

銀行系統專題報告

本專題旨在開發一個桌面銀行系統應用程式，以滿足銀行內部員工日常營運管理任務，並提供客戶便捷的帳戶查詢與基本金融交易服務。此系統採用分層架構設計，結合C#與SQLite技術，實現了多角色使用者認證、客戶帳戶管理、內部員工管理等核心功能。

指導老師: 江家頡

成員: B10090105 詹永旭

B11170019 巫宏珉

研究背景與目的

1

研究背景

隨著科技的飛速發展，金融服務業正經歷前所未有的數位轉型。銀行系統作為金融機構的核心，其安全性、效率與使用者體驗直接影響銀行營運及客戶滿意度。

2

研究動機

傳統銀行業務面臨線上化、自動化的挑戰，同時也需要因應日益嚴峻的資安威脅。本專題將嘗試解決如何在有限資源下，打造一個兼顧功能、效能與安全性的銀行系統。

3

研究目的

設計並實作一個安全、高效且直觀的桌面銀行系統，支援多角色使用者認證與授權，實現核心銀行業務功能，建立詳盡的交易記錄與審計日誌追蹤機制。

問題陳述與預期貢獻

問題陳述

現有銀行系統可能存在操作複雜、介面不直觀、權限管理不夠細緻等問題。對於中小型銀行或特定業務場景，尋求輕量級、易於部署且功能齊全的桌面應用程式，以簡化日常業務流程並提高效率，是一個實際的需求。

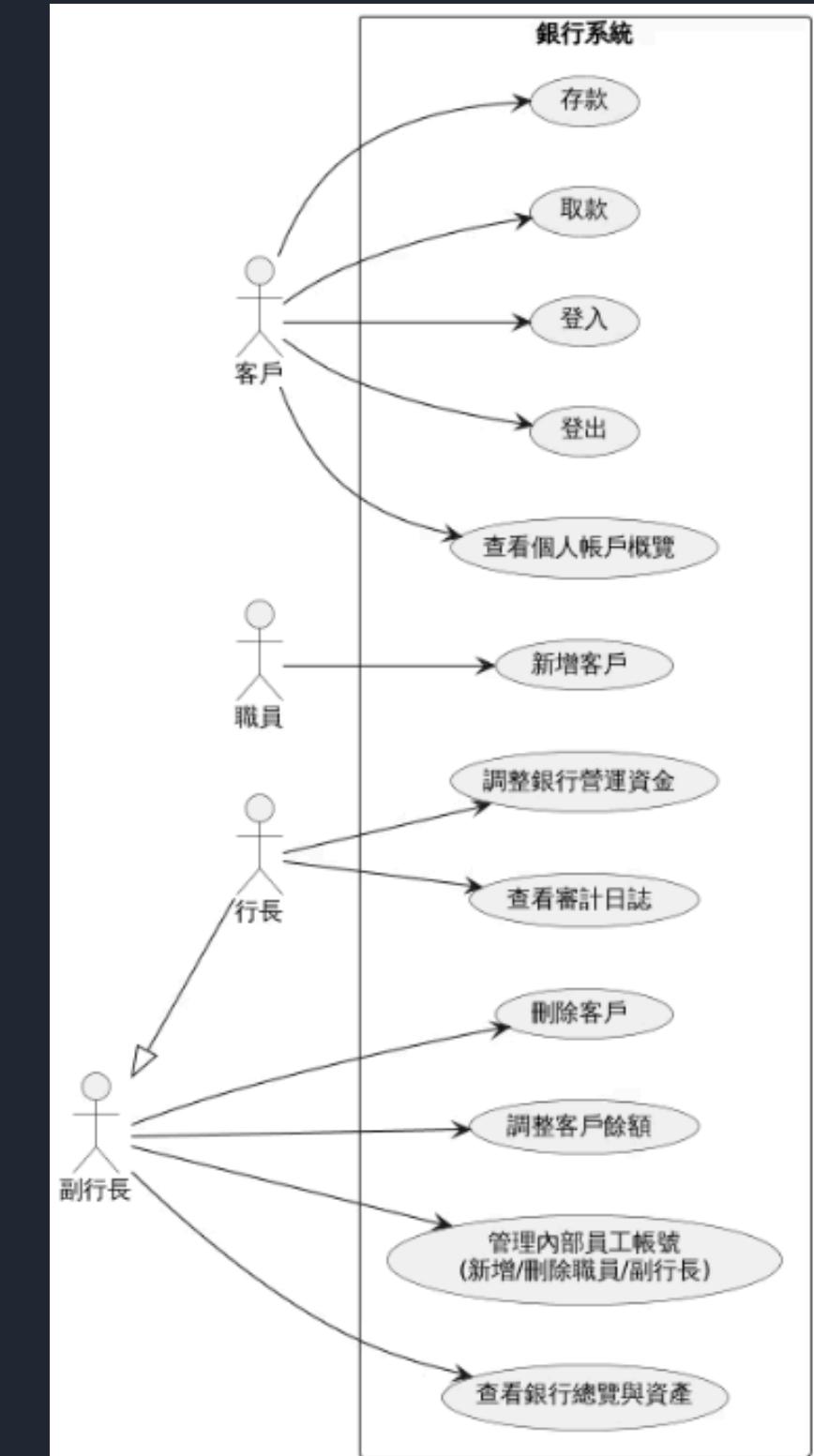
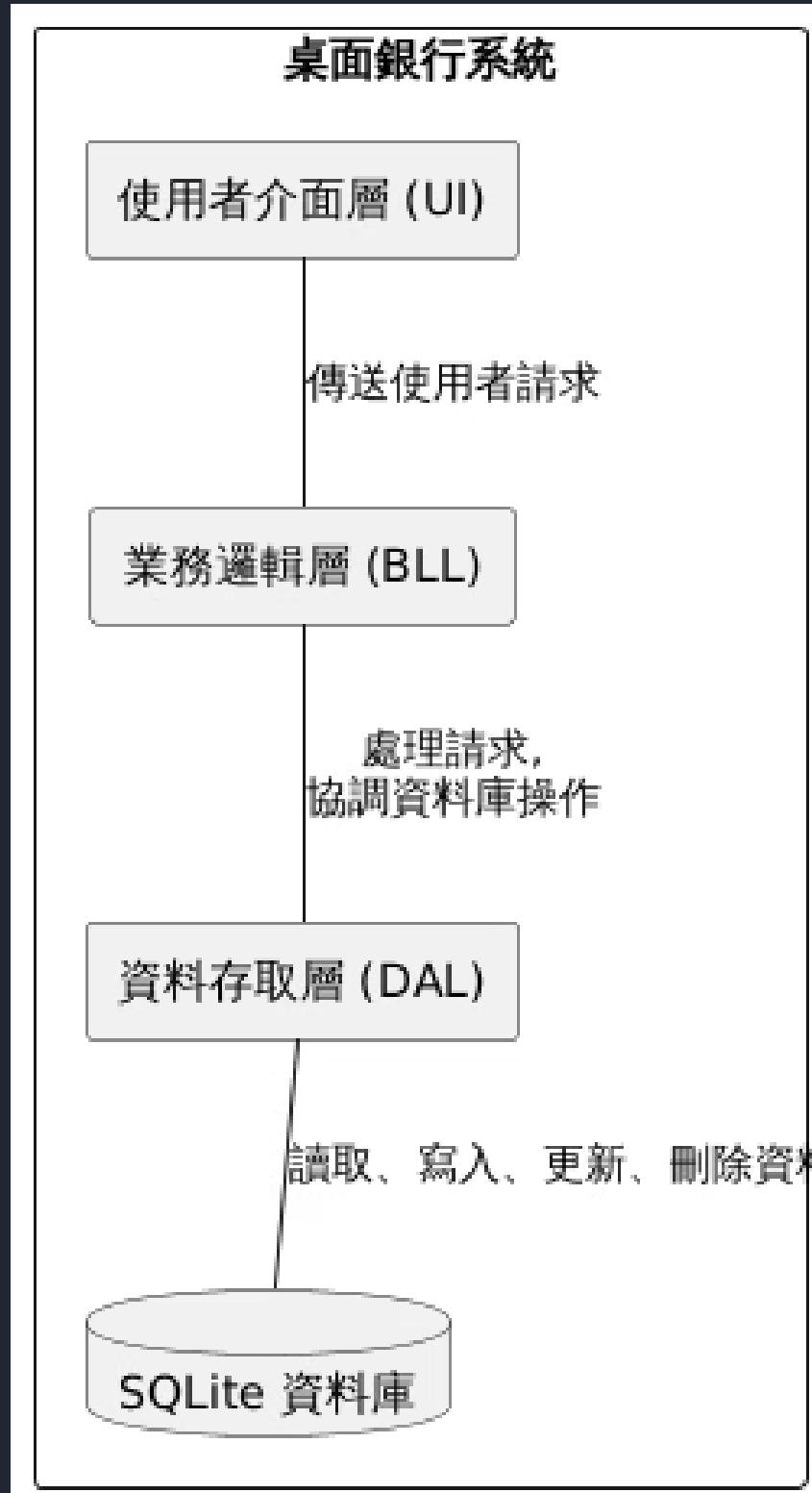
現有大型銀行系統的問題包括：操作複雜性與介面直觀性不足、權限管理不夠細緻、部署與資源限制高昂，以及缺乏示範與參考價值。

預期貢獻

本專題的成果預期能提供一個可作為示範的桌面銀行系統應用程式，展現如何運用C#和SQLite技術實現銀行核心業務功能。這將有助於：

1. 為未來的銀行系統開發專案提供參考設計與實作經驗
1. 提升內部員工的操作效率，降低人為錯誤
1. 為客戶提供更便捷的自助服務管道
1. 為銀行提供一套基礎的營運監控與審計工具

系統架構設計



功能模組說明



使用者認證模組

處理登入、登出、角色驗證及權限導向，確保系統安全性與適當的存取控制。



客戶帳戶管理模組

包括帳戶概覽、存款、取款、新增客戶、刪除客戶、調整客戶餘額等核心銀行業務功能。



銀行內部員工管理模組

負責新增、刪除、修改職員及副行長帳號，維護內部人員資料。



銀行總覽與資產查詢模組

顯示所有客戶帳戶列表、銀行營運資金及銀行總資產，提供全面的財務視圖。

此外，系統還包含銀行營運資金調整模組，允許行長調整營運資金並進行雙重驗證；以及審計日誌查詢模組，記錄並顯示所有敏感操作的詳細日誌，確保系統操作的可追溯性與合規性。

資料庫設計 - 使用者與帳戶

本系統採用SQLite作為資料儲存解決方案，輕量且易於部署。資料庫設計主要包含以下：

Users (使用者)

儲存所有系統用戶（客戶、職員、副行長、行長）的帳號、密碼雜湊值、姓名和角色等資訊。此表是系統用戶管理的核心，支援多角色認證與授權機制。

資料表：Users					
	Id	Username	PasswordHash	Name	Role
1	1	admin	5e884898da28047151d0e56f8dc629277360...	銀行行長	3
2	4	Clerk1	18d6ad41a00b1d39fd5263681142ea6199ac...	Clerk1	1
3	7	User1	27a534a25cf745b6c985eb782079a6fe8641...	User1	0
4	14	Clerk2	7b6e18f27102391ff468734cf073fa62c446...	Clerk2	1
5	16	Vice	9a9036ab0f9ad5acbb24c0eb4741fe01d082...	Vice	2
6	17	User2	0e238ae88aef5a81ba9d297b5df67e74af15...	User2	0

Accounts (帳戶)

儲存客戶的銀行帳戶資訊，包含帳號、所屬客戶ID和當前餘額。此表與Users表關聯。

資料表：Accounts					
	Id	UserId	AccountNumber	Balance	CreationDate
1	2	7	ACC20250605221247216	1000.3	2025-06-05 22:12:47
2	9	17	ACC20250605224113743	1000.0	2025-06-05 22:41:13

資料庫設計 - 審計與營運資金

The screenshot shows a database interface with three tables:

- Users**: A table listing users with columns: Id, Username, PasswordHash, Name, Role, and CreationDate.
- BankOperationalFunds**: A table named "Funds" with columns: Id and Amount. It contains a single row with Id 1 and Amount 100000.0.
- Transactions**: A table with columns: Id, AccountId, Type, Amount, Timestamp, and Description. It lists six transactions involving accounts 6, 11, 12, 13, 14, and 15.

Id	Username	PasswordHash	Name	Role	CreationDate
1	admin	5e884898da28047151d0e56f8dc629277360...	銀行行長	3	2025-06-04 2
2	Clerk1	18d6ad41a00b1d39fd5263681142ea6199ac...	Clerk1	1	2025-06-05 0
3	User1	27a534a25cf745b6c985eb782079a6fe8641...	User1	0	2025-06-05 2
4	Clerk2	7b6e18f27102391ff468734cf073fa62c446...	Clerk2	1	2025-06-05 2
5	Vice	9a9036ab0f9ad5acbb24c0eb4741fe01d082...	Vice	2	2025-06-05 2
6	User2	0e238ae88aef5a81ba9d297b5df67e74af15...	User2	0	2025-06-05 2

Funds	
Id	Amount
1	100000.0

Id	AccountId	Type	Amount	Timestamp	Description
1	6	2	2	1000.0	2025-06-05 22:26:57 由 admin 調整: 開戶金額
2	11	9	2	1000.0	2025-06-05 22:41:22 由 Vice 調整: 開戶金額
3	12	9	0	1000.0	2025-06-05 23:24:15 由 User2 執行存款
4	13	9	1	500.0	2025-06-05 23:24:22 由 User2 執行取款
5	14	9	1	500.0	2025-06-05 23:24:34 由 User2 執行取款
6	15	2	0	0.3	2025-06-05 23:38:14 由 User1 執行存款

AuditLogs (審計日誌)

記錄所有關鍵系統操作，包含日誌類型、操作者、操作時間、詳細描述和相關資訊。此表對於系統安全性和合規性至關重要，提供完整的操作追蹤記錄，有助於問題排查和安全審計。

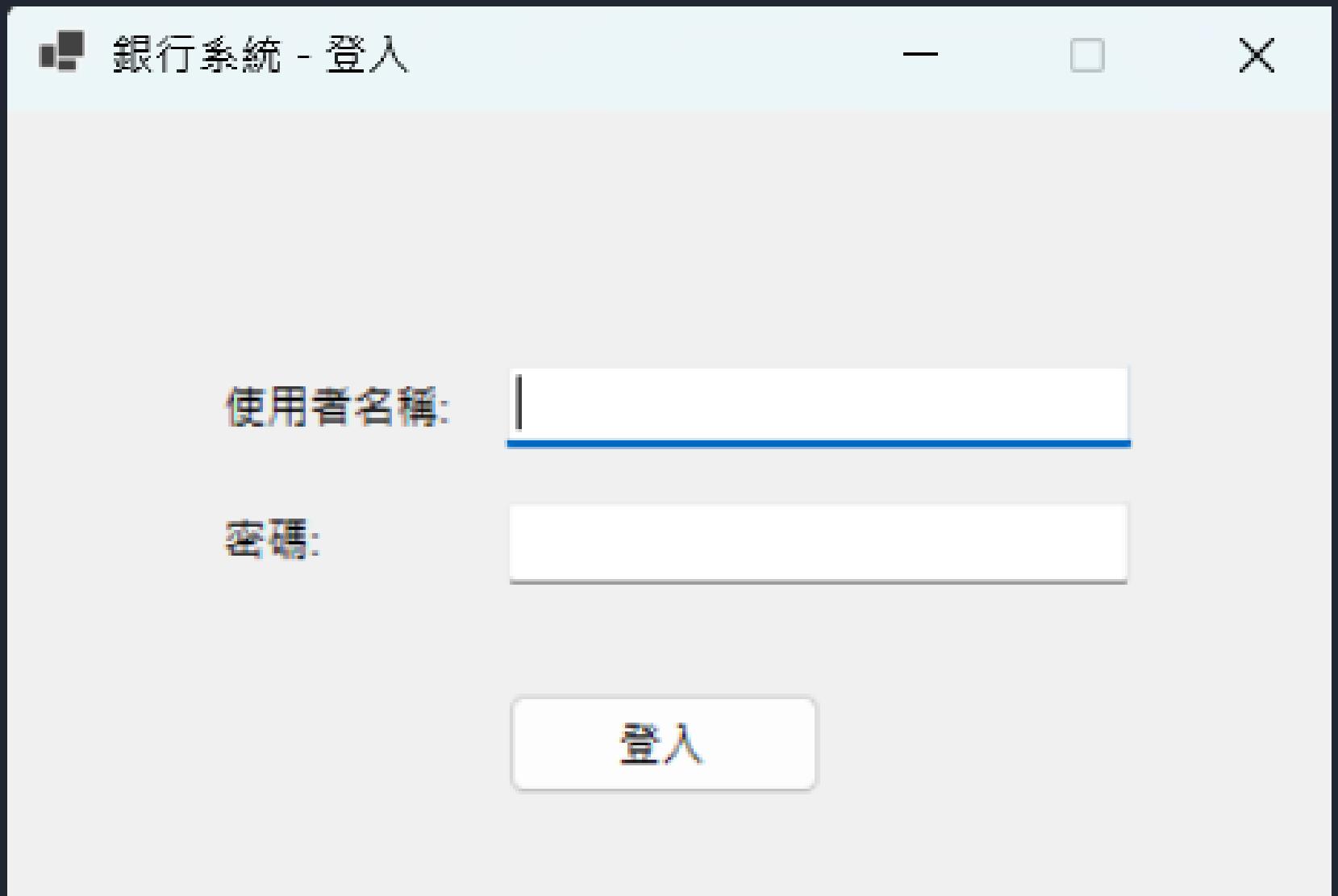
BankFunds (銀行營運資金)

單一記錄，儲存銀行目前的營運資金金額。此表雖然結構簡單，但在銀行系統中扮演重要角色，記錄銀行可用於營運的資金總額，是銀行財務狀況的重要指標。

Transactions (交易)

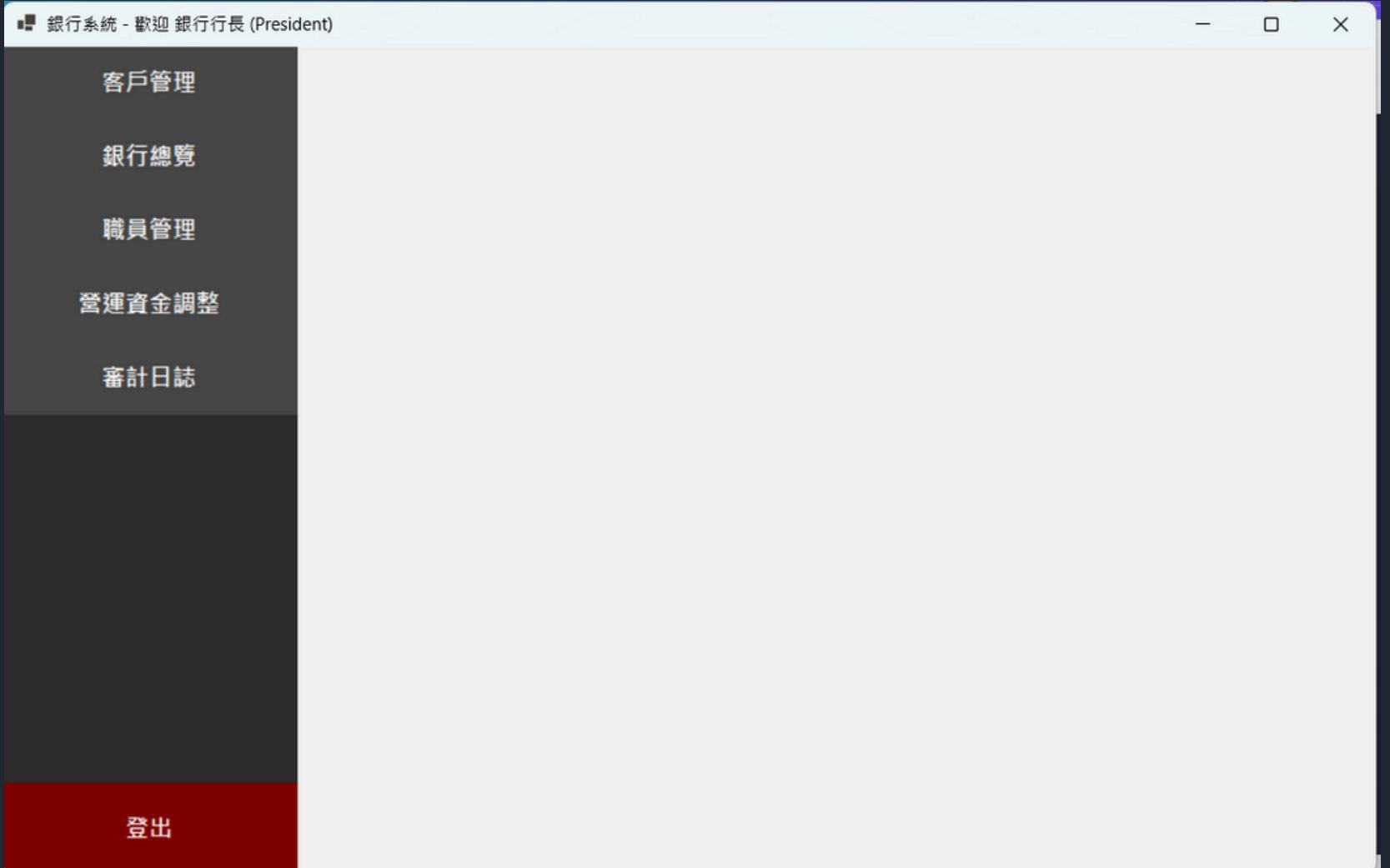
記錄所有客戶帳戶的金融交易，包含交易ID、帳戶ID、交易類型、金額、時間和描述，確保所有財務活動可追蹤。

系統畫面展示 登入畫面



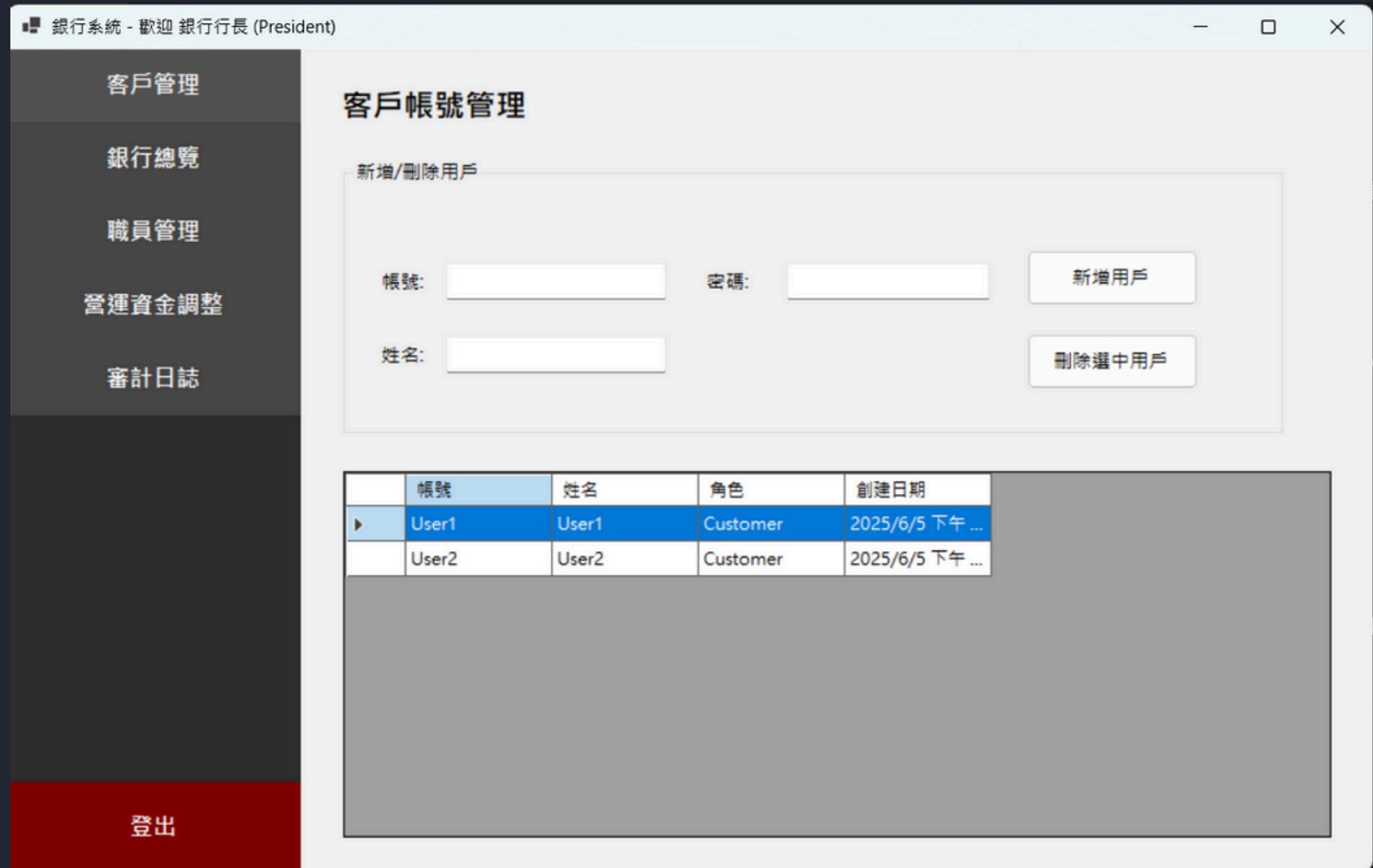
1. 密碼輸入時應以星號 (*) 或圓點 (•) 遮蔽顯示，保護使用者隱私。
2. 當登入失敗時（例如：帳號或密碼錯誤、網路連線問題），系統會在此處顯示明確且友善的錯誤訊息，引導使用者進行修正。
3. 密碼雜湊儲存：確保即使資料庫被洩露，密碼也無法直接被讀取。

主要操作畫面



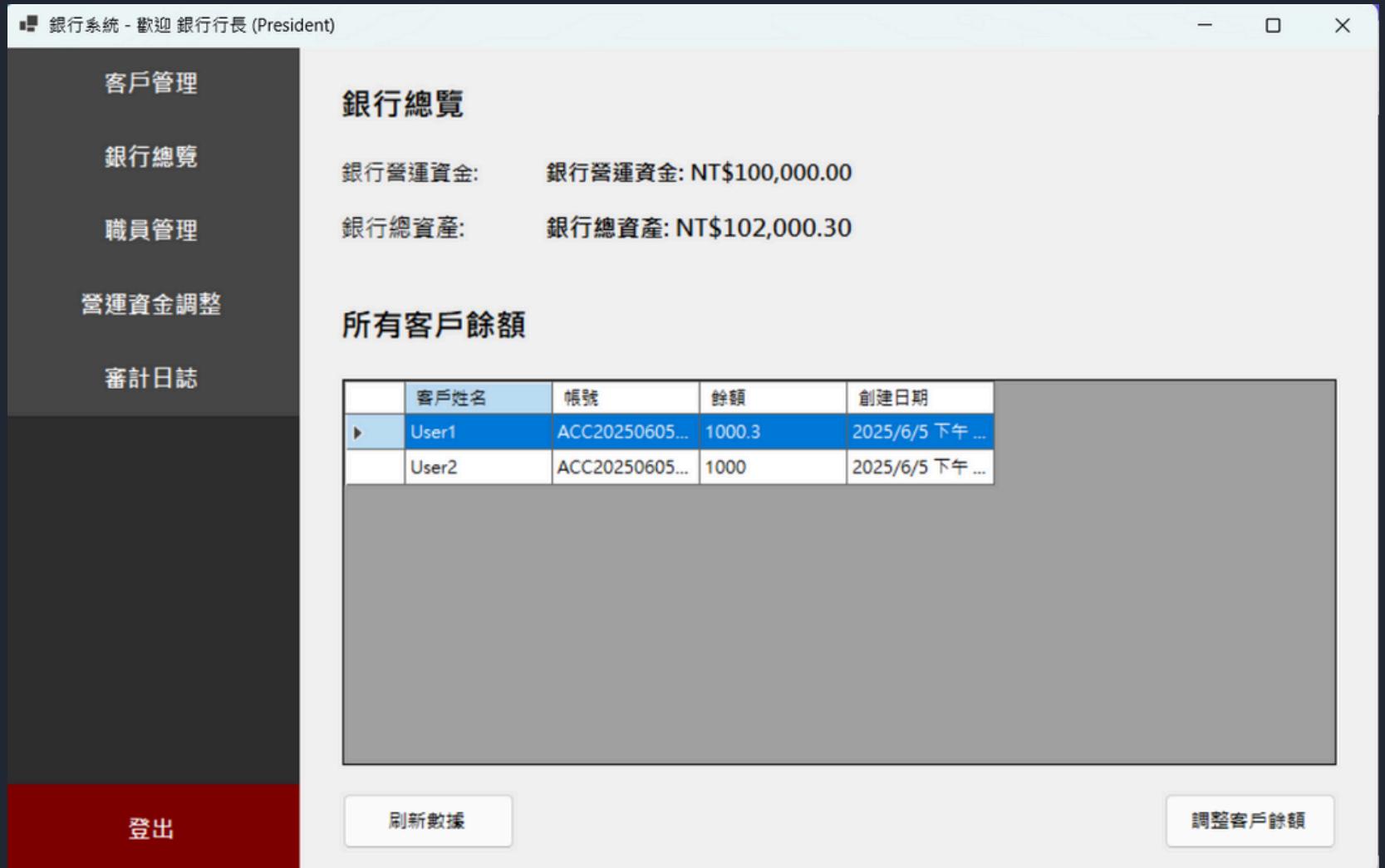
1. 左側導航菜單區 (Left Sidebar Navigation)：一個垂直的菜單列表，根據當前登入使用者的角色權限動態顯示可用的功能模組。
2. 主畫面會根據登入的使用者角色，動態調整其可見的菜單選項和主要內容區的默認呈現

客戶帳號管理畫面

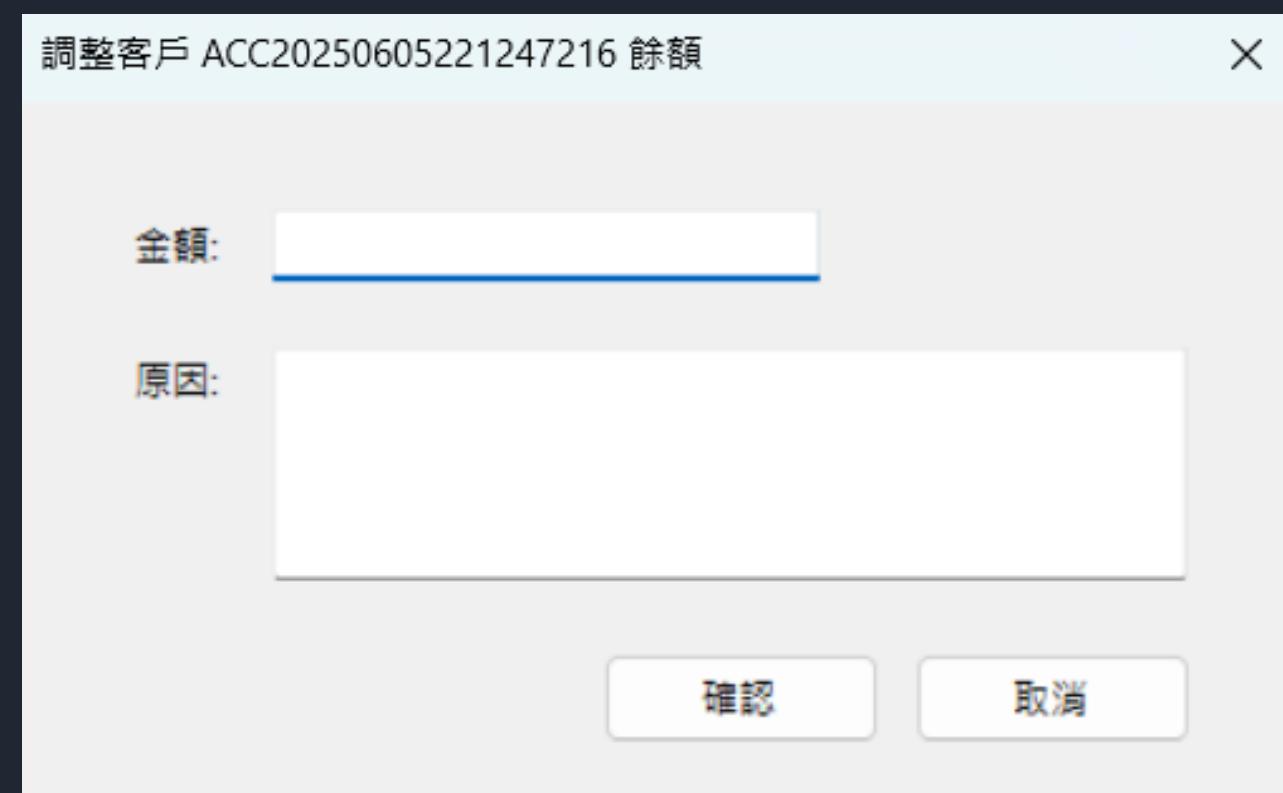


1. 允許為銀行客戶創建/刪除帳號，僅限職員及以上身份
2. 檢查帳戶是否已被註冊過
3. 列出所有客戶的帳號、姓名、身分、創建日期

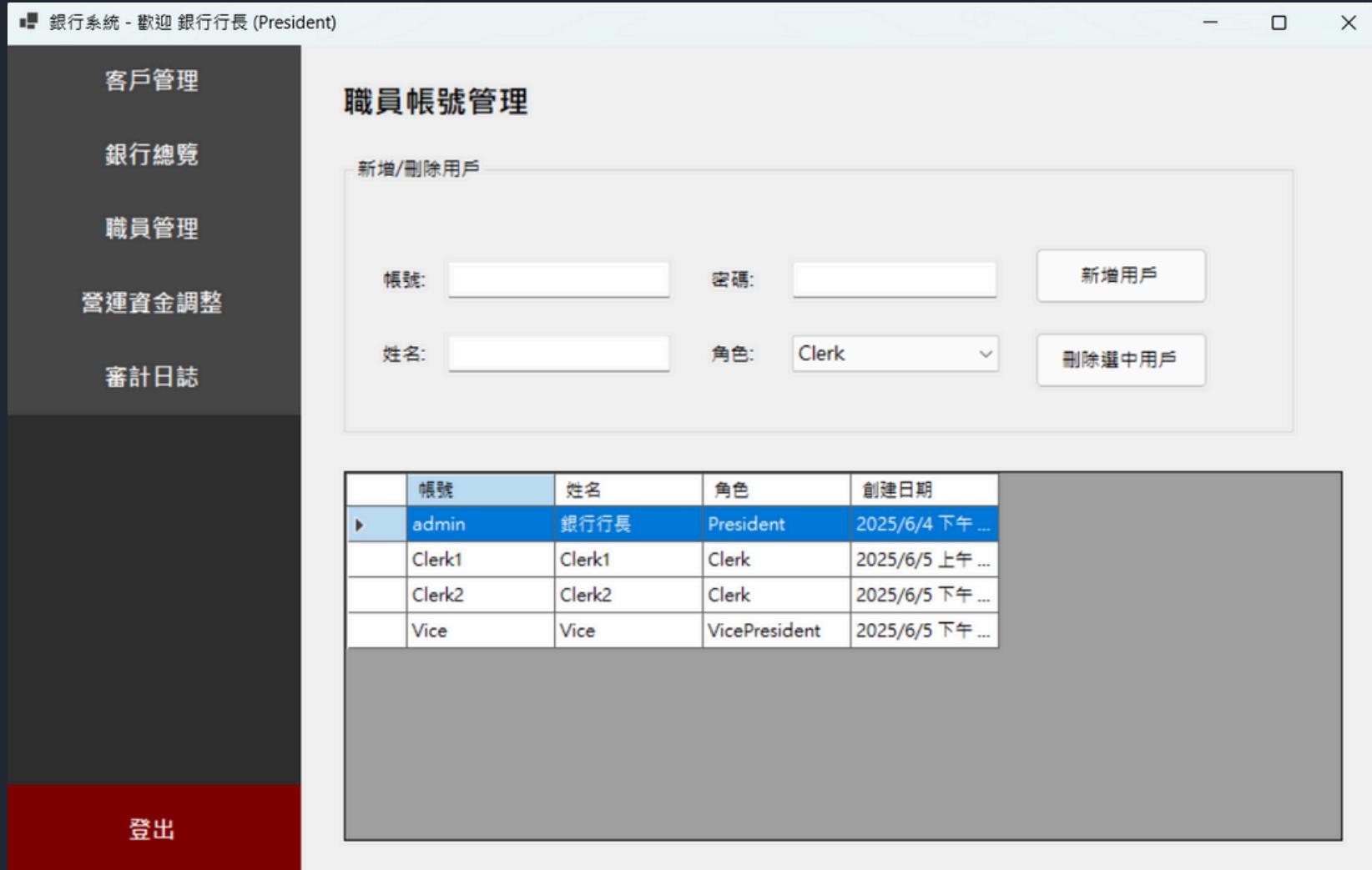
銀行總覽畫面



- 顯示所有客戶的姓名、帳號和當前餘額。行長可以瀏覽這些客戶帳戶。
- 顯示銀行當前用於日常營運的資金總額
- 計算並顯示所有客戶存款的匯總金額
- 從所有現有客戶中選擇一個特定的客戶帳號，輸入調整金額及原因
- 更改會記錄到審計日誌



職員管理



- 僅限副行長及以上身份
- 清晰列出所有現有職員的資訊
- 新增職員表單區，職員帳號、初始密碼、確認密碼、職員姓名、角色選擇
- 從所有現有職員中選擇一個特定的帳號進行刪除

營運資金調整



1. 顯示銀行當前營運資金。
2. 僅限行長操作
3. 輸入一位副行長的帳號和對應的密碼
4. 在「新營運資金金額」輸入框中填入期望的目標金額。
5. 在「操作說明/原因」文字區塊中，詳細填寫本次調整的具體原因。
6. 詳細記錄此次「營運資金調整」操作到審計日誌中

審查日誌



- 此功能僅限於行長 (President) 使用
- 列出所有審計日誌記錄
- 不應允許任何使用者（包括行長）進行修改或刪除
- 所有關鍵操作都應被記錄，不遺漏任何潛在的敏感事件

個人帳戶



1. 此畫面僅限於客戶 (Customer) 角色使用
2. 顯示客戶帳戶的實時餘額
3. 「存款」按鈕，系統會切換到存款操作介面
4. 「取款」按鈕，系統會切換到取款操作介面
5. 瀏覽交易歷史：客戶可以滾動瀏覽其交易歷史列表，查看過去的金融活動。

研究成果與專題限制



研究成果總結

成功開發基於C#和SQLite的桌面銀行系統應用程式



專題限制

SQLite資料庫限制、安全性未臻完善、桌面應用限制、即時性問題



系統價值

提供參考設計與實作經驗，提升操作效率，降低人為錯誤

本專題成功實現了多角色使用者認證、客戶帳戶管理、內部員工管理、銀行總資產查詢以及審計日誌追蹤等核心功能。系統設計採用分層架構，提高了程式碼的可維護性和可測試性。通過單元測試和整合測試，驗證了系統功能的正確性和穩定性。

然而，SQLite作為檔案型資料庫不適用於高併發環境，限制了系統擴展性；安全機制尚未包含多因素認證等進階功能；作為桌面應用也限制了系統的擴展性和即時性。

未來改進方向



資料庫升級

考慮升級至SQL Server、PostgreSQL或MySQL，支援高併發和分散式部署



增強安全性

導入多因素認證、細粒度存取控制、敏感資料加密



介面優化

採用現代UI框架、加入資料匯出功能、優化報表生成



模組化與擴展性

將業務邏輯抽象為服務、考慮微服務架構



性能優化

資料庫查詢索引優化、頻繁操作緩存處理

未來還可增加更多的自動化測試案例，包括UI自動化測試，以提升測試效率和系統品質。這些改進方向將使系統更加穩健、安全、高效，並能更好地適應未來可能的業務擴展需求。