|  |
| --- |
| 國立臺北商業大學 |
| 資訊管理系 |
| 1 1 4 資訊系統專案設計 |
| 系統手冊 |
|  |
| 組 別：第114408組  題 目：E筆勾銷  指導老師：蒯思齊老師  組 長：11146023 翁益德  組 員：1114C007 周品君 11146020鄭名棋 11146031葉勇杰 |
| 中華民國1 1 4年n月n日 |

目錄

[第１章 前言 4](#_Toc197783454)

[1-1 背景介紹 4](#_Toc197783455)

[1-2 動機 4](#_Toc197783456)

[1-3 系統目的與目標 4](#_Toc197783457)

[1-4 預期成果 4](#_Toc197783458)

[第2章 營運計畫 5](#_Toc197783459)

[2-1 可行性分析 5](#_Toc197783460)

[2-1-1 技術可行性 5](#_Toc197783461)

[2-1-2 市場可行性 5](#_Toc197783462)

[2-1-3 經濟可行性 5](#_Toc197783463)

[2-2 商業模式－Business model 6](#_Toc197783464)

[2-3 市場分析－STP 6](#_Toc197783465)

[2-3-1 市場區隔(Segmentation) 6](#_Toc197783466)

[2-3-2 目標市場(Targeting) 7](#_Toc197783467)

[2-3-3 產品定位(Positioning) 7](#_Toc197783468)

[2-3-4 競爭力分析－SWOT 7](#_Toc197783469)

[第3章 系統規格 8](#_Toc197783470)

[3-1 系統架構 8](#_Toc197783471)

[3-2 系統軟、硬體需求與技術平台 9](#_Toc197783472)

[3-3 開發標準與使用工具 9](#_Toc197783473)

[第4章 專案時程與組織分工 11](#_Toc197783474)

[4-1 專案時程 11](#_Toc197783475)

[4-2 專案組織與分工 11](#_Toc197783476)

[4-3 Github上傳紀錄 13](#_Toc197783477)

[第5章 需求模型 14](#_Toc197783478)

[5-1 需求清單 14](#_Toc197783479)

[5-2 使用個案圖(Use case diagram) 15](#_Toc197783480)

[5-3 使用個案描述 15](#_Toc197783481)

[5-4 分析類別圖(Analysis class diagram) 15](#_Toc197783482)

[第6章 設計模型 16](#_Toc197783483)

[6-1 循序圖(Sequential diagram) 16](#_Toc197783484)

[6-2 設計類別圖(Design class diagram) 16](#_Toc197783485)

[第7章 實作模型 17](#_Toc197783486)

[7-1 佈署圖(Deployment diagram) 17](#_Toc197783487)

[7-2 套件圖(Package diagram) 17](#_Toc197783488)

[7-3 元件圖(Component diagram) 17](#_Toc197783489)

[7-4 狀態機(State machine) 17](#_Toc197783490)

[第8章 資料庫設計 18](#_Toc197783491)

[8-1 資料庫關聯表 18](#_Toc197783492)

[8-2 表格及Mata data 19](#_Toc197783493)

圖目錄

[圖：2-1‑1系統可行性分析 5](#_Toc197783604)

[圖：2-2‑1商業模式 6](#_Toc197783605)

[圖：2-3‑1市場分析STP 6](#_Toc197783606)

[圖：3-1‑1系統架構圖 8](#_Toc197783607)

[圖：4-1‑1甘特圖 11](#_Toc197783608)

[圖：8-1‑1資料庫關聯圖 18](#_Toc197783609)

表目錄

[表：3-2‑1系統軟、硬體需求 9](#_Toc197783614)

[表：3-3‑1開發工具 9](#_Toc197783615)

[表：4-2‑1專案組織與分工表 11](#_Toc197783616)

[表：5-1‑1功能性需求 14](#_Toc197783617)

# 前言

## 背景介紹

近年來的資訊科技發展蓬勃，各式行業都紛紛開始與資訊科技做出結合。但其中所使用的許多資料的收集和儲存仍是以紙本保存，紙本的媒介要轉換成電子資訊的過程則要花費大量人力和時間來處理。在這背景之下E筆勾銷便是為了解決這點而產生的，為了減少花費在轉換儲存媒介的時間的繁複的手續

## 動機

在這資訊化的社會中，許多產業開始積極地開始資訊轉型。而會計仍然是在這過程中較為傳統的一環，我們認為其中有著可以改善的空間，於是我們選擇了發票、傳票等利用紙本儲存的會計憑證，利用RPA進行將其改善並結合生成式AI進行效益的提升。於是E筆勾銷便由此而生，為了改善轉換儲存媒介所需時間的繁複的手續和花費的人力成本。

## 系統目的與目標

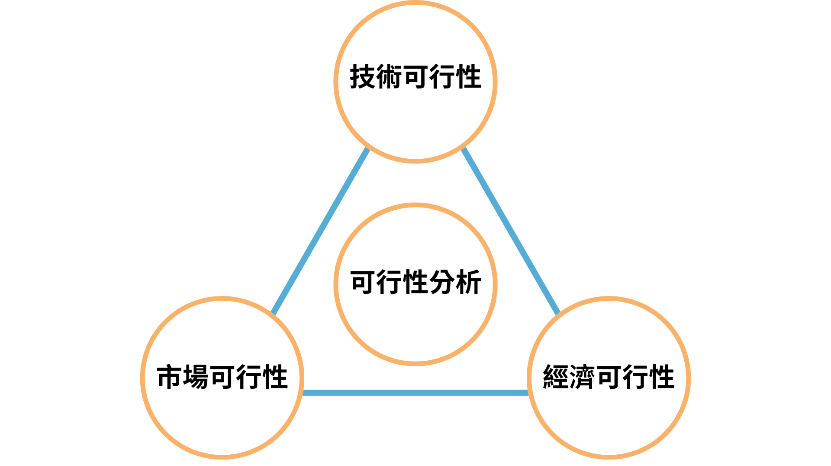
* 改善目前發票傳換至會計系統的流程
* 創造會計流程自動化

## 預期成果

* 改善目前會計流程並將其自動化

# 營運計畫

## 可行性分析



圖：2-1‑1系統可行性分析

### 技術可行性

E筆勾銷需要運用多種技術和知識來實現開發，目前具備的技術和知識如：會計流程自動化的設計、了解系統分析的概念，了解該如何分析系統架構、熟知fimga並能設計出人性化的介面， OCR 技術與 AI 辨識，還須支援多種發票類型，包括電子發票、傳統紙本發票、收據等。最後還須利用RAG和SNN的技術搭配生成式AI用以確保發票辨識的準確性與完整性。

### 市場可行性

基本上任何組織只要涉及金錢都會需要會計和憑證管理人員，而其工作內容去卻大多為重複且容錯率低的文書工作。若有一個可以輕鬆管理和解決這些繁雜工作的網頁那會有多好，E筆勾銷為了使用者會計需求能夠滿足而設計了下列功能: 響應式網頁(RWD)適用於各式平台和各種遠端設備使用、使用者可透過手機拍照或檔案上傳等方式上傳發票、並可查詢報表，隨時掌握會計數據。

### 經**濟可行**性

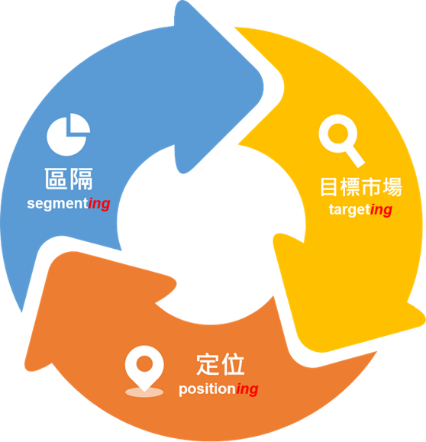
E筆勾銷可以在為使用者節省需要花在傳統會計流程中發票報帳或審閱上所需花費的大量時間和沉重的人事成本，透過生成式AI的幫助下E筆勾銷還可以為發票或憑證中有的錯誤進行自動修正。如:錯別字或是會計項目的錯誤。E筆勾銷可以為使用者節省下大量的時間和更多的人事成本，更還能防範未來可能發生錯誤所造成的損失。

## 商業模式－Business model



圖：2-2‑1商業模式

## 市場分析－STP



圖：2-3‑1市場分析STP

### 市場區隔(Segmentation)

* E筆勾銷鎖定企業財務部門、中小企業主、會計師事務所以及對發票管理有需求的個人或組織。

### 目標市場(Targeting)

* E筆勾銷的目標市場是以那些尚未完全電子化會計流程仍依賴紙本發票處理且有一定會計需要的人或組織。

### 產品定位(Positioning)

* 讓企業能夠輕鬆管理發票、簡化會計流程、將其RPA化的財務控管轉型。
* 為使用者提供更簡單、更方便的發票管理工具，讓使用者可以用更低的成本和更少的時間。

|  |  |
| --- | --- |
| **Strengths 優勢**  **跨平台網頁開發無APP上架費用：**節省支出  **發票自動辨識：**無須手動輸入  **操作簡易：**使企業導入使用容易  **兼容多個發票種類：**含傳統、三聯式發票  **無資料隱私洩漏問題：**AI語言模型皆運行於本地  **平台基本功能免費：**免費即可享有大部分功能 | **Weaknesses 劣勢**  **依賴網際網路：**網頁需運行在有網路的環境  **依賴GPU運算：**部分功能無GPU運算時，效能將不如預期  **辨識之影像需清楚：**若上傳之影像失真，將降低辨識成功率  **設備剛需：**若無能攝影之裝置，部分功能無法使用 |
| **Opportunities 機會**  **AI技術成熟：**能實現預期的流程  **公司行號皆有報帳需求：**適用於各公司行號  **競爭對手皆須付費：**市面上相同的服務，大部分都須付費且價格不透明 | **Threats 威脅**  **使用客群不滿意：**平台提供之服務，使用客群不滿意  **其他大型競爭者出現：**大型競爭者推出相似之服務或產品，將威脅本平台利益  **未來發票類型轉換：**發票不再使用紙本，皆轉換成電子資訊，本平台的辨識功能將被淘汰 |

### 競爭力分析－SWOT

### TOWS策略

**SO 策略**

1. 以免費+跨平台特性吸引用戶：利用平台基本功能免費、跨平台無上架費用的優勢，吸引有報帳需求、預算有限的中小企業或新創公司。
2. 推廣在地AI保障隱私：強調本地運行無隱私洩漏風險，搭配AI技術成熟的契機，建立用戶對資訊安全的信心。
3. 利用操作簡易打入報帳市場：結合企業普遍有報帳需求與平台操作簡便的優勢，快速導入企業內部流程。

**ST 策略**

1. 持續優化辨識準確率：因應未來發票轉型風險，透過支援多種發票格式與本地AI模型更新能力，提升未來適應性。
2. 強調免費與隱私保護對抗競爭者：面對大型競爭者可能加入市場，主打平台免費與資料不經雲端的優勢差異化。
3. 建構使用者回饋機制：因應使用者不滿意風險，透過簡易操作介面與客製化設定，快速回應使用需求並迭代改進。

**WO 策略**

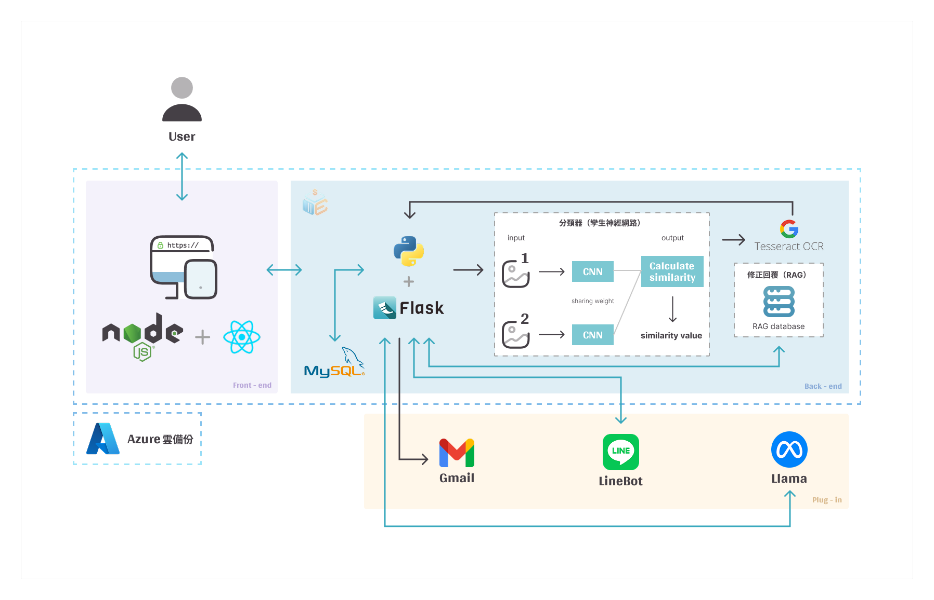
1. 導入離線模式/緩存技術：減少對網路依賴，提升在不穩網路環境下的可用性，抓住AI導入契機。
2. 尋求雲端GPU合作方案：針對缺乏GPU資源的使用者，考慮提供雲端計算選項，讓更多企業得以使用。
3. 優化影像前處理功能：利用AI技術發展，加入自動影像強化工具，提高模糊圖片辨識成功率。

**WT 策略**

1. 多元化輸入方式：因應設備限制與影像辨識問題，增加語音輸入等替代方式。
2. 持續監測市場發票轉型趨勢：提早佈局電子發票資料串接功能，降低紙本辨識功能被淘汰的風險。
3. 強化教育與支援資源：提供完善的系統操作手冊，並協助企業進行員工訓練（例如：使用流程講座）。

# 系統規格

## 系統架構

****

圖：3-1‑1系統架構圖

本系統功能主要分為一般使用者和管理者、使用說明如下：

1. 一般使用者
   * 透過使用者註冊將帳號和密碼提供給E筆勾銷後台後，登記進後台的資料庫後，使用者即可登入E筆勾銷
   * 使用者登入E筆勾銷照護者登入後台後，首頁會有報帳紀錄表，報帳申請查詢、新增報帳和相機掃描等功能。導覽列則會有帳戶資訊、核銷紀錄、FQA智能回覆、其他設定和介面色調的選項
   * 點選新增報帳即可上傳發票檔案至E筆勾銷，發票會經由網站進入分類器分辨是否為發票，是發票的話則會進入OCR開始辨識，不是則進入到其他的分類中。如：明細、傳票。經過RAG修正後則會呈現辨識結果提供到待報銷紀錄中。
   * 點選相機掃描則能透過鏡頭直接將發票上傳至E筆勾銷，發票會經由網站進入分類器分辨是否為發票，是發票的話則會進入OCR開始辨識，不是則進入到其他的分類中。如：明細、傳票。經過RAG修正後則會呈現辨識結果提供到待報銷紀錄中。
   * 選擇FQA智能回覆後，即可開始輸入想了解的問題後。機器人會透過生成式AI生成答案並透過RAG修正後回答給使用者。
2. 管理者

* 管理者登入E筆勾銷照護者登入後台後，首頁會有支出占比，預算上限表、近期審核及AI支出分析建議畫面。導覽列則會有帳戶資訊、核銷報帳、報表專區、其他設定和介面色調的選項
  + 點選核銷報帳即可查閱發票的詳細資訊如：標題、報帳日期、發票種類、金額、申請人等。報帳附件則會下載申請的發票，發票確認完後則能決定是否核銷。
  + 點選報表專區則能看到核銷發票的列表和透過發票資訊所整理出的各項資訊所繪製的圖表。如：前三支出種類、核銷金額變動圖、核銷金額分配圖等。最後則能看到各月份對於發票所整理出的事件簿。

## 系統軟、硬體需求與技術平台

表：3-2‑1系統軟、硬體需求

|  |  |
| --- | --- |
| 基本軟體需求 | |
| 作業系統 | Android 7.0、iOS 11.3、Windows 10、macOS 10.13以上 |
| 基本硬體需求 | |
| 基本設備 | Wi-Fi / 網際網路 |
| 可拍攝的鏡頭 |

## 開發標準與使用工具

表：3-3‑1開發工具

|  |  |  |
| --- | --- | --- |
| 系統開發環境 | | |
| 作業系統 | Windows 7支援即將於明天到期，又有證據還可以免費升級| iThomemacOS Tipps: Privatsphäre-Einstellungen in macOS vornehmen | Windows、macOS |
| 撰寫工具 | PyCharm - 維基百科，自由的百科全書 歡迎使用Colab - Colab | Visual Studio Code、MySQL、Pycharm、Google Colab |
| 程式開發語言 | | |
| 前端 | 如何在Ubuntu 11.04 上設定node.js 的開發環境| DreamersLab | Node.js |
| 後端 | Python - 維基百科，自由的百科全書 | Python |
| 資料庫 |  | MySQL |
| 網頁介面設計工具 | | |
| 介面 |  | Figma |
| 文件及簡報工具 | | |
| 文件 |  | Microsoft Word |
| 簡報 |  | Microsoft PowerPoint、Canva |
| 專案管理及版本控制工具 | | |
| 應用程式 |  | Fork |
| 版本控制 | GitHub - YouTube | Git、Github |
| 專案管理 |  | Notion |

# 專案時程與組織分工

## 專案時程



|  |  |  |  |
| --- | --- | --- | --- |
| 預想進度 |  | 實際進度 |  |

圖：4-1‑1甘特圖

## 專案組織與分工

表：4-2‑1專案組織與分工表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 11146023  翁益德 | 1114C007  周品君 | 11146020  鄭名棋 | 11146031  葉勇杰 |
| 資料庫 | 資料表規劃 | ● | ○ | ○ |  |
| 伺服器 | 虛擬機架設 |  |  |  |  |
| DNS、HTTPS |  |  |  |  |
| 開發環境架設 | Flask |  |  |  |  |
| Node.js |  | ● |  |  |
| Docker |  |  | ● |  |
| MySQL |  |  |  |  |
| LOGO | LOGO設計 |  | ● |  |  |
| 前台前端畫面 | UI/UX與介面設計 |  | ● |  |  |
| 註冊及登入 |  | ● |  |  |
| 核銷管理 |  | ● |  |  |
| 過去核銷記錄 |  | ● |  |  |
| 服務操作提示框 |  | ● |  |  |
| 其他設定畫面 |  | ● |  |  |
| API串接 |  | ● |  |  |
| 後台管理核銷 | UI/UX與介面設計 |  | ● |  |  |
| 註冊及登入 |  | ● |  |  |
| 服務操作提示框 |  | ● |  |  |
| 主頁資訊及儀表板 |  | ● |  |  |
| 報帳申請詳細資訊 |  | ● |  |  |
| 編輯及新增預算 |  | ● |  |  |
| 核銷報帳 |  | ● |  |  |
| 報表專區 |  | ● |  |  |
| 核銷圖表 |  | ● |  |  |
| 事件簿 |  | ● |  |  |
| 其他設定畫面 |  | ● |  |  |
| API串接 |  | ● |  |  |
| 後端功能開發 | 使用者管理 |  |  |  |  |
| 核銷邏輯開發 |  |  |  |  |
| Line Bot互動設計 |  |  |  |  |
|  | ChatGPT API互動設定 |  |  |  |  |
|  |  |  |  |  |  |
| 影像辨識 | OCR設定 |  |  |  |  |
| 文件撰寫 | 統整 |  |  |  | ● |
| 第1章 前言 |  |  |  | ● |
| 第2章營運計畫 |  | ○ |  | ● |
| 第3章系統架構 |  | ○ |  | ● |
| 第4章專案時程與組織分工 |  | ○ |  | ● |
| 第5章需求模型 |  | ○ |  | ● |
| 第7章實作模型 |  |  |  | ● |
| 第8章資料庫設計 |  | ○ |  | ● |
| 競賽 | 報名資料整理 |  |  |  | ● |
| 報告 | 簡報製作 |  |  |  | ● |
| 競賽 | 文件製作 |  |  |  | ● |
| 影片製作 |  |  |  | ● |

## Github上傳紀錄

# 需求模型

## 需求清單

表：5-1‑1功能性需求

|  |  |
| --- | --- |
| 功能名稱 | 功能敘述 |
| 1. 帳戶 | * 使用Gmail 註冊與登入 |
| 1. 設定 | * 調整網頁介面色調 * 常用會計項目查詢 * 調整個人資料 |
| 1. 報帳申請 | * 發票上傳 * 發票掃描資訊展示 * 顯示和通知報帳結果 |
| 1. 報帳紀錄查詢 | * 發票核銷結果展示 |
| 1. 核銷管理 | * 發票核銷許可 * 詳細發票資訊查詢 * 發票核銷事件簿 |
| 1. 預算管理 | * 預算上限設定 * 支出分析圖表 * 支出上限警示 |
| 1. FQA管理 | * 常見問答回覆 * 生成式AI回覆非常見問答 |

非功能性需求

1. 模組化程式碼
2. 穩定運行，不崩潰或出現錯誤
3. 在時間內辨識出結果
4. 設計方便好用的用戶介面、說明和指引
5. 能夠容納更多的用戶和數據

## 使用個案圖(Use case diagram)

## 使用個案描述

## 分析類別圖(Analysis class diagram)

# 設計模型

## 循序圖(Sequential diagram)

## 設計類別圖(Design class diagram)

# 實作模型

## 佈署圖(Deployment diagram)

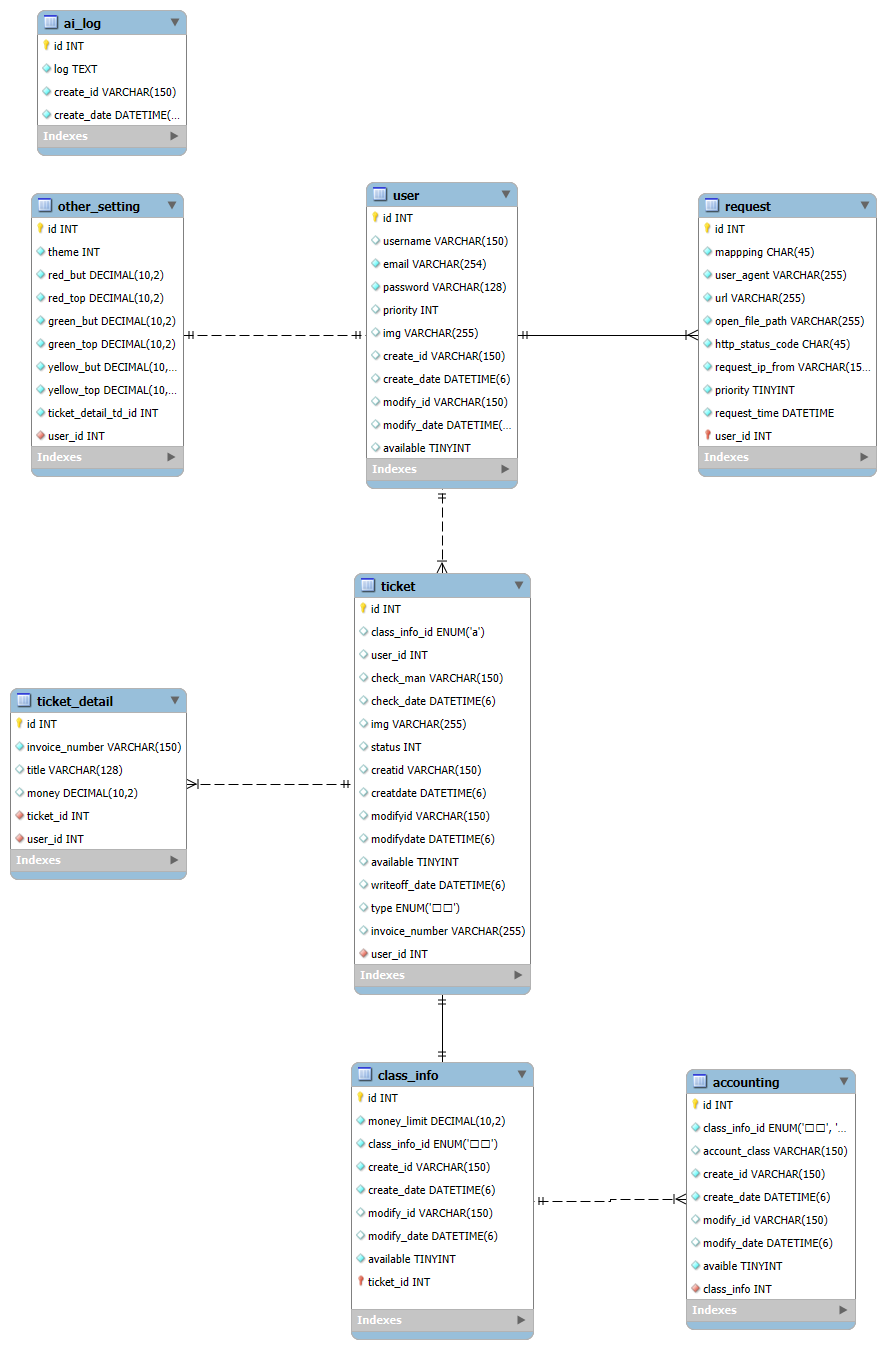
## 套件圖(Package diagram)

## 元件圖(Component diagram)

## 狀態機(State machine)

# 資料庫設計

## 資料庫關聯表



圖：8-1‑1資料庫關聯圖

## 表格及Mata data