學習到很多。