

Kanban

Software Design Document (SDD)

Version: 1.0

Team 6

Name	ID	E-mail
劉柏宏	103590001	lucy933626@gmail.com
鄭鴻仁	107598013	hong012078@gmail.com
陳巧宜	107598025	jammy1129@gmail.com
洪子軒	107598042	gay88358@yahoo.com.tw

**Department of Computer Science & Information Engineering
National Taipei University of Technology**

11/28/2018

Table of Contents

Table of Contents	1
Section 1 System Models/System Architecture	3
1.1 Decomposition of the Functional Requirement to Subsystem Requirement and Interfaces	3
1.1.1 Functional Requirement	3
1.1.2 System Interfaces	4
1.2 Establish Technical Solution Criteria	4
1.3 Describe Alternative Solution.....	5
1.4 Select System Solution	6
1.5 System Internal/External Interface	7
Section 2 Design Issues and Solutions	8
2.1 Board Subsystem	8
2.1.1 Subsystem Characteristics	8
2.1.2 Establish Technical Solution Criteria.....	8
2.1.3 Selected Subsystem Solution.....	8
2.1.4 Error Detection and Recovery	9
2.2 User and Team Management Subsystem.....	9
2.2.1 Subsystem Characteristics	9
2.2.2 Establish Technical Solution Criteria.....	9
2.2.3 Selected Subsystem Solution.....	10
2.2.4 Error Detection and Recovery	10
2.3 Report Generating Subsystem.....	11
2.3.1 Subsystem Characteristics	11
2.3.2 Establish Technical Solution Criteria.....	11
2.3.3 Selected Subsystem Solution.....	11
2.3.4 Error Detection and Recovery	11
Section 3 Detailed of Subsystem and Interface Description	12
3.1 Detailed System	12
3.1.1 Use Cases Analysis	12
3.1.2 User Interfaces Analysis.....	18
3.1.3 Static Model	26
3.1.4 Dynamic Model	29
3.2 Subsystem Internal/External Interface Requirement and Design	33
3.2.1 Internal Interface Requirements	33
3.2.2 External Interface Requirements	33
3.3 Traceability Matrix – Requirements vs Components.....	34

Glossary	35
References.....	36
Appendices.....	37
A. Tracebility Matrix Use Cases v.s. Classes	37
B. Tracebility Matrix Classes v.s. Classes	37

Section 1 System Models/System Architecture

1.1 Decomposition of the Functional Requirement to Subsystem Requirement and Interfaces

1.1.1 Functional Requirement

看板系統（Kanban）是為了讓團隊能視覺化正在運行的流程，以利團隊觀察各個階段的狀態，並進一步改善流程，是一種流程改善的工具。本系統包含三個子系統，分別為：

1. 使用者和團隊管理子系統：負責管理帳號、權限和團隊相關資料
2. 看板子系統：負責看板內容的操作及呈現
3. 報表製作子系統：紀錄看板內個工作項目的變化，並繪製成 CFD。

KBS 的 Function Requirement 如下：

需求編號	優先順序	需求描述
KBS-F-01	1	提供看板，且能夠管理 task。
KBS-F-02	1	提供團員管理。
KBS-F-03	1	提供多個看板功能。
KBS-F-04	1	提供報表產生管理功能。
BS-F-01	1	使用者必須先登入才可修改資料。
BS-F-02	1	提供專案管理功能，包括專案的新增、修改、刪除和查詢。
BS-F-03	1	提供專案工作管理，包括工作的新增、修改、刪除和查詢。
BS-F-04	1	設定專案工作的相關性、開始結束時間。
BS-F-05	1	提供追縱專案工作狀態，包括確認專案目前完成狀態、工作完成狀態。
BS-F-06	1	記錄每個對工作的操作紀錄，可供使用者觀看變動紀錄。
UTMS-F-01	1	提供使用者登入，進行身份辨識。
UTMS-F-02	1	使用者可以設定/修改自己的個人資訊。
UTMS-F-03	1	設定帳戶權限。
UTMS-F-04	1	安全認證機制（密碼加密）。
RGS-F-01	1	使用者必須有專案權限才可瀏覽該專案資料。
RGS-F-02	1	產生專案 CFD。

1.1.2 System Interfaces

圖 2-1 為 KBS 的系統架構圖，說明了 KBS 與其包含的數個子系統之間的互動關係。

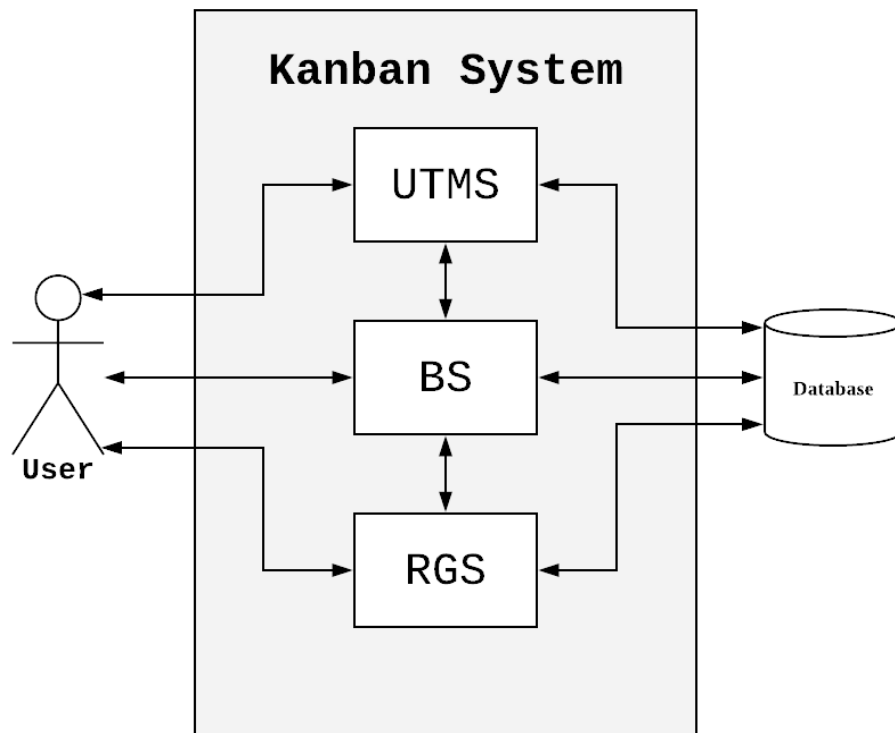


圖 2-1 Kanban 系統架構圖

本系統主要分為三個部分，分別為使用者和團隊管理子系統（UTMS, User Account and Team Management Subsystem）、看板子系統（Board Subsystem）、報表製作子系統（Report Generating Subsystem）。

1.2 Establish Technical Solution Criteria

有關 Solution Criteria 分面，專案針對 KBS 擬定了將來會遇到的各種限制，包括：

- 易學性：考慮選擇的應用軟體是否容易上手。
- 可攜性：考慮是否能在各種作業系統上運行。
- 安全性：考慮應用軟體設計上的安全保密性。
- 擴充性：考慮後續的擴充是否容易。
- 維護性：考慮後續的維護是否容易。

上述的限制問題，基本上與系統所用的軟體是否有關。

1.3 Describe Alternative Solution

在本系統設計前，要考慮各種可能影響系統架構的因素，包括系統架構、程式設計架構、資料庫軟體系統、與程式語言等的選擇。

(一) 由本專案的系統架構，開發者提供了可行的系統架構，分別為單機系統、主從系統、與網頁系統 架構，以下列出與上述限制比較表：

	單機系統	主從系統	網頁系統
易學性	高，開發時間較短	低，開發需要較長時間	低，開發需要較長時間
可攜性	低，必需因應每個平台而修改	中，必需額外開放使用	高，使用 Browser 連上即可
安全性	高，單機資料保護較容易	中，有被盜取資料的風險	中，有被盜取資料的風險
擴充性	彈性小，不能完全符合使用者需求	彈性大，可完全符合使用者需求	彈性大，可完全符合使用者需求
維護性	高，由開發者自行維護，且發展也極為不易。安裝較為簡單	高，由開發者自行維護。安裝較為複雜	低，由供應商提供維護作業。安裝容易

(二) 可用來開發系統的程式語言約有 C#、Java、C++：

	C#	Java	JavaScript
易學性	高，現代物件導向語言，易學	高，現代物件導向語言，易學	高，無型別語言，語法簡單，易學
可攜性	低，僅支援 Windows 作業系統	高，JVM 可支援各個平台	高，Nodejs 可支援各個平台
安全性	中，物件導向語言可實現資料封裝	中，物件導向語言可實現資料封裝	中，物件導向語言可實現資料封裝
擴充性	彈性中，部分符合開發者需求	彈性大，可完全符合開發者需求	高，有廣大社群提供各種不同的 API
維護性	高，物件結構較易維護	高，物件結構最易維護	中，無型別語言，不容易確認資料型態

(三) 系統的 User Interface 所使用 AJAX 框架目前有三種選擇，分別為 Prototype、JQuery、Vue.js，以下列出與上述限制比較表：

	Prototype	JQuery	Vuejs
易學性	中，使用率最高，網路範例較多	高，每個 API 都有完整的例子	整的例子低，大量的 UI 框架，較為複雜
可攜性	無，無可攜性需求	無，無可攜性需求	無，無可攜性需求
安全性	無，無安全性考量	無，無安全性考量	無，無安全性考量
擴充性	低，無相關 Plug-in	高，有大量用戶開發的 Plug-in 可供使用	高，模塊化實現，可擴展性強
維護性	高，每個 Action 都由單一檔案發佈	高，簡潔架構和簡短的語法，容易維護	高，元件物件化較可重複使用，易維護

1.4 Select System Solution

(一) 系統架構選擇

	單機系統	主從系統	網頁系統
易學性	4	2	3
可攜性	2	3	4
安全性	4	2	3
擴充性	2	4	4
維護性	4	4	3
加總	16	15	17

(二) 可用來開發系統的程式語言約有 C#、Java、C++：

	C#	Java	JavaScript
易學性	4	4	4
可攜性	3	4	4
安全性	3	3	3
擴充性	3	4	5
維護性	4	4	3
加總	17	19	19

(三) 系統的 User Interface 所使用 AJAX 框架目前有三種選擇，分別為 Prototype、JQuery、Vue.js，以下列出與上述限制比較表：

	Prototype	JQuery	Vuejs
易學性	4	4	4
可攜性	0	0	0
安全性	0	0	0
擴充性	2	4	4
維護性	4	4	5
加總	10	12	13

Priorities {Scale: 1 = Worst(difficult), 5 = Best(Easy), 0 = none necessary

Vuejs 前端框架能夠封裝物件，進而重複利用，相較於傳統網頁，提升開發速度與可維護性，因此選擇 Vuejs 作為前端主要框架，也搭配 Bootstrap 使用。而後端使用 Nodejs 及 mongoDB，開發速度較快，且可以再任意平台使用。

1.5 System Internal/External Interface

需求編號	優先順序	需求描述
KBS-N-11	1	UTMS 與資料庫間必預能傳送與接收專案資料。
KBS-N-12	1	BS 與資料庫間必預能傳送與接收專案資料。
KBS-N-13	1	RGS 必預能取得資料庫的專案資料但不能更改。
KBS-N-14	1	UTMS 允許使用者登入，更改帳戶資料。
KBS-N-15	1	BS 允許組員新增或瀏覽既有的專案資訊。
KBS-N-16	1	BS 允許組員可即時操作看板。
KBS-N-17	1	RGS 需產生專案報表可供使用者查詢。
KBS-N-18	1	透過網路傳送資料。
KBS-N-19	1	BS 能夠向 UTMS 取得使用者登入 Session 資訊。
KBS-N-20	1	BS 能夠向 UTMS 取得帳號權限資訊。
KBS-N-21	1	BS 能夠向 RGS 取得 CFD 報表資料。
KBS-N-22	1	RGS 能夠向 UTMS 取得使用者登入 Session 資訊。
KBS-N-23	1	RGS 能夠向 UTMS 取得帳號權限資訊。

Section 2 Design Issues and Solutions

2.1 Board Subsystem

2.1.1 Subsystem Characteristics

- i. 看板清單管理 (Board List Management) 之功能為建立、修改、刪除專案。
 - ii. 看板內容管理 (Board Item Management) 之功能為建立、修改、刪除看板內容。
 - iii. 工作管理 (Task Management) 之功能為建立、修改、刪除看板內容之工作項目。
- 資料庫 (Database) 可儲存專案資料。

2.1.2 Establish Technical Solution Criteria

資料庫系統儲存的比較表：目前資料庫有 MongoDB、MySQL 兩個進行比較。

MongoDB		MySQL
易學性	高，原來的使用者具有一定的基礎	高，原來的使用者具有一定的基礎
可攜性	高，免費資料庫，容易安裝	高，企業用免費系統容易安裝。
安全性	中，安全性目前無重大漏失	中，MySQL 安全性目前無重大漏失
擴充性	高，可以自動切割資料實現快速擴充	中，儲存量適於中小型的資料量儲存
維護性	高，不需要自行維護	低，需要自行維護

2.1.3 Selected Subsystem Solution

MongoDB		MySQL
易學性	4	4
可攜性	4	3
安全性	3	3
擴充性	5	3
維護性	4	4
加總	20	17

Priorities {Scale: 1 = Most bad (difficult), 5 = Most good (easy)} 根據上表評估，決定採用 MySQL 作為 MongoDB 的儲存資料庫。

2.1.4 Error Detection and Recovery

Error	直接實作登入驗證機制
使用者為登入	跳出警告視窗
使用者輸入資料錯誤	於使用者輸入資料錯誤的項目顯示訊息以供更正
資料不一致(資料庫端)	本系統的主程式將於使用者新增資料的時候自動檢查帳號資料與群組資料的一致性當系統偵測出不一致性時，會顯示相關訊息，且使用者無法進入登入畫面
資料不一致(使用者輸入端)	於前端的資料送至伺服器端的應用程式時，再進行檢查。當子系統一偵測出不一致性時，相關的訊息將由本子系統由前端通知使用者。

2.2 User and Team Management Subsystem

2.2.1 Subsystem Characteristics

- 帳號管理 (Account Management)之功能為建立、修改、刪除所有使用者的登入帳戶。
- 團隊管理 (Team Management) 可建立、修改、刪除專案。
- 登入管理 (Login Management) 的主要功能為驗證系統使用者是否以註冊。

2.2.2 Establish Technical Solution Criteria

	RBAC	直接實作登入驗證機制
易學性	低，RBAC 使用較為複雜	高，直接實作角色權限不需 另外學習
可攜性	中，必須處理安置 Session Key 的儲存位置	高，沒有需要額外負擔的處理
安全性	高，RDBC 控制權是直接賦予 角色較安全	低，直接實作控制權限，漏洞很多
擴充性	中，必須依靠 RBAC 內建功能 來擴充	高，不足部份直接實作即可
維護性	高，RDBC 的角色權限較容易維護	中，定義一個新的角色必須 撰寫新的權限功能

2.2.3 Selected Subsystem Solution

	RBAC	直接實作登入驗證機制
易學性	2	5
可攜性	3	5
安全性	5	2
擴充性	3	3
維護性	4	3
加總	17	18

Priorities {Scale: 1= Most bad(differcult), 5= Most good(easy), 0= Not necessary}

2.2.4 Error Detection and Recovery

Error	直接實作登入驗證機制
使用者輸入密碼錯誤	跳出警告視窗
使用者輸入資料錯誤	於使用者輸入資料錯誤的項目顯示訊息以供更正
資料不一致(資料庫端)	本系統的主程式將於使用者新增資料的時候自動檢查帳號資料與群組資料的一致性當系統偵測出不一致性時， 會顯示相關訊息，且使用者無法進入登入畫面
資料不一致(使用者輸入端)	於前端的資料送至伺服端的應用程式時，再進行檢查。 當子系統一偵測出不一致性時，相關的訊息將由本子 系統由前端通知使用者。

2.3 Report Generating Subsystem

2.3.1 Subsystem Characteristics

累積流程圖 (Cumulative Flow Diagram, CFD) 之功能為從資料庫抓取資料並產生出目前專案之累積流程圖。

2.3.2 Establish Technical Solution Criteria

Jquery 報表		圖片
易學性	高，簡單的 JS 語法跟 HTML 語法	低，必須學習使用撰寫產生圖片報表的 Library
可攜性	高，在主流的 Browser 都可以使用	高，圖片不會因 Browser 而改變
安全性	無，無安全性考量	無，無安全性考量
擴充性	高，可以對表格做修改	低，使用者修改報表困難
維護性	高，修改 JS 跟 HTML 都很容易	中，必須重新配置圖片資料

2.3.3 Selected Subsystem Solution

Jquery 報表		圖片
易學性	5	2
可攜性	5	5
安全性	0	0
擴充性	5	1
維護性	5	2
加總	20	10

Priorities {Scale: 1= Most bad(difficult), 5= Most good(easy), 0= Not necessary}

2.3.4 Error Detection and Recovery

Error	直接實作登入驗證機制
使用者未登入	無法產生報表，並且導入到登入畫面
使用者輸入資料錯誤	於使用者輸入資料錯誤的項目顯示訊息以供更正

Section 3 Detailed of Subsystem and Interface Description

3.1 Detailed System

3.1.1 Use Cases Analysis

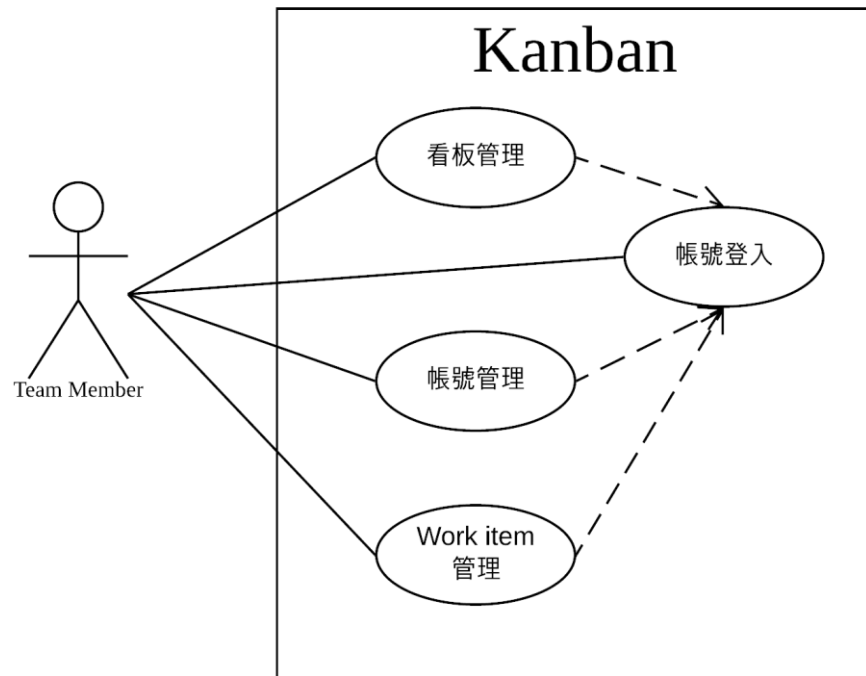


圖 3-1 Use case diagram

UC#01. 帳號登入

Use Case Session	Comment
Scope	Kanban 系統
Level	Subfunction-level
Primary Actor	- Team member
Stakeholders and Interests	- Team members: 登入以進入團隊看板
Preconditions	none
Success Guarantee	完成登入可以進入系統操作看板
Main Success Scenario	<ol style="list-style-type: none"> 1. Team member 開啟登入頁面 2. Team member 輸入帳號及密碼 3. Team member 點選登入 4. 進入 Kanban 系統
Extensions	<p>2a. 尚未註冊帳號：</p> <ol style="list-style-type: none"> 1. Team member 點選註冊帳號 2. 系統顯示帳號註冊畫面 3. Team member 填寫使用者資訊及帳號和密碼 <p>如果資料不完整或帳號已經被申請過，回到 2a.3 繼續編輯</p> <ol style="list-style-type: none"> 4. 系統顯示完成註冊，並將 Team member 導回登入頁面 5. Team member 回到步驟 2 <p>3a. 帳號或密碼輸入錯誤：</p> <ol style="list-style-type: none"> 1. 系統顯示登入錯誤訊息 2. 回到步驟 2，Team member 重新填寫帳號及密碼
Special Requirements	none
Technology and Data Variations List	none
Frequency of Occurrence	經常

UC#02. Workitem 管理

Use Case Session	Comment
Scope	Kanban 系統
Level	Subfunction-level
Primary Actor	- Team member
Stakeholders and Interests	- Team members: 編輯 Work item 內容
Preconditions	<ol style="list-style-type: none"> 1. Team member 為登入狀態 2. Team member 擁有團隊的看板 3. Team member 進入團隊看板
Success Guarantee	Team member 完成想要編輯的 Work item
Main Success Scenario	<ol style="list-style-type: none"> 1. Team member 在看板內點選新增 Work item 2. Team member 開啟新增的 Work item 3. Team member 編輯 title 和 description 4. Team member 點選儲存
Extensions	<ol style="list-style-type: none"> 3a. 編輯 TodoList : <ol style="list-style-type: none"> 1. Team member 點選新增待辦事項 2. Team member 填寫代辦事項內容 3b. 編輯工作分配 : <ol style="list-style-type: none"> 1. Team member 點選工作分配 2. 系統顯示看板內所有成員 3. Team member 選擇或取消分配的成員 3c. 編輯標籤 : <ol style="list-style-type: none"> 1. Team member 點選新增標籤 2. 系統顯示可選擇的顏色及填寫欄位 3. Team member 選擇顏色及填寫標籤內容 3d. 編輯到期日 : <ol style="list-style-type: none"> 1. Team member 點選到期日 2. 系統顯示 Date picker 3. Team member 選擇到期日期和時間 4a. 刪除 Work item : <ol style="list-style-type: none"> 1. Team member 點選刪除 2. 系統顯示刪除確認對話框

	3. Team member 點選確認 4. 系統關閉編輯畫面，並移除看板內的 Work item
Special Requirements	None
Technology and Data Variations List	none
Frequency of Occurrence	經常

UC#03. 看板管理

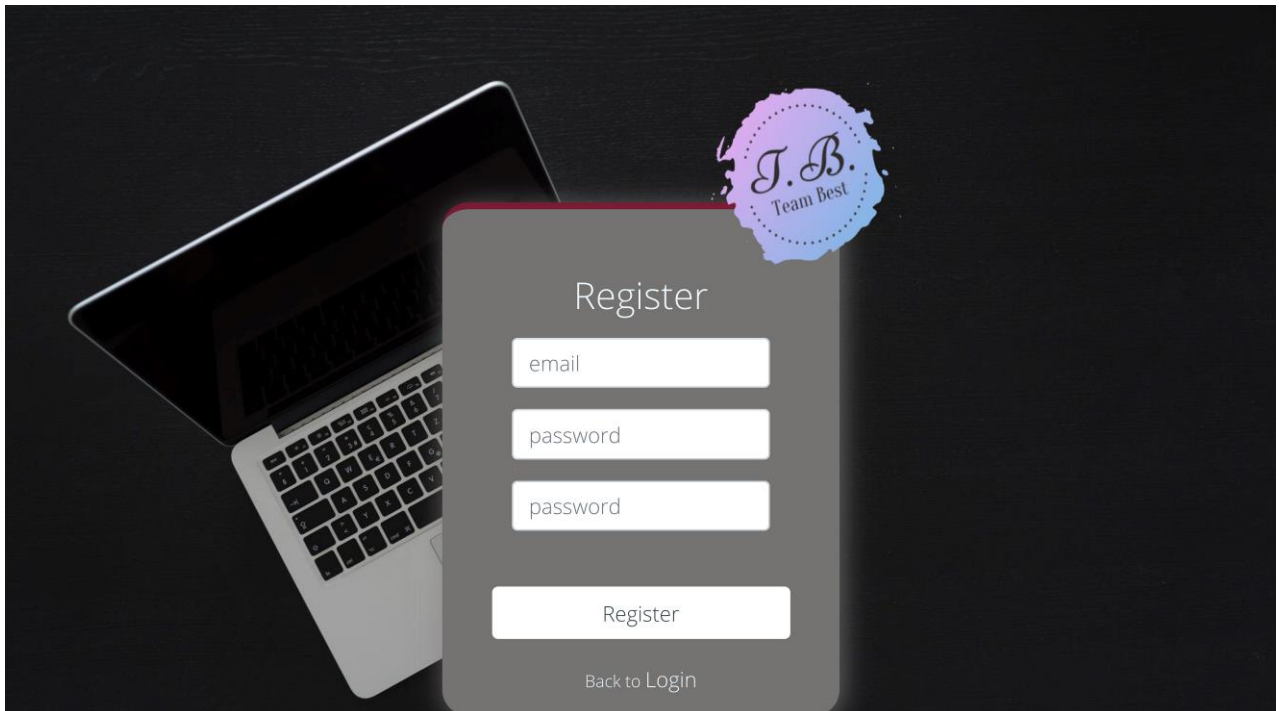
Use Case Session	Comment
Scope	Kanban 系統
Level	User goal
Primary Actor	- Team member
Stakeholders and Interests	- Team members: 編輯看板內容
Preconditions	Team member 為登入狀態
Success Guarantee	Team member 完成想要編輯的看板內容
Main Success Scenario	<ol style="list-style-type: none"> 1. Team member 點選進入看板 2. Team member 編輯項目 3. 系統更新編輯項目
Extensions	<p>2a. 成員管理：</p> <ol style="list-style-type: none"> 1. Team member 點擊加入成員 2. 系統顯示 email 欄位 3. Team member 輸入新成員的 email <p>2b. Work item 管理參考 UC02</p> <p>2c. 移動 Work item 工作階段：</p> <ol style="list-style-type: none"> 1. Team member 選擇 Work item，並移動到目的工作階段 <p>2d. 查看報表：</p> <ol style="list-style-type: none"> 1. Team member 點擊產生報表 2. 系統顯示報表 <p>2e. 離開團隊：</p> <ol style="list-style-type: none"> 1. Team member 點選個人頭像及退出
Special Requirements	none
Technology and Data Variations List	none
Frequency of Occurrence	經常

UC#04: 帳號管理

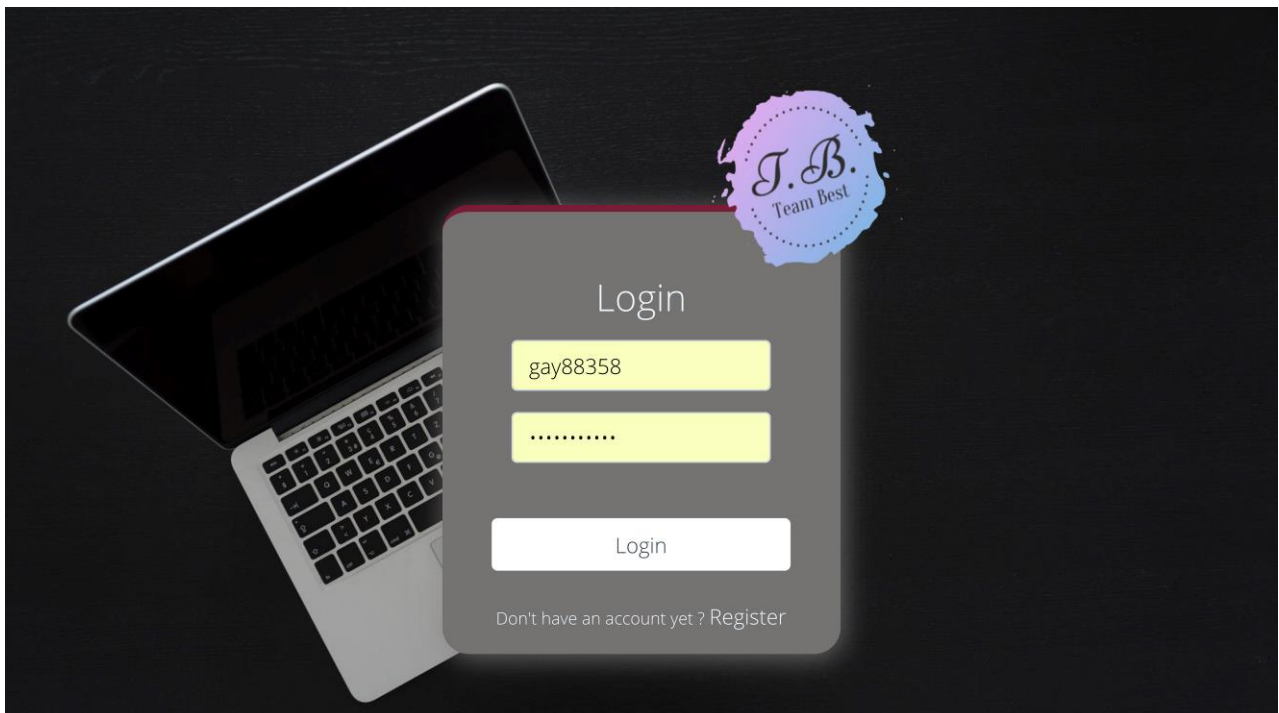
Use Case Session	Comment
Scope	Kanban 系統
Level	User goal
Primary Actor	- Team Member
Stakeholders and Interests	- Team members: 編輯個人資料
Preconditions	Team member 為登入狀態
Success Guarantee	完成個人資料編輯
Main Success Scenario	<ol style="list-style-type: none"> 1. Team member 點選個人頭像進入個人資料編輯頁面 2. Team member 變更個人資料（姓名、電話.....） 3. 點擊儲存
Extensions	2a. 新增看板： <ol style="list-style-type: none"> 1. Team member 點選新增看板 2. 系統顯示 title 欄位 3. Team member 填寫新看板名稱 4. 系統顯示新看板
Special Requirements	none
Technology and Data Variations List	none
Frequency of Occurrence	經常

3.1.2 User Interfaces Analysis

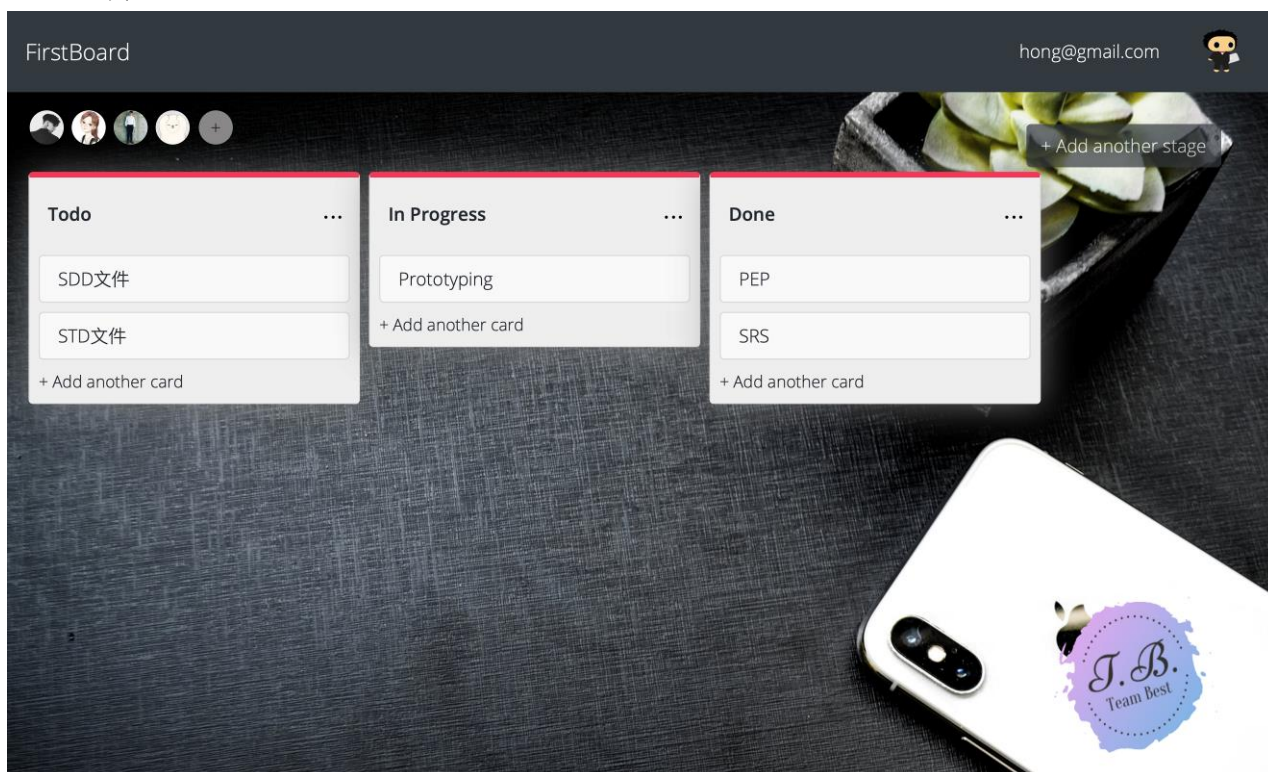
一、註冊介面



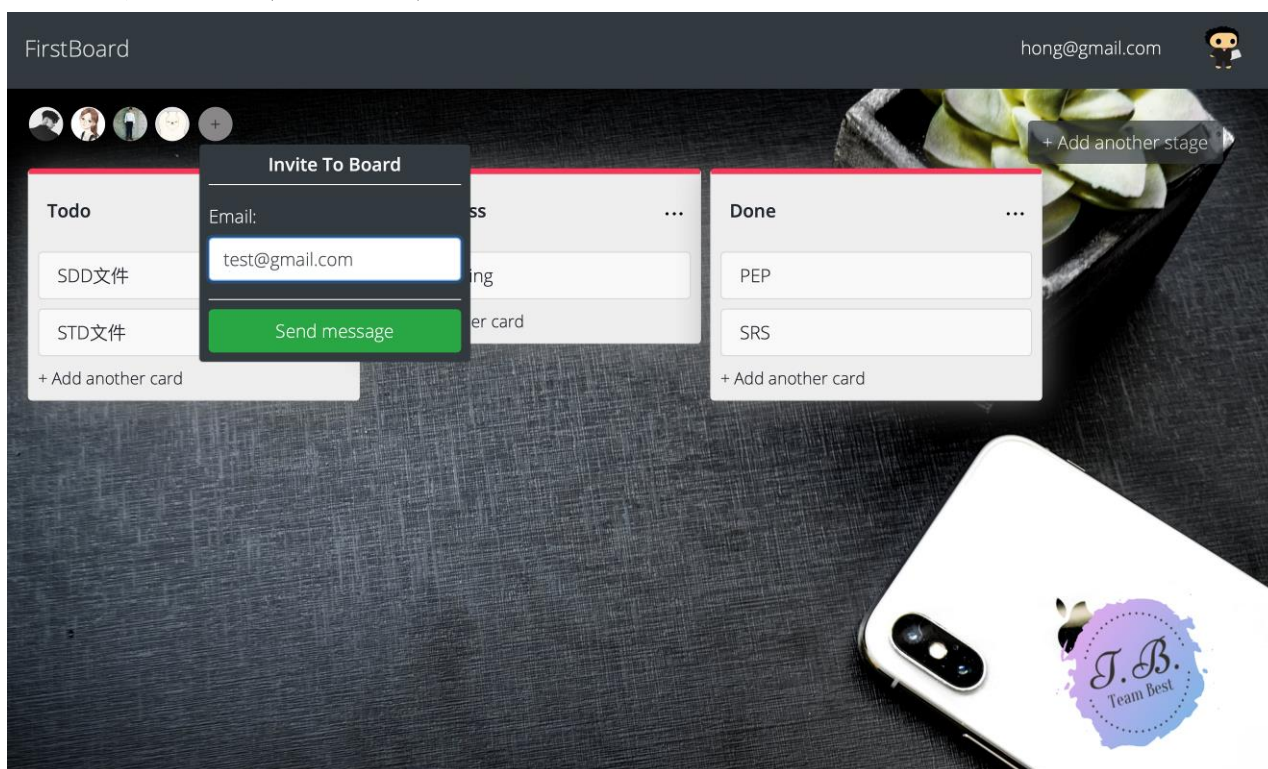
二、登入介面



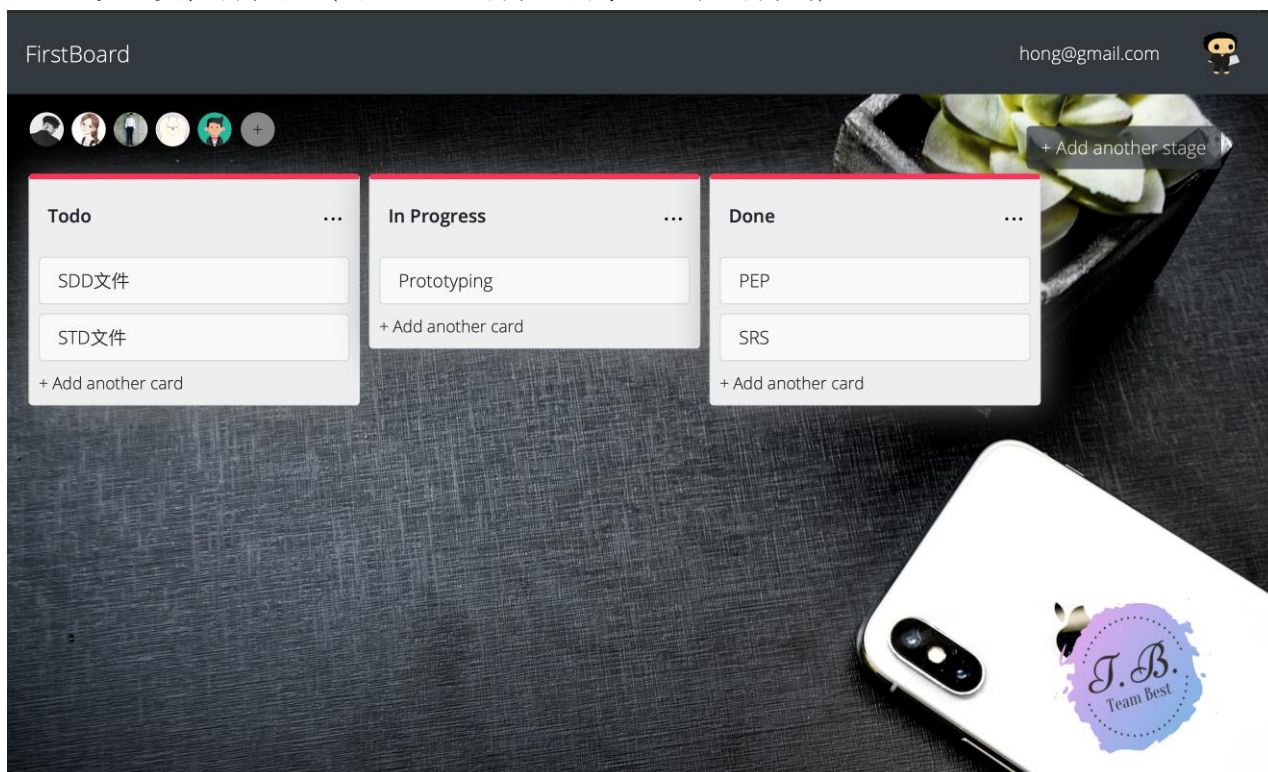
三、首頁



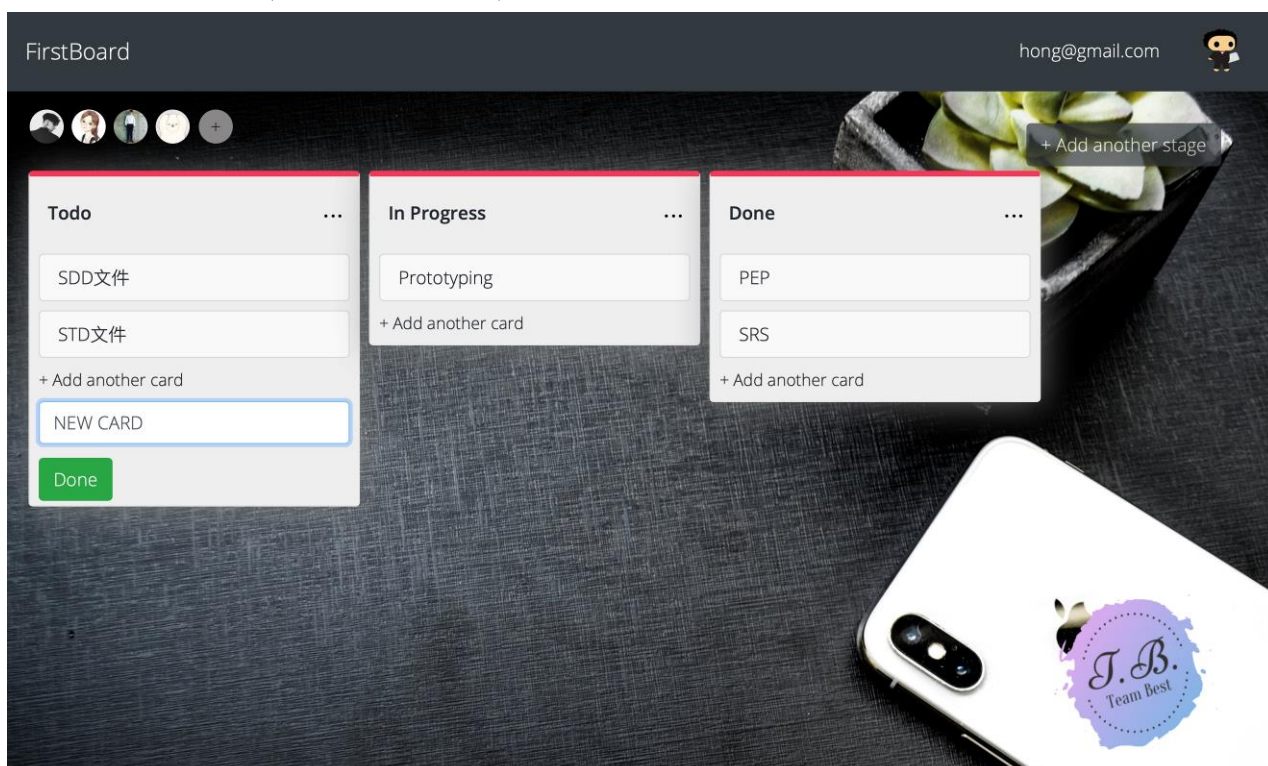
四、邀請團員介面(輸入 email)



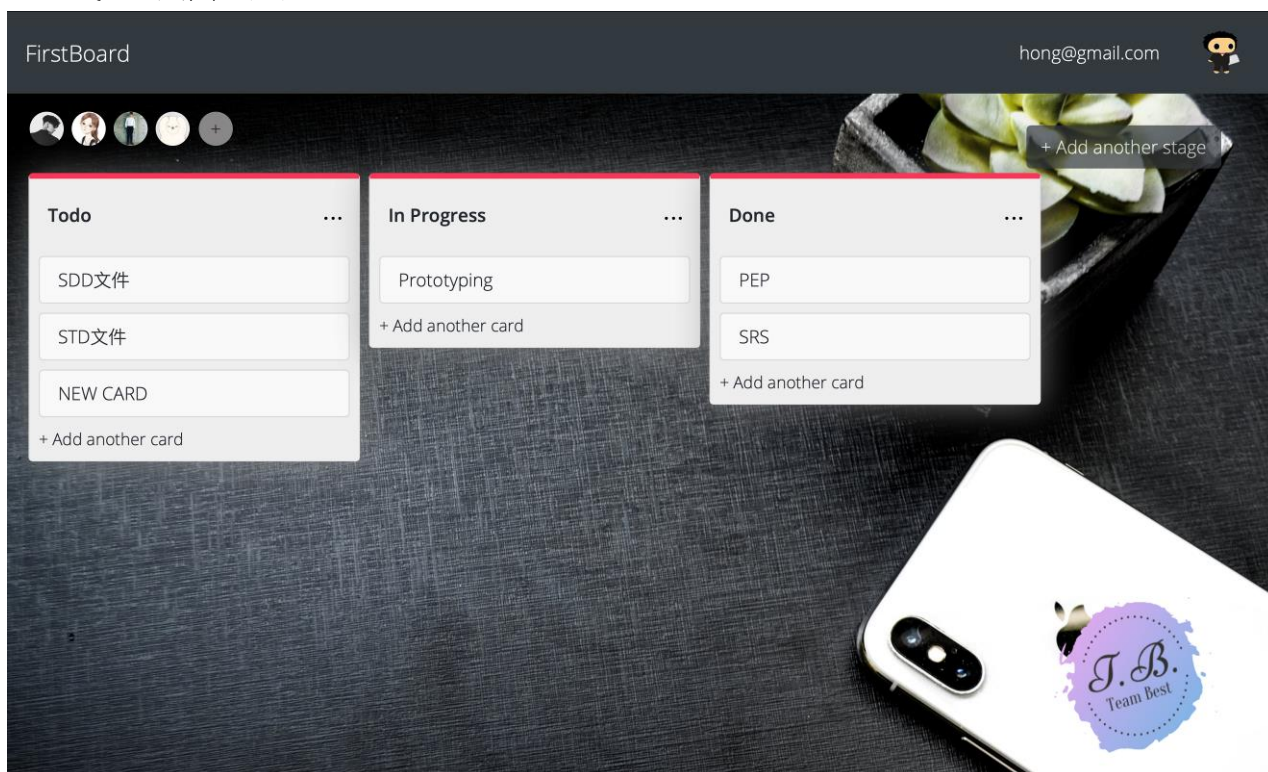
五、成功邀請團員介面(新加入的團員的頭像出現在團員列)



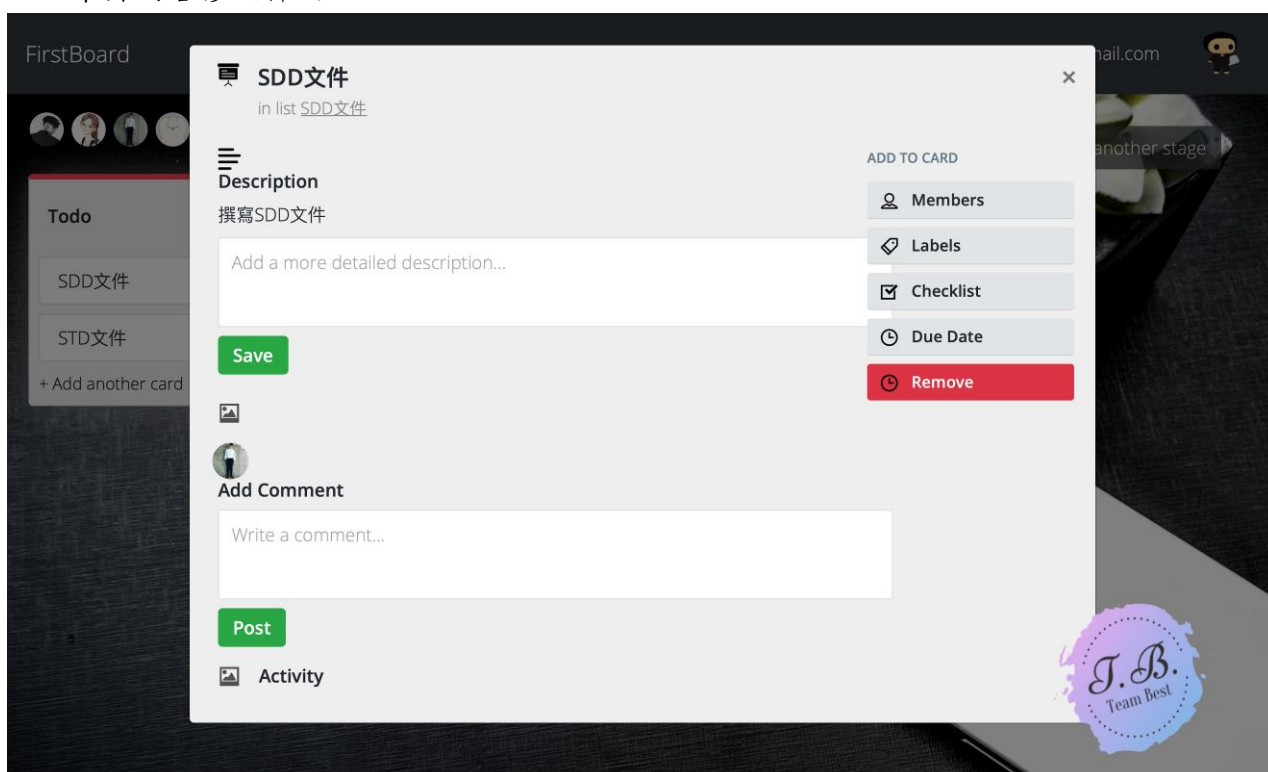
六、新增卡片介面(輸入新卡片名稱)



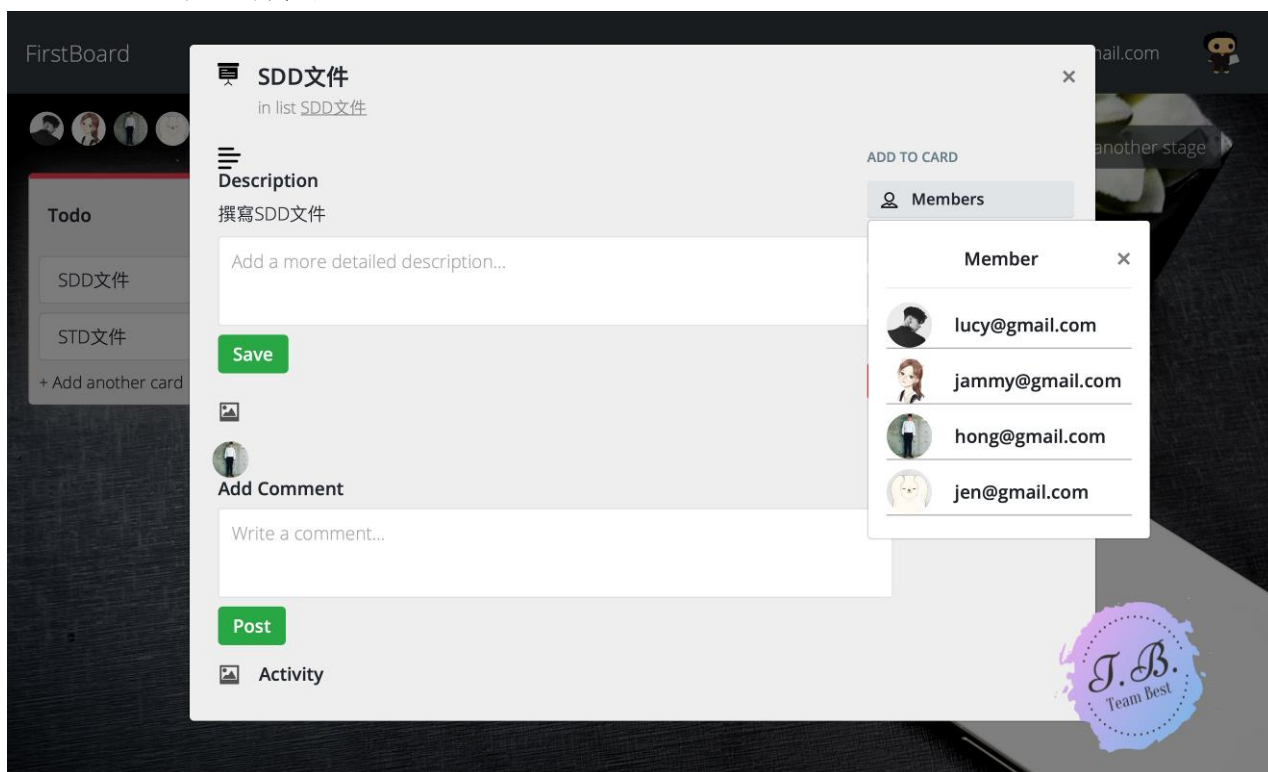
七、成功新增卡片介面



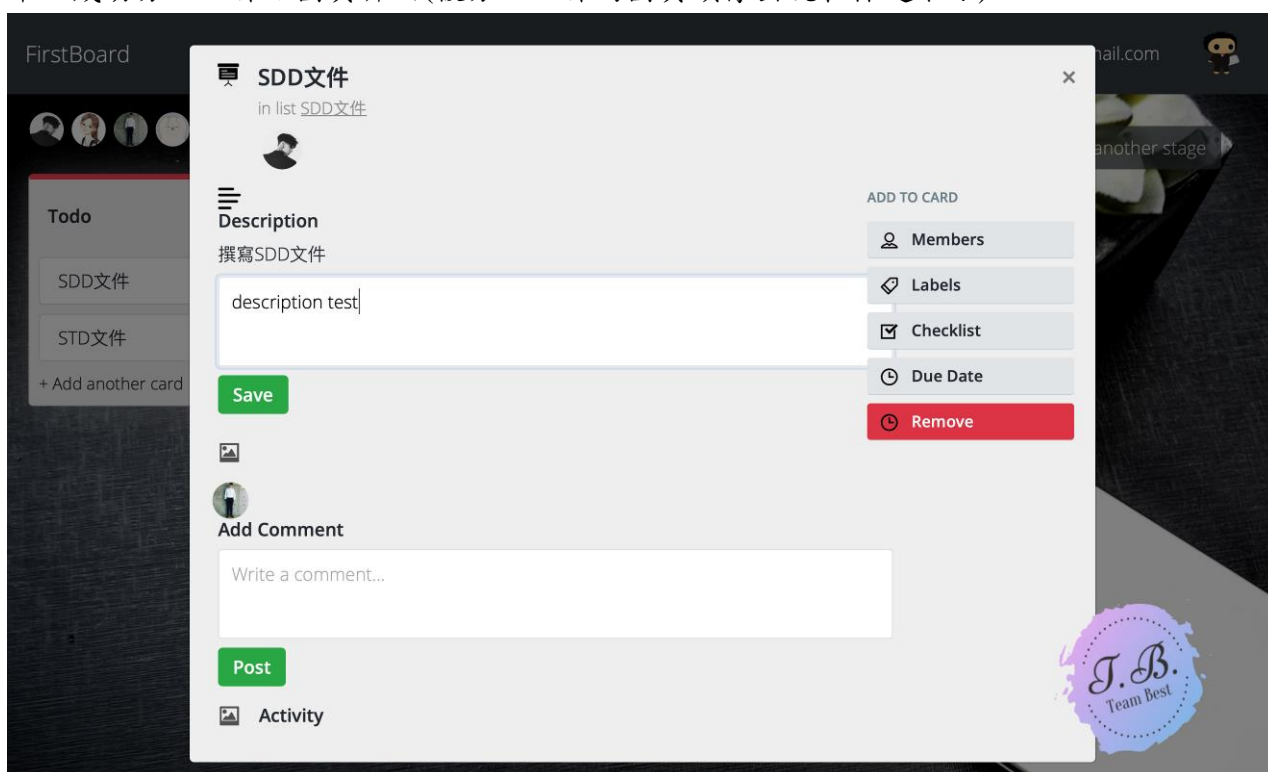
八、卡片內容修改介面



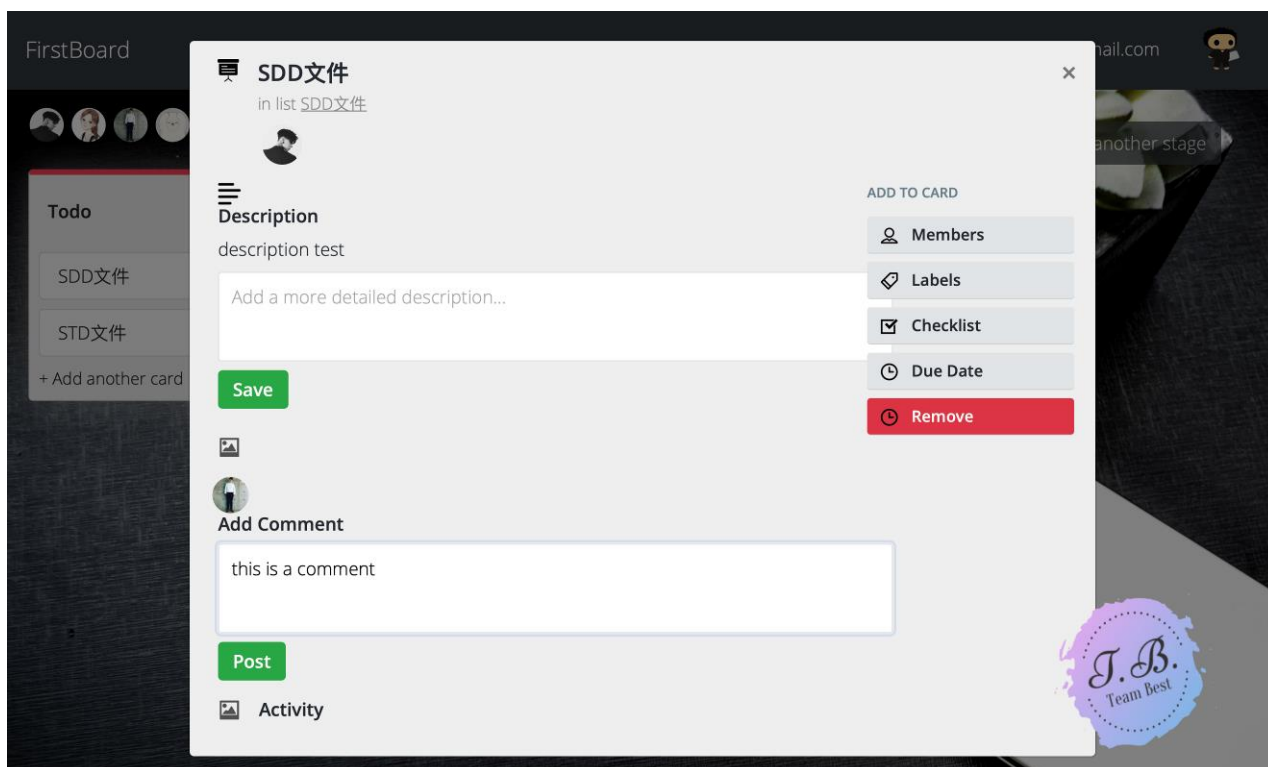
九、分配工作給團員介面



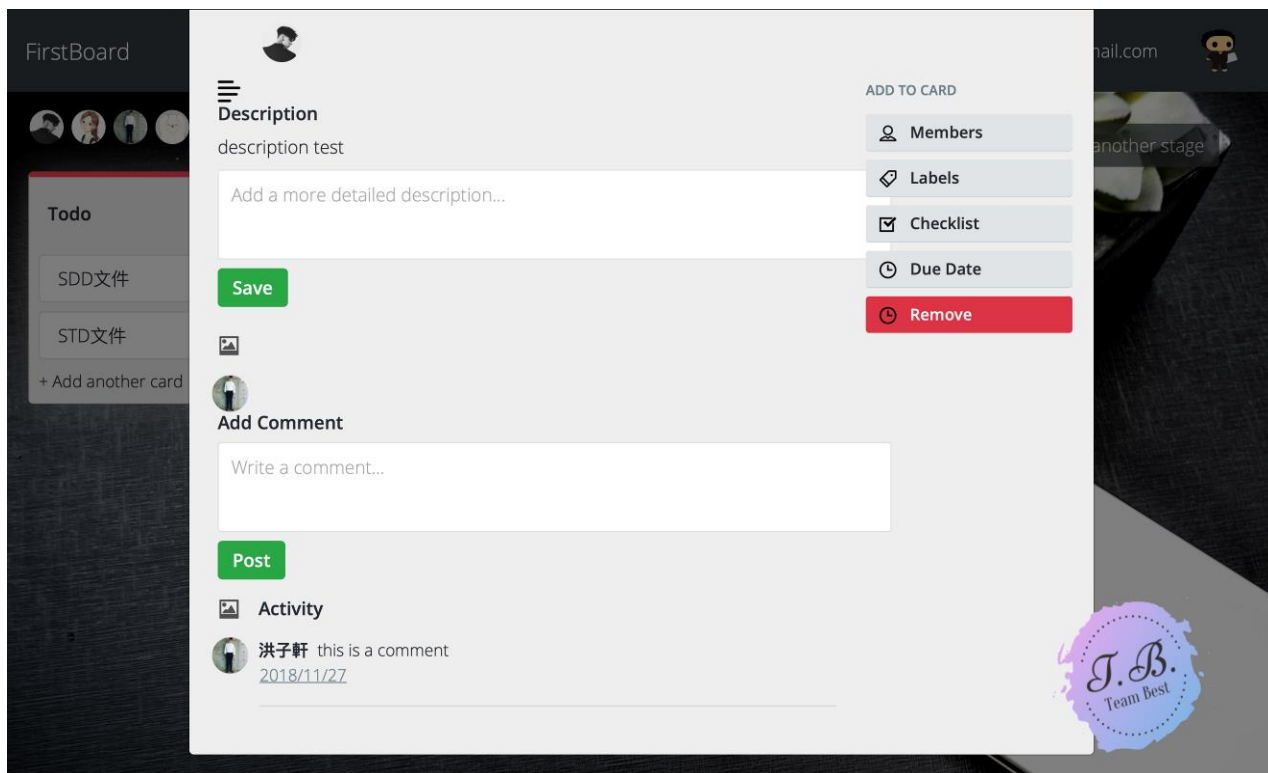
十、成功分配工作給團員介面(被分配工作的團員頭像出現在標題下方)



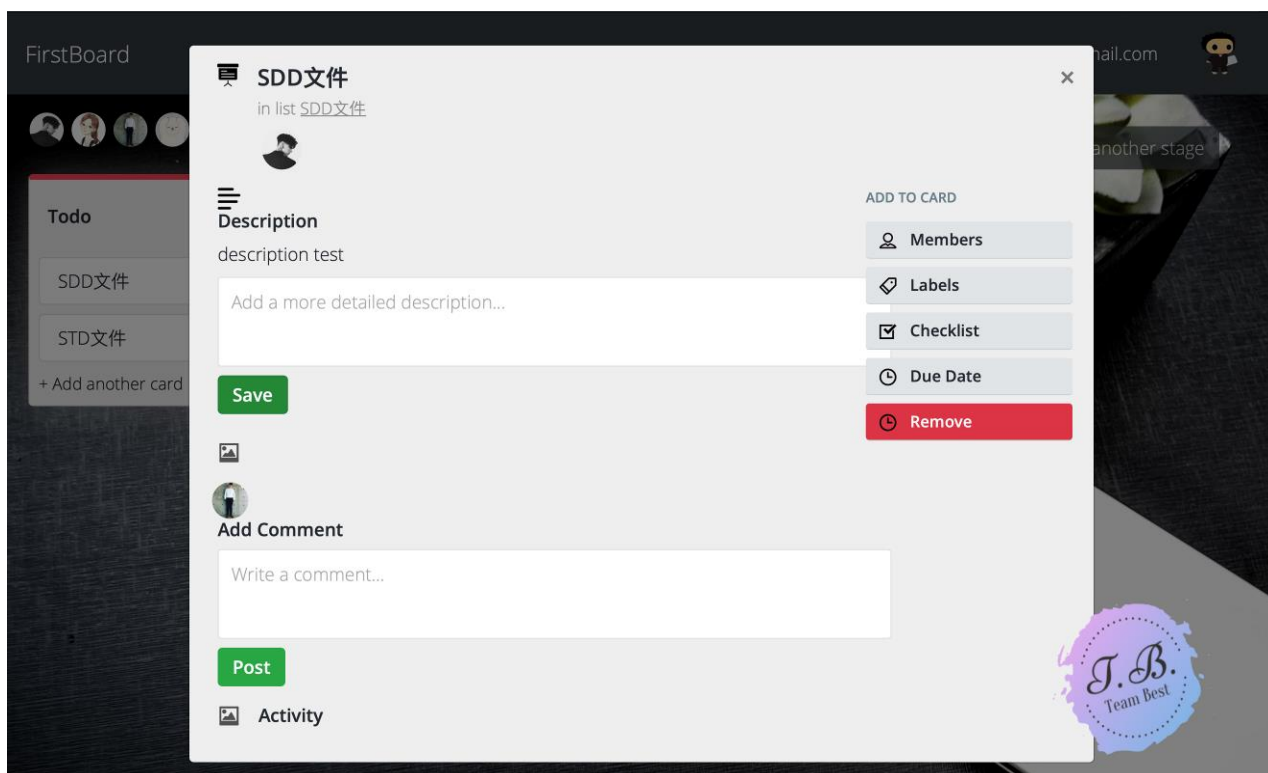
十一、卡片留言介面(在留言欄位裡輸入文字)



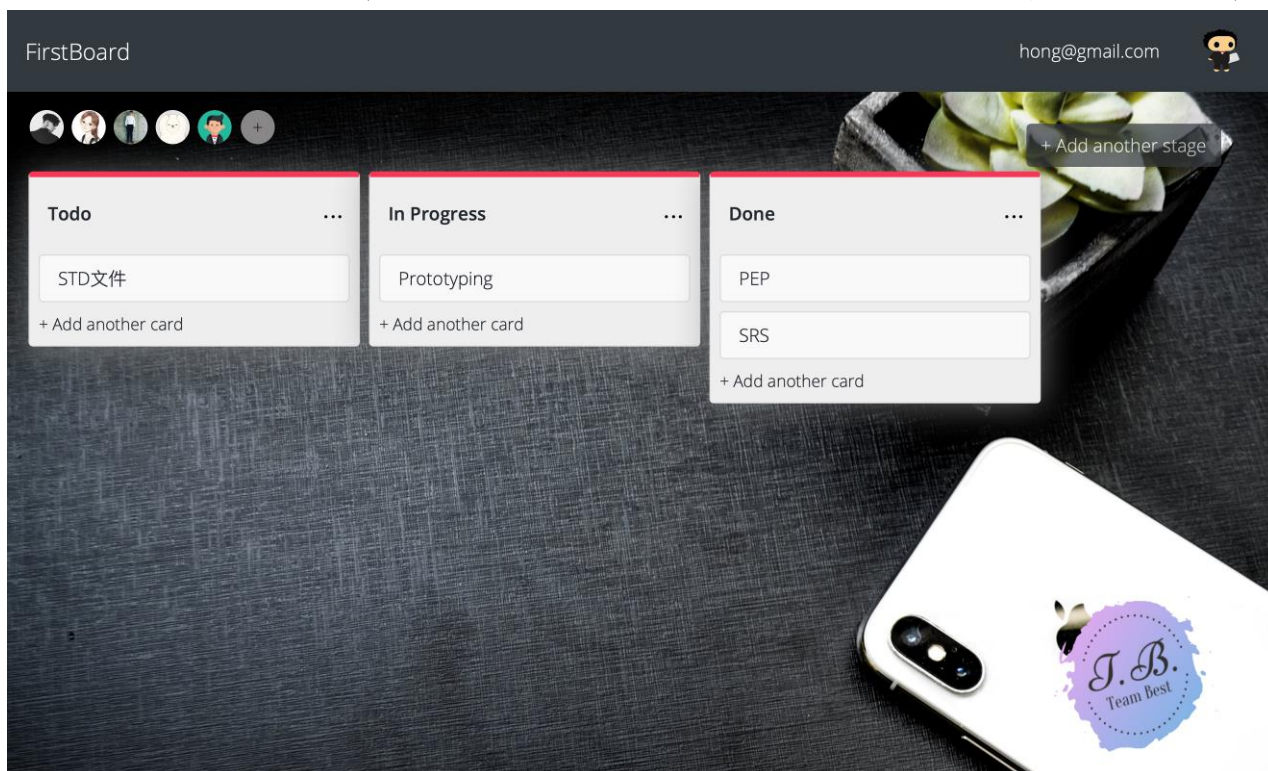
十二、成功在卡片內容留言介面(留言出現在 Activity 裡)



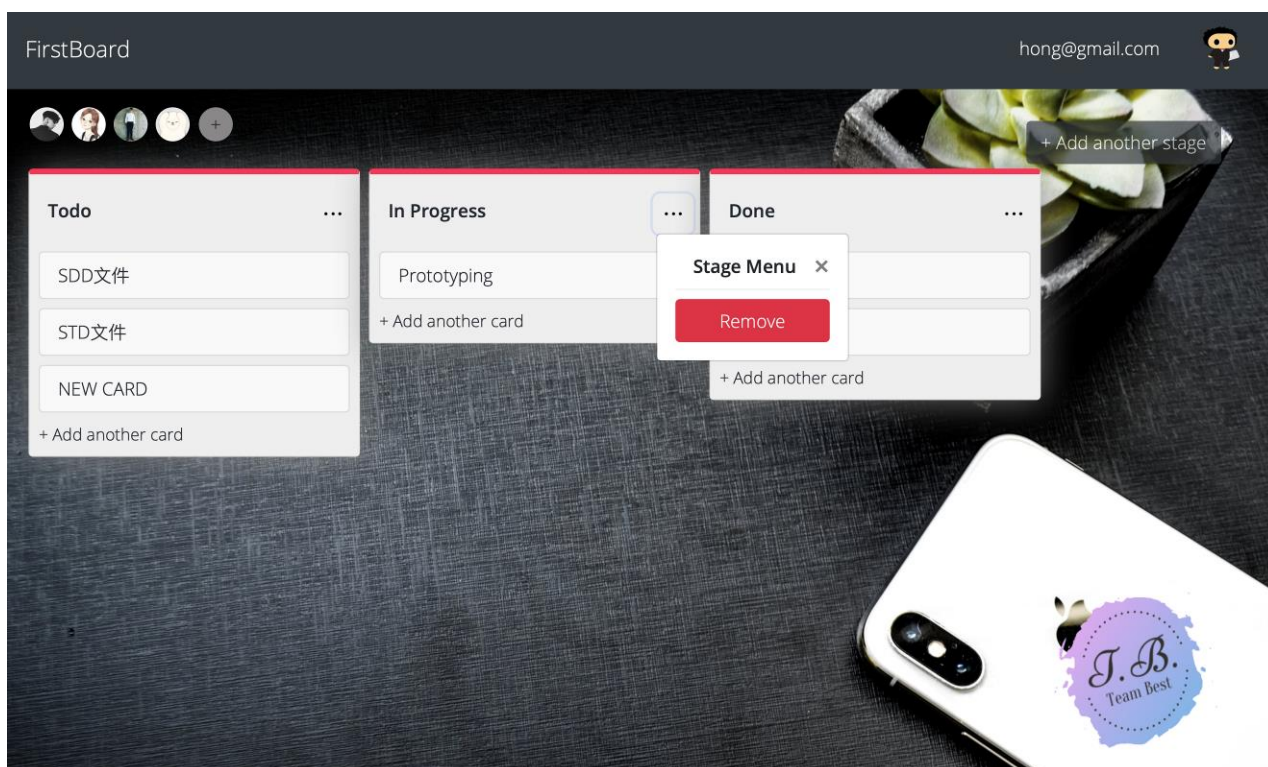
十三、移除卡片介面(點選 Remove 按鈕)



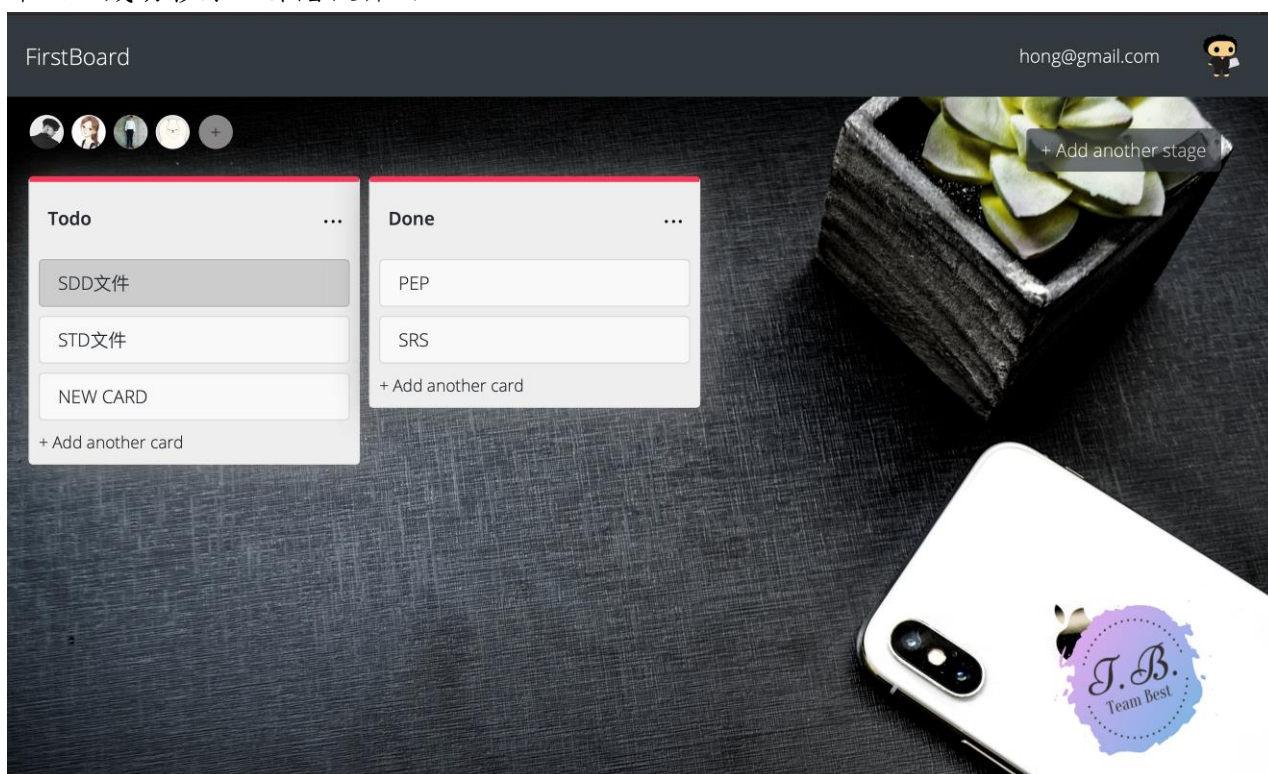
十四、成功移除卡片介面(系統關閉卡片修改視窗，被移除的卡片從原所屬工作階段移除)



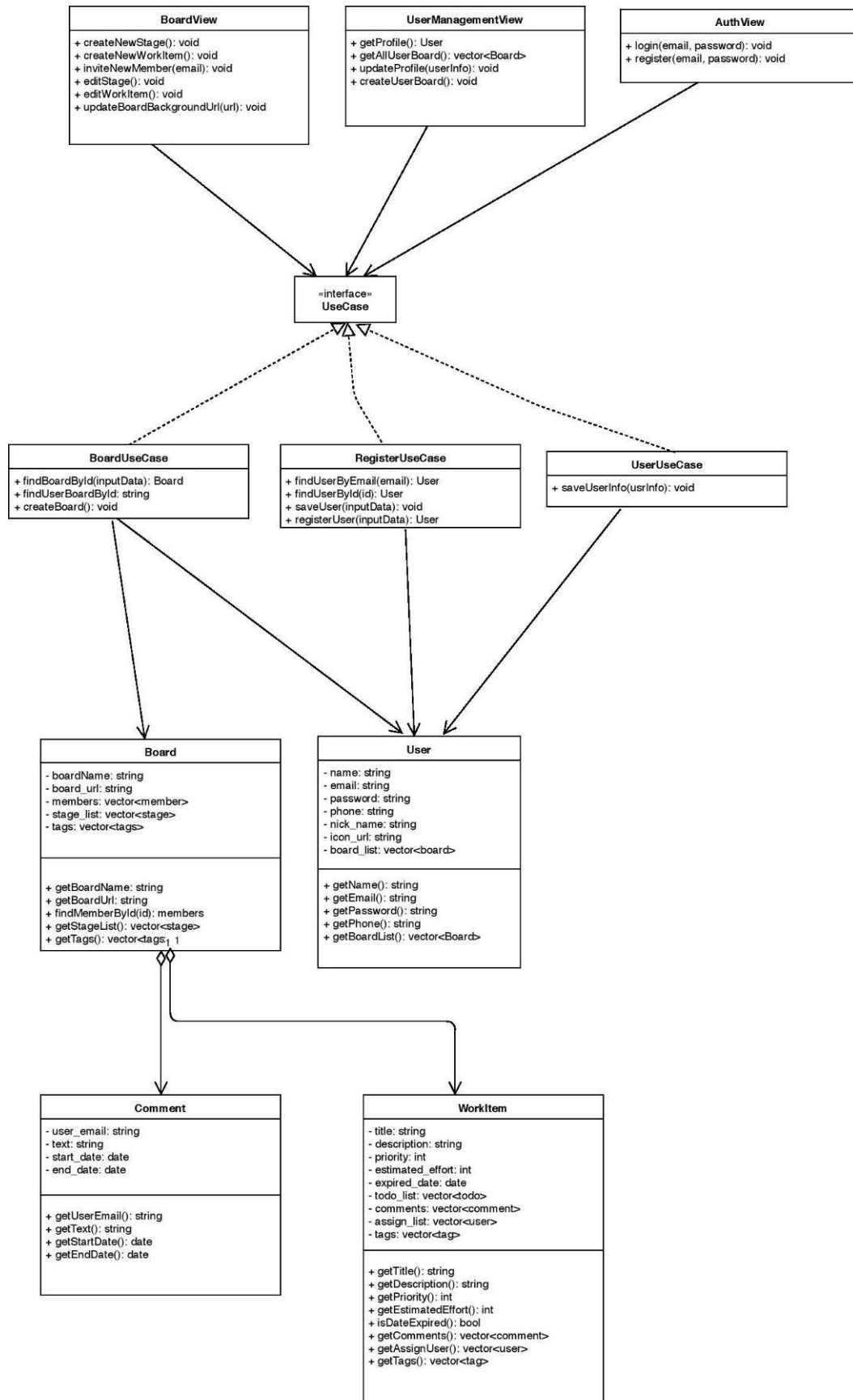
十五、移除工作階段介面(點選工作階段標題旁的選項按鈕，介面出現刪除按鈕)



十六、成功移除工作階段介面



3.1.3 Static Model



我們用表格描述 Class diagram 中的物件，以幫助設計者瞭解與使用物件：

Class No.	KBS-CL01
Class Name	BoardView
Responsibility	負責提供一個介面讓使用者操作看板內容。
Collaboration	UseCase
Related Subsystem	BS Module, UTMS Module, RGS Module

Class No.	KBS-CL02
Class Name	UserManagementView
Responsibility	負責提供一個介面讓使用者管理帳戶資訊。
Collaboration	UseCase
Related Subsystem	BS Module, UTMS Module

Class No.	KBS-CL03
Class Name	AuthView
Responsibility	負責處理使用者的註冊和登入。
Collaboration	UseCase
Related Subsystem	UTMS Module

Class No.	KBS-CL04
Class Name	UseCase
Responsibility	負責提供一個實作工作介面。
Collaboration	None
Related Subsystem	BS Module, UTMS Module, RGS Module

Class No.	KBS-CL05
Class Name	BoardUseCase
Responsibility	負責處理看板操作。
Collaboration	UseCase, Board, User
Related Subsystem	BS Module, UTMS Module, RGS Module

Class No.	KBS-CL06
Class Name	RegisterUseCase
Responsibility	負責處理使用者註冊。
Collaboration	UseCase, User
Related Subsystem	BS Module, UTMS Module

Class No.	KBS-CL07
Class Name	UserUseCase
Responsibility	負責處理使用者修改帳戶資訊。
Collaboration	UseCase, User
Related Subsystem	BS Module, UTMS Module

Class No.	KBS-CL08
Class Name	Board
Responsibility	負責處理看板的細節操作。
Collaboration	Comment, WorkItem
Related Subsystem	BS Module, UTMS Module, RGS Module

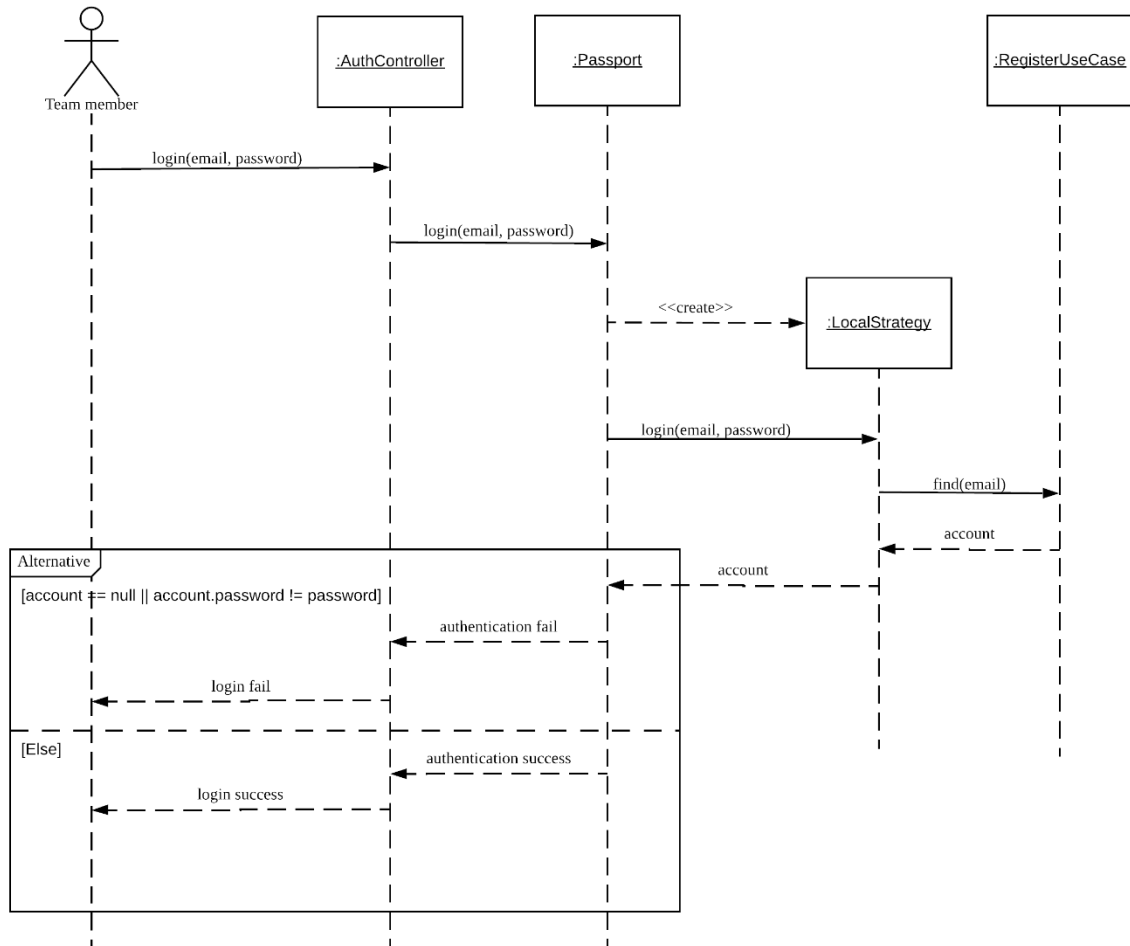
Class No.	KBS-CL09
Class Name	User
Responsibility	負責處理使用者的細節操作。
Collaboration	None
Related Subsystem	BS Module, UTMS Module

Class No.	KBS-CL10
Class Name	Comment
Responsibility	負責處理留言的細節操作。
Collaboration	None
Related Subsystem	BS Module, UTMS Module, RGS Module

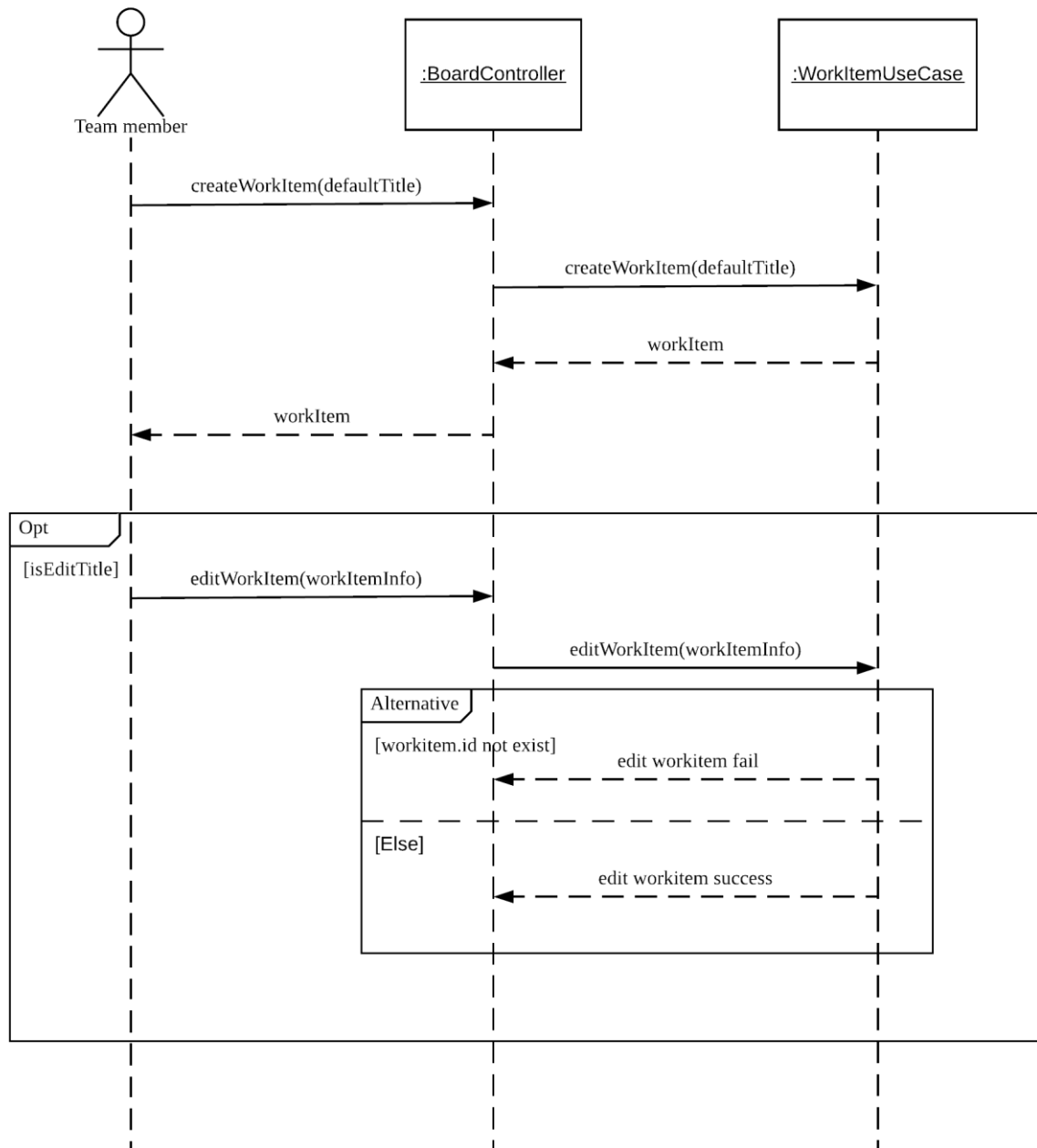
Class No.	KBS-CL11
Class Name	WorkItem
Responsibility	負責處理卡片的細節操作。
Collaboration	None
Related Subsystem	BS Module, UTMS Module, RGS Module

3.1.4 Dynamic Model

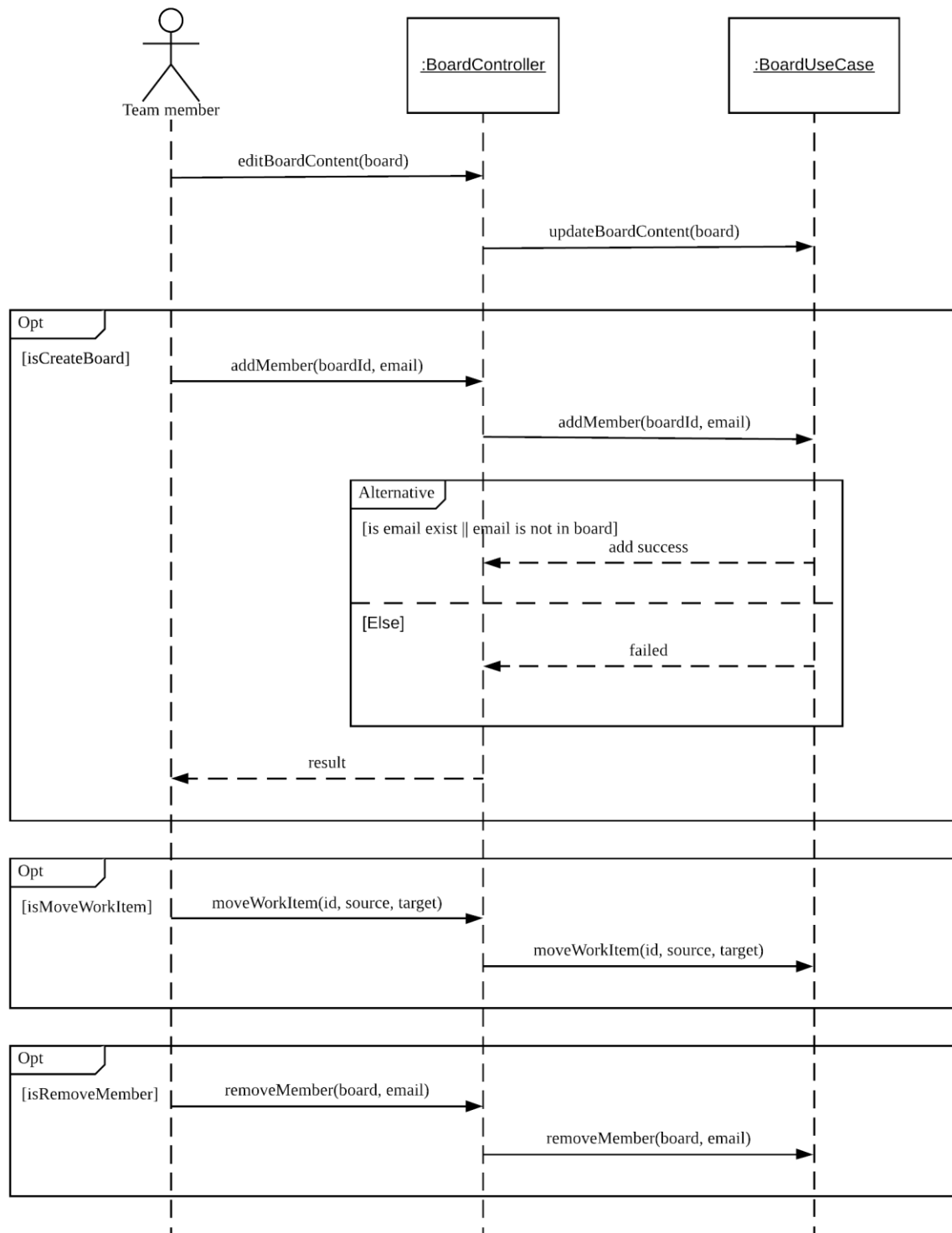
UC#01 帳號登入



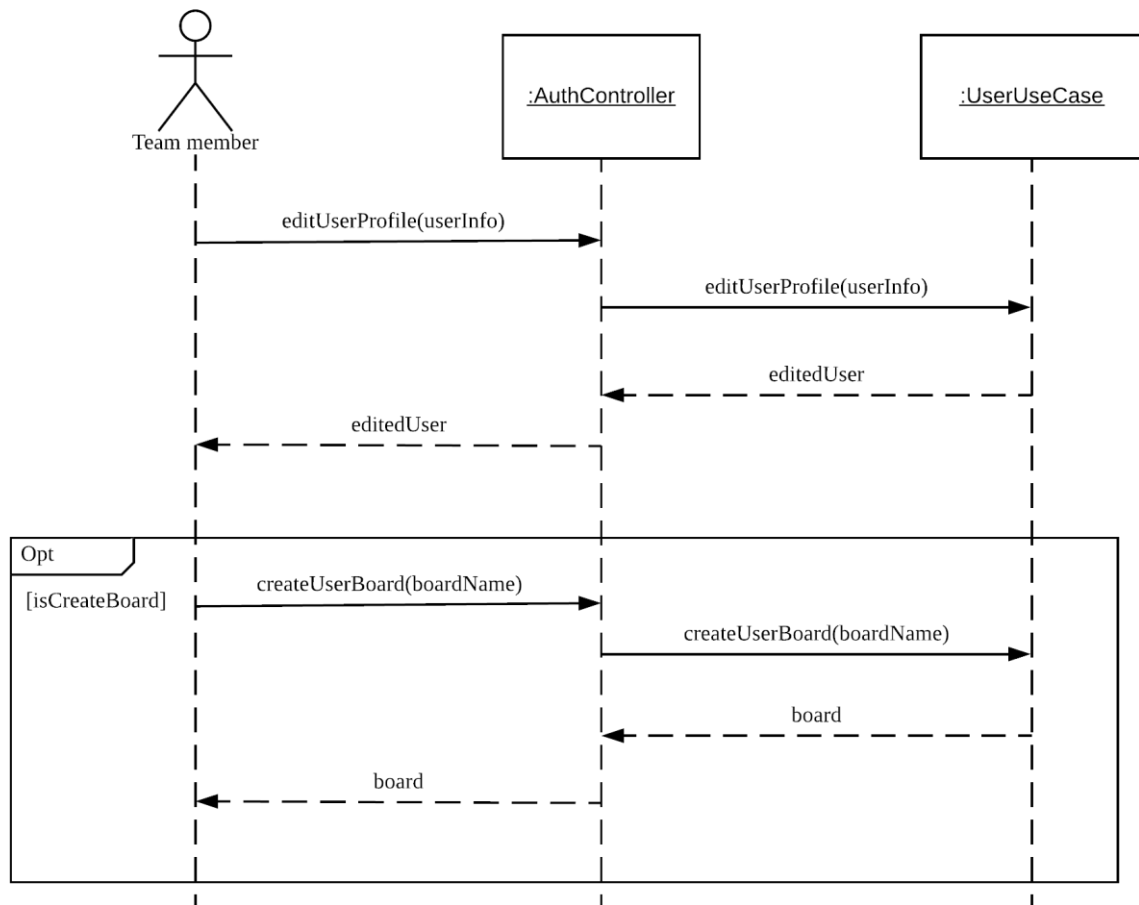
UC#02 Work item 管理



UC#03 看板管理



UC#04 帳號管理



3.2 Subsystem Internal/External Interface Requirement and Design

3.2.1 Internal Interface Requirements

需求編號	優先順序	需求描述
KBS-I-01	1	UTMS 與資料庫間必預能傳送與接收使用者資料。
KBS-I-02	1	BS 與資料庫間必預能傳送與接收專案工作資料。
KBS-I-03	1	RGS 與資料庫間必預能傳送與接收專案資料但不能更改。

3.2.2 External Interface Requirements

需求編號	優先順序	需求描述
KBS-E-01	1	UTMS 提供圖形化介面供使用者操作。
KBS-E-02	1	UTMS 允許使用者登入，更改帳戶資料。
KBS-E-03	1	BS 提供圖形化介面供使用者操作。
KBS-E-04	1	BS 允許使用者新增或檢視既有的專案資訊。
KBS-E-05	1	BS 允許特定使用者即時操作專案工作。
KBS-E-06	1	RGS 提供圖形化介面供使用者操作。
KBS-E-07	1	RGS 需產生專案報表可供使用者查詢。

3.3 Traceability Matrix – Requirements vs Components

	KBS- CL01	KBS- CL02	KBS- CL03	KBS- CL04	KBS- CL05	KBS- CL06	KBS- CL07	KBS- CL08	KBS- CL09	KBS- CL10	KBS- CL11
KBS-F-01	O	O		O	O			O		O	O
KBS-F-02	O	O	O	O	O		O				O
KBS-F-03		O		O	O		O	O			
KBS-F-04	O			O	O			O			
BS-F-01	O		O	O	O		O	O	O	O	
BS-F-02	O			O	O			O		O	O
BS-F-03	O			O	O			O		O	O
BS-F-04	O			O	O			O		O	O
BS-F-05	O		O	O	O			O		O	O
BS-F-06	O		O	O	O			O		O	O
UTMS-F-01		O	O	O		O	O		O		
UTMS-F-02	O	O	O	O		O	O		O		
UTMS-F-03		O	O	O			O		O		
UTMS-F-04		O	O	O							
RGS-F-01	O	O	O	O	O			O		O	O
RGS-F-02	O			O	O			O			

Glossary

Database	記錄資料的地方，提供新增、刪除、更新記錄的功能。
MySQL	MySQL 是一個開放源程式碼的關聯式資料庫管理系統，由於性能高、成本低、可靠性好，已經成為最流行的開源資料庫，被廣泛地應用在 Internet 上的中小型網站中。隨著 MySQL 的不斷成熟，它也逐漸用於更多大規模網站和應用。
SQL	Structured Query Language，關連式查詢語言，用來定義資料庫的結構，或是利用 SQL 對資料庫執行一些查詢、增加、刪除、更新記錄，到目前為止，SQL 是第一個，也是唯一的標準資料庫語言，受到廣泛的接受。
jQuery	jQuery 是一款免費且開放原始碼的 JavaScript 程式函式庫，由 John Resig 創建。授權協定為 GPL 和 MIT 許可證雙協定。jQuery 1.3 版中，引入全新的 CSS 選擇器引擎 Sizzle。目前已有 Google AJAX Libraries API 等公開站台提供 JQuery 的 js 的參照服務。
UI	用戶介面(User Interface)是介於使用者與硬體而設計彼此之間互動溝通相關軟體，目的在使得使用者能夠方便有效率地去操作硬體以達成雙向之互動，完成所希望借助硬體完成之工作，用戶介面定義廣泛，包含了人機互動與圖形使用者介面，凡參與人類與機械的信息交流的領域都存在著用戶介面。
JavaScript	JavaScript 是一種進階的，解釋執行的程式語言。JavaScript 是一門基於原型、函式先行的語言，是一門多範式的語言，它支援物件導向編程，指令式程式設計，以及函數語言程式設計。它提供語法來操控文字、陣列、日期以及正規表示式等，不支援 I/O，如網路、儲存和圖形等，但這些都可以由它的宿主環境提供支援。

References

- [1] 洪肇奎, 郭耀煌, 李允中, *Light-Weight Capability Maturity Model Integration (CMMI) for National Science Council (NSC) Open Source Project*, CSIE Program, Department of Engineering and Applied Science, NSC, 2004.

Appendices

A. Traceability Matrix Use Cases v.s. Classes

	KBS- CL01	KBS- CL02	KBS- CL03	KBS- CL04	KBS- CL05	KBS- CL06	KBS- CL07	KBS- CL08	KBS- CL09	KBS- CL10	KBS- CL11
UC#01		O	O	O		O	O		O		
UC#02	O			O	O			O		O	O
UC#03	O			O	O			O		O	O
UC#04		O		O			O		O		

B. Traceability Matrix Classes v.s. Classes

	KBS- CL01	KBS- CL02	KBS- CL03	KBS- CL04	KBS- CL05	KBS- CL06	KBS- CL07	KBS- CL08	KBS- CL09	KBS- CL10	KBS- CL11
KBS- CL01	O		O	O	O			O		O	O
KBS- CL02		O	O	O	O	O	O		O		
KBS- CL03		O	O								
KBS- CL04	O	O	O	O							
KBS- CL05	O				O			O		O	O
KBS- CL06		O	O	O		O					
KBS- CL07		O		O			O		O		
KBS- CL08	O	O		O	O			O		O	O
KBS- CL09		O	O	O			O		O		
KBS- CL10	O	O		O	O					O	O
KBS- CL11	O	O		O	O					O	O