

# AMM

## 設計文件

專案名稱	AMM 開發輔助工具
撰寫日期	2022 / 11 / 21
發展者	簡蔚驊 鄧晔宣 余威霆 林佳何 唐劭賢



介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-02	註冊帳號	Account Management Module	Any
連結方式	輸入資料		輸出資料
	email, password		註冊成功後跳轉頁面
對應介面之要求			
接收 email 及 password，交由 firebase 註冊。			

介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-03	忘記密碼	Account Management Module	User
連結方式	輸入資料		輸出資料
	email		驗證完後跳回登入頁
對應介面之要求			
接收 email，交由 firebase 驗證，並由 firebase 寄送驗證信到使用者上。			

介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-04	修改個人資料	Account Management Module	User
連結方式	輸入資料		輸出資料
	avatar url, nickname, bio		顯示新的個人資料
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-05	修改密碼	Account Management Module	User
連結方式	輸入資料		輸出資料
	old password, new password		跳到登入畫面
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-06	Google 日曆授權	Account Management Module	User
連結方式	輸入資料		輸出資料
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-EI-07	取得通知	Account Management Module	User
連結方式	輸入資料		輸出資料
對應介面之要求			
接收資料後			

### 2.1.2 內部介面

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-01	/signin	Account Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "email": "Email (string)", "password": "Password (string)" }		{ "status": "Status (string, 200)", "JWT": "JWT token (string)" }
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-02	/signup	Account Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "email": "Email (string)", "password": "Password (string)" }		{ "status": "Status (string, 200)", "JWT": "JWT token (string)" }
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-03	/forget	Account Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "email": "Email (string)", }	{ "status": "Status (string, 200)", }	
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-04	/person	Account Management Module	User
連結方式	輸入資料		輸出資料
PUT	{ "JWT": "JWT token (string, cookie)", "avatar": "Avatar (base64)", "nickname": "Nickname (string)", "bio": "bio (string)" }		{ "status": "Status (string, 200)", }
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-05	/person	Account Management Module	User
連結方式	輸入資料		輸出資料
GET	{ "JWT": "JWT token (string, cookie)", "email": "Email (string)", }		{ "status": "Status (string, 200)", } 
對應介面之要求			
獲取資料			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-06	/person/reset	Account Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "JWT": "JWT token (string, cookie)", "oldPassword": "Password (string)", "newPassword": "Password (string)", }		{ "status": "Status (string, 200)", } 
對應介面之要求			
改密碼			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-07	/person/calendar	Account Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "JWT": "JWT token (string, cookie)", }		{ "status": "Status (string, 200)", }
對應介面之要求			
接收資料後			

介面編號	介面名稱	介面提供者	介面使用者
AMM-II-08	/notify	Account Management Module	User
連結方式	輸入資料		輸出資料
GET	{ "JWT": "JWT token (string, cookie)", }		{ "status": "Status (string, 200)", } }
對應介面之要求			
接收資料後			

## 2.2 專案管理子系統

### 2.2.1 外部介面

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-01	列出所有專案	Project Management Module	User
連結方式	輸入資料		輸出資料
			列出所有專案的資料
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-02	建立新專案	Project Management Module	User
連結方式	輸入資料		輸出資料
	Project Name, Development Mode		新增完專案後，跳到專案畫面。
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-03	新建 Readme	Project Management Module	User
連結方式	輸入資料		輸出資料
			建立 Readme 檔案
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-04	專案設定修改	Project Management Module	User
連結方式	輸入資料		輸出資料
	Project Name, Development Mode		修改完專案設定後， 跳到專案畫面。
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-05	邀請加入專案	Project Management Module	User
連結方式	輸入資料		輸出資料
	User Mail		
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-06	移除成員	Project Management Module	User
連結方式	輸入資料		輸出資料
	User Mail		
對應介面之要求			



介面編號	介面名稱	介面提供者	介面使用者
PMM-EI-07	刪除專案	Project Management Module	User
連結方式	輸入資料		輸出資料
			刪除完專案，跳回到專案列表。
對應介面之要求			

### 2.2.2 內部介面

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-01	/projects	Project Management Module	User
連結方式	輸入資料		輸出資料
GET			{ "projects": " 專案列表 (Array<string, projectid>)" }
對應介面之要求			
獲取專案列表			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-02	/project	Project Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "projectname": "project name (string)", "developmentmode": "development mode (string)", }		
對應介面之要求			
新建專案			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-03	/project/readme	Project Management Module	User
連結方式	輸入資料		輸出資料
POST			
對應介面之要求			
加 readme			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-04	/project	Project Management Module	User
連結方式	輸入資料		輸出資料
PUT	{ "projectname": "project name (string)", "developmentmode": "development mode (string)", }		
對應介面之要求			
修改專案設定			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-05	/project/member	Project Management Module	User
連結方式	輸入資料		輸出資料
GET			
對應介面之要求			
獲取此專案的成員資料			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-06	/project/member	Project Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "mail": "User mail (string)", }		
對應介面之要求			
邀請成員			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-07	/project/member	Project Management Module	User
連結方式	輸入資料		輸出資料
DELETE	{ "usermail": "User mail (string)", }		
對應介面之要求			
刪除成員			

介面編號	介面名稱	介面提供者	介面使用者
PMM-II-03	/project	Project Management Module	User
連結方式	輸入資料		輸出資料
DELETE			
對應介面之要求			
移除專案			

## 2.3 Repo 管理子系統

### 2.3.1 外部介面

介面編號	介面名稱	介面提供者	介面使用者
RMM-EI-01	列出所有 REPO	Repo Management Module	User
連結方式	輸入資料		輸出資料
			列出所有 REPO
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
RMM-EI-02	新增 REPO	Repo Management Module	User
連結方式	輸入資料		輸出資料
	reponame, repourl		repoid
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
RMM-EI-02	刪除 REPO	Repo Management Module	User
連結方式	輸入資料		輸出資料
	repoid		
對應介面之要求			

### 2.3.2 內部介面

介面編號	介面名稱	介面提供者	介面使用者
RMM-II-01	/project/repos	Repo Management Module	User
連結方式	輸入資料		輸出資料
GET	{ "projectid": "Project ID (string)" }		{ "repos": "REPO 列表" }
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
RMM-II-02	/project/repo	Repo Management Module	User
連結方式	輸入資料		輸出資料
POST	{ "projectid": "Project ID (string)" "reponame": "REPO 名稱 (string)" "repourl": "REPO URL (string)" }		{ "repoid": "REPO ID (string)" }
對應介面之要求			
新增 repo			

介面編號	介面名稱	介面提供者	介面使用者
RMM-II-03	/project/repo	Repo Management Module	User
連結方式	輸入資料		輸出資料
PUT	{ "projectid": "Project ID (string)" "repoid": "REPO ID (string)" "reponame": "REPO 名稱 (string)" }	{ "repoid": "REPO ID (string)" }	
對應介面之要求			
修改 repo			

介面編號	介面名稱	介面提供者	介面使用者
RMM-II-02	/project/repo	Repo Management Module	User
連結方式	輸入資料		輸出資料
DELETE	{ "projectid": "Project ID (string)" "repoid": "Project ID (string)" }		
對應介面之要求			
刪除 repo			

## 2.4 文件管理子系統

### 2.4.1 外部介面

介面編號	介面名稱	介面提供者	介面使用者
DMM-EI-01	列出所有文件	Doc Management Module	User
連結方式	輸入資料		輸出資料
			列出所有文件
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
DMM-EI-02	新增文件	Doc Management Module	User
連結方式	輸入資料		輸出資料
	文件類型		
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
DMM-EI-03	編輯文件	Doc Management Module	User
連結方式	輸入資料		輸出資料
	文件類型		
對應介面之要求			

介面編號	介面名稱	介面提供者	介面使用者
DMM-EI-04	刪除文件	Doc Management Module	User
連結方式	輸入資料		輸出資料
	文件類型		
對應介面之要求			

### 3. 流程設計 (Process Design)

可用 UML 之 Activity diagram / State Transition diagram 或一般的程式流程圖描述所開發的系統流程，目的是運用流程圖或狀態圖設計整個系統的完整運作。

此部份應是操作概念的細部設計。

使用者的使用流程之前的流程圖 + 細部設計

唐

### 4. 使用者畫面設計 (User Interface Design)

可用各種畫面設計方式，如發展網頁 prototype、繪製 PowerPoint 投影片等，去設計你的主要系統畫面。此部份應是 SRS 中使用者介面分析的進階設計版本。

線框圖? 包含所有 UI 設計版型顏色等等

CLI 工具 UI 設計

何

### 5. 資料設計 (Data Design)

設計此系統的資料庫 schema、檔案結構、XML schema、JSON (JavaScript Object Notation) 物件等。

NoSQL 類似 json 格式

簡

## 6. 類別圖設計 (Class Diagram)

發展 UML 之 class diagram (可對應 C4 model 中的 Code diagram)，明確設計系統中包含哪些類別，以及這些類別之間的關係。

若遇到非一般物件類別或特殊物件類別，如 HTML、JavaScript、JSP、Servlet 等，可用 stereotype 表達，如 «JavaScript»、«Servlet»。

可把重點放在定義程式結構：訂出所有類別名稱、拉出所有類別關係，再加入重要的 attribute/operation 即可。而特殊的類別職責設計可加上文字說明。

StarUML

前後端都要

## 7. 實作方案 (Implementation Languages and Platforms)

說明系統之平台 (如網站或行動 App)

說明預計採用的程式語言或技術 (如後端 Java、node.js、PHP 等，前端 JS+Bootstrap、Android、iOS 等)

說明預計採用的框架 (Framework，如前端 React、Angular、Vue，後端 Spring、SpringBoot 等)、函式庫 (如前端 jQuery、d3.js，後端 jsoup、iText)、Servless 服務 (Firebase) 等。

輕輕鬆酥

## 8. 設計議題 (Design Issue)

將設計此系統過程中遭遇的議題紀錄下來，包含 (1) 議題內容、(2) 可能解決方案 (alternative solutions)、(3) 最後解決方案 (final solution) 與理由 (rationale)。

唐劭賢輕輕鬆酥

大家有問題都丟這邊ㄟ