Software Testing and Verification

Tasks

Test Design Specification

資工碩一 107598010 陳郁欣

資工碩一 107598013 鄭鴻仁

資工碩一 107598020 林冠璋

資工碩一 107598025 陳巧宜

資工碩一 107598042 洪子軒

|  |  |
| --- | --- |
| **Version：2.00** | **Date: 2019/4/14** |

**Version history**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | Author | Reviewer | Status | Description |
| 1.00 | 19.3.26 | Whole Team | Whole Team | Draft | Initial draft |
| 2.00 | 19.4.14 | Whole Team | Whole Team | final | Fix use cases |

**Table of contents**

* [**Test design specification identifier**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#identifier)
* [**Features to be tested**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#tested)
* [**Approach refinements**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#approach)
* [**Testing identification**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#identification)
* [**Feature pass/fail criteria**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#pass_fail)
* [**References**](http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/MUPePe/t1/testing/test_design.html#references)

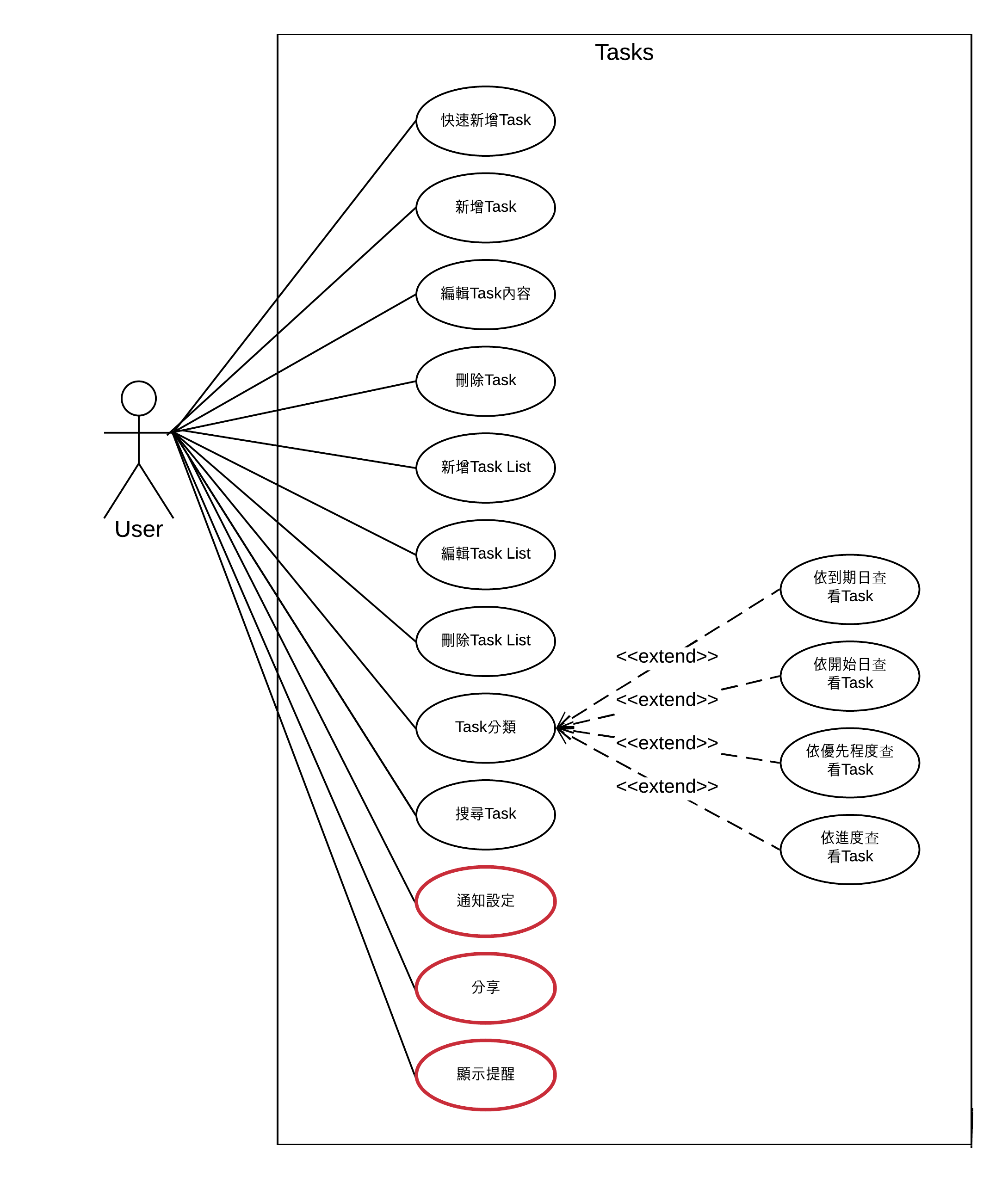
**Test design specification identifier**

Identifier for document is **TASKS-TDS-001 1.00**.

This is document is associated with test plan **TASKS-STP-001**.

**Features to be tested**

The following use case diagram from [2] shows the basic functionality of the program.



[2]

* **Each use case will be tested by at least one test case and all test cases should be passed.**

|  |  |
| --- | --- |
| Feature | Priority |
| Create List | High |
| Delete List | Low |
| Update List | Low |
| Create Task(Quick\_Add) | High |
| Create Task(Normal\_Add) | High |
| Delete Task | High |
| Update Task | High |
| Search Task | Medium |
| Classification | Medium |

The following combinations will be used:

|  |
| --- |
| {Create List、Create Task(Quick or Normal)}, {Create Task(Quick or Normal)、Delete Task}, {Create List、Delete List}, {Create Task(Quick or Normal)、Classisfication}, {Create Task(Quick or Normal)、Search Task}, {Create Task(Quick or Normal)、Update Task}, {Create List Delete List}, {Create List、Update List} |

**Approach refinements**

The four levels of the V-model will be tested separately.

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Responsibility** | **Refinement** | **Identification** |
| Acceptance testing | Client | Take the software to the client | Client sees, that the software is created according to demands. |
| System testing | Test group | Create new tests, run them | Manual tests cases produce visible result |
| Integration testing | Developer | Insert new module | System works visually and compiles, automated tests return clear result |
| Module testing | Developer | Create new module, test changed module | Module compiles and **junit** testing passes. |

Test cases for phase 1 will be created using common sense and can be analyzed by anyone with experience in computer programming and java.

Results of the automated testing tools, Appium, will not be analyzed beyond bug reports and phase 1.

**Testing identification**

|  |  |
| --- | --- |
| (TC01)Test Scenario : Quick add task | |
| User scenario | 1. 使用者點擊Task List。 2. 系統顯示+符號。 3. 點擊+符號，彈出快速新增視窗。 4. 使用者輸入Title並且按下“SAVE“按鈕。 5. 成功新增Task。 |
| Test Input | 使用者輸入Title名稱。 |
| Test Output | 顯示新增的Task，且Task名稱為使用者輸入內容。 |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | 新增一個Task。 |

|  |  |
| --- | --- |
| (TC02)Test Scenario : Continuously quick add task | |
| User scenario | 1. 使用者點擊Task List。 2. 系統顯示+符號。 3. 點擊+符號，彈出快速新增視窗。 4. 使用者輸入Title並且按下“SAVE AND CONTINUE“按鈕。 5. 系統顯示Task Created訊息。 6. 重複步驟4，直到不再新增時改按“Save“按鈕。 |
| Test Input | 輸入多個Task的Title。 |
| Test Output | 顯示新增的三個Task，名稱為依序輸入的內容。 |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | 新增一個Task。 |

|  |  |
| --- | --- |
| (TC03)Test Scenario : Create a task | |
| User scenario | 1. 使用者點擊右下角+號Button。 2. 系統彈出視窗。 3. 使用者輸入Task內容並且按下確認。 4. 系統成功新增Task。 |
| Test Input | 輸入Task內容。 |
| Test Output | Task Summary Page顯示輸入的內容。 |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | 新增一個Task。 |

|  |  |
| --- | --- |
| (TC04)Test Scenario : Update title of task | |
| User scenario | 1. 使用者點擊要更新的Task。 2. 系統顯示Task Summary頁面。 3. 使用者點選編輯按鈕。 4. 系統顯示Task編輯頁面。 5. 使用者輸入要更新的欄位。 6. 使用者點擊Save按鈕。 |
| Test Input | 使用者針對選擇的Task，要更新的內容。 |
| Test Output | 點選修改的Task，系統把此Task修改成使用者輸入的更新內容。 |
| Precondition | 1. 打開應用程式，Activity為MainActivity。 2. List裡面的Task不為空。 |
| Postcondition | 成功更新Task的Title。 |

|  |  |
| --- | --- |
| (TC05)Test Scenario : Delete task | |
| User scenario | 1. 使用者點選List，系統顯示此List裡的Task。 2. 使用者點選要刪除的Task，系統彈出視窗。 3. 使用者點選右上角第一個設定按鈕選擇刪除並且按下確定。 4. Task被完整刪除。 |
| Test Input | 新增一個Task 。 |
| Test Output | 系統顯示的Task裡，沒有被刪除的Task。 |
| Precondition | 1. 打開應用程式，Activity為MainActivity。 2. List裡面的Task不為空。 |
| Postcondition | 選擇的Task被完整刪除。 |

|  |  |
| --- | --- |
| (TC06)Test Scenario : add List | |
| User scenario | 1. 使用者點選右上角menu按鈕，系統顯示menu選項視窗。 2. 使用者點選Displayed Lists，系統顯示Displayed Lists頁面。 3. 使用者點選右上角+按鈕，系統顯示list name輸入視窗。 4. 使用者輸入List名稱，並且按下確認按鈕。 5. 系統顯示新增的list名稱及list顏色。 6. 使用者點選儲存按鈕。 7. 系統成功新增List。 |
| Test Input | 使用者輸入List名稱。 |
| Test Output | 系統新增新的List，名稱為使用者輸入的內容。 |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | List成功新增。 |

|  |  |
| --- | --- |
| (TC07)Test Scenario : Delete list | |
| User scenario | 1. 使用者點選右上角menu按鈕，系統顯示menu選項視窗。 2. 使用者點選Displayed Lists，系統顯示Displayed Lists頁面。 3. 使用者點選要刪除的List右邊設定按鈕，系統顯示list名稱及顏色的預覽視窗。 4. 使用者點選Delete List，系統顯示確認刪除對話框。 5. 使用者點選確認刪除按鈕。 6. 系統成功刪除List。 |
| Test Input | 新增一個List。 |
| Test Output | 系統顯示的List，沒有被刪除的List。 |
| Precondition | 1. 打開應用程式，Activity為MainActivity。 2. 應用程式List數量不為0。 |
| Postcondition | 系統List的數量比原本少1。 |

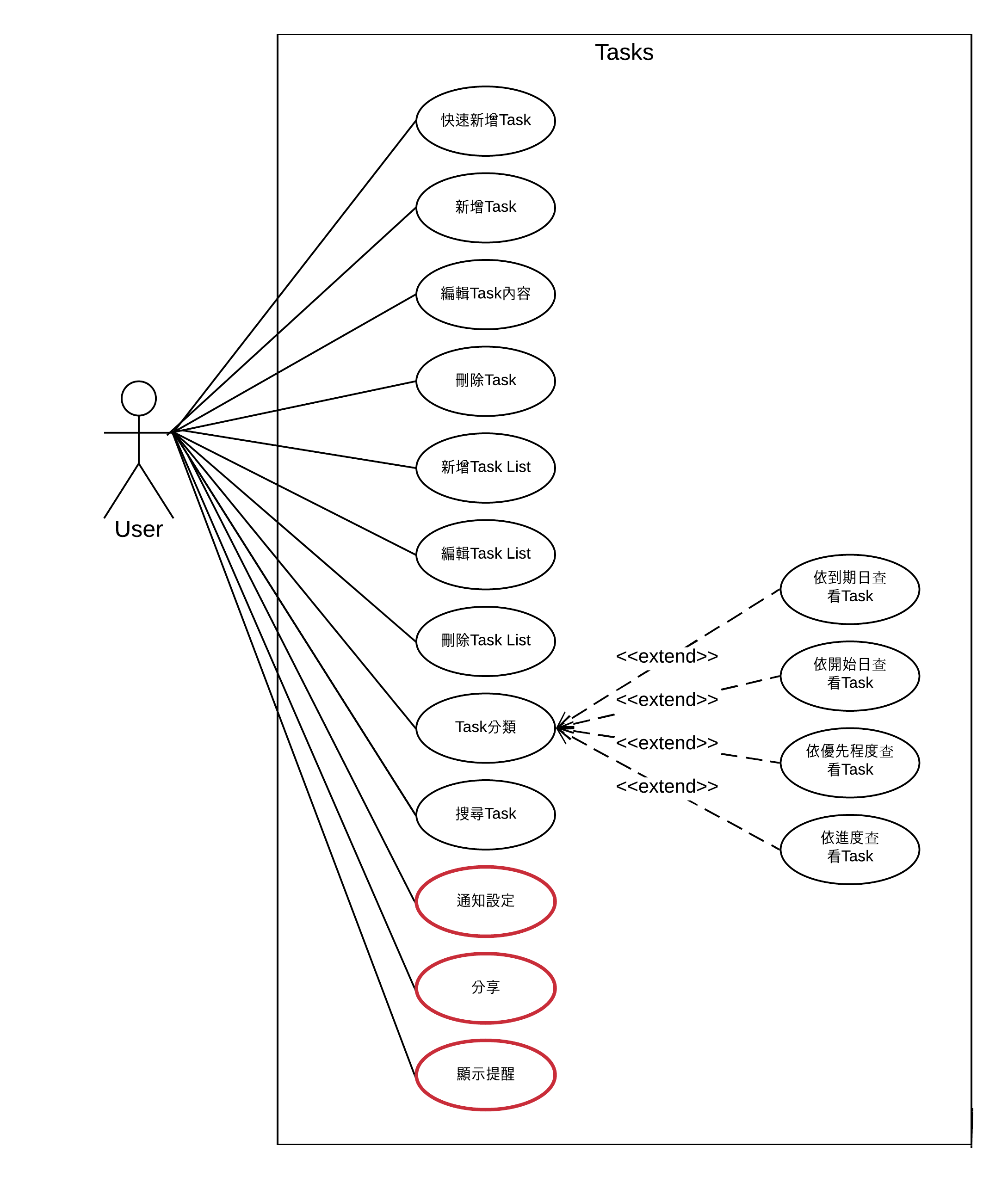
|  |  |
| --- | --- |
| (TC08)Test Scenario : Refresh list name | |
| User scenario | 1. 使用者點選右上角menu按鈕，系統顯示menu選項視窗。 2. 使用者點選Displayed Lists，系統顯示Displayed Lists頁面。 3. 使用者點選要更新的List右邊設定按鈕，系統顯示List名稱及顏色的預覽視窗。 4. 使用者點選Name，系統顯示List name修改視窗。 5. 使用者輸入List的新名稱，並且按下確定。 6. 系統顯示List名稱及顏色的預覽視窗。 7. 使用者點選List Color，系統顯示Color選者視窗。 8. 使用者點選視窗下方的顏色組合，系統顯示顏色組合的選項。 9. 使用者點選顏色，系統顯示List名稱及顏色的預覽視窗。 10. 使用者點選儲存按鈕。 11. 系統成功更新List名稱及顏色。 |
| Test Input | 使用者輸入List要更新的名稱。 |
| Test Output | 對於更新的List，系統顯示更新後的名稱。 |
| Precondition | 1. 打開應用程式，Activity為MainActivity。 2. 應用程式List數量不為0。 |
| Postcondition | List名稱確實被修改。 |

|  |  |
| --- | --- |
| (TC09)Test Scenario : Edit task | |
| User scenario | 1. 使用者點擊List，系統顯示此List所包含的Task。 2. 使用者點擊要修改的Task，系統彈出視窗並顯示此Task之內容。 3. 使用者修改Task內容並且確認。 4. 系統成功修改Task內容。 |
| Test Input | 使用者輸入要編輯的Task內容。 |
| Test Output | 對於更新過的Task，系統顯示修改後的內容。 |
| Precondition | 1. 打開應用程式，Activity為MainActivity。 2. 應用程式Task數量不為0。 |
| Postcondition | Task成功被修改。 |

|  |  |
| --- | --- |
| (TC10)Test Scenario : Search task | |
| User scenario | 1. 使用者點選TabList最右邊的搜尋按鈕，系統彈出搜尋視窗。 2. 使用者輸入要搜尋之Task名稱並且按下確認。 3. 系統回傳搜尋之Task。 |
| Test Input | 輸入要尋找的Task名稱。 |
| Test Output | 系統顯示尋找到的Task。  系統顯示My Task、系統無顯示任何Task |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | 系統顯示搜尋之Task。 |

|  |  |
| --- | --- |
| (TC11)Test Scenario : Classification | |
| User scenario | 1. 使用者點擊右下角新增按鈕。 2. 輸入Task分類屬性跟內容，並且按下儲存。 3. 使用者點選不同的分類，系統根據不同的分類顯示相對應的Task。 |
| Test Input | 新增Task設定Start、Due、Priority、Progress屬性(不用每個屬性都要有值)。 |
| Test Output | 在相對應的分類可以看到依屬性新增的Task。 |
| Precondition | 打開應用程式，Activity為MainActivity。 |
| Postcondition | 系統顯示依據的分類。 |

**Feature pass/fail criteria**  
  Each use case will be tested by at least one test case and all test cases should be passed.



**References**

[1]  IEEE Standard for Software Test Documentation, IEEE Std 829, 09/1998