|  |
| --- |
| 113學年度程式設計期末專題  **National Kaohsiung University of Science and technology**AI智慧小幫手 食材管理沒煩 |
| 指導老師：謝文川 |
| 成員：C112156202蕭宛宜  　　　C112156203黃玲煒  　　　C112156204李閔珊  　　　C112156209程惠明  中華民國114年1月 |

[零、摘要 3](#_Toc187104257)

[壹、緒論 3](#_Toc187104258)

[一、前言 3](#_Toc187104259)

[二、研究目的 3](#_Toc187104260)

[三、研究方法 3](#_Toc187104261)

[四、研究目標 4](#_Toc187104262)

[貳、文獻探討 4](#_Toc187104263)

[一、智能提醒系統的現狀與應用 4](#_Toc187104264)

[二、LINE Bot與人工智慧的整合 4](#_Toc187104265)

[三、食品管理系統的技術挑戰 4](#_Toc187104266)

[參、系統架構 4](#_Toc187104267)

[一、架構概述 5](#_Toc187104268)

[二、實作方法 5](#_Toc187104269)

[肆、實作畫面 6](#_Toc187104270)

[伍、遇到困難 6](#_Toc187104271)

[陸、未來期許 6](#_Toc187104272)

零、摘要

本專題旨在設計並實現一套基於LINE Bot的智能提醒與資源管理系統，幫助用戶高效管理食材資訊並減少浪費。系統利用Flask作為後端框架，結合SQLite資料庫存儲用戶數據，並使用Google Generative AI技術生成個性化食譜建議。該系統提供如新增、查詢、刪除、修改食材等功能，並可自動提醒即將過期的食材資訊。透過雲端與本地資料結合的架構設計，系統實現高效性、可擴展性及用戶友好的操作介面。

壹、緒論

一、前言

食品管理問題是現代家庭常見的挑戰，特別是在高頻率採購的情境下，食材容易過期浪費。此外，隨著消費者對健康與飲食多樣性的需求增長，尋找合適的食譜也變得日益重要。本研究結合LINE Bot的即時互動特性與人工智慧技術，設計一個簡單易用的解決方案，幫助用戶方便地管理食材並獲得飲食靈感。

二、研究目的

本專題旨在達成以下目標：

1. 提供便捷的食材管理工具，幫助用戶追蹤食材保存期限。
2. 設計自動化的提醒功能，避免食材浪費。
3. 利用人工智慧技術生成個性化食譜，提高食材利用率。
4. 探討本地化數據存儲與雲端資料整合的最佳實踐。

三、研究方法

1. 需求分析：調查用戶的日常食材管理痛點，確定功能需求。
2. 技術選型：基於Python開發環境，選擇Flask框架搭建後端，結合LINE Messaging API與SQLite資料庫。
3. 系統設計：以模組化設計為核心，確保系統的可維護性與可擴展性。
4. 測試與優化：進行單元測試、整合測試與使用者體驗調查，持續優化功能。

四、研究目標

1. 建立一個高效、友好的食材管理與提醒系統。
2. 整合Google Generative AI進行食譜建議生成，提供智能化增值服務。
3. 研究本地化資料存儲與雲端計算在小型應用中的應用場景。

貳、文獻探討

一、智能提醒系統的現狀與應用

智能提醒系統已被廣泛應用於多個領域，如醫療健康（用藥提醒）、教育（學習計劃管理）與日常生活（事項提醒）。然而，多數系統過於依賴硬體（如IoT設備）或特定平台（如專屬App），無法滿足用戶的跨平台需求。本專題選用LINE Bot作為交互界面，結合Python進行系統開發，降低硬體與軟體依賴。

二、LINE Bot與人工智慧的整合

LINE Bot因其普及性與即時互動性，成為設計智能應用的理想工具。當結合Google Generative AI等人工智慧技術後，Bot能更深入了解用戶需求，生成符合情境的文字建議或內容。過去研究表明，這種結合不僅提升用戶體驗，還可大幅提高系統智能化水平。

三、食品管理系統的技術挑戰

食品管理系統的主要挑戰包括數據準確性、交互便捷性與提醒及時性。通過引入SQLite進行數據管理，並設計排程提醒功能，本研究有效解決了這些挑戰。此外，人工智慧技術的融入提升了系統的實用性與創新性。

參、系統架構

一、架構概述

1. 用戶端：使用LINE Bot作為用戶與系統的交互界面，支援文本指令如「新增」、「查詢」、「刪除」、「修改」、「食譜」等，實現高便捷性互動。
2. 伺服器端：
   * 基於Flask框架搭建，負責處理用戶請求、數據交互與邏輯運算。
   * 整合Google Generative AI進行語意生成與智能化食譜建議。
3. 資料存儲：
   * 使用SQLite本地資料庫存儲用戶與食材數據，支持快速查詢與更新。
   * 系統設計具備擴展性，可透過API整合雲端資料庫進行備份與同步。
4. 排程與提醒模組：
   * 使用Python的schedule模組實現定時任務，定期檢查資料庫中即將過期的食材。
   * 利用LINE Bot API實時推送提醒訊息至用戶端。
5. 安全性設計：
   * 環境變數（如API金鑰與資料庫位置）採用.env文件管理，確保敏感資訊的安全性。
   * 系統具備錯誤處理機制，記錄日誌並提供友好的錯誤回饋。

二、實作方法

系統開發步驟

1. 環境設置：
   * 使用Python開發環境，安裝必要的套件如Flask、line-bot-sdk與sqlite3。
   * 設置.env文件存儲API金鑰與LINE相關憑證。
2. 後端功能開發：
   * 使用Flask框架處理用戶請求，設計RESTful API實現與LINE Bot的互動。
   * 建立SQLite資料庫進行食材和用戶數據的存儲與管理。
3. 功能實現：
   * 新增功能：允許用戶新增食材名稱與有效日期，並進行格式驗證。
   * 查詢功能：以列表形式顯示當前所有食材及其有效日期。
   * 刪除功能：根據用戶輸入，移除指定的食材記錄。
   * 修改功能：支持修改食材名稱與有效日期，確保數據更新準確。
   * 食譜推薦：調用Google Generative AI，基於用戶食材輸入生成多樣化的菜單建議。
4. 排程提醒：
   * 設置定時任務，檢查資料庫內即將過期的食材，並透過LINE Bot推送提醒訊息。
5. 測試與優化：
   * 進行單元測試（Unit Test）驗證功能正確性。
   * 收集用戶反饋並迭代改進，提升用戶體驗。

肆、實作畫面

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

伍、遇到困難

陸、未來期許