

# Github 專案分析系統

## Project Execution Plan (PEP)

Version: 1.4

Team#2

Name	ID	E-mail
劉文揚	110598096	t110598096@ntut.org.tw
謝狄烽	110598087	t110598087@ntut.org.tw
林妤潔	110598055	t110598055@ntut.org.tw
曹智凱	110598085	t110598085@ntut.org.tw

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

10/15/2021

## 目錄 (Table of Contents)

版次變更記錄 (Change Log).....	3
Section 1 專案規劃及查核點 (Project Planning and Milestone Checking) .....	4
1.1 專案工作內容 (Project Work Description) .....	4
1.1.1 技術方法(Technical Approach).....	4
1.1.2 分工結構圖 (Work Breakdown Structure).....	4
1.1.3 工作分包與工作項目估算模型與方法 (Establish Estimates of Project Attributes).....	4
1.1.4 工作分包與工作項目總表 (List of Work Packages and Tasks). 錯誤! 尚未定義書籤。	
1.1.5 工作分包與工作項目內容說明 (Descriptions of Work Packages and Tasks).....	6
1.1.6 專案生命週期定義 (Project Life Cycle).....	12
1.2 預定時程及查核點 (Schedule and Milestone Checking) .....	12
1.2.1 預訂查核點說明 (Milestone Checking Description).....	12
1.2.2 預定時程 (Schedule).....	13
1.2.3 時程與進度審查監控機制說明 (Schedule & Progress Monitor and Control Mechanism).....	14
Section 2 專案成員工作指派 (Personnel)..... 錯誤! 尚未定義書籤。	
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).....	15
2.2 計畫成員指派 (Roles and Responsibilities) .....	1
Section 3 資源需求 (Resources).....	4
3.1 計畫經費預算說明 (Budget) .....	4
3.2 人事費用估算 (Estimations of Personnel Fee) .....	4
3.3 計畫經費預估表 (Project Cost Estimation) .....	5
3.4 預算監控機制說明 (Budget Monitor and Control Mechanism) .....	5
Section 4 資料管理規劃 (Data Management Plan).....	6
4.1 資料管理計畫 (Data Management Plan) .....	6
4.2 列管資料總表 (List of Managed Data) .....	6
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).....	6
Section 5 風險評估 (Risk Management).....	7
5.1 風險項目評估 (Risks Assessment) .....	7
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).....	7
Section 6 建構管理計畫 (Configuration Management Plan).....	8

6.1 目的 (Purpose) .....	8
6.2 建立基準 (Establish Baselines) .....	8
6.2.1 標示建構管理項目 (Identify Configuration Items).....	8
6.2.2 運用建立建構管理系統 (Establish a CM System).....	8
6.2.3 建立基準 (Create or Release Baselines).....	8
6.3 異動追蹤與控制 (Track and Control Changes) .....	8
6.3.1 異動追蹤 (Track Change).....	8
6.3.2 建構控制小組 (Configuration Control Board).....	8
6.3.3 異動控制 (Control Change).....	8
6.2.4 版本控制程式 (The Version Control Tool).....	9
6.4 達成完整性 (Establish Integrity) .....	9
6.4.1 建構管理記錄 (Establish Configuration Management Records).....	9
6.4.2 建構審核 (Perform Configuration Audits).....	9
Section 7 度量與分析計畫 (Measurement and Analysis Plan) .....	10
7.1 目的 (Purpose) .....	10
7.2 蒐集資訊的目的與資訊需求 (Information Needs and Objectives) .....	10
7.3 基礎度量 (Base Measurement) .....	10
7.4 度量與分析工具 (Measurement and Analysis Tool) .....	10
Section 8 流程與產品品質保證計劃 (PPQA Plan) .....	11
8.1 目的 (Purpose) .....	11
8.2 客觀檢視流程與產品 (Objectively Evaluate Process and Work Packages) .	11
8.3 專案目標洞察 (Project Objective Insight).....	11
8.4 管理架構 (Management Architecture) .....	11

## 版次變更記錄 (Change Log)

Revisions
-----------

Version	Primary Author(s)	Description of Version	Date Completed
1.0	王泓翔、諸政安、溫志嘉、林育德	專案初期規劃討論，專案大綱完成	2020/09/30
1.1	王泓翔、諸政安、溫志嘉、林育德	初期規劃，PEP 討論完成，更新時程評估及對應內容	2020/10/13
1.2	王泓翔、諸政安、溫志嘉、林育德	調整WBS的專案管理時程數值，以及更新相關時間圖表。	2020/12/08
1.3	劉文揚、謝狄烽、林妤潔、曹智凱	重構討論、未來可發展方向討論、專案大綱完成	2021/10/15
1.4	劉文揚、謝狄烽、林妤潔、曹智凱	修正錯誤	2022/01/17

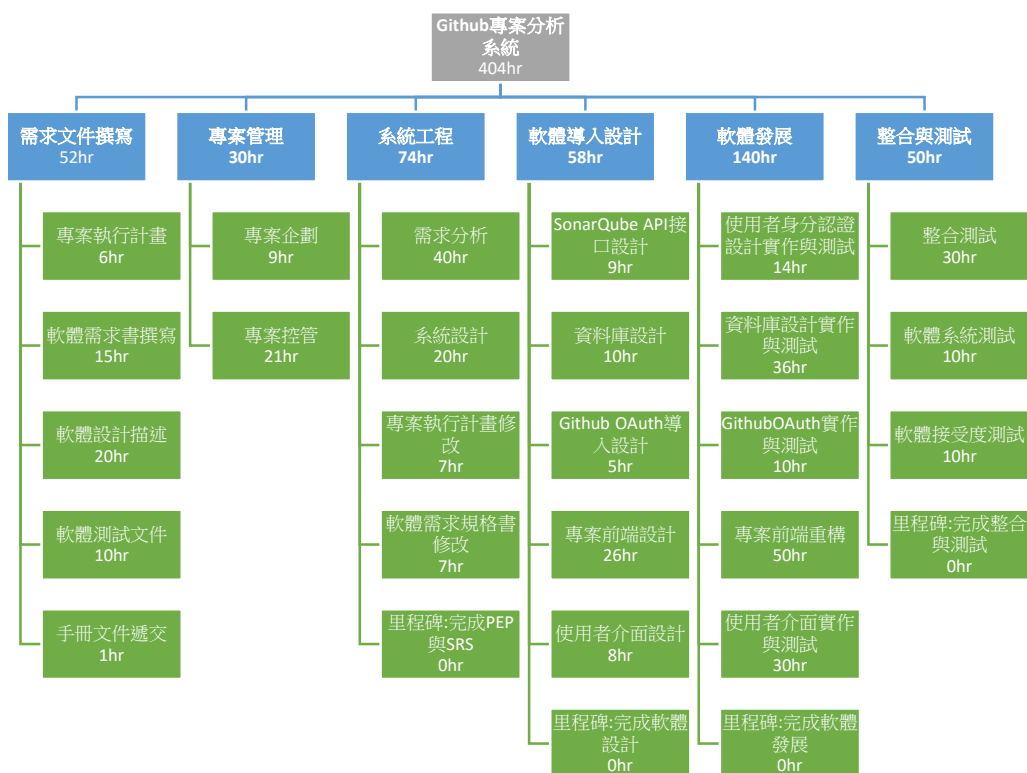
## Section 1 專案規劃及查核點 (Project Planning and Milestone Checking)

### 1.1 專案工作內容 (Project Work Description)

#### 1.1.1 技術方法(Technical Approach)

本系統採用將使用 Java 對於系統進行擴充重構，並且使用 Github 和 SonarQube 提供的 Restful API 取得 json 格式的資料，再進行資料視覺化以及優化前端介面，加入 Selenium 自動化測試網頁，並引入 Jenkins，建置自動化測試的環境。

#### 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)

- ☐ COCOMO
- ☐ COCOMOII
- ☐ 單元估算累加法 (Bottom-up，將工作劃分為較小單元進行估算，再行累加)
- ☐ 專家法(透過個人專業判斷，進行估算)
- ☒ 經驗法(根據歷史資料)
- ☐ 其他估算方式

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

任務名稱	工期	開始時間	完成時間	工時
<b>1 Github 專案分析系統</b>	62 工作日	2021/10/11	2022/01/04	404 工作小時
<b>1.1 需求文件撰寫</b>	8 工作日	2021/10/11	2021/10/20	52 工作小時
1.1.1 專案執行計畫	2 工作日	2021/10/11	2021/10/12	6 工作小時
1.1.2 軟體需求書撰寫	2 工作日	2021/10/13	2021/10/14	15 工作小時
1.1.3 軟體設計描述	2 工作日	2021/10/15	2021/10/18	20 工作小時
1.1.4 軟體測試文件	1 工作日	2021/10/19	2021/10/19	10 工作小時
1.1.5 手冊文件遞交	1 工作日	2021/10/20	2021/10/20	1 工作小時
<b>1.2 專案管理</b>	3 工作日	2021/10/21	2021/10/25	30 工作小時
1.2.1 專案企劃	1 工作日	2021/10/21	2021/10/21	9 工作小時
1.2.2 專案控管	2 工作日	2021/10/22	2021/10/25	21 工作小時
<b>1.3 系統工程</b>	9 工作日	2021/10/26	2021/11/05	74 工作小時
1.3.1 需求分析	4 工作日	2021/10/26	2021/10/30	40 工作小時
1.3.2 系統設計	2 工作日	2021/10/31	2021/11/02	20 工作小時
1.3.3 專案執行計畫修改	1 工作日	2021/11/03	2021/11/03	7 工作小時
1.3.4 軟體需求規格書修改	2 工作日	2021/11/04	2021/11/05	7 工作小時
1.3.5 里程碑:完成 PEP 與 SRS	0 工作日	2021/11/05	2021/11/05	0 工作小時
<b>1.4 軟體導入設計</b>	9 工作日	2021/11/06	2021/11/16	58 工作小時
1.4.1 SonarQube API 接口設計	1 工作日	2021/11/06	2021/11/06	9 工作小時
1.4.2 資料庫設計	2 工作日	2021/11/07	2021/11/08	10 工作小時
1.4.3 Github OAuth 導入設計	1 工作日	2021/11/09	2021/11/09	5 工作小時
1.4.4 專案前端設計	3 工作日	2021/11/10	2021/11/14	26 工作小時
1.4.5 使用者介面設計	2 工作日	2021/11/15	2021/11/16	8 工作小時
1.4.6 里程碑:完成軟體設計	0 工作日	2021/11/16	2021/11/16	0 工作小時
<b>1.5 軟體發展</b>	15 工作日	2021/11/17	2021/12/08	140 工作小時
1.5.1 使用者身分認證設計實作與測試	2 工作日	2021/11/18	2021/11/19	14 工作小時
1.5.2 資料庫設計實作與測試	2 工作日	2021/11/20	2021/11/23	36 工作小時
1.5.3 Github OAuth 實作與測試	1 工作日	2021/11/24	2021/11/24	10 工作小時
1.5.4 專案前端重構	4 工作日	2021/11/25	2021/11/30	50 工作小時
1.5.5 使用者介面實作與測試	6 工作日	2021/12/1	2021/12/08	30 工作小時
1.5.6 里程碑:完成軟體發展	0 工作日	2021/12/08	2021/12/08	0 工作小時
<b>1.6 整合與測試</b>	18 工作日	2021/12/09	2022/01/04	50 工作小時
1.6.1 整合測試	5 工作日	2021/12/09	2021/12/15	30 工作小時
1.6.2 軟體系統測試	3 工作日	2021/12/16	2021/12/20	10 工作小時
1.6.3 軟體接受度測試	10 工作日	2021/12/21	2022/01/04	10 工作小時
1.6.4 里程碑:完成整合與測試	0 工作日	2022/01/04	2022/01/04	0 工作小時

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

1.1 需求文件撰寫		
工作項目 編號/名稱	1.1.1 專案執行計畫	
工作內容說明	1.1.1 專案執行計畫	
工作產品/格式	PEP / 文件	
與其他工作的相依性		
需求技能與知識	專案分析、管理、軟體架構、軟體工程	
工作時程	2 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱	1.1.2 軟體需求規格書撰寫	
工作內容說明	撰寫軟體相關的規格書	
工作產品/格式	SRS / 文件	
與其他工作的相依性		
需求技能與知識	專案分析、管理、軟體架構、軟體工程	
工作時程	2 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱	1.1.3 軟體設計描述	
工作內容說明	描述軟體設計的架構	
工作產品/格式	SDD / 文件	
與其他工作的相依性		
需求技能與知識	專案分析、軟體架構、軟體工程	
工作時程	2 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱	1.1.4 軟體測試文件	
工作內容說明	描述如何測試軟體的文件	
工作產品/格式	STD / 文件	
與其他工作的相依性		
需求技能與知識	軟體架構、軟體測試、專業技術	
工作時程	1 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱		1.1.5 手冊文件繳交
工作內容說明		軟體使用說明文件
工作產品/格式		說明文件 / 文件
與其他工作的相依性		
需求技能與知識		溝通與談判
工作時程		1 工作日
資源	人力	1 人
	硬體設備	PC
	軟體工具	Microsoft Office

1.2 專案管理		
工作項目 編號/名稱		1.2.1 專案企劃
工作內容說明		規劃專案發展的企劃
工作產品/格式		PEP / 文件
與其他工作的相依性		1.1.1
需求技能與知識		專案分析、管理、軟體架構、軟體工程
工作時程		1 工作日
資源	人力	2 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱		1.2.2 專案控管
工作內容說明		專案執行過程中的版本控制
工作產品/格式		Git / code
與其他工作的相依性		1.4、1.5、1.6
需求技能與知識		程式設計
工作時程		2 工作日
資源	人力	4 人
	硬體設備	PC
	軟體工具	Github

1.3 系統工程		
工作項目 編號/名稱		1.3.1 需求分析
工作內容說明		確認專案需求，並做評估
工作產品/格式		技術文件 / 文件
與其他工作的相依性		1.1.1、1.2.1
需求技能與知識		軟體設計、軟體架構、專業技術
工作時程		4 工作日
資源	人力	4 人
	硬體設備	PC
	軟體工具	Miro

工作項目 編號/名稱		1.3.2 系統設計
工作內容說明		根據需求分析的結果，設計專案
工作產品/格式		技術文件 / 文件
與其他工作的相依性		1.3.1
需求技能與知識		軟體設計、軟體架構、專業技術
工作時程		2 工作日
資源	人力	3 人
	硬體設備	PC
	軟體工具	Draw.io



工作項目 編號/名稱	1.3.3 專案執行計畫修改	
工作內容說明	第一次修改 PEP	
工作產品/格式	PEP / 文件	
與其他工作的相依性	1.3.1、1.3.2	
需求技能與知識	專案分析、管理、軟體架構、軟體工程	
工作時程	1 工作日	
資源	人力	2 人
	硬體設備	PC
	軟體工具	Microsoft Office

工作項目 編號/名稱	1.3.4 專案需求規格書修改	
工作內容說明	第一次修改 SRS	
工作產品/格式	SRS / 文件	
與其他工作的相依性	1.3.1、1.3.2	
需求技能與知識	專案分析、管理、軟體架構、軟體工程	
工作時程	2 工作日	
資源	人力	2 人
	硬體設備	PC
	軟體工具	Microsoft Office

1.4 軟體導入設計		
工作項目 編號/名稱	1.4.1 SonarQube API 接口設計	
工作內容說明	規劃 SonarQube 如何和本系統進行連接。	
工作產品/格式	技術文件 / 文件	
與其他工作的相依性	1.4.2	
需求技能與知識	軟體架構、軟體設計、軟體系統	
工作時程	1 工作日	
資源	人力	1 人
	硬體設備	PC
	軟體工具	Miro

工作項目 編號/名稱	1.4.2 資料庫設計	
工作內容說明	設計與規劃 1. 設計規劃存放授權碼	
工作產品/格式	技術文件 / 文件	
與其他工作的相依性	1.4.1	
需求技能與知識	軟體架構、軟體設計、軟體系統	
工作時程	2 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Miro

工作項目 編號/名稱		1.4.3 Github OAuth 導入設計
工作內容說明		規劃 Github OAuth 如何和本系統進行連接
工作產品/格式		技術文件 / 文件
與其他工作的相依性		1.4.1、 1.4.2
需求技能與知識		軟體架構、軟體設計、軟體系統
工作時程		1 工作日
資源	人力	2 人
	硬體設備	PC
	軟體工具	Miro

工作項目 編號/名稱		1.4.5 使用者介面設計
工作內容說明		設計與規劃網站 UI 重構
工作產品/格式		技術文件 / 文件
與其他工作的相依性		1.4
需求技能與知識		軟體架構、軟體設計、軟體系統
工作時程		2 工作日
資源	人力	2 人
	硬體設備	PC
	軟體工具	Miro

工作項目 編號/名稱		1.4.4 專案前端重構
工作內容說明		規劃使用 ECharts 對前端進行優化
工作產品/格式		技術文件 / 文件
與其他工作的相依性		1.4.1、1.4.2、1.4.3
需求技能與知識		軟體架構、軟體設計、軟體系統
工作時程		3 工作日
資源	人力	4 人
	硬體設備	PC
	軟體工具	Miro

1.5 軟體發展		
工作項目 編號/名稱	1.5.1 SonarQube API 接口實作與測試	
工作內容說明	設計與規劃 1. SonarQube API 接口實作	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5.2、1.5.3	
需求技能與知識	SonarQube	
工作時程	2 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	Git、Sourcetree、IntelliJ

工作項目 編號/名稱	1.5.2 資料庫實作與測試	
工作內容說明	設計與規劃 1. 實作資料表存放授權碼	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5.1	
需求技能與知識	MySQL、JSP	
工作時程	2 工作日	
資源	人力	3 人
	硬體設備	PC
	軟體工具	Git、DataGrip、IntelliJ

工作項目 編號/名稱	1.5.3 GithubOAuth 實作與測試	
工作內容說明	設計與規劃 1. 實作Github OAuth和本系統進行連接	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5.1、1.5.2	
需求技能與知識	API	
工作時程	1 工作日	
資源	人力	3 人
	硬體設備	PC
	軟體工具	Git、sourcetree、IntelliJ

工作項目 編號/名稱	1.5.4 專案前端重構	
工作內容說明	實作使用 ECharts 對前端進行優化	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5.1、1.5.2	
需求技能與知識	JSP、JS	
工作時程	4 工作日	
資源	人力	3 人
	硬體設備	PC
	軟體工具	Git、sourcetree、IntelliJ

工作項目 編號/名稱	1.5.5 使用者介面設計	
工作內容說明	實作網站 UI 重構	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5.2	
需求技能與知識	UI、HTML、CSS、Javascript	
工作時程	6 工作日	
資源	人力	2 人
	硬體設備	PC
	軟體工具	Git、sourcetree、IntelliJ

1.6 測試與整合		
工作項目 編號/名稱	1.6.1 整合測試	
工作內容說明	測試 1. 使用者登入測試 2. 使用者修改個人資料與密碼實作 3. 使用者身分驗證測試 4. 專案內容導入測試 5. 專案資訊 dashboard 測試 6. 資料庫系統測試 7. 網頁跳轉測試 8. 網頁介面測試 9. 專案比對功能測試 10. SonarQube 功能測試 11. GithubOAuth 功能測試 12. ECharts 測試	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5	
需求技能與知識	JAVA、JSP	
工作時程	5 工作日	
資源	人力	4 人
	硬體設備	PC

軟體工具	Junit、IntelliJ
------	----------------

工作項目 編號/名稱	1.6.2 軟體系統測試	
工作內容說明	測試 1. Web application 加強測試 2. Server Loading 加強測試	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5	
需求技能與知識	JAVA、JSP	
工作時程	3 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	JUnit

工作項目 編號/名稱	1.6.3 軟體接受度測試	
工作內容說明	測試 1. 展示系統使用 2. 確認軟體需求 3. 供多個使用者測試，統計使用心得	
工作產品/格式	Code / Code	
與其他工作的相依性	1.5	
需求技能與知識	溝通與談判	
工作時程	10 工作日	
資源	人力	4 人
	硬體設備	PC
	軟體工具	JUnit

### 1.1.6 專案生命週期定義 (Project Life Cycle)

本系統一開始在做需求分析時，所有人專案人員都必須對每個需求有徹底的認知與了解，在分析過程中去了解如何實作，因我們團隊採用混合式的瀑布流程，在實作的階段會利用 Pair programming、Test-driven development 以及 Continuous Integration，同時更加了解使用者的需求，進一步去完成實作。

### 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	技術文件/產出物
M1	2021/10/20	完成 PEP	專案執行規劃書
M2	2021/11/09	完成 SRS	系統需求規格書
M3	2021/12/08	完成軟體設計	系統規劃設計書
M4	2022/01/04	完成軟體發展	各系統的程式碼
M5	2022/01/12	完成整合測試	系統的釋出

### 1.2.2 預定時程 (Schedule)

#	Task Name	工期	Start year/mm/dd	ETA year/mm/dd	2021 / 01 / 01 ~ 2021 / 01 / 30																	
					2021 / 10 / 11 ~ 2021 / 10 / 31	2021 / 11 / 01 ~ 2021 / 11 / 30					2021 / 12 / 01 ~ 2021 / 12 / 31											
					WEEK 12/34567/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7	WEEK 12/3456/7/12/3456/7					
1. Github 專案分析系統																						
1.1	需求文件撰寫	8 days	2021/10/11	2021/10/20																		
1.1.1	專案執行計畫	2 days	2021/10/11	2021/10/12																		
1.1.2	軟體需求書撰寫	2 days	2021/10/13	22/10/14																		
1.1.3	軟體設計描述	2 days	2021/10/15	2021/10/18																		
1.1.4	軟體測試文件	1 days	2021/10/19	2021/10/19																		
1.1.5	手冊文件遞交	1 days	2021/10/20	2021/10/20																		
1.2	專案管理	3 days	2021/10/21	2021/10/25																		
1.2.1	專案企劃	1 days	2021/10/21	2021/10/21																		
1.2.2	專案監督	2 days	2021/10/22	2021/10/25																		
1.3	系統工程	9 days	2021/10/26	2021/11/05																		
1.3.1	需求分析	4 days	2021/10/26	2021/10/30																		
1.3.2	系統設計	2 days	2021/10/31	2021/11/02																		
1.3.3	專案執行計畫修改	1 days	2021/11/03	2021/11/03																		
1.3.4	軟體需求規格書修改	2 days	2021/11/04	2021/11/05																		
1.3.5	里程碑完成PEP與SRS	0 days	2021/11/05	2021/11/05																		
1.4	軟體導入設計	9 days	2021/11/06	2021/11/16																		
1.4.1	SourceCode API接口設計	1 days	2021/11/06	2021/11/06																		
1.4.2	資料庫設計	2 days	2021/11/07	2021/11/08																		
1.4.3	Github OAuth導入設計	1 days	2021/11/09	2021/11/09																		
1.4.4	專案前端設計	3 days	2021/11/10	2021/11/14																		
1.4.5	使用者介面設計	2 days	2021/11/15	2021/11/16																		
1.4.6	里程碑完成軟體設計	0 days	2021/11/16	2021/11/16																		
1.5	軟體發展	15 days	2021/11/17	2021/12/08																		
1.5.1	使用者身分認證設計實作與測試	2 days	2021/11/18	2021/11/19																		
1.5.2	資料庫設計實作與測試	2 days	2021/11/20	2021/11/23																		
1.5.3	Github OAuth實作與測試	1 days	2021/11/24	2021/11/24																		
1.5.4	專案前端重構	4 days	2021/11/25	2021/11/30																		
1.5.5	使用者介面實作與測試	6 days	2021/12/01	2021/12/08																		
1.5.6	里程碑完成軟體發展	0 days	2021/12/08	2021/12/08																		
1.6	整合與測試	18 days	2021/12/09	2022/01/04																		
1.6.1	整合測試	5 days	2021/12/09	2021/12/15																		
1.6.2	軟體系統測試	3 days	2021/12/16	2021/12/20																		
1.6.3	數據庫安裝測試	10 days	2021/12/21	2022/01/04																		
1.6.4	里程碑完成整合與測試	0 days	2022/01/04	2022/01/04																		

### 1.2.3 時程與進度審查監控機制說明 (Schedule & Progress Monitor and Control Mechanism)

本專案對於進行中的工作採用一周 meeting 一次(日期)，由團隊所有人針對所完成的工作比例進行進度審視的方式。而需要實施矯正措施時，其基準為：當進度落後 10% 必須實施修正措施。

修正措施由團隊召開會議，討論如何修改時程規劃，並實行。

## 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：

☐ 歷史資料法

☒ 專家法(透過個人專業判斷，進行估算)

☐ 其他估算法

※ 參數：

a. 文件： 1 頁 / 1 人時

b. 系統功能： 1 個 / 8 人時

c. 假設條件：以人事行政局公佈的年度上班時間為工作日

d. 一日工作時數為 8 小時（加班視專案及課業程度而自行調整）

※ 專業技能需求：

專業技能	需求人數
需求文件撰寫	4
專案管理	4
系統工程	4
專業技能	需求人數
軟體設計	4
軟體發展	4
整合與測試	4



## 2.2 計畫成員指派 (Roles and Responsibilities)

姓名	縮寫
劉文揚	WY
謝狄烽	DF
林好潔	UJ
曹智凱	ZK

專案人員需求規格表			
WBS	活動與交付項目	負責人員	所需的知識與技能
1.1.1	專案執行計畫	WY、DF、UJ、ZK	專案分析、管理、軟體架構、軟體工程
1.1.2	軟體需求規格書撰寫	WY、DF、UJ、ZK	專案分析、管理、軟體架構、軟體工程
1.1.3	軟體設計描述	WY、DF、UJ、ZK	專案分析、軟體架構、軟體工程
1.1.4	軟體測試文件	WY、DF、UJ、ZK	軟體架構、軟體測試、專業技術
1.1.5	手冊文件遞交	WY、DF、UJ、ZK	溝通與談判
1.2.1	專案企劃	WY、DF、UJ、ZK	專案分析、管理、軟體架構、軟體工程
1.2.2	專案控管	WY、DF、UJ、ZK	程式設計
1.3.1	需求分析	WY、DF、UJ、ZK	軟體設計、軟體架構、專業技術
1.3.2	系統設計	WY、DF、UJ、ZK	軟體設計、軟體架構、專業技術
1.3.3	專案執行計畫修改	WY、DF、UJ、ZK	專案分析、管理、軟體架構、軟體工程
1.3.4	軟體需求規格書修改	WY、DF、UJ、ZK	專案分析、管理、軟體架構、軟體工程

1.3.5	里程碑： 完成 PEP 與 SRS	WY、DF、UJ、ZK	Check
1.4.1	SonarQube API 接口設計	WY、DF、UJ、ZK	軟體架構、軟體設計、軟體系統
1.4.2	資料庫設計	WY、DF、UJ、ZK	軟體架構、軟體設計、軟體系統
1.4.3	Github OAuth 導入設計	WY、DF、UJ、ZK	軟體架構、軟體設計、軟體系統
1.4.4	專案核心功能設計	WY、DF、UJ、ZK	軟體架構、軟體設計、軟體系統
1.4.5	使用者介面設計	WY、DF、UJ、ZK	軟體架構、軟體設計、軟體系統
1.4.6	里程碑：完成軟體設計	WY、DF、UJ、ZK	Check
1.5.1	使用者身份認證實作與測試	WY、DF、UJ、ZK	SonarQube
1.5.2	資料庫設計實作與測試	WY、DF、UJ、ZK	MySQL、JSP
1.5.3	GithubOAuth 實作與測試	WY、DF、UJ、ZK	API
1.5.4	專案前端重構	WY、DF、UJ、ZK	JSP、JS
1.5.5	使用者介面實作與測試	WY、DF、UJ、ZK	UI、HTML、CSS、Javascript
1.5.6	里程碑：完成軟體發展	WY、DF、UJ、ZK	Check
1.6.1	整合測試	WY、DF、UJ、ZK	JAVA、JSP
1.6.2	軟體系統測試	WY、DF、UJ、ZK	JAVA、JSP
1.6.3	軟體接受度測試	WY、DF、UJ、ZK	溝通與談判
1.6.4	里程碑：完成整合與測試	WY、DF、UJ、ZK	Check

### 2.3 調整專案成員 (Adjustments)

團隊成員狀態穩定，無特殊情況基本上應該是不會異動，和樂融融感情很好，因此理論上不會出現調整情形。

### 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
JAVA	4	0	本專案的成員已有相關經驗
JSP	4	1	本專案部分的成員無相關經驗
MySQL	4	1	本專案部分的成員無相關經驗
JavaScript	4	1	本專案部分的成員無相關經驗
CSS+HTML	4	2	本專案部分的成員無相關經驗
UI design	4	3	本專案部分的成員無相關經驗
Angular ?	4	4	本專案的成員無相關經驗

### 2.5 訓練計畫表 (Training Plan)

將採用做中學的方式，由已經有相關經驗的成員以 mob 方式帶領無經驗的成員進行開發，並從中攝取經驗與相關技術知識，不須特地開設課程做訓練。

### 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)

專案開發過程中，各開發階段的 Increment 會做一次監控確認，以瞭解進度是否有按照規劃日期運行。當落後預期進度時，將召開會議討論將目前最緊急的任務列為首要完成對象；當開發進度超前之時，將召開回顧會議，把之前因重要性關係而被略過的任務，再次加回來實作。

## Section 2 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	合計
1.1.1	專案執行計畫	60000	5000	65000
1.1.2	軟體需求規格書撰寫		2000	2000
1.1.3	軟體設計描述		2000	2000
1.1.4	軟體測試文件		2000	2000
1.1.5	手冊文件遞交		500	500
1.2.1	專案企劃		4000	4000
1.2.2	專案控管		5000	5000
1.3.1	需求分析		3000	3000
1.3.2	系統設計		4000	4000
1.3.3	專案執行計畫修改		3000	3000
1.3.4	專案需求規格書修改		1000	1000
1.3.5	里程碑：完成 PEP 與 SRS		500	500
1.4.1	SonarQube API 接口設計		1500	1500
1.4.2	資料庫設計		1000	1000
1.4.3	Github OAuth 導入設計		2500	2500
1.4.4	專案前端重構		1500	1500
1.4.5	使用者介面設計		1000	1000
1.4.6	里程碑：完成軟體設計		500	500
1.5.1	SonarQube API 接口實作與測試		3000	3000
1.5.2	資料庫設計實作與測試		6000	6000
1.5.3	Github OAuth 實作與測試		3000	3000
1.5.4	專案前端重構		2000	2000
1.5.5	使用者介面實作與測試		3000	3000
1.5.6	里程碑：完成軟體發展		500	500
1.6.1	整合測試		3000	3000
1.6.2	軟體系統測試		3000	3000
1.6.3	軟體接受度測試		500	500
1.6.4	里程碑：完成整合與測試		500	500
合計		60000	64500	124500
			(單位:NT 新台幣)	

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

工作計畫需求人力:455 人時		總人事費用：404000	
職級	單位(時)	人事費概算	備註
研究生工作員(4 人)	404 小時	404000	系統設計開發、UI 設計、專案控管
其他(加班費)	0 小時	0	

### 3.3 計畫經費預估表 (Project Cost Estimation)

經費項目	預定金額	說明
研究設備費	60000	伺服器與相關設備
業務費	25000	一般業務或特殊業務之用，例如客戶訪談
人事費	000	專案人員共 5 名之研究經費，包含系統設計、UI 設計、專案控管
管理費	2300	專案管理以及其他經費
合計	441000	

### 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：

- a. 監控頻率：每周監控一次。
- b. 實施調整之基準及其措施：實際支出超過預算 5%即必須實施調整措施。調整措施為開會決定如何取得資金，或是刪減專案功能，減少支出項目。

## Section 3 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將分為三種：

- A. 原始程式碼：程式碼以 Git 做版本控制，託管於 Github 伺服器。
- B. 電子文件及可執行檔：由 Google 提供的免費 google 雲端，可以做上傳 file 的動作，因此只要將報表結果或是圖檔之類的文件，直接利用系統上做管理。
- C. 紙本文件或光碟資料：由組長負責做管理。

### 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	否	否	密	Event	C	團隊	團隊
系統需求規格書	否	否	密	Event	C	團隊	團隊、使用者
系統設計規格書	否	否	密	Event	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	否	否	密	Event	B	團隊	團隊、測試者
系統測試報告	否	否	密	Event	C	團隊、測試者	團隊、測試者
系統接受度報告	否	否	密	Event	B	測試、使用者	測試、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 4 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方法
人事變動	1%	極低	團體成員平常積極交流，建立良好的互動，如有問題則保持溝通，避免產生心結。拒絕謾罵與人身攻擊，與人建立良好關係。
版本衝突	10%	低	利用 Git 版本控管，以解決版本不一致情況
人員的訓練不足	15%	中	平日多充實自我能力、組員間彼此討論幫忙, 共同提升
資料庫需求變更	10%	高	降低程式的相依度
資料庫伺服器毀損	0.1%	高	定期備份
版控伺服器毀損	0.1%	高	Client 端之間交叉比對資料
伺服器被入侵	1%	極低	定期備份、設置使用者權限
資料外洩	1%	極低	網站採用會員制，須經由系統管理員核准才可進入網站使用

### 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影響程度的發生都是不可預期的事件，因此只能在面對風險時才能做適時的處理，以下針對高影響程度議題提出基本的處理方案：

※ 資料庫伺服器毀損—

解決方法：做 RAID，以及當需求有重大變更時也要做一次資料庫備份。

※ 資料庫需求變更—

解決方法：即時開會並盡快提出新版資料庫的規劃。

※ 版控伺服器毀損—

解決方法：團員每日均會開發程式並上傳至 server，因此自身 client 端的程式也只會跟 server 有一天的誤差，當發生資料毀損時，通過團員間互相交叉比對即可獲得正確、統一的版本。

## Section 5 建構管理計劃 (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，可能會有開發的工作，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做個別修改，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	系統需求規格書	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
3	系統設計規格書	否	規格書	Event	團隊	團隊
4	整合設計計畫書	否	規格書	Event	團隊	團隊
5	原始程式碼	是	原始碼	Weekly	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	團隊、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 Git 進行建構管理

#### 6.2.3 建立基準 (Create or Release Baselines)

由表 6.2.1 得知，規格書隨著系統開發跟著更新，測試報告書確認後將不會異動，因此皆不需要版本控制，只有原始程式碼要用 Github 進行版本控制。

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

#### 6.3.1 異動追蹤 (Track Change)

- 提出異動申請。
- 由建構管理人員評估影響層面，並通知 Member。
- 由組長邀集受影響單位進行評估，並決定是否准予異動。
- 追蹤異動的狀態。

#### 6.3.2 建構控制小組 (Configuration Control Board)

此小組由團隊本身進行監控。

#### 6.3.3 異動控制 (Control Change)

- 對於要異動的項目對該版本提出一個分支(branch)。
- 再次確認其正確性。
- 確認異動後，必須對此次做異動記錄以及異動原因。



#### 6.2.4 版本控制程式 (The Version Control Tool)

Git，是一個開放原始碼的分散式版本控制軟體。

資料來源：<https://zh.wikipedia.org/wiki/Git>

### 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。而紀錄項目以及資料僅採用 Git 本身的紀錄欄位。

#### 6.4.2 建構審核 (Perform Configuration Audits)

為達成對於建構系統中的分支擴充性，團隊們必須將子系統開發到一定的整合後，再推出一分支，以利後期的除錯。

## Section 6 度量與分析計畫 (Measurement and Analysis Plan)

### 7.1 目的 (Purpose)

為達成對於建構系統中的 branch 擴充性，團隊們必須將子系統開發到一定的整合後，必須再推出一隻 branch，以利後期的 debug。

### 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	里程碑完成狀況、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	系統或功能品質、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	里程碑完成狀況、工作單元進度	里程碑完成的時間、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	系統或功能品質、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)

本專案使用 **功能點分析** 作為系統或功能品質分析。

## Section 7 流程與產品品質保證計劃 (PPQA Plan)

### 8.1 目的 (Purpose)

本系統的流程與產品品質保證計畫主要是提供專案所有人員對於產品的品質能深入瞭解其運作流程，亦可當作系統接受度測試的檢視，以告知使用者關於本系統的品質趨向。

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

以客觀角度評估該流程領域的流程和工作項目，本專案的檢視流程為：

- (1) 由專案負責人及團隊共同檢視該系統整體執行流程、單元審查與單元測試皆符合依照執行架構運行，並將每個頁面運作過程對照使用者手冊及系統規格書進行複查。
- (2) 在驗證流程中共同確認是否有達到 Product Owner 所提出之功能與需求。
- (3) 為增進建構項目之安全性與加強變更之控管，輔助維持專案開發內容之一致性，同時檢視四份文件(PEP、SRS、SDD 及 STD)，於展示的同時進行複查，以確保正確與完整性。

### 8.3 專案目標洞察 (Project Objective Insight)

- (1) 團隊與 Stakeholder 在討論過程中需同步確認需求，以讓本身能更瞭解對方真正的需求。
- (2) 確保團隊對於需求的認知無誤，專案負責人需將個人對於專案需求內容的認知詳細告知團隊，而團隊成員如在專案進行中有不確定或疑慮皆可向專案負責人詢問相關細節。
- (3) 由於需求的變更對整體專案的執行會產生影響，為確保專案計畫與產出物需求一致，將討論內容動紀錄動於 Github 上，成員可隨時視察執行進度與結果。

### 8.4 管理架構 (Management Architecture)

本系統由成員共同負責，沒有職位高低之分，為一個自組織的團隊。當系統實作過程中，覺得系統有任何可以改善的地方，均可提出作為一個系統需求來做評估。

專案執行上雖以瀑布式流程為底，但仍採用 XP 執行精神，於每個工作階段皆是團員們互相分工合作，並不需要特別指派工作分工細項。