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**



**組 別：第113205組**

**題 目：紙張小精靈paper genie系統**

**指導老師：蒯思齊老師**

**組 長：11236010 周松霆**

**組 員：11236014 康海晴 11236015 黃凱綸**

**11236023 鍾昀臻**

**中華民國113年5月29日**

**目錄**

[**第一章 前言** 1](#_Toc167815827)

[1-1 背景介紹 1](#_Toc167815828)

[1-2 動機 2](#_Toc167815829)

[1-3 系統目的與目標 2](#_Toc167815830)

[1-4 預期成果 2](#_Toc167815831)

[**第二章 營運計畫** 4](#_Toc167815832)

[2-1 可行性分析 4](#_Toc167815833)

[2-2 商業模式 5](#_Toc167815834)

[2-3 市場分析-STP 8](#_Toc167815835)

[2-4 競爭力分析 SWOT-TOWS 11](#_Toc167815836)

[**第三章 系統規格** 13](#_Toc167815837)

[3-1 系統架構 13](#_Toc167815838)

[3-2 系統軟硬體需求與技術平台： 15](#_Toc167815839)

[3-3 使用標準與工具： 16](#_Toc167815840)

[**第四章 專案時程與組織分工** 18](#_Toc167815841)

[4-1 專案時程 18](#_Toc167815842)

[4-2 專案組織與分工 19](#_Toc167815843)

[**第五章 需求模型** 22](#_Toc167815844)

[5-1 使用者需求 22](#_Toc167815845)

[5-2 使用個案圖 25](#_Toc167815846)

[5-3 使用個案描述 26](#_Toc167815847)

[5-4 分析類別圖 33](#_Toc167815848)

[**第六章 設計模型** 34](#_Toc167815849)

[6-1 循序圖 34](#_Toc167815850)

[6-2 設計類別圖 37](#_Toc167815851)

[**第七章 實作模型** 38](#_Toc167815852)

[7-1 佈署圖 38](#_Toc167815853)

[7-2 套件圖 38](#_Toc167815854)

[7-3 元件圖 39](#_Toc167815855)

[7-4 狀態機、時序圖 39](#_Toc167815856)

[**第八章 資料庫設計** 42](#_Toc167815857)

[8-1 資料庫關聯表 42](#_Toc167815858)

[8-2 表格及其Meta data 42](#_Toc167815859)

**圖目錄**

[圖2-1-1 商業模式九宮格 7](#_Toc167828072)

[圖2-3-1 STP 9](#_Toc167828110)

[圖3-1-1 系統架構圖 13](#_Toc167828221)

[圖3-1-2 系統流程圖 14](#_Toc167828222)

[圖3-2-1 2023年下半年手機銷售量 15](#_Toc167828320)

[圖 4-2-1 Github 頁面 20](#_Toc167828354)

[圖 5-2-1 使用個案圖 25](#_Toc167845304)

[圖 5-3- 1 註冊之活動圖 26](#_Toc167845233)

[圖 5-3-2 登入之活動圖 27](#_Toc167845234)

[圖 5-3-3 瀏覽、搜尋個人檔案之活動圖 28](#_Toc167845235)

[圖 5-3-4 上傳下載(公告)之活動圖 29](#_Toc167845236)

[圖 5-3-5 上傳下載(表單區)之活動圖 30](#_Toc167845237)

[圖 5-3-6 助教搜尋檔案之活動圖 31](#_Toc167845238)

[圖 5-3-7 學生上傳表單檔案審核之活動圖 32](#_Toc167845239)

[圖 6-1-1 循序圖-註冊 34](#_Toc167828428)

[圖 6-1-2 循序圖-登入 34](#_Toc167828429)

[圖 6-1-3 循序圖-學生個人檔案查詢 35](#_Toc167828430)

[圖 6-1-4 循序圖-學生公告上傳檔案 35](#_Toc167828431)

[圖 6-1-5 循序圖-學生公告下載檔案 36](#_Toc167828432)

[圖 6-1-6 循序圖-學生個人檔案搜尋 36](#_Toc167828433)

[圖 6-1-7 循序圖-學生個人檔案瀏覽 37](#_Toc167828434)

[圖 7-1-1 佈署圖 38](#_Toc167828516)

[圖 7-2-1 套件圖 38](#_Toc167828522)

[圖 7-3-1 元件圖 39](#_Toc167828527)

[圖 7-4-1 狀態機-登入 39](#_Toc167828639)

[圖 7-4- 2 狀態機-註冊 40](#_Toc167828640)

[圖 7-4-3 狀態機-上傳檔案 40](#_Toc167828641)

[圖 7-4-4 狀態機-表單檔案審核 41](#_Toc167828642)

[圖 8-1-1 資料庫關聯表 42](#_Toc167828753)

**表目錄**

[表1-4-1 系統實施比較**(期初)** 3](#_Toc167820280)

[表1-4-2 系統實施比較**(期中)** 3](#_Toc167820281)

[表1-4-3 成本比較報表 3](#_Toc167820282)

[表2-3-1 功能比較表 10](#_Toc167828983)

[表3-2-1 系統軟硬體需求-手機 15](#_Toc167828995)

[表3-3- 1 系統開發環境 17](#_Toc167829009)

[表 4-1-1 專案時程表 18](#_Toc167829026)

[表 4-2-1 專案分工表 19](#_Toc167829040)

[表 4-2-2 分工貢獻度表 21](#_Toc167829041)

# 第1章 前言

## 1-1 背景介紹

根據觀察，當前普遍的教育機構在處理學生事務方面仍普遍依賴紙本表單，如學生加退選課單和請假單。這種傳統方式不僅耗時耗力，也增錯誤的機會，並影響了教育資源的有效分配，而在企業、校園對於永續發展環保意識抬頭的趨勢下，無「紙」化漸漸成為了趨勢，根據美國環境紙張網路（Environmental PaperNetwork）及美國環保協會（Environmental DefenseFund）等非政府組織合作研究，計算出只要生產 1 萬張影印紙，就需要消耗 196,642 公升的水和砍伐 1,690 棵樹，並且產生將近 200 公斤的廢棄物，生產紙張的過程消耗的不僅僅是物料、能源成本，更是對環境的傷害。

接著，我們將焦點著重於資管系辦當前的狀況，我們發現每次新學期開始時，系辦內經常擠滿了人，進一步了解才發現，8成的學生是為了處理文件資料，而當中處理加退選文件的幾乎就佔了一半，這讓有些需要辦理即時重要文件的同學，需要長時間等待，可能會造成一些同學的困擾，於是我們開始思考是甚麼原因造成這樣的狀況，我們明明都有線上的加退選系統了。藉由詢問當事學生，我們得出了結論，當資管系學生要修資管系以外的課程時，就必須填寫紙本的加退選課單，或是一些例外狀況，也需要學生填寫紙本的單子，接著期中的退選也需要填寫紙本，這些因素的存在，就需要紙本填寫，助教花費大量時間來處理這些表單，不僅耗費人力成本，也造成了成本資源的消耗。為了提高效率並減少成本，我們提出了流程自動化的系統來優化當前的狀況。下段通過對目前流程和新系統的成本進行分析，估算實施新系統後每月能節省的成本。

以112學年學生紙本加退選課單為例，依據北商教務處註冊課務組給出的資料統計，在112學年資管系在學總人數為**544**人，這個數值包含五專部、二技部、四技部的全體學生，假設每學期學生使用紙本加退選的次數，期初有50%的學生進行加退選操作：544 \* 50% = **272人**，期中則是10%的學生進行加退選操作：544 \* 10% = 54.4 ≈ **54人，**接著假設助教每次加退選操作需要5分鐘（0.0833小時），假設助教時薪250元，期初處理時間成本：272人 \* 0.0833小時 \* 250元/小時 = 5,665元，期中處理時間成本：54人 \* 0.0833小時 \* 250元/小時 = 1,124.775元 ≈ 1,125元，總處理時間成本：5,665元 + 1,125元 = **6,790**元，接著考慮列印成本，紙張成本為0.1到0.2元，我們取平均值0.15元，一組黑色標準碳粉匣的容量可以列印約 1,500 張，換算下來黑白列印單張成本大概是 1.1 元，先不加上電費和器材折舊，這樣期初影印成本：272人 \* (0.15元 + 1.1元) = 353.6元，期中影印成本：54人 \* (0.15元 + 1.1元) = 70.2元，總影印成本為**423.8**元，使當前總成本：6,790元 + 423.8元 = **7,213.8**元，而這些操作成本發生在短短1到2周內發生，造成明顯的時間成本和工作負擔增加，是典型的高峰期（Rush Hour）問題。這種集中爆發的工作量會嚴重干擾其他日常任務的進行，對於開學時期最忙的助教，我們想極力避免這種雪上加霜的情況。

## 1-2 動機

為了解決上述提到的問題，即紙本文件繳送耗時耗力，以及有較重要問題的同學難以優先得到解決，而上述只提到**加退選課單**，資管系學生使用的紙本文件少說也有十來張，在我們看來許多學校仍然依賴於傳統的紙本管理方式進行學生事務的處理，如課程加退選、請假流程等。這種方式不僅消耗大量的紙張，而且效率低下。尤其是對於學校的行政人員和助教來說，手動處理和歸檔這些紙本文件是一個耗時且繁瑣的工作，加上助教收集完學生的紙本文件還要回送到教務處，運送過程也不排除有遺失風險。於是我們就想設計一個系統來優化這段校務流程，提高行政效率同時減少錯誤，減少紙張耗材使用保護環境，最重要的是，幫助校園節約營運的成本。

## 1-3 系統目的與目標

本系統旨在通過引入OCR（光學字符識別）技術結合生成式AI來解決傳統紙本管理方式的局限。系統的主要目標包括：

* 文檔處理：利用OCR技術自動識別和提取學生提交的紙本表單上的手寫或印刷文字，包括但不限於姓名、學號、請假事由等資訊。
* 導入生成式AI：透過OCR提取完學生紙本表單的資訊後，當模型辨識字形又誤時，使用生成式AI校正。
* 導入Line bot：Line bot結合ChatGPT，精準回應使用者問的種種問題。
* 提高行政效率：我們設計了一個流程，讓助教能減少檔案整理和修正工作，大幅提高學校行政人員和助教的工作效率。
* 降低錯誤率：生成式AI的輔助使資料識別和處理的錯誤大幅下降，確保資料的準確性。
* 推動校園數位化：為學校提供一個向數位化的平台，促進學校現代化和環保。
* 流程自動化：及時、紀錄、同步，降低出錯的機會，並且使整個文件申請過程更加流暢和透明。

## 1-4 預期成果

透過實施這個校務系統，預將達到以下成果：

* 效率顯著提升：預計行政人員和助教在處理學生事務（如課程加退選、請假等）時的工作效率將提高50%以上。
* 數據處理錯誤率降低：自動化過程將大幅減少人為錯誤，提高資料處理的準確性。
* 環保效益：減少紙張使用，促進校園環保意識，符合可持續發展的目標。
* 降低助教處理文件的工作量：幫助助教節省文件處理時間。
* 改善學生文件繳送的效率：文件繳送更快速、更便捷，減少有較重要問題的同學難以優先得到解決的情況。
* 節省成本: 我們希望透過我們系統能節省成本。(不考慮電費，其他雜費)

我們希望減少高峰期（兩周）處理成本，使加退選流程操作時間從原本5分鐘（0.0833小時）縮短為1~2分鐘（0.0167到0.0333小時）。

表1-4-1系統實施比較**(期初)**

|  |  |  |
| --- | --- | --- |
| **112學年資管(期初)** | **紙本處理** | **系統實施** |
| **總學生人數** | 544人 | 544人 |
| **50%加退選操作學生人數** | 272人 | 272人 |
| **預計每人處理時間：** | 5分鐘（0.0833小時） | 1.5分鐘（0.025小時） |
| **預計總時間成本：** | 272人 \* 0.0833小時 \* 250元/小時 = 5,665元 | 272人 \* 0.025小時 \* 250元/小時 = 1,700元 |
| **預計總影印成本：** | 272人 \* (0.15元 + 1.1元) = 353.6元 | 0元 |

表1-4-2系統實施比較**(期中)**

|  |  |  |
| --- | --- | --- |
| **112學年資管(期中)** | **紙本處理** | **系統實施** |
| **總學生人數** | 544人 | 544人 |
| **10%加退選操作學生人數** | 54人 | 54人 |
| **預計每人處理時間：** | 5分鐘（0.0833小時） | 1.5分鐘（0.025小時） |
| **預計總時間成本：** | 54人 \* 0.0833小時 \* 250元/小時 = 1,125元 | 54人 \* 0.025小時 \* 250元/小時 = 337.5元 |
| **預計總影印成本：** | 54人 \* (0.15元 + 1.1元) = 70.2元 | 0元 |

表1-4-3 成本比較報表

|  |  |  |  |
| --- | --- | --- | --- |
| **成本比較報表** | | | |
| **學生人數** | 544人 | | |
| **項目：** | **目前流程** | **新系統流程** | **成本節省 (元)** |
| **預計期初總成本** | 6,018.6元 | 1,700元 | 4,318.6元 |
| **預計期中總成本** | 1,195.2元 | 337.5元 | 857.7元 |
| **預計總成本** | 7,213.8元 | 2,037.5元 | 5,176.3元 |

我們希望在選課加退選的高峰期（兩周內），傳統紙本系統導致的高成本和大量時間投入會顯著影響助教其他工作的進行。通過引入我們設計的系統，我們預期可以大幅減少助教在高峰期的工作量，並節省約5,176.3元的成本。這不僅提高了選課加退選的效率，減少了助教的工作負擔，還降低了紙張和列印成本，對學校和環境均有良好的影響。

# 第2章 營運計畫

## 2-1可行性分析

* + 時程可行性：本組先篩選系上較常用的表單做為掃描目標。
  + 技術可行性：讓學生可以簡單掃描上傳並使用ChatGPT的技術將所掃描文件進行修正存放到資料庫。
  + 收益可行性：人工收紙本的時間大幅降低不僅省了時間也更加環保，從文章探討了企業如何通過無紙化和ESG轉型提升競爭力。
  + 時間成本可行性：在學期開始是選課尖峰時段因此會耗費大量的時間成本，自動化流程將花費時間大量減少。

以下是幾個關鍵點：

* 環境效益與成本節省： 無紙化減少紙張使用，節省購買和處理成本對環境時間皆有益。
* 提高效率：縮短處理時間，提升行政效率。
* 法遵和安全性：確保數據安全和合規性，特別是在金融服務業。

總之，無紙化和ESG轉型是提升企業運營效率和市場競爭力的有效策略，有助於達成環保與省時的目標並提升企業形象。

## 2-2商業模式

以下為本組開發系統所需的資源：

Key Partners(關鍵合作夥伴)：

* 資管系系上學生：若學生需要選課及請假。
* 系上教師與助教：教師可以線上審核假單，助教可以審核選課單。
* 本組組員：負責系統的開發和維護，確保項目按計劃進行。

Key Activities(關鍵活動)：

* 選課單審核與系上確認整體流程：原本紙本流程為填寫完畢交給系上審核再繳交至教務處。
* 請假單審核與教師確認整體流程：原本紙本流程為未附證明者須填寫完假別交給班導師做審核再繳交至學務處生輔組做後續請假流程。

Key Resources(關鍵資源)：

* 人力資源：包括開發人員和管理團隊，確保項目順利進行。
* 開發工具：必要的軟硬件資源支持開發過程。
* 伺服器空間：用於存儲和處理大量文件數據，保證系統的正常運行。

Value Propositions(主要價值)：

* 快速管理及掃描文件：提供文件處理功能，提升用戶的工作效率。
* 自動化：將審核的紙本流程透過自動化去完成。
* 花費時間減少：助教與教師省去許多紙本審閱的時間，學生也減少當下排隊等候處理的時間。
* 修正錯誤：使用ChatGPT去修正表格上的一些瑕疵與錯誤，在降低文件上的失誤率。

Customer Relationships(客戶關係)：

* 良好的使用體驗：讓師生皆可省下大量的時間並減上大量用紙。

Channels(通路)：

* 北商資管系：五專部(1-5年級)、二技部(1-2年級)、四技部(1-4年級)，及教師與助教。

Customer Segments(目標客戶)：

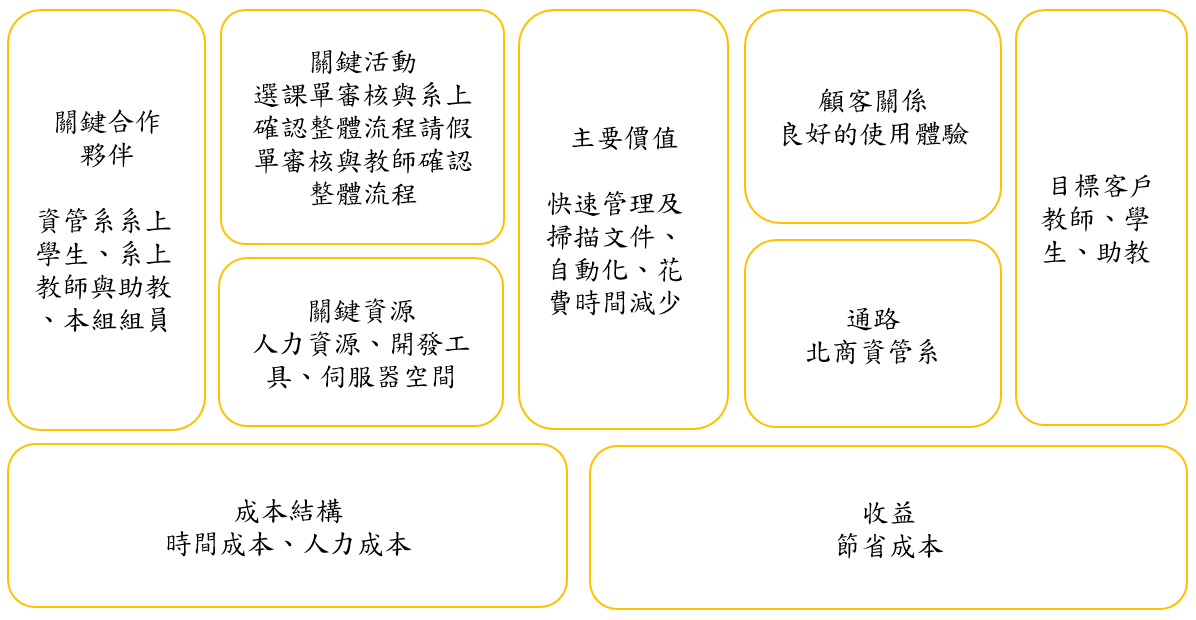
* 教師：可使用該系統可進行對學生所上傳的資料做審閱。
* 學生：可使用該系統進行資料的上傳和查閱審閱進度到哪。
* 助教：可使用該系統可進行對學生所上傳的資料做審閱。

Cost Structure(成本結構)：

* 時間成本：助教與教師停下手邊工作進行選課或請假審核則學生旁邊等候處理時間。
* 人力成本：助教與教師做審閱的動作與學生跑流程的動作。

Revenue Streams(收入來源)：

節省成本：大量降低人力成本的消耗，比如在選課週助教忙系上學生選課的事情就耗費大量時間，學生在旁邊等待也同時再浪費時間，使用系統上傳可以讓學生不用在旁邊空等，助教、教師也可以不用被打斷原本正在忙的事情。

圖2-1-1商業模式九宮格

根據以上資源，本組的目標為通過環保意識提高且工作效率提高的看法，先以資管系師生可以使用此系統，以下是一份簡單的總結，建議學校師生使用無紙化系統，除了提升環保意識外還可以增加工作效率，具體好處如下：

* 環保貢獻：透過減少紙張使用，我們能夠直接減少對自然資源的消耗。
* 提升效率：文件審批和存檔大幅縮短處理時間提高工作速度。
* 便捷存取：電子文件存取和管理，無論師生身在何處都能快速查閱和處理文件，特別適合當前追求遠程教育與工作的環境。

鑑於以上優點，我們誠摯推薦資管系的師生使用這一項系統。這不僅能提升我們的日常工作效率，同時也是向所有師生展示實踐環保理念的具體行動。

## 2-3 市場分析-STP

Segmentation 市場區隔:

* 依文件類型區分(選課單、請假單)：

主要處理假單、選課單，這是學校中最常使用的文件類型之二。學生提交請假單請假，教師和行政人員處理和存檔這些請假單。因此，假單的處理流程固定且需求量大，非常適合系統初期的應用。選課單也是學校中高頻使用的文件，學生每學期需提交選課申請，助教需審核並管理這些選課記錄。處理選課單能夠顯著提高教務工作的效率和準確性。

* 依處理複雜度區分(高/低)：

處理手寫文件，利用文件辨識技術結合ChatGPT進行資料的修正，有效地完成識別適用於假單與選課文件。

* 依處理流程(紙本繳交/非紙本繳交)：

支持將紙本文件掃描數字化，支持直接上傳電子文件便於存檔和查閱，也簡化文件提交和管理流程，適用於現代化的數字校園環境。

Targeting 目標市場(師生（教授、學生）、行政人員（助教）)：

教授、學生及助教是主要目標用戶。教師需要管理和審核學生的假單，助教需要管理及審核學生選課單，皆都需要管理、文件存檔和歸檔工作系統能夠幫助他們高效地處理和管理各類文件，學生則需經常提交

請假單和選課單。因此，這些文件的頻繁使用和處理需求使師生成為系統的重要用戶群體。

Positioning 定位:

* 簡單易用：

介面簡單、易於操作讓師生和行政人員快速上手，無需額外學習，提升用戶體驗。

* 流程設計更加準確：系統設計了精確的流程處理機制，確保處理和文件的管理更加準確，降低時交耗費與錯誤率，提升整體工作效率。
* 流程自動化：從文件掃描、修正到數據處理和查詢，所有步驟均自動完成，減少了師生和行政人員的時間花費，提升整體運營效率。

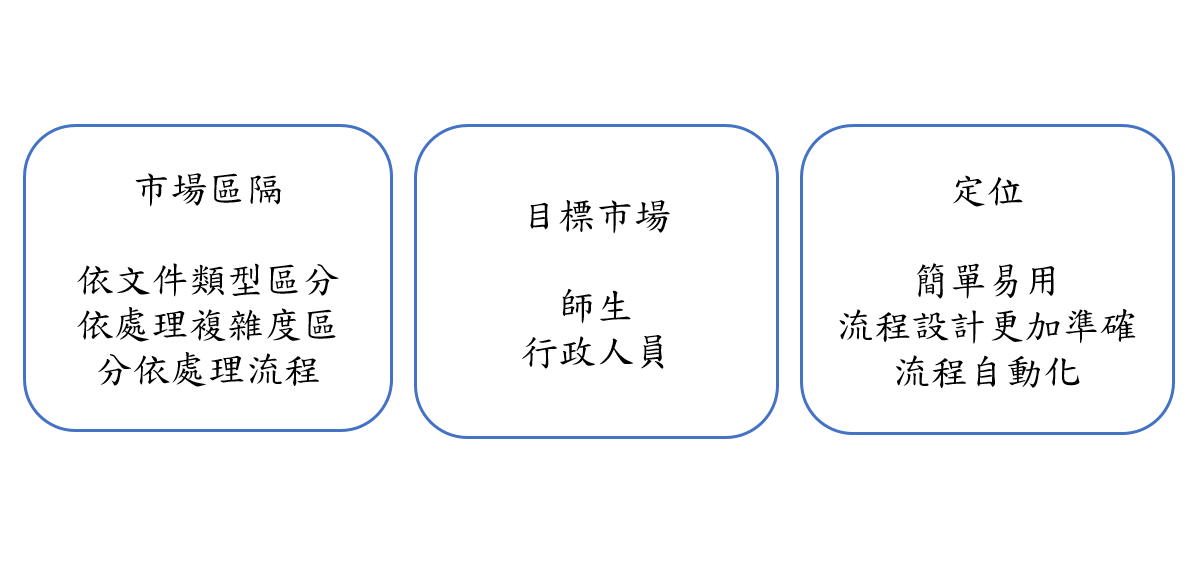


圖2-3-1 STP

市場上現有服務與本系統提供服務相似的應用程式"全能掃描王"、"Adobe Scan"，

前者有類似系統功能表

表2-3-1 功能比較表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能/系統 | 紙張小精靈paper genie系統 | 全能掃描王 | PDF Scanner |
| Logo |  | https://lh7-us.googleusercontent.com/oX4QdK-OZDob5T-cG4NyBCPGyf7uVaPmZVlHR6A0lEs-loAtVSl4KhiBRojLX02qlQPk7NeUDhSoY-dqZ4L9Yy9-bktrOsPk77CBIq0vohos-MKZKAOAhnCZbpe_ex5DBtHph_Ny6YoZ4E_ZHooSwO8 | https://lh7-us.googleusercontent.com/RRGMpMgS5eI9CeSLqdTAUqj-iBFN0kpMBAMjBN5CDycq5HNfHYbh5SxC_fE1Jvmq810oTzGElj6geH8GeqxJ8M1mDX_r3LkP5mCLncWw-btwO1lMixPxjJdkDQWIethMx2982Mk9Xr33Oy-QYk58ado |
| 跨平台使用 | ✔ | ✔ | ✔ |
| 文件編輯功能 | ✔ | ✔ | ✔ |
| 公告提醒 | ✔ | ✘ | ✘ |
| 校園公告 | ✔ | ✘ | ✘ |
| 無需訂閱 | ✔ | ✘ | ✘ |
| 校園表單整合 | ✔ | ✘ | ✘ |
| 學生資料管理 | ✔ | ✘ | ✘ |
| LineBot | ✔ | ✘ | ✘ |

從功能比較表中可以看出，紙張小精靈paper genie本系統在多個方面具有顯著優勢，特別是在校園專屬功能上。本系統不僅文件上傳功能、校園公告、無紙化流程、校園表單整合和學生資料管理等多項校園專屬功能。相較之下，全能掃描王和PDF Scanner雖然也提供了許多功能，但在校園專屬功能方面明顯不足。這使得本系統成為校園環境下的最佳選擇，能夠更好地滿足校園內部文件管理和交流的需求，提升工作效率和資料管理的便利性。

## 2-4 競爭力分析 SWOT-TOWS

優勢Strengths(S):

* 簡單易用的使用界面：本系統設計了簡單直觀的使用界面，讓師生能夠輕鬆上手，提高了使用體驗。
* 專業的學術支持：本系統提供專業的學術支持和定制化服務，滿足學校師生對文件處理的需求。

劣勢Weakness(W):

* 時間有限：所以選擇處理圖表上有限，先選擇較常使用的表單假單與選課單。

機會Opportunity(O):

* 擴大其他學系或行政單位使用：本系統自動化流程可以減少人力成本與時間耗費，可以推廣到其他系所或是行政處室提高全校使用率。
* 加強宣傳：本系統可以通過加強宣讓更多學生知道，比如新生訓練、系上活動，讓系上的學生都知道此APP提高使用率。

威脅Threats(T)

* 使用者抗拒：部分師生可能對新系統存在抗拒心理，習慣於使用傳統的紙質文件處理，導致本系統的使用率難以迅速提高。
* 數據安全：如果系統發生數據洩露師生的資料外流，會造成嚴重後果。

經過對本系統及其競爭者的詳細分析結論：

本系統在文件辨識技術上結合ChatGPT去修正內容，在省時省力層面和自動化流程層面優勢，能夠大大提升師生的使用體驗，這會使系上師生和行政人員中獲得了良好的反饋。雖然在時間有限的情況下，系統目前只能處理較常用的表單，但未來可以通過擴大使用範圍和加強宣傳來提高全校的使用率。然而，系統也面臨一些挑戰，包括使用者對新系統的抗拒和數據安全風險。總體來說，本系統在提升學校文件處理效率和減少人力成本方面有很大的潛力，但需要針對威脅制定相應的應對策略，以確保系統的順利推廣和安全運行。

# 第3章 系統規格

## 3-1 系統架構：

本系統由客戶端應用程式（App）和Line Bot結合，用戶可以通過這些渠道與系統交互。前端應用程序接收用戶輸入並進行相應處理。應用程式內部設有文件掃描辨識功能，師生可以使用該功能將文件掃描成數字和文字格式。掃描後的文件將被自動傳送至Chat GPT進行自然語言處理和智能修正。修正後的文本會返回至用戶，供其進一步使用。

此外，用戶還可以通過Line Bot進行常見問題的詢問，利用Chat GPT技術生成智能應答，幫助用戶解決問題。整個系統自動化運作，旨在減少師生在文件處理和信息查詢上花費的時間，提高工作和學習效率。

系統流程具體如下：

* 用戶通過App或上傳圖片。
* Line Bot可以進行詢問常見問題，透過ChatGPT修正並做回覆
* 上傳的圖片通過Flask API提交到後端。
* Flask API接收到圖片後，將其提交給OCR進行處理。
* OCR將圖像中的文字轉換為可編輯文本。
* Flask API接收OCR處理結果，並將文本提交給Chat GPT進行內容修正和自然語言處理，生成文本或智能應答。
* 經過處理的文本進一步轉換和分類，然後進行最終處理，並存儲或展示給用戶。

RAG流程說明 (如圖10.所示，而圖中的分類器，將會在下一個部分進行說明)

* 獲取圖像和OCR識別

首先，獲取需要處理的表單圖像，並將圖像送入OCR模型（如Google Cloud OCR或PaddlePaddle）以取得表格中的標記框，生成初步結果。

* 分類器分類表單種類：

使用訓練好的CNN模型對表單進行分類，辨識其為請假單或選課單，並輸出分類結果。

* 比對文字欄位

將OCR識別出的文字還有知識圖譜（如科目、日期、姓名等）送入ChatGPT進行欄位比對，找出需要檢查的欄位，如科目欄位。

* 比對文字內容

找到欄位之後，將該欄位的文字內容與知識圖譜送入ChatGPT進行比對，找出需檢查的文字內容，如程式設計。

* 修正文字內容：

接著，OCR辨識的字串內容、知識圖譜找到的文字內容送入ChatGPT進行比對，若比對文字內容有錯誤，則ChatGPT會進行修正。然後回到比對文字欄位步驟將OCR辨識的字串進行多次檢查，直到沒有錯誤為止。

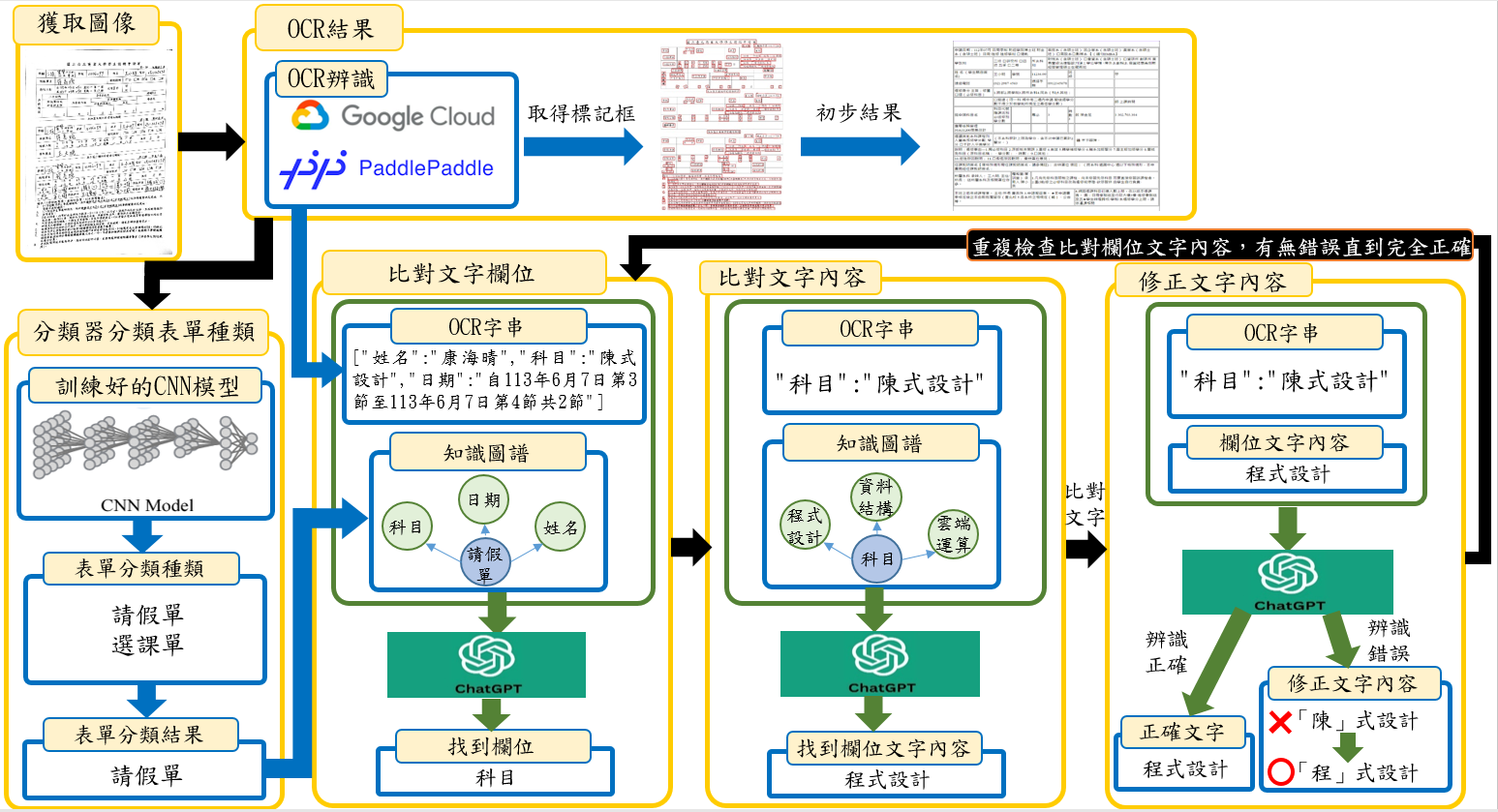


圖3-1-2 RAG流程圖

自動分類表單總覽說明

當學生將填寫好的表單上傳至系統時，系統會對表單進行分類，辨識上傳的表單是請假單還是選課單，以便後續提交給Chat GPT進行內容修正。

首先說明表單分類原理，系統中有兩種類型的表單，即請假單和選課單。我們提取表單中的標記框，並將這些標記框放置在空白的圖像上，這樣我們便得到了請假單與選課單僅包含標記框的圖像。接著，我們將請假單和選課單的標記框圖像進行比對，發現這兩種圖像的標記框排列方式不同。而當我們將不同請假單的標記框圖像進行比對時，會發現它們的標記框排列方式相同，反之亦然。

這樣一來，我們就可以通過比對標記框圖像的方式，準確地將上傳的表單分類為請假單或選課單。

以下進行流程說明

* 分類器製作流程

首先，獲取需要處理的表單圖像，並將圖像送入OCR模型（如Google Cloud OCR或Paddle OCR）以取得表格中的標記框。然後提取這些標記框的座標，並將其存儲。根據這些座標，在全白圖像上繪製標記框，生成標記框圖像。隨後，對生成的標記框圖像進行資料增強，以增加訓練數據的多樣性。最後，將增強後的標記框圖像送入CNN模型進行訓練，最終生成能夠自動分類表單的模型，如圖11.所示。

* 自動分類表單流程

整個流程從獲取需要處理的表單圖像開始。首先，將這些圖像輸入OCR模型（如Google Cloud OCR或Paddle OCR）進行識別，提取表格中的標記框座標。接著，根據這些座標在全白圖像上繪製標記框，生成標記框圖像。這部分流程與分類器製作的前半部分相似。

在生成標記框圖像後，進入後半部分流程。我們使用已訓練好的CNN模型對請假單或選課單的標記框圖像進行分類。通過比對標記框圖像與表單模型，計算每個表單屬於請假單或選課單的概率，最終輸出分類結果，確定表單類型，如圖12.所示。

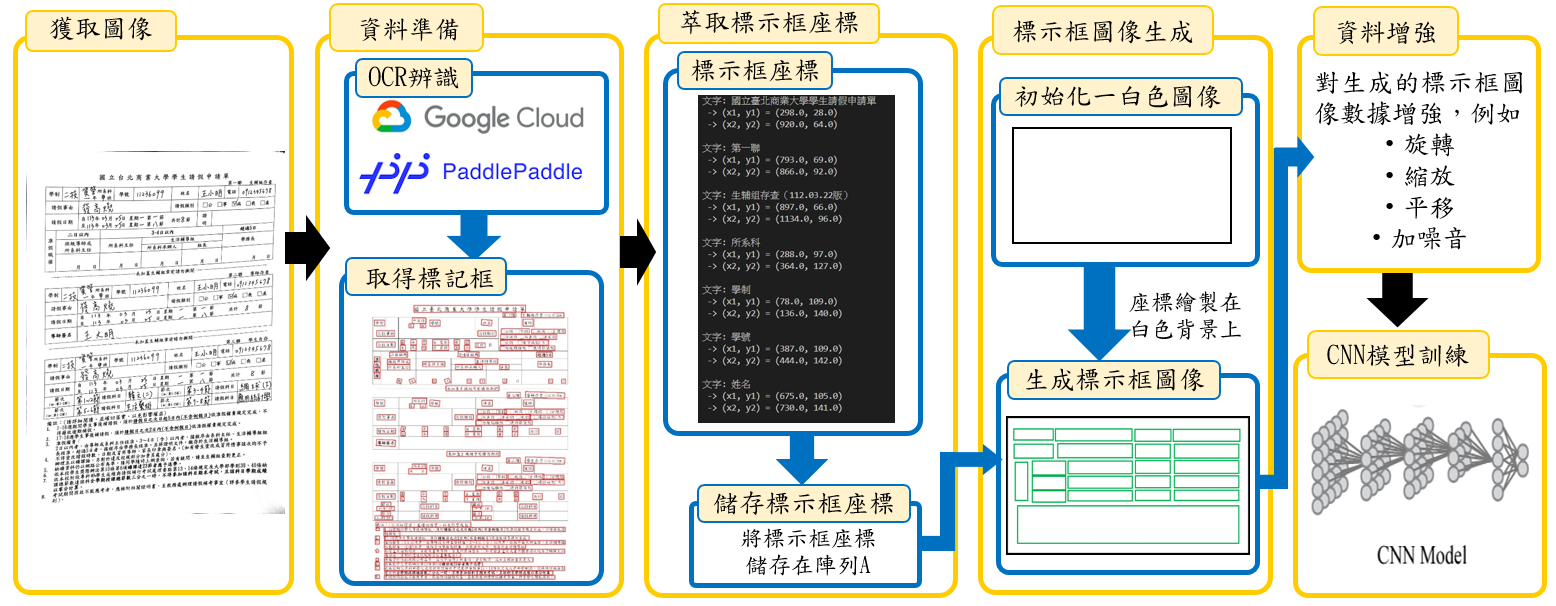


圖 11. 分類器製作流程圖

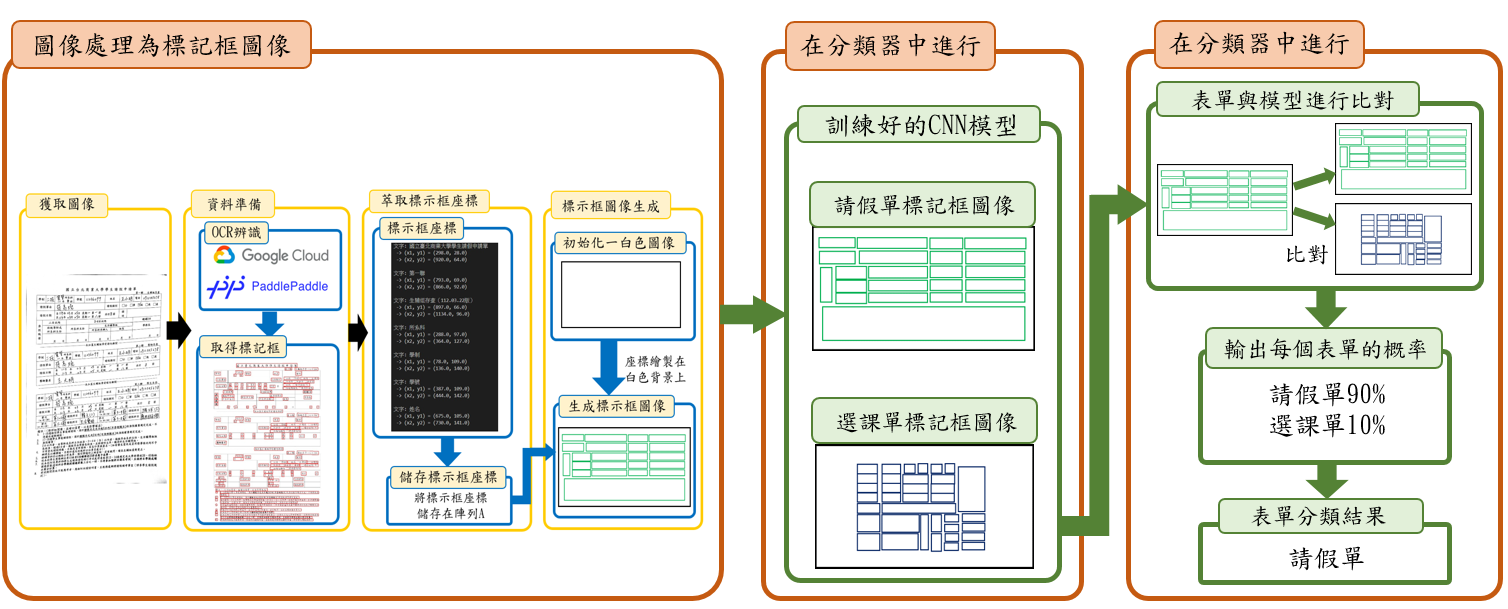


圖 12. 自動分類表單流程圖

表單自動辨識與內容修正流程說明

* 獲取圖像：使用者首先拍攝一張表單的照片並上傳到系統中。
* OCR辨識：系統使用Google Cloud OCR與Paddle OCR技術對上傳的圖像進行文字識別，並且取得表單中的文字框（欄位）資料。
* 比對表單欄位：系統會將OCR結果中的文字框位置與標準表單進行比對，透過孿生神經網路來比較使用者上傳的表單與標準樣本的相似度，判斷表單類型，

如果結果相似，則表示表單類型一致，進行下一步。

* 比對文字內容：在判斷表單類型後，系統會將OCR提取的表單文字內容與知識圖譜中的相關資料進行比對，驗證內容的正確性。例如，「科目名稱」、「日期」等欄位的資料將對照知識庫中的標準數據，檢查是否有誤。
* 修正文字內容：如果發現OCR識別的內容不正確，系統會進一步將不正確的文字內容與標準資料進行比對和修正。ChatGPT會根據比對結果自動進行欄位內容的糾正，直到修正無誤為止。
* 後端自動化處理：修正後的表單內容會自動進行處理，無需再次取出紙本進行檢查或處理。這樣大幅減少了紙本文件的重複使用，達成無紙化的效益。

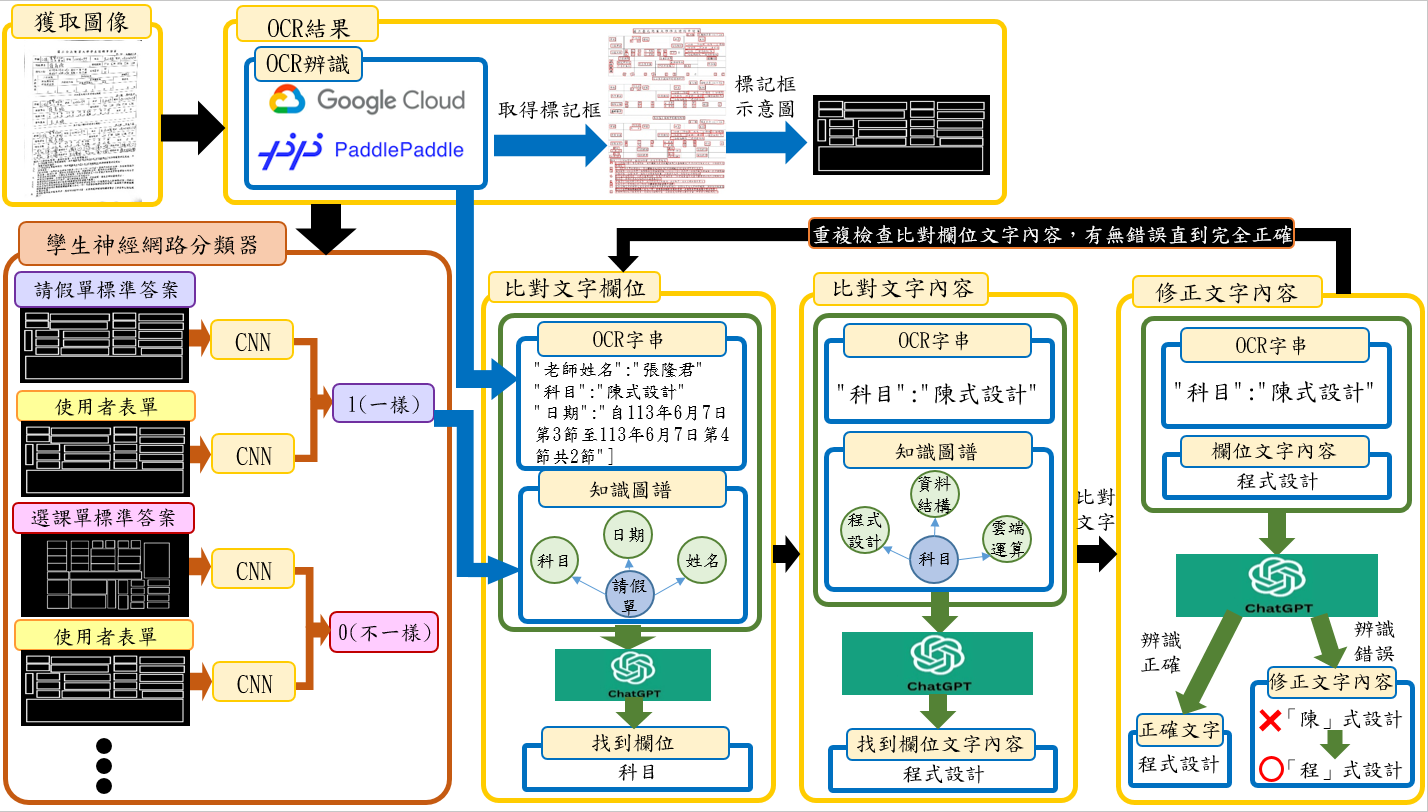


圖 表單自動辨識與內容修正流程圖

LINE Bot 與 ChatGPT 問答流程說明

* APP 部分：用戶在手機 APP 中查看表單（表單 ID：A01）。
* 用戶透過 APP 跳轉到 LINE Bot 查詢表單的處理狀態。
* 知識圖譜資料庫：系統將表單 ID（如 A01）與相關資料（如課程設計、康海晴等屬性）聯繫起來，並且通過知識圖譜進行匹配和查詢。
* 在 LINE Bot 問問題：用戶在 LINE Bot 中輸入問題，例如「請假單通過沒有」。
* ChatGPT 回答：LINE Bot 將問題傳送給 ChatGPT。
* ChatGPT 生成回應，然後透過 LINE Bot 回答用戶問題，顯示如「通過」的結果。

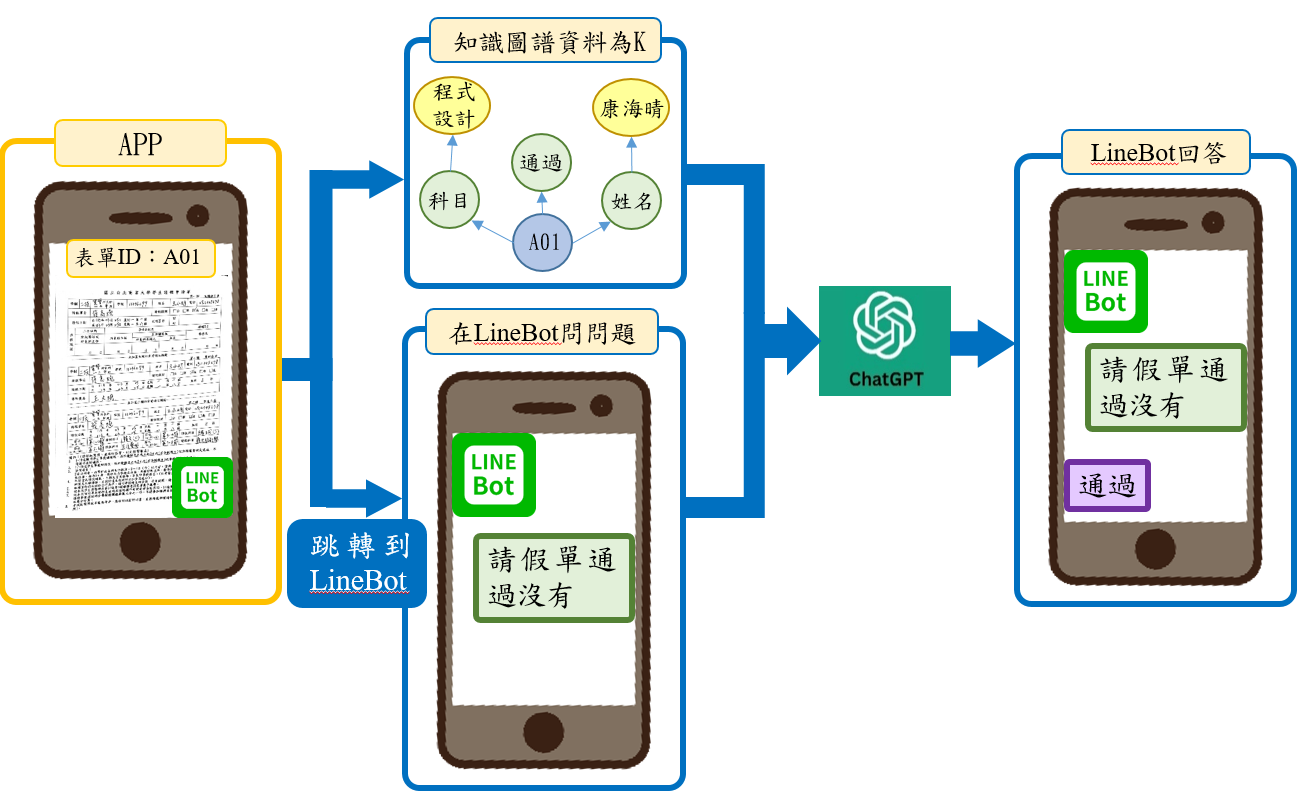


圖 LINE Bot 與 ChatGPT 問答流程圖

LINE BOT多輪問答的流程

當使用者提出問題時，系統將問題傳送給 ChatGPT，並根據資料庫返回答案。隨後，如果用戶繼續提問，系統會將之前的問題和答案一起傳送給 ChatGPT，以提供完整的上下文進行回答。為了避免對話過程變得過於冗長，當使用者提出更多問題時，系統會選擇性地刪除過去不再相關的對話記錄，只保留最近的問題和回覆，避免累積過多無用訊息。

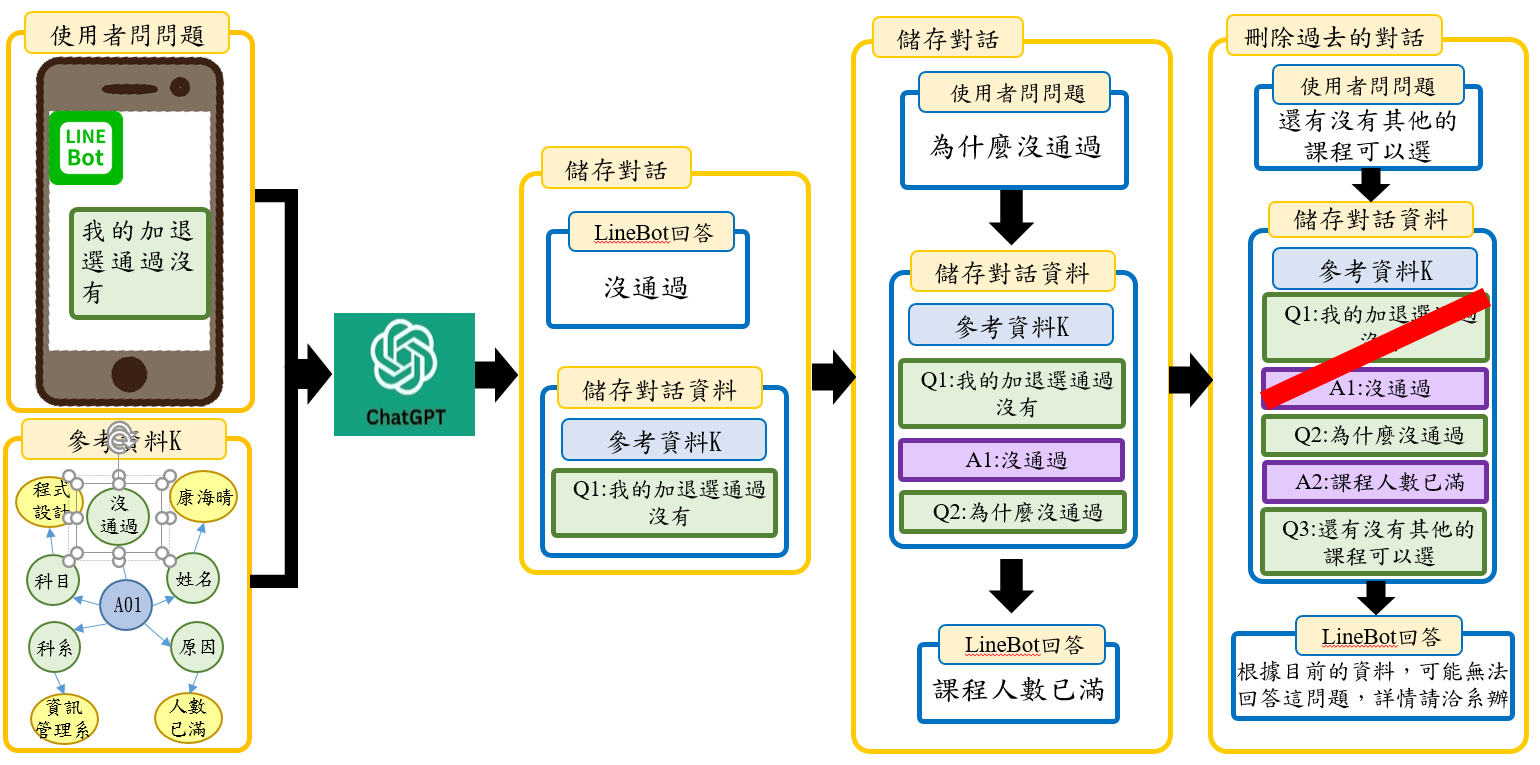


圖 LINE BOT多輪問答的流程圖

APP

## 3-2 系統軟硬體需求與技術平台：

根據傑昇通訊的銷售統計，我們針對銷量排名前兩名的手機品牌進行分析，並基於大眾需求開發應用程序。我們選擇使用 Flutter 作為開發工具，以其跨平台、高性能、豐富的組件庫和活躍的社區支持來提高開發效率和應用質量。針對其中一家

圖3-2-1 2023年下半年手機銷售量

|  |  |
| --- | --- |
| 表3-2-1 系統軟硬體需求-手機 | |
| 系統軟硬體需求-手機 | |
| 手機版本 | Android 12.0 (Marshmallow) |
| 網路需求 | Wi-Fi/3G/4G/5G網路 |

綜合以上分析及市面android系統的版本狀況，我們選擇的手機開發版本為Android 12.0 (Marshmallow)以上版本，需具備Wifi/3G/4G/5G網路。

## 3-3使用標準與工具：

下表為本組的開發工具，各項工具的選用說明如下：

* Python：語法簡潔好閱讀,有許多可供機器學習的套件,寫OCR技術效能較佳,作為OCR技術的開發工具。
* Flutter：是Google開發的跨平台行動應用程式開發工具包，使用 Dart 語言。它提供快速開發和熱重載功能，讓開發者可以輕鬆建立美觀、原生效果的應用程式。
* Mysql Workbench：圖形化的 MySQL 資料庫設計和管理工具，提供了直觀的介面和多種功能，包括資料庫設計、SQL 開發、管理和監控。
* Neo4j：是一種圖形資料庫(graph database,GDB),使用圖結構進行語意查詢的資料庫。他開源，支持 ACID ，並有Web的管理工具、Rest API介面，讓使用者好上手。也具有企業部署的高可用性叢集 是近幾年開始流行的圖形資料庫。
* VS code：包含許多 Python 數據庫、套件的及虛擬環境建置的工具,主要做為機器學習、資料分析等數據科學用途。
* google cloud：是 Google 提供的雲端計算平台，包括 Google Cloud Platform（GCP）和 Google Workspace。它提供了各種雲端基礎設施和服務，包括虛擬機器、資料庫、機器學習等，可滿足不同需求的企業和開發者。
* Flask 是一個由 Python 編寫的微框架，用於構建 Web 應用和 API。它以簡單且可擴展為設計哲學，使開發者能靈活地選擇和配置所需的組件。Flask 的輕量級特性、易於學習和使用、強大的擴展性和模組化設計，使其成為快速開發 Web 應用的理想選擇。
* ChatGPT 是 OpenAI 開發的基於 GPT 架構的語言模型，具備強大的語言生成能力和上下文理解能力，可應用於對話系統、自動寫作、語言翻譯等多個領域。它能生成連貫且有邏輯的文本，適應多種語境，並可根據特定需求進行微調。
* OCR（光學字符識別）技術則是將印刷或手寫的文本圖像轉換為機器可讀文本的技術，被廣泛應用於文檔數字化、自動數據輸入等領域。現代 OCR 系統具備高精度、多語言支持和自動化文本提取功能。這三種技術在現代應用中有著廣泛的用途，並且經常相互結合使用，以構建更強大和智能的系統。
* Github：建立共同合作的軟體開發平台,結合 Fork 工具,可方便管理團隊與成員的資料夾,能清楚追蹤與紀錄每位成員的進度與動態,兼具審視檔案的功能,提供團隊開發更多的便利性,為本組管理專案進度的工具。

表3-3- 1 系統開發環境

|  |  |
| --- | --- |
| 系統開發環境 | |
| 作業平台 | Windows10、Windows11 |
| 程式語言 | Python、Dart |
| 資料庫 | MySQL、Neo4j |
| 開發環境 | Python、Visual Studio Code、google cloud |
| 框架與工具 | Flutter、Flask API |
| 技術 | ChatGPT、OCR技術 |
| 文件製作 | Microsoft Word、Canva |
| 圖表 | Draw.io、Figma |
| 專案管理 | GitHub |
| 檔案存放 | GitHub、Google Drive |

# 第4章 專案時程與組織分工

## 4-1 專案時程

表 4-1-1 專案時程表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 113年 | | | | | | | | | |
| 任務描述 | 1月 | 2月 | 3月 | 4月 | 5月 | 6月 | 7月 | 8月 | 9月 | 10月 |
| 機器學習探討與實作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 訂定主題 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| OCR探討與實作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| App流程設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 頁面框架編寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前端開發 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前後端串接 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 後端開發 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 伺服器架設 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文件製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 簡報製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 海報製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 影片製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 預期進度 | |  | 實際進度 | |  |  |  |

## 4-2 專案組織與分工

表 4-2-1 專案分工表

| 項目/組員 | | 11236011 周松霆 | 11236014 康海晴 | 11236016 黃凱綸 | 11236023 鍾昀臻 |
| --- | --- | --- | --- | --- | --- |
| 資 | 資料庫設計-MySQL |  |  |  |  |
| 料 | 資料庫設計-Neo4j |  |  |  |  |
| 庫 | 資料庫建置-MySQL |  |  |  |  |
|  | 資料庫建置-Neo4j | 〇 |  | ● |  |
| 後 | 伺服器架設 | 〇 |  | ● |  |
|  | 雲端伺服器架設 |  |  |  |  |
| 端 | 雲端OCR辨識 | 〇 |  | ● |  |
| 開 | 本地OCR辨識 | 〇 |  | ● |  |
| 發 | 本地OCR串接 | 〇 |  | ● |  |
|  | Line bot串接 | ● |  |  |  |
|  | ChatGPT串接 | ● |  |  |  |
|  | 分類器(孿生神經網路) |  |  |  |  |
| Line bot | Line bot 建立 | ● |  |  |  |
|  | Ngrok 部署 | ● |  |  |  |
|  | Line bot 串接Ngrok | ● |  |  |  |
|  | API撰寫 |  |  | ● |  |
| APP | 登入/登出 | 〇 |  | ● |  |
|  | 上傳檔案 | ● |  |  |  |
|  | 審查進度 |  |  |  |  |
|  | 掃描文件 |  | 〇 | ● | 〇 |
|  | 修正內容 |  |  |  |  |
|  | 審查撤銷 |  |  |  |  |
|  | Line bot 跳轉 |  |  |  |  |
|  | 搜尋 |  |  |  |  |
|  | 歷史紀錄 |  |  |  |  |
|  | 公告 |  |  |  |  |
|  | 助教、教師審查 |  |  |  |  |
| 前  端  開  發 | Android 開發環境架設 |  |  | 〇 | ● |
|  | ~~登入介面~~ |  |  |  |  |
|  | ~~文件辨識介面~~ |  | 〇 |  | ● |
|  | ~~上傳檔案介面~~ |  | 〇 | 〇 | ● |
|  | ~~審查進度介面~~ |  | 〇 | 〇 | ● |
|  | ~~歷史紀錄介面~~ |  | 〇 | ● | ● |
|  | ~~搜尋介面~~ |  | 〇 | ● | ● |
|  | ~~公告介面~~ |  |  |  |  |
|  | ~~助教、教師審查介面~~ |  |  |  |  |
|  | App黑暗模式 |  |  |  | ● |
|  |  | ● |  |  | 〇 |
|  | API接入 |  |  | ● | 〇 |
| 系  統  流  程  圖  設  計 | App流程設計 | 〇 | ● |  | 〇 |
|  | RAG流程圖 |  |  |  |  |
|  | 分類器製作流程圖 |  |  |  |  |
|  | 自動分類表單流程圖 |  |  |  |  |
|  | 流程圖1 |  |  |  |  |
|  | 流程圖2 |  |  |  |  |
|  | 多個問答流程圖3 |  |  |  |  |
| 美  術  設  計 | UI/ UX |  | 〇 |  | ● |
| APP介面設計 |  | 〇 |  | ● |
| 色彩設計 |  | 〇 |  | ● |
| Logo設計 |  | 〇 |  | ● |
| 素材設計 |  | 〇 |  | ● |
| 文  件  撰  寫 | 統整 |  | ● |  |  |
|  | 第1章 前言 | ● |  |  |  |
|  | 第2章 營運計畫 |  | 〇 |  | ● |
|  | 第3章 系統規格 | 〇 |  |  | ● |
|  | 第4章 專題時程與組織分工 |  | ● | 〇 |  |
|  | 第5章 需求模型 | ● | 〇 | 〇 |  |
|  | 第6章 設計模型 |  | ● |  |  |
|  | 第7章 實作模型 | 〇 | ● |  |  |
|  | 第8章 資料庫設計 | 〇 |  | ● |  |
|  | 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 | 〇 |
|  | 影片製作 |  | ● | 〇 | 〇 |
|  | 海報製作 |  |  |  |  |

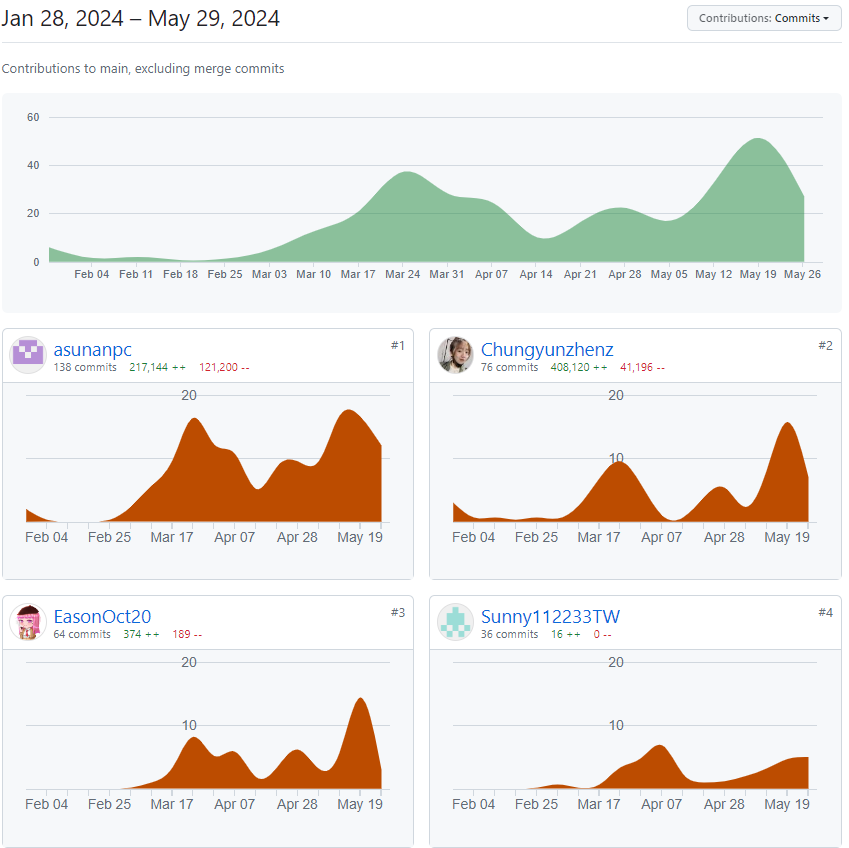
⚫ Github 各組員 commit 的次數: 

圖 4-2-1 Github 頁面

表 4-2-2 分工貢獻度表

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容<各限100字以內> | 貢獻度 |
| 1 | 組長  周松霆 | 一同參與主題發想、製作文件，主要前端負責line Bot跳轉功能  後端負責:Line Bot創建、line Bot與ChatGPT串接，line Bot與資料庫串接。ngrok部屬，ngrok與line Bot串接。 | **25 %** |
| 2 | 組員  康海晴 | 一同參與主題發想、製作文件，主要負責設計APP功能流程、協助製作部分前端介面 | **25 %** |
| 3 | 組員  黃凱綸 | 一同參與主題發想、製作文件，主要負責雲端及本地OCR辨識、後端程式撰寫、API撰寫、建置資料庫和建置伺服器 | **25 %** |
| 4 | 組員  鍾昀臻 | 一同參與主題發想、製作文件，主要負責製作APP前端介面、API接入、美術設計 | **25 %** |
|  | | | 總計:100% |

# 第5章 需求模型

## 5-1 使用者需求

* 登入(UC-D002 )

學生輸入帳號密碼登入。

* 表單上傳與下載(UC-D006)

學生將選定好的表單進行下載或是將以填好資料的表單拍照上傳至系統。

* 上傳表單資料確認(UC-D007)

上傳之後，經過系統處理，學生可以在APP介面上看到資料，並進一步確認資料。

* 修改資料(UC-D008)

當表單內容有誤時，學生可以直接在系統內更改。

* 瀏覽、搜尋個人檔案(UC-D003)

學生可以點選個人檔案按鈕，進入個人檔案區，瀏覽和搜尋曾經上傳過的檔案。

* 瀏覽首頁的公告 (UC-D004)

學生可以瀏覽首頁的公告，某些公告，像是選課公告，裡面進行表單的上傳或是下載。

* 瀏覽常用表單區塊(UC-D005)

學生在常用表單區塊，可以選擇指定的上傳或是下載項目，像是(請假單、選課單、傷病單)。

* Line bot機器人(UC-D009)

點擊Line bot按鈕，跳轉到Line bot介面。

助教方:

* 助教、教師登入(UC-D0010)

助教輸入帳號密碼登入。

* 助教查看申請單(UC-D0011)

助教可以查看學生上傳的申請單，並進一步審核。

* 學生上傳表單檔案搜尋(UC-D0012)

助教、老師輸入關鍵字以搜尋指定的表單。

* 學生上傳表單檔案審核(UC-D0013)

助教、老師審核表單內容，如有問題，透過email和app發送修改通知給學生。

* ~~標註錯誤的地方(UC-D0014)~~

~~助教、老師審核表單內容過程，可以選擇是否標註(框)，學生上傳資料中的錯誤內容。~~

* 選擇輸入退件原因(UC-D0015)

選擇是的話，助教輸入退件原因，並透過email或app發送通知給學生，選擇否的話，則略過。

* ~~助教新增公告(UC-D0016)~~

~~助教可以新增一些公告，注意事項等資訊給學生。~~

* ~~助教修改公告(UC-D0017)~~

~~助教可以更改一些公告。~~

* ~~助教刪除公告(UC-D0018)~~

~~可刪除改一些公告。~~

## 5-2 使用個案圖C:\Users\sunny\Downloads\使用個案圖.jpg

圖 5-2-1 使用個案圖

## 5-3 使用個案描述

* 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

  自動產生的描述功能描述**:使用者利用電子郵件登入系統**

圖 5-3- 1 登入之活動圖

* 一張含有 螢幕擷取畫面, 文字, 數字, 軟體 的圖片

  自動產生的描述功能描述:**瀏覽、搜尋個人檔案**

圖 5-3-2 瀏覽、搜尋個人檔案之活動圖

* 一張含有 螢幕擷取畫面, 文字, 軟體, 設計 的圖片

  自動產生的描述功能描述: **上傳下載(公告)**

圖 5-3-3上傳下載(公告)之活動圖

* **一張含有 螢幕擷取畫面, 文字, 軟體, 數字 的圖片

  自動產生的描述**功能描述: **上傳下載(表單區)**

圖 5-3-4上傳下載(表單區)之活動圖

助教

* 一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

  自動產生的描述功能描述: **助教、老師搜尋檔案**

圖 5-3-5助教搜尋檔案之活動圖

* 一張含有 螢幕擷取畫面, 文字, 軟體, 多媒體軟體 的圖片

  自動產生的描述功能描述**:學生上傳表單檔案審核**

圖 5-3-6學生上傳表單檔案審核之活動圖

圖 5-3-7之活動圖

## 一張含有 螢幕擷取畫面, 文字, 字型, 設計 的圖片 自動產生的描述5-4 分析類別圖

圖5-4-1 分析類別圖

# 第6章 設計模型

## 6-1 循序圖

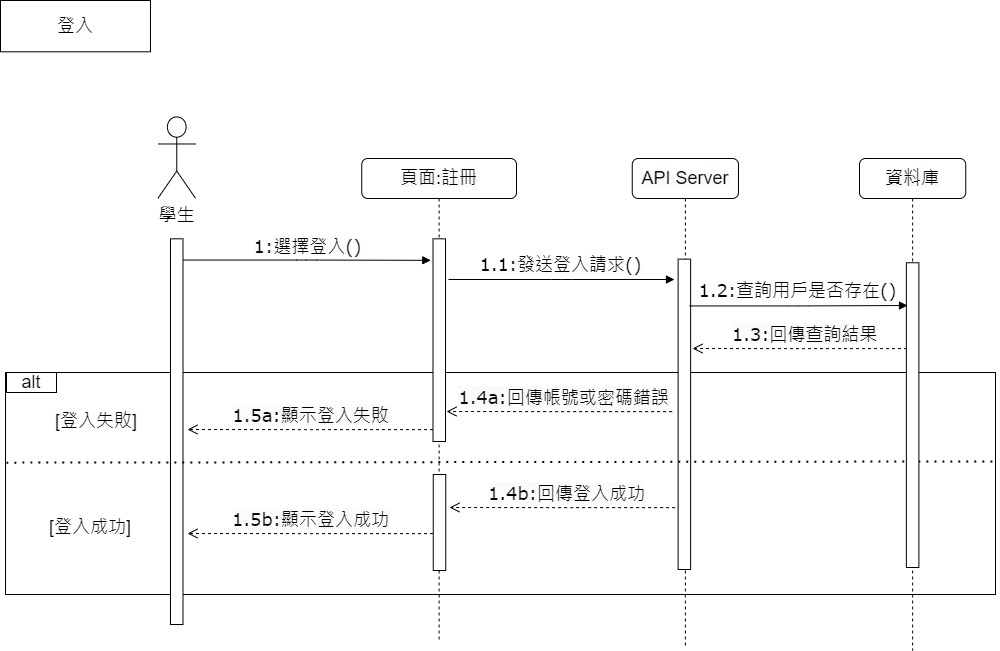


圖 6-1-2 循序圖-登入



圖 6-1-3 循序圖-學生個人檔案查詢



圖 6-1-4 循序圖-學生公告上傳檔案



圖 6-1-5 循序圖-學生公告下載檔案



圖 6-1-6 循序圖-學生個人檔案搜尋



圖 6-1-7 循序圖-學生個人檔案瀏覽

## C:\Users\sunny\Downloads\設計類別圖.jpg6-2 設計類別圖

圖 6-2-1 設計類別圖

# 第7章 實作模型

## 7-1 佈署圖

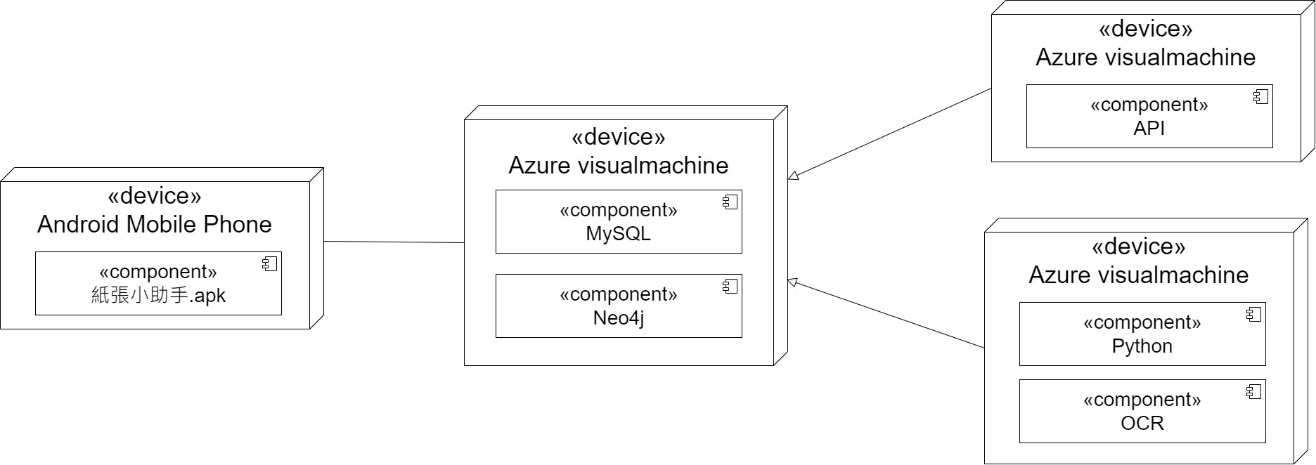


圖 7-1-1 佈署圖

## 7-2 套件圖

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

自動產生的描述

圖 7-2-1 套件圖

## 7-3 元件圖



圖 7-3-1 元件圖

## 7-4 狀態機、 時序圖



圖 7-4-1狀態機-登入



圖 7-4-3 狀態機-上傳檔案

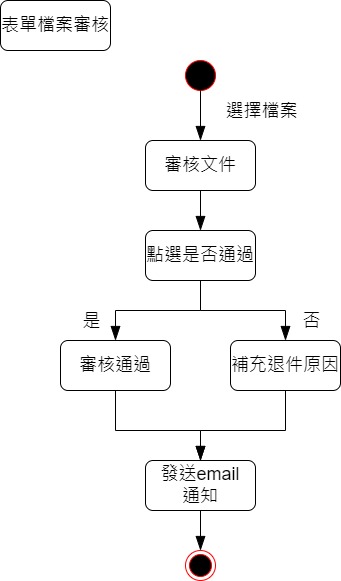


圖 7-4-4 狀態機-表單檔案審核

# 第8章 資料庫設計

## 8-1 資料庫關聯表

圖 8-1-1 資料庫關聯表

## 8-2 表格及其Meta data

表 8-2-1 使用者表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T01 Users使用者 | | | | |
| 欄位名稱 | 欄位中文名稱 | 資料型態 | 資料長度 | 索引 |
| **NationalID** | 流水號 | VARCHAR | 32 | 🗸 |
| **Password** | 密碼 | VARCHAR | 64 |  |
| **Name** | 使用者姓名 | VARCHAR | 16 |  |
| **Phone** | 連絡電話 | VARCHAR | 16 |  |
| **BirthDate** | 生日 | DATE | - |  |
| **StudentID** | 學號 | VARCHAR | 16 |  |
| **Role** | 身分組 | ENUM | - |  |
| **Academic** | 學制 | ENUM | - |  |
| **Department** | 科系別 | ENUM | - |  |
| **line\_id** | N/A | VARCHAR | 512 |  |

表 8-2-2 JSON資料儲存表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T02 json\_data資料儲存 | | | | |
| 欄位名稱 | 欄位中文名稱 | 資料型態 | 資料長度 | 索引 |
| **id** | 流水號 | VARCHAR | 64 | 🗸 |
| **data** | 資料 | JSON | - |  |
| **UploadDate** | 上傳時間 | DATETIME | - |  |
| **UploadedBy** | 上傳者 | VARCHAR | 32 |  |

表 8-2-3 圖片資料儲存表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T03 ImageUploads圖像儲存 | | | | |
| 欄位名稱 | 欄位中文名稱 | 資料型態 | 資料長度 | 索引 |
| **id** | 流水號 | VARCHAR | 64 | 🗸 |
| **Image** | 圖像資料 | LONGBLOB | - |  |
| **UploadDate** | 上傳時間 | DATETIME | - |  |
| **UploadedBy** | 上傳者 | VARCHAR | 32 |  |
| **state** | 狀態 | ENUM |  |  |

表 8-2-4 公告資訊表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T04 announcement公告 | | | | |
| 欄位名稱 | 欄位中文名稱 | 資料型態 | 資料長度 | 索引 |
| **id** | 流水號 | INT | - | 🗸 |
| **Title** | 標題 | VARCHAR | 64 |  |
| **content** | 內文 | LONGTEXT | - |  |
| **UploadedBy** | 上傳者 | VARCHAR | 32 |  |
| **UploadDate** | 上傳時間 | DATETIME |  |  |

表 8-2-5 linebot紀錄表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T05 user\_messages\_linebot紀錄 | | | | |
| 欄位名稱 | 欄位中文名稱 | 資料型態 | 資料長度 | 索引 |
| **id** | 流水號 | INT | - | 🗸 |
| **user\_id** | 用戶id | VARCHAR | 512 |  |
| **message** | 內容 | text | - |  |
| **timestamp** | 對話時間 | DATETIME | - |  |

# 第9章 程式

## 9-1 元件清單及其規格描述。

表 元件清單及其規格描述表(前端)

|  |  |  |  |
| --- | --- | --- | --- |
| Dart檔案(前端) | | | |
| 群組 | 編號 | 檔案名稱 | 功能 |
|  |  | login\_page.dart | 登入畫面 |
| APP介面編寫 |  | s.dart  t.dart  z.dart | 學生主畫面  教師主畫面  助教主畫面 |
|  |  | file\_upload.dart | 上傳檔案畫面 |
|  |  | historical\_record.dart | 查詢歷史記錄畫面 |
|  |  | ReviewCoursePage.dart | 審核假單畫面 |
|  |  | ReviewLeavePage.dart | 審核選課單畫面 |
|  |  | announcement\_page.dart | 公告畫面 |
|  |  | manual\_page | 使用手冊畫面 |
|  |  | stu\_review | 學生審核畫面 |
|  |  | t\_download\_page.dart | 教師下載歷史紀錄畫面 |
|  |  | z\_download\_page.dart | 助教下載歷史紀錄畫面 |

表 元件清單及其規格描述表(後端)

|  |  |  |  |
| --- | --- | --- | --- |
| python檔案(後端) | | | |
| 群組 | 編號 | 檔案名稱 | 功能 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 9-2 其他附屬之各種元件。

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | login\_page.dart |
| 目的 | 三種使用者分別登入，建立登入頁面，讓學生、教師及助教登入，並切換至對應的角色頁面 | | |
| 部分程式碼 | | | |
| 部分程式碼  // 建立登入頁面，包含輸入帳號、密碼及角色登入按鈕  表 9-2-3 部分程式碼 — login\_page.dart  編號 程式名稱  1-1-9 login\_page.dart  目的  建立登入頁面，讓學生、教師及助教登入，並切換至對應的角色頁面  部分程式碼  // 建立登入頁面，提供輸入帳號、密碼及角色登入按鈕  class LoginPage extends StatefulWidget {  const LoginPage({Key? key}) : super(key: key);  @override  LoginPageState createState() => LoginPageState();  }  class LoginPageState extends State<LoginPage> {  final TextEditingController \_studentIdController = TextEditingController();  final TextEditingController \_passwordController = TextEditingController();  String \_errorMessage = '';  // 學生登入  Future<void> \_loginStudent() async {  final user = {'Role': '學生', 'Name': '示例用戶'};  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => StudentPage(user: user)));  }  // 教師登入  Future<void> \_loginTeacher() async {  final user = {'Role': '教師', 'Name': '示例用戶'};  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => TeacherPage(user: user)));  }  // 助教登入  Future<void> \_loginAssistant() async {  final user = {'Role': '助教', 'Name': '示例用戶'};  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => AssistantPage(user: user)));  }  @override  Widget build(BuildContext context) {  return Scaffold(  body: Center(  child: Column(  mainAxisSize: MainAxisSize.min,  children: [  // 輸入帳號  TextField(  controller: \_studentIdController,  decoration: InputDecoration(labelText: '輸入帳號'),  ),  // 輸入密碼  TextField(  controller: \_passwordController,  decoration: InputDecoration(labelText: '輸入密碼'),  obscureText: true,  ),  // 學生登入按鈕  ElevatedButton(onPressed: \_loginStudent, child: Text('學生登入')),  // 教師登入按鈕  ElevatedButton(onPressed: \_loginTeacher, child: Text('教師登入')),  // 助教登入按鈕  ElevatedButton(onPressed: \_loginAssistant, child: Text('助教登入')),  // 錯誤訊息顯示  Text(\_errorMessage, style: TextStyle(color: Colors.red)),  ],  ),  ),  // 浮動按鈕切換主題  floatingActionButton: FloatingActionButton(  onPressed: () => Provider.of<ThemeNotifier>(context, listen: false).toggleTheme(),  child: Icon(Provider.of<ThemeNotifier>(context).isDarkMode ? Icons.dark\_mode : Icons.light\_mode),  ),  );  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | s.dart |
| 目的 | 建立學生頁面，顯示個人資訊並提供功能選單 | | |
| 部分程式碼 | | | |
| 部分程式碼  // 建立學生頁面，顯示學生個人資訊及功能選單  class StudentPage extends StatefulWidget {  final String title;  final Map<String, dynamic> user;  final VoidCallback toggleTheme;  final bool isDarkMode;  const StudentPage({Key? key, required this.title, required this.user, required this.toggleTheme, required this.isDarkMode}) : super(key: key);  @override  \_StudentPageState createState() => \_StudentPageState();  }  class \_StudentPageState extends State<StudentPage> {  late String selectedProfileImage;  @override  void initState() {  super.initState();  selectedProfileImage = widget.user['ProfileImage'] ?? 'assets/a.png';  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text(widget.title),  actions: [  IconButton(  icon: Icon(widget.isDarkMode ? Icons.dark\_mode : Icons.light\_mode),  onPressed: widget.toggleTheme,  ),  ],  ),  drawer: \_buildDrawer(),  body: Center(child: Text('歡迎 ${widget.user['Name']}')),  floatingActionButton: FloatingActionButton(  onPressed: \_launchLineBot,  child: const Icon(Icons.chat),  ),  );  }  // 建立側邊選單  Widget \_buildDrawer() {  return ListView(  children: [  UserAccountsDrawerHeader(  accountName: Text(widget.user['Name']),  accountEmail: Text(widget.user['Role']),  currentAccountPicture: CircleAvatar(backgroundImage: AssetImage(selectedProfileImage)),  ),  ListTile(title: const Text('上傳圖片'), onTap: () => \_navigateTo(FormUploadPage())),  ListTile(title: const Text('審查進度'), onTap: () => \_navigateTo(ReviewListPage())),  ListTile(title: const Text('歷史紀錄'), onTap: () => \_navigateTo(HistoryPage())),  ListTile(title: const Text('公告'), onTap: () => \_navigateTo(AnnouncementPage(role: UserRole.student))),  ListTile(title: const Text('使用手冊'), onTap: () => \_navigateTo(ManualPage())),  ListTile(title: const Text('登出'), onTap: () => \_logout()),  ],  );  }  // 導航到指定頁面  void \_navigateTo(Widget page) {  Navigator.push(context, MaterialPageRoute(builder: (context) => page));  }  // 登出  void \_logout() {  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => const LoginPage()));  }  // 開啟 LINE Bot  Future<void> \_launchLineBot() async {  const url = 'https://line.me/R/ti/p/YOUR\_LINE\_BOT\_ID';  if (await canLaunch(url)) await launch(url);  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | t.dart |
| 目的 | 建立教師頁面，顯示個人資訊並提供功能選單 | | |
| 部分程式碼 | | | |
| 部分程式碼  // 建立教師頁面，顯示教師個人資訊及功能選單  class TeacherPage extends StatefulWidget {  final String title;  final Map<String, dynamic> user;  final VoidCallback toggleTheme;  final bool isDarkMode;  const TeacherPage({Key? key, required this.title, required this.user, required this.toggleTheme, required this.isDarkMode}) : super(key: key);  @override  \_TeacherPageState createState() => \_TeacherPageState();  }  class \_TeacherPageState extends State<TeacherPage> {  late bool isDarkMode;  @override  void initState() {  super.initState();  isDarkMode = widget.isDarkMode;  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text(widget.title),  actions: [  IconButton(  icon: Icon(isDarkMode ? Icons.dark\_mode : Icons.light\_mode),  onPressed: widget.toggleTheme,  ),  ],  ),  drawer: \_buildDrawer(),  body: Center(child: Text('歡迎 ${widget.user['Name']}')),  floatingActionButton: FloatingActionButton(  onPressed: \_launchLineBot,  child: const Icon(Icons.chat),  ),  );  }  // 建立側邊選單  Widget \_buildDrawer() {  return ListView(  children: [  DrawerHeader(  decoration: BoxDecoration(color: Colors.green[400]),  child: const Text('主選單', style: TextStyle(color: Colors.white, fontSize: 24)),  ),  ListTile(title: const Text('審核假單通知'), onTap: () => \_navigateTo(ReviewLeavePage())),  ListTile(title: const Text('所有班級請假單歷史紀錄'), onTap: () => \_navigateTo(FormDownloadPage())),  ListTile(title: const Text('使用手冊'), onTap: () => \_navigateTo(ManualPage())),  ListTile(title: const Text('新增公告'), onTap: () => \_navigateTo(AnnouncementPage(role: UserRole.teacher))),  ListTile(title: const Text('登出'), onTap: \_logout),  ],  );  }  // 導航到指定頁面  void \_navigateTo(Widget page) {  Navigator.push(context, MaterialPageRoute(builder: (context) => page));  }  // 登出  void \_logout() {  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => const LoginPage()));  }  // 開啟 LINE Bot  Future<void> \_launchLineBot() async {  const url = 'https://line.me/R/ti/p/YOUR\_LINE\_BOT\_ID';  if (await canLaunch(url)) await launch(url);  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | z.dart |
| 目的 | 建立助教頁面，顯示個人資訊並提供功能選單 | | |
| 部分程式碼 | | | |
| // 助教頁面，顯示個人資訊及功能選單  class AssistantPage extends StatefulWidget {  final String title;  final VoidCallback toggleTheme;  final bool isDarkMode;  final Map<String, dynamic> user;  const AssistantPage({Key? key, required this.title, required this.toggleTheme, required this.isDarkMode, required this.user}) : super(key: key);  @override  \_AssistantPageState createState() => \_AssistantPageState();  }  class \_AssistantPageState extends State<AssistantPage> {  bool isDarkMode = false;  @override  void initState() {  super.initState();  isDarkMode = widget.isDarkMode;  }  Future<void> \_toggleTheme() async {  setState(() {  isDarkMode = !isDarkMode;  widget.toggleTheme();  });  }  void \_logout() {  Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => const LoginPage()));  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text(widget.title),  actions: [  IconButton(  icon: Icon(isDarkMode ? Icons.dark\_mode : Icons.light\_mode),  onPressed: \_toggleTheme,  ),  ],  ),  drawer: \_buildDrawer(),  body: \_buildBody(),  floatingActionButton: FloatingActionButton(  onPressed: \_launchLineBot,  child: const Icon(Icons.chat),  ),  );  }  Widget \_buildDrawer() {  return ListView(  children: [  DrawerHeader(  decoration: BoxDecoration(color: Colors.orange[400]),  child: const Text('主選單', style: TextStyle(color: Colors.white, fontSize: 24)),  ),  \_buildDrawerItem('審核選課單通知', ReviewCoursePage()),  \_buildDrawerItem('選課單歷史紀錄', FormDownloadPage()),  \_buildDrawerItem('公告', AnnouncementPage(role: UserRole.assistant)),  ListTile(title: const Text('登出'), onTap: \_logout),  ],  );  }  Widget \_buildDrawerItem(String title, Widget page) {  return ListTile(title: Text(title), onTap: () => \_navigateTo(page));  }  Widget \_buildBody() {  return Column(  children: [  \_buildInfoCard(),  \_buildFeatureCard('審核選課單通知', Icons.upload\_file, ReviewCoursePage()),  \_buildFeatureCard('選課單歷史紀錄', Icons.download, FormDownloadPage()),  \_buildFeatureCard('新增公告', Icons.announcement, AnnouncementPage(role: UserRole.assistant)),  ],  );  }  Widget \_buildInfoCard() {  return Container(  padding: const EdgeInsets.all(20.0),  decoration: BoxDecoration(color: Colors.orange.shade200, borderRadius: BorderRadius.circular(15.0)),  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text('姓名: ${widget.user['Name']}', style: const TextStyle(fontSize: 18)),  Text('身份: ${widget.user['Role']}', style: const TextStyle(fontSize: 18)),  ],  ),  );  }  Widget \_buildFeatureCard(String text, IconData icon, Widget page) {  return GestureDetector(  onTap: () => \_navigateTo(page),  child: Container(  padding: const EdgeInsets.all(20.0),  decoration: BoxDecoration(color: Colors.orange.shade200, borderRadius: BorderRadius.circular(15.0)),  child: Row(  children: [  Icon(icon, size: 50),  SizedBox(width: 20),  Text(text, style: const TextStyle(fontSize: 18)),  ],  ),  ),  );  }  void \_navigateTo(Widget page) {  Navigator.push(context, MaterialPageRoute(builder: (context) => page));  }  Future<void> \_launchLineBot() async {  const url = 'https://line.me/R/ti/p/YOUR\_LINE\_BOT\_ID';  if (await canLaunch(url)) await launch(url);  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | file\_upload.dart |
| 目的 | 文件上傳頁面，提供文件選擇和上傳功能 | | |
| 部分程式碼 | | | |
| class FileUploadPage extends StatefulWidget {  @override  \_FileUploadPageState createState() => \_FileUploadPageState();  }  class \_FileUploadPageState extends State<FileUploadPage> {  File? \_file;  Future<void> pickFile() async {  if (await Permission.storage.request().isGranted) {  final result = await FilePicker.platform.pickFiles();  if (result != null) setState(() => \_file = File(result.files.single.path!));  }  }  Future<void> uploadFile() async {  if (\_file == null) {  \_showMessage('錯誤', '未選擇文件');  return;  }  var request = http.MultipartRequest('POST', Uri.parse('http://zct.us.kg:5000/upload'));  request.files.add(await http.MultipartFile.fromPath('file', \_file!.path));  var response = await request.send();  \_showMessage(response.statusCode == 200 ? '成功' : '失敗', response.statusCode == 200 ? '文件已成功上傳' : '文件上傳失敗');  }  void \_showMessage(String title, String message) {  showDialog(  context: context,  builder: (context) => AlertDialog(  title: Text(title),  content: Text(message),  actions: [TextButton(onPressed: () => Navigator.of(context).pop(), child: const Text('好的'))],  ),  );  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: const Text('上傳文件')),  body: Center(  child: Column(  mainAxisSize: MainAxisSize.min,  children: [  ElevatedButton.icon(onPressed: pickFile, icon: const Icon(Icons.folder\_open), label: const Text('選擇文件')),  ElevatedButton.icon(onPressed: uploadFile, icon: const Icon(Icons.cloud\_upload), label: const Text('上傳文件')),  Text(\_file != null ? '選擇的文件: ${\_file!.path.split('/').last}' : '未選擇文件'),  ],  ),  ),  );  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | historical\_record.dart |
| 目的 | 歷史紀錄頁面，提供查詢和顯示歷史資料的功能 | | |
| 部分程式碼 | | | |
| // 歷史紀錄頁面，提供查詢和顯示歷史資料的功能  class HistoryPage extends StatefulWidget {  @override  \_HistoryPageState createState() => \_HistoryPageState();  }  class \_HistoryPageState extends State<HistoryPage> {  List<dynamic> historyData = [];  final TextEditingController keywordController = TextEditingController();  String searchType = 'academic\_year';  bool isLoading = true;  String errorMessage = '';  @override  void initState() {  super.initState();  fetchHistoryData();  }  Future<void> fetchHistoryData() async {  setState(() {  isLoading = true;  errorMessage = '';  });  final response = await http.get(Uri.parse('http://zct.us.kg:5000/history'));  if (response.statusCode == 200) {  setState(() {  historyData = json.decode(response.body);  isLoading = false;  });  } else {  \_showError('Failed to load history data');  }  }  void \_showError(String message) {  setState(() {  errorMessage = message;  isLoading = false;  });  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(message)));  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: const Text('查詢歷史紀錄')),  body: Padding(  padding: const EdgeInsets.all(16.0),  child: Column(  children: [  Row(  children: [  Expanded(child: \_buildDropdown()),  const SizedBox(width: 16),  Expanded(child: \_buildSearchField()),  ],  ),  const SizedBox(height: 16),  isLoading ? const CircularProgressIndicator() : \_buildHistoryList(),  ],  ),  ),  );  }  Widget \_buildDropdown() {  return DropdownButtonFormField<String>(  value: searchType,  onChanged: (newValue) => setState(() => searchType = newValue!),  items: <String>['academic\_year', 'period', 'date', 'course\_name'].map((type) {  return DropdownMenuItem(value: type, child: Text(type));  }).toList(),  );  }  Widget \_buildSearchField() {  return TextField(  controller: keywordController,  decoration: InputDecoration(  labelText: '輸入查詢關鍵字',  border: OutlineInputBorder(),  suffixIcon: IconButton(  icon: const Icon(Icons.search),  onPressed: () {  // searchHistoryData implementation omitted  },  ),  ),  );  }  Widget \_buildHistoryList() {  if (historyData.isEmpty) return const Text('無歷史紀錄');  return Expanded(  child: ListView.builder(  itemCount: historyData.length,  itemBuilder: (context, index) {  final item = historyData[index];  return ListTile(  title: Text(item['title'] ?? '無標題'),  subtitle: Text('學年: ${item['academic\_year'] ?? '無'}'),  );  },  ),  );  }  }  void main() {  runApp(MaterialApp(home: HistoryPage()));  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | ReviewCoursePage.dart |
| 目的 | 選課單審核頁面，顯示審核狀態和提供更新功能 | | |
| 部分程式碼 | | | |
| class ReviewCoursePage extends StatefulWidget {  const ReviewCoursePage({Key? key}) : super(key: key);  @override  \_ReviewCoursePageState createState() => \_ReviewCoursePageState();  }  class \_ReviewCoursePageState extends State<ReviewCoursePage> {  List<Map<String, dynamic>> \_pendingRequests = [];  List<Map<String, dynamic>> \_returnedRequests = [];  List<Map<String, dynamic>> \_completedRequests = [];  bool \_isLoading = true;  @override  void initState() {  super.initState();  \_fetchCourseRequests();  }  Future<void> \_fetchCourseRequests() async {  setState(() => \_isLoading = true);  try {  final responses = await Future.wait([  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=審查中&title=選課單')),  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=退回&title=選課單')),  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=通過&title=選課單'))  ]);  if (responses.every((response) => response.statusCode == 200)) {  setState(() {  \_pendingRequests = \_parseResponse(responses[0].body);  \_returnedRequests = \_parseResponse(responses[1].body);  \_completedRequests = \_parseResponse(responses[2].body);  \_isLoading = false;  });  } else {  \_showError('加載選課單失敗');  }  } catch (e) {  \_showError('錯誤: $e');  }  }  List<Map<String, dynamic>> \_parseResponse(String body) {  return List<Map<String, dynamic>>.from(json.decode(body));  }  void \_showError(String message) {  setState(() => \_isLoading = false);  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(message)));  }  @override  Widget build(BuildContext context) {  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  title: const Text('查看選課單進度'),  bottom: const TabBar(tabs: [Tab(text: '審查中'), Tab(text: '退回'), Tab(text: '已完成')]),  ),  body: \_isLoading ? const Center(child: CircularProgressIndicator()) : \_buildTabViews(),  ),  );  }  Widget \_buildTabViews() {  return TabBarView(  children: [  \_buildTabContent(\_pendingRequests),  \_buildTabContent(\_returnedRequests),  \_buildTabContent(\_completedRequests),  ],  );  }  Widget \_buildTabContent(List<Map<String, dynamic>> requests) {  if (requests.isEmpty) return const Center(child: Text('沒有選課單'));  return ListView.builder(  itemCount: requests.length,  itemBuilder: (context, index) {  final request = requests[index];  return ListTile(  title: Text(request['title'] ?? '無標題'),  subtitle: Text(request['description'] ?? '無描述'),  onTap: () => \_showDetails(request),  );  },  );  }  void \_showDetails(Map<String, dynamic> details) {  showDialog(  context: context,  builder: (context) {  return AlertDialog(  title: Text('${details['submission\_date']} 選課單'),  content: Column(  crossAxisAlignment: CrossAxisAlignment.start,  mainAxisSize: MainAxisSize.min,  children: [  Text('課程名稱: ${details['course\_name']}'),  Text('選課時間: ${details['period']}'),  Text('學期: ${details['academic\_year']}'),  ],  ),  );  },  );  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | ReviewLeavePage.dart |
| 目的 | 審核請假單頁面，顯示審核狀態和提供更新功能 | | |
| 部分程式碼 | | | |
| class ReviewLeavePage extends StatefulWidget {  const ReviewLeavePage({Key? key}) : super(key: key);  @override  \_ReviewLeavePageState createState() => \_ReviewLeavePageState();  }  class \_ReviewLeavePageState extends State<ReviewLeavePage> {  List<Map<String, dynamic>> \_pendingRequests = [];  List<Map<String, dynamic>> \_returnedRequests = [];  List<Map<String, dynamic>> \_completedRequests = [];  bool \_isLoading = true;  @override  void initState() {  super.initState();  \_fetchLeaveRequests();  }  Future<void> \_fetchLeaveRequests() async {  setState(() => \_isLoading = true);  try {  final responses = await Future.wait([  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=審查中&title=請假單')),  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=退回&title=請假單')),  http.get(Uri.parse('http://zct.us.kg:5000/getLeaveRequests?status=完成&title=請假單'))  ]);  if (responses.every((response) => response.statusCode == 200)) {  setState(() {  \_pendingRequests = \_parseResponse(responses[0].body);  \_returnedRequests = \_parseResponse(responses[1].body);  \_completedRequests = \_parseResponse(responses[2].body);  \_isLoading = false;  });  } else {  \_showError('加載請假單失敗');  }  } catch (e) {  \_showError('錯誤: $e');  }  }  List<Map<String, dynamic>> \_parseResponse(String body) {  return List<Map<String, dynamic>>.from(json.decode(body));  }  void \_showError(String message) {  setState(() => \_isLoading = false);  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(message)));  }  @override  Widget build(BuildContext context) {  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  title: const Text('查看請假單進度'),  bottom: const TabBar(tabs: [Tab(text: '審查中'), Tab(text: '退回'), Tab(text: '已完成')]),  ),  body: \_isLoading ? const Center(child: CircularProgressIndicator()) : \_buildTabViews(),  ),  );  }  Widget \_buildTabViews() {  return TabBarView(  children: [  \_buildTabContent(\_pendingRequests),  \_buildTabContent(\_returnedRequests),  \_buildTabContent(\_completedRequests),  ],  );  }  Widget \_buildTabContent(List<Map<String, dynamic>> requests) {  if (requests.isEmpty) return const Center(child: Text('没有請假單'));  return ListView.builder(  itemCount: requests.length,  itemBuilder: (context, index) {  final request = requests[index];  return ListTile(  title: Text(request['title'] ?? '無標題'),  subtitle: Text(request['description'] ?? '無描述'),  onTap: () => \_showDetails(request),  );  },  );  }  void \_showDetails(Map<String, dynamic> details) {  showDialog(  context: context,  builder: (context) {  return AlertDialog(  title: Text('${details['submission\_date']} 請假單'),  content: Column(  crossAxisAlignment: CrossAxisAlignment.start,  mainAxisSize: MainAxisSize.min,  children: [  Text('請假原因: ${details['leave\_reason']}'),  Text('請假時間: ${details['period']}'),  Text('學期: ${details['academic\_year']}'),  ],  ),  );  },  );  }  } | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | announcement\_page.dart |
| 目的 | 提供公告的查看與新增功能 | | |
| 部分程式碼 | | | |
| // AnnouncementPage 類別：公告頁面  class AnnouncementPage:  屬性 role: 用戶角色  方法 initState():  調用 \_fetchAnnouncements()  方法 \_fetchAnnouncements():  獲取公告資料 (模擬資料)  \_sortAnnouncements()  更新狀態  方法 \_sortAnnouncements():  排序公告，根據時間升序或降序  方法 \_showAddAnnouncementDialog():  顯示新增公告對話框  方法 build(context):  返回 Scaffold:  - AppBar: "公告"  - 排序按鈕  - 如果角色不是學生，顯示新增公告按鈕  - ListView:  遍歷 announcements:  每個項目顯示公告標題，點擊可查看詳細內容 (省略實現) | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | manual\_page.dart |
| 目的 | 使用指南頁面 | | |
| 部分程式碼 | | | |
| class ManualPage:  方法 build(context):  返回 Scaffold:  - appBar: "使用指南"  - body:  - Stack:  - 背景漸變色  - Padding:  - Column:  - Row1:  - 功能卡片("功能介紹", Icons.settings, 功能介紹頁面)  - 功能卡片("幫助中心", Icons.help\_outline, 幫助中心頁面)  - Row2:  - 功能卡片("回饋與設定", Icons.feedback, 回饋表單)  // 建立背景漸變效果  方法 \_buildBackgroundGradient():  返回 Container (漸變色設置)  // 建立區塊卡片  方法 \_buildSectionCard(context, title, icon, page):  返回 Card:  - InkWell:  - 點擊打開新頁面  - Padding:  - Column:  - Icon: icon  - Text: title  // 建立功能介紹頁面  方法 \_buildFunctionIntroductionPage():  返回 Scaffold:  - appBar: "功能介紹"  - body: ListView:  - ExpansionTile: "功能介紹一 上傳功能"  - ExpansionTile: "功能介紹一 智能小幫手"  - ExpansionTile: "功能介紹二 上傳紀錄"  // 建立幫助中心頁面  方法 \_buildHelpCenterPage():  返回 Scaffold:  - appBar: "幫助中心"  - body: ListView:  - ListTile: "常見問題"  - ListTile: "更新公告"  // 展開項目  方法 \_buildExpansionTile(title, content):  返回 Card:  - ExpansionTile:  - 標題: title  - 內容: content  // 回饋表單類別  class FeedbackForm:  方法 build(context):  返回 Scaffold:  - appBar: "回饋表單"  - body: Padding:  - Column:  - TextField: "您的回饋"  - ElevatedButton: 點擊後顯示回饋訊息 | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | stu\_review.dart |
| 目的 | 學生查審核列表頁面 | | |
| 部分程式碼 | | | |
| class ReviewListPage:  方法 initState():  調用 fetchReviews()  方法 fetchReviews():  獲取審查數據，分為"審查中"、"退回"、"完成"三類  方法 build(context):  返回 Scaffold:  - AppBar: "查看所有審查進度"  - TabBar: "審查中"、"退回"、"完成"  - TabBarView:  - buildReviewList(審查中)  - buildReviewList(退回)  - buildReviewList(完成)  方法 buildReviewList(reviews, color):  如果 reviews 為空:  返回 "沒有項目"  否則:  返回 ListView:  每個審查項目:  Card:  - 標題: reviews[index]['title']  - 審核狀態: reviews[index]['review\_status']  - 點擊打開 ReviewDetailPage  方法 \_handleWithdraw(review):  發送撤回請求  更新審查列表  // ReviewDetailPage 類別：審查詳情頁面  class ReviewDetailPage:  方法 build(context):  返回 Scaffold:  - AppBar: 審查標題  - 內容: 請假內容、提交時間、審核時間、審核狀態  - 按鈕: "關閉"、"撤回審核"  方法 main():  啟動應用，顯示 ReviewListPage | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | t\_download.dart |
| 目的 | 教師端-各班學生請假單歷史下載頁面 | | |
| 部分程式碼 | | | |
| class FormDownloadPage:  屬性 \_selectedClass, \_searchKeyword, \_searchType, userRole  屬性 \_classData, \_studentData  方法 initState():  初始化，調用 \_fetchClassData()  方法 \_fetchClassData():  獲取班級資料 (模擬資料)  方法 \_fetchStudentData(className):  根據班級名稱獲取學生資料  方法 \_filteredStudents:  過濾已完成的學生表單，並根據關鍵字篩選  方法 requestPermissions():  請求存儲權限  方法 downloadFile():  確認權限後下載檔案  方法 \_showDetailsDialog(context, description):  顯示請假單詳情，並提供下載歷史資料選項  方法 build(context):  返回 Scaffold:  - AppBar: 顯示班級名稱或返回按鈕  - Body:  - 搜索和過濾欄位  - 顯示班級列表或學生列表  - 點擊班級時調用 \_fetchStudentData  - 點擊學生顯示詳情對話框 | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 | z\_download.dart |
| 目的 | 助教端-各班學生選課單歷史下載頁面 | | |
| 部分程式碼 | | | |
| class AssistantPage:  屬性 title, toggleTheme, isDarkMode, user  方法 initState():  初始化，調用 \_loadThemePreference()  方法 \_loadThemePreference():  獲取並設定主題偏好  方法 \_toggleTheme():  切換深色/淺色模式  方法 \_logout(context):  導航至 LoginPage 進行登出  方法 \_launchLineBot():  打開 LINE Bot 聯結  方法 build(context):  返回 Scaffold:  - AppBar: 包含標題和主題切換按鈕  - Drawer: 提供導航選項 (選課單審核、歷史紀錄、公告、登出)  - Body: 顯示助教個人資訊和功能卡片  - FloatingActionButton: 打開 LINE Bot  方法 \_buildFeatureCard(context, icon, text, page):  建立功能卡片，點擊時導航到相應頁面 | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

表 部分程式碼―

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 |  | 檔案名稱 |  |
| 目的 |  | | |
| 部分程式碼 | | | |
|  | | | |

# 第10章 測試模型

## 10-1 測試計畫

本系統的測試方向共分為學生、助教、教師使用紙張小精靈APP。在系統程式開發設計完成之後，測試系統功能是否能順利正常運行，確保每項功能皆能正確執行。

**A. 帳號**

1. 登入：學生、助教、教師是否可以成功登入到首頁

**B. 學生**

* + 1. 文件掃描：是否可以正確辨識出表單文字
    2. 文件上傳：是否可以成功上傳
    3. 搜尋檔案：是否可以成功搜尋到檔案
    4. 歷史紀錄：是否可以成功檢視曾經上傳的檔案
    5. 審核進度：是否可以成功檢視表單審核進度
    6. Line Bot問問題：Linebot是否可以成功回答問題
    7. 查看公告：是否可以成功檢視公告
    8. 學生個人資料：是否可以成功檢視學生自己的資料
    9. 使用手冊：是否可以成功檢視使用手冊

**C. 助教**

1. 審核表單通過：是否可以成功選擇審核通過
2. 表單退回原因：是否可以成功送出退回原因
3. 搜尋檔案：是否可以成功搜尋到檔案
4. 查看班級檔案：是否可以成功檢視各班級檔案
5. 助教個人資料瀏覽：是否可以成功檢視助教自己的資料
6. 使用手冊瀏覽：是否可以成功檢視使用手冊

**D. 教師**

1. 審核表單通過：是否可以成功選擇審核通過
2. 表單退回原因：是否可以成功送出退回原因
3. 搜尋檔案：是否可以成功搜尋到檔案
4. 查看班級檔案：是否可以成功檢視自己班級檔案
5. 教師個人資料瀏覽：是否可以成功檢視教師自己的資料
6. 使用手冊瀏覽：是否可以成功檢視使用手冊

## 10-2 測試個案與測試結果資料

**A. 帳號**

表 登入測試

|  |  |
| --- | --- |
| 功能名稱 | A1登入 |
| 測試目標 | 學生、助教、教師是否可以成功登入到首頁 |
| 測試流程 | 1. 輸入信箱與密碼 2. 登入 |
| 測試結果 |  |

**B. 學生**

表 文件掃描測試

|  |  |
| --- | --- |
| 功能名稱 | B1文件掃描 |
| 測試目標 | 是否可以正確辨識出表單文字 |
| 測試流程 | 1. 拍照選課單/請假單 2. 查看辨識出來的內容 3. 如果正確點擊確認 |
| 測試結果 |  |

表 文件上傳測試

|  |  |
| --- | --- |
| 功能名稱 | B2文件上傳 |
| 測試目標 | 是否可以成功上傳 |
| 測試流程 | 1. 選擇上傳表單 2. 選擇一張表單照片上傳 3. 點擊確認上傳 |
| 測試結果 | 例如上傳單據的清單 把清單的結果傳下來說 對應的放在這邊 代表說是正常的 進階(要時間)找範例資料請假單 丟進去是不是真的能分清楚這是請假單(測試10張辨識 可以告訴評審機率) |

表 搜尋檔案測試

|  |  |
| --- | --- |
| 功能名稱 | B3搜尋檔案 |
| 測試目標 | 是否可以成功搜尋到檔案 |
| 測試流程 | 1. 選擇選課單/請假單 2. 輸入關鍵字 3. 點擊搜尋 |
| 測試結果 |  |

表 歷史紀錄測試

|  |  |
| --- | --- |
| 功能名稱 | B4歷史紀錄 |
| 測試目標 | 是否可以成功檢視曾經上傳的檔案 |
| 測試流程 | 1. 進入歷史紀錄 2. 查看曾經上傳的表單照片及文件 |
| 測試結果 |  |

表 審核結果測試

|  |  |
| --- | --- |
| 功能名稱 | B5審核結果 |
| 測試目標 | 是否可以成功檢視表單審核結果 |
| 測試流程 | 1. 到審核結果 2. 點擊想看的表單 3. 查看審核結果 |
| 測試結果 |  |

表 Line Bot問問題測試

|  |  |
| --- | --- |
| 功能名稱 | B6 Line Bot問問題 |
| 測試目標 | Linebot是否可以成功回答使用者問題 |
| 測試流程 | 1. 點擊Line Bot圖示 2. 成功跳轉到Line 3. 輸入問題與Line Bot問問題 |
| 測試結果 |  |

表 查看公告測試

|  |  |
| --- | --- |
| 功能名稱 | B7查看公告 |
| 測試目標 | 是否可以成功檢視公告 |
| 測試流程 | 1. 進入公告 2. 查看公告 |
| 測試結果 |  |

表 學生個人資料測試

|  |  |
| --- | --- |
| 功能名稱 | B8學生個人資料 |
| 測試目標 | 是否可以成功檢視學生自己的資料 |
| 測試流程 | 1. 點擊個人資料 2. 查看個人資料 |
| 測試結果 |  |

表 使用手冊測試

|  |  |
| --- | --- |
| 功能名稱 | B9使用手冊 |
| 測試目標 | 是否可以成功檢視使用手冊 |
| 測試流程 | 1. 進入使用手冊 2. 查看使用手冊 |
| 測試結果 |  |

**C. 助教**

表 審核表單通過測試

|  |  |
| --- | --- |
| 功能名稱 | C1審核表單通過 |
| 測試目標 | 是否可以成功選擇審核通過 |
| 測試流程 | 1. 選擇表單通過 2. 到已完成區域確認該表單 |
| 測試結果 |  |

表 表單退回測試

|  |  |
| --- | --- |
| 功能名稱 | C2表單退回原因 |
| 測試目標 | 是否可以成功退回表單並送出退回原因 |
| 測試流程 | 1. 選擇表單退回 2. 輸入退回原因 3. 點擊送出 4. 到已退回區域確認該表單 |
| 測試結果 |  |

表 搜尋檔案測試

|  |  |
| --- | --- |
| 功能名稱 | C3搜尋檔案 |
| 測試目標 | 是否可以成功搜尋到檔案 |
| 測試流程 | 1. 輸入關鍵字 2. 點擊搜尋 |
| 測試結果 |  |

表 查看班級檔案測試

|  |  |
| --- | --- |
| 功能名稱 | C4查看班級檔案 |
| 測試目標 | 是否可以成功檢視各班級檔案 |
| 測試流程 | 1. 進入班級檔案 2. 進入其中一班 3. 查看該班級學生檔案 |
| 測試結果 |  |

表 助教個人資料測試

|  |  |
| --- | --- |
| 功能名稱 | C5助教個人資料 |
| 測試目標 | 是否可以成功檢視助教自己的資料 |
| 測試流程 | 1. 點擊個人資料 2. 查看個人資料 |
| 測試結果 |  |

表 使用手冊測試

|  |  |
| --- | --- |
| 功能名稱 | C6使用手冊 |
| 測試目標 | 是否可以成功檢視使用手冊 |
| 測試流程 | 1. 進入使用手冊 2. 查看使用手冊 |
| 測試結果 |  |

**D. 教師**

表 審核表單通過測試

|  |  |
| --- | --- |
| 功能名稱 | D1審核表單通過 |
| 測試目標 | 是否可以成功選擇審核通過 |
| 測試流程 | 1. 選擇表單通過 2. 到已完成區域確認該表單 |
| 測試結果 |  |

表 表單退回原因測試

|  |  |
| --- | --- |
| 功能名稱 | D2表單退回原因 |
| 測試目標 | 是否可以成功送出退回原因 |
| 測試流程 | 1. 選擇表單退回 2. 輸入退回原因 3. 點擊送出 4. 到已退回區域確認該表單 |
| 測試結果 |  |

表 搜尋檔案測試

|  |  |
| --- | --- |
| 功能名稱 | D3搜尋檔案 |
| 測試目標 | 是否可以成功搜尋到檔案 |
| 測試流程 | 1. 輸入關鍵字 2. 點擊搜尋 |
| 測試結果 |  |

表 查看班級檔案測試

|  |  |
| --- | --- |
| 功能名稱 | D4查看班級檔案 |
| 測試目標 | 是否可以成功檢視自己班級檔案 |
| 測試流程 | 1. 進入班級檔案 2. 查看該班級學生檔案 |
| 測試結果 |  |

表 教師個人資料測試

|  |  |
| --- | --- |
| 功能名稱 | D5教師個人資料 |
| 測試目標 | 是否可以成功檢視教師自己的資料 |
| 測試流程 | 1. 點擊個人資料 2. 查看個人資料 |
| 測試結果 |  |

表 使用手冊測試

|  |  |
| --- | --- |
| 功能名稱 | D6使用手冊 |
| 測試目標 | 是否可以成功檢視使用手冊 |
| 測試流程 | 1. 進入使用手冊 2. 查看使用手冊 |
| 測試結果 |  |

# 第11章 操作手冊

## 11-1 系統元件

表 系統安裝元件資訊

|  |  |
| --- | --- |
| 元件名稱 | 紙張小精靈paper genie |
| 檔案大小 |  |
| 費用 | 免費 |
| 版本需求 |  |
| 網路需求 | Wi-Fi/3G/4G/5G網路 |

## 11-2 系統下載及安裝

|  |  |
| --- | --- |
| 掃描QR code即可下載APP | |
| (網路空間 APK放進去)  圖QR code | |
| 步驟一 | |
|  | 開啟APP輸入帳號密碼後，點擊登入按鈕，登入至首頁 |
| 步驟二 | |
|  | 到首頁後即可開始使用APP |

# 第12章 使用手冊

## 12-1 使用者操作

(1)登入流程

|  |  |
| --- | --- |
| 圖 | 圖 |
| 1. 開啟APP輸入帳號密碼登入 | 1. 登入成功至首頁 |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# 第13章 感想

* 11236011周松霆
* 11236014康海晴
* 11236016黃凱綸
* 11236023鍾昀臻

在進入二技階段後，參與專題製作是我學術生涯中一次極具挑戰且成長的經驗。這次專題製作讓我深刻體會到，雖然一個人可能能夠快速完成任務，但真正穩定且長遠的成就，來自於團隊的力量與合作。不僅學會了技術性地完成一個專案，更重要的是，我學會了如何傾聽與吸收來自他人的意見。無論是來自教授、同學，還是團隊成員的反饋，這些建議都成為我不斷改進和提升自我的動力。透過這樣的反饋過程，我也學會了在團隊中如何保持開放的態度，接納不同的聲音，進而幫助專案朝更完善的方向前進。

此外，在專題過程中，為了讓研究結果更具體呈現，我進一步加強了整理重點與撰寫內文的能力，將複雜的技術內容轉化為簡潔明瞭的文字表達。

在專題製作的 12 個月內，雖然學習 Flutter 的時間有限，無法深入掌握所有細節，但我對該框架有了清楚的理解，並能夠將其應用到專案開發中，使應用程式的開發更具彈性，這不僅讓我能夠在聽取教授和組員討論時充分理解所談及的概念，還能夠就相關技術發展提出自己的看法與建議，為團隊貢獻一己之力。

總結來說，二技的專題製作不僅讓我技術上有所突破，更讓我在人際互動、資料搜集與整理、研究規劃和寫作技巧上獲得了寶貴的經驗。加上自學 Flutter 和深度學習理論的累積，這些能力將成為我未來職業生涯中的重要基石，讓我能夠在面對挑戰時更加從容應對。

# 第14章 參考資料

1. Flutter App開發教學

<https://medium.com/@w96284ps/flutter-app%E9%96%8B%E7%99%BC%E6%95%99%E5%AD%B8-%E4%B8%80-64661160906f>

2. LineBot儲存使用者傳送的圖片或影片

<https://steam.oxxostudio.tw/category/python/example/line-save-image.html#google_vignette>

3. Flutter介紹：把資料存進手機 - app database

<https://ithelp.ithome.com.tw/m/articles/10303711>

4. 幫 Line Bot 加上身份驗證

<https://ithelp.ithome.com.tw/m/articles/10262271>

5. Neo4j Course for Beginners

<https://www.youtube.com/watch?v=_IgbB24scLI>

6. 深度學習 最佳入門邁向AI專題實戰(書本)

<https://www.books.com.tw/products/0010898997>

7. Python機器學習超進化：AI影像辨識跨界應用實戰(書本)

<https://www.books.com.tw/products/0010870722>

# 附錄

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 無紙化的點在哪裡? 學生仍然需要填寫紙本再進行掃描，還是後續教職員整理中有節省紙本成本，可以再多補充這部分 | 目前學校的運作流程大部分仍以紙本為主，紙本仍是行政程序中重要的記錄方式。不僅在學校如此，銀行等機構也同樣面臨類似情況，即便擁有數位系統，紙本記錄依然不可或缺。  無紙化的效益主要體現在後續處理流程上。本系統通過將紙本文件數位化，縮短後端處理時間，並自動進行分類和內容處理，讓後續操作不再依賴實體紙本。例如，當使用者需要查詢文件進度或結果時，系統可直接從數位檔案中提取相關資訊，無需重新查找紙本資料，減少了紙張的使用。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 導入Line Bot的效益是什麼? 目前提到的功能都在自身系統中實現(ChatGPT會知道北商選課的流程?) | 導入Line Bot主要效益在於降低使用者的學習成本。使用者無需花時間學習如何操作APP中的各項功能，只需透過Line進行簡單的對話，即可查詢相關資訊，如請假單審核進度或原因。  Line Bot能自動從系統中獲取資料，提供即時回覆，讓使用者有更直接且方便的查詢管道。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 商業模式?(開發完之後是打算賣給學校嗎?還是免費使用) | 系統的應用不僅限於學校行政管理，許多企業與金融機構（如銀行）也需處理紙本文件，並耗費大量人力。  在商業模式上，針對學校的需求外，此技術也能推廣至企業與政府單位，幫助這些機構數位化後端流程，實現自動化並降低人力成本，提升運作效率。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 報告順序:核心功能開始 | 複審時，功能介紹會從本組主要研究項目—自動化分辨表單、AI修正辨識結果這兩項優先報告 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 功能性較薄弱，可考慮與學校現有系統做整合的可能性，以及單獨跨系選修電子化(請假單、選課單) | 我們系統的核心在於**利用 AI 技術自動化辨識表單種類以及修正辨識結果**。透過 OCR 掃描表單中的文字，系統會利用 AI 技術自動辨識表單的種類，並修正識別結果，提升紙本文件處理的自動化與精確度，著重於後端的自動化處理。  目前學校的請假系統由外包廠商維護，整合難度較高，未來如果學校的系統開放修改或整合，可以進一步的整合。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| OCR若遇手抖，辨識率高低? | 考慮到拍攝品質對辨識率的影響。若照片因手抖導致模糊，OCR的辨識率自然會降低，甚至無法正確抓取文字框。  那我們在系統的偵測階段便加入檢測的機制，例如：發現拍攝的圖片模糊，系統將提醒使用者重新拍攝，確保OCR能準確識別文字並提高整體辨識效率。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| OCR辨識後，拿表格名稱是否可取代分類器 | 由於文字在表單中的位置可能因拍攝角度或表單樣式的不同而有所變化，使得單靠表格名稱進行分類的準確性變得不可靠。例如，同一張表單送入OCR辨識，文字內容可能會因拍攝時的不同而出現在不同的格子中。  所以我們提取表格的形狀和框架，並透過AI技術進行分類的方式，能更準確地識別表格的結構，並有效處理文字內容的潛在錯誤，提高分類的準確率與穩定性。 |

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| RAG目前的設計細節 | 詳細細節請參考文件第15頁的第3章，其中提供了RAG流程圖及其說明 |