國立臺北商業大學

資 訊 管 理 系

114資訊系統專案設計

**系統手冊**



**組 別：第114403組**

**題 目：欸！愛多益**

**指導老師：林宏仁老師**

**組 長：11146077林若庭**

**組 員：11046090李明隆 11146063范旖旎**

**11146071何玗柔 11146072林志恆**

**中華民國114年5月28日**

**目錄**

[第1章 前言 6](#_Toc196087704)

[1-1 背景介紹 6](#_Toc196087705)

[1-2 動機 6](#_Toc196087706)

[1-3 系統目的與目標 6](#_Toc196087707)

[1-4 預期成果 8](#_Toc196087708)

[第2章 營運計畫 8](#_Toc196087709)

[2-1 可行性分析 8](#_Toc196087710)

[2-2 商業模式－Business model 9](#_Toc196087711)

[2-3 市場分析－STP 9](#_Toc196087712)

[2-4 競爭力分析SWOT-TOWS 10](#_Toc196087713)

[第3章 系統規格 11](#_Toc196087714)

[3-1 系統架構 11](#_Toc196087715)

[3-2 系統軟、硬體需求與技術平台 12](#_Toc196087716)

[3-3 使用標準與工具 13](#_Toc196087717)

[第4章 專案時程與組織分工 14](#_Toc196087718)

[4-1 專案時程 14](#_Toc196087719)

[4-2 專案組織與分工 15](#_Toc196087720)

[4-3 上傳GitHub紀錄 15](#_Toc196087721)

[第5章 需求模型 15](#_Toc196087722)

[5-1 使用者需求 15](#_Toc196087723)

[5-2 使用個案圖(Use case diagram) 19](#_Toc196087724)

[5-3 使用個案描述 19](#_Toc196087725)

[5-4 分析類別圖(Analysis class diagram)，甚至分析物件圖(Analysis object diagram)。 24](#_Toc196087726)

[第6章 設計模型 24](#_Toc196087727)

[6-1 循序圖(Sequential diagram)或通訊圖(Communication diagram)。 24](#_Toc196087728)

[6-2 設計類別圖(Design class diagram) 24](#_Toc196087729)

[第7章 實作模型 25](#_Toc196087730)

[7-1 佈署圖(Deployment diagram)。 25](#_Toc196087731)

[7-2 套件圖(Package diagram)。 25](#_Toc196087732)

[7-3 元件圖(Component diagram)。 25](#_Toc196087733)

[7-4 狀態機(State machine)，甚至時序圖(Timing diagram)。 25](#_Toc196087734)

[第8章 資料庫設計 25](#_Toc196087735)

[8-1 資料庫關聯表：需註明參考關係及限制(Constraints)。 25](#_Toc196087736)

[8-2 表格及其Meta data。 25](#_Toc196087737)

**圖目錄**

圖 2-2‑1商業模式圖 10

圖 3-1‑1系統架構圖 12

圖 3-1‑2系統功能架構圖 13

圖 4-1‑1專案時程甘特圖 16

圖 4-3‑1上傳GitHub紀錄 16

圖 5-2‑1使用個案圖 20

圖 5-3‑1註冊之活動圖 20

圖 5-3‑2登入之活動圖 21

圖 5-3‑3忘記密碼之活動圖 21

圖 5-3‑4修改密碼之活動圖 22

圖 5-3‑5測驗之活動圖 22

圖 5-3‑6查看學習報告之活動圖 23

圖 5-3‑7購買點數之活動圖 23

圖 5-3‑6使用點數生成文章之活動圖 24

圖 5-3‑6管理員管理題庫之活動圖 24

圖 6-2‑1設計類別圖 25

**表目錄**

表 1-3‑1本系統與其他學習平台之比較 8

表 2-4‑1 SWOT分析 11

表 2-4‑2 TOWS策略 11

表 3-2‑1 系統軟體需求表 13

表 3-2‑2 系統硬體需求表 14

表 3-2‑3 技術平台表 14

表 3-3‑1 使用工具表 14

# 前言

## 背景介紹

隨著數位學習的興起，越來越多學習者選擇透過線上平台提升語言能力。多益（TOEIC）測驗作為國際標準的英語能力評估工具，廣泛應用於求職與學術認證。然而，傳統的學習方式往往缺乏個人化，且學習者難以有效追蹤進度與掌握弱點。因此，建置一個專門針對多益學習的線上平台，能夠結合個人化推薦、即時測驗與數據分析，將有效提升學習效率。

## 動機

目前市場上的多益學習資源多以傳統書籍、線上影片課程或固定題庫的形式呈現，這些方式雖然提供基本的學習內容，但缺乏互動性與個人化學習機制，導致學習者難以找到弱點並根據自身需求有效學習。此外，多數學習者在準備多益考試時，往往缺乏系統化的進度追蹤與錯題分析機制，使得學習過程變得較為低效。因此，本系統的開發目標是透過數據分析與智慧推薦機制，幫助學習者有計畫地準備多益考試，並透過即時反饋提升學習成效。

## 系統目的與目標

近年來，隨著 TOEIC（多益）測驗在國際間的普及，許多學習者希望透過線上學習平台來提升應試能力。然而，目前市面上的 TOEIC 學習系統多以單一方式提供學習內容，例如單純的單字記憶、模擬測驗，或是社群討論，缺乏針對個別使用者需求的個人化學習機制。

本系統的開發目標如下：

1. 提供 AI 個人化學習體驗：根據使用者學習狀況，推薦適合的練習題目與學習計畫，提高學習效率。
2. 結合模擬測驗與即時回饋：使用者可進行多益模擬測驗，系統即時提供成績分析與建議。
3. 支援學習提醒與進度追蹤：透過 LINE 通知或 Email 提醒，幫助使用者維持學習習慣。
4. 創造點數機制：利用點數機制，讓使用者可依自身興趣生成AI文章測驗，提高作答及使用意願。

表 1-3‑1本系統與其他學習平台之比較

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能 | 本系統 | Duolingo | Memrise | OurScool |
| 學習內容 | 針對TOEIC設計，個性化推薦內容 | 多語言選擇，偏向日常語言學習 | 主要為單字記憶 | 多語言選擇，不同主題授課 |
| 模擬測驗 | 與正式TOEIC相近方式進行測驗 | 單元制，無完整測驗 | 以單字測驗為主 | 單字、文法選擇題 |
| 即時回饋 | 錯誤分析與學習建議 | 即時提供答案 | 提供學習建議 | 即時提供解答與說明 |
| 學習提醒 | LINE / Email 通知 | APP內部提醒 | APP內提醒 | 官方Line通知 |
| 適用對象 | TOEIC應試者 | 語言學習者 | 加強單字學習者 | 語言學習者 |
| 成本 | 免費/付費買點數 | 免費/付費高級版 | 免費/付費高級版 | 簽約式，每月繳費 |

## 預期成果

透過本系統的開發與實際應用，預期能夠幫助學習者提升多益應試能力，並透過個人化學習機制提高學習效率。使用者將能夠透過系統的學習建議與測驗回饋，掌握自身的學習進度，並針對弱點進行強化練習。此外，透過數據分析與視覺化的學習紀錄，學習者將能夠更有計畫地準備考試，提升學習的主動性與積極性。整體而言，本系統將為學習者提供一個高效、智能且個人化的多益學習環境，幫助他們更輕鬆地達成目標。

# 營運計畫

## 可行性分析

本系統的開發與推廣需要考量技術、經濟、操作及市場可行性，以確保能夠成功運行並達到預期目標。

技術可行性：

1. 本系統使用Django作為後端框架，確保系統穩定性與可擴展性。
2. 前端採用HTML、CSS、JavaScript，結合AI技術 提供個人化學習體驗。
3. 數據分析與測驗管理依靠機器學習演算法，優化學習推薦。

經濟可行性：

1. 開發成本主要為技術人員薪資、伺服器運行成本及行銷支出。
2. 採用初期免費，降低使用者進入門檻，功能升級（例：AI生成文章）則需要花費購買點數。
3. 透過企業合作（B2B）及廣告收益獲取另一收入。

操作可行性：

1. 系統採用直覺化UI設計，確保學習者能輕鬆操作。
2. 支援RWD設計，提高使用便利性。
3. 整合LINE / Email通知，提醒使用者學習進度。

市場可行性：

1. TOEIC為全球性英語考試，市場需求穩定且持續增長。
2. 針對在職人士、大學生、考試準備者提供個人化學習方案。
3. 透過數據分析，精準推送學習內容，提高用戶黏著度。

## 商業模式－Business model

圖 2-2‑1商業模式圖

## 市場分析－STP

1. 市場區隔（Segmentation）

依據學習需求與目標：將市場區分為學生、上班族、補習班學員。

依據學習方式：區分為自學者、輔導班學習者、企業內訓學習者。

1. 目標市場（Targeting）

主要目標族群：大學生、求職者、上班族，準備TOEIC考試提升競爭力。

次要目標族群：學校、補習班、企業，提供TOEIC訓練方案。

1. 市場定位（Positioning）

差異化優勢：以AI個人化推薦+TOEIC專業訓練+互動式學習機制，區隔於傳統學習平台（如 Duolingo、Memrise）。

品牌定位：「最適合應試者的AI TOEIC學習平台」，讓考生能夠快速提升TOEIC成績。

## 競爭力分析SWOT-TOWS

表 2-4‑1 SWOT分析

|  |  |
| --- | --- |
| 優勢Strengths   1. AI智能推薦，個人化學習 2. 針對TOEIC設計的專業內容 3. 模擬考試與即時反饋機制 | 劣勢Weaknesses   1. 需要大量測試數據來優化推薦系統 2. 初期品牌知名度較低 3. 需與現有競爭者競爭（如多鄰國） |
| 機會Opportunities   1. TOEIC市場需求穩定 2. 企業與學校有TOEIC教育需求 | 威脅Threats   1. 大型學習平台競爭激烈 2. 免費學習資源增多，影響付費意願 |

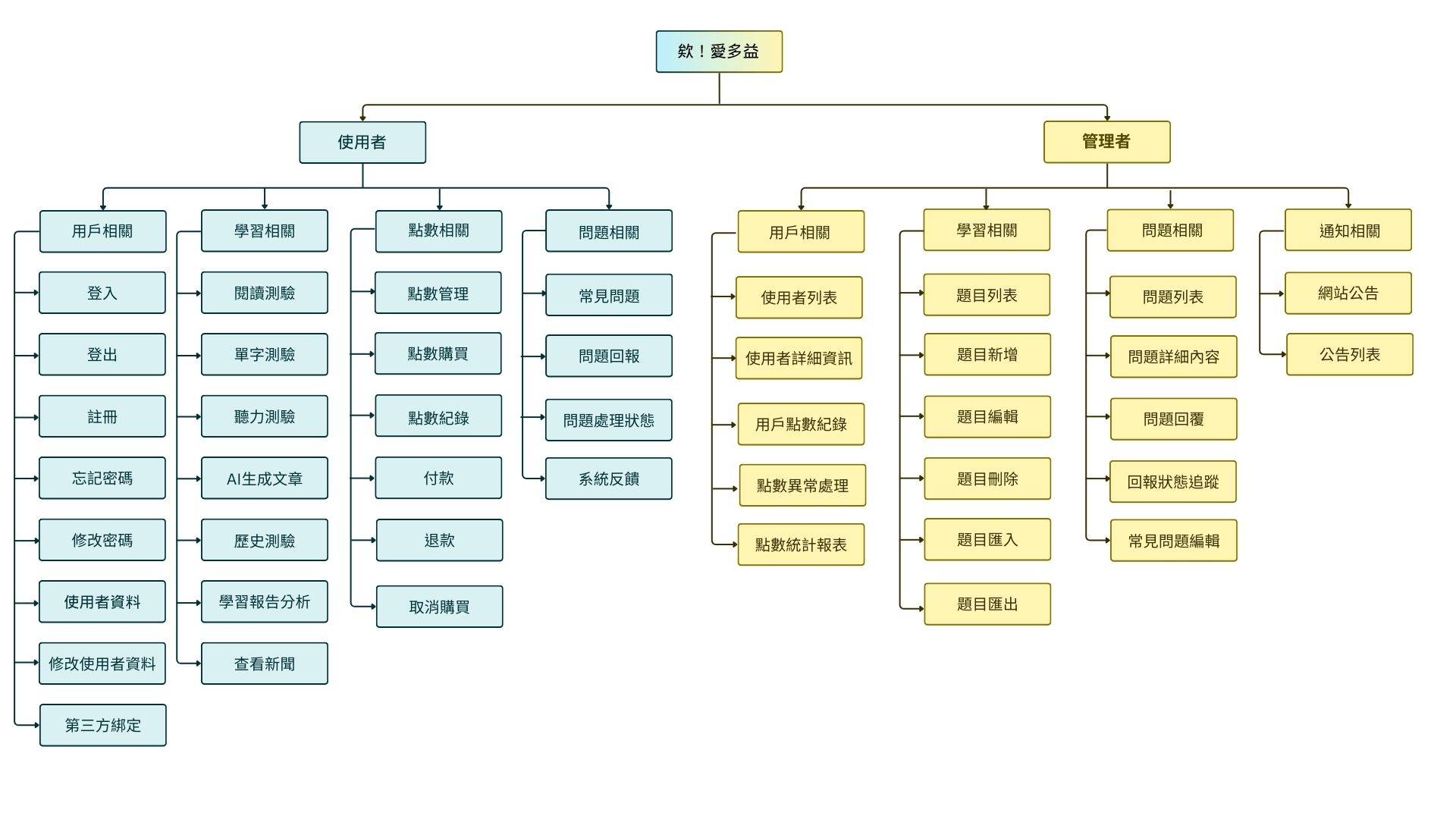
表 2-4‑2 TOWS策略

|  |  |
| --- | --- |
| **SO策略（利用優勢抓住機會）**   1. 透過AI智能學習機制，提供個人化學習建議，強化市場競爭力 2. 與TOEIC輔導班、學校合作，提供學習方案 | **WO策略（克服弱勢並抓住機會）**   1. 透過社群行銷（如YouTube、FB、IG）提高品牌知名度 2. 提供限量免費模擬測驗，再進一步吸引消費者花費購買文章 |
| **ST策略（利用優勢降低威脅）**   1. 持續更新TOEIC模擬試題，確保內容優勢 2. 強化學習回饋機制，提高用戶黏著度，降低流失率 | **WT策略（克服弱勢並降低威脅）**   1. 提供更高效的學習方法（如AI學習追蹤），對抗免費資源威脅 2. 持續優化使用者體驗，提升學習效率與吸引力 |

# 系統規格

## 系統架構

圖 3-1‑1系統架構圖

圖 3-1‑2系統功能架構圖

## 系統軟、硬體需求與技術平台

表 3-2‑1 系統軟體需求表

|  |  |
| --- | --- |
| **項目** | **說明 / 使用技術** |
| 作業系統 | 開發端：Windows 11 使用者端：Windows、macOS、Android、iOS |
| 開發工具 | Visual Studio Code |
| 程式語言 | Python 3.11（後端）、HTML / CSS / JS（前端） |
| Web 框架 | Django |
| 自動化流程工具 | n8n（流程設計與系統整合） |
| 資料庫系統 | MySQL 8.0 |
| API 介接方式 | RESTful API（使用 JSON 格式傳輸） |
| 網頁伺服器 | 開發：Django 內建伺服器 部署：Nginx + Gunicorn（預定） |
| 雲端儲存服務 | AWS S3 或 Google Cloud Storage（預定） |

表 3-2‑2 系統硬體需求表

|  |  |
| --- | --- |
| **項目** | **說明 / 規格建議** |
| 使用者端設備需求 | 一般桌機、筆電、平板、手機可正常使用 |
| 伺服器端需求 | CPU：雙核心以上 RAM：4GB 以上 儲存空間：建議 50GB |

表 3-2‑3 技術平台表

|  |  |
| --- | --- |
| **項目** | **說明 / 使用工具** |
| 雲端平台 | 預定部署於 AWS EC2 或 Google Cloud VM |
| 前端技術 | HTML、Tailwind CSS、JavaScript、Font Awesome |
| 版本管理 | 使用 GitHub |
| 整合工具 | n8n + Webhook + Django API + MySQL 資料庫整合 |

## 使用標準與工具

表 3-3‑1 使用工具表

|  |  |
| --- | --- |
| **開發輔助工具** | **工具名稱** |
| 程式碼編輯器 | Visual Studio Code |
| 版本控制工具 | Git, Github |
| 資料庫管理工具 | MySQL Client, HeidiSQL |
| API 測試工具 | Swagger |
| LLM 應用開發工具 | Langchain |
| 自動化流程工具 | n8n |
| LLM 模型 | OpenAI ChatGPT 4O |
| 製圖工具 | Medibang paint, Draw.io |
| 文件製作 | Microsoft 365 Word |
| 簡報製作 | Microsoft 365 PowerPoint, Canva |
| 溝通工具 | LINE, Email |

# 專案時程與組織分工

## 專案時程

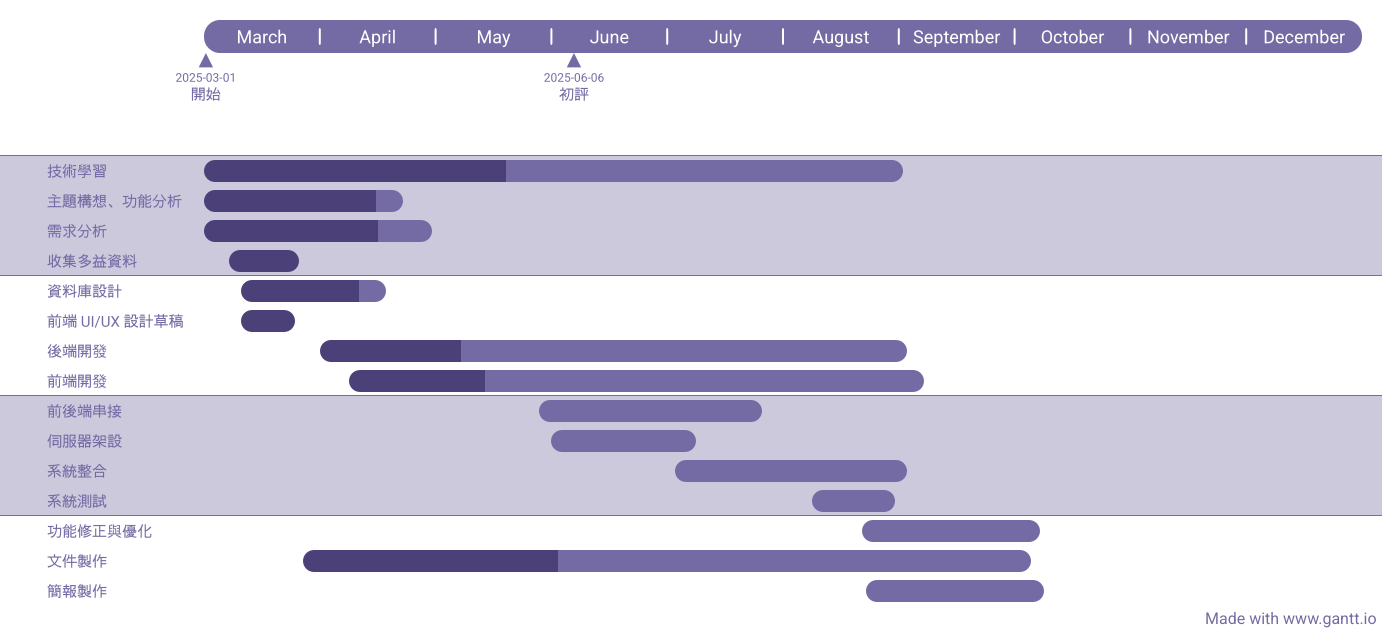


圖 4-1‑1專案時程甘特圖

## 專案組織與分工

## 上傳GitHub紀錄

11046090李明隆

11146072林志恆

11146071何玗柔

11146063范旖旎

11146077林若庭

圖 4-3‑1上傳GitHub紀錄

# 需求模型

## 使用者需求

5-1-1功能性需求：

1. 使用者端：
2. 帳號相關
   * 使用者可註冊新帳號。
   * 使用者可登入/登出帳號。
   * 使用者可查看與修改個人基本資料。
   * 使用者可變更密碼、申請忘記密碼功能。
   * 支援第三方登入（如 Google）。
3. 學習相關
   * 使用者可進行閱讀、單字、聽力測驗。
   * 使用者可查詢學習歷程與分析報告。
   * 使用者可回顧歷史錯題、收藏題目。
   * 使用者可使用 AI 產生練習文章內容。
4. 點數/交易相關
   * 使用者可購買點數並查看交易紀錄。
   * 使用者可使用點數進行測驗或兌換內容。
   * 使用者可進行付款、取消交易。
   * 點數購買後未入帳，可提出申訴處理。
5. 問題回報相關
   * 使用者可針對網站錯誤或內容錯誤提出回報。
   * 使用者可查看回報處理狀態。
6. 管理者端：
7. 使用者管理
   * 管理者可查詢／修改使用者資料。
   * 管理者可檢視使用者學習狀況與交易紀錄。
8. 題目與學習內容管理
   * 管理者可新增、修改、刪除題目。
   * 管理者可匯入／匯出題庫。
   * 管理者可設定題目分類與難易度。
9. 點數與交易管理
   * 管理者可查詢點數交易紀錄。
   * 管理者可針對特定帳號進行點數調整（如取消、補發點數）。
   * 管理者點數日誌紀錄操作人、原因及時間戳記。
   * 點數統計表可察看點數使用概況及銷售概況。
10. 問題回報管理
    * 管理者可查看所有使用者回報。
    * 管理者可回覆回報內容。
    * 管理者可維護常見問題（FAQ）。
11. 網站公告與內容管理
    * 管理者可新增與編輯首頁公告。
    * 管理者可發佈通知（如維護消息、活動資訊、更新通知）。

5-1-2非功能性需求：

1. 裝置與網路支援
   * 系統應支援主流瀏覽器（Chrome、Firefox、Safari、Edge）。
   * 使用者可透過行動裝置或桌機使用系統。
   * 系統應支援有線網路、WiFi、行動網路（3G / 4G / 5G）。
2. 系統穩定性
   * 系統應能承載高併發使用者流量（至少 XXX 使用者同時作答）。
   * 所有學習資料應即時儲存、避免資料遺失。
3. 安全性
   * 使用者密碼應加密儲存。
   * 所有交易與個資傳輸需透過 HTTPS。
   * 管理者操作應有權限驗證與記錄。

## 使用個案圖(Use case diagram)

圖 5-2‑1使用個案圖

## 使用個案描述

圖 5-3‑1註冊之活動圖

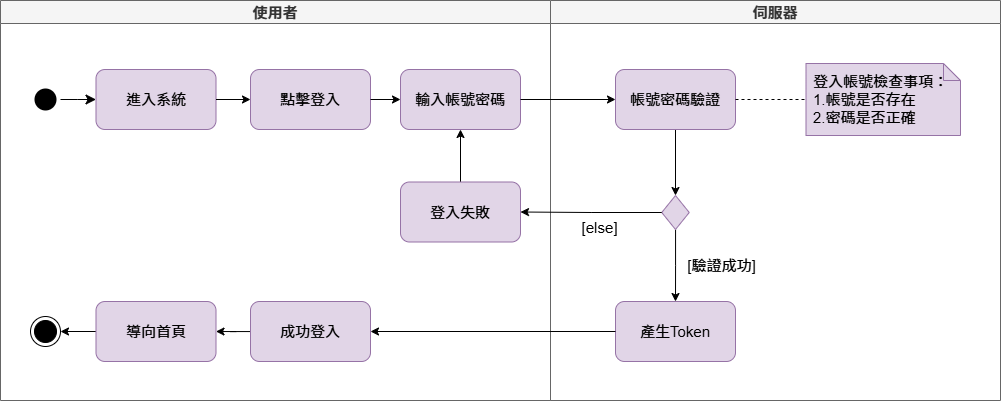
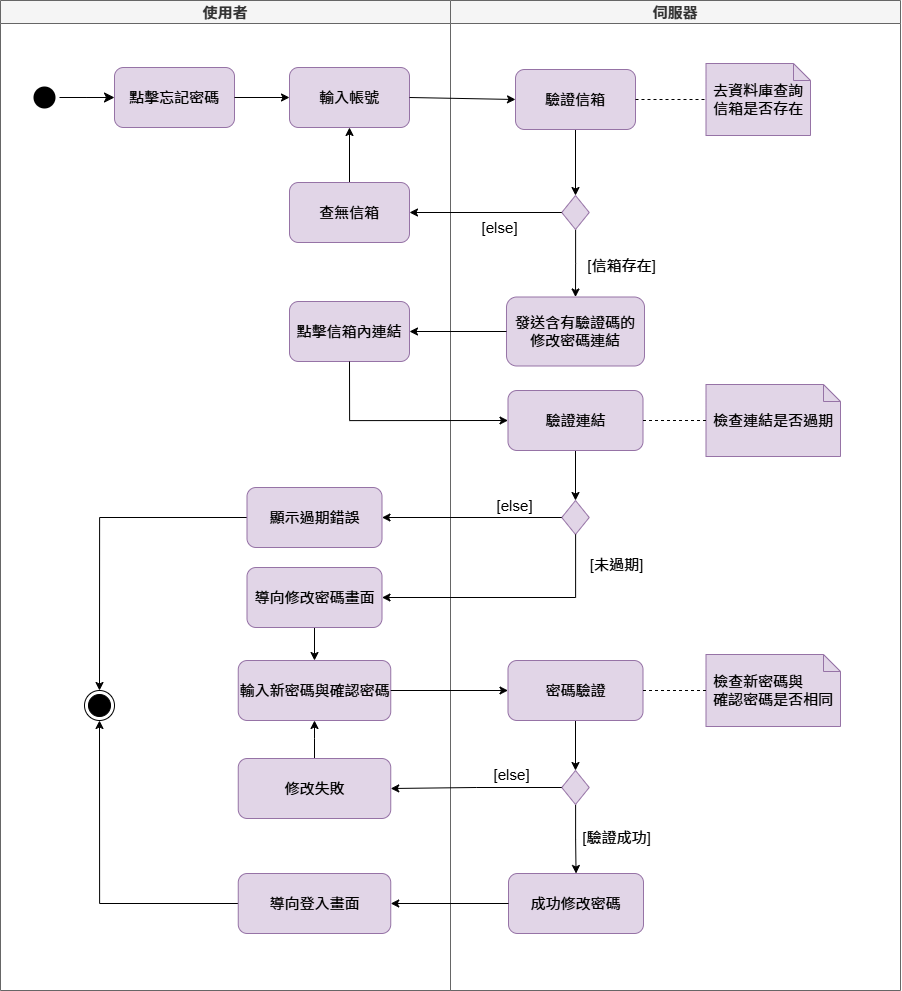
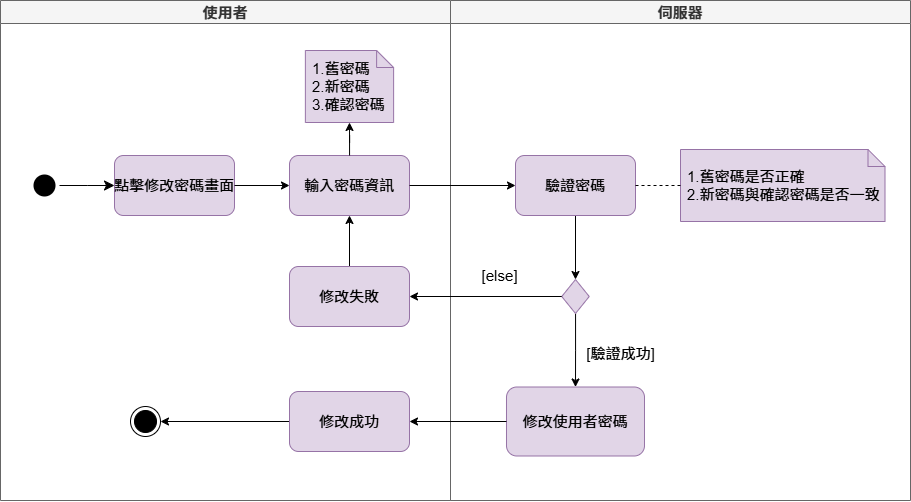
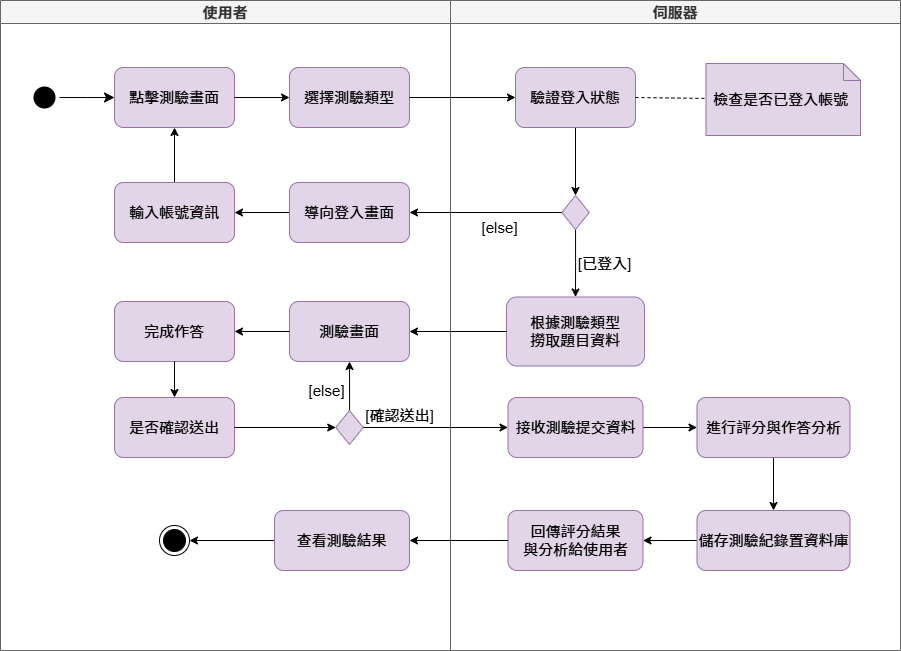
圖 5-3‑2登入之活動圖

圖 5-3‑3忘記密碼之活動圖

圖 5-3‑4修改密碼之活動圖

圖 5-3‑5測驗之活動圖

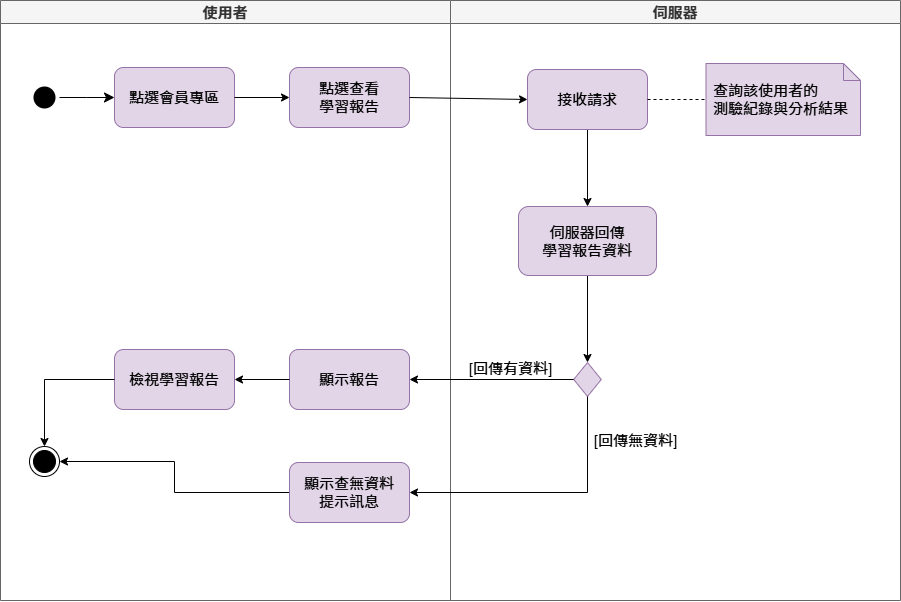
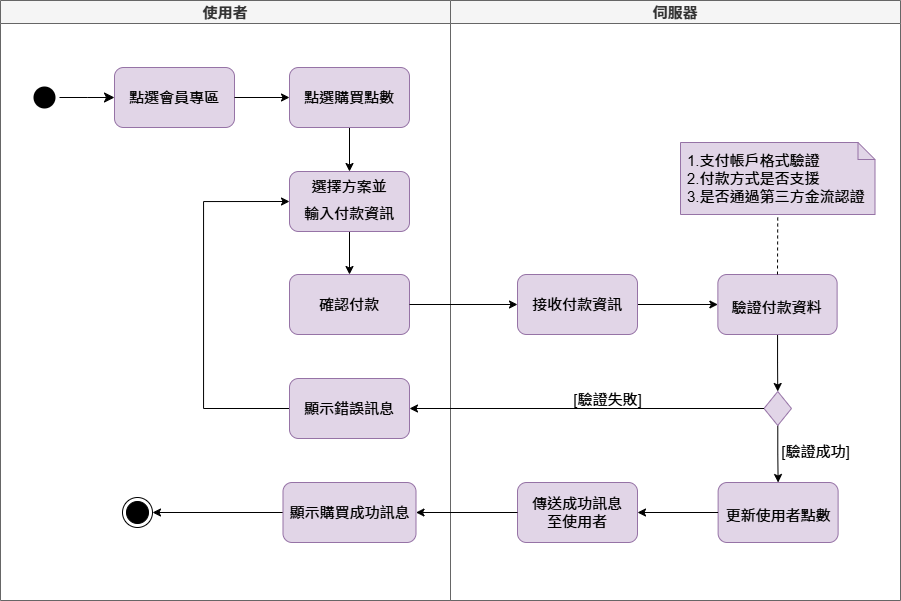
圖 5-3‑6查看學習報告之活動圖

圖 5-3‑7購買點數之活動圖

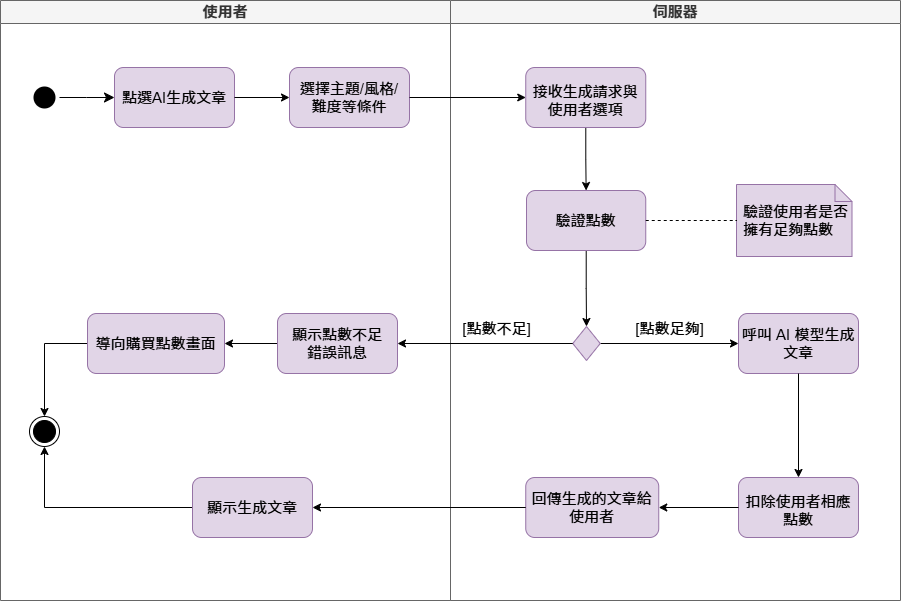
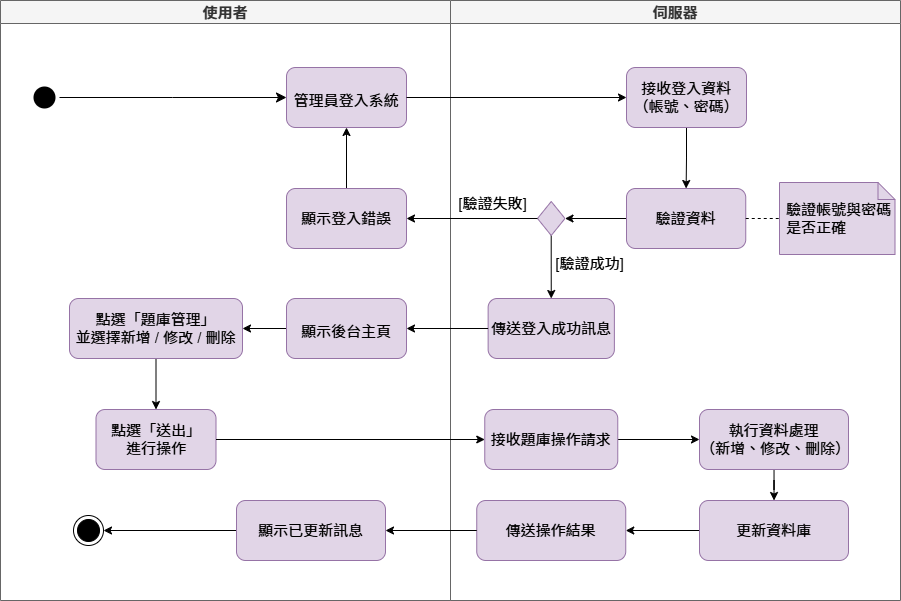
圖 5-3‑6使用點數生成文章之活動圖

圖 5-3‑6管理員管理題庫之活動圖

## 分析類別圖(Analysis class diagram)，甚至分析物件圖(Analysis object diagram)。

# 設計模型

## 循序圖(Sequential diagram)或通訊圖(Communication diagram)。

## 設計類別圖(Design class diagram)

圖 6-2‑1設計類別圖

# 實作模型

## 佈署圖(Deployment diagram)。

## 套件圖(Package diagram)。

## 元件圖(Component diagram)。

## 狀態機(State machine)，甚至時序圖(Timing diagram)。

# 資料庫設計

## 資料庫關聯表：需註明參考關係及限制(Constraints)。

## 表格及其Meta data。

**表4-2 專案組織與分工（範本）**

●主要負責人 〇次要負責人 (每一項只能有1位主要負責人，次要負責人最多2位)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 11146077林若庭 | 11046090李明隆 | 11146063范旖旎 | 11146071何玗柔 | 11146072林志恆 |
| 後端開發 | 資料庫建置 | ● |  |  |  |  |
| 伺服器架設 |  |  |  |  |  |
| Django整合 | ● | 〇 | 〇 |  |  |
| 帳號登入系統開發 | 〇 |  | ● |  |  |
| 後台管理界面設計 | ● |  |  |  |  |
| 測驗結果與成效追蹤 |  | ● |  | 〇 | 〇 |
| API 設計與開發 |  | ● |  |  | 〇 |
| 前端開發 | 首頁、登入與註冊 | 〇 |  | 〇 | ● |  |
| 模擬測驗 | 〇 |  |  | ● |  |
| 學習成果與報告 | 〇 |  |  | ● |  |
| RWD 響應式網頁設計 |  | ● |  | 〇 |  |
| 美術設計 | UI/ UX |  |  | 〇 | ● |  |
| Web介面設計 | 〇 |  |  | ● |  |
| 色彩設計 |  |  | 〇 | ● |  |
| 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 |  | 〇 | ● | 〇 |  |
| 第4章 專題時程與組織分工 | ● |  |  |  |  |
| 第5章 需求模型 | 〇 |  | 〇 |  | ● |
| 第6章 設計模型 |  | ● |  | 〇 |  |
| 第7章 實作模型 |  |  |  |  |  |
| 第8章 資料庫設計 | ● |  |  |  |  |
| 第9章 程式 |  |  | 〇 | 〇 |  |
| 第10章 測試模型 |  |  | ● |  |  |
| 第11章 操作手冊 |  |  |  | ● |  |
| 第12章 使用手冊 |  |  | ● |  |  |
| 報告 | 簡報製作 |  |  |  | ● | 〇 |

註：後端開發、前端開發及美術設計視各組專題功能新增項目，文件撰寫及報告則應固定欄位

**專題成果工作內容與貢獻度表**

**本組成員之工作內容與貢獻度(加總為100%)，請依組員人數自行調整欄位**

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容<各限100字以內> | 貢獻度 |
| 1 | 組長  林若庭 |  | % |
| 2 | 組員  李明隆 |  | % |
| 3 | 組員  范旖旎 |  | % |
| 4 | 組員  何玗柔 |  | % |
| 5 | 組員  林志恆 |  | % |
|  | | | 總計:100% |