



# Team #2

# Final Presentation

組長：余珮綺

組員：周雨柔、張書瑜、陸紀霖、吳思正、李維

# Table of contents

01

專案架構

02

User Story

03

DEMO

04

Test Case

05

Retrospective

06

about project

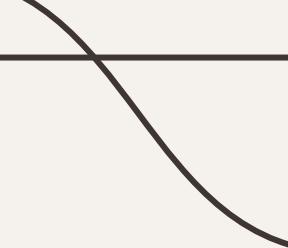
會議記錄與省思



# 01

# 專案架構

# Goals



01

便捷環境建置

02

全面系統功能

03

改善使用體驗

# System Requirements



## 建置Docker

方便其他開發人員建置符合  
系統的環境。

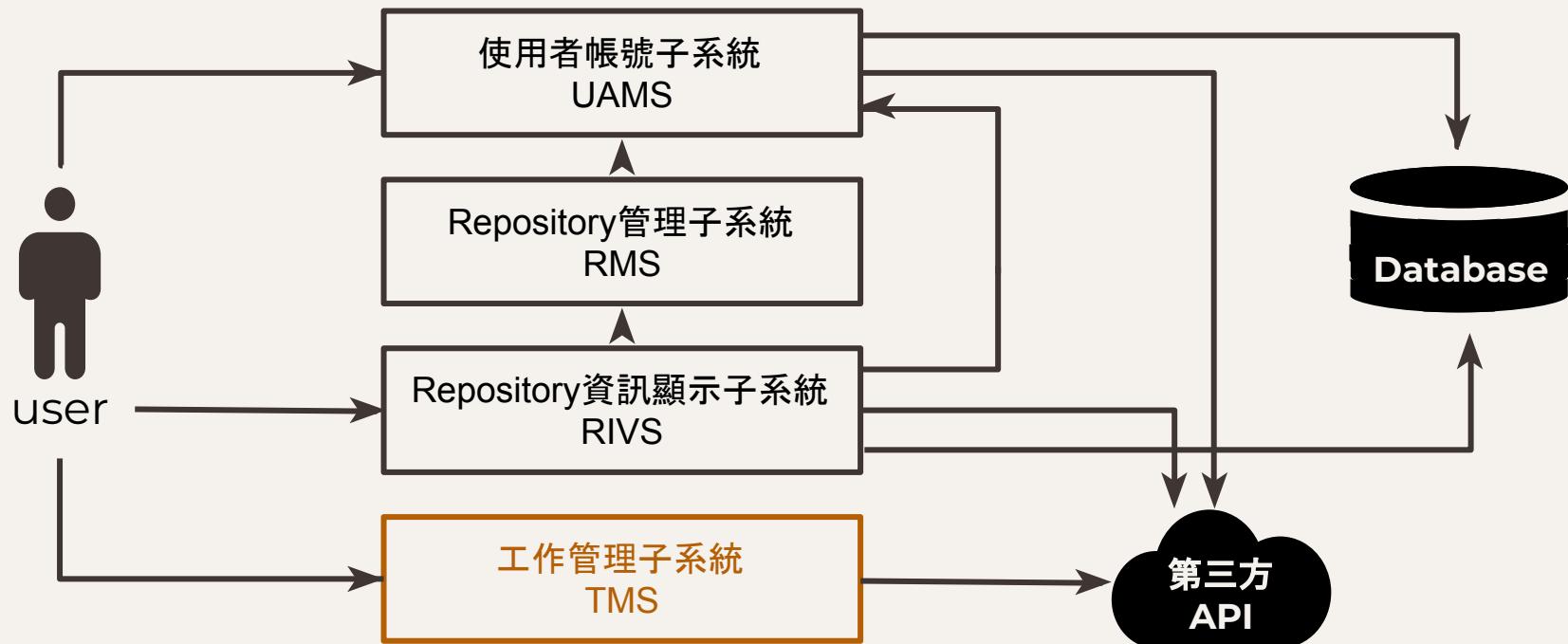


## TMS工作管理系統

新增 Trello 到專案，方便專  
案開發時的工作分配、進度討  
論。

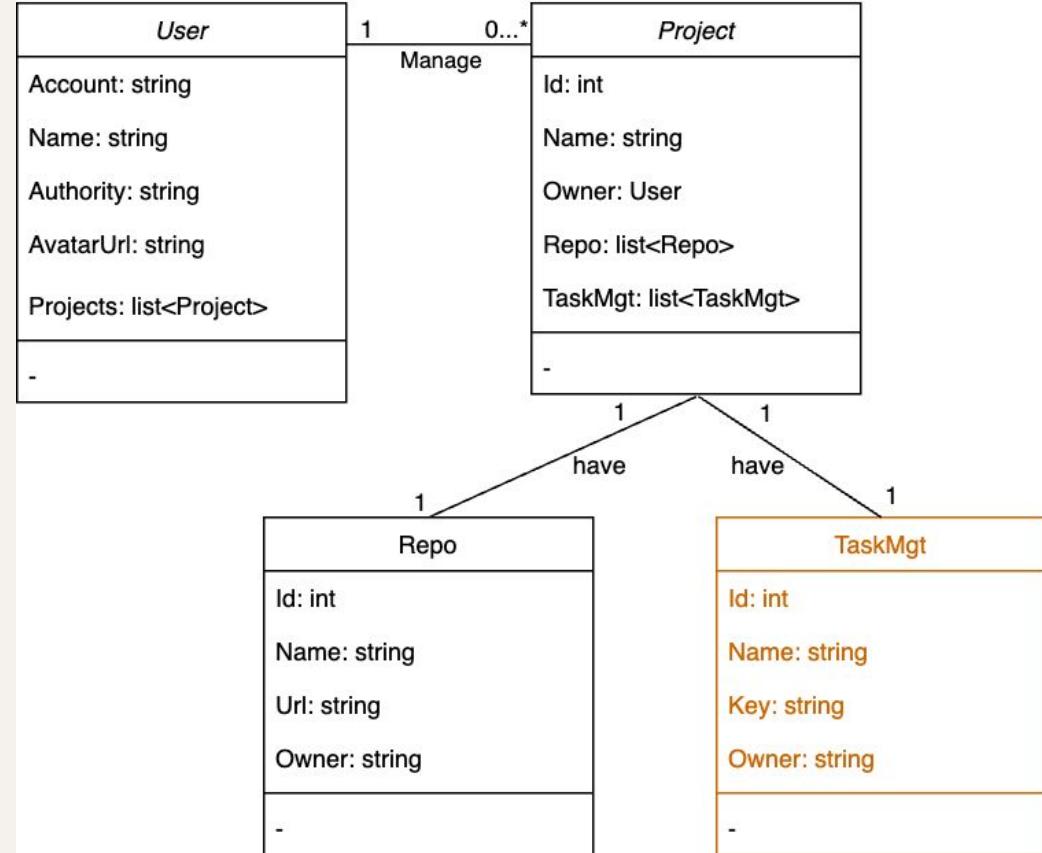
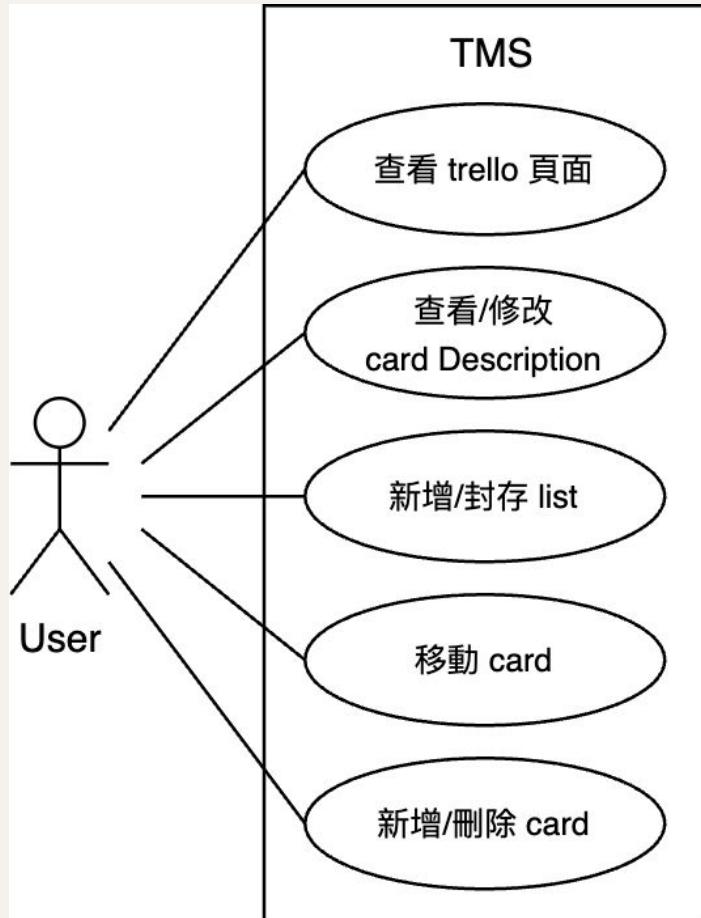


# 系統架構圖



## 2.6 功能性需求 (Functional Requirements)

需求編號	優先順序	需求描述
PD-F-01	1	提供帳戶管理及權限辯識功能。
PD-F-02	1	提供 Project 及 Repository 管理功能。
PD-F-03	1	提供 Repository 視覺化資訊功能。
PD-F-04	1	提供Task的管理功能。



查看trello頁面		修復資料庫 / 修改排版問題
查看/修改card description		修復按鈕功能
新增list		
封存list		
移動card		
新增card		修復位置錯誤問題
刪除card		

這些功能原先  
皆沒有  
屬於他們的  
unit test



# 02

# User Story

7+1

# TMS新增功能

身為一位使用者：

- 我希望能夠在 TMS系統上準確的連結上的Trello個人看板，並能及時更新。

驗收標準：

- 使用者能夠透過TMS系統順利連結到個人Trello看板，查看Trello看板。
- 資料庫能完整的存儲此系統連結 Trello看板所需的所有資訊

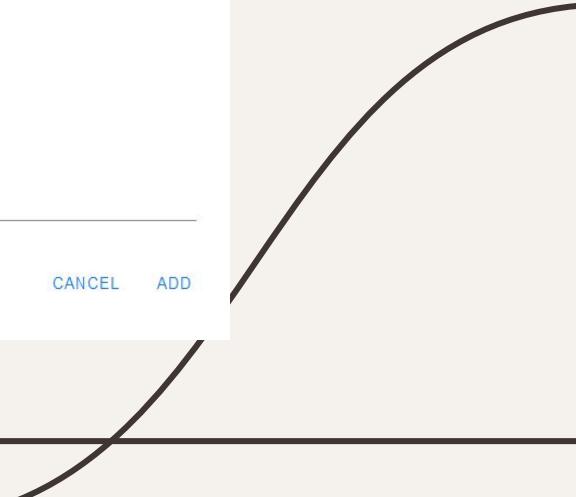
Bind Trello

Bind  
Status  


Trello Key

Trello Token

CLEAR EXISTED CANCEL ADD



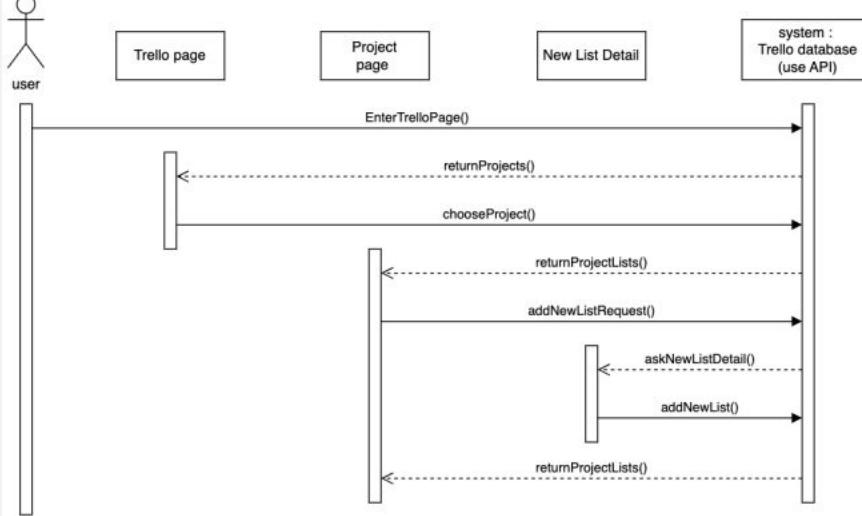
# TMS新增功能

身為一位使用者：

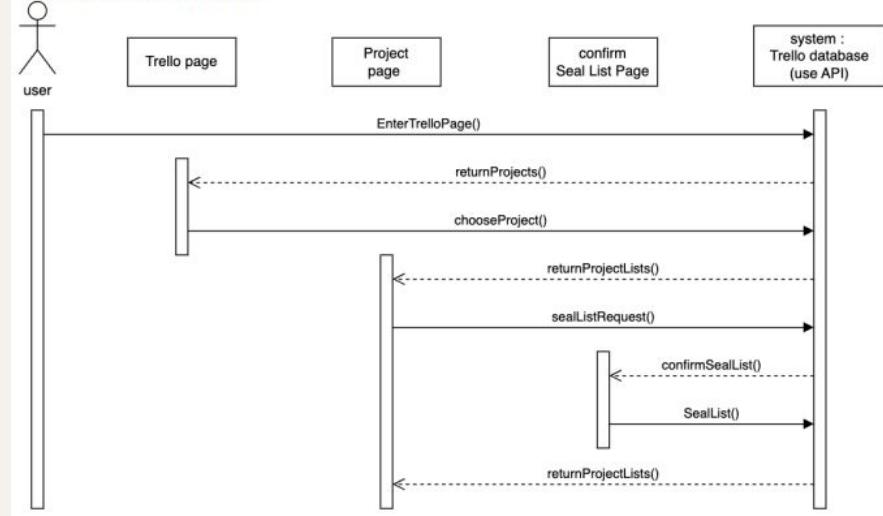
- 我希望能新增和封存列表，從而根據項目  
的發展動態調整任務組織。
- 驗收標準：使用者能夠在 Trello 上無障礙  
地新增新的列表，並將不再需要的列表封  
存。這些操作應該簡單直觀，並且能 夠即  
時反映在所有團隊成員的介面上。



新增list於Trello頁面中



封存Trello頁面中指定的list



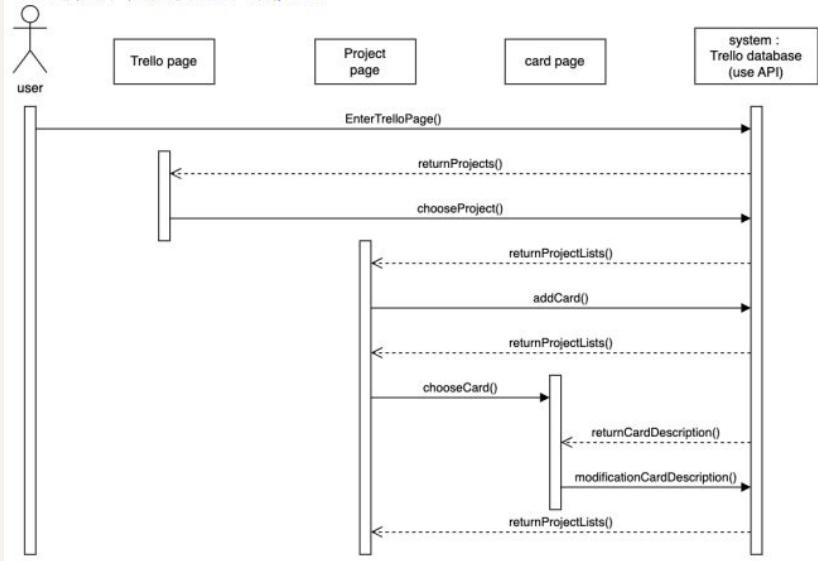


- 身為一位使用者，我需要新增卡片，以依據任務分配工作。
- 驗收標準：使用者能夠在 Trello 上無障礙地新增新的卡片。

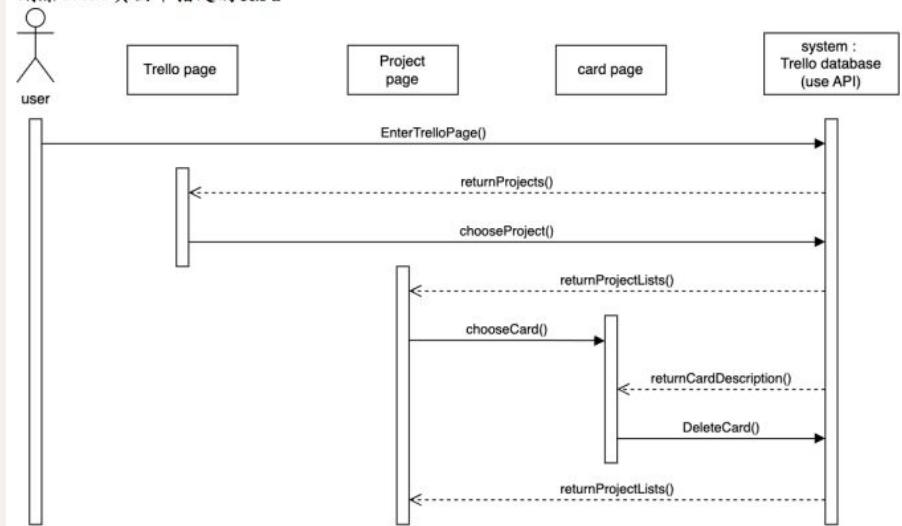


- 身為一位使用者，我需要刪除卡片，以刪除不需要的工作。
- 驗收標準：使用者能夠在 Trello 上無障礙地，並將不再需要的卡片刪除。

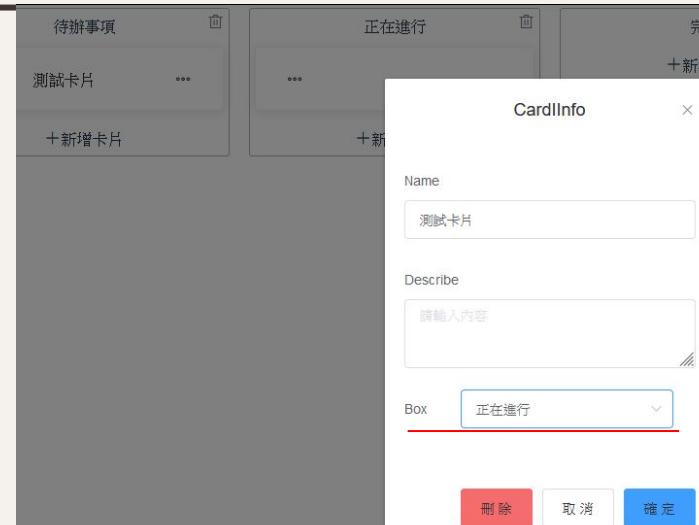
Trello頁面中在指定list中新增card



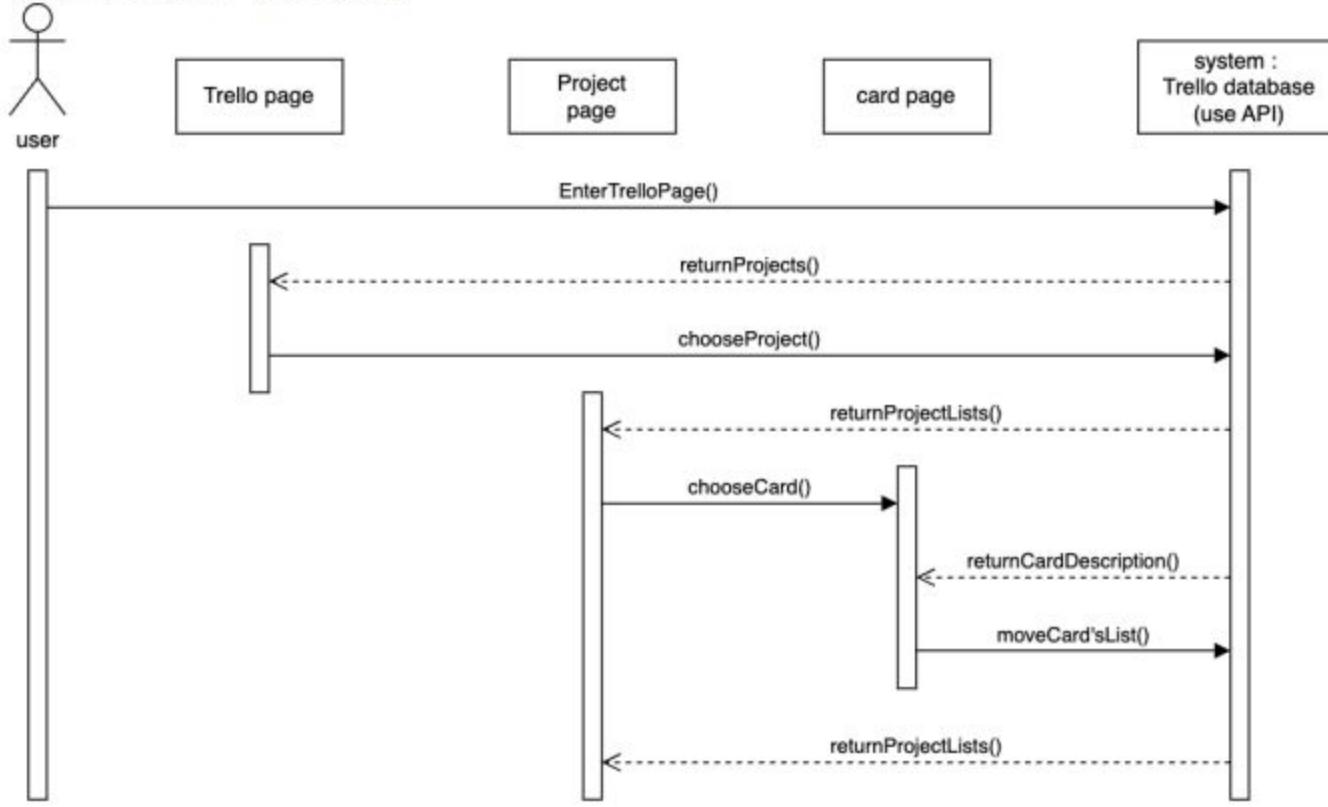
刪除Trello頁面中指定的card



- 身為一位使用者，我需要移動和管理卡片，以保持任務進度和優先級的靈活調整，從而提高團隊協作的效率。
- 驗收標準：使用者能夠輕鬆地在列表間移動卡片，以反映任務進度的變更。卡片的移動應該是直觀的，並且變更應即時顯示給所有相關團隊成員。



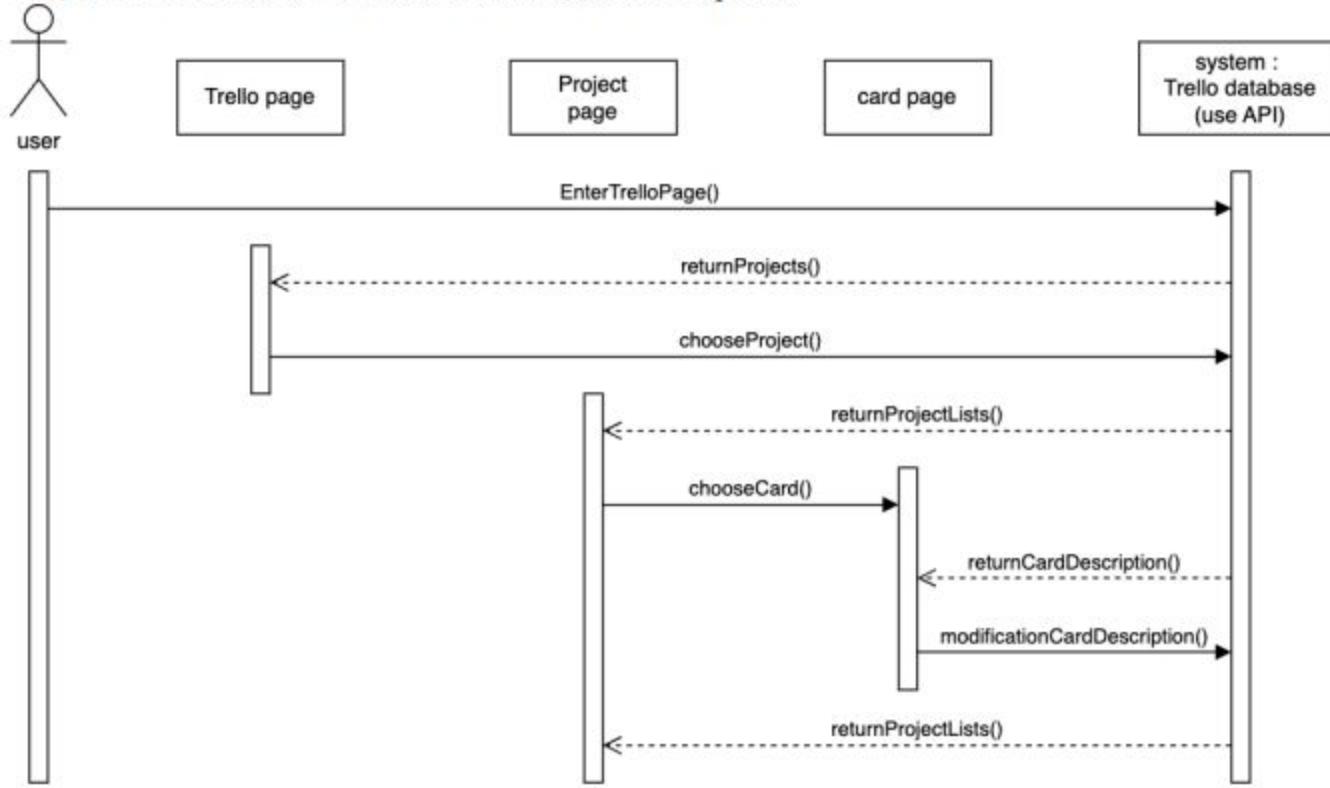
## 移動Trello頁面中指定的card



- 身為一位使用者，我需要查看描述，以確保任務資訊。
  - 驗收標準：使用者能夠輕鬆查看任何卡片的詳細描述。
- 
- 身為一位使用者，我需要修改卡片描述，以保持任務資訊的即時更新和準確性。
  - 驗收標準：使用者能夠修改卡片描述的功能應直觀易用，並且變更後的內容應即時更新並對所有團隊成員可見。

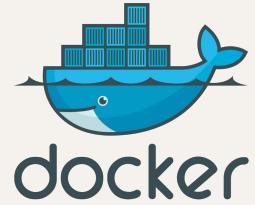


查看Trello Board頁面中指定card的詳細資訊(description)



# 環境建置方法

身為管理者，我想要當團隊成員交接時，不會有部署失敗的狀況。



- 驗收標準：使用者能獨立且順利的按照流程的建置好環境，並成功執行整個系統。

README.md

## ProjectManageSystemBackend

### Project Build

安裝visual studio 2022插件要裝.net 跟 asp.net

Visual Studio Installer

修改中 — Visual Studio Community 2017 — 15.9.58

工作負載 個別元件 語言套件 安裝位置

Windows (3)

.NET 桌面開發 使用 C#、Visual Basic 及 F# 建置 WPF、Windows Forms 與 主控台應用程式。

使用 C++ 的桌面開發 使用 Microsoft C++ 工具組、ATL 或 MFC 建置 Windows 傳統型應用程式。

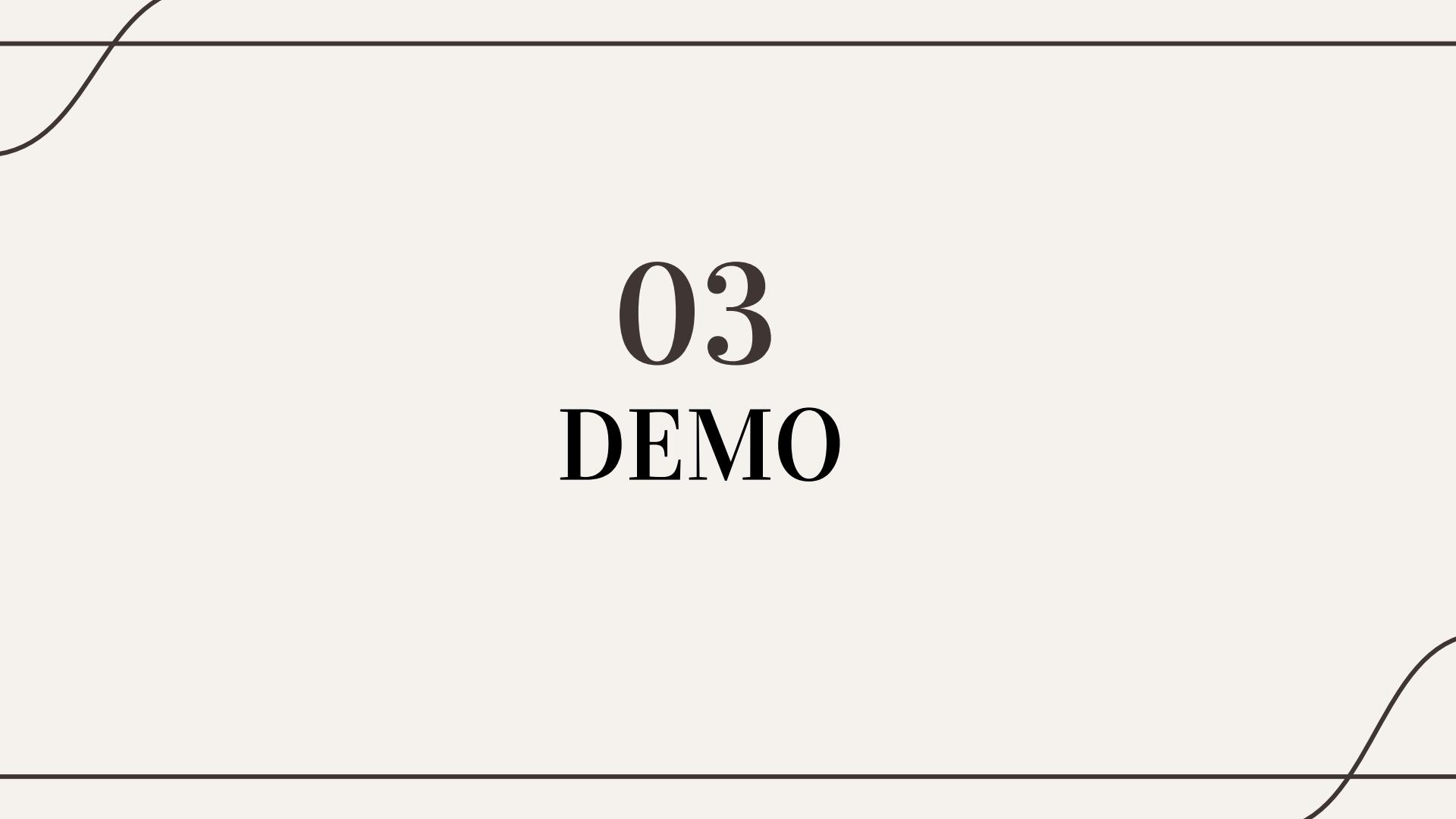
安裝詳細資料

.NET Framework 4 –

網頁程式開發的雲端

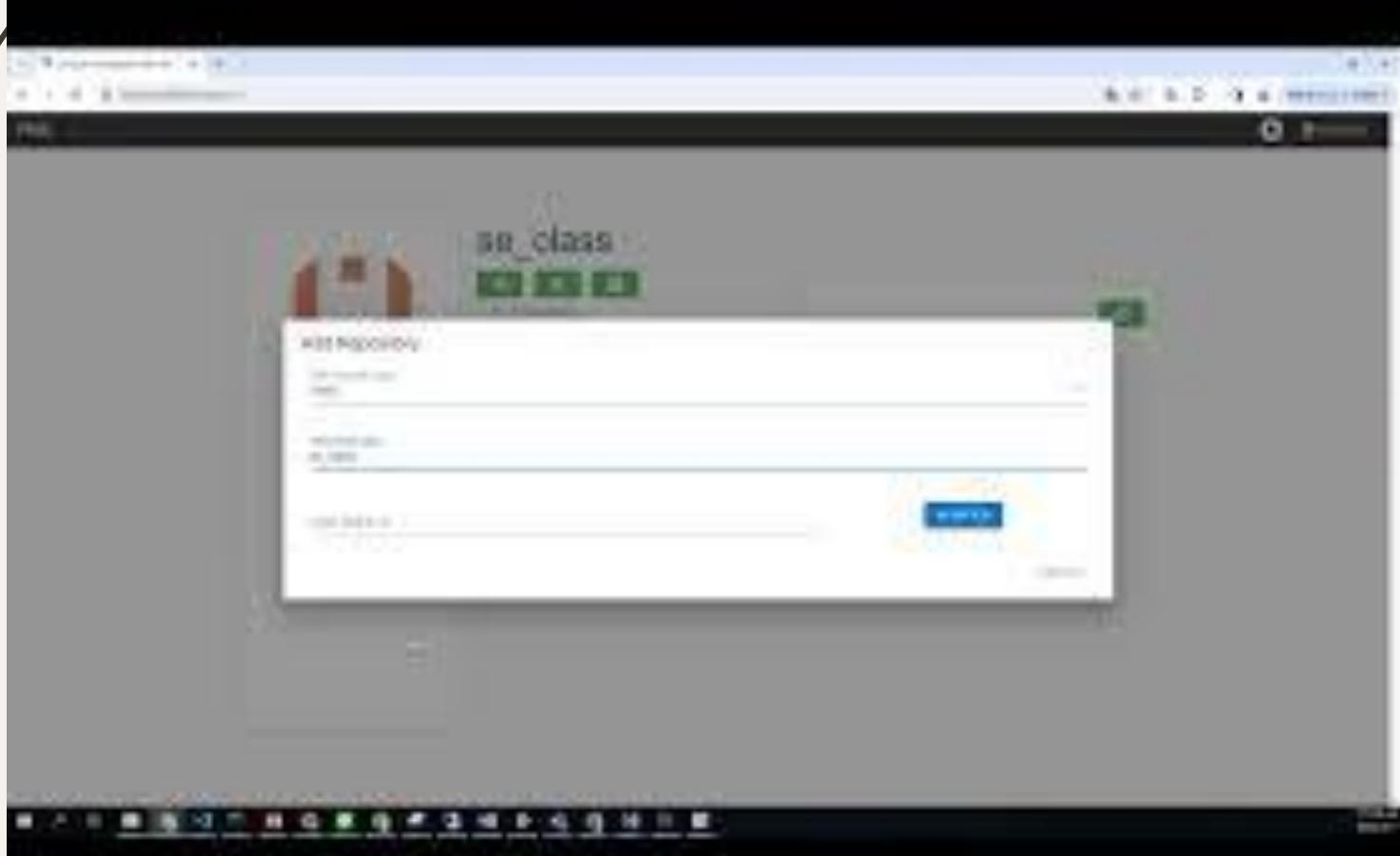
.NET 分析工具

Entity Framework 6 –



# 03

# DEMO



余

# 04

# Test Case

手動測試 7組  
unit test 21組

# 手動測試

Test Case Name		查看 Trello Board 頁面		Test Case ID	TMS-TC01	
Test Case Description		使用者查看自己專案中使用的 Trello Board		Serverity	1	
Pre-Requisite		登入系統並進入專案列表頁面		Post-Requisite	NA	
Test Execution Steps:						
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:23 ]: successful
2	點擊 Bind Trello 按鈕 新增 Trello Key 和 Trello Token	NA	顯示是否成功接上 Trello	顯示是否成功接上 Trello	Pass	[12/24/2023 15:26 ]: successful
3	點擊新增 repository 按鈕	NA	顯示是否成功接上選擇要連接的 Trello Board	顯示是否成功接上選擇要連接的 Trello Board	Pass	[12/24/2023 15:27 ]: successful
4	點擊新增的 repository	NA	顯示選擇接上 Trello Board	顯示選擇接上 Trello Board	Pass	[12/24/2023 15:28 ]: successful

<b>Test Case Name</b>	查看 Trello Card 頁面	<b>Test Case ID</b>	TMS-TC02																												
<b>Test Case Description</b>	使用者查看自己專案中使用的 Trello Card 裡的詳細內容(description)	<b>Serverity</b>	1																												
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																												
Test Execution Steps:																															
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:30 ]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:31 ]; successful</td> </tr> <tr> <td>3</td> <td>點擊所選的 card</td> <td>NA</td> <td>顯示 card 的詳細資訊</td> <td>顯示 card 的詳細資訊</td> <td>pass</td> <td>[12/24/2023 15:32 ]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:30 ]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:31 ]; successful	3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:32 ]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																									
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:30 ]; successful																									
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:31 ]; successful																									
3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:32 ]; successful																									

<b>Test Case Name</b>	新增 Trello List	<b>Test Case ID</b>	TMS-TC06																												
<b>Test Case Description</b>	使用者新增 Trello List	<b>Serverity</b>	1																												
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																												
Test Execution Steps:																															
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:46]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:47]; successful</td> </tr> <tr> <td>3</td> <td>點擊新增欄位按鈕，並輸入 List 名稱</td> <td>NA</td> <td>立即顯示新增的 List</td> <td>立即顯示新增的 List</td> <td>pass</td> <td>[12/24/2023 15:48]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:46]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:47]; successful	3	點擊新增欄位按鈕，並輸入 List 名稱	NA	立即顯示新增的 List	立即顯示新增的 List	pass	[12/24/2023 15:48]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																									
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:46]; successful																									
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:47]; successful																									
3	點擊新增欄位按鈕，並輸入 List 名稱	NA	立即顯示新增的 List	立即顯示新增的 List	pass	[12/24/2023 15:48]; successful																									

<b>Test Case Name</b>	新增 Trello Card	<b>Test Case ID</b>	TMS-TC03																												
<b>Test Case Description</b>	使用者新增 Trello Card	<b>Serverity</b>	1																												
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																												
Test Execution Steps:																															
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:34]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:35]; successful</td> </tr> <tr> <td>3</td> <td>點擊新增卡片按鈕，輸入 card 名稱</td> <td>NA</td> <td>立即在列表中顯示新添加的 card</td> <td>立即在列表中顯示新添加的 card</td> <td>pass</td> <td>[12/24/2023 15:36]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:34]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:35]; successful	3	點擊新增卡片按鈕，輸入 card 名稱	NA	立即在列表中顯示新添加的 card	立即在列表中顯示新添加的 card	pass	[12/24/2023 15:36]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																									
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:34]; successful																									
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:35]; successful																									
3	點擊新增卡片按鈕，輸入 card 名稱	NA	立即在列表中顯示新添加的 card	立即在列表中顯示新添加的 card	pass	[12/24/2023 15:36]; successful																									

<b>Test Case Name</b>	刪除 Trello Card	<b>Test Case ID</b>	TMS-TC04																																			
<b>Test Case Description</b>	使用者刪除所選擇的 Trello Card	<b>Serverity</b>	1																																			
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																																			
Test Execution Steps:																																						
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:37]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:38]; successful</td> </tr> <tr> <td>3</td> <td>點擊所選的 card</td> <td>NA</td> <td>顯示 card 的詳細資訊</td> <td>顯示 card 的詳細資訊</td> <td>pass</td> <td>[12/24/2023 15:39]; successful</td> </tr> <tr> <td>4</td> <td>點擊 card 中刪除按鈕</td> <td>NA</td> <td>立即顯示刪除指定 card 後的結果</td> <td>立即顯示刪除指定 card 後的結果</td> <td>pass</td> <td>[12/24/2023 15:40]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:37]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:38]; successful	3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:39]; successful	4	點擊 card 中刪除按鈕	NA	立即顯示刪除指定 card 後的結果	立即顯示刪除指定 card 後的結果	pass	[12/24/2023 15:40]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																																
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:37]; successful																																
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:38]; successful																																
3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:39]; successful																																
4	點擊 card 中刪除按鈕	NA	立即顯示刪除指定 card 後的結果	立即顯示刪除指定 card 後的結果	pass	[12/24/2023 15:40]; successful																																

<b>Test Case Name</b>	移動 Trello Card 所在 List	<b>Test Case ID</b>	TMS-TC05																																			
<b>Test Case Description</b>	使用者移動 Trello Card 至期望的 List 裡	<b>Serverity</b>	1																																			
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																																			
Test Execution Steps:																																						
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:42]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:43]; successful</td> </tr> <tr> <td>3</td> <td>點擊所選的 card</td> <td>NA</td> <td>顯示 card 的詳細資訊</td> <td>顯示 card 的詳細資訊</td> <td>pass</td> <td>[12/24/2023 15:44]; successful</td> </tr> <tr> <td>4</td> <td>點擊 Box 的下拉欄位，選擇要移動到的 List</td> <td>NA</td> <td>立即顯示移動 Card 遷動過後的結果</td> <td>立即顯示移動 Card 遷動過後的結果</td> <td>pass</td> <td>[12/24/2023 15:45]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:42]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:43]; successful	3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:44]; successful	4	點擊 Box 的下拉欄位，選擇要移動到的 List	NA	立即顯示移動 Card 遷動過後的結果	立即顯示移動 Card 遷動過後的結果	pass	[12/24/2023 15:45]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																																
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:42]; successful																																
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:43]; successful																																
3	點擊所選的 card	NA	顯示 card 的詳細資訊	顯示 card 的詳細資訊	pass	[12/24/2023 15:44]; successful																																
4	點擊 Box 的下拉欄位，選擇要移動到的 List	NA	立即顯示移動 Card 遷動過後的結果	立即顯示移動 Card 遷動過後的結果	pass	[12/24/2023 15:45]; successful																																

<b>Test Case Name</b>	封存 Trello List	<b>Test Case ID</b>	TMS-TC07																																			
<b>Test Case Description</b>	使用者封存所選擇的 Trello List	<b>Serverity</b>	1																																			
<b>Pre-Requisite</b>	登入系統並進入專案列表頁面	<b>Post-Requisite</b>	NA																																			
Test Execution Steps:																																						
<table border="1"> <thead> <tr> <th>S.No</th> <th>Action</th> <th>Inputs</th> <th>Expected Output</th> <th>Actual Output</th> <th>Test Result</th> <th>Test Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>點擊專案名稱進入 repo 列表</td> <td>NA</td> <td>顯示專案中的 repo</td> <td>顯示專案中的 repo</td> <td>Pass</td> <td>[12/24/2023 15:49]; successful</td> </tr> <tr> <td>2</td> <td>點擊所選的 repository</td> <td>NA</td> <td>顯示 Trello Board</td> <td>顯示 Trello Board</td> <td>pass</td> <td>[12/24/2023 15:50]; successful</td> </tr> <tr> <td>3</td> <td>點擊 List 左上角的垃圾桶 icon</td> <td>NA</td> <td>顯示詢問是否封存 List 的提示框</td> <td>顯示詢問是否封存 List 的提示框</td> <td>pass</td> <td>[12/24/2023 15:51]; successful</td> </tr> <tr> <td>4</td> <td>點擊提示框中的確認按鈕</td> <td>NA</td> <td>立即顯示刪除指定 List 的結果，同時跳出提示訊息</td> <td>立即顯示刪除指定 List 的結果，同時跳出提示訊息</td> <td>pass</td> <td>[12/24/2023 15:52]; successful</td> </tr> </tbody> </table>				S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments	1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:49]; successful	2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:50]; successful	3	點擊 List 左上角的垃圾桶 icon	NA	顯示詢問是否封存 List 的提示框	顯示詢問是否封存 List 的提示框	pass	[12/24/2023 15:51]; successful	4	點擊提示框中的確認按鈕	NA	立即顯示刪除指定 List 的結果，同時跳出提示訊息	立即顯示刪除指定 List 的結果，同時跳出提示訊息	pass	[12/24/2023 15:52]; successful
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments																																
1	點擊專案名稱進入 repo 列表	NA	顯示專案中的 repo	顯示專案中的 repo	Pass	[12/24/2023 15:49]; successful																																
2	點擊所選的 repository	NA	顯示 Trello Board	顯示 Trello Board	pass	[12/24/2023 15:50]; successful																																
3	點擊 List 左上角的垃圾桶 icon	NA	顯示詢問是否封存 List 的提示框	顯示詢問是否封存 List 的提示框	pass	[12/24/2023 15:51]; successful																																
4	點擊提示框中的確認按鈕	NA	立即顯示刪除指定 List 的結果，同時跳出提示訊息	立即顯示刪除指定 List 的結果，同時跳出提示訊息	pass	[12/24/2023 15:52]; successful																																

# Unit Test

```
// 實作檢查boardId跟userId綁定的trello key和token是否符合
public async Task<bool> CheckTrelloRepo(TrelloLoginDto dto, string userId)
{
    var user = _dbContext.Users.Include(u => u.Projects).ThenInclude(p => p.Project).FirstOrDefault(u => u.Account.Equals(userId));
    if (user == null)
    {
        Console.WriteLine("[CheckTrelloRepo] Can't Find This User");
        throw new Exception("Can't Find This User");
    }
    string boardId = await GetBoardIdByUrl(dto.BoardUrl, user.TrelloKey, user.TrelloToken);
    if (boardId == "None")
    {
        return false;
    }
}

// CheckTrelloRepo
[Fact]
● |0 個參考
public async void TestCheckTrelloRepoSuccess()
{
    string user_id = "github_shark";
    bool result = await _trelloService.CheckTrelloRepo(user_id);
    Assert.True(result);
}

[Fact]
● |0 個參考
public async Task TestCheckTrelloRepoNoUser()
{
    string user_id = "55688"; // 替換為不存在的userId

    var exception = await Assert.ThrowsAsync<Exception>(() => _trelloService.CheckTrelloRepo(user_id));
    Assert.Equal("Can't Find This User", exception.Message);
}

[Fact]
● |0 個參考
public async Task TestCheckTrelloRepoInvalidBoardUrl()
{
    string user_id = "github_shark";
    var exception = await Assert.ThrowsAsync<Exception>(() => _trelloService.CheckTrelloRepo(dtoForFail, user_id));
    Assert.Equal("Invalid BoardUrl", exception.Message);
}
```

The screenshot shows the Visual Studio IDE interface with several windows open:

- Code Editor:** Displays the implementation of the `CheckTrelloRepo` method.
- Test Explorer:** Shows the test results for the `TrelloServiceTest` class. It lists 21 tests, all of which have passed (indicated by green checkmarks). The results table at the bottom shows the execution times for three specific tests:
  - `TestCheckTrelloRepoInvalidBoardUrl`: 788毫秒
  - `TestCheckTrelloRepoNoUser`: 8毫秒
  - `TestCheckTrelloRepoSuccess`: 1.9秒
- Output Window:** Shows the command `dotnet test` being run.
- Task List:** Shows the tasks generated by the test runner.
- Properties Window:** Shows the properties for the `TrelloLoginDto` class, with the `BoardUrl` field highlighted.

# Unit Test

```
// 實作檢查boardId跟user綁定的trello key和token符合
public async Task<bool> CheckTrelloRepo(TrelloLoginDto dto, string userId)
{
    var user = _dbContext.Users.Include(u => u.Projects).ThenInclude(p => p.Project).FirstOrDefault(u => u.Account.Equals(userId));
    if (user == null)
    {
        Console.WriteLine("[CheckTrelloRepo] Can't Find This User");
        throw new Exception("Can't Find This User");
    }
    string boardId = await GetBoardIdByUrl(dto.BoardUrl, user.TrelloKey, user.TrelloToken);
    if (boardId == "None")
    {
        Console.WriteLine("[CheckTrelloRepo] Invalid BoardUrl");
        throw new Exception("Invalid BoardUrl");
    }
    bool result = await CheckTrelloReposExist(boardId, user.TrelloKey, user.TrelloToken);
    if (result)
    {
        Console.WriteLine("[CheckTrelloRepo] Trello Repo Info Correct");
        return true;
    }
}

// CheckTrelloRepo
[Fact]
● |0 個參考
public async void TestCheckTrelloRepoSuccess()
{
    string user_id = "github_shark";
    bool result = await _trelloService.CheckTrelloRepo(trelloLoginDto, user_id);
    Assert.True(result);
}

[Fact]
● |0 個參考
public async Task TestCheckTrelloRepoNoUser()
{
    string user_id = "55688";  

    var exception = await Assert.ThrowsAsync<Exception>(() => _trelloService.CheckTrelloRepo(trelloLoginDto, user_id));
    Assert.Equal("Can't Find This User", exception.Message);
}

[Fact]
● |0 個參考
public async Task TestCheckTrelloRepoInvalidBoardUrl()
{
    string user_id = "github_shark";
    var exception = await Assert.ThrowsAsync<Exception>(() => _trelloService.CheckTrelloRepo(dtoForFail, user_id));
    Assert.Equal("Invalid BoardUrl", exception.Message);
}
```

```
TrelloLoginDto trelloLoginDto = new TrelloLoginDto
{
    Name = "My Trello",
    BoardUrl = "https://trello.com/b/2G4Mm0Fj",
    ProjectId = 1
};

TrelloLoginDto dtoForFail = new TrelloLoginDto
{
    Name = "My Trello",
    BoardUrl = "https://trello.com/fail"  
    ProjectId = 1
};
```

✓ TestCheckTrelloRepoInvalidBoardUrl	788 毫秒
✓ TestCheckTrelloRepoNoUser	8 毫秒
✓ TestCheckTrelloRepoSuccess	1.9 秒

# Unit Test - code coverage

Name	Covered	Uncovered	Coverable	Total	Line coverage	Covered	Total	Branch coverage	
+ PMS_test	1431	213	1644	2947	87%	18	18	100%	
- project-manage-system-backend	1228	1139	2367	4873	51.8%	146	308	47.4%	

project_manage_system_backend.Services.AuthorizeService	73	2	75	125	97.3%		10	12	83.3%	
project_manage_system_backend.Services.BaseService	9	0	9	22	100%		0	0		
project_manage_system_backend.Services.InvitationService	42	2	44	74	95.4%		2	4	50%	
project_manage_system_backend.Services.JiraService	52	98	150	184	34.6%		12	30	40%	
project_manage_system_backend.Services.ProjectService	86	15	101	164	85.1%		24	30	80%	
project_manage_system_backend.Services.RepoInfoService	30	2	32	61	93.7%		2	4	50%	
project_manage_system_backend.Services.RepoService	83	10	93	145	89.2%		11	20	55%	
project_manage_system_backend.Services.SonarqubeService	59	0	59	92	100%		7	8	87.5%	
project_manage_system_backend.Services.TrelloService	0	213	213	375	0%		0	28	0%	
project_manage_system_backend.Services.UserService	76	21	97	151	78.3%		14	22	63.6%	

# Unit Test - code coverage

Name	Covered	Uncovered	Coverable	Total	Line coverage	Covered	Total	Branch coverage
+ PMS_test	1676	21	1697	2873	98.7%	18	18	100%
- project-manage-system-backend	1415	952	2367	4874	59.7%	165	308	53.5%
project_manage_system_backend.Services.AuthorizeService	73	2	75	125	97.3%	10	12	83.3%
project_manage_system_backend.Services.BaseService	9	0	9	23	100%	0	0	
project_manage_system_backend.Services.InvitationService	42	2	44	74	95.4%	2	4	50%
project_manage_system_backend.Services.JiraService	52	98	150	184	34.6%	12	30	40%
project_manage_system_backend.Services.ProjectService	86	15	101	164	85.1%	24	30	80%
project_manage_system_backend.Services.RepoInfoService	30	2	32	61	93.7%	2	4	50%
project_manage_system_backend.Services.RepoService	83	10	93	145	89.2%	11	20	55%
project_manage_system_backend.Services.SonarqubeService	59	0	59	92	100%	7	8	87.5%
project_manage_system_backend.Services.TrelloService	107	106	213	375	50.2%	9	28	32.1%
project_manage_system_backend.Services.UserService	76	21	97	151	78.3%	14	22	63.6%
project_manage_system_backend.Shares.DateHandler	16	1	17	29	94.1%	7	8	87.5%
project_manage_system_backend.Shares.JwtHelper	27	0	27	61	100%	0	0	
project_manage_system_backend.Shares.PMSContext	17	0	17	31	100%	0	0	
project_manage_system_backend.Startup	109	13	122	167	89.3%	5	10	50%

05

# 會議紀錄與反省

# Trello

The Trello board is organized into several lists:

- Check Points**: Contains a card for "11/14(二)期中報告" due "11月14日".
- To-Do**: Contains cards for "專案目標" and "user story".
- System Architecture Diagram**: A diagram showing a Frontend layer connected to a PVS layer, which in turn connects to a Database layer.
- WBS (Work Breakdown Structure)**: A detailed hierarchical diagram of project tasks.
- To-Do**: Contains cards for "準備口頭報告" and "找美麗的麗程模版".
- Work in progress**: Contains a card for "Docker的User story".
- Done**: Contains cards for "Docker的User story" and "找美麗的麗程模版".
- 會議記錄**: A list of completed tasks from 09/12 to 11/13, including "分組完畢", "認證組員", "決定繼承專案", "安裝專案", "PEP撰寫", "SRS撰寫", "環境建置", "討論擴充功能的實作", "期中報告", and "把期中報告做最後的討論與整理".

On the right side of the board, there is a sidebar with various Power-up icons and a "Share" button.

# 會議記錄

**09/12(一)分組完畢**

in list 會議記錄

Notifications  Watch

**10/02(一)安裝專案**

in list 會議記錄

Members Notifications  Watch

**Description**

組長：  
112598028 余彌綺  
組員：  
112598004 周雨柔  
112598021 張書渝  
112C72010 陸紀霖  
112C71003 吳思正  
112C71015 李維

• visual 2017  
• nodejs 16  
• npm 8  
• windows:安裝這個  
• 回編譯器：  
nvm install 16  
nvm use 16  
然後從這邊開始  
npm run serve

**11/13(一)晚上 報告預演**

in list 會議記錄

Members Notifications  Watch

**Description**

Add a more detailed description...

**Activity**

北科大-周雨柔 marked the due date complete Nov 14, 2023 at 12:23 AM

陸紀霖 changed the due date of this card to Nov 13, 2023 at 11:00 PM Nov 13, 2023 at 10:29 PM

陸紀霖 set this card to be due Nov 14, 2023 at 11:00 PM Nov 13, 2023 at 4:19 PM

陸紀霖 added 李維 to this card Nov 13, 2023 at 4:19 PM

**10/09(一)PEP撰寫**

in list 會議記錄

Members Notifications  Watching

**Description**

1. 決定新增的功能  
a. 團隊所有人的每日活躍度，依照commit次數（類似github個人首頁）用顏色區分每個人  
b. 整體專案完成度，需先分配每人負責的專案%數，再依每人commit後自行填寫完成度，來計算整體專案的完成度  
2. PEP文件分工  
a. section 2: 雨柔  
b. section 3: 書渝  
c. section 4, 5: 李維  
d. section 6: 紀霖  
e. section 7, 8: 思正  
f. 總整: 紹穎

**10/16(一)SRS撰寫**

in list 會議記錄

Members Notifications  Watching

**Description**

1. 共同撰寫文件  
2. 新增TMS系統

**10/23(一)環境建置**

in list 會議記錄

Notifications  Watch

**Description**

1. 互相於參與討論環境建置問題  
2. 統盤問題在環境建置手冊

**12/04(一)討論會議**

in list 會議記錄

Members Notifications  Watching

**Description**

1. 本地部署  
a. docker：了解後，可以當做外殼應用  
b. vue：建議選擇vue版式（ 工程工具）→ 時刻健壯影片，發佈並由監督人管理，無法達到成功API回饋  
c. Net：了解後，可以開始實驗部署，發現無法連接成功的HelloWorld→待解  
2. 資料架構  
a. 文本說明(DO) - 實作說明及說明  
b. 文本說明(DO) - use case model (需求) - static models 資料類別 - Dynamic Models (需求) - Interface Design - Data Base Design  
3. 本子系統  
a. 文本說明(DO) - use case model (需求) - static models 資料類別 - Dynamic Models (需求) - Interface Design (需求 + 定義)  
4. 進度管理  
a. 敏捷管理

**12/11(一) 討論會議**

in list 會議記錄

Notifications  Watch

**Description**

1. 分組報告  
a. docker：還在學習中，  
b. vue：建議選擇vue版式（ 工程工具）→ 時刻健壯影片，發佈並由監督人管理，無法達到成功API回饋  
c. Net：了解後，可以開始實驗部署，發現無法連接成功的HelloWorld→待解  
2. 資料架構  
a. 文本說明(DO) - 實作說明及說明  
b. 文本說明(DO) - use case model (需求) - static models 資料類別 - Dynamic Models (需求) - Interface Design - Data Base Design  
3. 進度管理  
a. 敏捷管理

**12/21(一)討論會議**

in list 會議記錄

Due date  Nov 20, 2023 at 3:00 PM Complete

Members Notifications  Watching

**Description**

1. 本地部署  
a. docker：建議選擇vue版式（ 工程工具）→ 時刻健壯影片，發佈並由監督人管理，無法達到成功API回饋  
b. vue：建議選擇vue版式（ 工程工具）→ 時刻健壯影片，發佈並由監督人管理，無法達到成功API回饋  
c. Net：了解後，可以開始實驗部署，發現無法連接成功的HelloWorld→待解  
2. 資料架構  
a. 文本說明(DO) - use case model (需求) - static models 資料類別 - Dynamic Models (需求) - Interface Design - Data Base Design  
3. 進度管理  
a. 敏捷管理

**12/18(一)討論會議**

in list 會議記錄

Notifications  Watch

**Description**

1. 分組報告  
a. docker-docker的ASP環境建置完成，Vue目前有點小問題，目前完成度約一半  
b. vue-完成使用點卡片式移動卡片  
c. CASP-目前正在研究移動卡片的API  
2. 進度報告  
3. 進度管理

**12/25(一)討論會議**

in list 會議記錄

Notifications  Watch

**Description**

1. 分組報告  
a. docker-還  
b. vue-建議選擇vue版式（ 工程工具）→ 時刻健壯影片，發佈並由監督人管理，無法達到成功API回饋  
c. Net-建議選擇NetCore API  
2. 進度報告  
3. 進度管理

**Activity**

Write a comment...

北科大-周雨柔 added this card to 會議記錄 Nov 13, 2023 at 11:19 PM

李維 added this card to 會議記錄 Dec 11, 2023 at 3:09 PM

余彌綺 added this card to 會議記錄 Dec 15, 2023 at 10:01 PM

# Retrospective

11/20反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

沒有做好的部分

- 之前都沒有確實執行scrum，尤其是retrospective的部分
- 沒有確實完成每次scrum應該要有的進度

學到的部分

- 每個scrum都是由一到多個user story 發起的
- user story不用一定要差實質文件上的功能

改進的部分

- 確實地去執行scrum應該要有的所有配套措施
- 增加一些deadline的強制性

無法更改的部分

- 每日的stand up meeting因為時間因素還是沒辦法執行

Activity

Hide Details

12/11反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 習慣使用user story，以此作為本週進度方向。
- 做好的同學會有紀錄或影片協助其他未跟上的同學。

沒有做好的部分

- 工作分工不明確
- jira算user story時，沒有把驗收標溝寫上去

學到的部分

- 了解jira的Scrum跟Trello的使用方式功能。
- 發現反省內容標題理解錯誤，做出修正。

需要改進的部分

- 增加更即時溝通，確實使用gitlab。

11/27反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 成功用jira去管理專案
- 成功去幫助問題

沒有做好的部分

- 沒有全面地去瞭解各自分配到的工作（vue, docker等）
- （vue框架不會）
- 沒有聽好在discord上進行平均的進度監督與合作（可考慮推奐）
- 應該要適時用trello進行專案管理

學到的部分

- 訂定有公車的開會流程
- 要詳細的scrum相關名詞（微畫面）

改進的部分

- 開會效率提升
- 主動分享學習到的技能，讓大家獲得更廣闊的軟體工程技巧（將jira中提升專案管理的連結）

無法更改的部分

- 無

12/04反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 成功去幫助數問題，達成目標

沒有做好的部分

- 進度起不來
- 對接手的專案認知不夠全面
- Jira的Scrum跟Trello的會被功能沒有統合，沒有起到協同作用的功能

學到的部分

- 大家對自己要做的事和自己的能力都有一定的瞭解了

改進的部分

- 開會方式上動機

無法更改的部分

- 大家依舊忙

12/18反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 有更新gitlab
- 新增list完成
- user story的切割做的不錯
- jira的合作模式受到肯定

沒有做好的部分

- 估算工作量的程度，需要改善
- 溝通量還是有一點點不足

學到的部分

- 了解別人的userstory如何使用，才能在jira顯示出更好的結果。

需要改進的部分

- 量力而為領取task

12/25反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 了解如何準備估點數
- 詳細即時溝通 完成Increment 2文件

沒有做好的部分

- 沒有將領取的task完成

學到的部分

- 善用身邊資源

需要改進的部分

- 量力而為領取task

1/5反省

in list Sprint Retrospective

Notifications  
④ Watch

Description

做好的部分

- 了解如何準備估點數
- 詳細即時溝通 完成Increment 2文件

沒有做好的部分

- 沒有將領取的task完成

學到的部分

- 善用身邊資源

需要改進的部分

- 量力而為領取task

Activity

Hide Details

Write a comment...

北科大-周雨柔 added this card to Sprint Retrospective Jan 5 at 3:35 PM

# 會議記錄

# 反省

- 1.錄製會議過程  
(開始使用Teams)
- 2.成員加入DC以方便合作
- 3.將Jira加入實作
- 4.開始執行retrospective

11 / 20

- 1.應嚴格遵循Scrum流程
- 2.每次sprint  
圍繞至少一個user story



Jira Software 您的工作 專案 篩選 儀表板 團

專案 / 北科軟體工程 時間軸

Q 📈 8 8 8 S 8 8 ⚙️ 狀態類別 大型工作

衝刺	十一月 '23
SCRUM-6 環境建置	20 21 22 23 24 25 26
SCRUM-22 文件撰寫	SCRUM 衝刺 1
SCRUM-27 學習知識	27

11 / 27

1. 確立會議行程

- a. 站立會議：15分鐘  
反思：15分鐘  
下個sprint planning：15分鐘

- 1.確立會議行程
- 2.站立會議、反思、  
下個sprint planning

學到的部分

1. 訂定有效率的開會流程
2. 更詳細的scrum相關名詞 (燃盡圖)

改進的部分

1. 開會效率提升

# 會議記錄

## 反省

### 1. 開始實作估點數

12  
11

#### 做好的部分

1. 習慣使用user story，
2. 做好的同學會有紀錄或
3. 開始估點數

#### 沒有做好的部分

1. 工作分工不明確
2. jira開user story時，沒有把驗收標準寫上去



The screenshot shows a Trello card for a user story. The card includes the following details:

- 標題: 身為專案開發者，我希望建能在PMS系統下的TMS子系統中新增一個按鈕，能夠在trello面板上新增list。
- 狀態: 完成
- 子議題: SERUM-51 在PMS的TMS子系統中，有個按... (完成 100%)  
SERUM-52 按下按鈕後，可以輸入list名稱 (完成 100%)

旁白文字: 採用子議題作為驗收標準 (可以為各個子議題分別估點數)

12  
18 ————— 12  
25

#### 沒有做好的部分

1. 估點數時使用太多大評分，導致稍許的不同難度變化無法細分



活動

聊天

團隊

所有團隊

1

## 112軟工報告

來自 您 及 周雨柔 的 67 則回覆

← 回覆

2023年12月4日



12/11 已開始

來自 您、周雨柔、陸紀霖 及 其他 2 位 的 48 則回覆

← 回覆

2023年12月11日



12/18 已開始

來自 您、周雨柔、陸紀霖 及 其他 3 位 的 28 則回覆

← 回覆

2023年12月18日



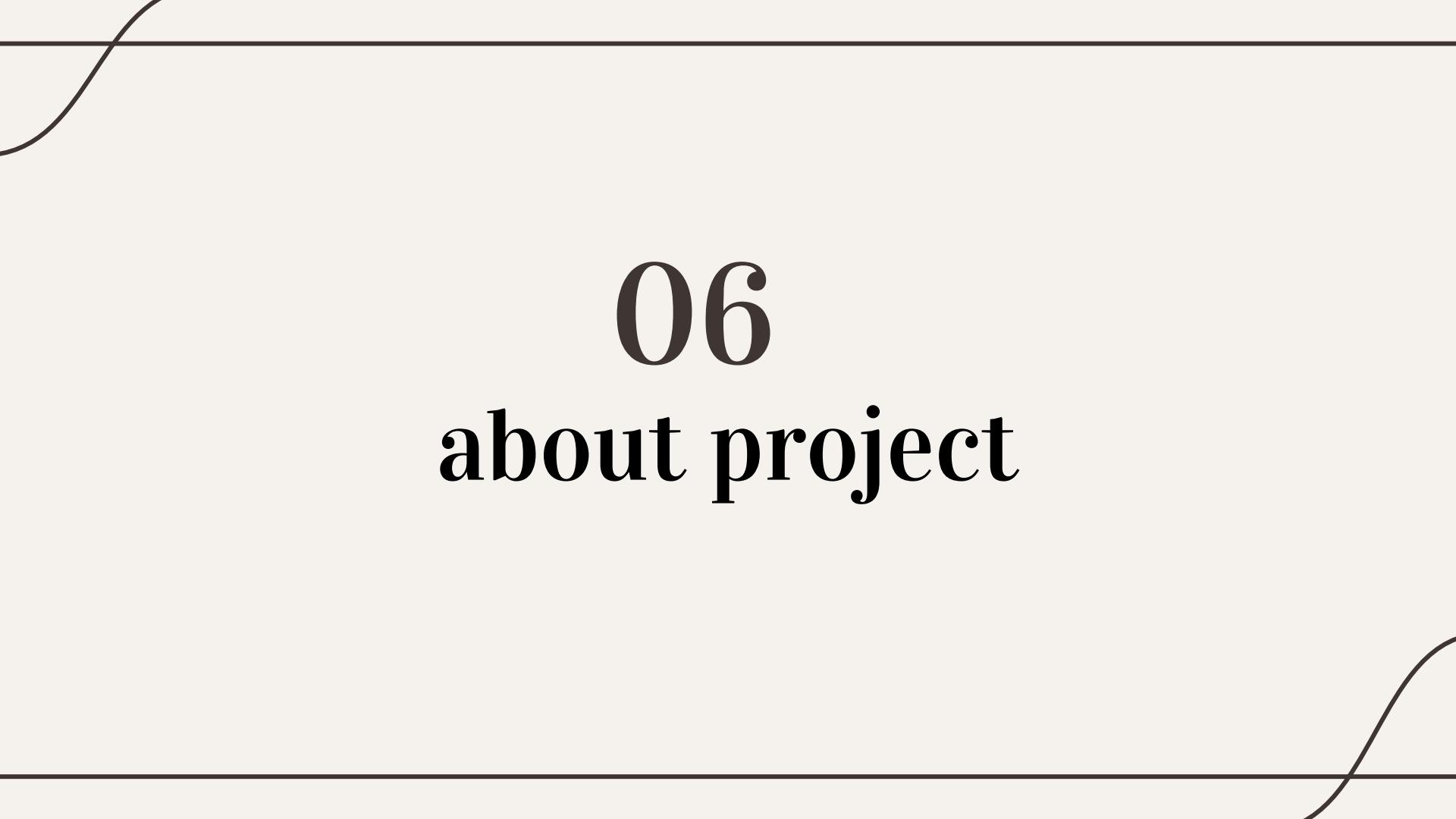
12/25 已開始

來自 您、周雨柔、陸紀霖 及 其他 3 位 的 26 則回覆

← 回覆

2023年12月25日

	名稱 ▾	修改時間 ▾	修改者 ▾
▶	「General」中的會議 -20231204_150625...	23年12月4日	周雨柔
▶	「General」中的會議-20231130_140717-...	23年11月30日	周雨柔
▶	01_05-20240105_152213-會議錄製.mp4	3 天前	周雨柔
▶	11_27-20231127_151608-會議錄製.mp4	23年11月27日	陸紀霖
▶	12_11-20231211_151247-會議錄製.mp4	23年12月11日	周雨柔
▶	12_18-20231218_153125-會議錄製.mp4	23年12月18日	周雨柔
▶	12_25-20231225_150420-會議錄製.mp4	23年12月25日	周雨柔
▶	demo invite user-20240108_024215-會...	11 小時前	余珮綺
▶	demo TMS-20240107_174444-會議錄製....	昨天 下午 05:47	余珮綺
▶	demo0104-20240104_203431-會議錄製....	4 天前	周雨柔
▶	demo-20240107_173532-會議錄製.mp4	昨天 下午 05:41	余珮綺
▶	新增頻道會議-20231120_153503-會議錄...	23年11月20日	張書瑜
▶	新增頻道會議-20231130_141820-會議錄...	23年11月30日	周雨柔



# 06

# about project

# Jira看板

追蹤工作項目  
各個Task可分屬不同  
Epic

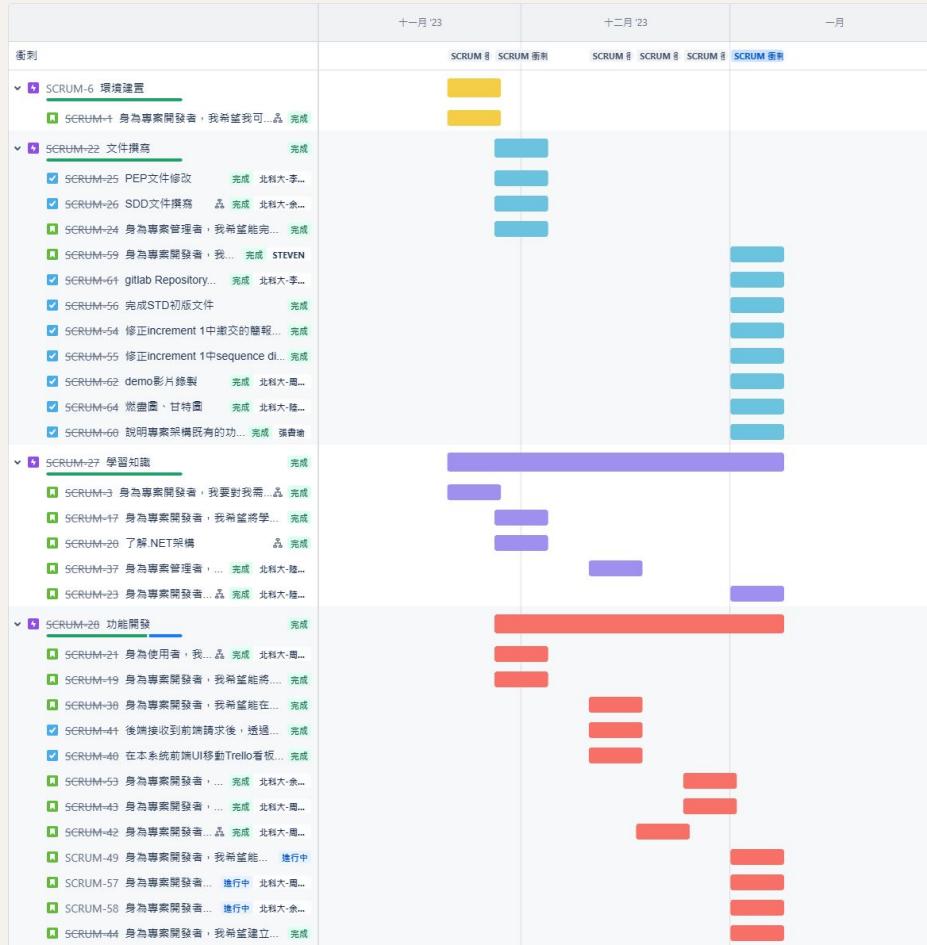
連結圖表分析  
輔助甘特圖、燃盡圖與  
儀表版分析

任務優先級  
故事點的分配資訊呈  
現

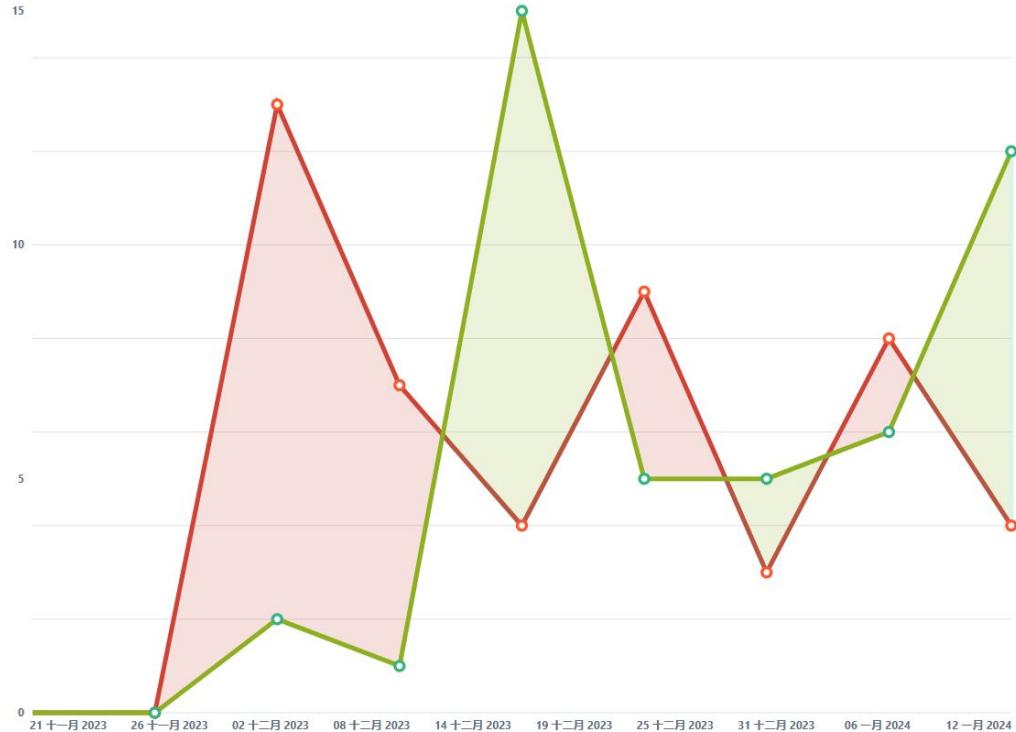
The screenshot shows a Jira Kanban board for 'SCRUM 衝刺 6'. The board is organized into three columns: 'TO DO', 'IN PROGRESS', and 'DONE'. Each column contains several cards, each with a title, status, and a checkmark icon.

- TO DO:**
  - 身為專案開發者，我希望能夠在TMS子系統上，看到Card活動紀錄。  
功能開發  
SCRUM-49
  - 身為專案開發者，我想要建立Unit test。  
功能開發  
SCRUM-57
  - 身為專案開發者，我想要算出code coverage  
功能開發  
SCRUM-58
- IN PROGRESS:**
  - 身為專案開發者，我想要統整RETROSPECTIVE、會議記錄和USER STORY，並使用jira和Trello截圖作為依據。  
文件匯總  
SCRUM-59
  - 身為專案開發者，我希望建立一個按鈕，能夠在TMS子系統上，刪除Card。  
功能開發  
SCRUM-44
  - gitlab Repository 的 README.md 中，本學期新增、完成、修改的功能（列表呈現），及fronted和backend中的環境建置readme  
文件匯總  
SCRUM-61
  - 燃盡圖、甘特圖  
文件匯總  
SCRUM-64
- DONE:**
  - 完成STD初版文件  
文件匯總  
SCRUM-56

# Gantt



## 已建立與已解決的比較圖表: 北科軟體工程



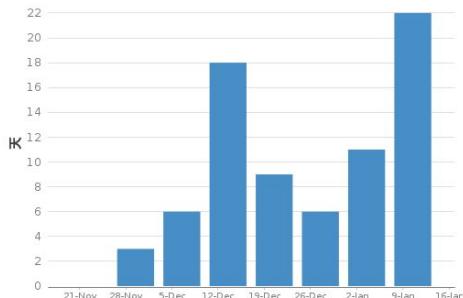
## 最後 50 天內的議題 (已分組 每週)

在議題導覽器中檢視

● 已建立議題 (48)

● 已解決的議題 (46)

## 解決時間: 北科軟體工程

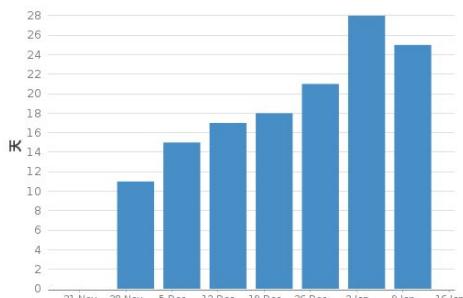


此圖表顯示了指定期間內已解決議題開放的平均天數。

期間：最後 50 天 (按每週分組)

刚刚

## 平均存在時間圖表: 北科軟體工程

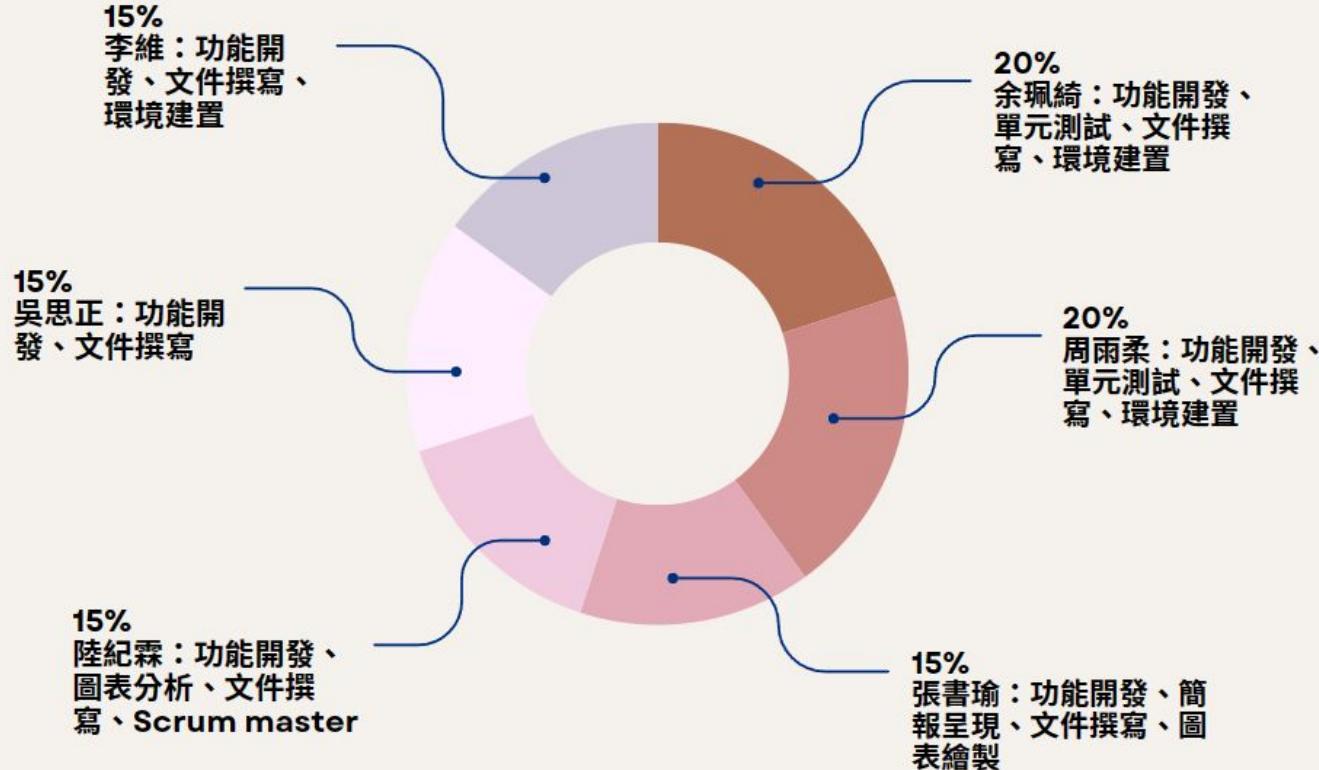


此圖表顯示指定期間內未解決議題的平均存在天數。

# project retrospective

做好的部分	學到的部分
<ol style="list-style-type: none"><li>需求規劃上選擇開發TMS功能眼光很不錯</li><li>對於使用trello和Jira幫助團隊合作的部分做得很好</li><li>Scurm執行的流程有越來越上手</li><li>劃分工作的技巧越加熟練，估點數的方法也有進步</li></ol>	<ol style="list-style-type: none"><li>對於開發的功能要連同Test Case也完成才算完成</li><li>User Story要配套驗收標準</li><li>使用對的合作方法也要有對的紀律才能完整體現方法的優勢</li></ol>
沒有做好的部分	需要改進的部分
<ol style="list-style-type: none"><li>一開始開發心態沒有準備好</li><li>對於可以幫助開發工作的軟體沒有探索很多</li><li>太晚發現寫 unit test 的重要性</li></ol>	<ol style="list-style-type: none"><li>可以之後以TDD為開發方法</li><li>早點寫測試可以引入CI/CD</li><li>量力而為領取Task</li></ol>

# 小組貢獻





Thank you