國立臺北商業大學

資 訊 管 理 系

113’資訊系統專案設計

**系統手冊**

一張含有 圖表, 設計 的圖片

自動產生的描述

**組 別：第113502組**

**題 目： CodeAlchemy系統**

**指導老師：楊進雄老師**

**組 長：10956010 童柏祐**

**組 員：10956011 周恩宇** **10956030 林妤玟**

**10956044 顏子綺** **10956051 范雨茜**

**中 華 民 國 1 1 3 年 5 月 2 7 日**

**目錄**

[第一章　前言 1](#_Toc165842861)

[1-1 背景簡介 1](#_Toc165842862)

[1-2 動機 1](#_Toc165842863)

[1-3 系統目的與目標 2](#_Toc165842864)

[1-4 預期成果 2](#_Toc165842865)

[第二章 營運計畫 3](#_Toc165842866)

[2-1 可行性分析 3](#_Toc165842867)

[2-2 商業模式 3](#_Toc165842868)

[2-3 市場分析-STP 3](#_Toc165842869)

[2-4 競爭力分析 SWOT-TOWS 4](#_Toc165842870)

[第三章 系統規格 5](#_Toc165842871)

[3-1系統架構 5](#_Toc165842872)

[3-2系統開發工具 5](#_Toc165842873)

[3-3系統使用環境開發工具 6](#_Toc165842874)

[第四章 專案時程與組織分工 7](#_Toc165842875)

[4-1專案時程圖 7](#_Toc165842876)

[4-2 專案組織與分工 8](#_Toc165842877)

[第五章 需求模型 11](#_Toc165842878)

[5-1 功能分解圖(Functional Decomposition Diagram) 11](#_Toc165842879)

[5-2 需求清單 12](#_Toc165842880)

[第六章 程序模型 13](#_Toc165842881)

[6-1 資料流程圖(Data flow diagram) 13](#_Toc165842882)

[6-2程序規格書(Process specification) 19](#_Toc165842883)

[第七章 資料模型 26](#_Toc165842884)

[7-1 實體關聯圖(Entity relationship diagram) 26](#_Toc165842885)

[7-2資料字典(Data dictionary) 27](#_Toc165842886)

[第八章 資料庫設計 28](#_Toc165842887)

[8-1資料庫關聯表(Constraints) 28](#_Toc165842888)

[8-2表格及其Meta data 29](#_Toc165842889)

**圖目錄**

[圖 3-1-1 系統架構 5](#_Toc167026211)

[圖 4-1-1 專案時程 7](#_Toc167026212)

[圖 5-1-1 功能分解圖 11](#_Toc167026213)

[圖 6-1-1 系統環境圖 13](#_Toc167026214)

[圖 6-1-2 DFD圖0 14](#_Toc167026215)

[圖 6-1-3 DFD圖1 15](#_Toc167026216)

[圖 6-1-4 DFD圖2 15](#_Toc167026217)

[圖 6-1-5 DFD圖3 16](#_Toc167026218)

[圖 6-1-6 DFD圖4 16](#_Toc167026219)

[圖 6-1-7 DFD圖5 17](#_Toc167026220)

[圖 6-1-8 DFD圖6 17](#_Toc167026221)

[圖 6-1-9 DFD圖7 18](#_Toc167026222)

[圖 6-1-10 DFD圖8 18](#_Toc167026223)

[圖 7-1-1 實體關聯圖 26](#_Toc167026224)

[圖 8-1-1 資料庫關聯圖 28](#_Toc167026225)

**表目錄**

[表 2-4-1 SWOT分析 4](#_Toc167023861)

[表 2-4-2 TOWS分析 4](#_Toc167023862)

[表 3-2-1 系統軟、硬體需求 5](#_Toc167023863)

[表 3-3-1 開發工具 6](#_Toc167023864)

[表 4-2-1 專案組織分工表 8](#_Toc167023865)

[表 4-2-2 專題成果工作內容與貢獻度表 10](#_Toc167023866)

[表 5-2-1 功能性需求 12](#_Toc167023867)

[表 5-2-2 非功能性需求 12](#_Toc167023868)

[表 6-2-1 程序規格書-題目作答 19](#_Toc167023869)

[表 6-2-2 程序規格書-收藏題目 19](#_Toc167023870)

[表 6-2-3 程序規格書-搜尋題目 19](#_Toc167023871)

[表 6-2-4 程序規格書-新增討論 20](#_Toc167023872)

[表 6-2-5 程序規格書-修改討論 20](#_Toc167023873)

[表 6-2-6 程序規格書-刪除討論 20](#_Toc167023874)

[表 6-2-7 程序規格書-創建比賽 21](#_Toc167023875)

[表 6-2-8 程序規格書-加入比賽 21](#_Toc167023876)

[表 6-2-9 程序規格書-新增會員 21](#_Toc167023877)

[表 6-2-10 程序規格書-會員資料 22](#_Toc167023878)

[表 6-2-11 程序規格書-新增反饋 22](#_Toc167023879)

[表 6-2-12 程序規格書-反饋結果 22](#_Toc167023880)

[表 6-2-13 程序規格書-新增題目 23](#_Toc167023881)

[表 6-2-14 程序規格書-修改題目 23](#_Toc167023882)

[表 6-2-15 程序規格書-刪除題目 23](#_Toc167023883)

[表 6-2-16 程序規格書-查詢題目 24](#_Toc167023884)

[表 6-2-17 程序規格書-新增討論群 24](#_Toc167023885)

[表 6-2-18 程序規格書-修改討論群 24](#_Toc167023886)

[表 6-2-19 程序規格書-刪除討論群 25](#_Toc167023887)

[表 6-2-20 程序規格書-反饋結果 25](#_Toc167023888)

[表 6-2-21 程序規格書-反饋內容 25](#_Toc167023889)

[表 7-2-1 資料字典 27](#_Toc167023890)

[表 8-2-1 user資料表 29](#_Toc167023891)

[表 8-2-2 feedback資料表 29](#_Toc167023892)

[表 8-2-3 discussion資料表 30](#_Toc167023893)

[表 8-2-4 problem資料表 30](#_Toc167023894)

[表 8-2-5 answer record資料表 31](#_Toc167023895)

[表 8-2-6 collection資料表 31](#_Toc167023896)

[表 8-2-7 contest problem資料表 31](#_Toc167023897)

[表 8-2-8 contest資料表 32](#_Toc167023898)

[表 8-2-9 contest participant 資料表 32](#_Toc167023899)

第一章　前言

**1-1 背景簡介**

在電腦領域已發明了上千種不同的程式語言，每年仍有新的程式語言誕生。很多程式語言需要用指令方式說明計算的程式，而有些程式語言則屬於宣告式程式設計，說明需要的結果，而不說明如何計算。

隨著數位時代的來臨，程式設計成為了現代社會不可或缺的一部分，對於學生、工程師和技術人員而言，具備良好的程式設計基礎和解題能力已經成為進入高科技行業和取得成功的必要條件之一。

在過去的學習和實踐過程中，我們曾花費大量的時間在尋找高質量的學習資源上。我們意識到現有的學習平台雖然提供了大量的題目和教材，但傳統的教育方式往往缺乏趣味性和互動性，使學生們難以找到與之相符的學習方法；然而對於職場工作者和專業技術人員來說，他們更需要的是能夠針對性解決問題的實用知識和技能，因此在個性化學習和社群建設方面還有很大的改進空間。

**1-2 動機**

我們的動機源於對技術教育的貢獻渴望，希望能打造一個開放、包容、高效的程式設計學習平台，為所有程式設計學習者提供一個共享知識、交流經驗、共同成長的空間。我們希望通過CodeAlchemy這個平台，解決現有程式設計學習資源的種種問題，讓每一位學習者都能夠輕鬆、有趣地學習程式設計，提升自己的程式技術水平。

**1-3 系統目的與目標**

1. 提供更多樣的學習資源：提供廣泛且高質量的程式設計學習資源，包括教材、練習題、項目案例等，以幫助使用者們掌握各種程式語言和技術。

2. 促進討論交流：提供一個互動交流的平台，讓學習者都能夠彼此交流學習心得、分享經驗、解決問題，從而促進學習的效果和效率。

3. 執行多種語言：系統可以同時支援C/C++、JAVA、Python、JavaScript...等多種語言。

4. 技術交流和創新：促進技術交流和創新，鼓勵學生參與共同開發項目、參加技術競賽等活動，從而激發創新潛力和培養團隊合作能力。

5. 使用dolos來分群識別不同題目的解法數量，可以快速且直觀的了解一個程式題目有多少種不同解法。

**1-4 預期成果**

本專題主要建立一個全面的學習資源平台，該平台將提供各種程式語言和技術的高質量教材和練習題，並打造互動交流的社群環境，讓學習者能夠彼此學習、共同成長。透過技術交流和實踐，激勵各種學生們參與創新項目和解決實際問題，從而促進技術創新和人才培養，為技術社群的發展注入新的活力和動力。

第二章 營運計畫

**2-1 可行性分析**

1. 程式碼相似度比對系統是一個幫助開發人員檢視代碼重複的程度、尋找代碼抄襲的功用。

2. 技術可行性評估：我們評估了 Dolos、Diff Checker、CodePen 和 pycode similar 這些現有的程式碼比對工具。由於每個工具產生的比對結果有時都有所不同，因此我們採取使用率較高的工具來產生最終結果。

3. 經濟可行性評估：需考慮系統開發和維護階段的成本，包括技術人員、資料庫和設備方面的費用。此外，還需考量系統的行銷成本，提升CodeAlchemy在市場上的知名度和使用率。值得一提的是，相關產品Codequiry和Copylacks兩個成功商品化的典型案例，將為本系統提供了寶貴的參考和啟示，以確保系統的成功發展和長期可持續性。

4. 風險評估：在開發過程中，主要面臨技術層面的挑戰，若是提供服務遜色於同類型的系統，將對我們構成競爭壓力。因此，系統需要不斷地提升技術水平、追蹤行業動態，以及保持創新和敏捷，以應對競爭帶來的挑戰，確保能在市場上的領先地位。

**2-2 商業模式**

1. 定位：目標是協助開發人員在開發過程中提升程式執行效能，即降低程式碼的重複程度。同時，CodeAlchemy也能夠確認程式碼是否存在抄襲疑慮，以確保程式碼的原創性。有助於提高程式碼的品質和可靠性。

2. 教育與學術市場模式：基於檢查抄襲的功能，廣泛應用於學術課堂。透過測驗來了解學生在編程方面的學習能力，同時預防學生的作弊行為。此外，為支持教學和學術活動，以滿足學術機構的需求。

**2-3 市場分析-STP**

1. 市場區隔：教育機構市場，提供管理課程、教學資源和分析學生學習情況的工具。

2. 市場目標：主要為學校、大學、培訓中心等。初學程式設計、技術面試準備者等。

3. 市場定位：網站提供的易用性、互動性和豐富的教育資源，以幫助學生更好地學習和提高技術水平。

**2-4 競爭力分析 SWOT-TOWS**

▼表2-4-1 SWOT分析

|  |  |
| --- | --- |
| 優勢(Strengths) | 弱勢(Weaknesses) |
| 1. 提供中文題庫，讓看不懂英文的人也能輕易讀懂題目。 2. 各種平台的題目，讓使用者可以學習到更多知識。 3. 提供dolos檢測程式碼抄襲的工具，可以更快速的了解一個程式題目有多少種不同解法。 | 1. 要持續開發和維護一個高質量的教育平台需要大量的技術和資源。 2. 無法及時調整和改進平台，可能會影響使用者的滿意度。 3. 系統無法處理某些程式語言，需要進一步改善。 |
| 機會(Opportunities) | 威脅(Threats) |
| 1. 可以透過不斷更新，以此來精進平台的精準度和性能，提供更加優質的服務。 2. 與教育機構合作，為學生創造新的教育機會，提高學習效率。 | 1. 新技術的出現和技術的變化可能會對平台的技術架構和功能產生影響，需要及時跟進和調整。 2. 程式設計教育市場已經飽和，存在過多的競爭對手和類似的教育平台，使得剛進入市場和吸引用戶變得更加困難。 |

▼表2-4-2 TOWS分析

|  |  |  |
| --- | --- | --- |
|  | S優勢 | W劣勢 |
| O  機會 | 邀請專業團隊來開發新的學習課程或是培訓計畫，來滿足需要更多課程的學生。 | 透過學術機構或是教育機構建立合作夥伴關係，來提供學生學會使用系統，收集更多回饋，以此來改進系統內容。 |
| T  威脅 | 利用中文題庫的優勢來吸引那些不懂英文的用戶，特別是在中文使用者眾多的地區或市場。 | 透過更多教育機構或技術公司合作，來共享資源和技術，以此減輕資源的不足和技術落後所帶來的威脅。 |

第三章 系統規格

**3-1系統架構**

**一張含有 圖表, Rectangle, 方案, 工程製圖 的圖片

自動產生的描述**

* 圖 3-1-1 系統架構

**3-2系統開發工具**

▼表3-2-1 系統軟、硬體需求

|  |  |
| --- | --- |
| 軟體需求 | |
| 作業系統需求 | Windows 10(版本1903或更新版本) |
| 硬體需求 | |
| 處理器 | 64位元處理器，最好是支援虛擬化技術的處理器，如Intel VT-x或AMD-V。 |
| 記憶體 | 至少8GB RAM。  建議配置16GB RAM或更高。 |
| 儲存空間 | 至少20GB可用空間。 |
| 虛擬化技術 | 需要啟用虛擬化技術。  可以在BIOS或UEFI設定中啟用它。 |
| 顯示器 | 最佳解析度為1024 x 768或更高。 |

**3-3系統使用環境開發工具**

▼表 3-3-1 開發工具

|  |  |  |
| --- | --- | --- |
| 系統開發環境 | | |
| 作業系統 |  | Windows、Ubuntu |
| 撰寫工具 |  | Visual Studio Code、MySQL workbench |
| 程式開發語言 | | |
| 前端 |  | HTML、Bootstrap、CSS |
| 後端 |  | Python、Flask |
| 資料庫 |  | MySQL |
| 介面及插圖繪製工具 | | |
| 插圖 | Adobe Illustrator - 維基百科，自由的百科全書 | Bootstrap Icons |
| 介面 |  | Figma |
| 文件及美化工具 | | |
| 文件 |  | Microsoft Word、ChatGPT |
| 簡報 |  | Microsoft PowerPoint |
| 專案管理及版本控制工具 | | |
| 應用程式 |  | Visual Studio Code |
| 版本控制 |  | Git |

第四章 專案時程與組織分工

**4-1專案時程圖**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 項目  時間 | 2024 | | | | | | | | | |
| 1月 | 2月 | 3月 | 4月 | 5月 | 6月 | 7月 | 8月 | 9月 | 10月 |
| 主題制訂與討論 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文件製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前端頁面撰寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 後端頁面撰寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 介面設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| LOGO設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 系統測試 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 簡報製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 影片製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 圖 4-1-1 專案時程

**4-2 專案組織與分工**

▼表 4-2-1 專案組織分工表

●主要負責人 ○次要負責人

| 項目/組員 | | 10956010  童柏祐 | 10956011  周恩宇 | 10956030  林妤玟 | 10956044  顏子綺 | 10956051  范雨茜 |
| --- | --- | --- | --- | --- | --- | --- |
| 後端開發 | 資料庫建置 | ● |  |  |  |  |
| 伺服器架設 | ○ | ● |  |  |  |
| 帳號管理 |  | ● |  |  |  |
| 題目管理 |  | ● |  |  |  |
| 比賽管理 |  |  |  | ● |  |
| 會員管理 |  | ● |  |  |  |
| 反饋管理 |  |  |  |  | ● |
| 排行榜 |  |  | ● |  |  |
| 爬蟲 | ● | ○ |  |  |  |
| 前端開發 | 註冊+登入 |  | ● |  |  |  |
| 解題介面 | ○ | ● |  |  |  |
| 系統主畫面 |  | ● |  |  |  |
| 排行榜 |  |  | ● |  |  |
| 新增比賽 |  |  |  | ● |  |
| 加入比賽 |  |  |  | ● |  |
| 會員資料 |  | ● |  |  |  |
| 程式碼分析畫面 |  | ● |  |  |  |
| 反饋 |  |  |  |  | ● |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 美術設計 | UI/ UX | ○ | ● |  |  |  |
| Web/APP介面設計 |  | ● |  |  |  |
| 色彩設計 |  |  |  | ● |  |
| Logo設計 |  |  |  | ● |  |
| 素材設計 |  |  |  | ● |  |
| 文件撰寫 | 統整 |  |  | ● | ○ |  |
| 第1章 前言 |  |  | ● |  |  |
| 第2章 營運計畫 |  |  | ● |  |  |
| 第3章 系統規格 |  |  |  | ● |  |
| 第4章 專題時程與組織分工 |  |  | ○ |  | ● |
| 第5章 需求模型 |  |  | ○ | ○ | ● |
| 第6章 程序模型 |  |  | ○ | ○ | ● |
| 第7章 資料模型 |  |  | ○ | ○ | ● |
| 第8章 資料庫設計 | ● |  | ○ |  |  |
| 第9章 程式 |  | ● |  |  |  |
| 第10章 測試模型 |  |  | ● |  |  |
| 第11章 操作手冊 |  |  | ● |  |  |
| 第12章 使用手冊 |  |  | ● |  |  |
| 其 他 | 簡報製作 |  |  | ● |  |  |
| 影片製作 |  | ● |  |  |  |

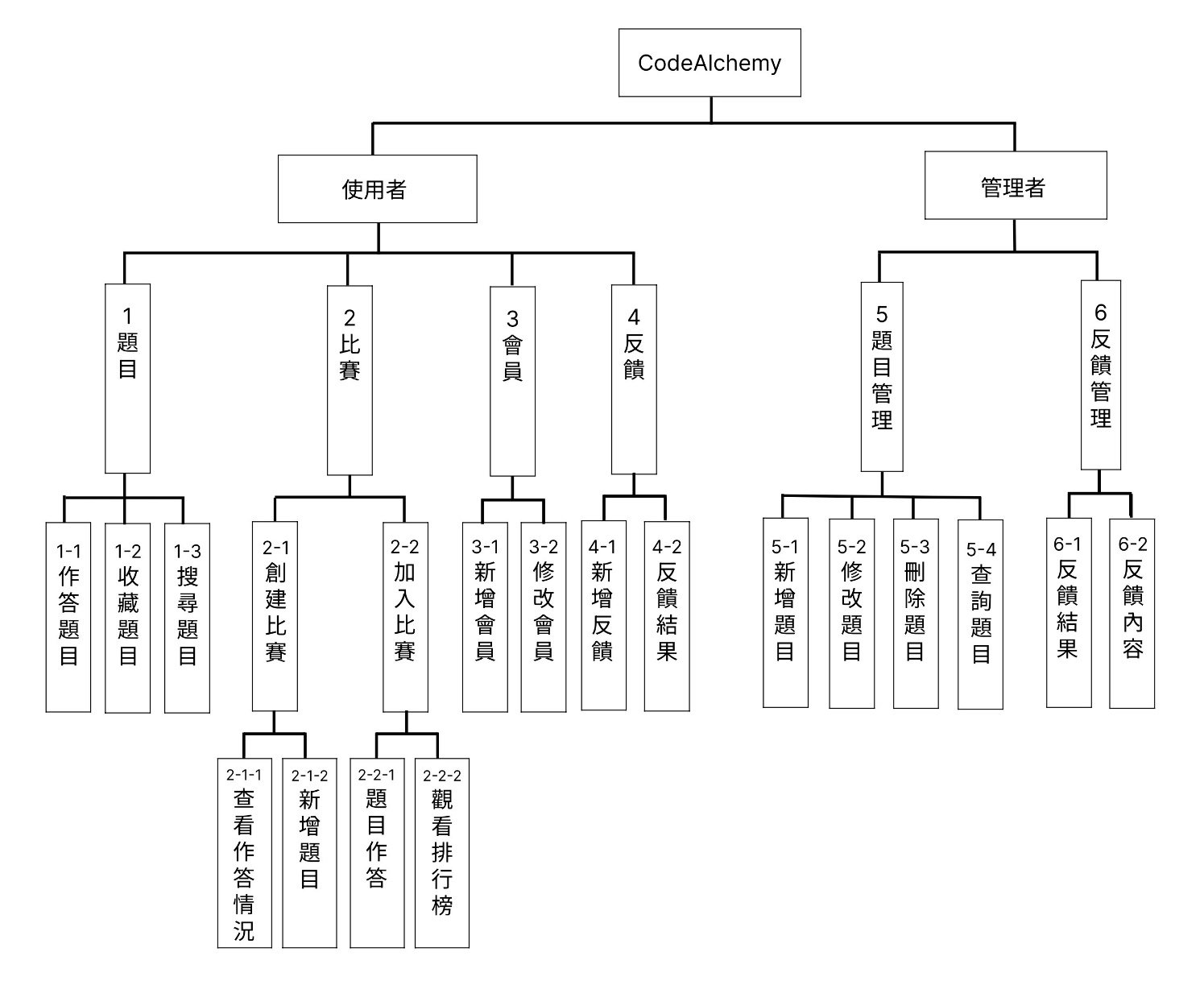
▼表 4-2-2 專題成果工作內容與貢獻度表

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容<各限100字以內> | 貢獻度 |
| 1 | 組長  童柏祐 | 建置資料庫、資料庫關聯圖、資料庫管理圖繪製、自動註冊OJ帳號、抓取OJ的題目、提交使用者程式碼並顯示結果、處理作答紀錄、爬蟲。 | 20 % |
| 2 | 組員  周恩宇 | 負責前端UI設計；logo發想；後端登入功能開發；使用Google帳號登入；串接前端後端及資料庫；使用者權限控管；研究flask-mail、codemirror、docker；dolos架設；串接dolos api。 | 20 % |
| 3 | 組員  林妤玟 | 總體文件統整、前言、SOWT分析、TWOS分析、專題時程與組織分工、需求清單、非需求清單、需求模型、程序模型、資料模型、系統簡介。 | 20 % |
| 4 | 組員  顏子綺 | logo設計、素材設計、系統規格、文件第八章、文件統整、簡報製作、創建比賽介面、創建比賽後端、加入比賽介面、加入比賽後端、系統架構圖、功能分解圖、資料流程圖。 | 20 % |
| 5 | 組員  范雨茜 | 功能分解圖、資料流程圖、實體關聯圖、專題時程繪製，修改資料庫關聯圖，建置資料庫，反饋與反饋管理的前後端程式與頁面撰寫。 | 20 % |
|  | | | 總計:100% |

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

第五章 需求模型

**5-1 功能分解圖(Functional Decomposition Diagram)**

* 圖 5-1-1 功能分解圖

**5-2 需求清單**

▼表5-2-1 功能性需求

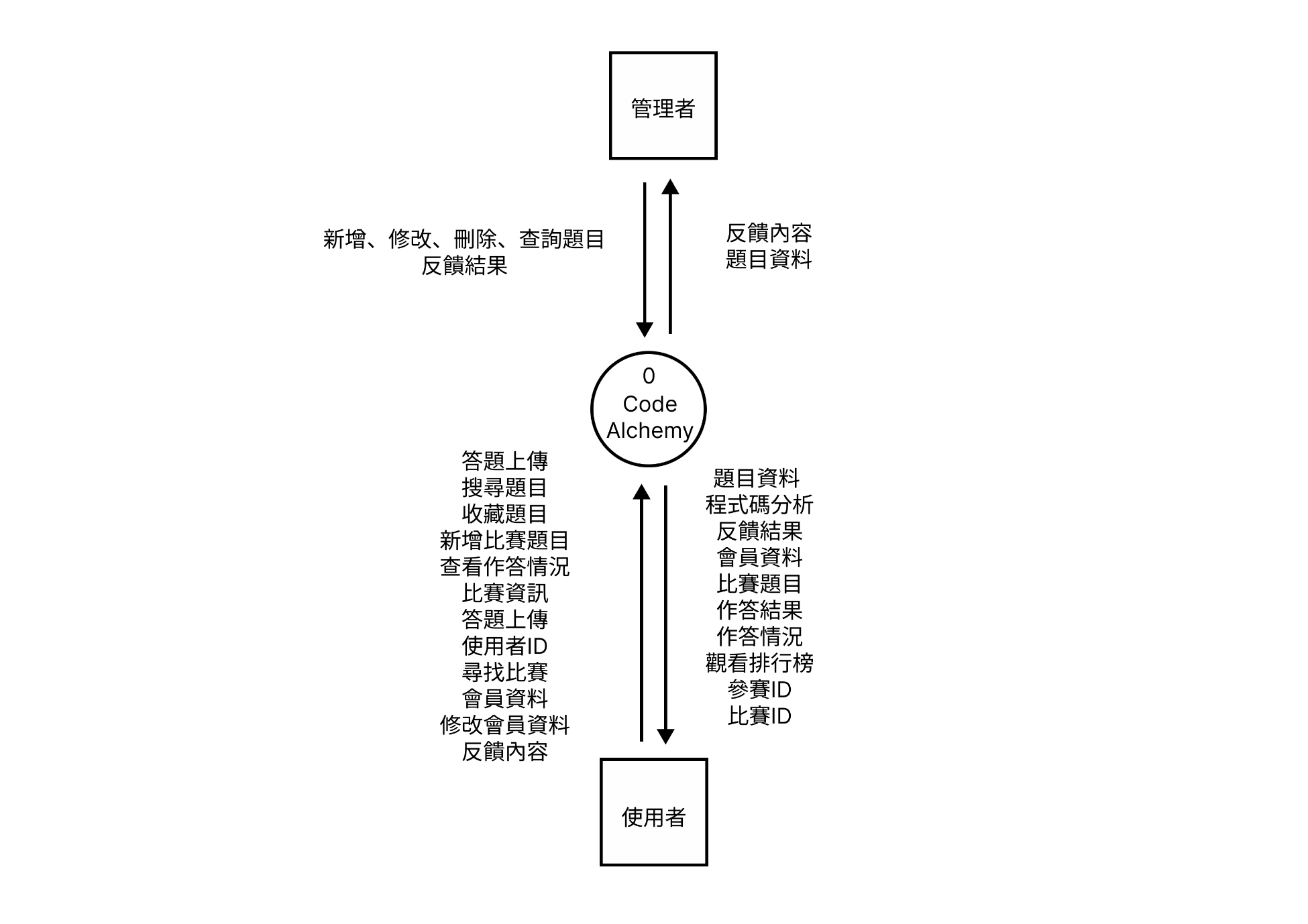
|  |  |  |
| --- | --- | --- |
| **編號** | **功能** | **功能敘述** |
| **1** | 題目 | 1-1作答題目：點擊題目，輸入程式即可上傳。  1-2收藏題目：為使用者儲存喜愛的題目。  1-3搜尋題目：使用者可透過搜尋功能來尋找想要的題目。 |
| **2** | 比賽 | 2-1創建比賽：使用者可創建比賽來考驗使用者的程式語言能力。  2-2加入比賽：可以觀看比賽內容來決定是否參加比賽。 |
| **3** | 會員 | 3-1新增會員：點擊”註冊”，使用者可以新增會員資料到系統內。  3-2修改會員：點擊”個人資料”即可查看使用者存到系統內的個人基本資料。 |
| **4** | 反饋 | 提供使用者來對系統意見反饋。 |
| **5** | 題目管理 | 5-1新增題目：管理者可以新增題目至系統內。  5-2修改題目：管理者可修改系統內的題目內容。  5-3刪除題目：管理者可以刪除題目。  5-4查詢題目：讓管理者更快速查詢到題目。 |
| **6** | 反饋管理 | 提供管理者來觀看使用者所提出意見，並加以改善系統。 |

▼表 5-2-2 非功能性需求

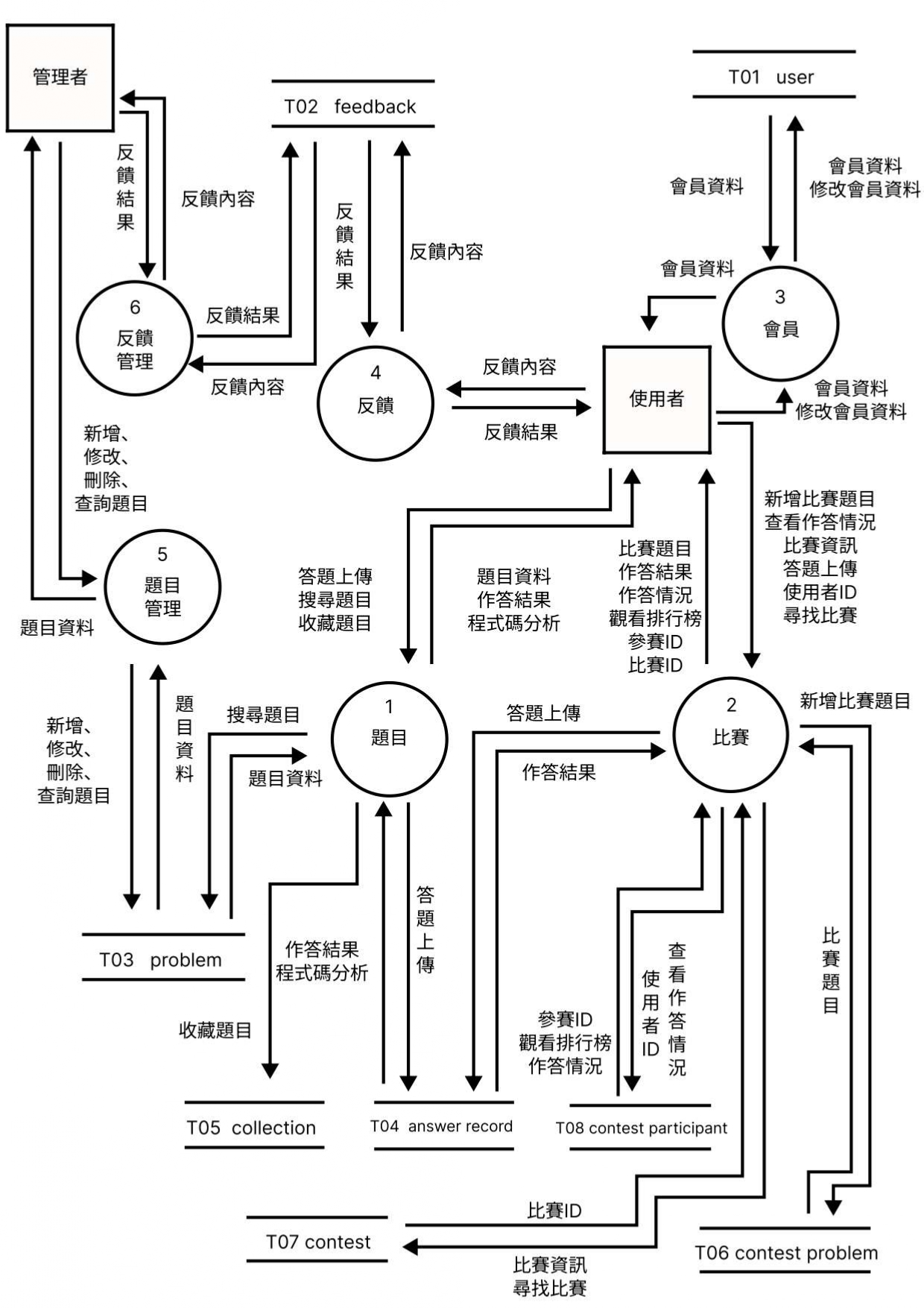
|  |  |
| --- | --- |
| **功能** | **功能敘述** |
| 效能需求 | 確保系統能夠快速、穩定地處理大量的用戶請求和數據操作。 最大限度地優化系統性能，盡量減少頁面加載時間和服務器延遲。 |
| 可用性 | 提供直覺友好的用戶界面，讓使用者能夠輕鬆找到所需要的功能並進行操作。 |
| 穩定性 | 測試系統的穩定性和可靠性，確保在高流量和高負載情況下仍能正常運行。 |

第六章 程序模型

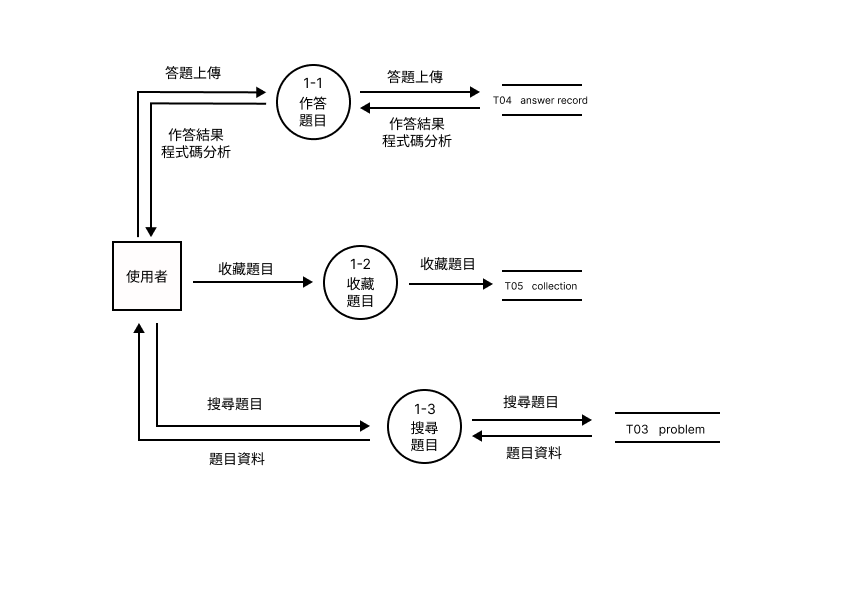
**6-1 資料流程圖(Data flow diagram)**



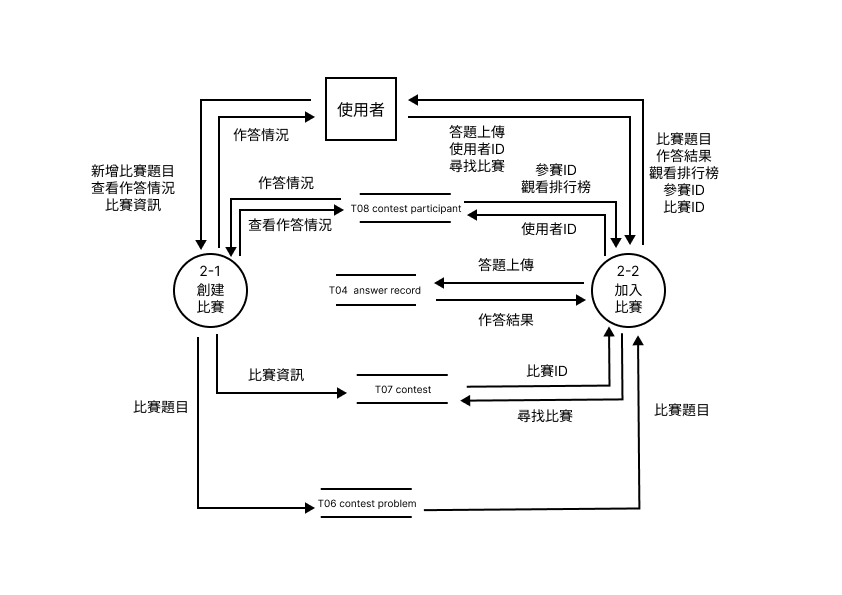
* 圖 6-1-1 系統環境圖



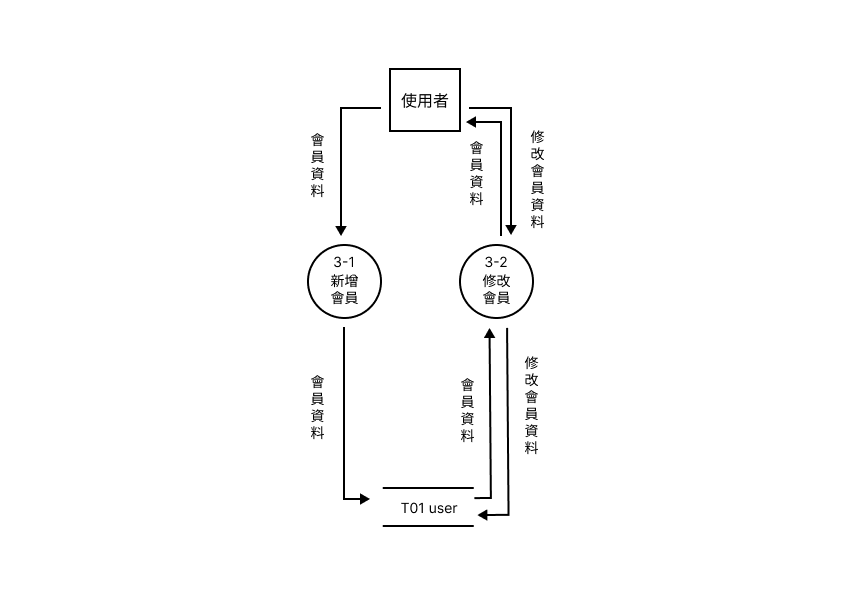
* 圖 6-1-2 DFD圖0



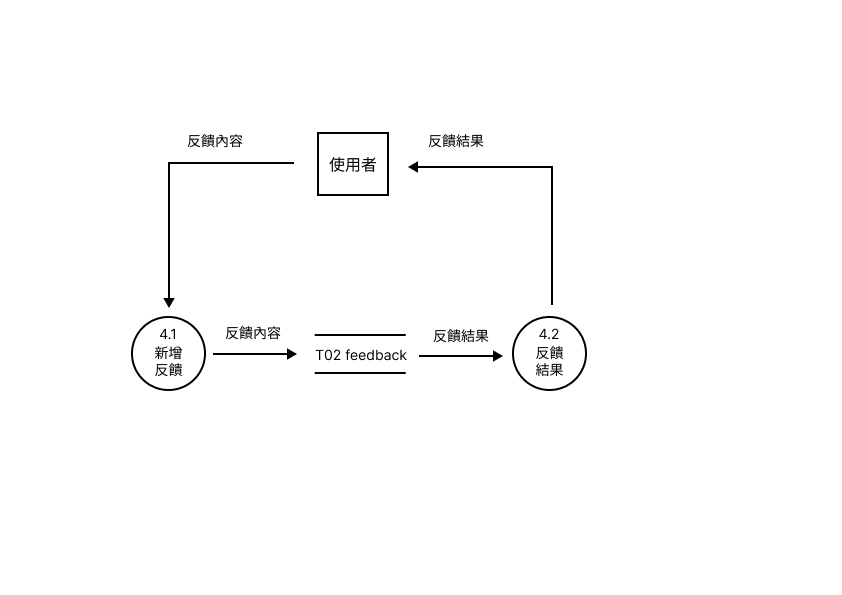
* 圖 1-1-3 DFD圖1



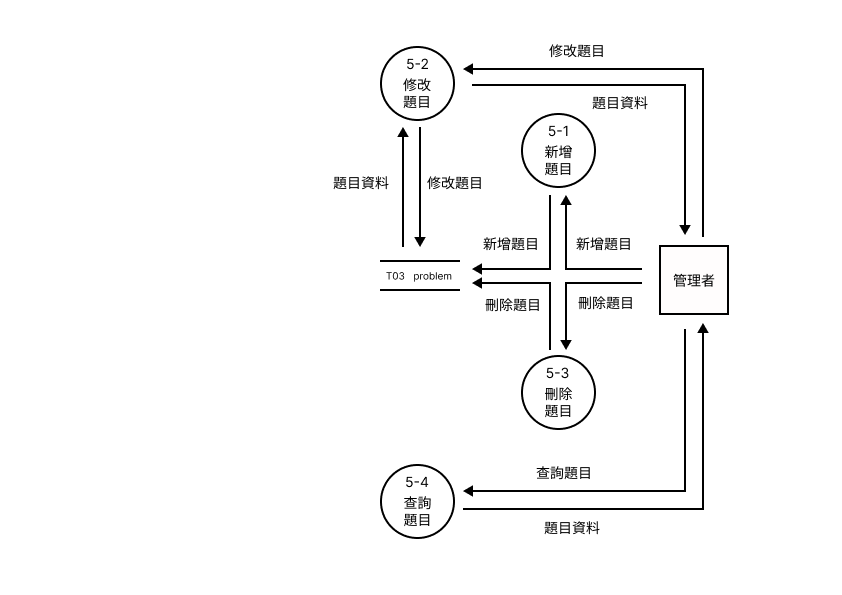
* 圖 6-1-4 DFD圖2



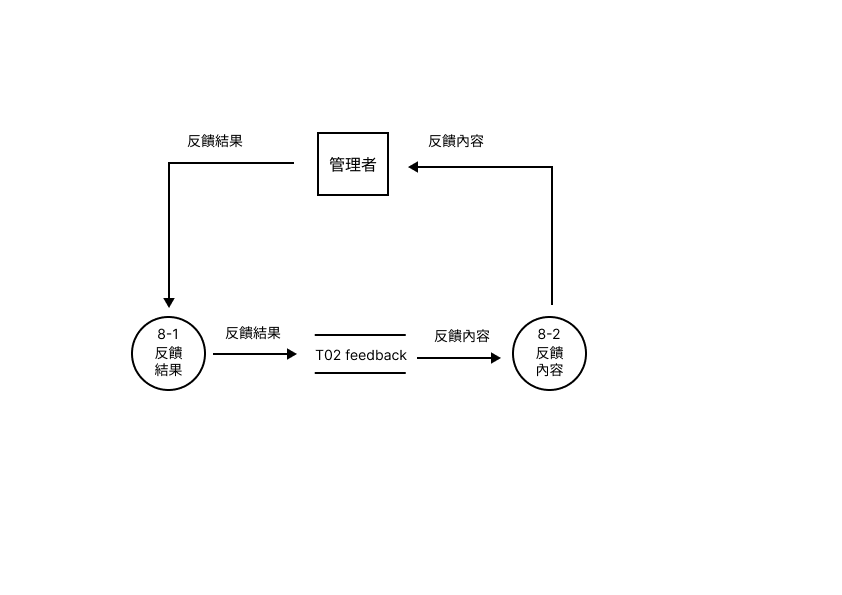
* 圖 6-1-5 DFD圖3



* 圖 6-1-6 DFD圖4



* 圖 6-1-7 DFD圖5



* 圖 6-1-8 DFD圖6

**6-2程序規格書(Process specification)**

▼表6-2-1 程序規格書-題目作答

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1 | 功能名稱 | 作答題目 |
| 功能說明 | 點擊題目，輸入程式即可上傳 | | |
| 輸入值 | 程式碼 | | |
| 輸出值 | 作答結果和程式碼分析 | | |

▼表 6-2-2 程序規格書-收藏題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2 | 功能名稱 | 收藏題目 |
| 功能說明 | 為使用者儲存喜愛的題目 | | |
| 輸入值 | 欲新增收藏題目資料 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-3 程序規格書-搜尋題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3 | 功能名稱 | 搜尋題目 |
| 功能說明 | 使用者透過搜尋功能來尋找想要的題目 | | |
| 輸入值 | 題目名稱 | | |
| 輸出值 | 查詢成功訊息 | | |

▼表 6-2-4 程序規格書-創建比賽

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1 | 功能名稱 | 創建比賽 |
| 功能說明 | 使用者可創建比賽來考驗使用者的程式語言能力 | | |
| 輸入值 | 新增比賽資料 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-5 程序規格書-加入比賽

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2 | 功能名稱 | 加入比賽 |
| 功能說明 | 可以觀看比賽內容來決定是否參加比賽 | | |
| 輸入值 | 是否參加比賽 | | |
| 輸出值 | 參加比賽成功訊息 | | |

▼表 6-2-6 程序規格書-新增會員

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 3-1 | 功能名稱 | 新增會員 |
| 功能說明 | 點擊註冊，使用者可以新增會員資料到系統內 | | |
| 輸入值 | 欲註冊資料 | | |
| 輸出值 | 註冊成功訊息 | | |

▼表 6-2-7 程序規格書-會員資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 3-2 | 功能名稱 | 修改會員 |
| 功能說明 | 點擊”修改會員資料”即可修改個人資料 | | |
| 輸入值 | 個人資料 | | |
| 輸出值 | 個人資料資訊 | | |

▼表 6-2-8 程序規格書-新增反饋

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 4-1 | 功能名稱 | 新增反饋 |
| 功能說明 | 新增系統反饋資料 | | |
| 輸入值 | 反饋內容 | | |
| 輸出值 | 反饋成功訊息 | | |

▼表 6-2-9 程序規格書-反饋結果

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 4-2 | 功能名稱 | 反饋結果 |
| 功能說明 | 查看反饋結果 | | |
| 輸入值 | 反饋結果 | | |
| 輸出值 | 反饋結果內容 | | |

▼表 6-2-10 程序規格書-新增題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-1 | 功能名稱 | 新增題目 |
| 功能說明 | 管理者可以新增題目至系統內 | | |
| 輸入值 | 題目標題與題目內容 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-11 程序規格書-修改題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-2 | 功能名稱 | 修改題目 |
| 功能說明 | 管理者可修改系統內的題目內容 | | |
| 輸入值 | 修改題目內容 | | |
| 輸出值 | 修改成功訊息 | | |

▼表 6-2-12 程序規格書-刪除題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-3 | 功能名稱 | 刪除題目 |
| 功能說明 | 管理者可以刪除題目 | | |
| 輸入值 | 刪除題目內容 | | |
| 輸出值 | 刪除成功訊息 | | |

▼表 6-2-13 程序規格書-查詢題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-4 | 功能名稱 | 查詢題目 |
| 功能說明 | 讓管理者更快速查詢到題目 | | |
| 輸入值 | 題目名稱 | | |
| 輸出值 | 查詢成功訊息 | | |

▼表 6-2-14 程序規格書-反饋結果

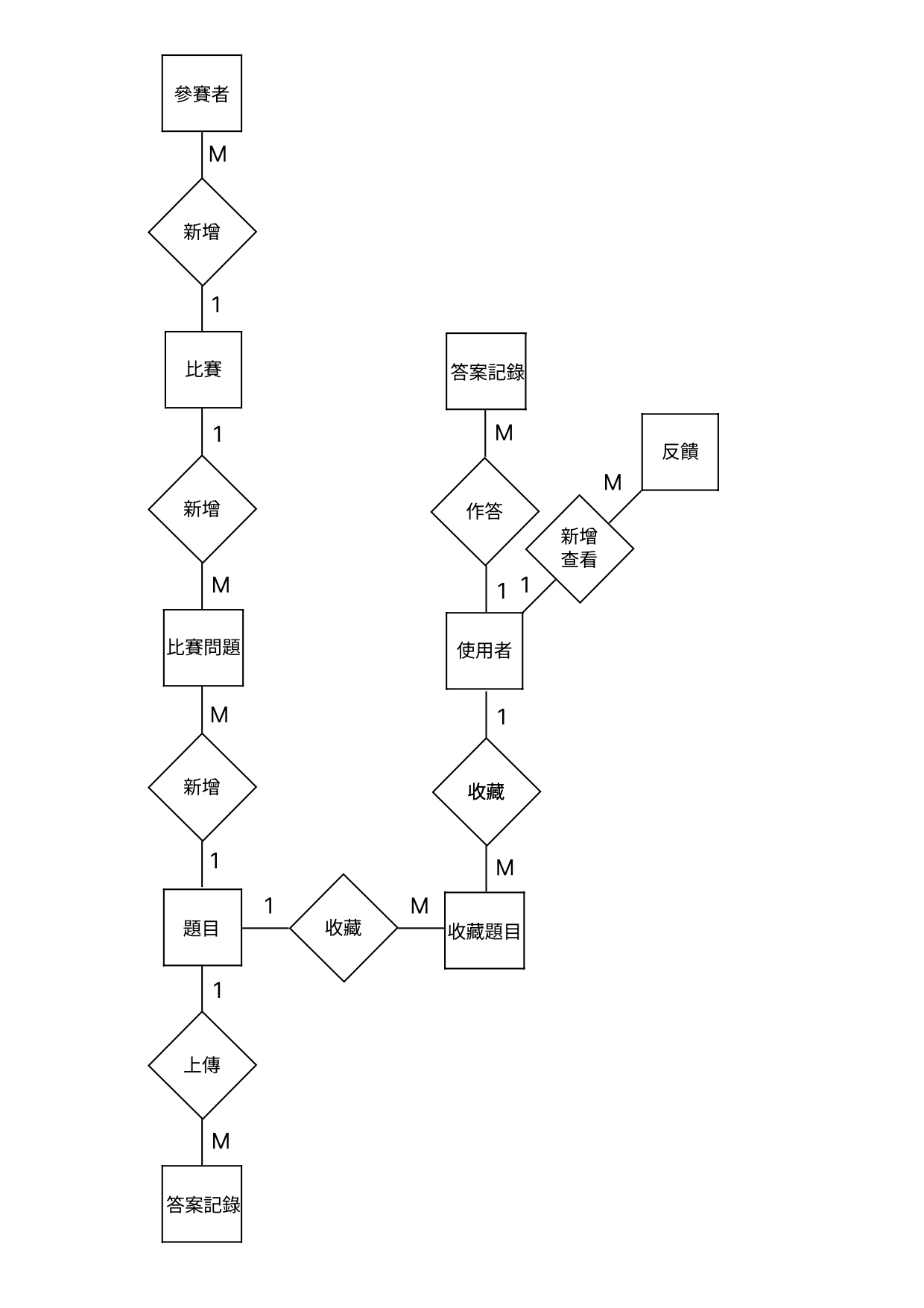
|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 6-1 | 功能名稱 | 反饋結果 |
| 功能說明 | 管理者將反饋結果發送給系統 | | |
| 輸入值 | 反饋結果 | | |
| 輸出值 | 反饋成功訊息 | | |

▼表 6-2-15 程序規格書-反饋內容

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 6-2 | 功能名稱 | 反饋內容 |
| 功能說明 | 系統將反饋結果的內容發送給使用者 | | |
| 輸入值 | 反饋內容 | | |
| 輸出值 | 反饋內容資料 | | |

第七章 資料模型

**7-1 實體關聯圖(Entity relationship diagram)**



* 圖 7-1-1 實體關聯圖

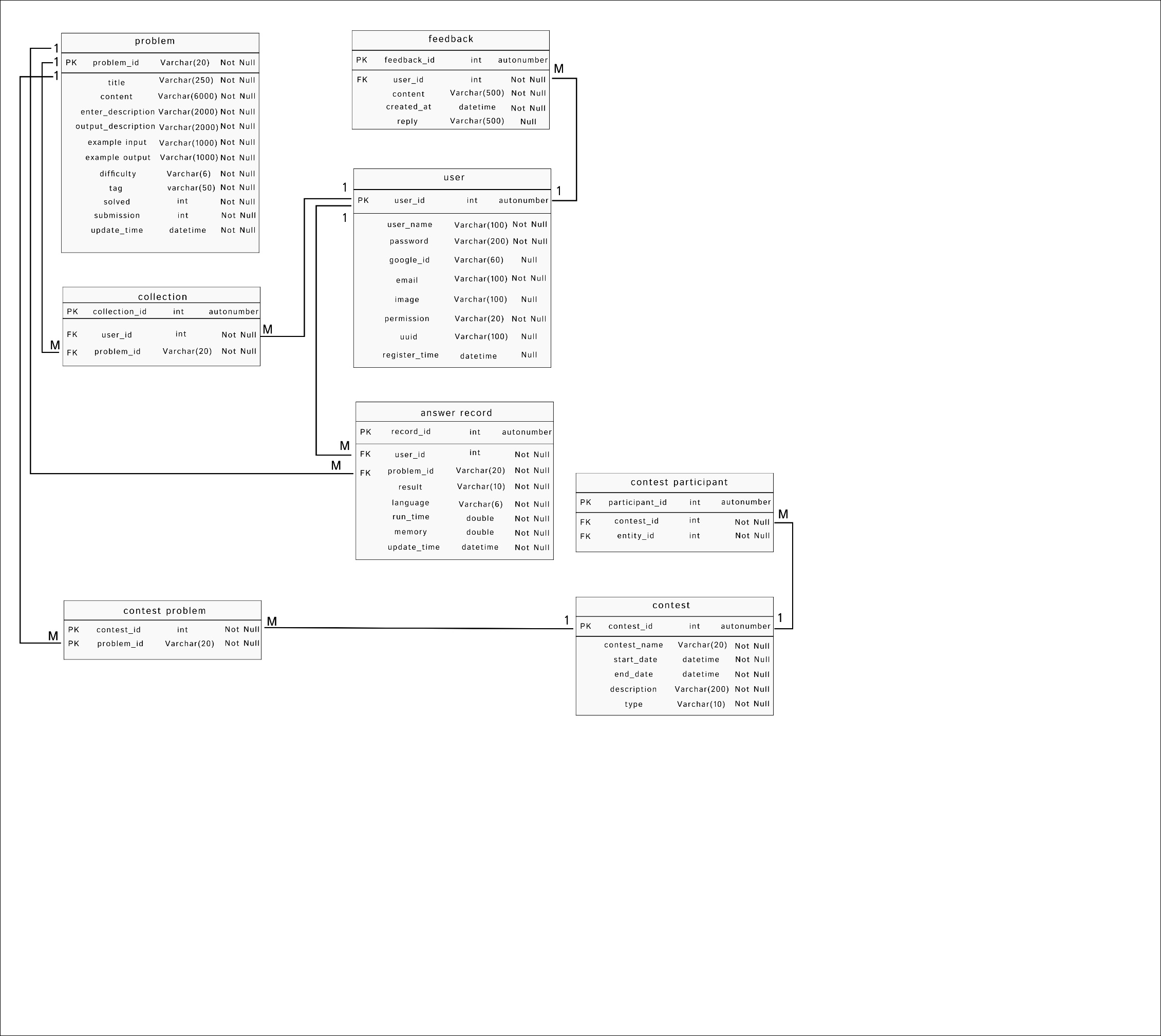
**7-2資料字典(Data dictionary)**

▼表 7-2-1 資料字典

|  |  |  |
| --- | --- | --- |
| **資料表編號** | **英文名稱** | **中文名稱** |
| **T01** | user | 使用者 |
| **T02** | feedback | 反饋 |
| **T03** | problem | 題目 |
| **T04** | answer record | 答案紀錄 |
| **T05** | collection | 收藏題目 |
| **T06** | contest problem | 比賽問題 |
| **T07** | contest | 比賽 |
| **T08** | contest participant | 參賽者 |

第八章 資料庫設計

**8-1資料庫關聯表(Constraints)**

****

* 圖 8-1-1 資料庫關聯圖

**8-2表格及其Meta data**

▼表 8-2-1 user資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T01 user | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| user\_id | int |  | ü | ü |  |  |
| user\_name | varchar | 100 | ü | ü |  |  |
| password | varchar | 200 | ü |  |  |  |
| google\_id | varchar | 60 |  |  |  |  |
| email | varchar | 100 | ü |  |  |  |
| image | varchar | 100 |  |  |  |  |
| permission | varchar | 20 | ü |  |  |  |
| uuid | varchar | 100 |  |  |  |  |
| register\_time | datetime |  |  |  |  |  |

▼表 8-2-2 feedback資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T02 feedback | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| feedback\_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| content | varchar | 500 | ü |  |  |  |
| created\_at | datetime |  | ü |  |  |  |
| reply | varchar | 500 |  |  |  |  |

▼表 8-2-3 problem資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T04 problem | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| problem\_id | varchar | 20 | ü | ü |  |  |
| title | varchar | 250 | ü |  |  |  |
| content | varchar | 6000 | ü |  |  |  |
| enter  \_description | varchar | 2000 | ü |  |  |  |
| output  \_description | varchar | 2000 | ü |  |  |  |
| example  input | varchar | 1000 | ü |  |  |  |
| example  output | varchar | 1000 | ü |  |  |  |
| difficulty | varchar | 6 | ü |  |  |  |
| tag | varchar | 20 | ü |  |  |  |
| solved | int |  | ü |  |  |  |
| submission | int |  | ü |  |  |  |
| update\_time | datetime |  | ü |  |  |  |

▼表 8-2-4 answer record資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T05 answer record | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| record\_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| problem\_id | varchar | 20 | ü |  | ü |  |
| result | varchar | 10 | ü |  |  |  |
| language | varchar | 6 | ü |  |  |  |
| run\_time | double |  | ü |  |  |  |
| memory | double |  | ü |  |  |  |
| update\_time | datetime |  | ü |  |  |  |

▼表 8-2-5 collection資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T06 collection | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| collection \_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| problem\_id | varchar | 20 | ü |  | ü |  |

▼表 8-2-6 contest problem資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T07 contest problem | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| contest\_id | int |  | ü | ü |  |  |
| problem\_id | varchar | 20 | ü | ü |  |  |

▼表 8-2-7 contest資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T08 contest | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| contest\_id | int |  | ü | ü |  |  |
| contest\_name | varchar | 20 | ü |  |  |  |
| start\_date | datetime |  | ü |  |  |  |
| end\_date | datetime |  | ü |  |  |  |
| description | varchar | 200 | ü |  |  |  |
| type | varchar | 10 | ü |  |  |  |

▼表 8-2-8 contest participant 資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T09 contest participant | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| participant \_id | int |  | ü | ü |  |  |
| contest\_id | int |  | ü |  | ü |  |
| entity\_id | int |  | ü |  | ü |  |

第九章 程式

**9-1 軟體架構與程式清單**

▼表 9-1-1 程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能名稱 |
| 1 | app.py | 系統主要功能 |
| 2 | templates\problem\_list.html | 首頁前端 |
| 3 | templates\problem.html | 題目作答頁面 |
| 4 | templates\rank.html | 排行榜頁面 |
| 5 | services\auth\app.py | 使用者認證相關 |
| 6 | templates\login.html | 查詢電子郵件是否已註冊 |
| 7 | templates\login\_password.html | 登入密碼頁面 |
|  | templates\forget\_password.html | 忘記密碼頁面 |

**9-2程式規格描述**

▼表 9-2-1 程式清單

|  |  |
| --- | --- |
| 路徑 | app.py |
| 用途 | 系統主要功能 |
| 部分程式碼 | |
| #主畫面  @app.route('/', methods=['GET'])  def index():  # 取得使用者的篩選條件  state = request.args.get('state', '\*', type=str)  onlinejudge = request.args.get('onlinejudge', '\*', type=str)  difficulty = request.args.get('difficulty', '\*', type=str)  search = request.args.get('search', '', type=str)    condition = ' WHERE ' # where前後的空格勿動  sql\_problem\_command = 'SELECT p.\*, acceptance\_data.acceptance\_rate FROM `113-CodeAlchemy`.problem AS p '  sql\_problem\_command += 'LEFT JOIN ('  sql\_problem\_command += """  SELECT  ar.problem\_id,  CONCAT(ROUND(SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) / COUNT(ar.result) \* 100, 2), '%') AS acceptance\_rate  FROM  `113-CodeAlchemy`.`answer record` AS ar  GROUP BY  ar.problem\_id  ) AS acceptance\_data ON p.problem\_id = acceptance\_data.problem\_id"""    if search != '':  condition += f'p.title LIKE "%{search}%" AND '  if onlinejudge != '\*':  condition += f'SUBSTRING\_INDEX(p.problem\_id, "-", 1) = "{onlinejudge}" AND '  if difficulty != '\*':  condition += f'p.difficulty = "{difficulty}" AND '  if condition == ' WHERE ':  condition = ''  else:  condition = condition[:condition.rfind(' AND ')]    sql\_problem\_command = sql\_problem\_command + condition  data=db.get\_data(sql\_problem\_command)  # 預設第一頁  page = request.args.get('page', 1, type=int)  # 每頁顯示15列  per\_page = 15  start\_page = max(1, page - 1)  end\_page = min(page+3,math.ceil(paginate(data,page, per\_page)[1]/per\_page)+1)  paginated\_data = paginate(data,page, per\_page)[0]  #渲染網頁  return render\_template('problem\_list.html',data=paginated\_data,page=page,start\_page=start\_page,end\_page=end\_page,state=state,onlinejudge=onlinejudge,difficulty=difficulty,search=search)  #題目  @app.route('/problem',methods=['GET','POST'])  def problem():  if request.method=="POST":  #從傳入封包取得資料  data = request.form  type = data.get('type')  problem\_id = data.get('problem\_id')  language = data.get('language')  code = data.get('code')  # 定義語言對應的文件擴展名字典  file\_extensions = {  'python': '.py',  'text/x-java': '.java',  'text/x-csrc': '.c',  'text/x-c++src': '.cpp'  }  # 生成 6 位數的亂碼  random\_code = str(uuid.uuid4())[:6]  # 生成隨機字串，第一個字母為英文字母  # 如果第一個字元不是英文字母，則重新生成，直到第一個字元為英文字母  while not random\_code[0].isalpha():  random\_code = str(uuid.uuid4())[:6]  # 構建文件路徑  file\_name = f'{random\_code}\_{problem\_id}{file\_extensions[language]}'  file\_path = os.path.join('./source', file\_name)  # 確保目錄存在  os.makedirs(os.path.dirname(file\_path), exist\_ok=True)  # 寫入內容到文件中  with open(file\_path, 'w') as file:  file.write(code)  if "ZJ" in file\_name:  # 調用 ZeroJudge\_submit 函數進行題目提交  score = ZeroJudge\_submit(file\_name,session['User\_id'])  if score and score[2]=='Accepted':  print()  elif "TIOJ" in file\_name:  run\_time=0  memory=0  score=TIOJ\_submit(file\_name,str(session['User\_id']))  if score and score[2]=='Accepted':  result=True  message="測試成功"  else :  result=False  message="測試失敗"  run\_time=score[3]  memory=score[4]  ensue=score[2]  if type == 'upload':  # 執行 SQL 插入語句  db.edit\_data(f'''  INSERT INTO `answer record` (user\_id, problem\_id, result, language,run\_time,memory, update\_time)  VALUES ('{session['User\_id']}','{problem\_id}','{ensue}', '{language}','{run\_time}','{memory}','{score[-1]}')  ''')    return jsonify({'result':result,  'message':message,  'run\_time':run\_time,  "memory":memory})  else:  problem\_id = request.args.get('problem\_id',type=str)  sql\_problem\_command=f"SELECT \* FROM problem where problem\_id='{problem\_id}'"  problem\_data=db.get\_data(sql\_problem\_command)  example\_inputs = problem\_data[0][5].split('|||')  example\_outputs = problem\_data[0][6].split('|||')  like = db.get\_data(f"SELECT IFNULL(COUNT(\*),0) FROM collection where problem\_id='{problem\_id}'")[0][0]  return render\_template('./problem.html',data=problem\_data,example\_inputs=example\_inputs,example\_outputs=example\_outputs,like=like)  @app.route('/answer\_record',methods=['GET'])  def answer\_record():  problem\_id = request.args.get('problem\_id',type=str)  sql\_problem\_command=f"""SELECT record\_id,u.user\_id,image,user\_name,result,language,run\_time,memory,ar.update\_time FROM `113-CodeAlchemy`.`answer record` as ar  left join `user` as u  on ar.user\_id=u.user\_id  where problem\_id='{problem\_id}'  order by update\_time desc"""  data=db.get\_data(sql\_problem\_command)  return render\_template('./answer\_record.html',data=data)    # 收藏  @app.route('/add\_to\_collection', methods=['POST'])  def add\_to\_collection():  data = request.get\_json()  problem\_id = data.get('problem\_id')  try:  user\_id = session['User\_id']  if db.get\_data(f"SELECT IFNULL(COUNT(\*),0) FROM collection where problem\_id='{problem\_id}' and user\_id='{user\_id}'")[0][0]==0:  sql\_command=f"INSERT INTO collection(user\_id,problem\_id) VALUES ('{user\_id}','{problem\_id}')"  else:  sql\_command=f"DELETE FROM collection WHERE problem\_id = '{problem\_id}' and user\_id='{user\_id}'"  db.edit\_data(sql\_command)  return jsonify({'message': 'Item added to collection'}), 201  except KeyError:  return jsonify({'error': 'Missing item\_id or user\_id'}), 400    @app.route('/dolos', methods=['GET'])  def problem\_dolos():  problem\_id=request.args.get('problem\_id',type=str)  zip=dolos.create\_zip(problem\_id)  url=dolos.submit\_to\_dolos(zip[0],zip[1])  return (redirect(url))  @app.route("/rank")  def rank():  data=db.get\_data(f'''  SELECT  u.user\_id,  u.user\_name,  u.image,  SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) AS 正確答題數,  CONCAT(ROUND(SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) / COUNT(\*) \* 100, 2), '%') AS 答題正確率,  round(avg(ar.run\_time),2) as 平均執行時間,  round(avg(ar.memory),2) as 平均使用記憶體  FROM  `113-CodeAlchemy`.`answer record` AS ar  LEFT JOIN  `user` AS u ON u.user\_id = ar.user\_id  GROUP BY  u.user\_id, u.user\_name  order by 正確答題數 desc;  ''')  return render\_template('./rank.html',data=data) | |

|  |  |
| --- | --- |
| 路徑 | templates\problem\_list.html |
| 用途 | 首頁前端 |
| 部分程式碼 | |
| <div class="container">  <div class="row mt-4">  <div class="col-3">  <select class="form-select" id="state" onchange="location.href=this.options[this.selectedIndex].value">  <option selected>狀態</option>  <option  value="?page={{ page }}&state=\*&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  All  </option>  <option  value="?page={{ page }}&state=accept&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  通過  </option>  <option  value="?page={{ page }}&state=error&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  錯誤  </option>  <option  value="?page={{ page }}&state=none&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  未做  </option>  </select>  </div>  <div class="col-3">  <select class="form-select" id="onlinejudge"  onchange="location.href=this.options[this.selectedIndex].value">  <option selected>OnlineJudge</option>  <option  value="?page=1&state={{ state }}&onlinejudge=\*&difficulty={{ difficulty }}&search={{ search }}">  All  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=ZJ&difficulty={{ difficulty }}&search={{ search }}">  ZeroJudge  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=TIOJ&difficulty={{ difficulty }}&search={{ search }}">  TIOJ  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=LeetCode&difficulty={{ difficulty }}&search={{ search }}">  LeetCode  </option>  </select>  </div>  <div class="col-3">  <select class="form-select" id="difficulty"  onchange="location.href=this.options[this.selectedIndex].value">  <option selected>難度</option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=\*&search={{ search }}">  All  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Hard&search={{ search }}">  Hard  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Normal&search={{ search }}">  Middle  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Easy&search={{ search }}">  Easy  </option>  </select>  </div>  <div class="col-3">  <form class="d-flex" action="" method="get">  <input type="hidden" name="page" value="1">  <input type="hidden" name="state" value="{{ state }}">  <input type="hidden" name="onlinejudge" value="{{ onlinejudge }}">  <input type="hidden" name="difficulty" value="{{ difficulty }}">  <input class="form-control me-2" name="search" placeholder="Search" aria-label="Search" id="search" value="{{ search }}">  </form>  </div>  </div>  <!-- ------- -->  <!-- 題目列表 -->  <!-- ------- -->  <div class="table-responsive">  <table id="problem\_list" class="table table-hover justify-content-center mt-4">  <thead>  <tr>  <td>通過</td>  <td>OJ</td>  <td>編號</td>  <td>名稱</td>  <td>難度</td>  <td>標籤</td>  <td>通過率</td>  <td>上傳日期</td>  {% if (session['Permission']=='Manager') %}  <td>修改</td>  <td>刪除</td>  {% else %}  {% endif %}  </tr>  </thead>  <tbody>  {% for item in data %}  <tr onclick="window.location='./problem?problem\_id={{item[0]}}'">  <td class="text-success">  {% if item['passed'] %}  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"  class="bi bi-check-lg" viewBox="0 0 16 16">  <path  d="M12.736 3.97a.733.733 0 0 1 1.047 0c.286.289.29.756.01 1.05L7.88 12.01a.733.733 0 0 1-1.065.02L3.217 8.384a.757.757 0 0 1 0-1.06.733.733 0 0 1 1.047 0l3.052 3.093 5.4-6.425z" />  </svg>  {% endif %}  </td>  <td>  {% if item[0].split('-')[0] == 'ZJ' %}  Zero<br>Judge  {% elif item[0].split('-')[0] == 'TIOJ' %}  TIOJ  {% endif %}  </td>  <td>{{ item[0] }}</td> <!--編號-->  <td>{{ item[1] }}</td> <!--名稱-->  <td>{{ item[7] }}</td> <!--難度-->  <td>{{ item[8] }}</td> <!--標籤-->  <td>  {{ item[12] }}<!--通過率-->  </td>  <td>{{ item[11].strftime('%Y-%m-%d') }}</td>  {% if (session['Permission']=='Manager') %}  <!-- 修改 -->  <td><a class="btn btn-success" href="/manager/problem?problem\_id={{item[0]}}" role="button"><svg  xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"  class="bi bi-pencil" viewBox="0 0 16 16">  <path  d="M12.146.146a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1 0 .708l-10 10a.5.5 0 0 1-.168.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0 0 1 .11-.168zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207zm1.586 3L10.5 3.207 4 9.707V10h.5a.5.5 0 0 1 .5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293zm-9.761 5.175-.106.106-1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0 1-.5-.5V11h-.5a.5.5 0 0 1-.468-.325" />  </svg></a></td>  <!-- 刪除 -->  <td><a class="btn btn-danger" href="/manager/delete?problem\_id={{item[0]}}" role="button"><svg xmlns="http://www.w3.org/2000/svg"  width="16" height="16" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">  <path  d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5m2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5m3 .5a.5.5 0 0 0-1 0v6a.5.5 0 0 0 1 0z" />  <path  d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1 1 0 0 1 1 1h3.5a1 1 0 0 1 1 1zM4.118 4 4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 1-1V4.059L11.882 4zM2.5 3h11V2h-11z" />  </svg></a></td>  {% else %}  {% endif %}  </tr>  {% endfor %}  </tbody>  </table>  </div>  <!-- 題目列表分頁 -->  <div class="d-flex justify-content-center">  <nav aria-label="Page navigation">  <ul class="pagination">  {% if page > 1 %}  <li class="page-item"><a class="page-link"  href="?page=1&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">第一頁</a>  </li>  {% else %}  <li class="page-item disabled"><span class="page-link">第一頁</span></li>  {% endif %}  {% for p in range(start\_page, end\_page) %}  {% if p == page %}  <li class="page-item active" aria-current="page"><span class="page-link">{{ p }}</span></li>  {% else %}  <li class="page-item"><a class="page-link"  href="?page={{p}}&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">{{  p }}</a></li>  {% endif %}  {% endfor %}  {% if page < end\_page-1 %} <li class="page-item disabled"><a class="page-link" href="#">...</a></li>  <li class="page-item"><a class="page-link"  href="?page={{page+1}}&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">下一頁</a>  </li>  {%else%}  <li class="page-item disabled"><span class="page-link">下一頁</span></li>  {%endif%}  </ul>  </nav>  </div>  </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\problem.html |
| 用途 | 題目作答頁面前端 |
| 部分程式碼 | |
| <div class="container-md">          <div class="row">              <div class="col">                  <div class="row">                      <div class="col-6">                          <div id="problem\_id" style="display: none;">{{data[0][0]}}</div>                          <h1 class="display-6 mt-4">{{data[0][0]}}<br>{{data[0][1]}}</h1>                      </div>                      <!-- 收藏按鈕 -->                      <div class="col-6 d-flex justify-content-center align-items-center">                          <button class="Btn">                              <span class="leftContainer">                                  <svg fill="white" viewBox="0 0 512 512" height="1em" xmlns="http://www.w3.org/2000/svg">                                      <path                                          d="M47.6 300.4L228.3 469.1c7.5 7 17.4 10.9 27.7 10.9s20.2-3.9 27.7-10.9L464.4 300.4c30.4-28.3 47.6-68 47.6-109.5v-5.8c0-69.9-50.5-129.5-119.4-141C347 36.5 300.6 51.4 268 84L256 96 244 84c-32.6-32.6-79-47.5-124.6-39.9C50.5 55.6 0 115.2 0 185.1v5.8c0 41.5 17.2 81.2 47.6 109.5z">                                      </path>                                  </svg>                                  <span class="like">Like</span>                              </span>                              <span class="likeCount">                                  {{like}}                              </span>                          </button>                      </div>                      <!-- 分頁 -->                      <div class="nav nav-tabs nav-justified">                          <button class="nav-link active" data-bs-toggle="tab" data-bs-target="#menu1"                              aria-selected="true">題目</button>                          <button class="nav-link" data-bs-toggle="tab" data-bs-target="#menu2"                              aria-selected="false">解題影片</button>                      </div>                      <div class="tab-content">                          <!-- 題目 -->                          <div class="tab-pane fade show active" id="menu1">                              <div class="col-12">                                  <h3 class="mt-4">題目說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.2|safe}}</h4>                                  <h3 class="mt-4">輸入說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.3|safe}}</h4>                                  <h3 class="mt-4">輸出說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.4|safe}}</h4>                              </div>                              {% for n in range(example\_inputs|length) %}                              <h5 class="mt-4 overflow-auto">範例{{n+1}}</h5>                              <div class="border-start border-4">                                  <h6 class="ms-2 mt-4 overflow-auto">輸入</h6>                                  <h6 class="ms-2 mt-4 overflow-auto code-font">{{ example\_inputs[n]|safe }}</h6>                                  <h5 class="ms-2 mt-4 overflow-auto">輸出</h5>                                  <h6 class="ms-2 mt-4 overflow-auto code-font">{{ example\_outputs[n]|safe }}</h6>                              </div>                              {%endfor%}                          </div>                          <!-- 教學影片 -->                          <div class="tab-pane fade" id="menu2">                              <div class="col-12">                                  <iframe width="560" height="315"                                      src="https://www.youtube.com/embed/nZqMjBUp3Fs?si=\_5ERs1-g2\_hBQgmT"                                      title="YouTube video player" frameborder="0"                                      allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"                                      referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>                              </div>                          </div>                      </div>                  </div>              </div>              <div class="col mt-5">                  <div class="row d-flex justify-content-center">                      <div class="btn-group" role="group" aria-label="Basic example">                          <div class="btn-group">                              <button type="button" class="btn btn-primary dropdown-toggle" data-bs-toggle="dropdown"                                  aria-expanded="false">                                  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                                      class="bi bi-code-square" viewBox="0 0 16 16">                                      <path                                          d="M14 1a1 1 0 0 1 1 1v12a1 1 0 0 1-1 1H2a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1zM2 0a2 2 0 0 0-2 2v12a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V2a2 2 0 0 0-2-2z" />                                      <path                                          d="M6.854 4.646a.5.5 0 0 1 0 .708L4.207 8l2.647 2.646a.5.5 0 0 1-.708.708l-3-3a.5.5 0 0 1 0-.708l3-3a.5.5 0 0 1 .708 0m2.292 0a.5.5 0 0 0 0 .708L11.793 8l-2.647 2.646a.5.5 0 0 0 .708.708l3-3a.5.5 0 0 1 0-.708l-3-3a.5.5 0 0 0-.708 0" />                                  </svg>                                  語言                              </button>                              <ul class="dropdown-menu" id="select\_language\_button">                                  <li><a class="dropdown-item" data-mode="python" value="python">Python</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-csrc" value="c">C</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-c++src" value="cpp">C++</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-java" value="java">Java</a></li>                              </ul>                          </div>                          <button type="button" class="btn btn-outline-primary" id="test\_btn" data-bs-toggle="modal"                              data-bs-target="#staticBackdrop">                              <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                                  class="bi bi-code-slash" viewBox="0 0 16 16">                                  <path                                      d="M10.478 1.647a.5.5 0 1 0-.956-.294l-4 13a.5.5 0 0 0 .956.294zM4.854 4.146a.5.5 0 0 1 0 .708L1.707 8l3.147 3.146a.5.5 0 0 1-.708.708l-3.5-3.5a.5.5 0 0 1 0-.708l3.5-3.5a.5.5 0 0 1 .708 0m6.292 0a.5.5 0 0 0 0 .708L14.293 8l-3.147 3.146a.5.5 0 0 0 .708.708l3.5-3.5a.5.5 0 0 0 0-.708l-3.5-3.5a.5.5 0 0 0-.708 0" />                              </svg>測試</button>                          <!-- Modal -->                          <!-- 進度條視窗 -->                          <div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false"                              tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">                              <div class="modal-dialog">                                  <div class="modal-content">                                      <div class="modal-header">                                          <h5 class="modal-title" id="staticBackdropLabel">測試中</h5>                                      </div>                                      <div class="modal-body">                                          <div aria-label="Orange and tan hamster running in a metal wheel" role="img"                                              class="wheel-and-hamster">                                              <div class="wheel"></div>                                              <div class="hamster">                                                  <div class="hamster\_\_body">                                                      <div class="hamster\_\_head">                                                          <div class="hamster\_\_ear"></div>                                                          <div class="hamster\_\_eye"></div>                                                          <div class="hamster\_\_nose"></div>                                                      </div>                                                      <div class="hamster\_\_limb hamster\_\_limb--fr"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--fl"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--br"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--bl"></div>                                                      <div class="hamster\_\_tail"></div>                                                  </div>                                              </div>                                              <div class="spoke"></div>                                          </div>                                      </div>                                      <div class="modal-footer">                                          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal"                                              id="stop\_btn">停止</button>                                      </div>                                  </div>                              </div>                          </div>                          <button type="button" class="btn btn-outline-success" id="upload\_btn" data-bs-toggle="modal"                              data-bs-target="#staticBackdrop"><svg xmlns="http://www.w3.org/2000/svg" width="16"                                  height="16" fill="currentColor" class="bi bi-cloud-arrow-up" viewBox="0 0 16 16">                                  <path fill-rule="evenodd"                                      d="M7.646 5.146a.5.5 0 0 1 .708 0l2 2a.5.5 0 0 1-.708.708L8.5 6.707V10.5a.5.5 0 0 1-1 0V6.707L6.354 7.854a.5.5 0 1 1-.708-.708z" />                                  <path                                      d="M4.406 3.342A5.53 5.53 0 0 1 8 2c2.69 0 4.923 2 5.166 4.579C14.758 6.804 16 8.137 16 9.773 16 11.569 14.502 13 12.687 13H3.781C1.708 13 0 11.366 0 9.318c0-1.763 1.266-3.223 2.942-3.593.143-.863.698-1.723 1.464-2.383m.653.757c-.757.653-1.153 1.44-1.153 2.056v.448l-.445.049C2.064 6.805 1 7.952 1 9.318 1 10.785 2.23 12 3.781 12h8.906C13.98 12 15 10.988 15 9.773c0-1.216-1.02-2.228-2.313-2.228h-.5v-.5C12.188 4.825 10.328 3 8 3a4.53 4.53 0 0 0-2.941 1.1z" />                              </svg>上傳</button>                      </div>                      <p></p>                      <a href="/dolos?problem\_id={{data[0][0]}}" class="btn btn-primary w-70" role="button"                          aria-disabled="true">程式碼分析</a>                      <p></p>                      <a href='./answer\_record?problem\_id={{data[0][0]}}' class="btn btn-primary w-70" role="button"                          aria-disabled="true">作答情況</a>                      <!-- 程式輸入框 -->                      <div class="row border border-3 round rounded-2 mt-5">                          <textarea id="editor" name="editor">  </textarea>                      </div>                      <div id="result\_container">                          <div id="passed" class="mt-5 bg-success p-3 text-white rounded" style="display: none;">                              <h4 id="passed\_status"></h4>                              <p id="passed\_run\_time"></p>                              <p id="passed\_memory"></p>                          </div>                          <div id="failed" class="mt-5 bg-danger p-3 text-white rounded" style="display: none;">                              <h4 id="failed\_status"></h4>                              <p id="failed\_error\_reason"></p>                          </div>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\rank.html |
| 用途 | 排行榜前端 |
| 部分程式碼 | |
| <div class="container">          <!-- ------- -->          <!-- 作答紀錄列表 -->          <!-- ------- -->          <div class="table-responsive">              <table id="problem\_list" class="table table-hover justify-content-center mt-4">                  <thead>                      <tr>                          <td>名次</td>                          <td>用戶名稱</td>                          <td>正確答題數</td>                          <td>正確率</td>                          <td>平均執行時間</td>                          <td>平均使用記憶體</td>                      </tr>                  </thead>                  <tbody>                      {% for item in data %}                      <tr>                          <td>{{ loop.index }}</td>                          <td>{% if item[2] is none %}                              <img src="./static/user\_icon/default.png" class="circle-img">                          {% else %}                              <img src="{{ item[2] }}" class="circle-img">                          {% endif %}                          {{ item[1] }}                          </td>                          <td>{{ item[3] }}</td>                          <td>{{ item[4] }}</td>                          <td>{{ item[5] }} ms</td>                          <td>{{ item[6] }} MB</td>                      </tr>                      {% endfor %}                  </tbody>              </table>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | services\auth\app.py |
| 用途 | 使用者認證相關 |
| 部分程式碼 | |
| # google登入  oauth = OAuth(app)  google = oauth.register(  name='google',  client\_id=GOOGLE\_CELENT\_ID,  client\_secret=GOOGLE\_CELENT\_SERRET,  client\_kwargs= {"scope": "openid email profile"},  server\_metadata\_url= 'https://accounts.google.com/.well-known/openid-configuration')  # 查詢電子郵件有沒有註冊過  @auth\_bp.route('/login', methods=['GET', 'POST'])  def login():  # 如果是 POST 請求，表示用戶提交了表單  if request.method == "POST":  # 從表單中獲取電子郵件  Email = request.form['Email']  # 構建 SQL 查詢命令，查詢使用者資料庫中是否存在這個電子郵件  sql\_user\_command = f"SELECT \* FROM user where email='{Email}'"  # 執行 SQL 查詢，獲取使用者資料  user\_data = db.get\_data(sql\_user\_command)  # 如果查詢結果有一條記錄，表示用戶已經註冊過  if len(user\_data) == 1:  # 將電子郵件存入 session  session['Email'] = Email  # 重定向到輸入密碼頁面  return redirect('/auth/login\_password')  # 如果用戶尚未註冊，將用戶重定向到註冊頁面  else:  return redirect('/auth/register')  else:  # 如果是 GET 請求，即用戶訪問該路由時，返回登入表單模板  return render\_template('login.html')  # Google 登入路由  @auth\_bp.route('/google')  def login\_google():  # 設置重定向 URI，用於 Google 登入完成後的跳轉  redirect\_uri = url\_for('auth.authorize', \_external=True)  # 呼叫 OAuth 库的 authorize\_redirect() 方法，將用戶導向 Google 登入頁面  return google.authorize\_redirect(redirect\_uri)  # Google 登入回調路由  @auth\_bp.route('/google-callback')  def authorize():  # 取得 Google 登入的存取令牌  token = google.authorize\_access\_token()  # 通過存取令牌向 Google API 發送請求，獲取用戶資訊  user\_info = google.get('https://www.googleapis.com/oauth2/v1/userinfo').json()  # 從用戶資訊中提取必要的數據  Goole\_ID = user\_info['id']  Email = user\_info['email']  imgs = user\_info['picture']  # 在數據庫中查詢是否有使用該郵箱註冊過  sql\_user\_command = f"SELECT \* FROM user where email='{Email}'"  user\_data = db.get\_data(sql\_user\_command)  # 如果郵箱已註冊  if len(user\_data) == 1:  # 將用戶資訊存入 session  session['User\_id']=user\_data[0][0]  session['Email'] = Email  session['logged\_in'] = True  session['User\_name'] = user\_data[0][1]  session['Permission']=user\_data[0][6]  session['google\_id'] = Goole\_ID  session['imgs'] = imgs  # 如果數據庫中的 Google ID 與 Google 返回的 ID 一致，則重定向到首頁  if user\_data[0][3] == Goole\_ID:  return redirect('/')  # 否則，重定向到連結 Google 賬戶頁面  else:  return redirect('/auth/connect\_google')  # 如果郵箱尚未註冊，則重定向到註冊頁面  else:  return redirect('/auth/register')  # 連結 Google 賬戶路由  @auth\_bp.route('/connect\_google',methods=['GET','POST'])  def connect\_google():  # 從 session 中獲取 Google ID、郵箱和用戶圖片  google\_id = session.get('google\_id')  Email = session.get('Email')  imgs = session.get('imgs')  # 如果是 POST 請求  if request.method=="POST":  # 獲取輸入的密碼  Password=request.form['Password']  # 查詢數據庫，確認密碼是否正確  sql\_common = f"SELECT \* FROM user where email='{Email}'"  user\_data = db.get\_data(sql\_common)  # 如果密碼驗證成功  if check\_password\_hash(db.get\_data(sql\_common)[0][2],Password):  # 將用戶標記為已登入  session['logged\_in'] = True  # 將用戶名稱存入 session  session['User\_name'] = user\_data[0][1]  # 將用戶 ID 存入 session  session['User\_id'] = user\_data[0][0]  session['Permission']=user\_data[0][6]  # 更新數據庫中的 Google ID 和圖片  sql\_user\_command = f"UPDATE user SET google\_id = '{google\_id}', image = '{imgs}' WHERE email = '{Email}'"  db.edit\_data(sql\_user\_command)  # 重定向到首頁  return redirect('/')  # 如果密碼驗證失敗  else:  result = '密碼錯誤'  return render\_template('connect\_google.html',result=result)  # 如果是 GET 請求，直接渲染連結 Google 賬戶頁面  return render\_template('connect\_google.html')  # 登入密碼頁面路由  @auth\_bp.route('/login\_password',methods=['GET','POST'])  def login\_password():  # 如果是 POST 請求  if request.method=="POST":  # 從 session 中獲取郵箱  Email = session.get('Email')  # 獲取輸入的密碼  Password=request.form['Password']  # 檢查是否勾選了「記住我」  try:  Rememberme=request.form['Rememberme']  Rememberme=1  except Exception:  Rememberme=0  # 查詢使用者資料  sql\_common=f"SELECT \* FROM user where email='{Email}'"  user\_data=db.get\_data(sql\_common)  # 如果密碼驗證成功  if check\_password\_hash(db.get\_data(sql\_common)[0][2],Password):  # 將用戶標記為已登入  session['logged\_in']=True  # 將用戶名稱存入 session  session['User\_id']=user\_data[0][0]  session['User\_name']=user\_data[0][1]  # 將用戶 ID 存入 session  session['User\_id']=user\_data[0][0]  # 將用戶權限存入 session  session['Permission']=user\_data[0][6]  # 如果使用者有勾選「記住我」  if Rememberme==1:  resp = make\_response(redirect('/'))  # 設置 cookie 有效期為 30 天  resp.set\_cookie('logged\_in','True',max\_age=60\*60\*24\*30)  resp.set\_cookie('user\_name',user\_data[0][1],max\_age=60\*60\*24\*30)  resp.set\_cookie('user\_id',str(user\_data[0][0]),max\_age=60\*60\*24\*30)  return resp  else:  return redirect('/')  # 如果密碼驗證失敗  else:  result='密碼錯誤'  return render\_template('login\_password.html',result=result)  else:  return render\_template('login\_password.html')  # 註冊頁面路由  @auth\_bp.route('/register' ,methods=['GET','POST'])  def register():  if request.method == "POST":  user\_name=request.form['Username']  Email=request.form['Email']  Password=request.form['Password']  # 檢查郵箱是否已經註冊過  if db.get\_data(f"SELECT \* FROM user where email='{Email}'"):  result='此Email已經註冊過'  return redirect('/auth/login')  else:  token=str(uuid.uuid4())  # 在數據庫中插入新用戶資料  sql\_user\_command=f"INSERT INTO user(user\_name,password,email,permission,uuid) VALUES ('{user\_name}','{generate\_password\_hash(Password)}','{Email}','Default user','{token}')"  db.edit\_data(sql\_user\_command)  html=f'http://140.131.114.141/auth/verify\_register?uuid={token}'  msg\_title = 'Welcome to CodeAlchemy'  msg\_recipients=[Email]  msg\_html =f'<p>親愛的 {user\_name}，<br>感謝您註冊成為我們平台的一員！為了確保您的帳戶安全，請點擊以下連結驗證您的電子郵件地址：<a href="{html}">驗證連結</a>。<br>如果您無法點擊上述連結，請將以下網址複製並粘貼到瀏覽器地址欄中：<a href="{html}">{html}</a>。<br>請完成這一步驟以啟用您的帳戶。如果您遇到任何問題或需要協助，請隨時聯繫我們的客戶服務團隊，我們將竭誠為您服務。<br>謝謝您的合作！<br>祝您有個愉快的體驗！</p>'  msg = Message(  subject=msg\_title,  sender = 'codealchemyproject@gmail.com',  recipients=msg\_recipients,  html=msg\_html,  charset='UTF-8'  )  mail.send(msg)  result='註冊成功'  return render\_template('register\_result.html',result=result)  else:  return render\_template('register.html')  # 註冊驗證頁面路由  @auth\_bp.route('/verify\_register',methods=['GET'])  def verify\_register():  # 獲得 uuid  uuid = request.args.get('uuid',None,type=str)  # 查詢是否存在該 uuid  sql\_command=f"SELECT \* FROM user where uuid='{uuid}'"  data=db.get\_data(sql\_command)  number=random\_string()+str(data[0])  if len(data)==1:  # 更新註冊時間  sql\_command = f"UPDATE user SET register\_time = NOW() WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  # 使用 threading 並行執行 TIOJ 和 ZeroJudge 註冊  t1 = threading.Thread(target=registration.TIOJ\_registration, args=(number,))  t2 = threading.Thread(target=registration.ZeroJudge\_registration, args=(number,))  # 開始執行這兩個線程  t1.start()  t2.start()  # 等待這兩個線程執行完畢  t1.join()  t2.join()  # 執行新增帳戶操作  registration.add\_account(number, number, './crawler/account.json')  # 將 uuid 設置為 Null  sql\_command = f"UPDATE user SET uuid = Null WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  result='驗證成功'  db.edit\_data(sql\_command)  return render\_template('verify\_register.html',result=result)  else:  result='驗證失敗'  return render\_template('verify\_register.html',result=result)  # 忘記密碼頁面路由  @auth\_bp.route('/forget\_password' ,methods=['GET','POST'])  def forget\_password():  if request.method == "POST":  Email=request.form['Email']  # 獲取使用者名稱  user\_name=db.get\_data(f"SELECT \* FROM user where email='{Email}'")[0][1]  # 生成驗證用的 UUID  token=str(uuid.uuid4())  # 將 UUID 更新到資料庫中  db.edit\_data(f"UPDATE user SET uuid = '{token}' WHERE email='{Email}'")  html=f'http://140.131.114.141/auth/verify\_forget\_password?uuid={token}'  msg\_title = 'Forget CodeAlchemy Password'  msg\_recipients=[Email]  msg\_html =f'<p>親愛的{user\_name},</p><p>我們注意到您最近嘗試登入您的帳號時遇到了一些問題。如果您忘記了您的密碼，請不要擔心，我們很樂意協助您重設密碼。</p><p>請點擊以下連結以重設您的密碼：</p><a href="{html}">重設密碼</a><p>如果點擊上述連結無法正常工作，請複製並粘貼以下網址至您的瀏覽器中：</p><p>{html}</p><p>請注意，此連結將在收到此郵件後的24小時內有效。請盡快完成密碼重設流程。</p><p>如果您沒有請求重設密碼，請忽略此郵件。您的帳號安全是我們的首要關注。</p><p>如果您有任何疑問或需要進一步協助，請隨時回覆此郵件與我們聯繫。</p>'.encode('utf-8')  msg = Message(  subject=msg\_title,  sender = 'codealchemyproject@gmail.com',  recipients=msg\_recipients,  html=msg\_html  )  mail.send(msg)  return redirect('/auth/login')  else:  return render\_template('forget\_password.html')  # 忘記密碼驗證頁面路由  @auth\_bp.route('/verify\_forget\_password',methods=['GET','POST'])  def verify\_forget\_password():  # 獲得 UUID  uuid = request.args.get('uuid',None,type=str)  if request.method == "POST":  uuid=request.form['uuid']  # 檢查 UUID 是否存在  sql\_command=f"SELECT \* FROM user where uuid='{uuid}'"  data=db.get\_data(sql\_command)  if len(data)==1:  if request.method == "POST":  uuid=request.form['uuid']  Password=request.form['Password']  # 新密碼加密  Password=generate\_password\_hash(Password)  # 更新密碼並將 UUID 設置為 Null  sql\_command = f"UPDATE user SET password = '{Password}' WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  sql\_command = f"UPDATE user SET uuid = Null WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  result='變更成功'  return render\_template('verify\_forget\_password\_result.html',result=result)  else:  return render\_template('verify\_forget\_password.html',uuid=uuid)  else:  result='錯誤'  return render\_template('verify\_forget\_password\_result.html',result=result)  # 更改密碼頁面路由  @auth\_bp.route('/change\_password',methods=['GET','POST'])  def change\_password():  if request.method == "POST":  Email=session.get('Email')  sql\_common=f"SELECT \* FROM user where email='{Email}'"  oldPassword=request.form['oldPassword']  newPassword=request.form['newPassword']  renewPassword=request.form['renewPassword']  # 檢查新密碼是否一致  if newPassword==renewPassword:  # 新密碼加密  newPassword=generate\_password\_hash(newPassword)  # 檢查舊密碼是否正確  if check\_password\_hash(db.get\_data(sql\_common)[0][2],oldPassword):  # 更新密碼  sql\_common= f"UPDATE user SET password = '{newPassword}' WHERE email='{Email}'"  db.edit\_data(sql\_common)  return render\_template('change\_password\_success.html')  else:  result='密碼錯誤'  return render\_template('change\_password.html',result=result)  else:  result='密碼不一致'  return render\_template('change\_password.html',result=result)  else:  return render\_template('change\_password.html')  # 登出頁面路由  @auth\_bp.route('/logout')  def logout():  session.clear()  resp = make\_response(redirect('/'))  resp.set\_cookie('logged\_in','',expires=0)  resp.set\_cookie('user\_name','',expires=0)  resp.set\_cookie('user\_id','',expires=0)  return resp | |

|  |  |
| --- | --- |
| 路徑 | templates\login.html |
| 用途 | 查詢電子郵件是否已註冊 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">登入或註冊</h2>                          <form action="/auth/login" method="POST">                              <div class="form-floating mb-3">                                  <input type="email" class="form-control" id="floatingInput"                                      placeholder="name@example.com" name="Email" required>                                  <label for="floatingInput">Email address</label>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">送出</button>                              </div>                          </form>                          <hr class="my-4">                          <div class="d-grid mb-2">                              <a href="/auth/google" class="btn btn-light">                                  <!-- 嵌入 Google 圖示 SVG -->                                  <img src="../static/imgs/google-icon-logo.svg" alt="Google 圖示" style="width: 5%;">                                  使用Google登入                                </a>                          </div>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\login\_password.html |
| 用途 | 登入密碼頁面 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">輸入密碼</h2>                          <form action="/auth/login\_password" method="POST">                              <div class="form-floating mb-3">                                  <input type="hidden" value="{{Email}}">                                  <input type="password" class="form-control" id="floatingPassword" placeholder="Password"                                      name="Password" required>                                  <label for="floatingPassword">Password</label>                                  <div class="invalid-feedback">                                      密碼錯誤                                  </div>                                  {% if result=='密碼錯誤' %}                                  <a href="./forget\_password" class="link-primary">忘記密碼？</a>                                  {% endif %}                              </div>                              <div class="form-check mb-3">                                  <label class="form-check-label" for="rememberPasswordCheck">                                      記住我                                  </label>                                  <input class="form-check-input" type="checkbox" value="1" id="rememberPasswordCheck"                                      name="Rememberme" checked>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">登入</button>                              </div>                          </form>                          <script>                              function handleBackendResult(result) {                                  const passwordInput = document.getElementById("floatingPassword");                                  if (result === "密碼錯誤") {                                      passwordInput.classList.add("is-invalid");                                  } else {                                      passwordInput.classList.remove("is-invalid");                                  }                              }                              const result = "{{ result }}"; // Get the result from Jinja2                              handleBackendResult(result);                          </script>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\forget\_password.html |
| 用途 | 忘記密碼頁面前端 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">忘記密碼</h2>                          <form action="/auth/forget\_password" method="POST">                              <div class="form-floating mb-3">                                  <input type="email" class="form-control" id="floatingInput"                                      placeholder="name@example.com" name="Email" required>                                  <label for="floatingInput">Email address</label>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">送出</button>                              </div>                          </form>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 |  |
| 用途 |  |
| 部分程式碼 | |
|  | |

第十章 測試模型

第十一章 操作手冊

第十二章 使用手冊