

國立臺北科技大學

資訊工程系

軟體測試與驗證

Software Test Plan



Turbo Editor

Team#6

專案成員：古兆瑋（108598019）

陳冠穎（108598034）

劉孝忠（108598044）

民國 109 年 6 月 23 日

Table of Contents

Table of Contents	2
Change History	3
1. Objectives and Overview	4
1.1 Test Plan Identifier	4
1.2 Test Scope	4
1.3 Test Approach	4
1.4 Test Deliverables	4
2. Test Items	5
2.1 Description	5
2.2 Use Cases	5
2.3 Features To Be Tested	9
2.4 Features Not To Be Tested	10
2.5 Item Pass/Fail Criteria	10
3. Environment Needs	11
3.1 Operational Environment	11
3.2 Hardware Specification and Configuration	11
3.3 Software Specification and Configuration	11
4. Personnel and Responsibility	12
4.1 Role and Responsibilities	12
4.2 Staffing and Training	12
5. Test Schedule	13
5.1 Schedule and Milestones	13
Glossary	14
Reference	15

Change History

Version	Primary Author(s)	Description of Version	Date Completed
1.0.0	古兆瑋、陳冠穎、劉孝忠	樣式及內容初稿。	2020/04/08
1.1.0	古兆瑋、陳冠穎、劉孝忠	依教授於期中報告提出之建議，於 Phase 3 修正 2.5 Item Pass/Fail Criteria；以及依更新後之行事曆修正 5. Test Schedule。	2020/05/20
1.2.0	古兆瑋、陳冠穎、劉孝忠	修正測試工具為 Espresso 及 UI Automator；以及依更新後之行事歷修正 5. Test Schedule。	2020/06/23



1. Objectives and Overview

1.1 Test Plan Identifier

NTUT-108-STV-TurboEditor_STP。

1.2 Test Scope

本次計畫將以一個在 Android 系統上的「Turbo Editor」APP 進行測試，範圍以 APP 所有功能為限。

1.3 Test Approach

本次計畫以 APP UI 測試自動化框架去進行，透過撰寫腳本進行自動化測試。

1.4 Test Deliverables

本次針對 Turbo Editor 的測試，預計會有以下 3 種產出物：

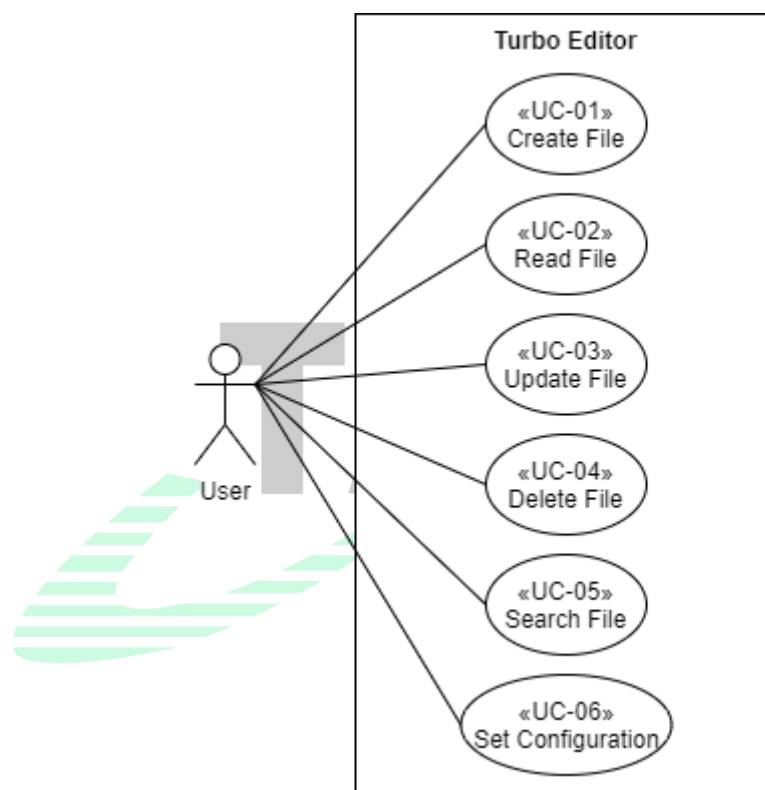
- ✧ Software Test Plan (STP)：(即為本文件)
- ✧ Test Design Specifications (TDS)：針對 Software Test Plan (STP)中的 Test Items 設計 Test Cases，並定義其測試的 Feature Pass/Fail Criteria。
- ✧ Test Case Specifications (TCS)：針對 Software Test Plan (STP)、Test Design Specifications (TDS)中的 Test Items/Test Cases 設計其 Input Specifications、Output Specifications，並記錄其 Intercase Dependencies。
- ✧ Test Result Summary (TRS)：執行那些根據上述文件的需求、設計、實作測試案例，其測試結果與分析報告。

2. Test Items

2.1 Description

本次測試將會以 SUT 中，以 Turbo Editor 的 Feature List 中，較常使用到的主要項目作為 Test Items，並將其歸類為相對應的 Use Cases，使往後 Test Cases 設計時有相對應的 Use Cases。

2.2 Use Cases



No	UC-01
Use Case Name	Create File
Scope	Turbo Editor
Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to Create a New File.
Preconditions	None.
Success Guarantee	A File Created or Overwrote.
Main success Scenario	1. User send “New File” request.

	<ol style="list-style-type: none"> 2. User input Filename. 3. User send “SAVE” request. 4. Turbo Editor will create a new File.
Extensions	<p>2a. User wants to overwrite an existed File.</p> <ol style="list-style-type: none"> 1. User send “SAVE” request. 2. Turbo Editor will ask the User whether to overwrite the existed File. 3. User send “OK” request. <p>2b. A File with the Filename that User just input is already exist.</p> <ol style="list-style-type: none"> 1. User send “SAVE” request. 2. Turbo Editor will create a new File but name it after a new alias.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

No	UC-02
Use Case Name	Read File
Scope	Turbo Editor
Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to Read an existed File.
Preconditions	None.
Success Guarantee	A File’s content showed.
Main success Scenario	<ol style="list-style-type: none"> 1. User select an existed File. 2. User send “Open a File” request. 3. File’s content and its Info showed.
Extensions	None.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

No	UC-03
Use Case Name	Update File
Scope	Turbo Editor

Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to Update an existed File.
Preconditions	None.
Success Guarantee	A File Updated.
Main success Scenario	<ol style="list-style-type: none"> 1. User select an existed File. 2. User send “Update” request. 3. User edit File’s content. 4. User send “Save” request. 5. A File Updated to its new state.
Extensions	<ol style="list-style-type: none"> 2a. User send “Rename” request. <ol style="list-style-type: none"> 1. User Input new Filename. 2. User send “SAVE” request. 3. File Updated to its new Filename. 3a. User send “Undo” request. <ol style="list-style-type: none"> 1. File recover to its previous state. 3b. User send “Redo” request. <ol style="list-style-type: none"> 1. File recover to its next state. 4a. User send “Save as” request. <ol style="list-style-type: none"> 1. User select Folder. 2. User Input new Filename. 3. User send “SAVE” request. 4. File Updated to its new Path. 4b. User send “SAVE” request. <ol style="list-style-type: none"> 1. File Updated. 4c. User send “CANCEL” request. <ol style="list-style-type: none"> 1. File not Updated.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

No	UC-04
Use Case Name	Delete File
Scope	Turbo Editor
Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to Delete an existed File.

Preconditions	None.
Success Guarantee	A File Deleted.
Main success Scenario	<ol style="list-style-type: none"> 1. User select an existed File. 2. User send “DELETE” request. 3. User send “YES” request. 4. A File Deleted.
Extensions	3a. User send “NO” request. <ol style="list-style-type: none"> 1. File not Deleted.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

No	UC-05
Use Case Name	Search File
Scope	Turbo Editor
Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to Search an existed File.
Preconditions	None.
Success Guarantee	A File Searched.
Main success Scenario	<ol style="list-style-type: none"> 1. User input File’s Info. 2. User send “Search” request. 3. A File Searched.
Extensions	3a. A File with that Searching Info do not exist. <ol style="list-style-type: none"> 1. File not Searched.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

No	UC-06
Use Case Name	Set Configuration
Scope	Turbo Editor
Level	User-goal
Primary Actor	User
Stakeholders and Interests	User: Intent to set Turbo Editor’s Configuration.

Preconditions	None.
Success Guarantee	A Configuration Set.
Main success Scenario	<ol style="list-style-type: none"> 1. User send “Preference” request. 2. User Set or Toggle Configuration. 3. User send “Set Configuration” request. 4. Configuration Set.
Extensions	<ol style="list-style-type: none"> 2a. User send “Default” request. <ol style="list-style-type: none"> 1. Configuration reset to its initial state. 3a. User send “Cancel” request. <ol style="list-style-type: none"> 1. Configuration not Set.
Special Requirements	None.
Technology	None.
Frequency of Occurrence	Often.
Miscellaneous	None.

2.3 Features To Be Tested

Feature ID	Description	Use Case
F-T-01	Create New File	Create File
F-T-02	Open File	Read File
F-T-03	Read File's Info	
F-T-04	Rename File	Update File
F-T-05	Edit File	
F-T-06	Undo Editing	
F-T-07	Redo Editing	
F-T-08	Search File with REGEX	Search File
F-T-09	Search File with MATCH CASE	
F-T-10	Search File with REPLACE	
F-T-11	Set Line Numbers	Set Configuration
F-T-12	Set Syntax Highlight	
F-T-13	Set Wrap Content	
F-T-14	Set Use Monospace	
F-T-15	Set Read Only	
F-T-16	Set Font Size	
F-T-17	Set Theme	
F-T-18	Set Accessory View	
F-T-19	Set Use the storage Access Framework	

F-T-20	Set Keyboard Suggestion and Swipe	
F-T-21	Set Auto Save	
F-T-22	Set Encoding	
F-T-23	Set Ignore Back Button	
F-T-24	Set Split The Text If Too Long	
F-T-25	Set Fullscreen Mode	
F-T-26	Set Open Last Viewed File At Startup	

2.4 Features Not To Be Tested

Feature ID	Description	Use Case
F-NT-01	Create New Folder	Create File
F-NT-02	Open Folder	Read File
F-NT-03	Share File	
F-NT-04	Rename Folder	Update File
F-NT-05	Edit Folder	

2.5 Item Pass/Fail Criteria

Test Item 的 Pass/Fail 判斷標準，分為下列兩種方式：

- 一、以 User Scenario Technique 設計之測試案例：以一功能之敘述句，在測試執行後是否成立為 Pass/Fail 標準，假使功能測試執行後，使敘述句成立則 Pass；反之則 Fail，而本次測試之目標為使所有與 Use Case 有關以及與 2.3 Features To Be Tested 有關之 Test Case 通過測試(Pass)，證明軟體有達成使用案例的需求。
- 二、以 Control-Flow Graph Coverage Technique 設計之測試案例：以 Basis Path Approach，須覆蓋所有分支，意即 Statement Coverage 及 Branch Coverage 是 100%，證明測試案例是有實際將該段程式碼執行過的。

不論判斷標準為何，最終希望本次測試能夠達成 80% 的 Code Coverage，證明因為測試涵蓋這麼高的覆蓋率，所以測試的結果是可以信任的。

3.Environment Needs

3.1 Operational Environment

本測試考量測試中觀測及紀錄(文字、圖像及影像)的方便性，將計畫於個人電腦(PC)上之 Android 模擬器，透過框架進行 APP UI 自動化測試，其硬體規格如 3.2；軟體規格如 3.3。

3.2 Hardware Specification and Configuration

種類	描述	備註
中央處理器	Intel® Core™ i5-4590	
圖形處理器	Intel® HD Graphics 4600	位於中央處理器內
硬碟	INTEL SSDSC2CT240A4	Intel® SSD 335 Series
記憶體	DDR3-1333	16 GB
顯示器	VW247N、VN289N	

3.3 Software Specification and Configuration

名稱	版本	備註
Microsoft Windows 10 專業版	1903	
Android Studio	3.6.2	SDK Platform Required
Espresso	3.1.0	
UI Automator	2.2.0	

4. Personnel and Responsibility

4.1 Role and Responsibilities

姓名	主要職稱/角色	主要職務/角色描述
古兆瑋	測試設計人員	進行測試設計，協助測試實作
陳冠穎	測試設計人員	調控專案時程，協助測試設計及實作
劉孝忠	測試實作人員	進行測試實作及環境建置

備註：角色因故無法執行職務時，得視情況調度其他人員取代或代理其職務。

4.2 Staffing and Training

訓練項目	Turbo Editor 操作流程	Appium 測試方法
訓練時間	(預計於 Phase 1 結束以前)	
訓練地點	宏裕科技大樓 1623 室	
講師	劉孝忠	
學員	古兆瑋、陳冠穎	

備註：訓練項目得同時或批次進行。

5. Test Schedule

5.1 Schedule and Milestones

日期/期間	行程	備註
2020/04/08	訓練 Turbo Editor 操作流程及 Appium 測試方法	無
2020/04/09 ～ 2020/04/21	撰寫 Test Design Specifications (TDS) 撰寫 Test Case Specifications (TCS) 籌備 Midterm Presentation	無
2020/04/22	里程碑：Midterm Presentation (Phase 2 due)	無
2020/04/23 ～ 2020/05/26	達成 Phase 3 需求	無
2020/05/27	里程碑：Phase 3 due	無
2020/05/28 ～ 2020/06/22	達成 Phase 4 需求	無
2020/06/23	里程碑：Phase 4 due	無
2020/06/24 ～ 2020/06/29	籌備 Final Presentation 達成 Final Submission 需求	無
2020/06/30	里程碑：Final Presentation (Final Report due)	無

備註：由於專案成員之身分為學生，針對課業之作業不定性，無法提供明確的時間點，故以各 Due Date 及期間為主要目標及里程碑劃分。

Glossary

SUT	軟體系統測試則通常是以被測系統（System Under Test，SUT）表示正在被測試的系統，目的是測試系統是否能正確操作。這一詞語常用於軟體測試中。[1]
Android	Android（讀音：英：['ændrɔɪd]，美：['æŋ,drɔɪd]），中文常譯作安卓或安致，是一個基於 Linux 核心的開放原始碼行動作業系統，由 Google（Google）成立的開放手機聯盟持續領導與開發，主要設計用於觸控螢幕行動裝置如智慧型手機和平板電腦與其他可攜式裝置。[2]
APP	A short of “Application”
Turbo Editor	Simple, Powerful and Open Source Text Editor for Android. Built following the latest Android Guidelines to make the experience the most intuitive on the Store, also on Tablets! [3][4]
Espresso	<p>The core API is small, predictable, and easy to learn and yet remains open for customization. Espresso tests state expectations, interactions, and assertions clearly without the distraction of boilerplate content, custom infrastructure, or messy implementation details getting in the way.</p> <p>Espresso tests run optimally fast! It lets you leave your waits, syncs, sleeps, and polls behind while it manipulates and asserts on the application UI when it is at rest. [5]</p>
UI Automator	The UI Automator testing framework provides a set of APIs to build UI tests that perform interactions on user apps and system apps. The UI Automator APIs allows you to perform operations such as opening the Settings menu or the app launcher in a test device. The UI Automator testing framework is well-suited for writing black box-style automated tests, where the test code does not rely on internal implementation details of the target app. [6]

Reference

[1] 被測系統(Wikipedia)

<https://zh.wikipedia.org/wiki/%E8%A2%AB%E6%B5%8B%E7%B3%BB%E7%BB%9F>

[2] Android(Wikipedia) <https://zh.wikipedia.org/wiki/Android>

[3] Turbo Editor(Google Play)

<https://play.google.com/store/apps/details?id=com.maskyn.fileeditor>

[4] Turbo Editor(Github) <https://github.com/vmihalachi/turbo-editor>

[5] Espresso <https://developer.android.com/training/testing/espresso>

[6] UI Automator <https://developer.android.com/training/testing/ui-automator>

« »

