**國立勤益科技大學學生專題成果報告**

運動紀錄系統

執 行 期 間 ：　113　學年度下學期

專題參與人員：

部別：進修部

組別：第 三 組

班別：甲班 學號：9B117007 姓名：張雋宏

甲班 學號：9B117006 姓名：劉雲頂

甲班 學號：9B117015 姓名：鍾曜宇

指導老師：朱孝國 講師

中　華　民　國　　114　 年　　6 　月　　13　 日

**目次**

第壹章、 專題動機

1.1前言

1.2響應式網頁設計(Responsive Web Design)

1.3研究目標

第貳章、 背景資料：

2.1相關系統或文獻介紹與比較

2.2 Python的相關套件

2.3 運動數據的依據

第參章、 系統架構

第肆章、 系統環境與開發工具

第伍章、 實作方法

5.1 資料流程圖 Data Flow Diagram

5.2 ER Model

5.3 流程圖

第柒章、 實作心得

第玖章、 附註

**圖目次**

圖 1 系統架構圖

圖 2 Bootstrap

圖 3 Flask

圖 4 SQLite

圖 5 DFD圖

圖 6 ERD圖

圖 7 流程圖

**表目次**

表 1 專案差異比較

表 2 各類運動的卡路里消耗數據

表 3 Django與Flask後端工具比較

表 4 SQLite與MySQL資料庫比較

第壹章、 專題動機

**1.1前言**

在現代生活中，運動已經成為許多人日常的一部分，但要有效追蹤運動成果並保持進步，卻不是件容易的事。這個專題的核心理念，是打造一個能夠 主動記錄運動時間 的系統，並透過數據分析和視覺化，讓使用者可以直覺地看到自己的進步、調整訓練節奏，甚至追蹤減重成果。

**1.2響應式網頁設計(Responsive Web Design)**

響應式網頁設計（Responsive Web Design，簡稱 RWD）是一種網頁設計方法，旨在使網頁在不同設備和螢幕尺寸上都有良好的顯示效果，無論是桌上型電腦、筆記本電腦、平板電腦還是智能手機。RWD 的主要目標是提高使用者體驗，確保網頁在不同設備上都能自適應調整佈局和內容。

**優點：**

1. **跨設備適應性**：

RWD 能夠確保網站在各種設備和螢幕尺寸上都有良好的顯示效果，提供一致的使用者體驗。

1. **提高 SEO 效果**：

使用 RWD 設計的網站對搜索引擎友好，有助於提高搜索引擎排名，因為搜索引擎喜歡具有一致且優質內容的網站。

1. **減少維護成本**：

只需要一個版本的網站，可以減少設計、開發和維護多個版本的成本和時間。

1. **未來適應能力**：

RWD 的彈性設計使網站能夠應對未來新設備和螢幕尺寸的出現，無需重新設計。

**缺點：**

1. **初期設計成本較高**：

RWD 需要在設計和開發初期投入更多的時間和資源來確保所有元素在不同設備上都能良好顯示。

1. **性能問題**：

如果沒有妥善優化，RWD 可能會導致加載時間變長，特別是圖片和視頻內容較多時。

1. **複雜性**：

設計和開發 RWD 需要較高的技術水準，對於一些小型團隊或個人開發者來說，可能會面臨較大的挑戰。

1. **兼容性問題**：

需要考慮到不同瀏覽器和設備的兼容性，有時需要特別的技巧來處理某些設備或瀏覽器的問題。

**1.3研究目標**

1. 開發一個 **直覺且功能完善的運動記錄網站**，透過 **主動記錄運動時長** 與 **數據視覺化分析**，幫助使用者追蹤 **減重與運動表現**，並依據數據調整訓練策略，以提升運動的效率與持續性。

第貳章、 背景資料：

**2.1相關系統或文獻介紹與比較**

表 1 專案差異比較

|  | Health App | 本系統專案 |
| --- | --- | --- |
| 運動類型支援 | 基本運動（步數、跑步、游泳等） | 多種運動類型（跑步、騎行、游泳、健身等） |
| 運動數據記錄 | 距離、卡路里、心率等 | 距離、卡路里、心率 |
| 數據可視化 | 基本趨勢分析 | 多種圖表 |
| 目標設定與達成 | 健康趨勢追蹤 | 自訂運動目標（如每月跑100公里） |
| 社交互動 | 家人共享健康數據 | 好友 ID 搜尋競賽、社群挑戰 |
| 平台限制 | 僅限 iOS & Apple 設備 | 跨裝置支援（手機、平板、電腦） |

隨著運動科技的發展，越來越多使用者希望透過數據分析來提升訓練效率、達成健康目標。然而，現有的運動紀錄系統，如 iPhone Health App，雖然能夠提供基礎數據，但在運動目標設定、社交互動、以及裝置支援方面仍有一定限制。本專案則針對這些痛點進行優化，打造出一套 更具靈活性與互動性 的運動紀錄系統。

**2.1.1核心特色**

本專案不僅能記錄多種運動數據，還提供 自訂運動目標，讓使用者能根據自身需求設立挑戰，確保運動方向明確。此外，還能 加入好友並與他們比較運動表現，透過競爭與激勵機制，提高使用者的自律性與持續性。

**2..1.2跨裝置支援**

不同於 iPhone Health App 受限於特定裝置，本專案 不受設備限制，只要有網路，就能在 手機、平板或電腦 上使用，極大提升了系統的便利性與可及性。

**2.1.3數據準確度與限制**

本專案的數據來源主要來自 開放資料推測，因此在精確度上可能 不如 Health App 直接從穿戴式裝置獲取數據。目前系統 無法與智慧手錶或其他穿戴設備整合，因此在即時心率監測等功能上略顯不足。

**2.2 Python的相關套件**

**flask：**

* + **Flask：建立 Flask 應用主體。**
  + **render\_template：渲染 HTML 模板（Jinja2）。**
  + **request：處理來自使用者的請求（GET、POST）。**
  + **redirect：導向其他路由。**
  + **url\_for：生成路由的 URL。**
  + **session：使用者的 session 管理。**
  + **jsonify：返回 JSON 格式的資料（常用於 API）。**
  + **flash：顯示一次性的提示訊息（通常用於登入、註冊等操作結果）。**

**flask\_sqlalchemy：**

* + **SQLAlchemy：Flask 整合的 ORM 工具，用來與資料庫互動（例如新增、查詢、修改資料等）。**

**werkzeug.security：**

* + **generate\_password\_hash：將密碼轉成雜湊值（避免明文儲存）。**
  + **check\_password\_hash：檢查使用者輸入的密碼是否與儲存的雜湊值相符。**

**datetime：**

* + **datetime, timedelta：處理時間與日期（例如計算使用者登入時間、文章發布時間等）。**

**sqlalchemy：**

* + **func：SQLAlchemy 的內建函數（如 func.count()、func.now() 等，可生成 SQL 語法中常見的函數）。**

**json：**

* + **用來處理 JSON 資料的編碼與解碼。**

**os：**

* + **處理作業系統相關的功能，例如讀取環境變數、檔案路徑等。**

**2.3 運動數據的依據**

依據參考了政府開放資料平台中-各類運動的卡路里消耗數據

表 2 各類運動的卡路里消耗數據

| 運動項目 | 消耗熱量(大卡/公斤體重/小時) | 40公斤運動30分鐘所消耗的熱量 (大卡) | 50公斤運動30分鐘所消耗的熱量 (大卡) | 60公斤運動30分鐘所消耗的熱量 (大卡) | 70公斤運動30分鐘所消耗的熱量 (大卡) |
| --- | --- | --- | --- | --- | --- |
| 慢走(4公里/小時) | 3.5 | 70 | 87.5 | 105 | 122.5 |
| 快走、健走(6.0公里/小時) | 5.5 | 110 | 137.5 | 165 | 192.5 |
| 下樓梯 | 3.2 | 64 | 80 | 96 | 112 |
| 上樓梯 | 8.4 | 168 | 210 | 252 | 294 |
| 慢跑(8公里/小時) | 8.2 | 164 | 205 | 246 | 287 |
| 快跑(12公里/小時) | 12.7 | 254 | 317.5 | 381 | 444.5 |
| 快跑(16公里/小時) | 16.8 | 336 | 420 | 504 | 588 |
| 騎腳踏車(一般速度，10公里/小時) | 4 | 80 | 100 | 120 | 140 |
| 騎腳踏車(快，20公里/小時) | 8.4 | 168 | 210 | 252 | 294 |
| 騎腳踏車(很快，30公里/小時) | 12.6 | 252 | 315 | 378 | 441 |
| 拖地 | 3.7 | 74 | 92.5 | 111 | 129.5 |
| 園藝 | 4.2 | 84 | 105 | 126 | 147 |
| 使用工具製造或修理(如水電工) | 5.3 | 106 | 132.5 | 159 | 185.5 |
| 耕種、牧場、漁業、林業 | 7.4 | 148 | 185 | 222 | 259 |
| 搬運重物 | 8.4 | 168 | 210 | 252 | 294 |
| 瑜珈 | 3 | 60 | 75 | 90 | 105 |
| 跳舞(慢)、元極舞 | 3.1 | 62 | 77.5 | 93 | 108.5 |
| 跳舞(快)、國際標準舞 | 5.3 | 106 | 132.5 | 159 | 185.5 |
| 飛盤 | 3.2 | 64 | 80 | 96 | 112 |
| 排球 | 3.6 | 72 | 90 | 108 | 126 |
| 保齡球 | 3.6 | 72 | 90 | 108 | 126 |
| 太極拳 | 4.2 | 84 | 105 | 126 | 147 |
| 乒乓球 | 4.2 | 84 | 105 | 126 | 147 |
| 棒壘球 | 4.7 | 94 | 117.5 | 141 | 164.5 |
| 高爾夫 | 5 | 100 | 125 | 150 | 175 |
| 溜直排輪 | 5.1 | 102 | 127.5 | 153 | 178.5 |
| 羽毛球 | 5.1 | 102 | 127.5 | 153 | 178.5 |
| 游泳(慢) | 6.3 | 126 | 157.5 | 189 | 220.5 |
| 游泳(較快) | 10 | 200 | 250 | 300 | 350 |
| 籃球(半場) | 6.3 | 126 | 157.5 | 189 | 220.5 |
| 籃球(全場) | 8.3 | 166 | 207.5 | 249 | 290.5 |
| 有氧舞蹈 | 6.8 | 136 | 170 | 204 | 238 |
| 網球 | 6.6 | 132 | 165 | 198 | 231 |
| 足球 | 7.7 | 154 | 192.5 | 231 | 269.5 |
| 跳繩(慢) | 8.4 | 168 | 210 | 252 | 294 |
| 跳繩(快) | 12.6 | 252 | 315 | 378 | 441 |

第參章、 系統架構

**3.1系統架構圖**

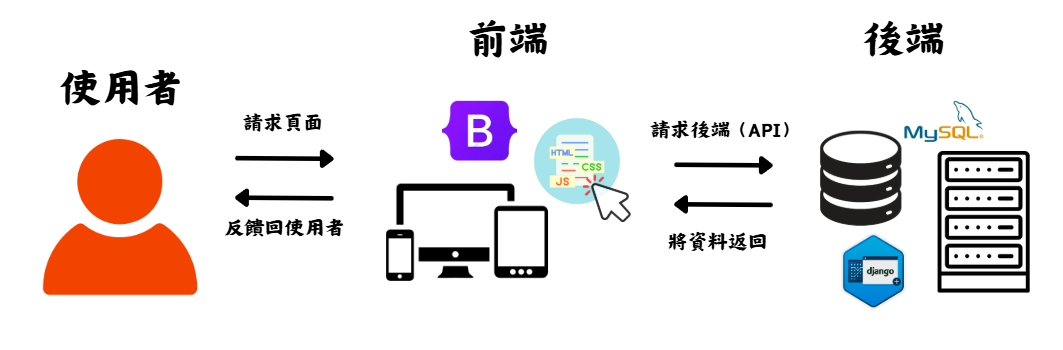


圖 1 系統架構圖

**3.1.1前端部分**

前端是用戶與系統交互的窗口，我採用了Bootstrap框架和HTML/CSS/JS技術，設計了多個UI組件，包括新增運動紀錄、運動統計、趨勢分析和歷史紀錄查詢等功能。這些設計讓使用者能夠方便地操作我的網站，不管是手機、電腦還是平板上都能有良好的使用體驗。

**3.1.2後端部分**

後端功能處理是整個系統的心臟，包括搜尋好友功能、帳號管理功能等，這些功能處理與前端UI緊密結合，確保數據的準確傳遞與處理。我使用了Flask框架來實現這些後端功能，並且依賴穩定的伺服器來支持系統的運行。

**3.1.3資料庫**

資料庫是系統的數據倉庫，我們選擇了SQLite來存儲用戶信息、運動紀錄等，我設計了多個資料表來組織和管理數據，這些表之間通過外鍵、外鍵關係進行聯結，確保數據的一致性和完整性。

第肆章、 系統環境與開發工具



圖 2 Bootstrap

**4.1前端Bootstrap 框架**

Bootstrap 是一個基於HTML、CSS 和 JavaScript 技術，所開發的開源前端框架。提供了大量的預設樣式和組件，如按鈕、導航欄、彈窗等，並且有內建響應式網格系統，讓網站在不同大小的螢幕上都能有良好的顯示效果，再加上擁有龐大的社區和豐富的資源，可以使開發者快速搭建網站。



圖 3 Flask

**4.2後端Flask框架**

Flask 是一個使用Python 語言的輕量級Web 應用框架，可以幫助開發者快速建立安全且可擴展的網頁應用，而且基於Python語言，擁有龐大的第三方庫和工具，讓應用具備極高的擴展性和靈活性，非常適合小型到中型的專案。

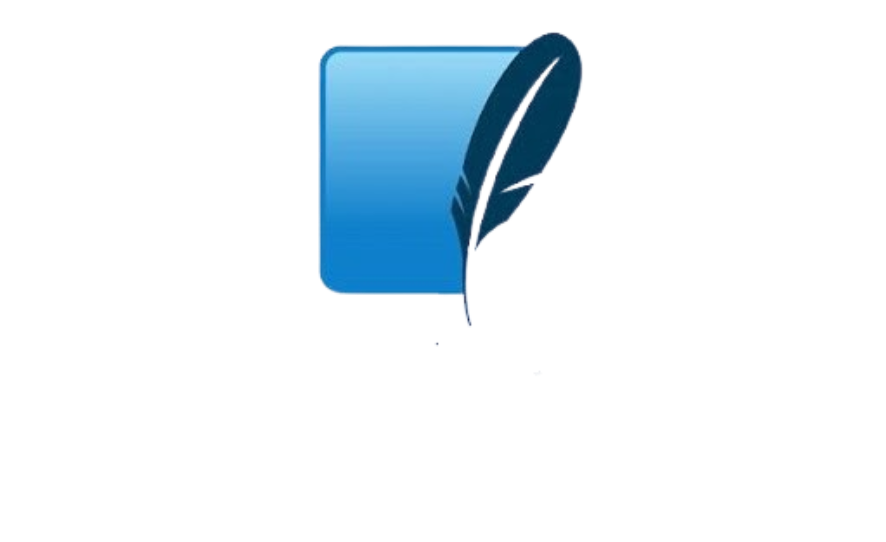


圖 4 SQLite

**4.3資料庫 SQLite**

**SQLite** 是一個開源的關聯型數據庫管理系統，使用結構化查詢語言 (SQL) 來管理數據。它具備高效能、可靠性和靈活性，廣泛應用於各類網頁和應用程式中，能夠處理從小型至中型網頁應用的數據需求。無論是個人專案還是大型網站，**SQLite**都能提供穩定的性能。

**4.4 Django與Flask串接工具比較**

表 3 Django與Flask後端工具比較

| 特性 | Django | Flask |
| --- | --- | --- |
| 框架類型 | 全功能框架 | 輕量級微框架 |
| 適用場景 | 大型應用、企業級系統 | 小型應用、API、原型開發 |
| 內建功能 | 內建 ORM、管理介面、身份驗證、表單處理 | 僅提供基本路由與請求處理 |
| 擴展性 | 適合大型專案，內建許多功能 | 適合小型專案，可自由選擇擴展 |
| 性能 | 較重，適合複雜應用 | 較輕量，適合高效 API |

**4.4.1 Django**

Django 是一款 強大的 Python 網頁框架，它讓開發者可以 快速搭建安全、可擴展 的網站。採用 MTV 架構（模型-模板-視圖），能夠有效分離資料處理、介面設計和邏輯運作，讓開發更有條理，最大的優勢是 擁有豐富的內建功能，像是 身份驗證、管理介面、資料庫處理 都是現成的，不需要額外安裝太多套件。

表 4 SQLite與MySQL資料庫比較

| 特性 | Django | Flask |
| --- | --- | --- |
| 架構 | 無伺服器，所有數據存儲在單一檔案中 | 客戶端-伺服器架構，需要獨立的數據庫伺服器 |
| 適用場景 | 小型應用、個人專案 | 大型網站、企業級應用系統 |
| 性能 | 輕量級，適合小型數據處理 | 高效能，適合大規模數據處理 |
| 安全性 | 較簡單，無內建使用者管理 | 內建使用者管理與權限控制 |
| 安裝與管理 | 無需額外安裝，直接嵌入應用程式 | 需要伺服器安裝與管理 |

**4.4.2 MySQL**

MySQL 是一款 開源的關聯型數據庫管理系統（RDBMS），使用 SQL（結構化查詢語言） 來管理和操作數據。它以 高效能、可靠性、可擴展性 著稱，並且廣泛應用於各類網站、應用程式和企業級系統。

第伍章、 實作方法

**5.1 資料流程圖 Data Flow Diagram**

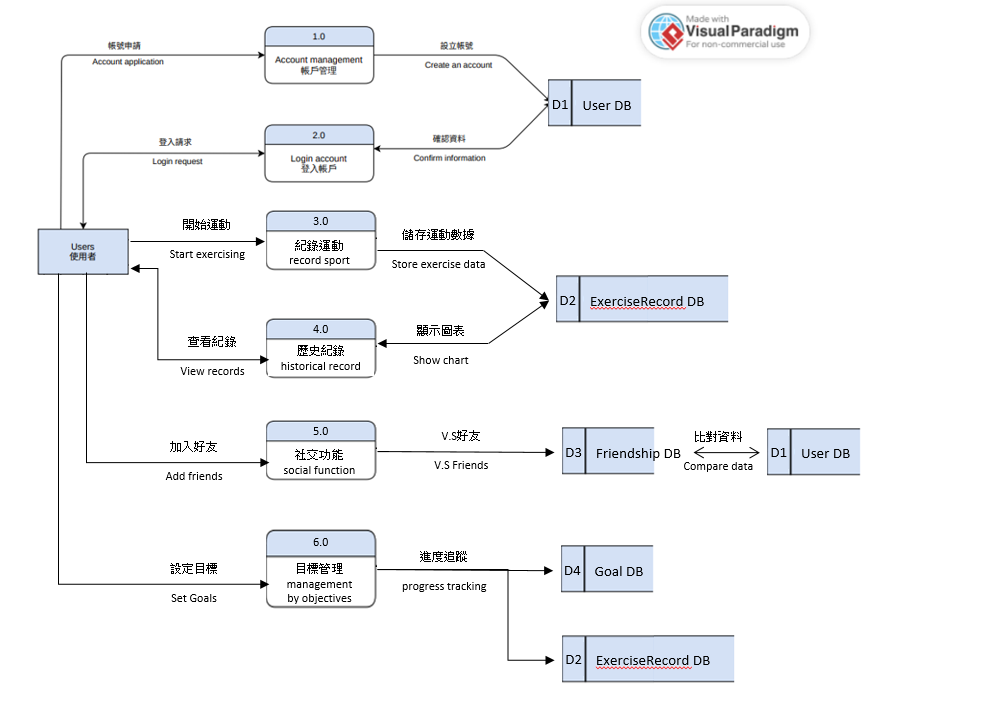


圖 5 DFD圖

**過程分解 (Process Decomposition) – 第 1 級**

**5.1.1 帳戶管理 (1.0)**

* **帳號申請 (Account application)：** 使用者透過介面發送帳號申請。
* **設立帳號 (Create an account)：** 系統接收申請，並在 User DB (D1) 中建立新帳號。

**5.1.2 登入帳戶 (2.0)**

* **登入請求 (Login request)：** 使用者輸入帳號密碼並送出登入請求。
* **確認資料 (Confirm information)：** 系統向 User DB (D1) 查詢並驗證使用者身份。

**5.1.3 紀錄運動 (3.0)**

* **開始運動 (Start exercising)：** 使用者開始運動並自行記錄。
* **儲存運動數據 (Store exercise data)：** 系統將即時運動資料寫入 ExerciseRecord DB (D2)。

**5.1.4 歷史紀錄 (4.0)**

* **查看紀錄 (View records)：** 使用者請求檢視過往運動紀錄。
* **顯示回表 (Display historical data)：** 系統從 ExerciseRecord DB (D2) 撈取資料並以圖表或列表方式呈現。

**5.1.5 社交功能 (5.0)**

* **加入好友 (Add friends)：** 使用者發送加好友請求。
* **建立關係 (V.S Friends)：** 系統將好友關係寫入 Friendship DB (D3)，並參照 User DB (D1) 比對使用者資料。

**5.1.6 目標管理 (6.0)**

* **設定目標 (Set goals)：** 使用者在介面上設定運動目標，系統將目標記錄至 Goal DB (D4)。
* **進度追蹤 (Progress tracking)：** 系統自 ExerciseRecord DB (D2) 讀取運動紀錄，與 Goal DB (D4) 中的目標比對，並更新達成進度。

**5.2 ER Model**



圖 6 ERD圖

### 5.2.1 使用者（Users）

### Users 表用來儲存所有註冊使用者的基本資料。

* **主鍵**：id
* **username（用戶名）**：唯一鍵，用於識別與登入
* **password\_hash（加密密碼）**：雜湊後的密碼，確保帳戶安全
* **weight（體重）**：紀錄使用者當前體重，以便進行目標追蹤與消耗估算
* **created\_at（註冊時間）**：帳號建立時間，方便後續管理與統計

### 5.2.2 運動紀錄（ExerciseRecord）

ExerciseRecord 表記錄使用者每次運動的詳細資訊。

* + **主鍵**：id
  + **user\_id（使用者ID）**：外鍵，對應 Users.id，用於關聯該筆紀錄屬於哪位使用者
  + **exercise\_type（運動類型）**：例如「跑步」、「游泳」、「重訓」等
  + **duration（運動時間）**：以分鐘為單位
  + **heart\_rate（心率）**：平均心率，可選欄位
  + **intensity（運動強度）**：如「低」「中」「高」
  + **notes（備註）**：使用者補充說明
  + **calories（消耗熱量）**：大卡為單位，估算運動消耗
  + **created\_at（記錄時間）**：該筆運動何時被寫入系統

### 5.2.3 目標（Goal）

Goal 表管理使用者為自己設定的運動目標，以及完成狀態。

* + **主鍵**：id
  + **user\_id（使用者ID）**：外鍵，對應 Users.id，標記目標的擁有者
  + **goal\_type（目標類型）**：例如「距離」、「時長」、「熱量」
  + **goal\_value（目標值）**：具體數值，如 5 km、30 min、500 kcal
  + **period（週期）**：目標週期，例如「每日」「每週」「每月」
  + **name（目標名稱）**：使用者自訂的易識別名稱
  + **completed（是否完成）**：布林值（true/false）
  + **created\_at（創建時間）**：該目標建立於何時

### 5.2.4 好友關係（Friendship）

Friendship 表用以管理使用者間的社交連結。

* **主鍵**：id
* **user\_id（使用者ID）**：外鍵，對應 Users.id，代表加好友的發起者
* **friend\_id（好友ID）**：外鍵，對應 Users.id，代表被加為好友的對象
* **created\_at（建立時間）**：紀錄好友關係何時形成

**5.3 流程圖**

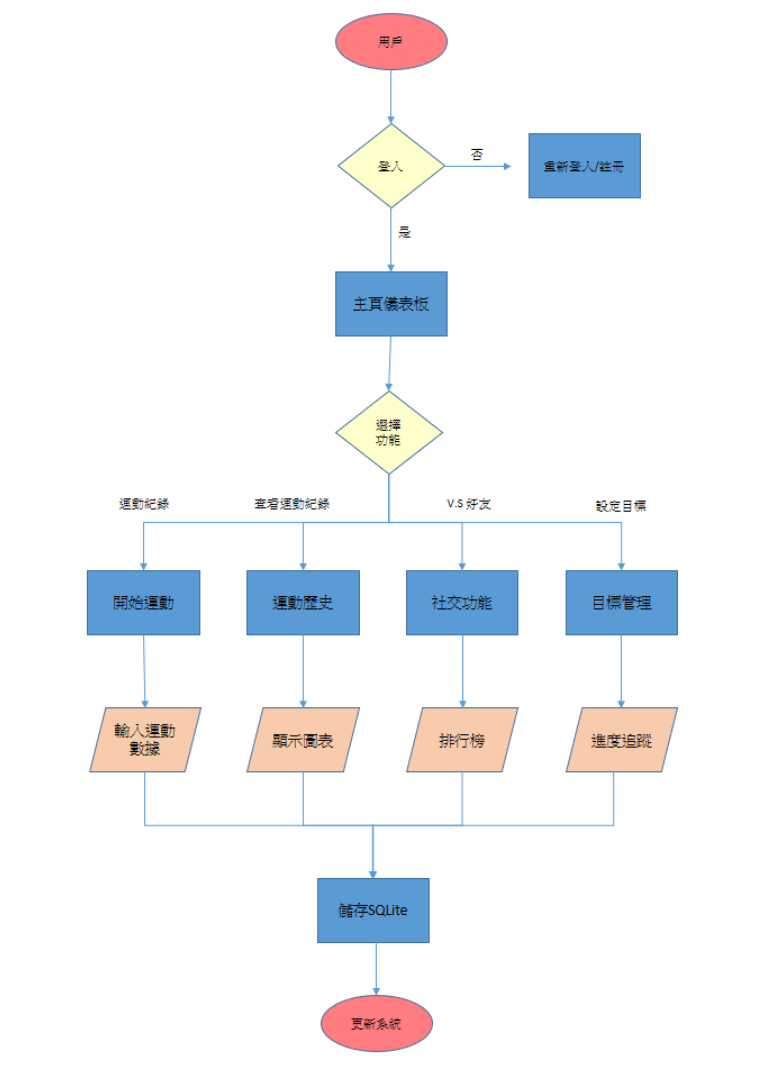


圖 7 流程圖

**5.3.1系統概述**

本系統為一個運動紀錄與目標管理平台，整合四大功能模組：

1. 運動紀錄輸入
2. 歷史運動圖表
3. 社交排行榜
4. 運動目標設定與追蹤

系統以 SQLite 作為資料儲存核心，並透過使用者登入維持資料安全性與個人化操作。

## 5.3.2使用者流程說明

### 登入驗證階段

* 使用者進入系統界面。
* 系統進行登入狀態判斷：
  + **已登入** → 導入主頁／儀表板。
  + **未登入** → 顯示登入或註冊頁面，完成後重新導向主頁。

### 5.3.3 儀表板（主頁）功能選擇

使用者登入後進入儀表板，可以快速查看：

* 運動儀錶板
* 運動趨勢
* 最近運動

## 5.3.4 功能模組與資料流程

各功能模組最終都將操作結果儲存至 SQLite，並觸發系統更新流程：

### 3.1 開始運動

* 使用者點選「開始運動」功能。
* 系統跳轉至「輸入運動數據」介面。
* 使用者完成輸入後，資料送至 SQLite 資料庫。

### 3.2 查看運動歷史

* 使用者在儀表板選擇「數據分析」。
* 系統從 SQLite 提取相關歷史紀錄。
* 將資料透過圖表視覺化呈現。

### 3.3 社交功能（VS 好友）

* 使用者選擇「社交功能」。
* 系統抓取使用者與好友的運動數據，計算排行順序。
* 呈現排行榜介面，顯示個人分數或總量排行。

### 3.4 目標管理

* 使用者點選「目標管理」功能，進入「進度追蹤」介面。
* 使用者設定運動目標（例如每週跑步 20 公里、燃燒 X 卡），並設定時間、目標名稱。
* 系統根據使用者的歷史數據（來自 SQLite）計算達成百分比／比例。
* 進度與完成度視覺化呈現，幫助使用者追蹤成果。

第柒章、 實作心得

**7.1 張雋宏**

在這次專題中，我首次負責前端開發過程中遇到不少挑戰。從版面配置到資料串接，我透過查閱資料和與組員討論慢慢完成任務。這段經歷讓我學會前端技術與團隊合作的重要性。雖然過程不易但看到成果非常值得。

**7.2 劉雲頂**

這次專題我主要是負責簡報和文件的部分，不過為了讓整體方向更清楚，一開始我就先規劃出我們系統應該要有的功能，像是有哪些主要的操作流程、使用者需要什麼等等。除此之外，我也先做了一個前後端的雛形框架，方便後續負責程式的同學能夠依照這個架構去開發，減少來回修改的時間。雖然我不是主要負責寫程式的那位，但我還是盡力在前期協助定好架構跟方向，讓整個專題可以比較順利地進行，也方便大家之後做功能開發跟整合。

**7.3 鍾曜宇**

在這次的專題製作上，我主要負責網頁後端製作與開發，開發我是使用Flask搭配SQL來製作，主要在端點的設計、架構與回應格式的設計與定義、錯誤處理的設計與實作，在後端開發時我遇到很多的問題都是出在API、資料格式不一致、不過也讓我在這次實作中更加熟悉後端開發的流程，也增強我對於API設計和即時錯誤回應對整體專案進展重要性的͏理解͏。

透過這次做͏，我對網頁後端的發展有了更深的了解，也學會在面對複雜問題時，如何冷靜的分析和有效解決。還有，我也很感謝組員們在過程中協助與討論，讓我能順利完成͏負責的一部分並從中得到寶貴的經驗.

第玖章、 附註

**8.1 參考文獻**

(1) 各類運動消耗熱量表

取自<https://data.gov.tw/dataset/8494>

**8.2 工作**

以下為初步分工方案，前後端的部份大家都彼此交叉合作、共同涉略。

**張雋宏：**前端設計

**鍾曜宇：**後端設計

**劉雲頂：**簡報、文件