

Project Dashboard

Project Execution Plan (PEP)

Version: 1.1

Team#02

Name	ID	E-mail
余珮綺	112598028	t112598028@ntut.org.tw
周雨柔	112598004	t112598004@ntut.org.tw
張書瑜	112598021	t112598021@ntut.org.tw
陸紀霖	112C72010	t112C72010@ntut.org.tw
吳思正	111C71003	t111C71003@ntut.org.tw
李維	111C71015	t111C71015@ntut.org.tw

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

11/27/23

目錄 (Table of Contents)

版次變更記錄 (Change Log)	1
Section 1 專案規劃及查核點 (Project Planning and Milestone Checking)	2
1.1 專案工作內容 (Project Work Description)	2
1.1.1 技術方法(Technical Approach)	2
1.1.2 分工結構圖 (Work Breakdown Structure)	2
1.1.3 工作分包與工作項目估算模型與方法 (Establish Estimates of Project Attributes)	2
1.1.4 工作分包與工作項目總表 (List of Work Packages and Tasks)	3
1.1.5 工作分包與工作項目內容說明 (Descriptions of Work Packages and Tasks)	4
1.1.6 專案生命週期定義 (Project Life Cycle)	6
1.2 預定時程及查核點 (Schedule and Milestone Checking)	6
1.2.1 預訂查核點說明 (Milestone Checking Description)	6
1.2.2 預定時程 (Schedule)	7
1.2.3 時程與進度審查監控機制說明 (Schedule & Progress Monitor and Control Mechanism)	7
Section 2 專案成員工作指派 (Personnel)	8
2.1 工作項目或工作分包預估需求與估算之假設條件 (Task Estimation Assumptions)	

2.2 計畫成員指派 (Roles and Responsibilities)	8
2.3 調整專案成員 (Adjustments)	9
2.4 專案專業知識與技能需求 (Requirements of Knowledges and Skills)	9
2.5 訓練計畫表 (Training Plan)	10
2.6 成員參與情況監控機制說明(Monitor and Control Mechanism)	10
Section 3 資源需求 (Resources)	11
3.1 計畫經費預算說明 (Budget)	11
3.2 人事費用估算 (Estimations of Personnel Fee)	12
3.3 計畫經費預估表 (Project Cost Estimation)	12
3.4 預算監控機制說明 (Budget Monitor and Control Mechanism)	12
Section 4 資料管理規劃 (Data Management Plan)	13
4.1 資料管理計畫 (Data Management Plan)	13
4.2 列管資料總表 (List of Managed Data)	13
4.3 列管資料監控機制說明 (Monitor and Control Mechanism)	13
Section 5 風險評估 (Risk Management)	14
5.1 風險項目評估 (Risks Assessment)	14
5.2 風險監控機制說明 (Risk Monitor and Control Mechanism)	
(此項目為必要監控項目)	14
Section 6 建構管理計畫 (Configuration Management Plan)	15

6.1 目的 (Purpose)	15
6.2 建立基準 (Establish Baselines)	15
6.2.1 標示建構管理項目 (Identify Configuration Items)	15
6.2.2 運用建立建構管理系統 (Establish a CM System)	15
6.2.3 建立基準 (Create or Release Baselines)	15
6.3 異動追蹤與控制 (Track and Control Changes)	15
6.3.1 異動追蹤 (Track Change)	15
6.3.2 建構控制小組 (Configuration Control Board)	15
6.3.3 異動控制 (Control Change)	16
6.4 版本控制程式 (The Version Control Tool)	16
6.5 達成完整性 (Establish Integrity)	16
6.5.1 建構管理記錄 (Establish Configuration Management Records)	16
6.5.2 建構審核 (Perform Configuration Audits)	16
Section 7 度量與分析計畫 (Measurement and Analysis Plan)	17
7.1 目的 (Purpose)	17
7.2 蒐集資訊的目的與資訊需求 (Information Needs and Objectives)	17
7.3 基礎度量 (Base Measurement)	17
7.4 度量與分析工具 (Measurement and Analysis Tool)	17
Section 8 流程與產品品質保證計劃 (PPQA Plan)	18
8.1 目的 (Purpose)	18

8.2客觀檢視流程與產品 (Objectively Evaluate Process and Work Packages)	18
8.3 專案目標洞察 (Project Objective Insight)	18
8.4管理架構 (Management Architecture)	18

1. 版次變更記錄 (Change Log)

R

Version	Primary Author(s)	Description of Version	Date Completed
1.0	余珮綺 周雨柔 張書瑜 陸紀霖 吳思正 李維	初期需求分析、進度規劃與討論	23/10/05
1.1	余珮綺 周雨柔 張書瑜 陸紀霖 吳思正 李維	初期需求分析、進度重新規劃與討論	23/11/21

Section 1 專案規劃及查核點 (Project Planning and Milestone

Checking)

1.1 專案工作內容 (Project Work Description)

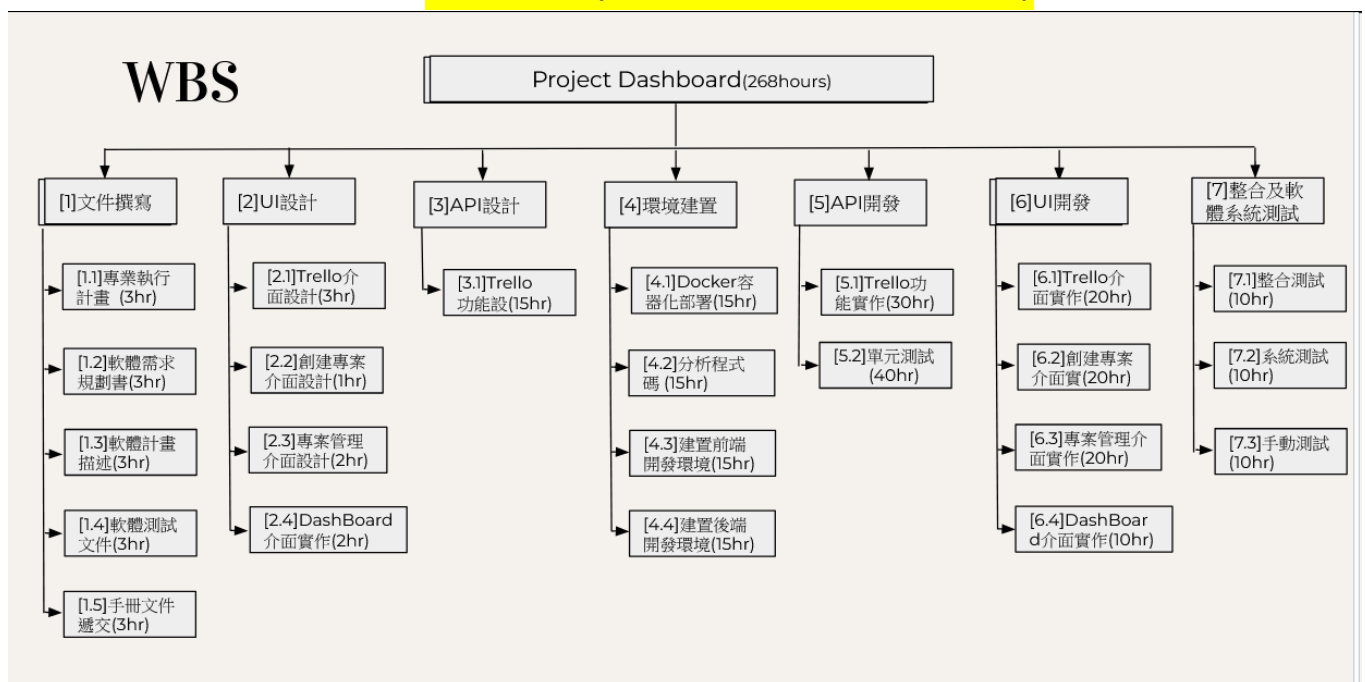
1.1.1 技術方法(Technical Approach)

本系統目的是提供一個可以管理多項專案的網頁平台，讓使用者可以簡單的追蹤公開的 Repository 資訊，例如 **GitLab** 上的 Repository 之 Issue、Commit 等。並透過視覺化資訊圖表，讓使用者更容易進行專案管理。

此系統對於專案的發展可以持續保持著追蹤，透過瀏覽器的互動式介面，將資訊輸出成專案進度報表，並透過各種不同方式查看所呈現的圖表，藉此得知專案的狀態，並適時地對專案做調整控制，來達到專案管理追蹤的效果。另外該平台也能追蹤 Trello 上的報表，為使用者提供更多的資訊。

此系統沿用學長先前所開發的系統進行改良，開發環境會 ASP.NET Core 中開發。前端使用 TypeScript 語言和 Vue.js 框架來開發，後端使用 C#語言撰寫，資料庫使用 SQLite 來管理。

1.1.2 分工結構圖 (Work Breakdown Structure)



1.1.3 工作分包與工作項目估算模型與方法 (Establish Estimates of Project Attributes)

- 專家法（透過個人專業判斷，進行估算）
- 經驗法（根據歷史資料）
- 時間法（根據交付時間，考慮技術可行性，逆推各階段所需要時間）

1.1.4 工作分包與工作項目總表 (List of Work Packages and Tasks)

項目名稱	開始時間	完成時間	工期 (天)	工時 (小時)
Project Dashboard	2023/10/05	2023/01/10	66	268
1 需求文件撰寫	2023/10/05	2023/01/10	4	12
1.1 專案執行計畫	2023/10/05	2023/10/11	1	3
1.2 軟體需求規格書	2023/11/02	2023/11/15	1	3
1.3 軟體計劃描述	2023/11/30	2023/12/13	1	3
1.4 軟體測試文件	2023/12/24	2023/12/27	1	3
1.5 手冊文件遞交	2023/12/28	2023/01/10	0	0
2 UI 設計	2023/10/19	2023/11/15	4	8
2.1 Trello 資訊介面設計	2023/10/19	2023/11/01	1	3
2.2 創建專案介面設計	2023/10/19	2023/11/01	1	1
2.3 管理專案介面設計	2023/10/19	2023/11/01	1	2
2.4 DashBoard 介面設計	2023/11/02	2023/11/15	1	2
3 API 設計	2023/11/02	2023/11/29	4	15
3.1 Trello 功能設計	2023/11/02	2023/11/29	4	15
4 環境建置	2023/11/16	2023/11/30	14	60
4.1 Docker 容器部屬	2023/11/16	2023/11/20	4	15
4.2 分析程式碼	2023/11/18	2023/11/20	2	15
4.3 建置前端開發環境	2023/11/20	2023/11/30	10	15
4.4 建置後端開發環境	2023/11/20	2023/11/30	10	15
5 API 開發	2023/12/1	2023/12/21	20	120
5.1 Trello 實作與設計	2023/12/1	2023/12/10	5	30
5.2 單元測試	2023/12/1	2023/12/21	7	40
6 UI 開發	2023/12/21	2024/1/03	14	90
6.1 Trello 資訊介面實作	2023/12/21	2023/12/24	3	20
6.2 創建專案介面實作	2023/12/22	2023/12/27	3	20
6.3 管理專案介面實作	2023/12/22	2024/1/03	3	20
6.4 DashBoard 介面實作	2023/11/30	2023/1/03	2	10

7 整合及軟體系統測試	2024/1/04	2024/01/10	6	20
7.1 整合測試	2024/1/04	2024/01/10	2	10
7.2 系統測試	2024/1/04	2024/01/10	2	10
7.2 手動測試	2024/1/04	2024/01/10	2	10

1.1.5 工作分包與工作項目內容說明 (Descriptions of Work Packages and Tasks)

1 需求文件撰寫				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
1.1 專案執行計畫	撰寫專案執行計畫	PEP	對專案的了解、軟體工程	MS Word
1.2 軟體需求規格書	撰寫軟體相關的規格書	SRS	專案分析、軟體架構、軟體工程	MS Word
1.3 軟體計劃描述	描述軟體設計的架構	SDD	專案分析、軟體架構、軟體工程	MS Word
1.4 軟體測試文件	描述如何測試軟體的文件	STD	軟體測試、軟體架構	MS Word
1.5 手冊文件遞交	確認手冊送達		溝通技巧	

2 UI 設計				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
2.1 Trello 資訊介面設計	根據需求設計 Trello 資訊介面	介面圖	介面設計	MS PowerPoint
2.2 創建專案介面設計	根據需求設計創建專案介面	介面圖	介面設計	MS PowerPoint
2.3 管理專案介面設計	根據需求設計管理專案介面	介面圖	介面設計	MS PowerPoint

2.5 DashBoard 介面設計	根據需求設計 DashBoard 介面	介面圖	介面設計	MS PowerPoint
--------------------------	------------------------	-----	------	------------------

3 API 設計				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
3.1 Trello 功能設計	決定要從 Trello 獲得的資訊	code	介面設計	VS Code

4 API 設計				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
4.1 Docker	決定 Docker 建置內容	code	軟體架 構	Docker
4.2 分析程式碼	了解 Docker 建置 內容	code	軟體架 構	Docker
4.3 建構前端開發環 境	Docke r 建置前 端內容	code	軟體架 構	Docker
4.4 建構後端開發環 境	Docke r 建置後 端內容	code	軟體架 構	Docker

5 API 開發				
----------	--	--	--	--

項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
5.1 Trello 實作與設計	實作與測試	code	軟體設計、軟體架構	VS Code
5.2 單元測試	測試	code	單元測試	VS Code
5.3 創建專案介面實作	實作與測試	code	介面設計，軟體設計	VS Code
5.4 管理專案介面實作	實作與測試	code	介面設計，軟體設計	VS Code
5.5 Dashboard 介面實作	實作與測試	code	介面設計，軟體設計	VS Code

6 UI 開發				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
6.1 Trello 資訊介面實作	呈現有關 Trello 資訊，進行實作與測試	code	介面設計，軟體設計	VS Code
6.2 創建專案介面實作	實作與測試	code	介面設計，軟體設計	VS Code
6.3 管理專案介面實作	實作與測試	code	介面設計，軟體設計	VS Code
6.4 Dashboard 介面實作	實作與測試	code	介面設計，軟體設計	VS Code

7 整合及軟體系統

測試				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
7.1 整合測試	實作與測試	code	介面設計，軟體設計	VS Code
7.2 系統測試	實作與測試	code	介面設計，軟體設計	VS Code
7.3 手動測試	實作與測試	code	介面設計，軟體設計	VS Code

8 整合及軟體系統測試				
項目編號/名稱	內容說明	產品	需求技能與知識	軟體工具
8.1 整合測試	Web Application 測試	code	軟體架構、軟體測試	VS Code
8.2 系統測試	Unit test, Integration test	code	軟體架構、軟體測試	VS Code

1.1.6 專案生命週期定義 (Project Life Cycle)

使用 Scrum Sprint 來定義專案生命週期。每一個 Sprint 定為二週。

1.2 預定時程及查核點 (Schedule and Milestone Checking)

1.2.1 預訂查核點說明 (Milestone Checking Description)

查核點	預定時間	查核點概述	產品
M1	2023/10/11	完成 PEP	專案執行計畫書
M2	2023/11/01	完成 SRS & Prototyping	系統需求規格書 UI 設計
M3	2023/11/29	Design & Construction of Increment 1	Use Case 和系統架構 軟體計劃描述 (SDD)

M4	2023/12/26	Design & Construction of Increment 2	分析和設計 Model&實作 整合及軟體系統測試軟 體測試文件 (STD)
-----------	------------	---	---

M5	2023/01/03	完成 SDD & STD	Final submission
----	------------	--------------	------------------

1.2.2 預定時程 (Schedule)

Sprint	時間範圍	工作項目
Sprint 1	2023/10/05 ~ 2023/10/11	專案執行計畫 (PEP)
Sprint 2	2023/10/12 ~ 2023/11/01	UI 設計 軟體需求規格 (SRS)
Sprint 3	2023/11/02 ~ 2023/11/15	UI 設計 API 設計
Sprint 4	2023/11/16 ~ 2023/11/29	API 設計 API 開發 軟體計劃描述 (SDD)
Sprint 5	2023/11/30 ~ 2023/12/13	API 開發 UI 開發
Sprint 6	2023/12/14 ~ 2023/12/26	UI 開發 整合及軟體系統測試 軟體測試文件 (STD)
Sprint 7	2023/12/27 ~ 2023/01/03	整合及軟體系統測試 手冊文件遞交

1.2.3 時程與進度審查監控機制說明 (Schedule & Progress

Monitor and Control Mechanism)

本專案中，工作進度以 Sprint 做為基本單位，每兩週安排一次組員討論時間（星

落後的情況發生，則會調整下個 Sprint 的工作進度，確保按預期進行進度。

Section 2 專案成員工作指派 (Personnel)

2.1 工作項目或工作分包預估需求與估算之假設條件

(Task Estimation Assumptions)

※ 工作分包預估方式：

- 專家法（透過個人專業判斷，進行估算）
- 經驗法（根據歷史資料）
- 時間法（根據交付時間，考慮技術可行性，逆推各階段所需要時間）

※ 參數：

- a. 文件： 1 頁 / 1 人時
- b. 系統功能： 1 個 / 8 人時
- c. 每周工作時數為 10 小時（加班視專案及課業程度自行調整）

2.2 計畫成員指派 (Roles and Responsibilities)

姓名	縮寫
余珮綺	PC
陸紀霖	JL
吳思正	WU
李維	LW
周雨柔	YJ
張書瑜	SY

專案人員需求規格

表

活動與交付項目	負責人員	所需的知識與技能
1. 專案執行計畫	PC, JL, WU, LW, YJ, SY	專案分析、軟體架構、軟體工程
1. 軟體需求規格書	PC, JL, WU, LW, YJ, SY	專案分析、軟體架構、軟體工程
1. 軟體計劃描述	PC, JL, WU, LW, YJ, SY	專案分析、軟體架構、軟體工程
1. 軟體測試文件	PC, JL, WU, LW, YJ, SY	系統測試相關知識
1. 手冊文件遞交	PC	溝通與談判
2. Jenkins 資訊介面設計	PC, JL, WU, LW, YJ, SY	軟體設計、軟體架構

2.4 Jira 資訊介面設計	PC, JL, WU, LW, YJ, SY	軟體設計、軟體架構
2.4 創建專案介面設計	PC, JL, WU, LW, YJ, SY	軟體設計、軟體架構
2.4 管理專案介面設計	PC, JL, WU, LW, YJ, SY	軟體設計、軟體架構
2.5 DashBoard 介面設計	PC, JL, WU, LW, YJ, SY	軟體設計、軟體架構
3.1 Jenkins 功能設計	PC, JL, WU, LW, YJ, SY	設計規範、UX 思維、設計思考
3.2 Jira 功能設計	PC, JL, WU, LW, YJ, SY	設計規範、UX 思維、設計思考
3.3 專案資訊視覺化介面設計	PC, JL, WU, LW, YJ, SY	設計規範、UX 思維、設計思考
4.1 Jenkins 實作與設計	PC, JL, WU, LW, YJ, SY	程式設計、後端框架
4.2 Jira 實作與設計	PC, JL, WU, LW, YJ, SY	程式設計、後端框架
4.3 專案資訊視覺化實作與設計	PC, JL, WU, LW, YJ, SY	程式設計、後端框架
4.4 單元測試	PC, JL, WU, LW, YJ, SY	軟體架構、軟體測試
5.1 Jenkins 資訊介面實作	PC, JL, WU, LW, YJ, SY	程式設計、前端架構
5.2 Jira 資訊介面實作	PC, JL, WU, LW, YJ, SY	程式設計、前端架構
5.3 創建專案介面實作	PC, JL, WU, LW, YJ, SY	程式設計、前端架構
5.4 管理專案介面實作	PC, JL, WU, LW, YJ, SY	程式設計、前端架構
5.5 DashBoard 介面實作	PC, JL, WU, LW, YJ, SY	程式設計、前端架構
6.1 整合測試	PC, JL, WU, LW, YJ, SY	軟體架構、軟體測試
6.2 系統測試	PC, JL, WU, LW, YJ, SY	軟體架構、軟體測試

2.3 調整專案成員 (Adjustments)

小組人員在開發過程中基本上不會有更換，但我們也會依照每位成員的任務進度，視情況調整任務。

2.4 專案專業知識與技能需求 (Requirements of Knowledges and Skills)

專業技能及知識	預估需要人數	預計受訓人員	說明
C#	6	6	本專案成員皆無相關經驗
SQLite	6	4	本專案成員有相關經驗 (WU, JL)
TypeScript	6	6	本專案成員皆無相關經驗
Vue. js	6	6	本專案成員皆無相關經驗
HTML 與 CSS	6	2	本專案成員有相關經驗 (SY, YC, WU, PC)
UI design	6	4	本專案成員有相關經驗 (SY, LW)
Testing	6	5	本專案成員有相關經驗 (YC)

2.5 訓練計畫表 (Training Plan)

團隊成員利用組員討論以外的時間進行學習，從網路或書籍來學習技術和知識，在專案的實作過程中慢慢累積經驗。在組員討論時間，由有相關經驗的人帶領討論，成員之間互相學習，將自己學到的知識分享給其他成員。

2.6 成員參與情況監控機制說明(Monitor and Control Mechanism) (此項目為必要監控項目)

(說明監控成員參與情況之實施矯正措施基準及機制)

每週皆會舉辦一次組員討論，會審視所有成員的工作參與情況，以瞭解進度是否有按照規劃日期運行。若某位成員無法達到預期目標時，我們會找出問題並適當調整人員工作內容，並且在能夠滿足客戶需求的最低標準下，依重要程度篩選之後要實作的系統，盡可能趕上所規劃的進度。

Section 3 資源需求 (Resources)

3.1 計畫經費預算說明 (Budget)

工作編號	經費描述	設備費用	管理費用	其他費用	合計
1.1	專案執行計畫		500		500
1.2	軟體需求規格書		500		500
1.3	軟體計劃描述		500		500
1.4	軟體測試文件		500		500
1.5	手冊文件遞交		500		500
2.1	Jenkins 資訊介面設計		500		500
2.2	Jira 資訊介面設計		500		500
2.3	創建專案介面設計		500		500
2.4	管理專案介面設計		500		500
2.5	DashBoard 介面設計		500		500
3.1	Jenkins 功能設計		500		500
3.2	Jira 功能設計		500		500
3.3	專案資訊視覺化介面設計		500		500
4.1	Jenkins 實作與設計		500		500
4.2	Jira 實作與設計		500		500
4.3	專案資訊視覺化實作與設計		500		500
4.4	單元測試		500		500
5.1	Jenkins 資訊介面實作		500		500
5.2	Jira 資訊介面實作		500		500
5.3	創建專案介面實作		500		500
5.4	管理專案介面實作		500		500
5.5	DashBoard 介面實作		500		500
6.1	整合測試		500		500
6.2	系統測試		500		500
合計			12000		12000

3.2 人事費用估算 (Estimations of Personnel Fee)

工作計畫需求人力： 200 小時		總人事費用：240,000	
職級	單位 (小時)	人事費概算	備註
北科學生 (6 人)	300	240,000	薪水
其他 (加班費)	300	0	

3.3 計畫經費預估表 (Project Cost Estimation)

經費項目	預定金額	說明
設備費	60,000	個人電腦補助
業務費	30,000	文件產生的費用
人事費	240,000	薪水
管理費	30,000	專案中各類產物
合計	360,000	

3.4 預算監控機制說明 (Budget Monitor and Control Mechanism)

本專案有關於預算的監控機制為：

- 監控頻率：每學期監控一次。
- 實施矯正之基準及其措施：預算使用超過 10% 即必須實施矯正措施。矯正措施為開會決定如何取得資金，或是刪減專案活動。

Section 4 資料管理規劃 (Data Management Plan)

4.1 資料管理計畫 (Data Management Plan)

本計畫資料管理與儲存方式分為三種：

- A. **原始程式碼：**程式碼以 Git 做版本控制，託管於 GitLab 伺服器。
- B. **電子文件及可執行檔：**保存於 Google 雲端硬碟共用資料夾。
- C. **紙本文件或光碟資料：**由組長負責保管。

4.2 列管資料總表 (List of Managed Data)

資料名稱	版控	建構 管理	機密 等級	產生週期	儲存方式	資料 提供者	資料 使用者
專案執行規劃書	是	否	密	Event	B、C	團隊	團隊
系統需求規格書	是	否	密	Event	B、C	團隊	團隊
系統設計規格書	是	否	密	Event	B、C	團隊	團隊
原始程式碼	是	是	密	Daily	A	團隊	團隊
整合測試計畫書	是	否	密	Event	B、C	團隊	團隊
系統測試報告	是	否	密	Event	B、C	團隊	團隊
系統接受度報告	是	否	密	Event	B、C	使用者	團隊、 使用者

4.3 列管資料監控機制說明 (Monitor and Control Mechanism) (此項目 為必要監控項目)

(說明監控列管資料之實施矯正措施基準及機制)

本專案監控列管資料之矯正措施基準與機制為：

※ 監控頻率：每月監控一次。

實施矯正之基準及其措施：資料管理所列管的所有資料都必須按照資料管理計畫的方式進行，如果發現任何的資料未按資料管理計畫保管或備份，都必須立刻進行矯正，矯正措施為立即增補。

Section 5 風險評估 (Risk Management)

5.1 風險項目評估 (Risks Assessment)

風險項目	發生可能性	影響程度	風險發生處理或避免法
人事變動	1%	高	彼此關心，若遇到困難，則互相幫助，共同解決。
版本衝突	50%	低	解 conflict，並檢討是否需求分得不夠仔細，導致同一區程式有不同變動。
人員訓練不足	15%	低	平日組員應自我充實，且採用 mob 開發，集合大家長處、互相學習。
資料庫毀損	1%	低	平時要定期備份，並妥善保存資料庫文件，如:建置用 sql、ER model 等。
版控伺服器毀損	0.1%	高	養成定期更新本地端 repository 的習慣。
伺服器被入侵	1%	低	定期備份並設置使用者權限。
設備故障	1%	高	使用替代電腦。

5.2 風險監控機制說明 (Risk Monitor and Control Mechanism) (此項目為必要監控項目)

本專案目前推估的高風險事件均為不可預期的，因此只能在面對風險時才能做適時的處理，以下針對高風險議題提出基本的處理方案：

※ 人事變動—

解決方法：團隊採 scrum 開發流程，於 planning 與 meeting 時，了解彼此狀況，若遇到困難，則互相幫助，共同解決。

※ 版控伺服器毀損—

解決方法：版控伺服器毀損機率極低，此外團隊採 mob programming 的方式開發，因此 client 端有原始碼，可直接回復至其他版控伺服器，如: 暫時從 GitLab 移至 GitHub，直至問題解決。

※ 設備故障—

解決方法：使用替代電腦，如實驗室電腦、其他公用電腦。

Section 6 建構管理計畫 (Configuration Management Plan)

6.1 目的 (Purpose)

本專案系統開發為釋出讓使用者去使用，因此可能需要有長時間維護的工作，或是當新的需求被提出要加入時，可能需要有開發的工作，而一個良好的建構管理，即可在一邊開發的同時也可以針對早期版本發現的問題作個別修改，因此才需要此計畫。

6.2 建立基準 (Establish Baselines)

6.2.1 標示建構管理項目 (Identify Configuration Items)

ID	資料名稱	版本控管	建構類別	產生週期	資料提供者	資料使用者
1	專案執行規劃書	否	規格書	Event	團隊	團隊
2	系統需求規格書	否	規格書	Event	團隊	團隊、使用者
3	系統設計規格書	否	規格書	Event	團隊	團隊
4	整合測試計畫書	否	規格書	Event	團隊	團隊、測試者
5	原始程式碼	是	原始碼	Daily	團隊	團隊
6	系統測試報告	否	報告資料	Event	團隊、測試者	團隊、測試者
7	系統接受度報告	否	報告資料	Event	測試、使用者	測試、使用者

6.2.2 運用建立建構管理系統 (Establish a CM System)

本專案系統採用 Google Drive 進行文件的管理，GitLab 進程式的管理。

6.2.3 建立基準 (Create or Release Baselines)

只有原始程式碼使用 GitLab 進行版本控制，因其他沒有版本控管的書面資料，雖會隨系統的開發增加新的內容，但確認後將不會異動，因此皆不需要版本控制。

6.3 異動追蹤與控制 (Track and Control Changes)

6.3.1 異動追蹤 (Track Change)

- 提出 pull request。
- 由團隊人員 code review 後確認並 approve。

6.3.2 建構控制小組 (Configuration Control Board)

小組由開發團隊本身自行監控。

6.3.3 異動控制 (Control Change)

- a. 對於異動的項目對該版本提出一個分支 (branch)。
- b. 該分支 (branch) merge master 確認無 conflict。
- c. 對 master 發出 pull request。
- d. master 完成 merge 後確定功能如同預期。

6.3.4 版本控制程式 (The Version Control Tool)

本專案使用 Git 來進行版本控制。

6.4 達成完整性 (Establish Integrity)

6.4.1 建構管理記錄 (Establish Configuration Management Records)

此管理記錄為建立與維護用來描述建構管理項目的紀錄。參考 Git 的 Commit Log 以及 GitLab PR Page。

6.4.2 建構審核 (Perform Configuration Audits)

審核採用 Pull Request 和 Code review。

Section 7 度量與分析計畫 (Measurement and Analysis Plan, MAP)

7.1 目的 (Purpose)

軟體的度量和分析計畫 (Measurement and Analysis Plan, 簡稱 MAP) 是一份結構化文件，概述了收集、分析和解釋與軟體開發和維護活動相關的數據的目標、方法論和流程。軟體的度量和分析計畫的主要目的是讓組織能夠做出明智的決策，改進其軟體開發流程，並實現其質量和性能目標。

7.2 蒐集資訊的目的與資訊需求 (Information Needs and

Objectives)

序號	目的	資訊需求
1	客戶滿意度	客戶的反應、支援客戶的狀況
2	時程與進度	工作完成狀況、工作單元進度
3	資源與成本	支出、各項資源支援的程度
4	產品品質	系統或成功品質、使用者介面的良劣
5	客戶需求的穩定程度	客戶需求的異動
6	產品大小	每個子系統的大小、功能多寡

7.3 基礎度量 (Base Measurement)

序號	度量	因子
1	客戶滿意度度量	經由客戶問題的反應、與客戶互動的時間
2	工作完成狀況、工作單元進度	工作完成的時間、階層中工作單元完成度
3	支出、各項資源支援的程度	專案人員投入的工作時數、實際支出數
4	系統或成功品質、使用者介面的良劣	系統或功能之錯誤數、使用者反應介面問題
5	客戶需求的異動	客戶需求異動個數、個數、無法修改個數
6	每個子系統的大小、功能多寡	每個子系統的程式行數、功能數

7.4 度量與分析工具 (Measurement and Analysis Tool, MAT)

度量和分析工具 (Measurement and Analysis Tool, 簡稱 MAT) 是一套軟體工具，旨在收集、分析和呈現與軟體開發過程和產品相關的數據和指標。這些工具幫助組織和項目團隊深入了解其軟體項目，監控進展，評估質量並做出明智決策。我們會以 SonarQube 來進程式碼檢測，做為度量分析的依據。

Section 8 流程與產品品質保證計劃 (Process and Product Quality

Assurance, PPQA Plan)

8.1 目的 (Purpose)

PPQA 的主要目的是確保軟體開發過程受控並達到高質量標準，同時確保最終的軟體產品符合質量要求並滿足客戶需求。通過實施 PPQA，組織可以提高軟體項目的成功機會，減少問題和風險，並提供高品質的軟體產品。

8.2 客觀檢視流程與產品 (Objectively Evaluate Process and Work Packages)

- a. 團隊成員共同檢視系統執行流程，並依照使用者手冊和規格書去檢查產品。
- b. 確認產品符合客戶的需求。
- c. 團隊成員共同審查，檢視四份文件（PEP、SRS、SDD、STD）是否無誤。

8.3 專案目標洞察 (Project Objective Insight)

- a. 開發團隊要與 Product owner 討論、確認需求。
- b. Product owner 將需求告知團隊成員並審查開發流程。

8.4 管理架構 (Management Architecture)

由余珮綺擔任組長，其他人為組員。在系統實做與改進的過程中，若認為系統有任何可以改善的地方，均可提出想法，讓團隊共同評估是否要加入系統需求。專案執行上採用 Scrum，成員之間共同合作。