視窗程式設計

期末報告

題目

食物相剋造成原因及解決方法

組員：陳蔡安　B11170015

　　　黃紫綺　B11170067

指導老師：江家頡

目錄

[第一章 緒論 3](#_Toc200741409)

[研究背景與動機 3](#_Toc200741410)

[研究目的 3](#_Toc200741411)

[問題陳述 3](#_Toc200741412)

[預期貢獻 3](#_Toc200741413)

[第二章 系統設計 4](#_Toc200741414)

[系統架構圖 4](#_Toc200741415)

[功能模組說明 4](#_Toc200741416)

[循序圖、類別圖、使用案例圖等UML圖表 5](#_Toc200741417)

[第三章 系統實現 7](#_Toc200741418)

[開發環境 7](#_Toc200741419)

[核心功能代碼解析 7](#_Toc200741420)

[關鍵技術實現說明 17](#_Toc200741421)

[界面設計與使用者操作流程 20](#_Toc200741422)

[第四章 結論與未來工作 27](#_Toc200741423)

[研究成果總結 27](#_Toc200741424)

[專題限制 27](#_Toc200741425)

[未來改進方向 27](#_Toc200741426)

[附錄 參考文獻 28](#_Toc200741427)

第一章 緒論

研究背景與動機

擁有豐富選擇、多元飲食料理的台灣，各式各樣的搭配無奇不有，但食物之間是有可能發生衝突的，為了避免讓身體感受到不適還需要去尋找解決辦法，所以我們做出一個系統能夠讓使用者查詢食物之間是否衝突並賦予解決辦法，甚至記錄自己的飲食狀況，以避免食物中毒。

研究目的

本系統要探討食物之間有哪些容易造成身體不適，甚至是若誤食該如何應對。

本系統目標達成的結果如下：

1. 支援登入系統及切換使用者
2. 列出常見食物按鈕清單，可搜尋關鍵字，可新增提供資料
3. 提供使用者紀錄自己在何時吃了什麼
4. 給予記錄表單查詢過去吃了什麼
5. 食物搭配

問題陳述

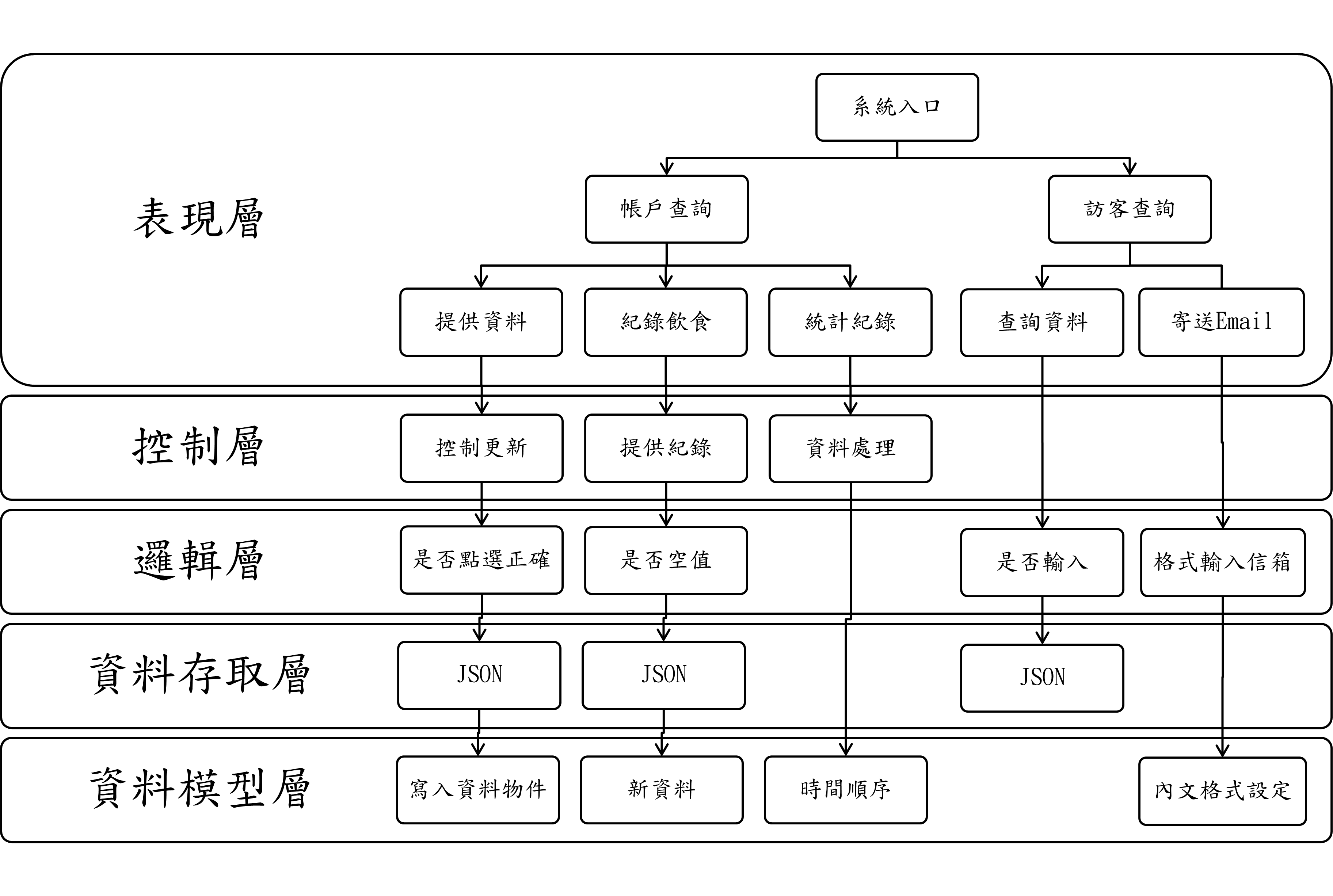
多數人對於食材衝突沒有概念，像是在吃到飽、自助餐尤為常見，有時是食材處理不當、有時是食材屬性不合，引發身體不適。然而市面上大多應用程式是用來計算熱量，雖然網絡搜尋有許多數據及資料，但步驟繁瑣、說法不一致，因此我們設計一個能夠直覺的查詢食物衝突狀況與紀錄飲食情況的系統，協助使用者關注自身健康，也減少突發腸胃不適的情形。

預期貢獻

本系統預期可建立一套簡易且實用的視窗應用程式平台，協助使用者透過內建食材衝突比對，提供飲食發生衝突的警示與解決方法，並且整合許多常見食物衝突資料，以便使用者即時了解資訊，更可以進行日常飲食記錄。在未來可能也能夠加裝進手機，讓使用者能夠更即時方便的紀錄與查詢。

第二章 系統設計

系統架構圖



功能模組說明

(介面一)系統入口：有訪客及登入兩種模式，可建立新帳戶。

(介面二)相剋查詢－訪客模式：使用者僅可查閱食物相剋資料。

(介面三)相剋查詢－帳戶模式：使用者可查詢使用，並擁有更多樣的權限。

(介面四)提供資料：使用者可將得知的新資料輸入至系統，也可刪除錯誤資料。

(介面五)紀錄飲食：使用者可記錄個人飲食，且無需特別設定時間。

(介面六)飲食統計：使用者可點閱自己的紀錄，得知紀錄的頻率、常吃的食物。

循序圖、類別圖、使用案例圖等UML圖表

一張含有 文字, 筆跡, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 圖表, 平行, 方案 的圖片

AI 產生的內容可能不正確。一張含有 圖表, 行, 文字, 圓形 的圖片

AI 產生的內容可能不正確。

第三章 系統實現

開發環境

* 輸出類型：Windows應用程式
* 目標Framework：.NET 8.0
* 目標作業系統：Windows
* 目標OS版本：7.0
* 支援作業系統版本：7.0
* UI框架選擇：啟用Windows Forms

核心功能代碼解析

* Form1.cs

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

AI 產生的內容可能不正確。

統一定義三個 JSON 檔案的完整路徑，運用Application.StartupPath 可取得執行檔所在資料夾，並利用 Path.Combine 避免手動組合字串造成錯誤，確保一致性，也方便在整個專案中存取需要的資料。

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

AI 產生的內容可能不正確。

儲存整體系統共用資料，讓所有表單能直接讀取與修改資料庫(字典)資訊，使用static 讓資料存在於應用程式生命週期中，不會因表單關閉而消失，並且能夠自動更新。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。

集中處理使用者在主畫面輸入帳號名稱後的三種特例情況，防止輸入重複帳號、啟動訪客模式（GuestForm）、顯示帳號選單供選擇（Choose 模式）。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

當使用者輸入尚未存在的帳號名稱時，自動將此帳號新增至帳號清單確保儲存資料可讀性與安全字元處理後，儲存至 JSON 檔，並同時更新介面提供用戶後續操作。

* Accountform.cs

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

當使用者點擊某個食物按鈕時，確保觸發事件的是合法的按鈕，並且該按鈕有對應的原始文字（食物名稱），將該食物的詳細資料（優點、相剋、後果、解決辦法）顯示在文字區域 txtList 中，讓使用者可以快速查閱食物的完整資訊。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

當使用者將滑鼠游標移到食物按鈕上時，確認事件來源是已知的按鈕，避免非預期物件觸發，顯示對應食物的圖片並隱藏文字，提升視覺效果與使用者體驗。

* EmailForm.cs

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

讓使用者能將查詢到的食物資訊或警告信件，透過 Gmail 自動發送給指定收件人。此功能可應用於提醒、資料分享等場景。

string fromEmail = "FoodIncompatibilityManager@gmail.com" 寄件人帳號。建議更換為使用者自己的 Gmail 並開啟應用程式密碼功能。

string appPassword = "kxckpxmlkdkgavye" Gmail 專用應用程式密碼（非登入密碼）。需到 Google 帳戶設定產生。

new SmtpClient("smtp.gmail.com", 587) 建立 Gmail SMTP 客戶端。587 是 TLS 預設連接埠。

EnableSsl = true 啟用 SSL 加密以確保傳輸安全。

Credentials = new NetworkCredential(...) 提供寄件人帳號與密碼作為登入驗證。

MailMessage 建立 Email 主體。設定寄件人、主旨、內文。

IsBodyHtml = true 允許 HTML 格式排版。

Body = $"<html><body><pre>{body}</pre></body></html>" 使用 <pre> 標籤保持格式與換行。

mail.To.Add(toEmail) 加入收件者電子郵件地址。

mail.AlternateViews.Add(...) 同時附加純文字版本，以便某些 Email 客戶端無法渲染 HTML。

smtpClient.Send(mail) 實際執行發送動作。送出郵件給伺服器。

* GuestForm.cs

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

負責初始化訪客帳號介面，其核心目的是：

指定帳號為「訪客」時，讓系統進入受限功能的模式，繼承共用功能（如按鈕生成、資料讀取）但客製化顯示與行為。使用 Load 事件綁定方式避免覆寫父類載入方法，確保其他子類不受影響。

base("訪客") 呼叫父類 AccountForm 的建構子，並傳入帳號名稱「訪客」，做為辨識身份的依據

this.Load += GuestForm\_Load 掛載事件處理器而不直接覆寫父類 AccountForm\_Load，避免破壞父類通用邏輯

base.AccountForm\_Load(...) 明確呼叫父類的 UI 初始化邏輯（例如載入食物資料與按鈕）

labelAccount.Text = ... 修改畫面上帳號身份標籤，顯示為「訪客（僅能查詢）」以提醒使用者權限受限

* ProvideForm.cs

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

在 ProvideForm 載入時，初始化畫面，並從 food.json 讀取所有現有食物資料，將其名稱載入到 comboBoxDelete 下拉選單中，供使用者選擇欲刪除的食物項目。

一張含有 文字, 螢幕擷取畫面, 字型, 文件 的圖片

AI 產生的內容可能不正確。

用來新增或修改一筆食物資料，將使用者輸入的食物名稱、優點、相剋、後果與解決辦法，儲存到 food.json，並更新主系統中的共用資料字典與介面（像是刪除下拉選單、食物按鈕區）。

設定不可有空白名稱，並用逗號作為分隔，載入現有資料或新建字典，若 JSON 檔存在則讀取，否則新建空字典。

* FoodLogForm.cs

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

在使用者輸入食物與餐別後，儲存該筆飲食紀錄至 records.json，僅保留最近14天的紀錄，並更新當前畫面顯示的清單與日期選單，讓資料即時反映。

一張含有 文字, 螢幕擷取畫面, 字型, 文件 的圖片

AI 產生的內容可能不正確。

將目前帳號的「上週」與「本週」飲食紀錄匯出成 CSV 格式，並儲存在桌面上，方便用戶查看或備份飲食紀錄。

* StatisticsForm.cs

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

從 records.json 載入指定帳號的飲食紀錄，並統計各種餐別（如早餐、午餐等）出現的次數，顯示於 listBoxMeal 中。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

統計使用者在飲食紀錄中的行為特徵，包含：常出現的食物（LoadFoodStatistics）：分析最常食用的項目，有助於健康建議與飲食追蹤。一週內各天的紀錄次數（LoadWeekdayStatistics）：檢視使用者填寫頻率與習慣，判斷是否穩定紀錄。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

檢查並列出上週與本週共 14 天中，使用者尚未填寫的飲食紀錄日期，幫助使用者追蹤填寫習慣，強化紀錄完整性。

records.ContainsKey(accountName) 確認帳號有紀錄可查，否則提示錯誤

.Select(r => r.日期).Distinct().ToHashSet() 建立已填寫日期的集合，快速比對

DateTime today = DateTime.Today 取得今天日期，作為參考基準

thisSunday = today.AddDays(-todayDay) 計算本週週日日期

lastSunday = thisSunday.AddDays(-7) 計算上週週日（即 14 天前的起點）

for (int i = 0; i < 14; i++) 循環過去 14 天的日期

if (!recordedDates.Contains(dateStr)) 若該天未填寫，則加入缺漏列表

"日一二三四五六"[(int)date.DayOfWeek] 根據 DayOfWeek 顯示對應中文星期

listBoxEmptyDays.Items.Add(...) 顯示缺漏日期及星期資訊於介面

關鍵技術實現說明

AccountForm.cs

建構子:public AccountForm(string account) 指定帳號名稱，設定事件；protected AccountForm() 是預設訪客帳號。

多表單應用程式:使用 new ProvideForm().Show()、new FoodLogForm(accountName).Show()、new EmailForm(...).ShowDialog()。

事件處理:各種按鈕事件：FoodButton\_Click, buttonSearch\_Click, btnsignout\_Click 等皆透過委派連接與完整實作。

委派:所有事件處理如 btn.Click += FoodButton\_Click 等，都是委派形式的事件繫結。

例外處理:雖然此檔案未直接使用 try-catch，但在 FoodButton\_MouseEnter 有用 FileStream 開啟圖檔並間接處理失敗路徑情況。

抽象化設計（預留多型）:多個方法標記為 protected virtual（例如查詢、登出、載入、點擊），可被子類別 GuestForm 繼承與覆寫。

表單滑鼠控制事件:滑鼠移入移出事件：MouseEnter, MouseLeave 改變圖片與按鈕文字。

表單鍵盤事件:textBox1\_KeyDown 處理 Enter 鍵查詢食物資料。

動態控制項生成:AddFoodButtonAtFront(string foodName) 方法中動態建立 Button 並加到

tableLayoutPanelFood。

檔案與資料夾處理:使用 File.Exists, FileStream, Bitmap 載入圖片做為快取圖片來源。

Form1.cs

建構子:Form1() 中設定 FormBorderStyle、關閉最大化、綁定 Timer 事件並呼叫 InitializeComponent()。

部分類別:public partial class Form1 : Form，實際由設計器 .Designer.cs 分離介面程式碼。

多表單應用程式:使用 new GuestForm().Show() 與 new AccountForm(selected).Show() 完整開啟並切換不同視窗。

事件處理:事件如 Form1\_Load, btnConfirm\_Click, btnLogin\_Click, timerScroll\_Tick 等明確註冊並完整實作。

委派:timerScroll.Tick += timerScroll\_Tick 等使用 Windows Forms 中事件的委派機制完成 UI 操作綁定。

例外處理多個 try-catch 包裝 JSON 讀取動作，若出錯則顯示 MessageBox，避免程式崩潰並提示錯誤內容。

檔案與資料夾處理:使用 File.Exists, File.ReadAllText, File.WriteAllText, Path.Combine 處理帳號與紀錄存取。

靜態類別成員:public static Dictionary<string, FoodData> foodInfo 與 records 用於多表單共用資料並在初始化時讀取。

JSON 處理與序列化:使用 JsonSerializer.Serialize/Deserialize 處理帳號、食物與紀錄資料的儲存與還原，有實作存檔與更新邏輯。

GuestForm.cs

繼承:GuestForm : AccountForm 表示訪客版是一般帳號的特化版

實作多型:使用 override 覆寫父類別的 buttonProvide\_Click、buttonEat\_Click 方法

事件處理:自行綁定 GuestForm\_Load，並手動呼叫 base.AccountForm\_Load 來沿用父類邏輯

EmailForm.cs

建構子:public EmailForm(string foodName) 接收食物名稱，設定視窗標題與初始狀態。

事件處理:使用 btnSend\_Click, btnClear\_Click, btnBack\_Click 等事件對應按鈕操作。

例外處理:使用 try-catch 處理寄信過程中可能發生的錯誤（如 SMTP 錯誤、一般例外）。

檔案與資料夾處理:從 Form1.foodInfo 讀取資料，是先前由 JSON 載入的靜態資料結構。

靜態資料共用:使用 Form1.foodInfo 來查詢食物資料。這是跨表單共用資料的設計。

ProvideForm.cs

建構子:public ProvideForm() 綁定載入與縮放事件。

事件處理:包含 ProvideForm\_Load, buttonSave\_Click, btnDeleteFood\_Click, buttonBack\_Click 等典型事件流程。

檔案與資料夾處理:使用 File.Exists, File.ReadAllText, File.WriteAllText, Path.Combine 處理 food.json 的新增/修改/刪除。

JSON 處理與序列化:使用 JsonSerializer.Serialize/Deserialize 完整處理食物資料格式，並格式化輸出。

靜態資料共用:使用 Form1.foodInfo 作為全域資料來源與更新對象（動態更新全系統資料）。

FoodLogForm.cs

建構子:public FoodLogForm(string account) 接收帳號名稱、綁定 Load 和 Resize 事件、初始化資料。

事件處理:共包含 13 個事件處理方法，如 btnSaveRecord\_Click, comboBoxDaySelector\_SelectedIndexChanged 等，功能完整。

例外處理:雖未顯式使用 try-catch，但所有檔案操作皆包含 File.Exists 判斷與資料驗證，間接防呆處理錯誤。

檔案與資料夾處理:使用 File.ReadAllText, File.WriteAllText, Path.Combine 處理 records.json 儲存與載入。

JSON 處理與序列化:使用 JsonSerializer.Serialize/Deserialize 進行紀錄儲存、清理與重新載入。

靜態資料共用:透過 Form1.records 全域靜態字典實現跨表單資料同步與修改。

FormResizer.cs

類別與物件導向設計:自訂類別 FormResizer，封裝 UI 縮放邏輯，透過成員方法供多個表單呼叫，強化程式模組化與可重用性。

事件處理與委派:使用 Timer.Tick += (s, e) => { ... } 建立匿名方法委派，實現縮放延遲執行。

StatisticsForm.cs

建構子:public StatisticsForm(string account) 初始化資料與事件綁定，並立即載入所有統計方法。

事件處理:共實作多個事件處理器：Load, Resize, btnBack\_Click, btnRefresh\_Click 等，涵蓋畫面初始化與使用者操作。

檔案與資料夾處理:使用 File.Exists, File.ReadAllText, Path.Combine 讀取 records.json，實作記錄讀取功能。

JSON 處理與序列化:使用 JsonSerializer.Deserialize<Dictionary<string, List<Form1.Record>>> 載入資料。

靜態資料共用:使用 records 靜態變數（來自 Form1），與其他表單共享紀錄資料。

界面設計與使用者操作流程

* Form1.cs

一張含有 文字, 食物, 原料, 綠葉蔬菜 的圖片

AI 產生的內容可能不正確。使用者輸入帳號名稱，若是Guest(不限大小寫)則跳轉到訪客選單。

若為Choose(不限大小寫)則顯示下方切換帳後供選擇，若accounts.json裡無輸入的名稱則會新增帳號，也能夠將以儲存帳號進行刪除。

一張含有 文字, 食物, 菜餚, 原料 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 食物, 原料, 綠葉蔬菜 的圖片

AI 產生的內容可能不正確。

* GuestForm.cs

一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

AI 產生的內容可能不正確。輸入為Guest時，將會跳轉到這個頁面，此頁面只能夠點選按鈕查詢食物相關資料、輸入食物名稱查詢以及將食物資料傳至Email，資料更新與飲食紀錄將被禁止。

* 一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

  AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

  AI 產生的內容可能不正確。AccountForm.cs

選擇帳後登入後將能夠使用所有訪客能使用的內容以及資料更新和飲食紀錄。一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

AI 產生的內容可能不正確。

* EmailForm.cs

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

AI 產生的內容可能不正確。寄送Email至信箱會顯示相關資料。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 標誌, 字型 的圖片

AI 產生的內容可能不正確。

* ProvideForm.cs

一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。資料更新頁面中能夠自行添加食物資料也能夠進行刪除。

* FoodLogForm.cs

一張含有 文字, 螢幕擷取畫面, 陳列, 數字 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 陳列, 圖表 的圖片

AI 產生的內容可能不正確。在飲食紀錄中能夠讓使用者記錄當天的飲食情況，有餐別選擇、食物名稱、食物備註，會記錄在旁邊視窗中供閱讀，也能夠查詢進14天的飲食紀錄，隨時能選擇後刪除，並且有統計與匯出功能。

一張含有 文字, 螢幕擷取畫面, 數字, 陳列 的圖片

AI 產生的內容可能不正確。

* StatisticsForm.cs

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

AI 產生的內容可能不正確。統計頁面能夠計算使用者的餐別、食物出現次數、星期出現次數以及未記錄日期。

一張含有 文字, 螢幕擷取畫面, 陳列, 圖表 的圖片

AI 產生的內容可能不正確。匯出功能則會匯出一個CSV檔供使用者保存

一張含有 文字, 螢幕擷取畫面, 字型, Rectangle 的圖片

AI 產生的內容可能不正確。

BuildCsvSection(...) 將週期紀錄格式化為 CSV 區段字串

Path.Combine(..., fileName) 建立匯出檔案的桌面路徑

File.WriteAllText(...) 將組裝好的 CSV 內容寫入指定路徑的檔案中

System.Diagnostics.Process.Start(...) 開啟匯出後的 CSV 檔案供即時檢視

MessageBox.Show(...) 彈出訊息框告知匯出成功與檔案名稱

第四章 結論與未來工作

研究成果總結

本研究圍繞健康飲食與食物相剋的相關知識，研發出一套擁有圖形化操作介面的《食物相剋系統》，並將帳號管理、資料查詢、新增以及紀錄等功能整合於其中。系統以 C# Windows Forms 作為開發平台，運用 JSON 格式進行資料的儲存與讀取，最終實現下列成果：

* 使用者能夠不登入即可使用的訪客模式，可分別查詢並保存寄送至個人電子郵件中。
* 可輸入與查詢各項食物資訊，包括其優點、相剋食物、可能產生的後果以及對應的解決方式。
* 系統亦設有建立個人帳號，登入系統，並可擁有多項功能如下。
* 飲食紀錄功能，依照每週與餐別，詳細記錄每日所攝取的食物，並將資料存入專屬帳戶中。
* 資料提供介面，為資料庫補充項目，讓使用者能夠查詢更多品項。
* 統計分析模組，呈現用戶的用餐次數、偏好食物類型、缺漏記錄之天數等多項指標，協助用戶掌握自身飲食狀況，甚至可匯出CVS檔案，進一步提升整體的彈性與擴展能力。

專題限制

* 資料來源有限：食物相剋資料主要由團隊自行查找與輸入，尚未連接正式營養資料庫，資料正確性仍有待驗證。
* 缺乏使用者驗證機制：目前的建立帳號未實作密碼保護機制，存在安全性風險。
* 多語系與平台限制：本系統僅支援中文介面與 Windows 平台，無法支援行動裝置或跨平台作業系統使用。
* 操作流程未最佳化：部份按鈕或流程尚未達到最直觀的使用邏輯，新手使用者可能需要花時間適應。
* 食物關聯判斷未自動化：目前需手動查詢是否有相剋關係，尚未實作自動比對功能。

未來改進方向

我們計畫近一步整合食物營養的相剋資料庫，透過連接公開的營養學與中醫學資料來源，自動導入完整且正確的食物資訊，提升系統的準確性與可靠度。為了強化健康管理功能，系統將加入自動相剋偵測機制，能依據使用者輸入的每日飲食紀錄，主動判斷是否存在食物相剋的情況，並及時提出警示，協助使用者避免潛在風險。

在帳號管理方面，將導入使用者驗證與多帳號機制，透過帳號密碼登入提升資料的隱私性與安全性，同時預留雲端同步功能的擴充可能，讓使用者可以在不同裝置間同步個人資料與飲食紀錄。為因應行動化趨勢，系統將考慮採用 Xamarin 或 MAUI 等跨平台開發工具，使應用程式能順利執行於 Android 與 iOS 裝置上，提升可用性與覆蓋範圍。

此外，系統將導入 AI 模型，根據使用者的飲食習慣與歷史紀錄進行分析，進一步預測可能的健康風險，並提供個人化的飲食建議與推薦菜單。最後，為提升使用體驗與資料的可讀性，我們將加強視覺化呈現能力，透過長條圖、圓餅圖等動態圖表展示飲食統計資訊，使使用者能更直觀地掌握自身的飲食狀況與健康趨勢。這些改進方向將有助於打造一套更智慧、個人化且實用的飲食管理系統。

附錄 參考文獻

[旗標]新觀念Microsoft Visual C# 程式設計範例教本 第六版

跑馬燈教學：<https://ftandsu.blogspot.com/2018/07/c.html>

[C#]如何使用 Gmail 免費發信：<https://blog.hungwin.com.tw/cs-gmail/>

[C#]常用控制項：<https://medium.com/@felicianote>

[C#]實作：<https://ithelp.ithome.com.tw/users/20163217/ironman/6816>

[C#]CSON相關知識：<https://so.csdn.net/so/search?spm=1001.2101.3001.4498&q=C%23&t=&u=>

如何使用JSON檔案：<https://vocus.cc/article/64d9d0d3fd89780001c01793>

選擇日期功能：<https://jerry5217.pixnet.net/blog/post/226747753>

匯出Excel：<https://vocus.cc/article/64c233d5fd897800012010ac>

提示視窗：<https://rmotex.blogspot.com/2016/10/c_14.html>

避免重複啟動：<https://blog.csdn.net/xggbest/article/details/104231935>