



電機工程系

110學年度專題製作期末報告

Python應用-社群APP開發

學號： U07127138

學生： 曾 楷 歲

學號： U07127139

學生： 黃 于 紘

指導老師： 邱機平老師

2022 年 5 月

明志科技大學電機工程系授權書

本授權書所授權之專題製作期末報告全文與電子檔，為本組於明志科技大學電機工程系撰寫之專題製作期末報告。本組同意開放紙本與電子檔全文以非專屬、無償授權明志科技大學電機工程系，基於推動讀者間「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、光碟、網路或其它各種方法收錄、重製、與發行，或再授權他人以各種方法重製與利用。以提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

專題名稱： Python 應用-社群 APP 開發

指導老師：邱機平 老師

專題生簽名：

學號：

日期：民國 年 月 日

備註：

1. 本授權書請填寫並親筆簽名後，裝訂於各紙本專題製作期末報告之書名頁後。
2. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列專題製作期末報告，應依著作權法相關規定辦理。

表號：A120130305

誌謝

在專題的過程中，我們特別感謝邱機平教授的指導，讓我們在專題的各個過程中知道如何解決所遇到的難題。感謝系上所舉辦的各種活動好讓我們團隊有經費購買硬體設備。

摘要

本專題預計開發出一款社群平台，終結乏味生活，鼓勵人們嘗試不同事物，因此決定發展一個具有話題性的主題『挑戰』作為新平台。透過對生活的挑戰，希望能把群眾拉進現實場景，體驗更實質上的互動性，以話題搭起人與人之間的橋樑，建立更緊密的連結。

關鍵字：社群、平台、挑戰、話題

目 錄

明志科技大學電機工程系授權書	i
誌謝	ii
摘要	iii
第一章 前言	1
1.1 研究動機	1
1.2 實作方法	2
第二章 相關資料	5
2.1 Python 程式語言介紹	5
2.2 Kivy 程式語言介紹	5
2.3 OPENVPN 介紹	6
2.4 Material design 介紹	6
2.5 虛擬機器 VM 介紹	7
2.6 Figma 介紹	7
2.7 MSSQL 介紹	8
2.8 MSSQL-Agent 介紹	8
2.9 DOCKER 介紹	9
2.10 NAS 720+介紹	9
2.11 路由器介紹	10
第三章 實作步驟	11
3.1 系統架構規劃	11
3.2 環境建置	11
3.2.1 後端資料庫架設	11
3.3 VPN server	13
3.4 NAS 檔按存取	14
3.5 UI 設計	14
3.5.1 設計 FIGMA 繪製 UI 介面	14
3.5.2 UI 程式	15
3.6 資料庫功能參數設計	17
3.7 前端與後端 api 串接	18
3.8 程式碼打包	19
3.8.1 建立 Ubuntu 虛擬環境	19
3.8.2 使用 Buildozer 打包程式(.APK)	19
3.8.3 設定 buildozer.spec，相關使用到的套件皆須明確填寫 ...	19
3.8.4 連接手機，手機要開啟開發者模式	19
3.8.5 程式閃退問題	19
3.9 UI 功能	20

3.9.1 登入.....	20
第四章 實作成果.....	24
4.1 製作成果.....	24
第五章 結論.....	27
參考文獻.....	28

圖目錄

圖 1- 1 資料庫參數規劃、API 呼叫規劃	2
圖 1- 2 資料庫數據建模結構	2
圖 1- 3 資料庫存取	3
圖 1- 4 資料庫自動化運行模組	3
圖 1- 5 Figma 設計畫面	4
圖 1- 6 邏輯處理.py	4
圖 1- 7 畫面處理.kv	4
圖 1- 8 Kivy 程式碼應用	5
圖 2- 1 VM 架構	7
圖 2- 2MSSQL	8
圖 2- 3 SQL Server Agent 代理	8
圖 2- 4 Docker 結構	9
圖 2- 5NAS720+	9
圖 2- 6 Docker 結構	10
圖 3- 1 系統架構規劃	11
圖 3- 2 安裝 Docker	11
圖 3- 3MSSQL 參數	12
圖 3- 4SQL 運行於 Docker	12
圖 3- 5SSMS 登入	13
圖 3- 6OPENVPN 設定檔	13
圖 3- 7NAS 網路設定	13
圖 3- 8Figma 設計畫面	14
圖 3- 9Google Fonts	14
圖 3- 10Materia Design icons	15
圖 3- 11Material Design	15
圖 3- 12Kivymd	15
圖 3- 13 前端開發資料夾分類	16
圖 3- 14 元件	16
圖 3- 15 畫面分類	16
圖 3- 16 資料庫、NAS、UI 頁面管理	16
圖 3- 17 前端程式架構	17
圖 3- 18 資料庫運行狀況	17
圖 3- 19 使用者關聯圖	17
圖 3- 20 挑戰任務關聯圖	18

圖 3- 21pymssql 套件.....	18
圖 3- 22DB 串接指令.....	18
圖 3- 23 圖預存程序呼叫指令.....	18
圖 3- 24 使用者資料庫資料.....	20
圖 3- 25 登入介面.....	20
圖 3- 26 註冊介面.....	21
圖 3- 27 接收&上傳介面.....	22
圖 3- 28 發起介面.....	22
圖 3- 29 評核介面.....	23
圖 3- 30 獎勵介面.....	23
圖 4- 1 登入.....	24
圖 4- 2 註冊.....	24
圖 4- 3 主頁.....	24
圖 4- 4 挑戰列表.....	24
圖 4- 5 挑戰主題.....	25
圖 4- 6 推薦挑戰.....	25
圖 4- 7 用戶貼文.....	25
圖 4- 8 上傳表單.....	25
圖 4- 9 新增挑戰.....	26

第一章 前言

1.1 研究動機

現今社會，人人都有社群帳號，『分享、按讚、追蹤』已經是最常聽見的口號，自媒體經營風潮的崛起，讓社群平台成為一個很好投放廣告平台，這是一個廠商與自媒體互惠的趨勢，不過當今，『發圖文、短片、直播』的內容太雜亂，即使有『#hashtsg』之類的標記，不過多數人很少利用此功能，來瀏覽相關內容。

因此，團隊決定選擇具有話題性的主題『挑戰』作為新平台的核心價值，然而以往的社群平台是以上傳影片為主要分享方式，我們希望能把群眾拉進現實場景，更有實質上的互動性，平台命名為『Challenger APP』。

1.2 實作方法

在製作 APP 過程中我們將重點放在規劃。良好的規劃能夠減少很多不必要的功能開發。

必要參數.txt - 記事本	
檔案(E) 編輯(E) 格式(O) 檢視(V) 說明	
4. 接受的任務主題ID	"AcceptMissionTopicID"
5. 接受日期	"UploadDate"
6. 完成狀態	"Finish"
使用者APP使用時間(UserAPPUseTime)	
1. 使用者ID	"UserAPPUseTimeID"
2. 累積使用時間	"CumulativeUseTime"
使用者上傳影片 or 照片(UserUploadVideoOrPhoto)	
1. 擁有者名稱ID	"UserUploadVideoOrPhotoID"
2. 影片 or 照片 位址	"VideoPictureURL"
3. 影片 or 照片 標題	"Title"
4. 影片 or 照片上傳時間	"DataUpdateTime"
5. 挑戰任務ID	"ChallengeMissionID"
6. 挑戰任務類型ID	"ChallengeMissionTypeID"
7. 挑戰任務主題ID	"ChallengeMissionTopicID"
*****有關評核的部分*****	
評核(Assessment)	
1. 被評核者	"Evaluated"
2. 評核者	"Rater"
3. 被評核任務ID	"TaskAssessedID"
4. 被評核任務類型ID	"TaskAssessedTypeID"
5. 被評核任務主題ID	"TaskAssessedTopic"
6. 給予分數(1:喜歡.. -1:不喜歡)	"JudgePoint"
7. 資料新增時間(方便做排名分析及評核分數計算)	"DataUpdateTime" 很重要
8. 影片 or 照片網址	"VideoPictureURL"
9. 被評核任務種子碼	"ChallengeMissionSeed"
評核有效排名紀錄(AssessmentRank)	
1. 被評核任務對應種子碼	"ChallengeMissionSeed"
2. 評核分數	"JudgePoint"
3. 資料新增時間	"DataUpdateTime"
評核結果紀錄時間(AssessmentTimeRecord)	
1. 被評核任務對應種子碼	"ChallengeMissionSeed" 抓ChallengeMissionUserRecord資料表的任務種子
2. 評核排名截止時間	"TimeDeadLine" 先固定為24hr

圖 1- 1 資料庫參數規劃、API 呼叫規劃

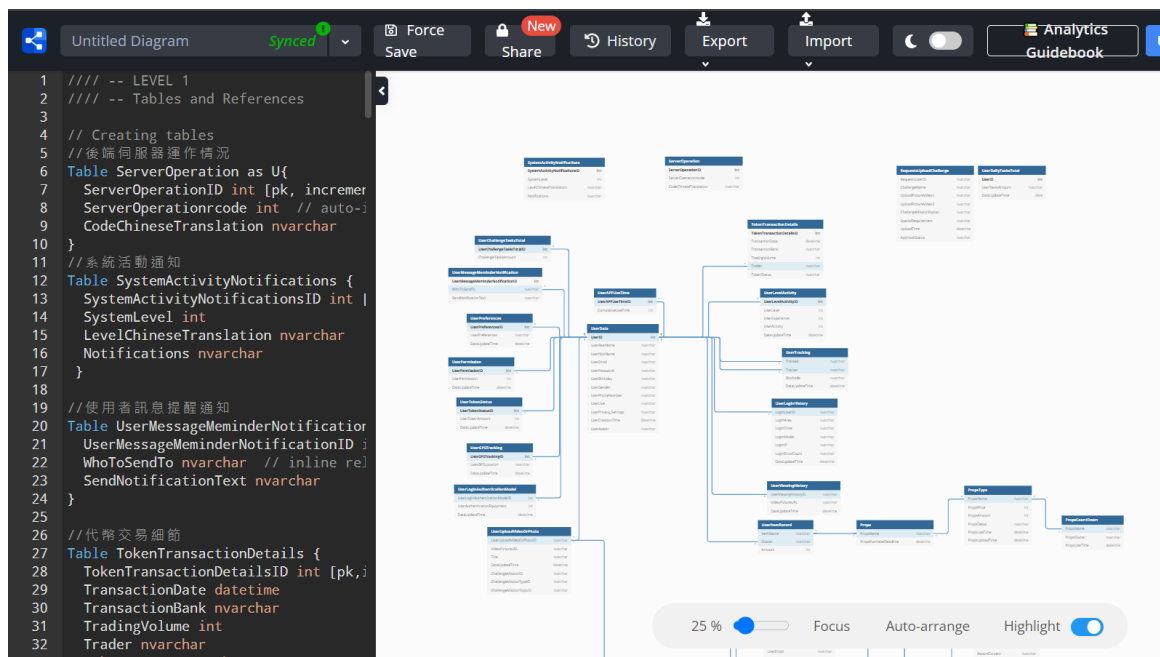


圖 1- 2 資料庫數據建模結構

我們的資料庫選用 Microsoft SQL Server，進行資料存取、分析及自動化運行，並透過 API 將資料傳遞給 Python Kivy(前端)。

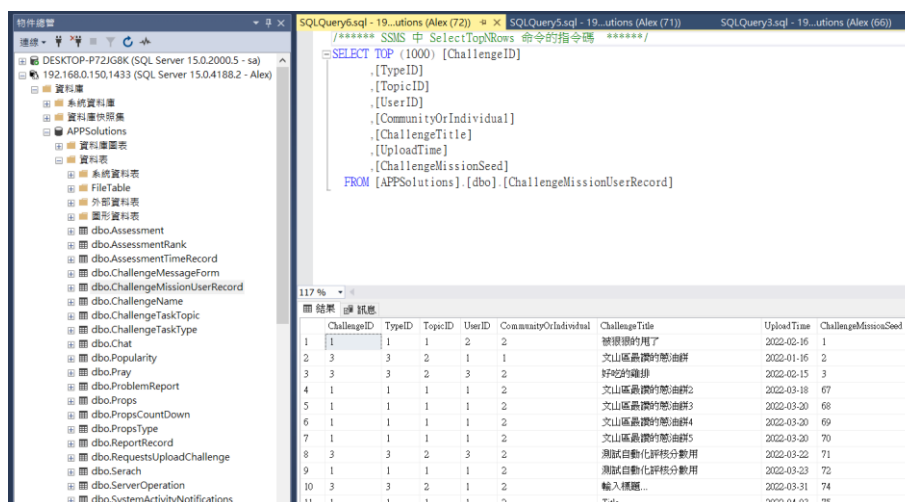


圖 1- 3 資料庫存取

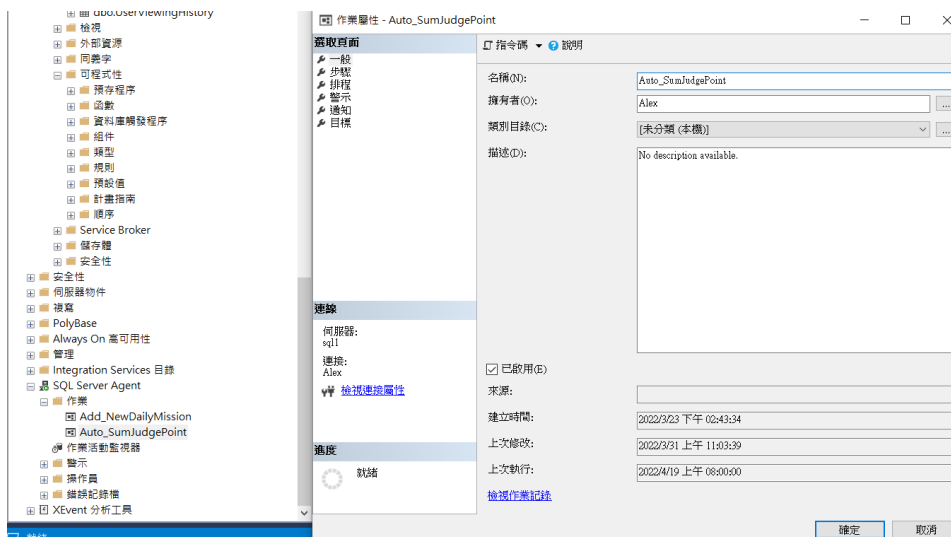


圖 1- 4 資料庫自動化運行模組

第二章 相關資料

2.1 Python 程式語言介紹

Python 是一種易學、功能強大的程式語言。它有高效能的高階資料結構，也有簡單但有效的方法去實現物件導向程式設計。Python 優雅的語法和動態型別，結合其直譯特性，使它成為眾多領域和大多數平臺上，撰寫腳本和快速開發應用程式的理想語言。

Python 有幾項特點：

1. 容易撰寫
2. 功能強大
3. 跨平台
4. 容易擴充

2.2 Kivy 程式語言介紹

Kivy 是開源 Python 函式庫，用於開發行動應用程式和其它採用自然使用者界面的多點觸控應用軟體。Kivy 致力於描述使用者界面和與使用者互動的一種語言。就像 QML，它能夠簡單的建造整個使用者界面並加上互動。它能夠採用 Python 快速的造出基本的控制項，之後用 kivy 造出使用者界面。

```
#:kivy 1.4.0
<LoadDialog>:
    BoxLayout:
        size: root.size
        pos: root.pos
        orientation: "vertical"
        FileChooserListView:
            id: filechooser

        BoxLayout:
            size_hint_y: None
            height: 30
            Button:
                text: "Cancel"
                on_release: root.cancel()

            Button:
                text: "Load"
                on_release: root.load(filechooser.path, filechooser.selection)
```

圖 1- 8 Kivy 程式碼應用

2.3 OPENVPN 介紹



OpenVPN 是一種廣泛使用且高度安全的協議。這種開源協議運行在 TCP 或 UDP 網際網路協議之上，TCP 這種通訊協議能確保數據以正確的順序完整傳輸，而 UDP 則專注於更快的速度。

優點：

1. 開放源碼：這表示程式碼是公開的，任何人都能檢查程式碼內隱藏的後門，或可能危及 VPN 安全的漏洞。
2. 多功能：這種協議能與一系列不同的加密和通訊協定一起使用，能配置為不同用途，也能根據需要保持安全和輕便。
3. 安全性：非常安全，能搭配各種加密協議。
4. 輕鬆繞過防火牆：某些防火牆不允許 VPN 流量通過，然而使用 OpenVPN 能輕鬆繞過防火牆進行傳輸。

2.4 Material design 介紹

Material Design 是 Google 創建的設計系統，內含元件和工具系統，清楚的定義出許多元件的規範，減少了設計人員與開發人員的溝通時間，幫助團隊可以快速的建構出精美的產品，讓使用者在操作介面時能有高品質的使用體驗。

2.5 虛擬機器 VM 介紹

虛擬機器是實體電腦的虛擬呈現或模擬環境。虛擬化可讓您在單一實體機器上建立多個虛擬機器，每個虛擬機器都有自己的作業系統（OS）和應用程式。虛擬機器無法與實體電腦直接互動。而是需要藉助一個叫做 Hypervisor 的輕量型軟體層，在虛擬機器和底層的實體硬體之間進行協調。Hypervisor 負責將實體運算資源（例如處理器、記憶體及儲存設備）配置給每個虛擬機器。

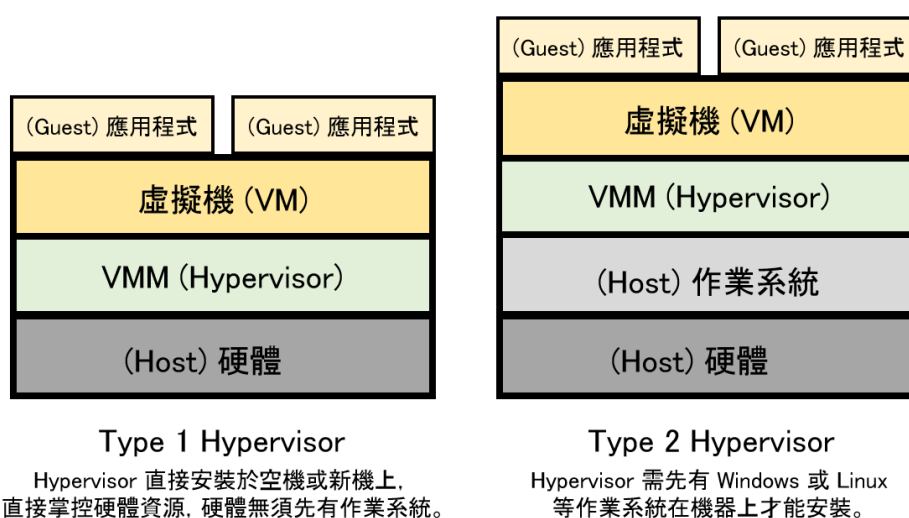


圖 2- 1 VM 架構

2.6 Figma 介紹

Figma 是線上介面設計工具，只需網路即可直接在瀏覽器上操作的 UI 設計，因此不限系統類型，不佔電腦空間，可多位設計師共同編輯設計檔，版本歷史功能更直覺切換。Figma 是，允許和團隊進行多人協同合作，在瀏覽器中進行設計，共同繪製使用者介面，UI、UX、網頁、App 應用程式的介面原型，只要有一個成員變更內容，其他使用者都能同步查看變更和加入編輯，讓團隊可以共同討論，創作出最有共識的產品。

2.7 MSSQL 介紹

SQL Server 是 Microsoft 公司推出的關係型資料庫管理系統。具有使用方便可伸縮性好與相關軟體集成程度高等優點，使用集成的商業智慧型 (BI) 工具提供了企業級的數據管理。Microsoft SQL Server 資料庫引擎為關係型數據和結構化數據提供了更安全可靠的存儲功能，使您可以構建和管理用於業務的高可用和高性能的數據應用程式。

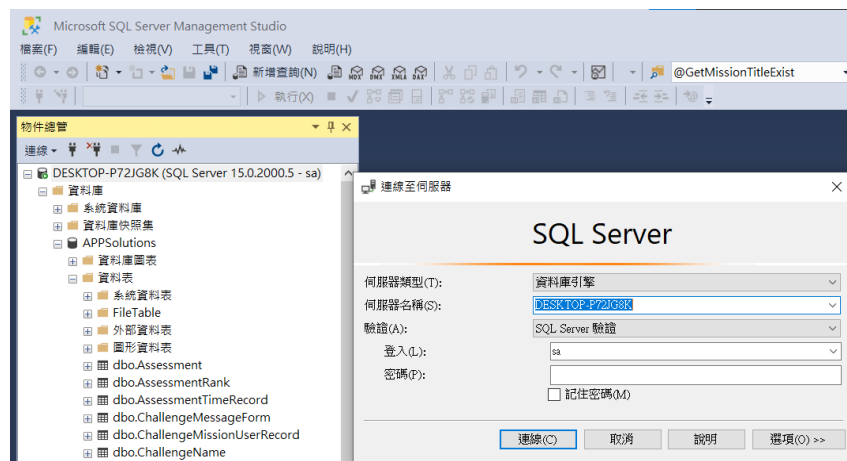


圖 2- 2MSSQL

2.8 MSSQL-Agent 介紹

SQL Server 代理程式是一項 Microsoft Windows 服務，會執行排程的系統管理工作，這些工作會在 SQL Server 中呼叫。SQL Server Agent 會使用 SQL Server 來儲存作業資訊。作業包含了一個或多個作業步驟。每一個步驟包含它自己的工作。

執行作業有數種方式：

1. 依照一個或多個排程
2. 回應一個或多個警示
3. 執行 sp_start_job 預存程序

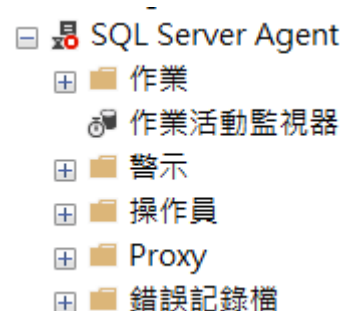


圖 2- 3 SQL Server Agent 代理

2.9 DOCKER 介紹

Docker 是一個開源專案，提供開發、轉移和執行應用程式。Docker 讓基礎建設的管理更容易，就如同管理應用程式一樣。讓部署、測試程式碼更加便利，並將應用程式獨立於 Container 當中，省去搭建作業系統的成本，使得程式運行更快速。

Docker Components:

1. Docker Image (映像檔)
2. Docker Container (容器)
3. Docker Registry (倉庫)

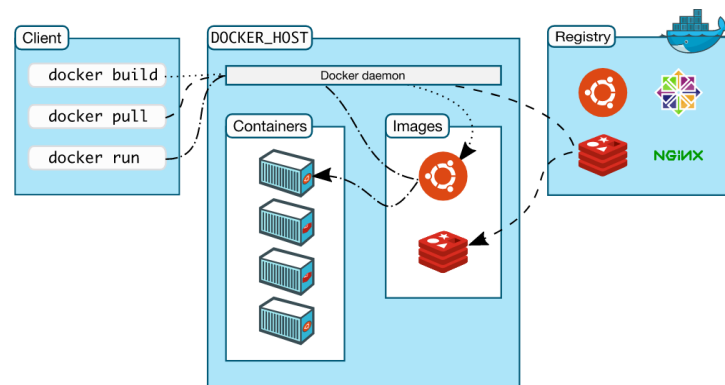


圖 2- 4 Docker 結構

2.10 NAS 720+介紹

NAS 720+ 是一台小型網路連接儲存設備，可做備份、Docker、網站 server、VPN server 等架設。

1. 備份
2. Docker
3. 網站伺服器
4. VPN 伺服器



2.11 路由器介紹

此專題使用的路由器是 TP-LINK，透過通訊埠的設定，能夠使外部網路連接到架設好的伺服器。



圖 2-6 路由器

第三章 實作步驟

3.1 系統架構規劃

先進行系統架構的設計，本團隊使用的工具是 Draw.io，它可以直接線上設計。

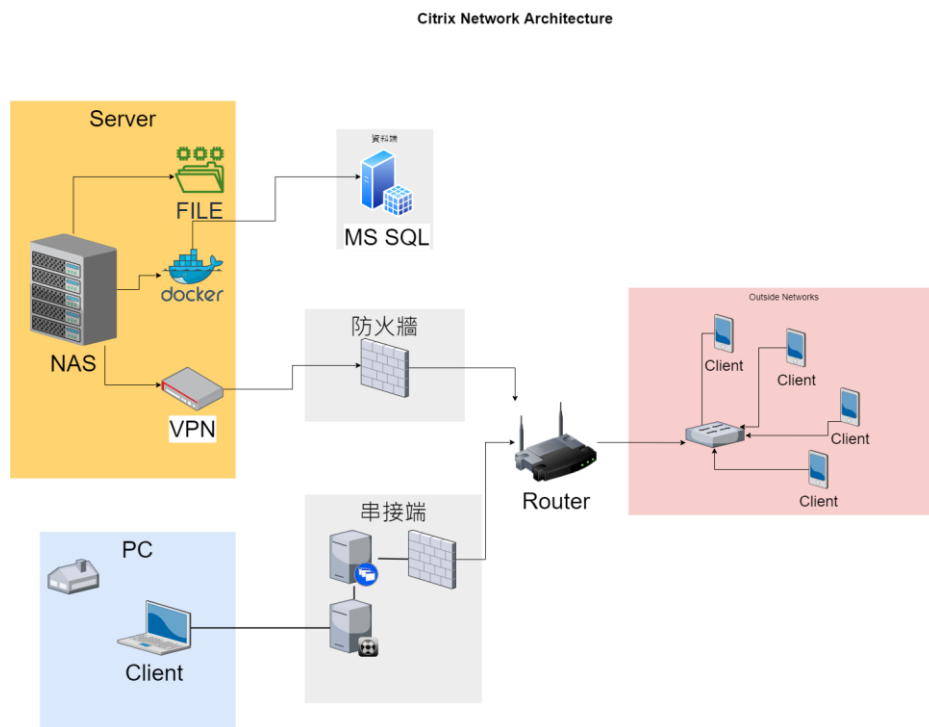


圖 3- 1 系統架構規劃

3.2 環境建置

3.2.1 後端資料庫架設

Step1 在 NAS 安裝 Docker

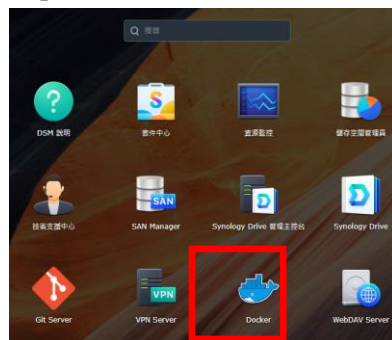


圖 3- 2 安裝 Docker

Step2 確認 NAS SSH 功能有開啟，也確認 port number，確認使用

的帳號是否有管理權限

Step3 透過 SSH 登入 NAS

- `ssh <username>@<nas_ip> -p <port>`

Step4 使用 root 權限

- `sudo -i`

Step5 下載 MSSQL 映像檔到 Docker 容器中

- `sudo docker pull`
`mcr.microsoft.com/mssql/server:2019-latest`

Step6 設定 MSSQL 與 docker 的相關 port 及帳號密碼

- `sudo docker run -e "ACCEPT_EULA=Y" -e \`
`"SA_PASSWORD=<YourStrong@Passw0rd>" \`
`-p 1433:1433 --name sql1 -h sql1 \`
`-dmcr.microsoft.com/mssql/server:2019-latest`

參數	描述
<code>-e "ACCEPT_EULA=Y"</code>	將 <code>ACCEPT_EULA</code> 變數設定為任何值，以確認是否接受 End-User 授權合約。此為 SQL Server 映像的必要設定。
<code>-e "SA_PASSWORD=<YourStrong@Passw0rd>"</code>	請指定至少為八個字元且符合 SQL Server 密碼需求的強式密碼。此為 SQL Server 映像的必要設定。
<code>-p 1433:1433</code>	將主機環境上的 TCP 連接埠 (第一個值) 對應至容器中的 TCP 連接埠 (第二個值)。在本範例中，SQL Server 正在接聽容器中的 TCP 1433 且對主機上的連接埠 1433 公開。
<code>--name sql1</code>	為容器指定自訂名稱，而不使用隨機產生的名稱。若您執行數個容器，就無法使用此相同名稱。
<code>-h sql1</code>	用來明確設定容器主機名稱。如果您未指定主機名稱，則會預設為容器識別碼，這是隨機產生的系統 GUID。
<code>mcr.microsoft.com/mssql/server:2019-latest</code>	SQL Server 2019 Ubuntu Linux 容器映像。

圖 3- 3MSSQL 參數

Step7 運行確認

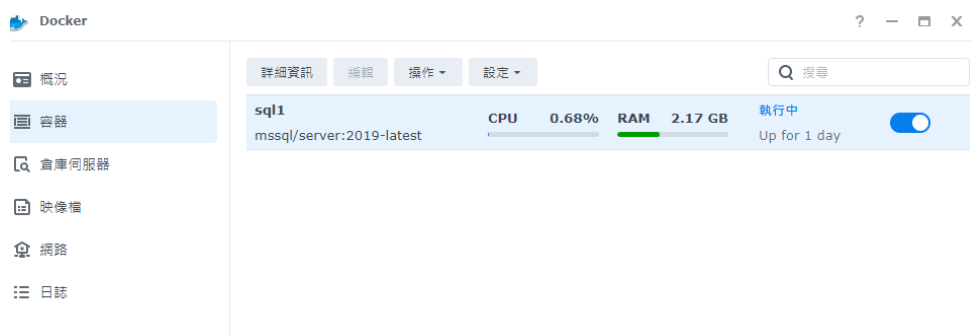


圖 3- 4SQL 運行於 Docker

Step8 利用 SSMS 連線 MSSQL

- server: <nas_ip>, <port>
- account: sa <--default
- password: 建立 docker 時設定的密碼

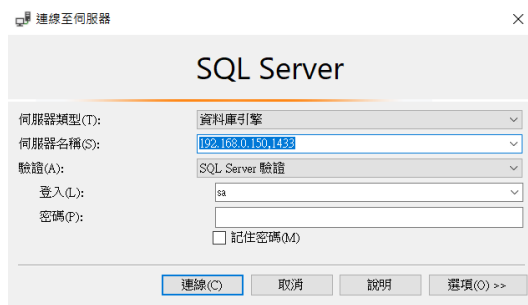


圖 3- 5SSMS 登入

3.3 VPN server

Step1 在 NAS 上安裝 VPN server 套件

Step2 設定 IP 範圍、連接埠、通訊協定，匯出設定檔

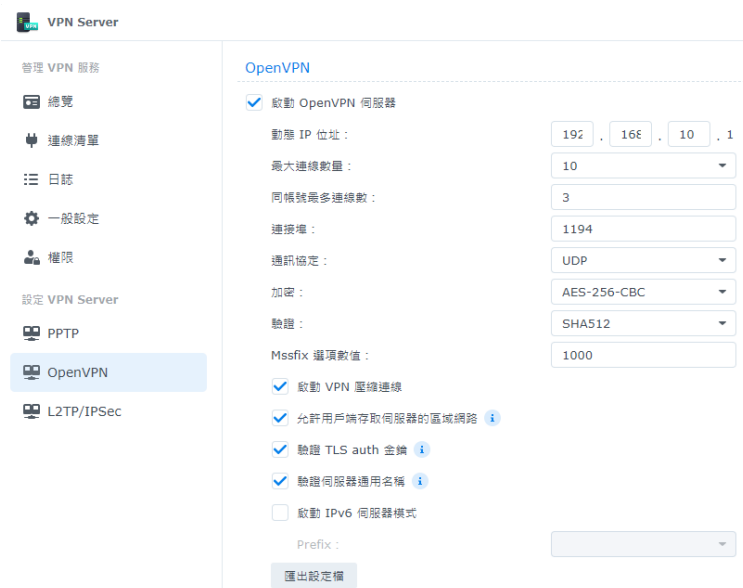


圖 3- 6OPENVPN 設定檔

Step3 NAS 網路環境設定

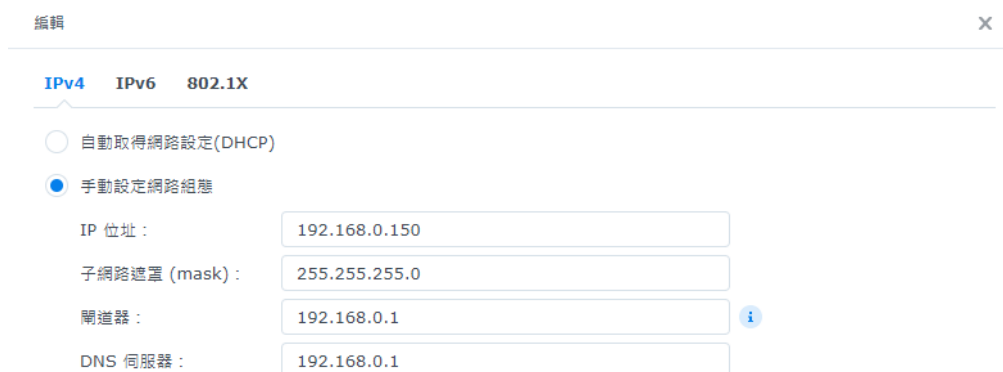


圖 3- 7NAS 網路設定

Step4 DDNS 申請及設定(無固定 IP，以 DDNS 方式設定)

Step5 路由器虛擬通訊埠需開通，防火牆也要確認能通過

3.4 NAS 檔按存取

Step1 存取資料夾連線設定

Step2 存取權限設定

Step3 透過 NAS API 做讀取寫入

3.5 UI 設計

3.5.1 設計 FIGMA 繪製 UI 介面

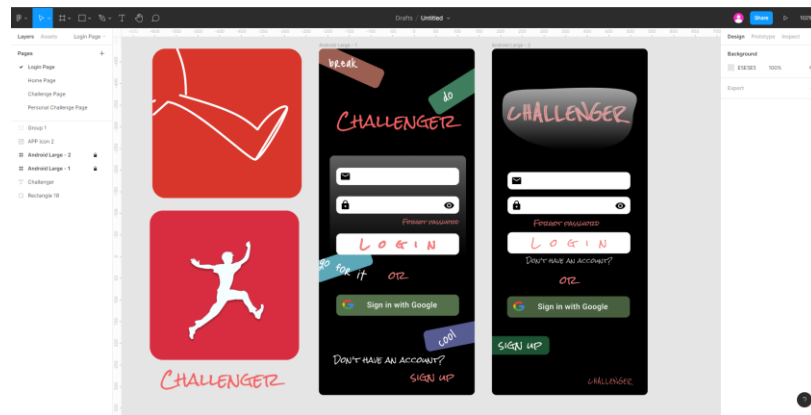


圖 3- 8Figma 設計畫面

3.5.1.1 設計參考

字型：Google Fonts



圖 3- 9Google Fonts

圖示：Material Design icons

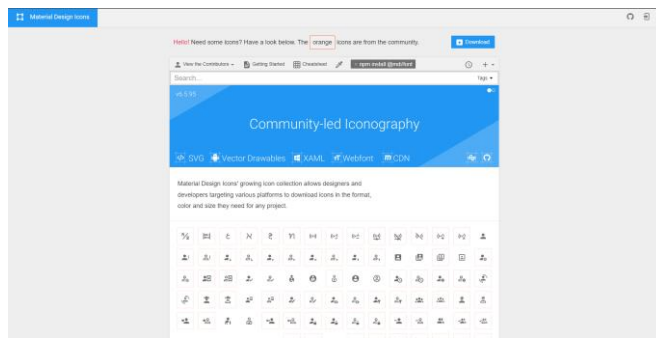


圖 3- 10Material Design icons

UI 元件：Material Design、Kivymd

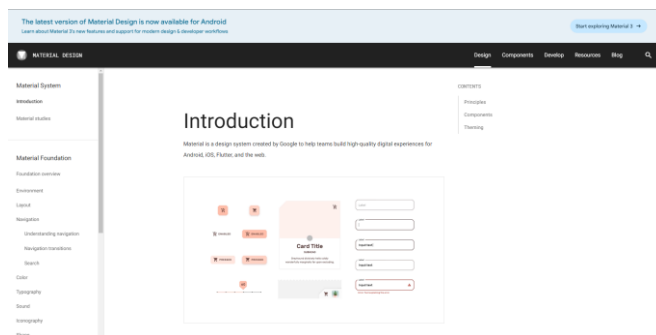


圖 3- 11Material Design

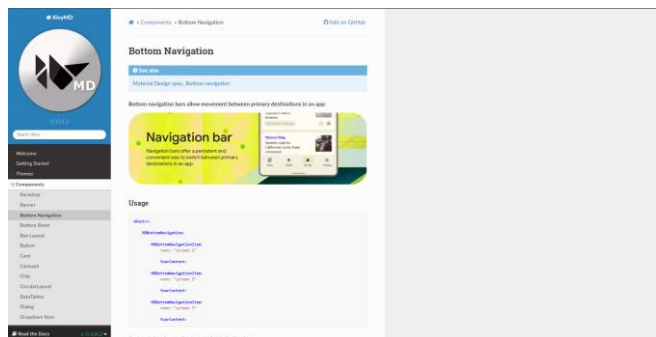


圖 3- 12Kivymd

3.5.2 UI 程式

語言：kivy

Step1 建立資料夾(分類)

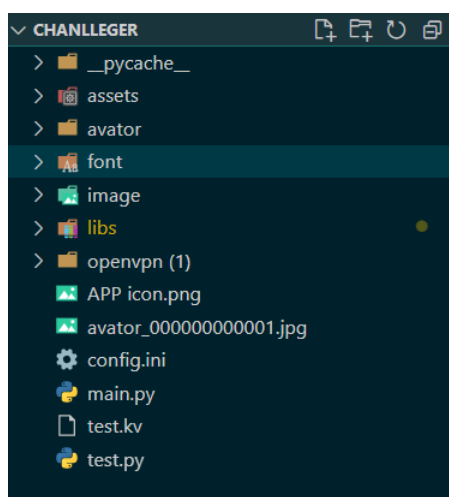


圖 3- 13 前端開發資料夾分類

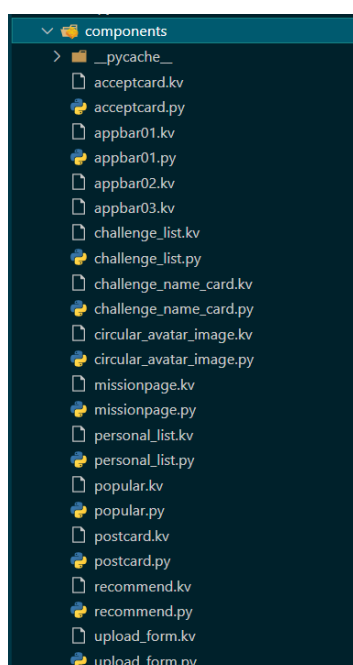


圖 3- 14 元件

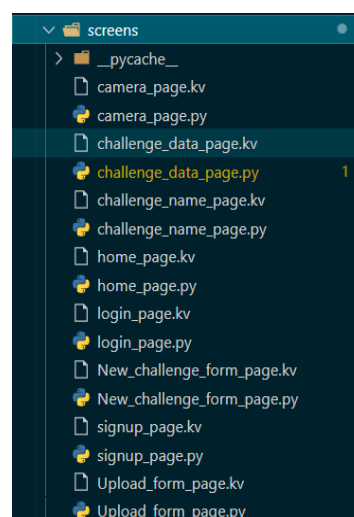


圖 3- 15 畫面分類

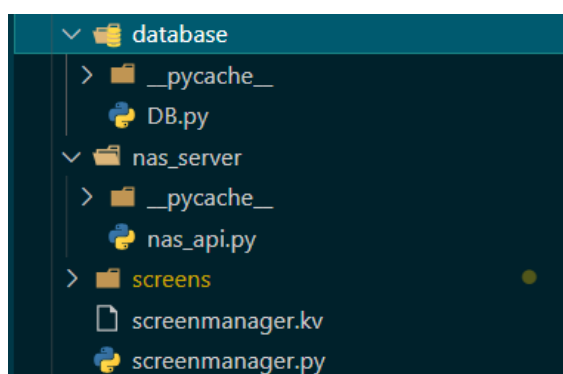


圖 3- 16 資料庫、NAS、UI 頁面管理

3. 挑戰任務相關設計

此設計稍微複雜，每個挑戰都有各個分支，並且也有層級聯繫，先主題→類型→挑戰任務，雖然看似很亂但又有其中的關連性，只要透過特定語法即可將這些資料串起來。

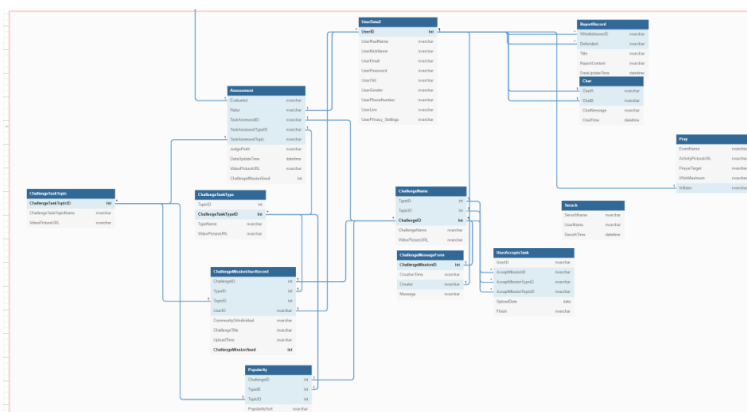


圖 3- 20 挑戰任務關聯圖

3.7 前端與後端 api 串接

利用 Microsoft 提供的 Python pymssql 套件與 MS SQL 做串接，並在資料庫端設定好預存程序(Stored Procedure)，此後只需引用預存程序即可呼叫指定動作。

```
import pymssql as database
```

圖 3- 21pymssql 套件

```
#連接資料庫 server名稱 userID password密碼 編碼方式(基本不改回utf8) 資料庫名稱 port圖
db=database.connect(server='DESKTOP-P72JG8K', user='sa', password='08408449', charset='utf8', database='APPSolutions', port='1433')
#接著利用連結conn圖cursor()
cursor=db.cursor()
```

圖 3- 22DB 串接指令

```
1.GetUserID
#執行SQL SERVER 預存程序Login_Check，並丟入2個參數Email、Password
cursor.execute(f"exec Login_Check @Email=%s,@Password=%s",(useremail,password))
#fetchall取得所有結果列，以陣列或物件方式回傳
results = cursor.fetchall()

2.GetUserNickName
#執行SQL SERVER 預存程序GetUserNickName 抓取使用者NickName
cursor.execute(f"exec GetUserNickName @UserID=%s",(UserID))
#fetchall取得所有結果列，以陣列或物件方式回傳
results = cursor.fetchall()

3.GetUserLevel
#執行SQL SERVER 預存程序GetUserLevel，抓取使用者經驗、等級、活躍度
cursor.execute(f"exec GetUserLevel @UserID=%s",(UserID))
#fetchall取得所有結果列，以陣列或物件方式回傳
results = cursor.fetchall()

4.GetHomePageMissionList
#執行SQL SERVER 預存程序GetHomePageMissionList，抓取首頁推薦的隨機任務
cursor.execute(f"exec GetHomePageMissionList @UserPreferences=%s",(UserID))
#fetchall取得所有結果列，以陣列或物件方式回傳
results = cursor.fetchall()
```

圖 3- 23 圖預存程序呼叫指令

3.8 程式碼打包

3.8.1 建立 Ubuntu 虛擬環境

Step1 下載 Oracle VM VirtualBox

Step2 下載 Ubuntu 20.04.3 映像檔

3.8.2 使用 Buildozer 打包程式(.APK)

- `sudo apt install git`
- `git clone https://github.com/kivy/buildozer.git`
- `cd buildozer`
- `sudo apt-get install python3.9`
- `sudo apt-get install -y python3-setuptools`
- `sudo python3 setup.py install`
- `cd ..`
- (可有可無)`git clone https://github.com/`(此步驟可由 git download 或 抓取自行專案檔案位置放入)
- `cd Challenger`
- `buildozer init` → 產生 `buildozer.spec`
- `sudo apt update`
- `sudo apt install -y git zip unzip openjdk-8-jdk python3-pip autoconf libtool pkg-config zlib1g-dev libncurses5-dev libncursesw5-dev libtinfo5 cmake libffi-dev`
- `pip3 install --user --upgrade cython virtualenv`
- `sudo apt-get install cython`
- #缺少套件 看需要甚麼 `pip3 install` 什麼

3.8.3 設定 buildozer.spec，相關使用到的套件皆須明確填寫

3.8.4 連接手機，手機要開啟開發者模式

- `buildozer android debug deploy run`

3.8.5 程式閃退問題

- 需下載 Android Debug Bridge(adb)，進入手機內部查看 log
- #進入 adb 模式
- `./adb shell`
- #進入 myapp 目錄

- `run-as org.test.myapp`
- `cd files/app/.kivy/logs/`
- `cat ${log 文件名}`

3.9 UI 功能

3.9.1 登入

UserID	UserRealN...	UserNickN...	UserEmail	UserPassword
1	HUANG	Zombie	U07127139@o365.cut.edu.tw	Huang00000

圖 3- 24 使用者資料庫資料

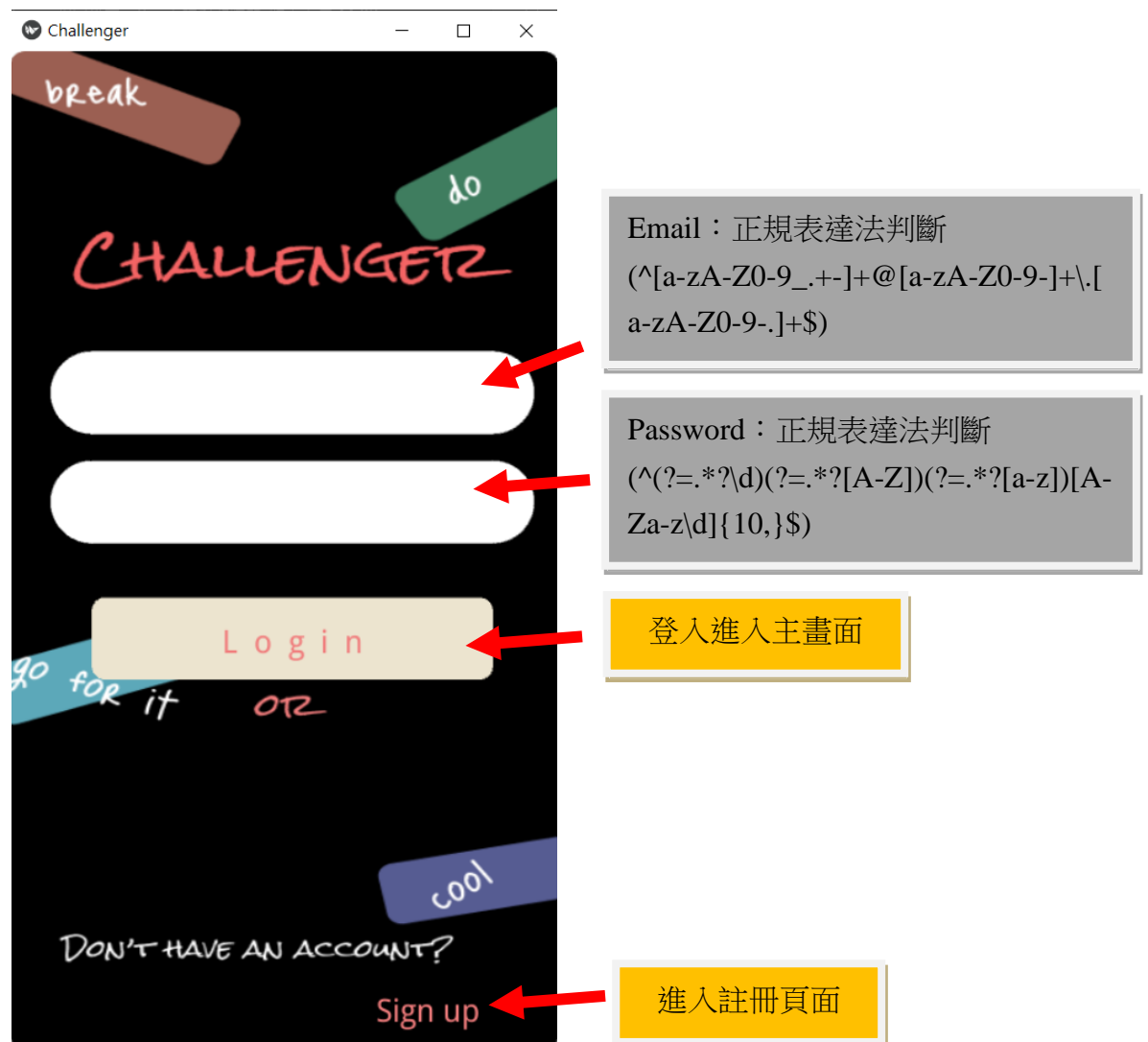


圖 3- 25 登入介面

3.9.2 註冊

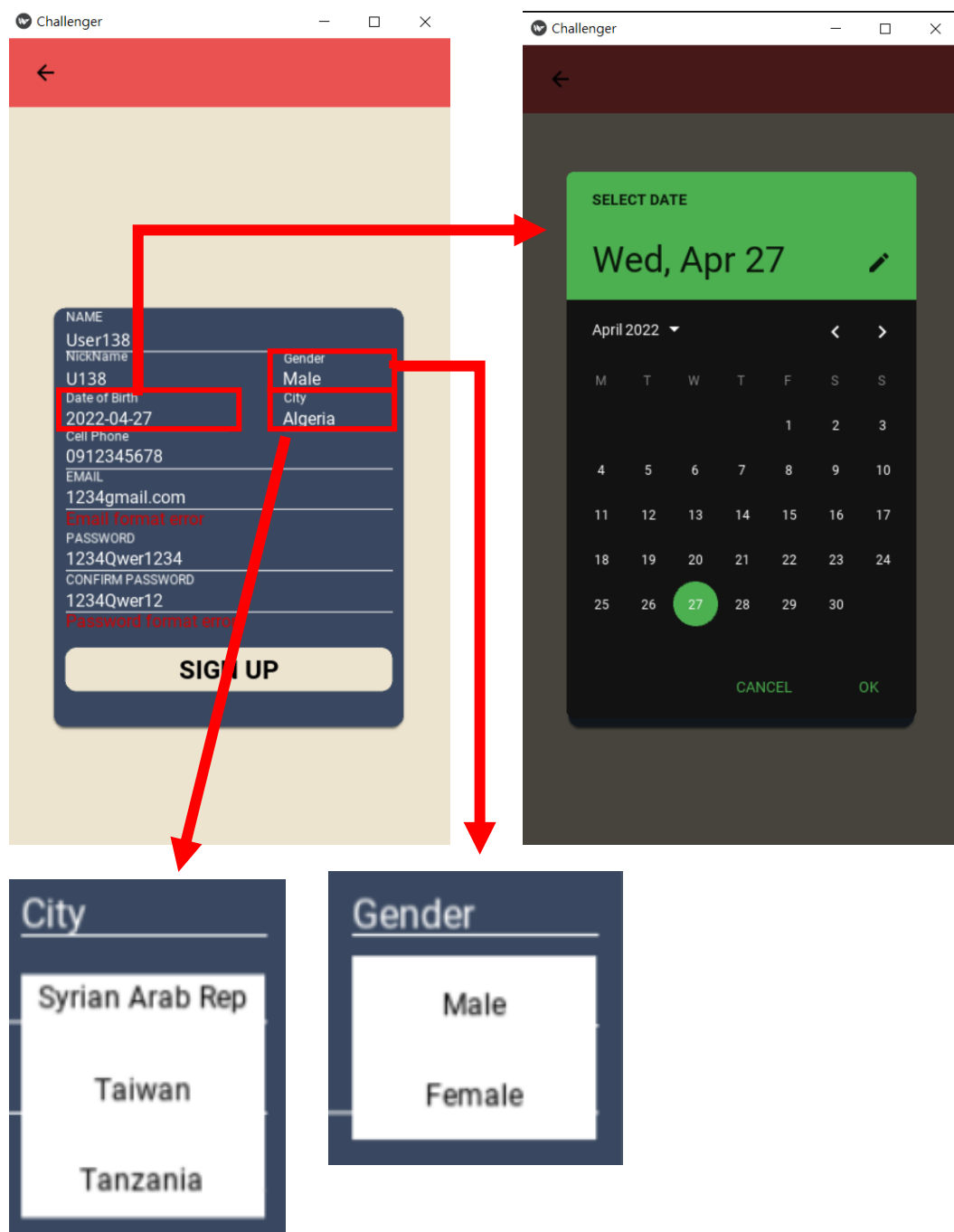


圖 3- 26 註冊介面

3.9.3 接收任務 & 上傳任務

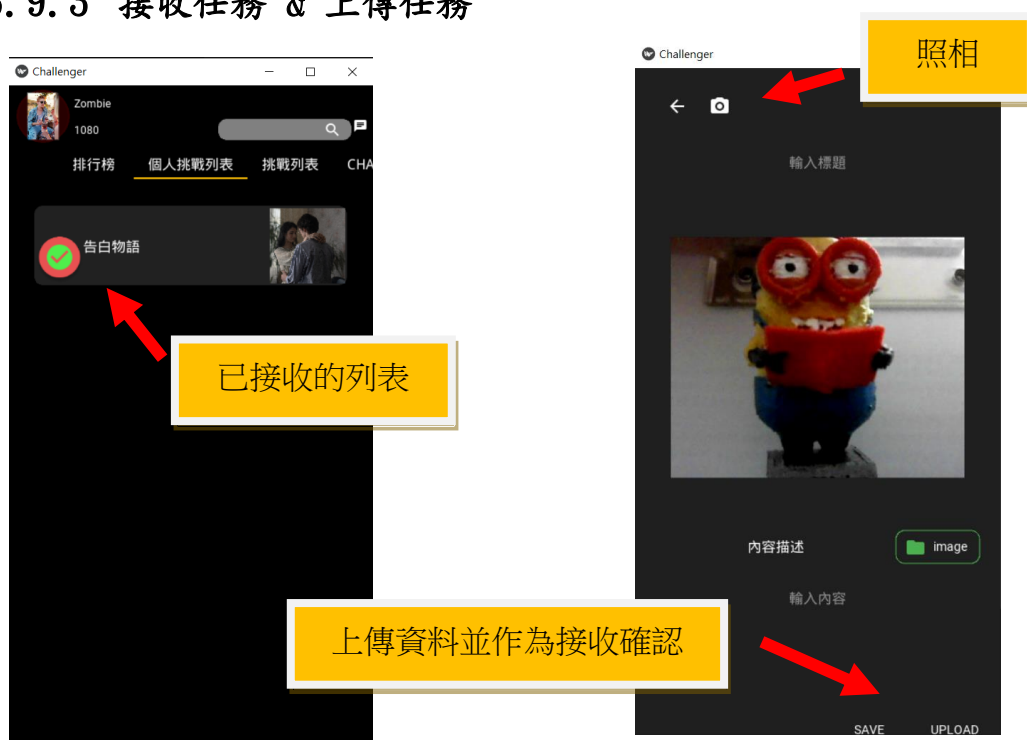


圖 3- 27 接收&上傳介面

3.9.4 發起任務

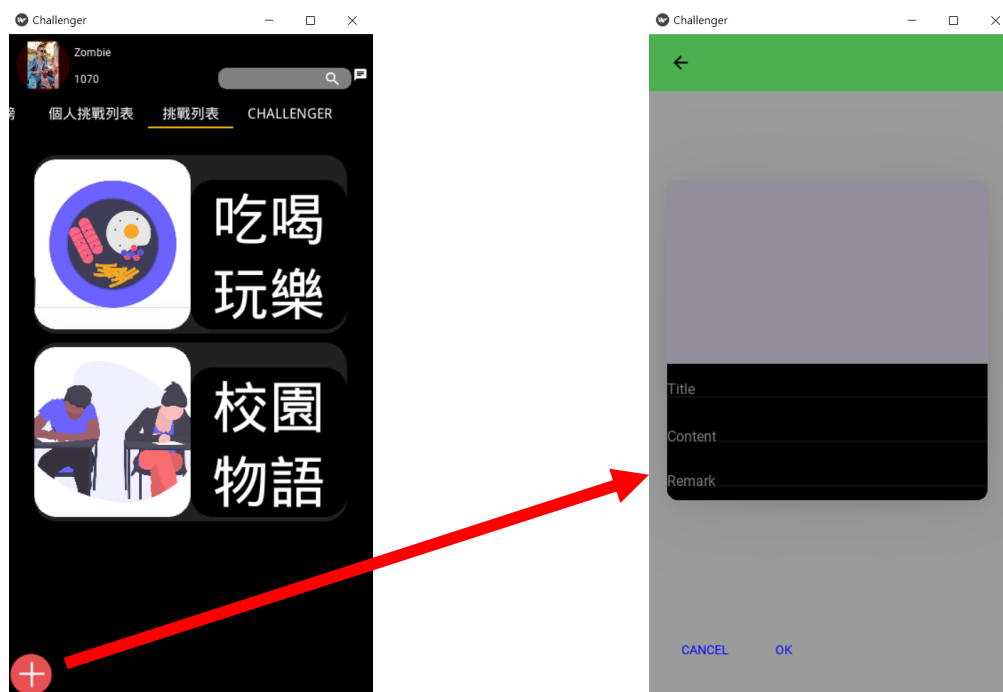


圖 3- 28 發起介面

3.9.5 評核機制

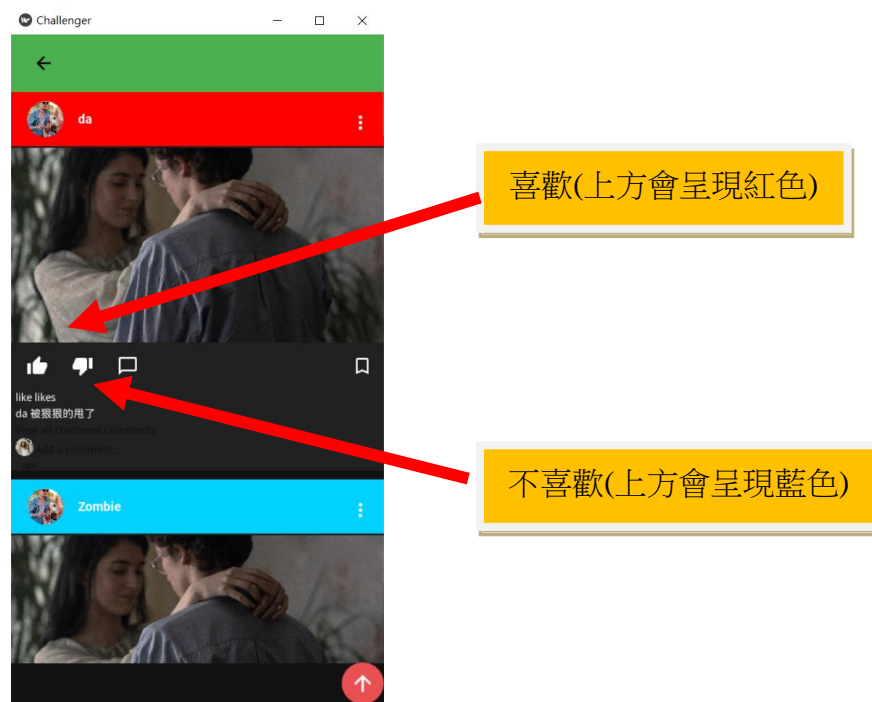


圖 3- 29 評核介面

3.9.6 獎勵機制



圖 3- 30 獎勵介面

第四章 實作成果

4.1 製作成果



圖 4- 1 登入

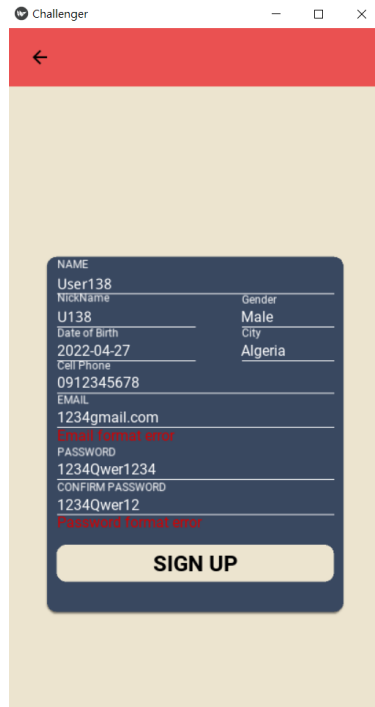


圖 4- 2 註冊



圖 4- 3 主頁

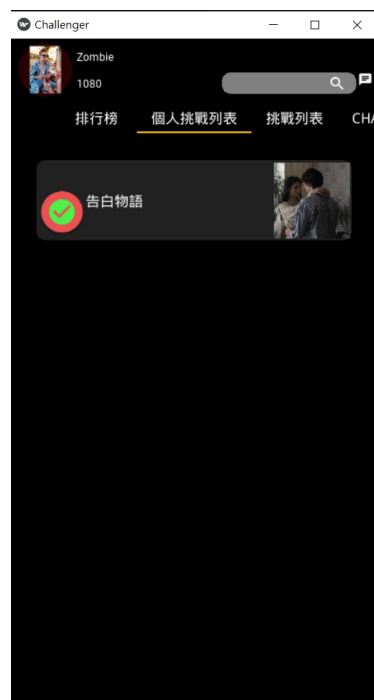


圖 4- 4 挑戰列表



圖 4- 5 挑戰主題



圖 4- 6 推薦挑戰

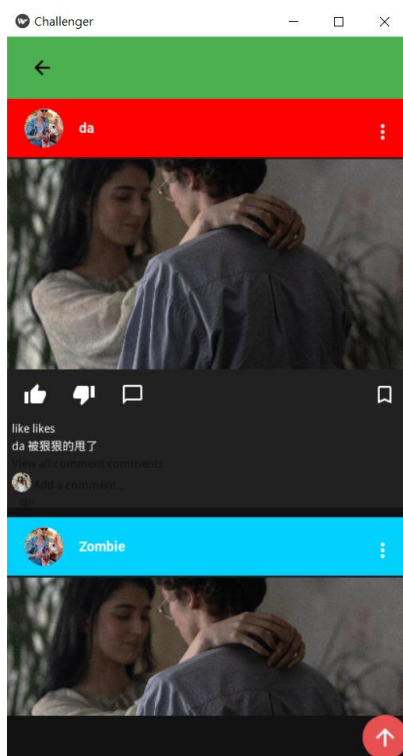


圖 4- 7 用戶貼文

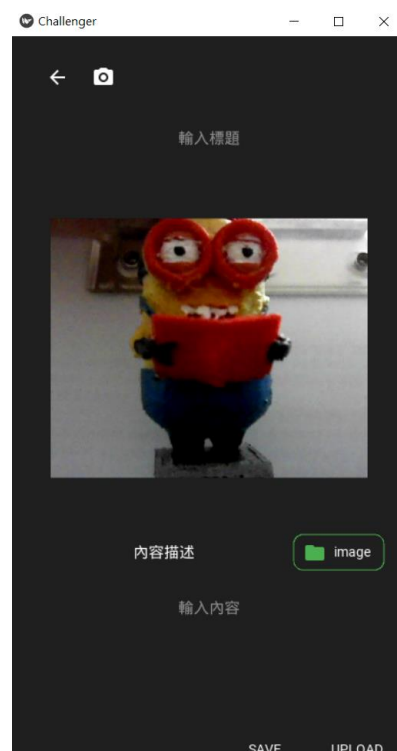


圖 4- 8 上傳表單

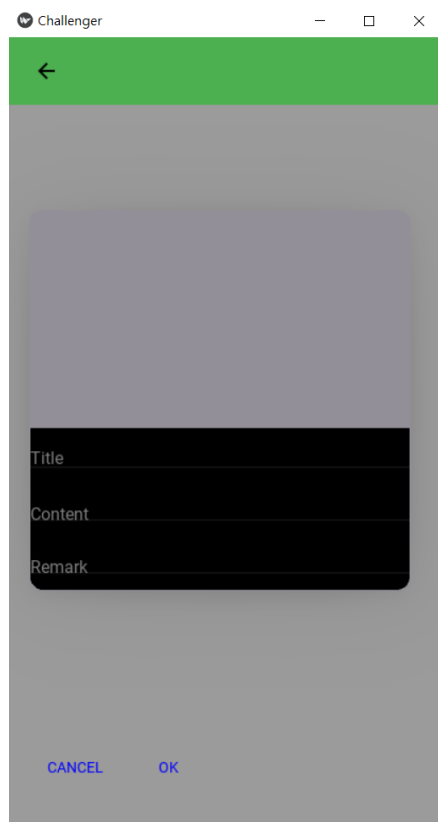


圖 4- 9 新增挑戰

第五章 結論

現在想想還是非常感慨萬千，從零到完成作品的路途上伴隨著無數個大小問題，無論是大問題或者是小問題每個都足以困擾我們好幾天，這幾天唯一能做的就是上網或者從書上找解答，印象最深刻的莫過於建置伺服器和網路串接，因為我們團隊沒有一位有這種經驗所以頻頻遇到障礙，為了更進一步理解架構甚至購買了 NAS 來當作伺服器，但真正的問題現在才開始。

我們試著安裝 Docker 並在其他建置 MS SQL，但第一步就遇到資料儲存的問題，後續更是引起權限的不足導致無法啟用部分功能 or 套件。VPN 對外串接更是頭痛，速度提不上去造成負面的操作感等等..。在我們團隊的合作努力下一一克服了所有困難，雖然這一路上走來非常坎坷，但也要謝謝有遇到這些問題，這些問題幫助我們更進一步地成長，無論是在心境或者知識上都帶給我們無可取代的經驗，此外非常感謝邱機平教授悉心教導，讓我們順利完成專題。

參考文獻

1. [Python] 與 MSSQL 資料庫做串接與應用 @ K_程式人 :: 痞客邦 ::
(pixnet.net)
2. 【免安裝】線上流程圖繪製軟體，開發模型、專案架構圖，Draw.io 免費製作程式。 | 痞凱踏踏 | PKstep
3. Docker 基礎教學與介紹 101. 何謂容器虛擬化、介紹 Docker 三元素、手把手建立 Docker... | by Cheng-Wei Hu | 胡程維 | Medium
4. 程式設計師如何畫好架構圖？ (uetie.com)
5. How enable SQL Server Agent in docker container existing - Stack Overflow
6. 免費圖庫相片 · Pexels
7. Noto Home - Google Fonts
8. UI 設計師要如何出 App icon 的圖給 iOS 及 Android 系統呢？ | by Joyceh | Medium
9. Humaaans: Mix-&-Match illustration library
10. Open Doodles - Free sketchy illustrations
11. Material Design Icons
12. Color Tool - Material Design
13. mtu 值怎樣設定才網速最快 - 程式人生 (796t.com)
14. Upgrade Python to latest version (3.10) on Ubuntu Linux
(cloudbytes.dev)
15. Android adb 基本用法教學 | ShengYu Talk (shengyu7697.github.io)
16. GitHub - subhranil2699/instagram_clone
17. python 下載/上載文件到 Synology 群暉_容沁風的博客-CSDN 博客
18. Mobile App Architecture Guide for 2022 | Net Solutions

19. [Need a Faster Mobile App? 8 Ways to Optimize App Server Performance \(moveoapps.com\)](#)
20. [\[原來後端要知道\] 什麼是前後端分離? #前端後端是什麼 #軟體發展歷程 #軟體開發思維 - iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天 \(ithome.com.tw\)](#)
21. [Widget interactions between Python and Kv \(inclem.net\)](#)
22. [GitHub - Jamie-/openvpn-api: A Python API for the OpenVPN management interface.](#)
23. [通過 Py2exe 將自己的 python 程式打包成 .exe/.app 的方法 - IT 閱讀 \(itread01.com\)](#)
24. [python 打包 apk 闪退问题以及美化_python__reported 的博客-CSDN 博客_kivy 打包后闪退](#)
25. [Kivy Base — Kivy 2.1.0 documentation](#)
26. [Bottom Navigation — KivyMD 0.104.2 documentation](#)
27. [python - No module named 'picamera' - Stack Overflow](#)
28. [How to install android module? : learnpython \(reddit.com\)](#)
29. [kivy/camera_android.py at master · kivy/kivy · GitHub](#)
30. [彻底解决 INSTALL_FAILED_UPDATE_INCOMPATIBLE 的安装错误、安装包与之前设备上的安装包签名不一致_vickyleexy 的博客-CSDN 博客](#)
31. [\[Termux\] ImportError with dlopen failed · Issue #4355 · python-pillow/Pillow · GitHub](#)
32. [Android 圖片儲存到指定路徑和相簿的方法 - IT 閱讀 \(itread01.com\)](#)