

Lab 4 Report

Name: 陳日揚

Student ID: 106598014

Date: 2018/5/21

1 Test Plan

1.1 Summary

因應現今網頁應用程式(Web Applications)發展快速且應用範圍廣泛，本次 Lab4 實驗我們以「Socify」網站作為測試的目標。Socify 為一功能類似 Facebook、微博及噗浪的社群網站，使用者在註冊帳號後便可透過該網站發布個人的心情狀態貼文、舉辦活動、追蹤朋友並對朋友的動態發表評論及按讚，Socify 為一個方便架設與使用開源社群交友平台。

根據題目要求，本次網頁的測試必須使用「Selenium」這套網頁的自動化測試工具來進行實作。Selenium 最大的優點為可透過視覺化的測試過程及錄製、回放等的功能來讓測試人員方便地完成測試的目的。為了同時學習 Python 程式語言及 Selenium 工具，本次實驗我將採取 Selenium with Python 來進行測試。

1.2 Feature to be tested

根據實驗要求及對 Socify 應用程式進行分析過後，我認為登入/登出功能較不重要因此不列入本次的待測範圍內。另外，Find Friends 功能因為實際的系統並未提供搜尋欄位讓使用者輸入，因此亦不列入待測範圍內。

以下列出本次實驗待測的 Features：

(1) Status feature:

- Create new status
- Edit status
- Delete status

(2) Event feature:

- Create new event
- Edit event
- Delete event

(3) Post feature:

- Like post
- Unlike post
- Comment post
- Delete comment

(4) Follow friends feature:

- Follow friend
- Unfollow friend

(5) Edit profile feature:

- Edit profile

1.3 Success criteria of completing the test

因為本次設計的層級較高階，屬於 UI 測試及整合測試階段，因此我將採用 User scenario 作為測試案例設計的方法，而測試案例通過的標準(也就是系統否能夠 Release 的標準)為每個分析過後的 User scenario(包含 Happy paths 及 Alternative paths)都至少要被一個測試案例所包含，並且測試結果應全數通過。

1.4 Test environment and infrastructure

(1) PC(Personal Computer) hardware:

- CPU: Intel Core i7-4790 CPI 3.60GHz
- RAM: 4GB
- SSD: 250G
- HD: 1000G

(2) PC(Personal Computer) software:

- OS: Windows 10 Enterprise
- PyCharm IDE: 2018.1
- Selenium: 3.12.0
- Python: 3.6.5

1.5 Test Approach

如同 *section 1.3* 所述，本次測試屬於高階的整合測試階段，因此我們將先觀察使用者操作系統時的 **User scenarios**，接著便分析出這些 User scenarios 的 **preconditions**、**input actions**、**expected output** 及例外發生時的 **alternative scenarios**。最終，將分析出來的結果整合成執行測試的 Test scenarios，詳細的 Test scenarios 請參照 *section 2*。

1.6 Test Tasks

No.	Activity Name	Plan hours	Schedule Date
1	Familiar with Socify web app	1 Hr	2018/5/19
2	Analysis user scenario and design test scenarios	3 Hr	2018/5/19
3	Implement test cases & perform tests	10 Hr	2018/5/20- 2018/5/21
4	Complete Lab4 report	3 Hr	2018/5/21

2 Test Design

(1) Status feature:

Create new status	
Precondition	● 使用者已登入本系統。
User scenario	1. 使用者輸入 status 的內容。 2. 使用者確認發布 status。 3. 系統顯示使用者發佈的最新 status。
Expected output	● 使用者成功地新增 status。
Alternative scenario	1a. 使用者選擇夾帶圖片。 1. 使用者選擇欲上傳的圖片。 2. 系統顯示使用者夾帶之圖片縮圖預覽。 2a. 使用者在沒輸入內容的情況下確認送出 status。 1. 系統提示使用者在沒輸入內容的情況下無法送出 status。 2. 回到 User scenario 步驟 1。

Edit status	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者已發布欲編輯的 status。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲編輯的 status。 2. 輸入新的狀態內容。 3. 使用者確認更新 status。 4. 系統顯示使用者更新後的 status。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地編輯 status。
Alternative scenario	<ol style="list-style-type: none"> 3a. 使用者在沒輸入內容的情況下確認送出 status。 <ol style="list-style-type: none"> 1. 系統將不會更新狀態的內容。 2. 系統顯示更新前的 status 內容。

Delete status	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者已發布欲刪除的 status。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲刪除的 status。 2. 系統頁面更新並將不顯示使用者已刪除的 status。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地刪除 status。
Alternative scenario	None

(2) Event feature:

Create new event	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇進入 create event 頁面。 2. 使用者輸入 event name 及 event date。 3. 使用者確認建立 event。 4. 系統顯示使用者發佈的最新 status。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地新增 event。
Alternative scenario	<ol style="list-style-type: none"> 3a. 使用者在沒輸入 event name 或 event date 的情況下確認送出 status。 <ol style="list-style-type: none"> 1. 系統將不會新增 event，並且將畫面回到 create event 頁面。

Edit event	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者已發布欲編輯的 event。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲編輯的 event。 2. 輸入新的狀態 event name 及 event date。 3. 使用者確認更新 event。 4. 系統顯示使用者更新後的 event。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地新增 event。
Alternative scenario	<ol style="list-style-type: none"> 3a. 使用者在沒輸入 event name 或 event date 的情況下確認送出 status。 <ol style="list-style-type: none"> 1. 系統將不會新增 event，並且將畫面回到舊的 event 貼文頁面。

Delete event	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者已發布欲刪除的 event。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲刪除的 event。 2. 系統頁面更新並將不顯示使用者已刪除的 event。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地刪除 status。
Alternative scenario	None

(3) Post feature:

※ Post 包含 status 及 event。

Like post	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者或朋友已發布欲按讚的 post。
User scenario	<ol style="list-style-type: none"> 1. 使用者對 post 進行按讚。 2. 系統頁面更新並顯示按讚後的 post。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地按讚 post，且按讚按鈕中的文字從「like」更新為「unlike」。
Alternative scenario	None

Unlike post	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者或朋友已發布欲取消按讚的 post。
User scenario	<ol style="list-style-type: none"> 1. 使用者對 post 取消按讚。 2. 系統頁面更新並顯示取消按讚後的 post。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地取消按讚 post，且按讚按鈕中的文字從「unlike」更新為「like」。
Alternative scenario	None

Comment post	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者或朋友已發布欲取消按讚的 post。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲撰寫 comment 的文章。 2. 使用者輸入 comment 內容。 3. 使用者送出 comment。 4. 系統頁面更新並顯示 post 的 comment。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地對 post 撰寫 comment。
Alternative scenario	<ol style="list-style-type: none"> 1a. 使用者在沒輸入內容的情況下送出 comment。 <ol style="list-style-type: none"> 1. 系統將不會新增 comment，並且將畫面回到舊的 post 頁面。

Delete comment	
Precondition	<ul style="list-style-type: none"> ● 使用者已登入本系統。 ● 使用者已發布欲刪除的 comment。
User scenario	<ol style="list-style-type: none"> 1. 使用者選擇欲刪除的 comment。 2. 系統頁面更新並將不顯示使用者已刪除的 comment。
Expected output	<ul style="list-style-type: none"> ● 使用者成功地刪除 comment。
Alternative scenario	None

(4) Follow friends feature:

Follow friends	
Precondition	<ul style="list-style-type: none">● 使用者已登入本系統。
User scenario	<ol style="list-style-type: none">1. 使用者選擇進入 Find Friends 頁面。2. 使用者對欲追蹤的朋友點選「follow」。3. 系統頁面更新並顯示使用者已 Follow 上一步驟所追蹤的朋友。
Expected output	<ul style="list-style-type: none">● 使用者成功地 follow 朋友，且該朋友旁邊按鈕中的文字從「follow」更新為「unfollow」。
Alternative scenario	None

Unfollow friends	
Precondition	<ul style="list-style-type: none">● 使用者已登入本系統。● 使用者已經有 Follow 欲 Unfollow 的朋友。
User scenario	<ol style="list-style-type: none">1. 使用者選擇進入 Friends 頁面。2. 使用者對欲取消追蹤的朋友點選「unfollow」。3. 系統頁面更新並顯示使用者已 unfollow 上一步驟欲取消追蹤的朋友。
Expected output	<ul style="list-style-type: none">● 使用者成功地 unfollow 朋友，且該朋友旁邊按鈕中的文字從「unfollow」更新為「follow」。
Alternative scenario	None

(5) Edit profile feature:

Edit profile	
Precondition	<ul style="list-style-type: none">● 使用者已登入本系統。
User scenario	<ol style="list-style-type: none">1. 使用者選擇進入 Edit profile 頁面。2. 使用者輸入欲更新的 profile 內容(包括 Avatar image、cover image、Name、About、Location、Phone number、Date of Birth 及 Sex)。3. 使用者確認送出更新。4. 系統將畫面導向最新動態頁面。
Expected output	<ul style="list-style-type: none">● 使用者成功地更新 profile 內容。

Alternative scenario	<p>3a. 若使用者在沒輸入任何資訊的情況下(所有欄位內容皆為空)送出更新。</p> <ol style="list-style-type: none"> 1. 系統將不會更新 profile 內容，並將畫面導向 Edit profile 畫面要求使用者重新輸入內容。 2. 回到 User scenario 步驟 2.
----------------------	---

3 Test Implementation

本次 Lab 採用 Selenium 自動化測試工具搭配 Python 及 Google Chrome 的 WebDrive，依照不同的 Feature 及相對應的 Test scenario 總共撰寫了 4 個 Test class 以及 16 個 Test cases。在撰寫完 test cases 後，我使用 Test loader 將 Test cases 載入到 Test suite 當中，最後再透過 TextTestRunner 的 run() Method 來執行 test suite。

詳細的 test implementation 程式碼請參考下列的 GitHub 超連結：

(1) StatusTest.py :

※ 包含 Status feature 及 Post feature，共 9 個 Test cases。

<https://github.com/JihYangChen/Socify-Test/blob/master/StatusTest.py>

(2) EventTest.py :

※ 包含 Event feature，共 4 個 Test cases。

<https://github.com/JihYangChen/Socify-Test/blob/master/EventTest.py>

(3) FriendTest.py :

※ 包含 Follow friends，共 2 個 Test cases。

<https://github.com/JihYangChen/Socify-Test/blob/master/FriendTest.py>

(4) ProfileTest.py :

※ 包含 Follow friends，共 1 個 Test cases。

<https://github.com/JihYangChen/Socify-Test/blob/master/ProfileTest.py>

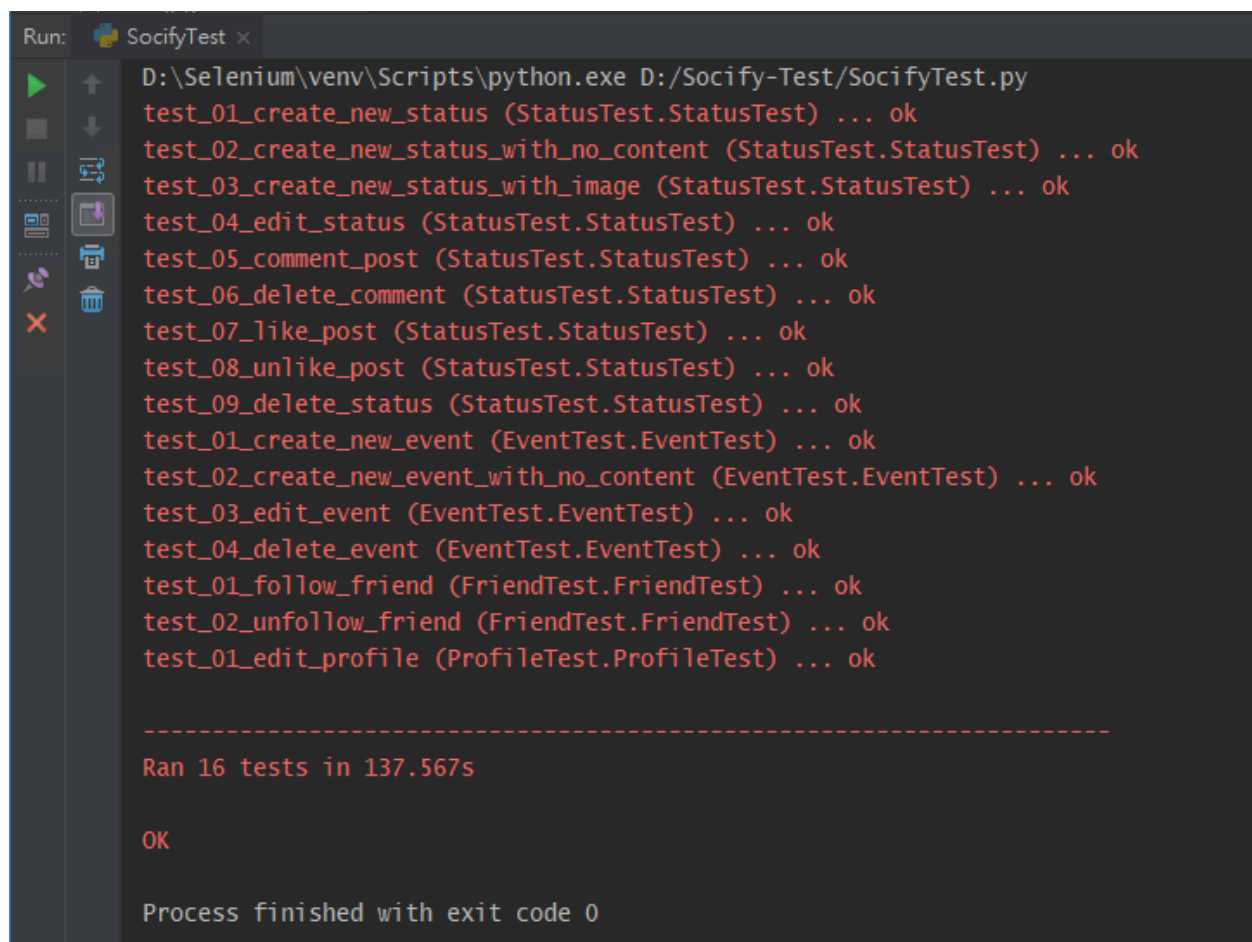
(5) SocifyTest.py :

※ 包含 Test Suite 的包裝及執行。

<https://github.com/JihYangChen/Socify-Test/blob/master/SocifyTest.py>

4 Test Results

本次測試 *section 2* 所分析出來的 13 個 Test scenario 皆有被 *section 3* 所設計的 16 個 Test cases 包含，最終的測試執行結果**全數通過**，執行結果可參考下方截圖：



```
Run: SocifyTest x
D:\Selenium\venv\Scripts\python.exe D:/Socify-Test/SocifyTest.py
test_01_create_new_status (StatusTest.StatusTest) ... ok
test_02_create_new_status_with_no_content (StatusTest.StatusTest) ... ok
test_03_create_new_status_with_image (StatusTest.StatusTest) ... ok
test_04_edit_status (StatusTest.StatusTest) ... ok
test_05_comment_post (StatusTest.StatusTest) ... ok
test_06_delete_comment (StatusTest.StatusTest) ... ok
test_07_like_post (StatusTest.StatusTest) ... ok
test_08_unlike_post (StatusTest.StatusTest) ... ok
test_09_delete_status (StatusTest.StatusTest) ... ok
test_01_create_new_event (EventTest.EventTest) ... ok
test_02_create_new_event_with_no_content (EventTest.EventTest) ... ok
test_03_edit_event (EventTest.EventTest) ... ok
test_04_delete_event (EventTest.EventTest) ... ok
test_01_follow_friend (FriendTest.FriendTest) ... ok
test_02_unfollow_friend (FriendTest.FriendTest) ... ok
test_01_edit_profile (ProfileTest.ProfileTest) ... ok

-----
Ran 16 tests in 137.567s

OK

Process finished with exit code 0
```

5 Conclusion

本次 Lab 是我第一次撰寫 Web Application 的測試，在這次的測試實作過程當中我學習到了 Selenium 與 Python。我認為 Web Application 測試最有挑戰的地方在於如何正確地定位到 DOM elements 以及如何取得 element 內的狀態或內容，本次我利用 xpath 下條件的方式來尋找 DOM elements。本次測試所撰寫的測試結果均全數通過，且符合 *section 1.3* 所訂定的 Success criteria。