

Tasks

Software Test Plan v1.0

組別：

成員： 鄭鴻仁
洪子軒
陳郁欣
陳巧宜
林冠璋

Content

Change History	3
1. Test plan identifier:	4
2. Introduction and References	4
3. Test Items	5
4. Features To Be Tested	5
5. Features Not To Be Tested	5
6. Test Approach	6
7. Item Pass/Fail Criteria:	6
8. Test Deliverables Identify	7
9. Testing Tasks	8
10. Environment Needs	8
11. Role and Responsibilities	9
12. Staffing and Training	10
13. Schedule and Milestones	10
14. Risks and contingencies	11
15. Approvals	12

Change History

Date	Item
2019/03/21	Prototype
2019/03/25	Add new testing task items
	Modify few feature

1. Test plan identifier:

Tasks-STP-01

Reference: app name: Tasks (Version 1.1.16)

- Project plan: 一個簡單容易操作的待辦事項管理應用程式，讓使用者能夠根據待辦事項的緊急程度，執行狀態以及特定範圍時間等條件來進行分類。
- System requirement: 此 app 版本支援 Android 4.0.3 以上的作業系統。
- Reference materials: Test design specification (TSD), Test Case Specification (TCS)

2. Introduction and References

關於我們的測試規劃，我們致力於達成覆蓋率 90%，目的是為了要檢測出 App 是否有存在其缺陷，以及找出 App 潛在的 Bug。

此次測試 App 的名稱叫做 Tasks，是一個可以建立代辦事項的 App，當中可以設定提醒時間、設定優先度、新增 check List 等功能。而我們將對以下兩個大 feature 去做測試：

1. Code Coverage：

- task 的 CRUD
- list 的 CUD
- task 的編輯
- task 的優先度排序
- 搜尋 task

2. Abnormal detection：

- 快速點擊
- 反轉
- Pause
- 翻轉
- 亂數輸入
- 測試輸入上下限值 or 輸入空值
- 手機環境改變
- 權限改變

3. Test Items

此專案生產的所有資料將被儲存在 Github 上進行版本控管，全部測試的項目都可在上面取得。以下為所有的測試項目：

- **Executable Code**

Tasks 可執行的程式碼將被測試

4. Features To Be Tested

1. Code Coverage：

- 新增 task：包含快速新增與一般新增
- 編輯 task：所有欄位更新包含：List、Title、Status、Location、Description、Check List、Start time、Due time、All Day、Priority、Privacy、URL
- 刪除 task
- 新增 task list
- 編輯 task list
- 刪除 task list
- Task 分類：包含 Due、Start、Priority、Task Progress
- 搜尋

2. Abnormal detection：

- 快速點擊
- 反轉
- Pause
- 翻轉
- 亂數輸入
- 輸入上下限值 or 輸入空值
- 手機環境改變
- 權限改變

5. Features Not To Be Tested

1. SMS
2. 同步
3. 分享

6. Test Approach

- Executable Code Testing

將透過人工測試檢查功能是否符合 Tasks 官方提供的說明。

- Automated testing tools

我們將使用 Appium 及 JUnit 等工具，進行 Tasks 的自動化測試。

- Defect Tracking

我們使用 Log 記錄腳本執行過程，裝置資訊，時間資訊，並且依紀錄出 Crash 或是 Abnormal 的事件。

- Change Manage

使用 Github 來進行版本控管

- Continuous integration(CI)

我們架設 Jenkin，以達到持續整合，並且定期建置專案，定期追蹤覆蓋率是否有無上升，進而調整測試計畫。

7. Item Pass/Fail Criteria:

1. Code Coverage:

- Suspension criteria:

當達到 40% 的 test cases 不通過時，所有測試人員暫停測試，並且集中處理所有沒通過的 test cases。

- Resumption criteria:

在解決 80% 不通過的 test cases 後，所有測試人員繼續進行接下來的測試。

- Approval criteria:

所有 test cases 必須通過，並且達到程式覆蓋率(Code coverage) 80%。

2. Abnormal Criteria:

- Suspension criteria:

當偵測到異常狀態，將 bug 送進 Bug tracking 系統，並且定期回報給開發團隊。

- Approval criteria:

所有 test cases 必須通過，並且偵測不到異常狀態。

8. Test Deliverables Identify

- **Phase1. Before testing**

Test Plans Document:

在測試的時候我們需要良好定義的測試規劃文件來確認我們目前的進度，測試規劃更像是一種規劃測試活動如何進行的藍圖。

Test Cases Document:

測試案例是好幾組條件跟變數應該在什麼樣的條件，測試者來確認系統行為是否有誤，因此我們透過 Test Case Document 來紀錄我們規劃的行為。

Test Design Specifications:

測試測試文件是一種在測試時會使用到的一種文件，它包含需要有效率去執行判斷程序的資訊和參數。

Test Data:

會透過收集 client-app 的元件 XPath，來當作我們的 Test Data

Test Tool:

我們會使用我們自己所開發的工具: Appium Keyword Base Tool，簡稱 AKB，Github: <https://github.com/likesm0887/STV-Project>。測試人員依照測試資料所撰寫的腳本，透過 AKB，就可以運行腳本，並且進行測試，並且產生覆蓋率，不須複雜的語言，簡單方便。

- **Phase2. During testing**

Test Scripts:

測試人員依照 Test Data 撰寫 High-Level Keyword 測試腳本，透過 AKB 執行腳本，並且進行驗收。

- **Phase3. After testing**

Test Results/reports :

用來觀看測試成功的程度，statement coverage、branch coverage

Change History:

Github: 追蹤程式碼版本。

Trello: 追蹤進度流程。

Log:

紀錄裝置資訊，異常狀態，事件執行順序，時間……等資訊。

9. Testing Tasks

請見附件 A

10. Environment Needs

1. Hardware (硬體) :

測試將被在下列硬體配置上執行:

- CPU: Intel P4 1.0 GHz.
- RAM: 256 MB.
- Hard Driver: At least 2 GB free spaces available.
- Monitor: 15' LCD, resolution: 1024 x 768, 16-bits color.
- Network Card: 100 Mbps Ethernet card.

2. Software (軟體) :

測試需要使用下列軟體:

- OS: Mac or Window 10.
- Compiler: JDK 1.8
- Project Build: Maven 3.9.
- IDE: IntelliJ 2018.
- Concurrent Versions System: Git and Github
- Bug tracking: Backlog.
- Unit testing: JUnit.
- Build Tool: Appium 1.11.0.

- Environment: Node.js 1.10 , ADB 1.0.40
- Documentation: Microsoft Word 2016 or Higher

11. Role and Responsibilities

Developer 陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒

責任:設計測試案例，完成測試需求。

Role	Candidate	Description
Test Analysts	鄭鴻仁	進行評估測試案例的可行性，並且是否有按照需求
Test Engineers	陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒	將測試案例撰寫成腳本
Tool Specialist	陳郁欣	架設 appium 環境 與 CI

12. Staffing and Training

下列人員需要執行這項專案的測試。每位人員也需要被指定各種訓練。

Test group (測試團隊)

Role	Number	Training
Test Analysts	1	<ul style="list-style-type: none">• 如何使用 appium,Uiautomator
Test Engineer	2	<ul style="list-style-type: none">• 如何使用 appium,Uiautomator• 學習 test case 撰寫• 學習 java Android app 結構
On Site Customer	1	<ul style="list-style-type: none">• 接受測試概念

13. Schedule and Milestones

請見附件 A

14. Risks and contingencies

Risk	Cause	Mitigation	Response	Severity	Probability
待測系統過於複雜	無法按照文件製作，無法使用工具測試	易讀的文件，好用的工具	訓練會議，"輔導"	High	Low
時間不足以完成所有測試	太多測試項目	設計少量的測試案例來包含大量的待測情況	提供更多的資源進行測試，smoke testing	Medium	High
測試系統失效	選擇經常失效或無法被使用的工具	選擇經常使用，且易於使用的工具	改變工具 (繁重的過程)	High	Low
太多錯誤	測試系統找到太多錯誤而不夠及時修正	及早進行測試，編寫好的程式碼	提供更多的資源，a)放棄失敗的產品，b)改變指派的工作份量	High	Low
測試經理辭職	缺乏動力，測試單倚賴一個人	激勵試驗經理，提供額外的報酬	派其他人到工作，在理解以前的設計之前，不要計畫新設計	High	Low
當測試時，缺乏人力	撰寫測試時因能力不足或時間不構	由資深同事幫忙撰寫	分發將兼差的人加入測試單位	Medium	Low
外部的培訓課程無法提供	沒有資金學習課外的測試技術	在網路上免費課程學習，並且帶回團隊交流	成立技術研討會	Medium	Medium
測試的環境失效	模擬器 crash，或是測試工具失效	重新架設環境	重建測試環境	Medium	Medium
程式碼的品質低落。表示在程式碼裡有太多錯誤	將問題反映到 bug 追蹤系統	呈報給開發團隊	將保持一個會議評論原始碼解決這個問題	High	Low

15. Approvals

Team Members	Date	Hours
陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒	2018/2/28	2.5 hr
陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒	2018/3/21	6 hr
陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒	2018/3/25	18 hr

16. 附件 A

Task	Predecessor Tasks	Special Skill	Responsibility	Effort	Finish
1.準備測試計畫	None		All Teams		2019/2/28
2.準備測試設計 規格	Task1	None	All Teams		2019/2/28
3.準備測試案例 規格	Task2	None	All Teams		2019/3/11
4.準備測試流程 規格	Task3	Maven Dependency Rule	All Teams		2019/3/11
5.建立測試環境	Task4	了解 Junit、Maven、 IntellJ、Android Simulator、Appium	All Teams		2019/3/18
6.驗證測試環境	Task5	了解 Junit、Maven、 IntellJ、Android Simulator、Appium	All Teams		2019/3/25
7.撰寫測試項目 並且上傳版本控 制	通過單元測 試	了解測試案例需求	All Teams		2019/5/21
8.檢查測試流程	Task7	None	All Teams		2019/5/21
9.執行批次測試 流程	Task8	None			2019/5/21
10.檢查批次測試 流程結果	Task9	None	All Teams		2019/5/21
11.產生測試報表	Task10	None	All Teams		2019/5/21
12.分析測試報表	Task11	None	All Teams		2019/5/21
13.重複步驟 7~12 直到所有測試案 例通過		None	All Teams		2019/5/21
14.撰寫系統概要 報表	Task13	None	All Teams		2019/6/11
15.封裝測試文件 及資料	Task15	None	All Teams		2019/6/11