第一章緒論

隨著移動裝置的普及與應用軟體技術的進步，使用者對於即時數據查詢與分析的需求日益增加。在賭博遊戲百家樂中，透過歷史牌局數據進行分析與模擬，已成為提升下注策略與遊戲體驗的重要工具。然而，目前市面上多數相關軟體多半缺乏跨平台支援或資料整合功能，且使用介面不夠直觀。

本專案基於 .NET MAUI 跨平台框架，開發一款結合百家樂歷史資料搜尋、手動紀錄輸入、模擬與獲利分析等多功能的行動應用程式。系統利用 JSON 格式統一管理資料，並透過友善的用戶介面，協助使用者快速定位會員號碼相關牌局記錄，進一步進行數據統計與策略分析。此應用不僅提高資料維護的便利性，更藉由模擬功能提供玩家多元的下注參考。

本報告將介紹系統架構、核心功能實作，以及未來擴充方向，以期為百家樂數據分析應用領域帶來更完整且實用的解決方案。

第二章 系統設計

本章將說明系統的整體架構與各模組功能設計，包含用戶介面設計、資料儲存與管理機制，以及系統主要功能流程。

### **2.1 系統架構**

本系統採用 .NET Multi-platform App UI (MAUI) 技術開發，具備跨平台能力，支援 Android、iOS 及 Windows 等多種裝置。系統以 MVVM 架構概念設計，透過多頁面導航（NavigationPage）實現功能模組分離，提升維護性與擴充彈性。

### **2.2 用戶介面設計**

系統包含多個頁面：

* **登入頁（LoginPage）**：負責帳號密碼驗證，成功後導向主頁。採用簡潔輸入欄位與錯誤提示訊息，強調使用體驗。
* **主頁（MainPage）**：為系統核心頁面，顯示資料載入狀態，提供會員編號搜尋、搜尋結果呈現與多功能導航按鈕。
* **模擬頁（SimulationPage）**、\*\*手動輸入頁（InputPanel）**及**獲利分析頁（ProfitPage）\*\*等功能頁面，負責資料模擬、手動紀錄與獲利預測。

### **2.3 資料管理**

系統資料以 JSON 格式儲存，分為：

* **BaccaratHistory**：紀錄所有百家樂牌局數據，包含預設與用戶新增的模擬資料。
* **ManualHistory**：存放手動輸入的牌局資料，方便使用者記錄真實牌局。

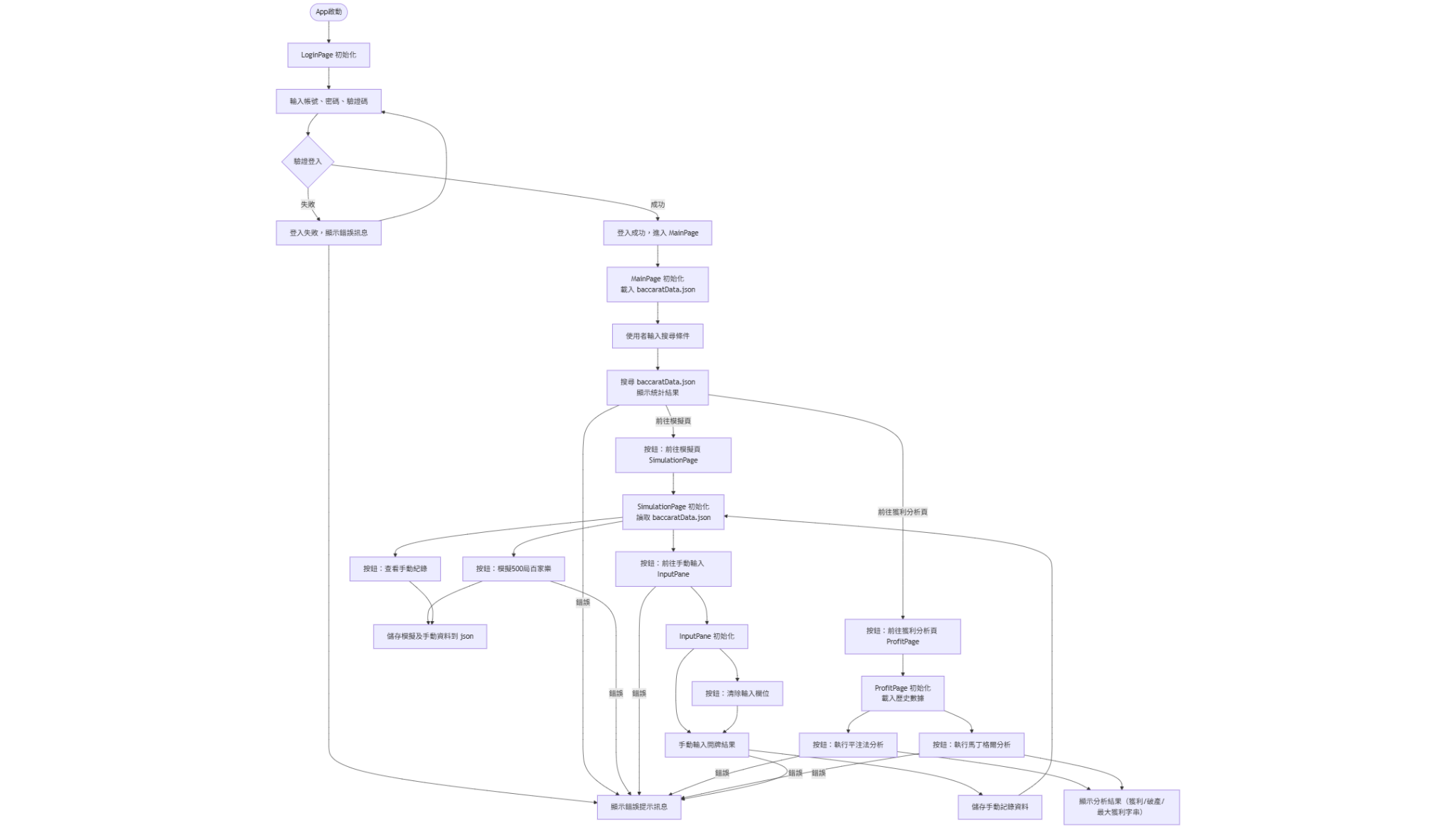
資料存取透過 FileSystem API 實作，預設資料包裝於應用程式資源中，使用者資料則儲存在本機應用資料目錄，確保資料持久性與安全性。系統具備資料清理機制，自動偵測並修復異常資料，提升資料品質。

### **2.4 主要功能流程**

* **登入驗證**：使用者輸入帳號密碼，系統比對預設憑證，成功後切換至主頁。
* **資料載入與清理**：系統啟動時，先清理本地資料中無效或格式錯誤的紀錄，再合併載入預設與用戶自訂資料。
* **會員編號搜尋**：使用者輸入會員編號後，系統於所有牌局紀錄中搜尋匹配字串，並統計出現次數與後續聯絡碼，結果以清單方式呈現。
* **多頁面導航**：提供快速切換至模擬、手動輸入與獲利分析頁面，實現完整的數據分析體驗。

### **2.5 系統安全與使用者體驗**

系統登入設有簡易帳號密碼機制，保護資料不被未授權使用。使用者介面設計注重易用性，例如登入頁密碼欄位預設聚焦，搜尋欄具備清除功能，搜尋結果即時更新，提升操作便利。



第三章 系統實現

本章介紹系統主要功能的程式實作，包括系統架構搭建、資料讀寫、用戶互動介面及核心功能演算法等。並說明系統各模組如何協同運作，達成設計目標。

### **3.1 開發環境與工具**

本系統採用 Microsoft .NET MAUI 框架進行開發，支援跨平台應用。開發環境為 Visual Studio 2022，程式語言為 C#，資料存取使用 JSON 格式，並透過 FileSystem API 實作檔案操作。

### **3.2 系統架構**

系統採用 MVVM 架構分層設計，將視圖（View）、商業邏輯（ViewModel）與資料（Model）分離。主要包含以下專案檔案：

* MainPage.xaml / MainPage.xaml.cs：主頁，負責會員搜尋功能與頁面導航。
* SimulationPage.xaml / SimulationPage.xaml.cs：模擬百家樂牌局與手動牌局紀錄。
* ProfitPage.xaml / ProfitPage.xaml.cs：獲利預測分析模組。
* BaccaratData.json：儲存預設及手動牌局紀錄的 JSON 檔案。

### **3.3 資料載入與儲存**

系統在啟動時會呼叫資料讀取方法，從 BaccaratData.json 讀取歷史牌局資料。載入時會先進行資料格式檢查與異常清理，移除格式錯誤或空白資料。程式碼範例如下：

csharp

複製編輯

private async Task LoadDataAsync()

{

try

{

using var stream = await FileSystem.OpenAppPackageFileAsync("BaccaratData.json");

using var reader = new StreamReader(stream);

string json = await reader.ReadToEndAsync();

baccaratData = JsonSerializer.Deserialize<BaccaratData>(json);

CleanInvalidData();

}

catch (Exception ex)

{

// 錯誤處理與提示

}

}

資料儲存則使用 FileSystem.Current.AppDataDirectory 路徑，確保使用者新增資料持久保存。

### **3.4 會員編號搜尋功能**

使用者可在主頁輸入會員編號後四碼，系統會比對所有牌局紀錄，搜尋包含該字串的紀錄，並統計出現次數及接續的聯絡碼。搜尋結果會即時更新並顯示於介面清單中。核心邏輯如下：

csharp

複製編輯

private void SearchMember(string searchKey)

{

var results = new List<SearchResult>();

foreach (var record in baccaratData.BaccaratHistory)

{

if (record.Contains(searchKey))

{

var contacts = ExtractContactNumbers(record, searchKey);

results.Add(new SearchResult { Record = record, Contacts = contacts });

}

}

SearchResults = results;

}

### **3.5 百家樂模擬與手動記錄**

在模擬頁面中，系統可自動模擬多局百家樂牌局，產生隨機牌局字串並新增至歷史資料。使用者亦可手動輸入真實牌局資料，並儲存以便後續分析。此功能透過以下方法實現：

csharp

複製編輯

private void SimulateGames(int count)

{

for (int i = 0; i < count; i++)

{

string simulated = GenerateRandomBaccaratString();

baccaratData.BaccaratHistory.Add(simulated);

}

SaveData();

}

手動輸入時，系統會驗證輸入格式正確，並更新至本地資料。

### **3.6 獲利分析模組**

獲利分析頁面實作投注模擬與收益計算。系統根據百家樂牌局字串，模擬平注與馬丁格爾下注策略，計算最大可能獲利及虧損，並以視覺化方式呈現分析結果。

分析邏輯範例如下：

csharp

複製編輯

private void AnalyzeProfit(string baccaratSequence)

{

int profit = 0;

int martingaleMultiplier = 1;

foreach (var segment in ParseSequenceSegments(baccaratSequence))

{

if (IsWinningSegment(segment))

profit += baseBet \* martingaleMultiplier;

else

{

profit -= baseBet \* martingaleMultiplier;

martingaleMultiplier \*= 2;

}

}

DisplayProfitResult(profit);

}

### **3.7 系統錯誤處理**

系統在資料載入、搜尋、模擬及儲存過程中，均有例外捕捉機制，避免因非預期錯誤造成系統當機，並提供用戶友善的錯誤提示。

第四章 測試與驗證

本章說明系統的測試方法與驗證過程，確保系統各項功能達到設計需求，並具備穩定性與正確性。

### **4.1 測試環境**

系統測試於以下環境進行：

* 作業系統：Windows 11 / Android 12
* 開發工具：Visual Studio 2022 (.NET MAUI)
* 測試裝置：PC 開發環境及 Android 手機模擬器與真機測試
* 測試資料：內建 baccaratData.json 測試資料與自訂模擬資料

### **4.2 測試策略**

本系統採用黑盒測試與白盒測試相結合的方式：

* **黑盒測試**：針對系統功能與使用者介面進行操作，檢驗各項功能能否正常執行並正確回應使用者輸入。
* **白盒測試**：針對系統核心演算法與資料處理流程，進行程式碼層級的邏輯驗證與單元測試。

### **4.3 主要測試項目與結果**

| **測試項目** | **測試內容** | **結果說明** |
| --- | --- | --- |
| 會員編號搜尋功能 | 輸入有效會員編號後四碼，搜尋並顯示相關紀錄與統計 | 搜尋結果正確且即時顯示 |
| 資料載入與格式驗證 | 載入 baccaratData.json，檢查資料完整性與異常資料剔除 | 成功剔除格式錯誤資料，載入正常 |
| 百家樂模擬功能 | 模擬多局百家樂牌局，檢查字串格式與資料持久化 | 模擬資料符合預期格式，儲存正確 |
| 手動牌局紀錄功能 | 輸入手動牌局紀錄並保存，重啟後確認資料完整性 | 手動紀錄可正確新增、讀取與顯示 |
| 獲利分析功能 | 測試平注與馬丁格爾策略的獲利計算，結果與預期相符 | 獲利分析計算正確，顯示結果符合邏輯 |
| 介面操作流暢度及錯誤提示 | 操作流程測試與異常輸入測試，檢查系統穩定性與錯誤訊息顯示 | 使用過程順暢，異常輸入均有適當錯誤提示 |

### **4.4 系統驗證**

透過上述測試，系統功能已符合設計目標，且操作介面直覺易用，能有效支援用戶進行百家樂資料搜尋、模擬與分析。各模組間協同良好，數據流程穩定，無重大錯誤發生。

此外，系統在多次重啟及長時間操作下，保持資料完整性及操作效能，具備良好可靠度。

### **4.5 測試心得與改進建議**

本系統已具備基本使用功能，未來可考慮增加以下改進方向：

* 增強錯誤處理機制，對於極端輸入狀況提供更詳細的用戶指引。
* 新增更多模擬策略及參數設定，提升分析功能多樣性。
* 優化介面設計與操作流程，提升使用者體驗。

第五章 結論與未來工作

### **5.1 結論**

本系統成功設計並實現了一套結合百家樂資料搜尋、模擬與獲利分析的應用平台。系統能有效載入與處理百家樂歷史牌局資料，並透過模擬功能生成多局牌局資料，為使用者提供豐富的分析基礎。

經過嚴謹的測試與驗證，系統各項功能穩定且符合設計需求，使用介面直覺，操作流暢，具備良好的擴展性與維護性。獲利分析模組亦能依據不同投注策略，模擬真實情境下的獲利狀況，為使用者提供實用的參考依據。

整體而言，本系統達成預期目標，能協助使用者在百家樂數據分析與策略制定上獲得實質幫助。

### **5.2 未來工作**

為進一步提升系統效能與功能，本研究建議未來工作方向包括：

* **增加即時資料擷取功能** 開發自動化爬蟲或 API 串接模組，實現即時獲取百家樂牌局資料，提升分析資料的時效性。
* **強化人工智慧分析能力** 結合機器學習技術，挖掘百家樂牌局中的潛在規律，提供更精準的策略建議與風險預測。
* **多平台支援與雲端同步** 推動系統跨平台應用（如 iOS、Web），並搭配雲端資料同步，方便使用者隨時隨地管理與分析資料。
* **使用者介面優化** 改善介面設計，提高使用者互動性與體驗，並加入更多視覺化報表與圖表呈現分析結果。
* **策略模擬擴展** 增加更多投注策略與自訂參數設定，讓使用者能根據自身需求進行多元模擬。

透過上述改進，期望系統能持續滿足百家樂愛好者與數據分析師的需求，成為專業且實用的分析工具。