**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

[**1.** **Test plan identifier:** 3](#_Toc4167667)

[**2.** **Introduction and References** 3](#_Toc4167668)

[**3.** **Test Items** 3](#_Toc4167669)

[**3.1** **Executable Code** 3](#_Toc4167670)

[**4.** **Features To Be Tested** 4](#_Toc4167671)

[**5.** **Features Not To Be Tested** 4](#_Toc4167672)

[**6.** **Test Approach** 4](#_Toc4167673)

[**7.** **Item Pass/Fail Criteria:** 4](#_Toc4167674)

[**8.** **Test Deliverables Identify** 5](#_Toc4167675)

[**9.** **Testing Tasks** 6](#_Toc4167676)

[**10.** **Environment Needs** 6](#_Toc4167677)

[**11.** **Role and Responsibilities** 6](#_Toc4167678)

[**12.** **Staffing and Training** 7](#_Toc4167679)

[**13.** **Schedule and Milestones** 7](#_Toc4167680)

[**14.** **Risks and contingencies** 8](#_Toc4167681)

[**15.** **Approvals** 9](#_Toc4167682)

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)

1. **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。而我們將對以下提到幾個 feature 去做測試 :

1. task 的 CRUD
2. list 的 CUD
3. task 的編輯
4. task 的優先度排序
5. 搜尋 task
6. **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：

* 1. **Executable Code**

Tasks可執行的程式碼將被測試

1. **Features To Be Tested**

* 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

1. **Features Not To Be Tested**
2. SMS
3. 同步
4. 分享
5. **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等工具幫助我們進行Tasks的自動化測試，並且架設Jenkin達到持續整合。

1. **Item Pass/Fail Criteria:**
2. 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。
3. 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。
4. Approval criteria: 所有test cases 必須通過，並且達到程式覆蓋率(Code coverage) 80%。
5. **Test Deliverables Identify**

* **Phase1. Before testing**

Test Plans Document:

在測試的時候我們需要有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，測試者鞥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。

* **Phase2. During testing**

Test Scripts:

撰寫High-Level Keyword測試腳本，透過本專案提供的測試轉換工具“…..”執行腳本，並且進行驗收。撰寫Appium測試腳本，並且使用Junit自動執行測試腳本，進行驗收。

Test Data:

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

* **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。

1. **Testing Tasks**
2. 安裝Appium
   1. 設定JAVA\_Home
   2. 設定Andriod\_Home
3. 安裝 Andorid Studio
   1. 安裝模擬器
   2. 安裝SDK
4. 熟悉Appium API
5. 安裝intellij
   1. 安裝Junit
6. **Environment Needs**

使用Andriod GUI自動化測試工具”Appium”，加上Java client side API，IDE使用Intellij，並且使用模擬機當作測試Device，SDK為23以上，JDK為1.8。

1. **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 |

1. **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 | * 如何使用appium,Uiautomator |
| Test Engineer | 2 | * 如何使用appium,Uiautomator * 學習test case 撰寫 * 學習java   Android app 結構 |
| On Site Customer | 1 | * 接受測試概念 |

1. **Schedule and Milestones**
2. **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 |

1. **Approvals**

|  |  |  |
| --- | --- | --- |
| Team Members | Date | Hours |
| 陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒 | 2018/5/1 | 2.5 hr |
| 陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒 | 2018/5/5 | 6 hr |
| 陳郁欣, 鄭鴻仁,林冠璋,陳巧宜,洪子軒 | 2018/6/15 - 22 | 65 hr |