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10月16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，當模型辨識字形有誤時，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：設計一個自動化流程，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。
* 資料處理錯誤率降低：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，符合持續發展的目標。
* 降低助教處理文件的工作量：流程自動化使助教節省文件處理時間。
* 改善學生文件繳送的效率：自動化使文件繳交過程更快速便捷，避免有重要問題的學生無法優先處理。
* 節省成本: 我們希望透過系統節省成本。(不考慮電費，其他雜費)

目標是減少高峰期（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 |

在選課加退選的高峰期（兩週內），傳統紙本通常會造成高成本和時間消耗，影響助教的其他工作。通過引入我們設計的自動化流程，預期可以顯著減輕助教的工作量，並節省約 5,176.3 元的成本。不僅提升了加退選效率，減少了助教的負擔，還能降低紙張和列印成本，對學校和環境都帶來正面的影響。

# 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：確保資料安全和合規性，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資料，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：系統設計了精確的自動化流程處理機制，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):

* 時間限制：考慮到時間有限，我們首先選擇了較常使用的假單和選課單。

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。
* 資訊安全：如果系統發生資料洩露，將會導致師生資料外流，可能造成嚴重的後果。

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自動化辨識表單種類以及修正辨識結果，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，使用者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。掃描後的文件將被孿生網路分類器自動分類表單，並傳送至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，使用者還可以通過Line Bot進行常見問題的詢問，利用Chat GPT技術生成智能應答，幫助使用者解決問題。整個系統的自動化運作，旨在師生減少文件處理和查詢上花費的時間，提高工作和學習效率。

**(1) 系統流程說明**

* 使用者通過App或上傳圖片。
* 在 Line Bot 詢問問題，系統將透過 ChatGPT 在資料庫中尋找答案並進行回覆
* 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，進行最終處理並存儲或展示給使用者。

**(2) RAG流程說明**

如圖3-1-1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：

使用訓練好的孿生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，辨識出的字串內容、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字串進行多次檢查，直到沒有錯誤為止。



圖3-1-1 RAG流程圖

**(3) 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3-1-2所示。



圖3-1-2 分類器製作流程圖

* 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孿生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3-1-3所示。



圖3-1-3 自動分類表單流程圖

**(4) 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3-1-4 表單自動辨識與內容修正流程圖

**(5) LINE Bot 與 ChatGPT 問答流程說明**

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）。
* 使用者透過 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「這張通過沒有」。
* 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3-1-5 LINE Bot 與 ChatGPT 問答流程圖

**(6) 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，如圖11. 所示。如圖上刪除過去的對話的步驟中，刪除早期的Q1、A1的對話，只保留最近的問題和回覆，避免累積過多無用訊息。



圖3-1-6 LINE BOT多輪問答流程圖

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是圖形化的 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。
* 孿生神經網路（Siamese Neural Network）是一種特殊的神經網路結構，用於比較兩個輸入數據的相似性。該網路由兩個共享相同參數和權重的子網組成，這兩個子網分別處理不同的輸入數據，通過提取特徵並將其映射到相同的特徵空間中，最終將它們的特徵進行比較。
* GitHub是一個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、孿生神經網路 |
| 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 |
| --- | --- | --- | --- | --- | --- |
| 資料庫 | 資料庫設計-MySQL | 〇 |  | ● |  |
| 資料庫設計-Neo4j |  |  | ● |  |
| 資料庫建置-MySQL | 〇 |  | ● |  |
| 資料庫建置-Neo4j |  |  | ● |  |
| Neo4j串GPT | ● |  |  |  |
| Neo4j Api | ● |  | 〇 |  |
| 後端開發 | 伺服器架設 | 〇 |  | ● |  |
| 雲端伺服器架設 |  |  | ● |  |
| 雲端OCR辨識 | 〇 |  | ● |  |
| 本地OCR建置 | ● |  |  |  |
| 本地OCR辨識 | ● |  | 〇 |  |
| 本地OCR串接 | 〇 |  | ● |  |
| CKIPtagger佈署 |  |  | ● |  |
| 域名佈署 |  |  | ● |  |
| 伺服器API佈署 |  |  | ● |  |
| API整合 |  |  | ● | 〇 |
| Line bot | Line bot串接 | ● |  |  |  |
| ChatGPT串接 | ● |  |  |  |
| Line bot 建立 | ● |  |  |  |
| Ngrok 部署 | ● |  |  |  |
| Line bot 串接Ngrok | ● |  |  |  |
| Line bot 圖文選單 | ● |  |  |  |
| UUID產生API | ● |  |  |  |
| UUID驗證和資料查詢的API | ● |  |  |  |
| 分類器 | 分類器(孿生神經網路)建置 | ● |  |  |  |
| 分類器(孿生神經網路)辨識 | ● |  |  |  |
| 分類器(孿生神經網路)辨識串接 | 〇 |  | ● |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 11236011 周松霆 | 11236014 康海晴 | 11236016 黃凱綸 | 11236023 鍾昀臻 |
| API建立與串接 | 登入/登出 |  |  | ● |  |
| 上傳檔案 |  |  | ● |  |
| 審查進度 |  |  | 〇 | ● |
| 掃描文件 |  |  | ● |  |
| 預覽文件 |  |  | ● |  |
| 修正內容 |  |  | ● |  |
| 審查撤銷 |  |  | 〇 | ● |
| Line bot | ● |  |  |  |
| 搜尋 |  |  | 〇 | ● |
| 歷史紀錄 |  |  | 〇 | ● |
| 公告 |  |  | ● |  |
| 助教審查 |  |  | 〇 | ● |
| 教師審查 |  |  | 〇 | ● |
| 學生審閱 |  |  | 〇 | ● |
| CKIP-NER預測 |  |  | ● |  |
| 前端開發 | Android 開發環境架設 |  |  | 〇 | ● |
| 登入介面 |  |  | 〇 | ● |
| 文件辨識介面 |  |  | ● | 〇 |
| 上傳檔案介面 |  |  | 〇 | ● |
| 審查進度介面 |  |  | 〇 | ● |
| 歷史紀錄介面 |  |  | 〇 | ● |
| 搜尋介面 |  |  | 〇 | ● |
| 公告介面 |  |  | 〇 | ● |
| 助教、教師審查介面 |  |  | 〇 | ● |
| App黑暗模式 |  |  |  | ● |
| 資料蒐集與系統測試 | 資料蒐集 |  |  | ● | 〇 |
| 系統測試 | ● | 〇 |  |  |
| 測試結果統整 |  | ● | 〇 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 11236011 周松霆 | 11236014 康海晴 | 11236016 黃凱綸 | 11236023 鍾昀臻 |
| 系統流程圖設計 | App流程設計 | ● | 〇 |  |  |
| RAG流程圖 |  | ● |  |  |
| 分類器製作流程圖 |  | ● |  |  |
| 自動分類表單流程圖 |  | ● |  |  |
| 表單自動辨識與內容修正流程圖 |  | ● |  |  |
| LINE Bot 與 ChatGPT 問答流程圖 |  | ● |  |  |
| LINE BOT多輪問答流程圖 |  | ● |  |  |
| 專題競賽申請書 |  | ● |  |  |
| 專題競賽切結書 |  | ● |  |  |
| 專題競賽作品構想書 |  | ● |  |  |
| 專題競賽報名表 |  | ● |  |  |
| 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 | ● | 〇 |  |  |
| 第2章 營運計畫 |  | 〇 |  | ● |
| 第3章 系統規格 | 〇 | 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 |  | ● |  |  |
| 附錄 |  | ● |  |  |
| 報告 | 簡報製作 | ● |  | 〇 | 〇 |
| 影片拍攝 |  |  | 〇 | ● |
| 影片腳本 | ● |  |  |  |
| 影片製作 |  | ● |  |  |
| 海報製作 |  | ● |  |  |

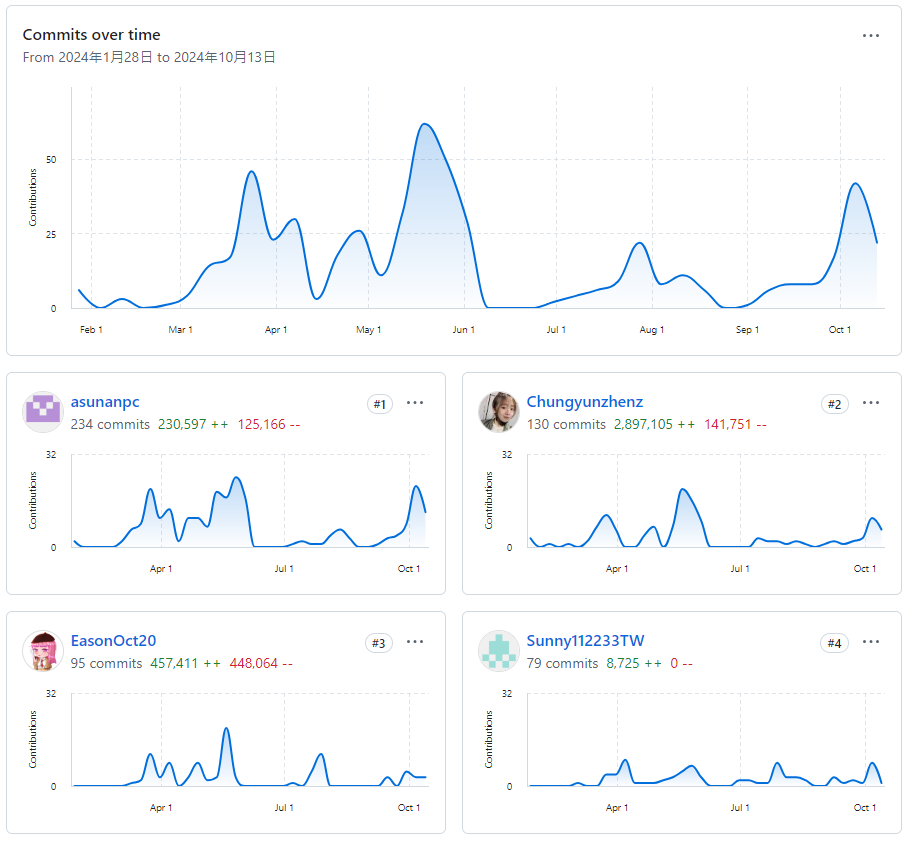
⚫ Github 各組員 commit 的次數: 

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，主要負責製作學校專題文件、校外競賽文件、設計APP功能流程、系統流程圖、海報 | **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(MySQL、Neo4j)和建置伺服器、雲端伺服器 | **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、APP大部分功能、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-D001 )

學生輸入帳號密碼登入。

* 查看審查進度(UC-D002)

學生點選審查進度，查看任一表單的審查進度。

* 審查撤銷(UC-D003)

學生點選審查進度，將任一表單的審查撤銷。

* 搜尋歷史紀錄(UC-D004)

學生點選歷史紀錄，輸入關鍵字，搜尋曾經上傳過的檔案。

* 掃描表單上傳(UC-D005)

學生將填好資料的表單先拍好照片，之後上傳至系統。

* 上傳表單資料確認(UC-D006)

上傳之後，經過系統處理，學生可以在APP介面上進一步確認資料。

* 瀏覽首頁的公告 (UC-D007)

學生可以瀏覽首頁的公告。

* Line bot機器人(UC-D008)

點擊Line bot按鈕，跳轉到Line bot介面，開始問問題。

助教、教師方:

* 助教、教師登入(UC-D009)

助教、教師輸入帳號密碼登入。

* 助教、教師查看申請單(UC-D0010)

助教、教師可以查看學生曾經上傳的申請單。

* 學生上傳表單檔案搜尋(UC-D0011)

助教、老師輸入關鍵字搜尋指定的表單。

* 助教、老師審核表單(UC-D0012)

助教、老師審核表單內容，如有問題將審核退回給學生。

* 輸入退件原因(UC-D0013)

助教、老師審核退回要輸入退件原因。

* 助教新增公告(UC-D0014)

助教可以新增一些公告，注意事項等資訊。

* 助教修改公告(UC-D0015)

助教可以更改公告。

* 助教刪除公告(UC-D0016)

助教可刪除改公告。

## 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

* 功能描述: **上傳下載**

**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**

圖 5-3-3上傳下載之活動圖

* 功能描述: 學生撤回表單

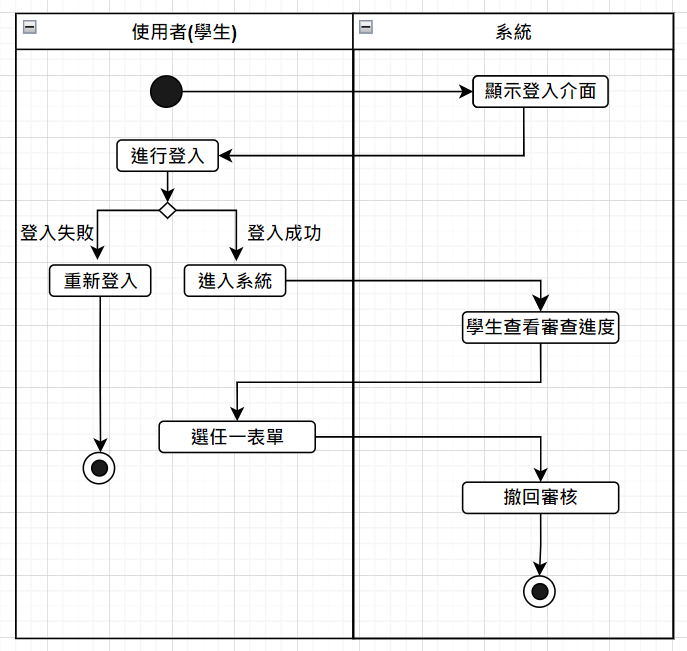


圖 5-3-4 學生撤回表單之活動圖

助教

* 功能描述: **助教、老師搜尋檔案**

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

圖 5-3-5助教搜尋檔案之活動圖

* 功能描述**:學生上傳表單檔案審核**

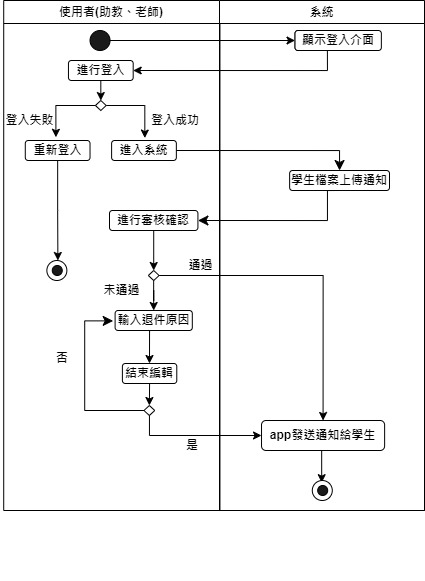


圖 5-3-6 學生上傳表單檔案審核之活動圖

## 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

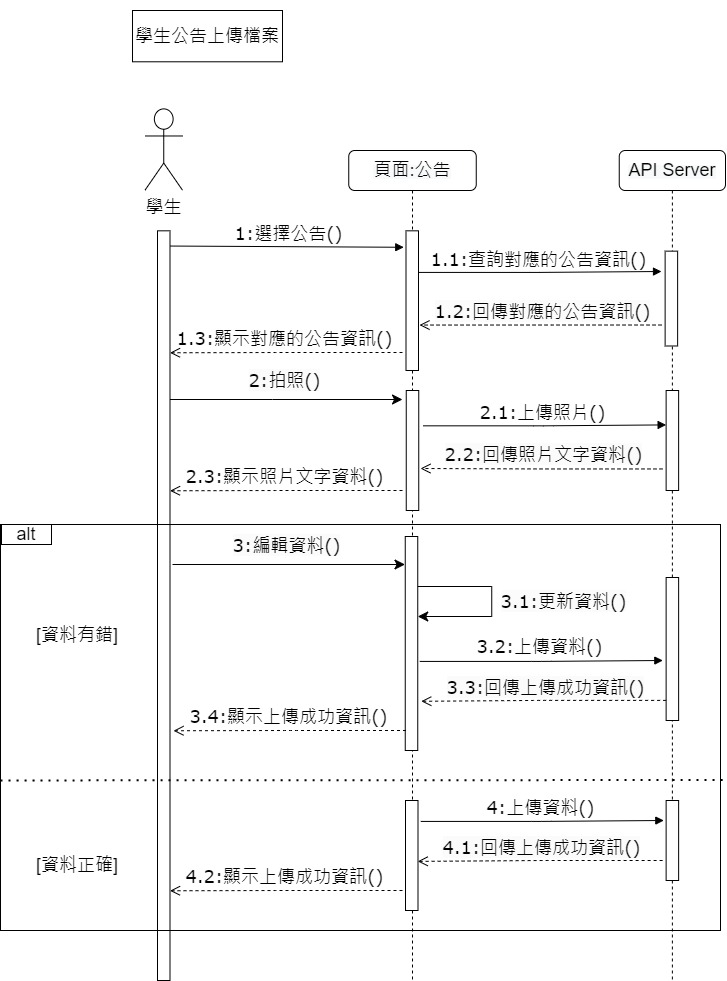


圖 6-1-4 循序圖-學生公告上傳檔案



圖 6-1-6 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(還沒放新的)

## 8-1 資料庫關聯表

圖 8-1-1 資料庫關聯表

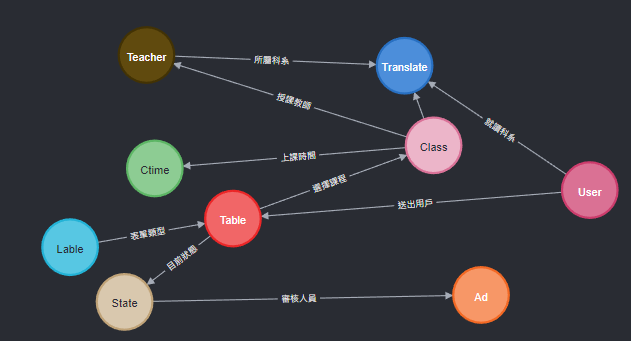


圖 8-1-2 Neo4j資料庫關聯表

## 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 9-1-1 元件清單及其規格描述表(前端)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dart檔案(前端) | | | | |
| 群組 | 編號 | 頁碼 | 檔案名稱 | 功能 |
| APP介面編寫 | 1-1-1 |  | login\_page.dart | 登入畫面 |
| 1-1-2 |  | s.dart  t.dart  z.dart | 學生主畫面  教師主畫面  助教主畫面 |
| 1-1-3 |  | file\_upload.dart | 上傳檔案畫面 |
| 1-1-4 |  | historical\_record.dart | 查詢歷史記錄畫面 |
| 1-1-5 |  | ReviewCoursePage.dart | 審核假單畫面 |
| 1-1-6 |  | ReviewLeavePage.dart | 審核選課單畫面 |
| 1-1-7 |  | stu\_review.dart | 學生審核畫面 |
| 1-1-8 |  | announcement\_page.dart | 公告畫面 |
| 1-1-9 |  | z\_download\_page.dart | 助教下載歷史紀錄畫面 |
| 1-1-10 |  | t\_download\_page.dart | 教師下載歷史紀錄畫面 |
| 1-1-11 |  | manual\_page.dart | 使用手冊畫面 |

表 9-1-2 元件清單及其規格描述表(後端)

| python檔案(後端) | | | | |
| --- | --- | --- | --- | --- |
| 群組 | 編號 | 頁碼 | 檔案名稱 | 功能 |
| 雙生  模型 | 2-1-1 |  | network8.py | 表單分類器 |
| OCR辨識自動分類表單與修正 | 2-2-1 |  | example6FetchFrame3.py | OCR核心程式，抓取表單文字框、和抓取json檔。 |
| 2-2-2 |  | train\_model2.py | OCR訓練程式 |
| 2-2-3 |  | form\_classification.py | 孿生模型—表單分類器 |
| 2-2-4 |  | mainNeo4j.py | Neo4j和ChatGPT串的api，透過GPT修正 OCR 錯誤資料的api |
| Line Bot | 2-3-1 |  | linerich\_menu.py | Line 圖文選單程式碼，建立圖文選單、綁定圖文選單給用戶進行操作 |
| 2-3-2 |  | uuid\_gen.py | 產生 UUID 的 API，用於驗證user身分依據 |
| 2-3-3 |  | linebot\_main.py | 主 Line Bot 功能的部分，主程式負責處理 Line Bot 的訊息處理邏輯，可以從已經產生的 UUID 中進行驗證和資料查詢等操作。 |
| API | 2-4-1 |  | img\_uploadapi.py | 圖片上傳、辨識、修正、儲存等相關功能 |
| 2-4-2 |  | announcement.py | 公告 |
| 2-4-3 |  | download.py | 下載檔案 |
| 2-4-4 |  | main\_app.py | Flask藍圖、api總成 |
| 2-4-5 |  | student\_review.py | 學生功能 |
| 2-4-6 |  | zuz.py | 單據功能 |
| 2-4-7 |  | z5.py | Ckip-ner |
| 2-4-8 |  | app.py | app登入 |

## 9-2 其他附屬之各種元件

表9-2-1 部分程式碼―login\_page.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-1 | 檔案名稱 | login\_page.dart |
| 目的 | 學生、教師及助教分別登入，並切換至對應的角色頁面 | | |
| 部分程式碼 | | | |
| class LoginPage:  方法 initState():  初始化帳號和密碼控制器  方法 \_loginStudent(), \_loginTeacher(), \_loginAssistant():  導航至學生、教師或助教頁面，傳遞對應角色信息  方法 build(context):  返回 Scaffold:  - Body: 包含輸入帳號、密碼及三個登入按鈕  - FloatingActionButton: 切換主題 | | | |

表9-2-2 部分程式碼―s.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-2 | 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 | | | |
| class StudentPage:  屬性 title, user, toggleTheme, isDarkMode  方法 initState():  初始化 selectedProfileImage  方法 build(context):  返回 Scaffold:  - AppBar: 顯示標題和主題切換按鈕  - Drawer: 顯示學生功能選單  - Body: 顯示歡迎信息  - FloatingActionButton: 啟動聊天機器人  方法 \_buildDrawer():  返回學生功能選單列表  方法 \_navigateTo(page):  導航至指定頁面  方法 \_logout():  導航至登入頁面  方法 \_launchLineBot():  啟動 Line 機器人 | | | |

表9-2-3 部分程式碼―file\_upload.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-3 | 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:  屬性 \_file  方法 pickFile():  確認存儲權限  若允許，選擇文件並設置 \_file  方法 uploadFile():  若未選擇文件則顯示錯誤提示  發送上傳請求，顯示成功或失敗提示  方法 \_showMessage(title, message):  顯示彈出框消息  方法 build(context):  返回 Scaffold:  - AppBar: 顯示標題  - Body: 中心顯示文件選擇和上傳按鈕，顯示選擇的文件名 | | | |

表9-2-4 部分程式碼―historical\_record.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-4 | 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 | | | |
| class HistoryPage:  屬性 historyData, keywordController, searchType, isLoading, errorMessage  方法 initState():  初始化，調用 fetchHistoryData()  方法 fetchHistoryData():  設置 isLoading 為 true  獲取歷史紀錄數據，設置 historyData，若失敗則調用 \_showError()  方法 \_showError(message):  設置 errorMessage 並顯示錯誤提示  方法 build(context):  返回 Scaffold:  - AppBar: 顯示標題  - Body: 包含搜尋選項和歷史紀錄列表  方法 \_buildDropdown():  返回下拉框選擇搜尋類型  方法 \_buildSearchField():  返回搜尋輸入框  方法 \_buildHistoryList():  若無歷史紀錄則顯示 "無歷史紀錄"，否則顯示歷史紀錄列表 | | | |

表9-2-5 部分程式碼―ReviewCoursePage.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-5 | 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:  屬性 pendingRequests, returnedRequests, completedRequests, isLoading  方法 initState():  初始化，調用 \_fetchCourseRequests()  方法 \_fetchCourseRequests():  獲取選課單數據，分為審查中、退回、已完成，設置 pendingRequests、returnedRequests、completedRequests  方法 \_showError(message):  顯示錯誤提示  方法 build(context):  返回 Scaffold:  - AppBar: 顯示標題和 TabBar  - Body: 根據 isLoading 顯示加載動畫或選課單列表  方法 \_buildTabViews():  返回包含三個 Tab 的 TabBarView:  - 審查中選課單  - 退回選課單  - 已完成選課單  方法 \_buildTabContent(requests):  若選課單數據為空顯示 "沒有選課單"，否則顯示選課單列表  方法 \_showDetails(details):  彈出對話框顯示選課單詳細信息，包括課程名稱、選課時間和學期 | | | |

表9-2-6 部分程式碼―ReviewLeavePage.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-6 | 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:  屬性 pendingRequests, returnedRequests, completedRequests, isLoading  方法 initState():  初始化，調用 \_fetchLeaveRequests()  方法 \_fetchLeaveRequests():  獲取請假單數據，分為審查中、退回、完成，設置 pendingRequests、returnedRequests、completedRequests  方法 \_showError(message):  顯示錯誤提示  方法 build(context):  返回 Scaffold:  - AppBar: 顯示標題和 TabBar  - Body: 根據 isLoading 顯示加載動畫或請假單列表  方法 \_buildTabViews():  返回包含三個 Tab 的 TabBarView:  - 審查中請假單  - 退回請假單  - 完成請假單  方法 \_buildTabContent(requests):  若請假單數據為空顯示 "沒有請假單"，否則顯示請假單列表  方法 \_showDetails(details):  彈出對話框顯示請假單詳細信息，包括請假原因、請假時間和學期 | | | |

表9-2-7 部分程式碼―stu\_review.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-7 | 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9-2-8 部分程式碼―t\_download.dart

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1-10 | 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9-2-9 部分程式碼―linebot\_main.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-3-3 | 檔案名稱 | linebot\_main.py |
| 目的 | 主 Line Bot 功能的部分，主程式負責處理 Line Bot 的訊息處理邏輯，可以從已經產生的 UUID 中進行驗證和資料查詢等操作。 | | |
| 部分程式碼 | | | |
| # 建立 Flask 應用程式  app = Flask(\_\_name\_\_)  # 初始化 LINE Bot API，用於處理用戶消息  access\_token = 'YOUR\_ACCESS\_TOKEN'  secret = 'YOUR\_SECRET'  line\_bot\_api = LineBotApi(access\_token)  handler = WebhookHandler(secret)  # 判斷是否是有效的 UUID  FUNCTION is\_valid\_uuid(uuid\_to\_test):  TRY:  uuid\_obj = uuid.UUID(uuid\_to\_test, version=4) #將輸入轉換為 UUID 格式  RETURN True  EXCEPT ValueError:  RETURN False  # 設定 LINE Bot 回調函數  @app.route("/callback", methods=['POST'])  FUNCTION linebot():  body = request.get\_data(as\_text=True) # 接收 POST 請求的消息體  signature = request.headers.get('X-Line-Signature', '') # 簽名驗證  TRY:  handler.handle(body, signature) # 驗證和處理消息  json\_data = json.loads(body) # 將消息轉為 JSON 格式  tk = json\_data['events'][0]['replyToken'] # 提取回覆 token  msg\_type = json\_data['events'][0]['message']['type'] # 判斷消息類型  # 連接資料庫  conn = mysql.connector.connect(\*\*db\_config)  cursor = conn.cursor()  IF msg\_type == 'text': # 如果用戶發送的是文字消息  msg = json\_data['events'][0]['message']['text'].strip() # 獲取文字消息  # 如果消息不是正確的 UUID 格式，則回覆錯誤提示  IF NOT is\_valid\_uuid(msg):  reply = "請輸入正確的UUID格式"  line\_bot\_api.reply\_message(tk, TextSendMessage(reply))  RETURN 'OK'  # 查詢資料庫中的 UUID 對應使用者  cursor.execute("SELECT user\_name, user\_id FROM UserInfo WHERE verification\_code = %s", (msg,))  user\_info = cursor.fetchone()  IF user\_info: # 如果找到對應的使用者  user\_name, user\_id = user\_info  reply = f"歡迎 {user\_name}！"  # 查詢該使用者的審核狀態  cursor.execute("SELECT status FROM verificationState WHERE user\_id = %s", (user\_id,))  verification\_state = cursor.fetchone()  IF verification\_state:  reply += f" 您的審核狀態為: {verification\_state[0]}" # 顯示審核狀態  ELSE:  reply += " 目前尚無審核資料。" # 無資料時的回覆  ELSE:  reply = "找不到此UUID的使用者資訊。" # 找不到用戶時回覆  # 回應用戶的消息  line\_bot\_api.reply\_message(tk, TextSendMessage(reply))  cursor.close() # 關閉游標  conn.close() # 關閉資料庫連接  EXCEPT LineBotApiError AS e: # 處理 LINE Bot API 錯誤  PRINT(f"LineBotApiError: {e.status\_code} {e.error.message}")  EXCEPT InvalidSignatureError:  abort(400) # 如果簽名驗證失敗  EXCEPT Exception AS e:  PRINT(f"Error: {e}")  PRINT(body) # 顯示錯誤和消息內容  RETURN 'OK'  # 主程式啟動 Flask 伺服器，監聽 5000 端口  IF \_\_name\_\_ == "\_\_main\_\_":  app.run(port=5000) | | | |

表9-2-10 部分程式碼―mainNeo4j.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2-4 | 檔案名稱 | mainNeo4j.py |
| 目的 | Neo4j和ChatGPT串的api，透過GPT修正 OCR 錯誤資料的api | | |
| 部分程式碼 | | | |
| # 建立一個 Flask 應用程式  建立 Flask 應用程式  # 設定 OpenAI API 金鑰  設定 OpenAI API 金鑰  # 連接到 Neo4j 資料庫  連接 Neo4j 資料庫  函數 generate\_prompt(欄位名稱, 原始文字, 修正後文字=None):  如果欄位名稱是 "學號":  建立學號檢查的 prompt  如果欄位名稱是 "姓名":  建立姓名檢查的 prompt  如果欄位名稱是 "請假事由":  建立請假事由檢查的 prompt  如果欄位名稱是 "申請科目":  建立申請科目檢查的 prompt  否則:  建立其他欄位的 prompt  返回 prompt  函數 correct\_with\_chatgpt(欄位名稱, 原始文字, 修正後文字=None):  根據欄位名稱和原始文字生成 prompt  使用 OpenAI ChatCompletion API 調用 ChatGPT 模型  返回 ChatGPT 的回應  函數 query\_neo4j(欄位名稱, 原始文字):  如果欄位名稱是 "學號":  執行查詢，根據學號比對資料庫  如果欄位名稱是 "科目":  執行查詢，根據科目名稱比對資料庫  否則:  返回 None  如果查詢結果存在:  返回查詢到的正確資料  否則:  返回 None  # 定義主路由，處理表單比對與修正  主路由 /check\_form:  嘗試:  指定要讀取的 JSON 檔案路徑  讀取 JSON 檔案  定義需要檢查的欄位 ["學號", "姓名", "請假事由", "申請科目"]  對每個欄位進行檢查:  找到對應的 OCR 結果  如果找到對應的 OCR 結果:  在 Neo4j 中查詢比對  如果 Neo4j 中有正確資料:  使用資料庫的正確資料  否則:  使用 ChatGPT 進行文本修正  儲存修正後的 JSON 檔案  返回成功訊息和修正後的資料  如果出錯:  返回錯誤訊息  # 啟動 Flask 應用程式，監聽 5005 端口 | | | |

表9-2-11 部分程式碼―app.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-4-8 | 檔案名稱 | app.py |
| 目的 | 處理app登入請求 | | |
| 部分程式碼 | | | |
| // 導入必要的模組  初始化 Blueprint 從 Flask  初始化 request 和 jsonify 從 Flask  初始化 mysql.connector  初始化 bcrypt 從 passlib.hash  // 設置 login Blueprint  定義 login\_bp 為 Blueprint，名稱為 'login\_bp'  // 配置 MySQL 連接  定義 db\_config，包含 host '127.0.0.1'，user 'root'，password，database  // 定義 login 路由  定義路由 '/login' 使用 POST 方法  // 登入函數  函數 login():  data = 從 request 獲取 JSON 數據  student\_id = 從 data 獲取 'StudentID'  password = 從 data 獲取 'Password'  如果 student\_id 或 password 缺失:  返回 JSON 響應，包含錯誤 '缺少 StudentID 或 Password' 和狀態 400  嘗試:  // 連接到 MySQL  cnx = 使用 db\_config 連接到 MySQL  cursor = 創建 cursor，dictionary=True  query = "SELECT \* FROM users WHERE StudentID = %s"  // 執行查詢  使用 student\_id 執行查詢  user = 從 cursor 獲取一個結果  如果 user 存在且 user['Password'] 存在且 bcrypt 驗證 password 和 user['Password']:  關閉 MySQL 連接  返回 JSON 響應，包含成功信息和用戶詳情，狀態 200  否則:  關閉 MySQL 連接  返回 JSON 響應，包含錯誤 '無效的 StudentID 或 Password' 和狀態 401  除了 MySQL 連接錯誤 當 err 發生:  返回 JSON 響應，包含來自 err 的錯誤信息，狀態 500 | | | |

表9-2-12 部分程式碼―img\_uploadapi.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-4-1 | 檔案名稱 | img\_uploadapi.py |
| 目的 | 接收上傳圖片並進行辨識，json生成並利用chatgpt修正，然後回傳預覽圖 | | |
| 部分程式碼 | | | |
| // 導入必要的庫和模組  導入 OpenAI API 庫  導入 JSON 處理庫  導入操作系統相關功能庫  導入臨時文件處理庫  導入 MySQL 資料庫連接器  從 Flask 中導入應用程序和請求相關功能  從 werkzeug.utils 中導入安全文件名處理函數  從 datetime 中導入日期時間處理功能  從 PaddleOCR 中導入 OCR 功能  導入 numpy 數值計算庫  導入 TensorFlow 及相關神經網絡模組  導入 OpenCV 圖像處理庫  導入隨機數處理庫  導入 io 流處理庫  從 PIL 中導入圖像處理功能  導入 opencc 進行簡體轉繁體  從 py2neo 中導入 Neo4j 資料庫連接器  // 設定 OpenAI API 金鑰（使用者需填寫自己的金鑰）  設置 openai.api\_key 為空字符串  // 連接到 Neo4j 資料庫（使用者需提供認證信息）  建立到 Neo4j 資料庫的連接，設置認證資訊  // 定義函數：生成提示語句  函數 generate\_prompt(field\_name, original\_text, correct\_text=None):  根據 field\_name 判斷需要生成的提示語句  返回對應的提示語句 prompt  // 定義函數：使用 ChatGPT 進行文本校正  函數 correct\_with\_chatgpt(field\_name, original\_text, correct\_text=None):  調用 generate\_prompt 生成 prompt  使用 OpenAI 的 ChatCompletion 介面進行文本校正  返回校正後的文本  // 定義函數：在 Neo4j 中查詢資料  函數 query\_neo4j(field\_name, original\_text):  根據 field\_name 構建查詢語句  執行查詢並取得結果  如果有匹配結果，返回相應的名稱  否則，返回 None  // 設置 Flask API 路由  創建 Blueprint 'img\_uploadapi'  // 配置資料庫連接資訊  設置資料庫連接配置，包括主機、使用者、密碼和資料庫名稱  // 設定隨機數種子以確保結果可重現  設置 TensorFlow、NumPy 和 random 的隨機數種子為固定值  // 初始化 PaddleOCR 用於文字識別  創建 PaddleOCR 實例，設定參數 use\_angle\_cls=True 和語言為中文  // 定義函數：創建基礎神經網絡  函數 create\_base\_network(input\_shape):  使用預訓練的 ResNet50 作為基礎模型  添加 Flatten 層和 Dense 層進行特徵提取  返回構建的模型  // 定義函數：計算餘弦距離  函數 cosine\_distance(vects):  對向量進行 L2 正則化  計算並返回 1 減去兩個向量的內積  // 定義函數：載入並預處理圖像  函數 load\_and\_preprocess\_image(image\_path, img\_width=224, img\_height=224):  讀取灰度圖像  調整圖像大小  將灰度圖像轉換為三通道  將像素值歸一化到 [0, 1] 範圍  返回預處理後的圖像數據  // 設置相似度模型的參數和結構  定義圖像高度和寬度為 224  設置輸入形狀 input\_shape  創建基礎網絡 base\_network  定義兩個輸入層 input\_a 和 input\_b  通過基礎網絡處理兩個輸入，得到 processed\_a 和 processed\_b  計算兩個處理後輸出的餘弦距離，得到 distance  構建最終的模型 model，並編譯模型  // 加載字體（檢查字體文件是否存在）  設定字體路徑為 "C:/Windows/Fonts/mingliu.ttc"  如果字體文件不存在，拋出錯誤提示  使用 ImageFont.truetype 加載字體，字體大小為 20  // 初始化簡體轉繁體的轉換器  創建 OpenCC 實例，設置轉換模式為 's2t'  // 定義上傳文件的 API 路由  在 img\_uploadapi 中定義路由 '/upload'，只接受 POST 請求  函數 upload\_file():  獲取上傳的文件  如果文件存在：  獲取安全的文件名  嘗試：  讀取文件內容並檢查是否為二進制格式  設置上傳者 ID 為 '213030'  連接到 MySQL 資料庫  查詢 users 表確認上傳者是否存在  如果不存在，返回錯誤信息  創建臨時文件保存上傳的圖像  使用預處理函數載入上傳的圖像，得到 ocr\_image  載入參考圖像 reference\_image，用於圖像分類  使用模型進行預測，確定圖像類型（'選課單' 或 '請假單'）  將圖像和相關信息插入到 imageuploads 資料表中  提交資料庫變更  查詢最近插入的記錄以獲取 imageupload\_id  使用 PaddleOCR 進行文字識別，獲取 ocr\_result  初始化 ocr\_data 列表  遍歷識別結果，進行簡體轉繁體，並將結果添加到 ocr\_data 中  定義需要校正的欄位列表 fields\_to\_check  遍歷 fields\_to\_check，對每個欄位進行校正：  在 ocr\_data 中查找對應的文本  如果找到，嘗試在 Neo4j 中查詢正確的文本  如果 Neo4j 無結果，使用 ChatGPT 進行校正  將校正後的文本添加到 ocr\_data 中  將 ocr\_data 轉換為 JSON 格式  將 JSON 數據插入到 json\_data 資料表中  創建新的圖像，並在圖像上繪製校正後的文本  將圖像保存到內存中，得到二進制數據 img\_blob  關閉資料庫連接和游標  刪除臨時文件  返回圖像響應，內容類型為 'image/png'  如果發生資料庫錯誤，返回錯誤信息  如果發生其他異常，返回錯誤信息  否則，返回錯誤信息表示未提供文件 | | | |

表9-2-13 部分程式碼―z5.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-4-7 | 檔案名稱 | z5.py |
| 目的 | 接收資料並進行Ckip-ner預測，並通過關聯式資料庫找出與預測結果相關的內容 | | |
| 部分程式碼 | | | |
| // 從 Flask 和其他模組中導入必要的函數和類  導入 Flask、request、jsonify、Blueprint、current\_app  從 ckiptagger 導入資料工具、WS、POS、NER、建構字典  導入 pymysql 用於 MySQL 資料庫連接  導入 logging 模組  導入 TensorFlow  導入 json 模組  從 neo4j 導入 GraphDatabase  從 passlib.hash 導入 bcrypt  導入 os 和 time 模組  // 建立 Flask Blueprint  建立名為 z5 的 Blueprint  // 設定 TensorFlow 的日誌等級  將 TensorFlow 的日誌等級設定為 ERROR  // 下載並載入 CKIPtagger 模型  嘗試：  下載 CKIPtagger 的資料到目前目錄  載入分詞模型 ws  載入詞性標註模型 pos  載入命名實體識別模型 ner  如果失敗：  記錄錯誤並拋出異常  // 資料庫配置  設定 MySQL 資料庫的連線配置，包括主機、使用者、密碼和資料庫名稱  設定 Neo4j 資料庫的連線配置，包括 URI、使用者和密碼  // 追蹤自訂字典檔案的最後修改時間  初始化 custom\_dict\_last\_modified 和 custom\_dict\_cache 為 None  // 定義函數 connect\_db():  嘗試連接到 MySQL 資料庫  如果成功，回傳連線對象  如果失敗，記錄錯誤並傳回 None  // 定義函數 connect\_neo4j():  嘗試連接到 Neo4j 資料庫  如果成功，回傳驅動對象  如果失敗，記錄錯誤並傳回 None  // 定義函數 load\_custom\_dict(file\_path):  初始化空的 custom\_dict 字典  嘗試開啟指定的文件，逐行讀取內容  將每行拆分為字詞、詞性和頻率，加入 custom\_dict 中  如果失敗，記錄錯誤  返回 custom\_dict  // 定義函數 check\_and\_load\_custom\_dict(file\_path):  取得文件的最後修改時間  如果 custom\_dict\_last\_modified 為 None 或檔案已被修改：  記錄訊息，表示重新載入自訂字典  呼叫 load\_custom\_dict 重新載入字典  更新 custom\_dict\_last\_modified  返回 custom\_dict\_cache  // 在 Blueprint z5 中定義路由 '/predict'，只接受 POST 請求  定義函數 predict():  從請求中取得 JSON 數據  記錄收到的數據  取得句子 sentence  如果未提供句子，返回錯誤  檢查並載入自訂字典 custom\_dict  使用 custom\_dict 建立使用者字典 use​​r\_dict  使用 CKIPtagger 的 ws 模型進行分詞，使用 user\_dict 作為推薦字典  使用 pos 模型進行詞性標註  將自訂詞性的結果應用到分詞和詞性標註結果中  記錄 WS 和 POS 的結果  執行命名實體識別 ner  記錄 NER 的結果  如果 NER 和 POS 結果都無法滿足要求，回傳錯誤  嘗試連接到 Neo4j 資料庫  如果連線失敗，回傳錯誤  初始化 persons, dates, neu\_categories, class\_categories 和 neo4j\_results 列表  從 NER 結果中提取 PERSON 和 DATE 類型的實體  從 POS 結果中提取 Neu 和 class 類型的詞語  使用 Neo4j 驅動，在會話中：  對於每個 person，在圖中查詢對應的節點和關係，並加入 neo4j\_results 中  對於每個 date，同樣進行查詢  對於每個 neu\_category，同樣進行查詢  對於每個 class\_category，同樣進行查詢  關閉 Neo4j 驅動  將 NER 結果轉換為 JSON 字串 ner\_results\_str  嘗試連接到 MySQL 資料庫  如果連線失敗，回傳錯誤  使用資料庫遊標，將句子和 NER 結果插入到 ner\_results 表中  提交資料庫更改  關閉遊標和資料庫連接  傳回成功訊息和結果，包括 NER 結果和 Neo4j 查詢結果  如果發生異常，記錄錯誤並傳回錯誤訊息 | | | |

表9-2-14 部分程式碼―student\_review.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-4-5 | 檔案名稱 | student\_review.py |
| 目的 | 學生相關功能 | | |
| 部分程式碼 | | | |
| // 從 flask 模組中導入 Blueprint、request、jsonify  從 flask 中導入 Blueprint、request、jsonify  // 導入 mysql.connector，用於連接 MySQL 資料庫  導入 mysql.connector  // 從 flask\_cors 中導入 CORS，用於處理跨域資源共享  從 flask\_cors 中導入 CORS  // 導入 threading 模組，用於多執行緒處理  導入 threading  // 導入 time 模組，用於時間相關操作  導入 time  // 建立 Flask 的 Blueprint，用於組織路由  創建名為 student\_review\_bp 的 Blueprint  // 資料庫配置  設定資料庫連接資訊，包括主機、使用者、密碼和資料庫名稱  // 初始化一個字典，用於儲存學生的更新狀態  初始化 student\_updates 為空字典  // 定義路由：查詢學生審查狀況  在 student\_review\_bp 中定義 GET 路由 '/getStudentReviews'  函數 get\_student\_reviews():  從請求中獲取參數 review\_status  // 驗證 review\_status 參數是否有效  定義有效的狀態列表 valid\_statuses，包括 "審查中"、"退回"、"通過"、"撤回"  如果提供了 review\_status，且不在 valid\_statuses 中：  返回錯誤訊息，指出無效的 review\_status 參數  // 連接到資料庫  嘗試：  使用資料庫配置連接到 MySQL 資料庫  如果連接失敗，返回錯誤訊息  // 創建資料庫游標，設定返回字典格式的結果  創建 cursor，dictionary=True  // 查詢 ReviewProgress 表，根據是否提供了 review\_status 進行過濾  如果提供了 review\_status：  執行 SQL 查詢，選擇符合指定 review\_status 的記錄  否則：  執行 SQL 查詢，選擇所有記錄  // 獲取查詢結果  使用 cursor.fetchall() 獲取結果  // 返回查詢結果的 JSON 表示  返回 jsonify(result)  如果發生資料庫錯誤，返回錯誤訊息  如果發生其他異常，返回錯誤訊息  最後：  關閉資料庫連接和游標  // 定義路由：更新審查狀態  在 student\_review\_bp 中定義 POST 路由 '/updateReviewStatus'  函數 update\_review\_status():  從請求中獲取 JSON 數據 data  獲取 data 中的 id 和 new\_status  // 驗證 new\_status 是否有效  定義有效的狀態列表 valid\_statuses，包括 "審查中"、"退回"、"通過"、"撤回"  如果 new\_status 不在 valid\_statuses 中：  返回錯誤訊息，指出無效的 new\_status 參數  // 檢查是否提供了 id  如果未提供 id：  返回錯誤訊息，指出缺少 id 參數  // 連接到資料庫  嘗試：  使用資料庫配置連接到 MySQL 資料庫  如果連接失敗，返回錯誤訊息  // 創建資料庫游標  創建 cursor  // 更新 ReviewProgress 表中指定 id 的記錄，設定 review\_status 為 new\_status  執行 SQL 更新語句  提交資料庫變更  // 檢查是否有記錄被更新  如果 cursor.rowcount 為 0：  返回錯誤訊息，指出未找到具有給定 id 的審查  // 更新 student\_updates 字典，記錄新的狀態和時間戳  將 student\_updates[id] 設為包含 status 和 timestamp 的字典  // 返回成功訊息，指出審查狀態已更新  如果發生資料庫錯誤，返回錯誤訊息  如果發生其他異常，返回錯誤訊息  最後：  關閉資料庫連接和游標  // 定義路由：撤回審查  在 student\_review\_bp 中定義 POST 路由 '/withdrawReview'  函數 withdraw\_review():  從請求中獲取 JSON 數據 data  獲取 data 中的 review\_id  // 檢查是否提供了 review\_id  如果未提供 review\_id：  返回錯誤訊息，指出缺少 review\_id 參數  // 連接到資料庫  嘗試：  使用資料庫配置連接到 MySQL 資料庫  如果連接失敗，返回錯誤訊息  // 創建資料庫游標  創建 cursor  // 更新 ReviewProgress 表中指定 id 的記錄，設定 review\_status 為 "撤回"  執行 SQL 更新語句  提交資料庫變更  // 檢查是否有記錄被更新  如果 cursor.rowcount 為 0：  返回錯誤訊息，指出未找到具有給定 id 的審查  // 更新 student\_updates 字典，記錄新的狀態和時間戳  將 student\_updates[id] 設為包含 status "撤回" 和 timestamp 的字典  // 返回成功訊息，指出審查狀態已撤回  如果發生資料庫錯誤，返回錯誤訊息  如果發生其他異常，返回錯誤訊息  最後：  關閉資料庫連接和游標  // 定義路由：查詢審查更新  在 student\_review\_bp 中定義 GET 路由 '/getReviewUpdates'  函數 get\_review\_updates():  從請求中獲取參數 id  // 檢查是否提供了 id  如果未提供 id：  返回錯誤訊息，指出缺少 id 參數  // 檢查 student\_updates 字典中是否有給定的 id  如果 id 不在 student\_updates 中：  返回錯誤訊息，指出給定 id 無可用更新  // 返回 student\_updates[id] 的 JSON 表示  返回 jsonify(student\_updates[id]) | | | |

# 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10-2-1 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 | 成功 |

**B. 學生**

表10-2-2 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 | 成功 |

表10-2-3 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10-2-4 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 | 成功 |

表10-2-5 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 | 成功 |

表10-2-6 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 | 成功 |

表10-2-7 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 | 成功 |

表10-2-8 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 | 成功 |

表10-2-9 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 | 成功 |

表10-2-10 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 | 成功 |

**C. 助教**

表10-2-11 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 | 成功 |

表10-2-12 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 | 成功 |

表10-2-13 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 | 成功 |

表10-2-14 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 | 成功 |

表10-2-15 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 | 成功 |

表10-2-16 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 | 成功 |

**D. 教師**

表10-2-17 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 | 成功 |

表10-2-18 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 | 成功 |

表10-2-19 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 | 成功 |

表10-2-20 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 | 成功 |

表10-2-21 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 | 成功 |

表10-2-22 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 | 成功 |

# 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 | |
| C:\Users\sunny\Downloads\UIH9TT_qrcode.png  圖11-2-1 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(補畫面)

## 12-1 使用者操作

(1)登入流程

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996650_0.jpg  圖 | C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 |
| 1. 開啟APP輸入帳號密碼選擇身分登入 | 1. 登入成功至首頁 |

(2)學生上傳圖片

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 | C:\Users\sunny\Downloads\S__5996645_0.jpg  圖 |
| 1.在學生首頁，點選上傳圖片 | 2.來到文件上傳畫面 |
|  |  |
| 3.確認表單 | 4.確認上傳 |

(3)學生查看審查進度

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 |  |
| 1.在學生首頁，點選審查進度 | 2.點選任一審查，即可看到審查進度 |
|  |  |

(4)學生查詢歷史紀錄

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 | C:\Users\sunny\Downloads\S__5996653_0.jpg  圖 |
| 1.在學生首頁，點選歷史紀錄 | 2.選擇關鍵字種類 |
|  |  |
| 3.輸入關鍵字搜尋 | 4.搜尋結果顯示 |

(5)學生查看公告

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 | 圖 |
| 1.在學生首頁，點選公告 | 2.即可看到最新公告 |
|  |  |

(6)師生查看使用手冊

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 |  |
| 1.在學生首頁，點選公告 | 2.即可看到最新公告 |
|  |  |

(7)學生在Line Bot問問題

|  |  |
| --- | --- |
| C:\Users\sunny\Downloads\S__5996651_0.jpg  圖 |  |
| 1.在學生首頁，點選訊息圖標 | 2.跳轉到Line Bot |
|  |  |
| 3. 在Line Bot開始問問題 |  |

(3)老師審查請假單(審查通過操作)

|  |  |
| --- | --- |
|  | C:\Users\sunny\Downloads\S__5996647_0.jpg  圖錯的 |
| 1.在老師首頁，點選查看請假單進度 | 2.點選其中一個請假單審查 |
| 圖 | 圖 |
| 3.點選審核通過 | 4.請假單會到已完成的地方 |

(3)續老師審查請假單(審查退回操作)

|  |  |
| --- | --- |
| 圖 | 圖 |
| 1.在老師首頁，點選查看請假單進度 | 2.點選其中一個請假單審查 |
| 圖 | 圖 |
| 3.點選返回審核 | 4.輸入退回原因並按下確定退回 |
| C:\Users\sunny\Downloads\S__5996647_0.jpg  圖 | 圖 |
| 5.請假單到退回的地方並點擊查看內容 |  |

# 第13章 感想

* 11236011周松霆

透過此次專題，收穫了不少，其中我認為收穫最多的是自身的成長，專題一路下來，我覺得除了技術方面以外，收穫最多的是心境上的成長。在專題的過程中，除了要應對各種技術難題，還需要學會如何有效地與團隊成員溝通和合作。在遇到困難時，不能再像以前一樣只著眼於問題本身，而是開始嘗試從不同的角度去思考解決的方法。

專題中所收穫的成長，讓我更加堅定了未來的目標和方向。我相信這段經歷將成為我職涯中的一個重要基石，幫助我在未來的工作和生活中更加從容自信地面對挑戰，並持續不斷地提升自己。

* 11236014康海晴

在這次專題中，我收穫了許多寶貴的經驗。剛開始時，我對機器學習幾乎毫無了解，但在老師的引導下，我們透過推薦的書籍自主學習，並每週進行討論交流，逐步掌握了基礎知識，開始認識到自己的能力範圍和可行的方向。透過循序漸進的學習，我們最終選定了合適且感興趣的主題。

在專題製作過程中，團隊成員各司其職，並透過互相溝通，學習如何更有效率地在限期內完成專題。我負責大部分文件撰寫與設計流程圖的工作，這使我學會了如何將複雜的研究內容轉化為更易理解的流程圖。在此過程中，我不僅需要深入了解研究框架，還必須融入自己的理解，並查閱相關資料來補充知識。為了清楚表達我們的核心研究，還需撰寫流程解釋，確保文件內容的重點不偏離研究主題。

專題製作過程充滿挑戰，但這次專題對我來說非常有意義。我從中學習到很多知識，不僅加深了對相關技術的理解，也提升了專業表達和文件撰寫的能力。特別感謝老師和組員的指導與幫助，讓我在這段旅程中能夠成長。期待未來能繼續學習與成長，並運用這次的經驗應對更多挑戰。

* 11236016黃凱綸

在這次專題的過程中，我學習到許多實用的技術與知識，特別是在後端API撰寫、伺服器架設、CKIP以及關聯式資料庫的使用上得到了寶貴的經驗。剛開始時，對這些技術並不熟悉，但隨著不斷學習與實踐，我逐漸掌握了它們的核心概念與應用技巧。這段學習旅程不僅僅是技術上的提升，更重要的是心態的轉變。面對複雜的技術問題時，我學會了如何冷靜分析，嘗試從多種角度思考，找到適當的解決方法。

這次專題的成功，離不開團隊成員之間的合作與溝通。我們每個人都各司其職，互相支持、協調，讓整個專案得以順利推進。在過程中，遇到了不少挑戰，比如技術難題或時間上的壓力，但我們都透過討論與交換意見，找到了解決方案，這也教會了我如何在團隊中保持開放心態，接納他人的建議。

當遇到挫折時，保持冷靜尤為重要。我學到了要暫時停下來，回顧問題的根本原因，並逐步拆解，這樣能更好地處理壓力。同時，也會適當調整自己的情緒，避免因壓力過大而影響專案進度。在這個過程中，我不僅提高了問題解決的能力，也學會了如何在壓力下保持專注。

總體而言，這次專題不僅讓我技術上有了成長，也讓我在團隊合作、時間管理以及情緒調整方面有了更深刻的體會。我相信，這些經驗將在我未來的職業生涯中為我打下良好的基礎，並讓我面對未來的挑戰時更加自信與從容。

* 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 |