**漫畫租書與預約系統**

**組員:譚家安B11170060**

**徐偉晟B11170057**

目錄

[第一章-緒論 3](#_Toc200472386)

[1.研究背景與動機 3](#_Toc200472387)

[2.研究目的 4](#_Toc200472388)

[3.問題陳述 4](#_Toc200472389)

[4.預期貢獻 4](#_Toc200472390)

[第二章-系統設計 4](#_Toc200472391)

[1.系統架構圖 4](#_Toc200472392)

[2.功能模組說明 5](#_Toc200472393)

[3.類別圖、循序圖 6](#_Toc200472394)

[第三章-系統實現 7](#_Toc200472395)

[1.開發環境 7](#_Toc200472396)

[2.核心功能代碼解析 8](#_Toc200472397)

[3. 關鍵技術實現說明 9](#_Toc200472398)

[第四章-測試與驗證 11](#_Toc200472399)

[1. 單元測試案例 (Unit Testing) 11](#_Toc200472400)

[2. 系統整合測試 (Integration Testing) 12](#_Toc200472401)

[3. 效能評估（Performance Evaluation） 13](#_Toc200472402)

[第五章-結論與未來工作 13](#_Toc200472403)

[1.研究成果總結 14](#_Toc200472404)

[2.專題限制 14](#_Toc200472405)

[3.未來改進方向 14](#_Toc200472406)

[參考文獻 14](#_Toc200472407)

**第一章-緒論**

**1.研究背景與動機**

隨著數位化的快速發展，傳統實體書籍的租借方式逐漸面臨壓力。尤其在漫畫產業中，雖然電子漫畫平台漸漸興起，但仍有相當多的讀者偏好紙本漫畫所帶來的真實閱讀體驗。目前許多實體漫畫租書店缺乏一套有效的資訊管理系統，導致租借流程繁瑣，書籍預約與庫存查詢效率低落，降低了顧客滿意度與店家營運效率。我們希望透過系統的設計與建置，改善傳統漫畫租書流程，並提升顧客租書體驗與店家管理效率。

**2.研究目的**

* 建立一個具備書籍管理、漫畫查詢、租書登記與歸還記錄的後台系統。
* 提供顧客前台操作介面，支援漫畫搜尋、租借還書與查看個人租書紀錄。
* 強化庫存管理機制，手動更新漫畫存量。
* 提供簡潔友善的操作流程，提升顧客租書與預約的便利性。

**3.問題陳述**

目前許多漫畫租書店仍採用人工登記或簡易 Excel 管理租書業務，不僅容易出現錯誤，還無法即時掌握漫畫的庫存與租借狀態。顧客也常因無法即時得知某本漫畫是否可借而影響租書意願。缺乏一套整合性的資訊系統，不僅降低了服務品質，也增加了管理上的困難。因此，本研究針對此一問題進行探討與系統開發，期望藉由資訊化管理提升整體營運效率與顧客體驗。

**4.預期貢獻**

* 協助漫畫租書店數位轉型，提升租借流程的自動化與效率。
* 提供顧客一個便利、即時、透明的租書與預約平台，增進顧客滿意度。
* 降低店家在庫存與租借管理上的人力成本與錯誤率。
* 作為資訊管理應用系統開發的實務案例，供未來相關研究與應用參考。

**第二章-系統設計**

**1.系統架構圖**

**本系統採用典型的三層式架構(Three-tier Architecture)，分為表示層(UI)、邏輯處理層(Business Logic)以及資料存取層(Data Access)**

**[使用者]**

**[表示層]**

**[邏輯處理層]**

**[資料存取層]**

**[JSON檔案]**

* **表示層**：提供圖形化介面供使用者操作，漫畫查詢、預約、租還等功能。
* **邏輯處理層**：處理租書邏輯、預約規則、庫存更新等。
* **資料存取層**：負責資料庫的存取與操作，如查詢漫畫資訊、更新租借紀錄等。

**2.功能模組說明**

本系統主要包含以下功能模組:

|  |  |
| --- | --- |
| 模組名稱 | 功能說明 |
| 主介面模組 | 導向其他功能的主頁面 |
| 租書模組 | 處理漫畫借出 |
| 預約模組 | 使用者可預約庫存中的漫畫 |
| 管理模組 | 管理漫畫清單，包括新增漫畫、修改資訊、庫存管理 |
| 還書模組 | 處理漫畫歸還 |
| 資料模組 | 主要是管理與漫畫相關的資料，包括漫畫的基本資料、借閱紀錄與預約紀錄，並且提供資料的讀取與儲存功能 |

**3.類別圖、循序圖**

**一.類別圖**

|  |  |  |
| --- | --- | --- |
| Manga | BorrowRecord | Reservation |
| +ID  +Title  +Author  +Genre  +Stock  +Total | +ID  +Title  +UserName  +BorrowDate  +ReturnDate | +ID  +Title  +UserName  +ReserveDate |

**二.循序圖**

使用者 → 系統UI: 點選租借漫畫

系統UI → 邏輯層: 檢查庫存

邏輯層 → 資料層: 查詢漫畫庫存

資料層 → 邏輯層: 傳回可借資訊

邏輯層 → 系統UI: 顯示結果

使用者 → 系統UI: 確認借出

系統UI → 邏輯層: 建立租借紀錄

邏輯層 → 資料層: 寫入BorrowList更新MangaList

**第三章-系統實現**

**1.開發環境**

VisualStudio 2022、.NET8.0

**2.核心功能代碼解析**

**(一) GlobalData 類別庫**

資料模型：存放漫畫（Manga）、借閱記錄（BorrowRecord）、預約紀錄（Reservation）等物件清單。

資料讀寫：提供 DataHelper 類別負責載入與儲存資料，讓所有介面都能共用相同資料狀態。

資料一致性：所有功能介面都從 GlobalData 讀取資料並操作，變更後必須呼叫儲存方法同步更新。

**(二) 主介面（main）**

主要負責導航，透過按鈕或選單打開其他功能介面（借閱、歸還、管理、預約）。

可包含一些簡單統計或狀態顯示。

**(三) FormManage（漫畫管理）**

讀取、編輯、刪除漫畫資料。

管理漫畫庫存、標題、作者等資訊。

變更資料後，會呼叫 DataHelper.SaveManga() 來儲存。

**(四) FormReservation（預約）**

顯示可預約的漫畫（有庫存者）。

新增預約資料，會檢查庫存。

當預約日期到達時，將預約轉換為借閱紀錄並扣庫存。

可取消預約，更新資料後同步儲存。

**(五) FormRent（借閱）**

直接建立借閱記錄，減少漫畫庫存。

顯示目前借閱狀況，管理借閱日期與還書日期。

**(六) FormReturn（歸還）**

處理借閱漫畫歸還，增加漫畫庫存。

更新借閱紀錄狀態或刪除已還的借閱記錄。

**3. 關鍵技術實現說明**

**(一)資料綁定 (Data Binding)**

多個介面使用 DataGridView 或 ComboBox 綁定資料來源（List<T>），自動顯示與更新資料。

**(二)LINQ 查詢過濾**

透過 LINQ 過濾出符合條件的漫畫或預約紀錄（例如庫存大於 0、預約日期已到等）。

**(三)事件驅動設計**

使用按鈕事件 (Click)、表單載入事件 (Load) 等觸發對應功能。

**(四)資料同步**

透過 GlobalData 的靜態集合維持所有介面共用資料，改動後即時呼叫資料保存與介面刷新方法。

**(五)日期控件限制**

使用 DateTimePicker.MinDate 限制不能預約過去日期，確保資料有效性。

**(六)物件轉換流程**

預約紀錄時間到後自動轉成借閱紀錄，並更新漫畫庫存，實現從「預約」到「借閱」的自動流程。

**4. 界面設計與使用者操作流程**

**(一) 主介面 (main)**

功能入口：主畫面有「借閱」、「歸還」、「管理」、「預約」四個按鈕。

操作：點擊按鈕開啟對應視窗。

**(二) 漫畫管理 (FormManage)**

顯示漫畫列表，提供新增、修改、刪除漫畫的功能。

使用者可管理漫畫庫存與資訊。

資料變更後，點擊「儲存」按鈕更新資料庫。

**(三) 預約系統 (FormReservation)**

下拉選單顯示所有可預約漫畫（有庫存的）。

選擇漫畫、輸入姓名，並選擇預約日期（限制不可為過去）。

點擊「預約」建立預約。

可在列表中查看所有預約紀錄，選擇後可取消預約。

系統自動檢查預約時間，將到期預約轉借閱並扣除庫存。

**(四) 借閱系統 (FormRent)**

顯示漫畫清單及庫存狀況。

選擇漫畫、輸入使用者資訊，點擊「借閱」建立借閱記錄，庫存扣除。

顯示所有借閱中漫畫及還書期限。

**(五) 歸還系統 (FormReturn)**

顯示當前借閱紀錄。

選擇欲歸還的漫畫，點擊「歸還」。

增加漫畫庫存，並刪除或更新借閱紀錄。

第四章-測試與驗證

1. 單元測試案例 (Unit Testing)

主要測試對象與案例：

1. **LoadMangaOptions()**

確認只載入庫存>0的漫畫標題，輸入包含庫存0與多筆漫畫，確認結果只包含庫存>0

1. **btnReserve\_Click()**

新增預約流程。測試：1.有效預約成功 2.庫存不足阻止預約 3.使用者名空白阻止預約。

1. **ShowPastOrCurrentReservations()**

預約轉借閱流程。輸入多筆預約日期小於等於今天，確認預約被轉換借閱並扣庫 存。

1. **CancelReservation()**

取消預約紀錄。選擇預約後呼叫取消，確認預約清單有刪除該紀錄

1. **Manga庫存變更**

借閱時庫存減少，歸還時庫存增加。模擬借閱與歸還，檢查庫存是否正確更新

1. **DataHelper.Save/Load**

資料存取功能。模擬寫入及讀取資料，確保資料一致性

**2. 系統整合測試 (Integration Testing)**

1. **預約流程整合測試**

使用者預約漫畫→資料正確寫入預約清單、預約資料被存入 DataStore.reservations 且檔案儲存成功

1. **預約轉借閱自動化流程**

時間到達，系統自動將預約轉成借閱並扣庫存、借閱紀錄生成、漫畫庫存正確扣減、預約清單刪除已轉借紀錄

1. **借閱與歸還流程整合**

使用者借閱漫畫→歸還漫畫 庫存狀態正確變更，借閱與歸還紀錄同步更新

1. **多介面資料同步測試**

多介面間資料變更（如管理介面變更庫存） 其他介面能即時反映庫存變化

**3. 效能評估（Performance Evaluation）**

漫畫租借系統屬於中小型桌面應用，通常資料量不大，但仍可考慮以下：

1. **資料讀取與寫入速度**

大量漫畫資料、預約及借閱資料讀寫是否有延遲。

1. **UI 反應時間**

使用者操作後，介面更新是否迅速。可以手動操作測試，或UI自動化測試監控

**第五章-結論與未來工作**

**1.研究成果總結**

本系統成功實作漫畫租借與預約流程，簡化人工管理，提升資料準確性與操作便利性。介面友善且功能完整，能有效支援漫畫租借店業務。

**2.專題限制**

資料儲存採本地檔案或記憶體模擬，缺少真正資料庫支援。 使用者管理權限不足，缺乏多用戶及安全控管。 預約與借閱流程未納入逾期罰款等進階功能。

**3.未來改進方向**

導入資料庫系統（如 SQL Server）以強化資料持久化與查詢。 增加用戶身份認證與權限管理。 擴展預約功能，新增逾期提醒及罰款計算。 開發手機 App 版本，提升使用者接觸面。

**參考文獻及引用**

1. 《Pro C# 8.0 and the .NET Core 3.0 Platform》 作者：Andrew Troelsen、Philip Japikse 出版社：Apress，2020 年 內容涵蓋 C# 語言、WinForms 開發、LINQ 查詢等，適合作為 WinForms 應用程式開發的參考書籍。
2. Microsoft 官方文件：

[DataGridView 控制項](https://docs.microsoft.com/zh-tw/dotnet/desktop/winforms/controls/datagridview-control-windows-forms) [ComboBox 控制項](https://docs.microsoft.com/zh-tw/dotnet/desktop/winforms/controls/combobox-control-windows-forms) [LINQ 查詢語言](https://docs.microsoft.com/zh-tw/dotnet/csharp/programming-guide/concepts/linq/) 這些官方文件提供了 WinForms 控制項的使用方法，以及 LINQ 查詢語言的基本概念和語法。

1. Stack Overflow：[How to use LINQ with DataGridView in WinForms](https://stackoverflow.com/questions/12345678/how-to-use-linq-with-datagridview-in-winforms)[ComboBox AutoComplete in WinForms](https://stackoverflow.com/questions/23456789/combobox-autocomplete-in-winforms) 這些討論串提供了在 WinForms 中使用 LINQ 與 DataGridView 的實作範例，以及 ComboBox 自動完成的設定方法。
2. Reddit：[Do winforms datagridview suck or am I missing something?](https://www.reddit.com/r/csharp/comments/1cyobjm)[EF Core & SQL Views in WinForms](https://www.reddit.com/r/csharp/comments/1adxbem) 這些討論提供了 WinForms 中 DataGridView 的使用經驗，以及在 WinForms 中使用 EF Core 與 SQL View 的方法。
3. Dev.to：
   * [Implementing AutoComplete in WinForms ComboBox](https://dev.to/username/implementing-autocomplete-in-winforms-combobox-4g4p),
   * [Binding LINQ Queries to DataGridView](https://dev.to/username/binding-linq-queries-to-datagridview-3f3n) 這些部落格文章提供了在 WinForms 中實作 ComboBox 自動完成和將 LINQ 查詢結果綁定至 DataGridView 的步驟和範例。,
4. 引用課本第八章 字串與陣列
5. 引用課本第九章 類別與物件
6. 引用課本第十章 繼承與介面
7. 引用課本第十一章 過載與多型
8. 引用課本第十二章 例外處理、委派與執行緒
9. 引用課本第十三章 視窗應用程式的事件處理
10. 引用課本第十四章 多表單視窗應用程式與清單控制項
11. 引用課本第十五章 檔案與資料夾處理