二手書販賣系統

Used Book Sales System

1063519 羅哲琛 1063524 邱寶萱 1063530 林欣彤

目錄

一、簡介.....3
1.1 一般描述資訊.....3
二、規格書內容描述.....3
2.1 系統概述.....3
2.1.1 系統目標.....3
2.1.2 系統範圍.....3
2.1.3 系統架構.....3
2.2 軟體架構.....4
2.3 軟體類別定義.....5
2.4 軟體類別描述.....6
2.5 軟體行為概述.....13
三、軟體設計描述.....16
3.1 設計概述.....16
3.2 軟體類別設計描述.....16
3.3 執行概念.....33
3.4 需求追溯.....33
3.5 重複使用性.....34
四、軟體介面設計描述.....34
4.1 介面設計概述.....34
4.2 外部介面定義.....34
4.3 函式介面定義.....36
4.4 類別介面定義.....38

圖目錄

圖一 系統功能架構圖.....4
圖二 Package Diagram.....5
圖三～圖七 Sequence Diagrams.....13
圖八 Client-Object Diagram.....15
圖九～圖十三 Collaboration diagram.....16
圖十四～圖十七 18
圖十八～圖二十一 軟體視窗規劃圖.....35

一、簡介

1.1 一般描述資訊

a. 議題日期和狀態：此版於 2020/05/10 尚在討論階段未正式完成。
b. 範圍：本規格書的內容以二手書販賣系統之操作介面等為主。
c. 議題機構：各大學校學生或有書籍販賣需求的人所建立之系統。
d. 參考：IEEE/EIA 12207.1-1997 內容：二手書系統之標準規格書。
e. 主體：包含有系統概述、各項功能定義說明、使用者介面介紹、資料定義等。
f. 規格書修改歷史：此版於 2020/05/10 為第 1 版。

二、規格書內容描述

2.1 系統概述

2.1.1 系統目標

- 讓需要販售書籍的人有簡潔明瞭的平台

  為保持市面上書籍更廣泛的流通，抱持的理念即是宣導「知識回收再利用」的觀念，提倡「有用的書籍不銷毀、不丟棄」的環保精神，並積極提供愛書人有品質的書籍和一個有品質的購書環境。也讓人整理自己的叢書，將家中看過的好書分享給他人。

- 操作介面清楚易懂易操作

  本系統使用者介面以清楚容易操作為特色，搭配各項功能說明，讓一般使用者可以更容易使用操作。

2.1.2 系統範圍

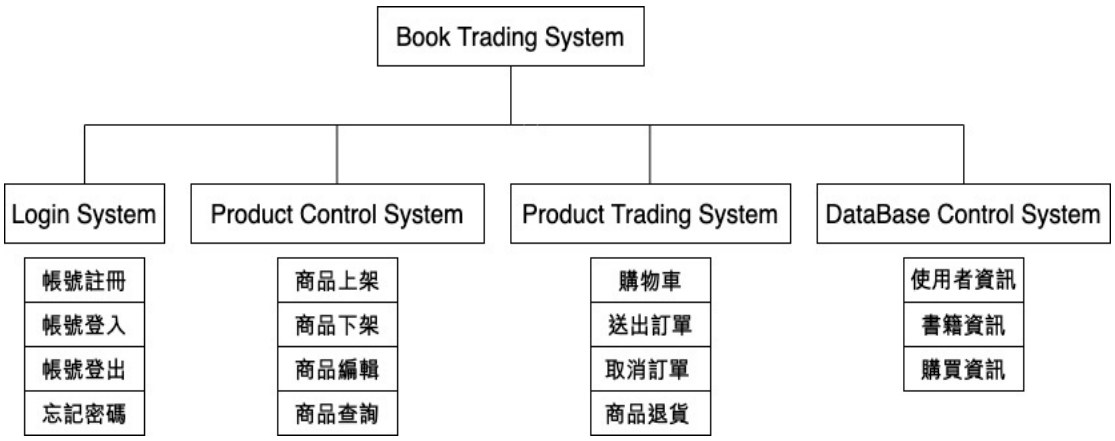
本二手書系統之範圍包括:

- 購買及退貨介面
- 商品簡介及庫存查詢
- 安全性防護

2.1.3 系統架構

使用者藉由簡單明瞭的操作介面進行操作，進行登入後使用者可以透過此系統來尋找自己想要的書，加入購物車時將購買資訊與資料進行比對，確認是否尚有庫存。使用者也可以上傳自己想要販賣的書籍，完整的達到書籍交流的目標。

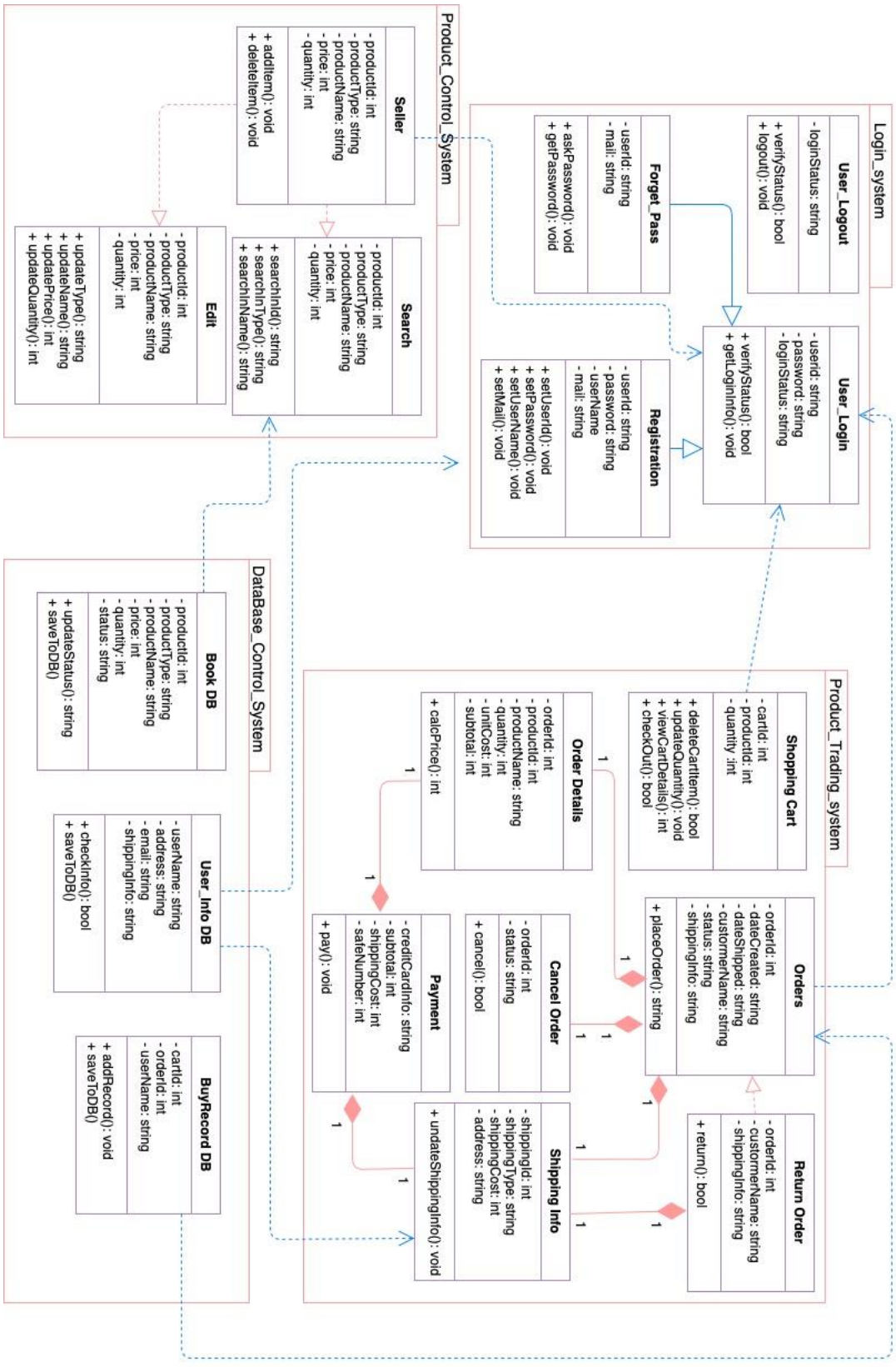
2.2 軟體架構



圖一、系統功能架構圖

- MENU
  - 使用目錄將資料分類，更快速的搜尋想要的書
    - 分類 button 因為是學校用的二手書交易系統將書分成各學院類別，以方便搜尋自己學院的書
    - 年級 button 再利用年級對學院的書進行更細的分類
    - 全部 button 若以上類別皆未找到想要的書，用全部尋找未經分類的書
- Search
  - 在 Search 輸入自己想要的書名
- Login
  - 跳轉登入介面
- 銷售畫面
  - 購物窗 有個產品的圖片，可以讓消費者對喜歡的產品下單
  - 詳細資料 button 點按查看產品詳細資料
- 加入購物車 button
  - 數量 button 透過加減選擇下單數量
  - 按下加入購物車 button，將產品加入購物車
  - 購物車讓我們可以購買產品

2.3 軟體類別定義



圖二、Package Diagram

2.4 軟體類別描述

Class User_Login
類別描述:
所有使用者皆需執行登入動作後才能進入系統首頁，使用本產品。
類別屬性:
➓ private string userId: 使用者帳號 ➓ private string password: 使用者密碼 ➓ private string loginStatus: 記錄使用者狀態，登入/登出
類別函式:
➓ public bool verifyStatus(string userId, string password): 驗證登入狀態 ➓ public bool getLoginInfo(string userId, string password): 取得輸入的帳號密碼訊息

Class User_Logout
類別描述:
使用者登出系統需執行此登出動作。
類別屬性:
➓ private string loginStatus: 記錄使用者狀態，登入/登出
類別函式:
➓ public bool verifyStatus(): 驗證登入狀態 ➓ public void logout(): 使用者登出

Class Registration
類別描述:
所有第一次使用本系統的使用者，皆需在首次使用前先執行註冊動作。
類別屬性:
➓ private string userId: 欲使用的使用者帳號 ➓ private string password: 欲使用的使用者密碼 ➓ private string userName: 欲使用的使用者姓名 ➓ private string mail: 使用者的合法信箱，作為系統與使用者溝通的媒介
類別函式:
➓ public void setUserId(string newUserId): 設定使用者帳號並檢查格式

➓ public void setPassword(string newPassword): 設定使用者密碼並檢查格式 ➓ public void setUserName(string newUserName): 設定使用者暱稱並檢查是否已被使用 ➓ public void setMail(string mail): 設定使用者電子郵件地址並檢查是否合法

Class Forget_Pass
類別描述:
忘記密碼，當使用者想尋回密碼時，使用此類別。
類別屬性:
➓ private string userId: 使用者帳號 ➓ private string mail: 使用者註冊時使用的合法信箱，用於系統傳回密碼資訊給使用者。
類別函式:
➓ public void askPassword(string userId): 要求取得密碼 ➓ public void getPassword(string userId): 取得密碼

Class Seller
類別描述:
使用者中欲販賣二手書籍的使用者，為賣家。此類別用於使用者上架及下架二手書籍。
類別屬性:
➓ private int productid: 商品編號 ➓ private string productType: 商品分類 ➓ private string productName: 商品名稱 ➓ private int price: 商品價錢 ➓ private int quantity: 商品數量
類別函式:
➓ public void addItem(string productType, string productName, int price, int quantity): 上架商品 ➓ public void deleteItem(int productid): 下架商品

Class Search
類別描述:
搜尋功能。可於搜尋欄輸入欲搜尋物品之相關資訊以快速獲取商品資訊。
類別屬性:
➓ private int productid: 商品編號 ➓ private string productType: 商品分類 ➓ private string productName: 商品名稱 ➓ private int price: 商品價錢 ➓ private int quantity: 商品數量
類別函式:
➓ public string searchInId(int productid): 輸入商品編號 ➓ public string searchInType(string productType): 輸入商品相關分類 ➓ public string searchInName(string productName): 輸入商品名稱關鍵字

Class Edit
類別描述:
使用者為賣家時，更新自己的販售商品資訊。
類別屬性:
➓ private int productid: 商品編號 ➓ private string productType: 商品分類 ➓ private string productName: 商品名稱 ➓ private int price: 商品價錢 ➓ private int quantity: 商品數量
類別函式:
➓ public string updateType(string productType): 更新商品分類 ➓ public string updateName(string productName): 更新商品名稱 ➓ public int updatePrice(int price): 更新商品價錢 ➓ public int updateQuantity(int quantity): 更新商品數量

Class Shopping Cart
類別描述:
此類別為購物車。使用者為買家時，將欲選購的二手書籍添加至此。購物車中能簡易的預覽商品資訊及修改已加入購物車的商品數量。確認訂單資訊後可送出訂單。
類別屬性:
➓ private int cartId: 購物車編號 ➓ private int productid: 欲購買之商品編號 ➓ private int quantity: 欲購買之商品數量
類別函式:
➓ public bool deleteCartItem(int productId): 移除已存於購物車的商品 ➓ public int updateQuantity(int productId, int quantity): 更新已存於購物車中欲購買之商品數量 ➓ public int viewCartDetails(): 確認購物車中已選取的商品之商品資訊（價錢、數量） ➓ public bool checkout(): 確認訂單內容後，送出訂單，進入結帳頁面

Class Orders
類別描述:
訂單資訊
類別屬性:
➓ private int orderId: 訂單編號 ➓ private int cartId: 購物車編號 ➓ private string dateCreated: 訂單成立日期時間 ➓ private string dateShipped: 訂單寄送日期時間 ➓ private string customerName: 收貨人姓名 ➓ private string status: 訂單狀態（訂單成立、備貨中、已出貨、已到貨、已領貨） ➓ private string shippingInfo: 訂單寄送資訊
類別函式:
➓ public string placeOrder(int cartId, string customerName, string shippingInfo): 確認收件人資訊及寄件資訊後，確認訂單，下訂單後紀錄訂單資訊。

Class Return Order
類別描述:
退貨。收到商品後，欲退還商品的使用者需使用此類別。
類別屬性:
❿ private int orderId: 欲退貨之訂單編號 ❿ private string customerName: 退貨人姓名 ❿ private string shippingInfo: 訂單退送資訊
類別函式:
❿ public bool return(int orderId): 退貨

Class Order Details
類別描述:
訂單詳細資訊
類別屬性:
❿ private int orderId: 訂單編號 ❿ private int productId: 訂單之商品編號 ❿ private string productName: 訂單之商品名稱 ❿ private int quantity: 訂單之商品數量 ❿ private int unitCost: 訂單中商品之商品單價 ❿ private int subtotal: 訂單之商品小計
類別函式:
❿ public int calcPrice(int unitCost): 計算商品總額

Class Cancel Order
類別描述:
取消訂單。商品尚未寄出前欲取消之訂單。
類別屬性:
❿ private int orderId: 欲取消之訂單編號 ❿ private string status: 欲取消之訂單當前狀態
類別函式:
❿ public bool cancel(int orderId): 取消訂單

Class Shipping Info
類別描述:
訂單寄送資訊
類別屬性:
➓ private int shippingId: 寄送編號 ➓ private string shippingType: 寄送類別（超商取貨、宅配、郵寄） ➓ private int shippingCost: 運費
類別函式:
➓ public void updateShippingInfo(int shippingId, string shippingType, int shippingCost): 更新寄送資訊

Class Payment
類別描述:
付款（本系統僅支持信用卡付款，不支持貨到付款、超商代碼繳費、轉帳等服務）。
類別屬性:
➓ private string creditCardInfo: 信用卡刷卡所需之資訊 ➓ private int subtotal: 訂單小計 ➓ private int shippingCost: 運費 ➓ private int safeNumber：安全碼
類別函式:
➓ public void pay(int credictCardID, int safeNumber, int subtotal): 付款

Class Book DB
類別描述:
系統賣場中的所有書籍。
類別屬性:
➓ private int productid: 商品編號 ➓ private string productType: 商品分類 ➓ private string productName: 商品名稱 ➓ private int price: 商品價錢 ➓ private int quantity: 商品數量 ➓ private string status: 商品狀態
類別函式:

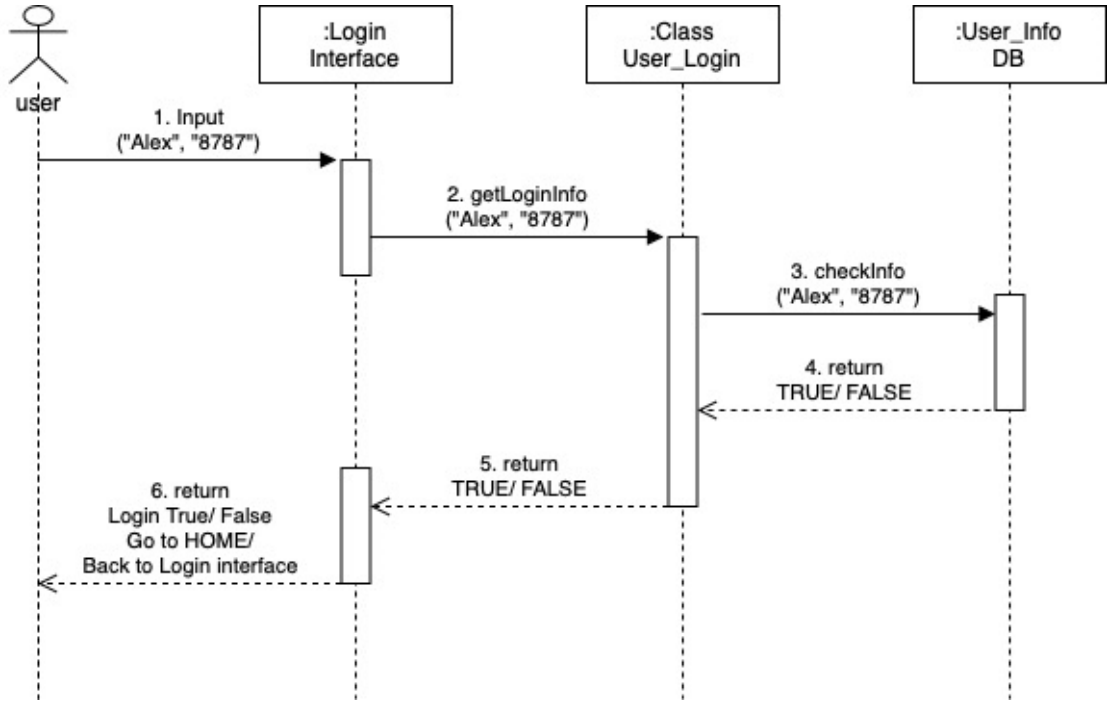
➓ public string updateStatus(int quantity): 更新商品資訊 ➓ public saveToDB(): 保存到資料庫

Class User_Info DB
類別描述:
本系統中所有使用者詳細資訊
類別屬性:
➓ private string customerName: 顧客暱稱 ➓ private string address: 住址 ➓ private string email: 電子郵件 ➓ private string shippingInfo: 寄送資訊
類別函式:
➓ public bool checkInfo(string userId, string password): 確認帳號密碼是否合法 ➓ public saveToDB(): 保存到資料庫

Class BuyRecord DB
類別描述:
紀錄每筆訂單資訊
類別屬性:
➓ private int cartId: 購物車編號 ➓ private int orderId: 訂單編號 ➓ private string userName: 購買人姓名
類別函式:
➓ public void addRecord(string userName, int cartId): 新增購買紀錄 ➓ public saveToDB(): 保存到資料庫

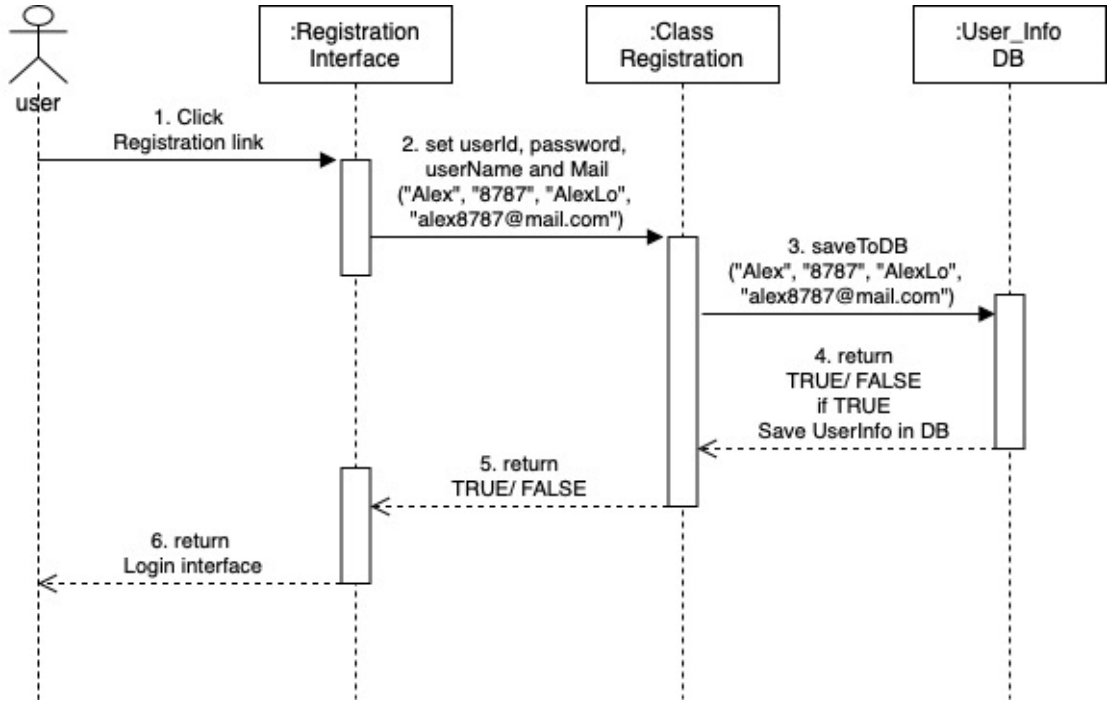
2.5 軟體行為概述

- 正常登入



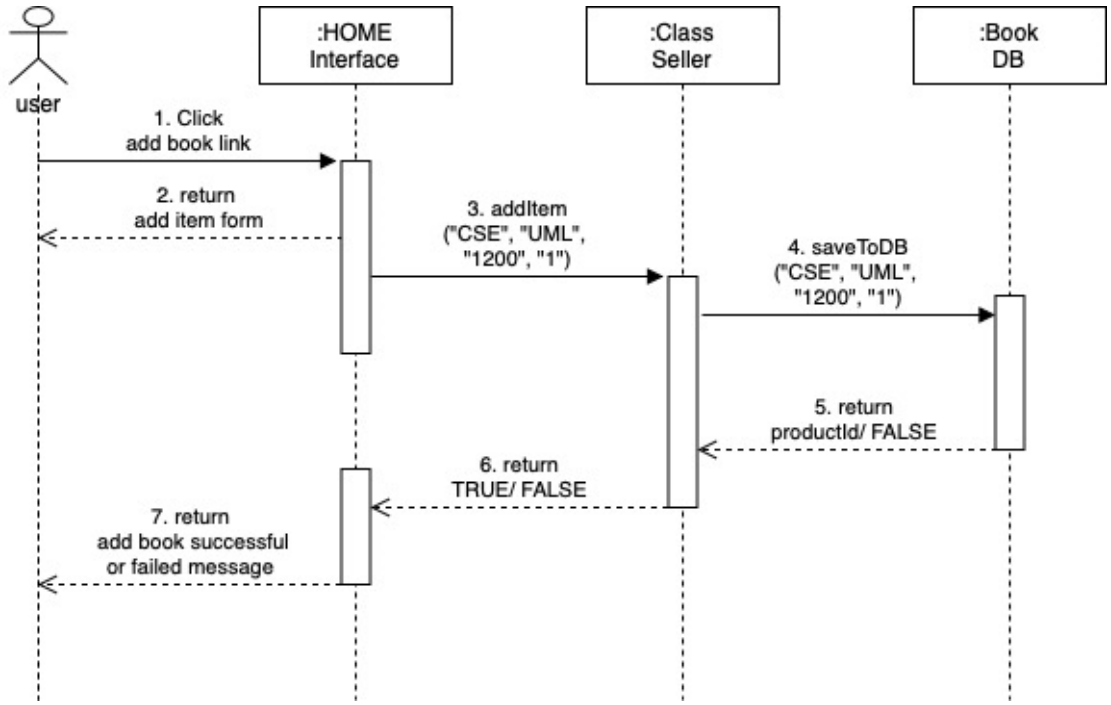
圖三、正常登入時序圖

- 註冊



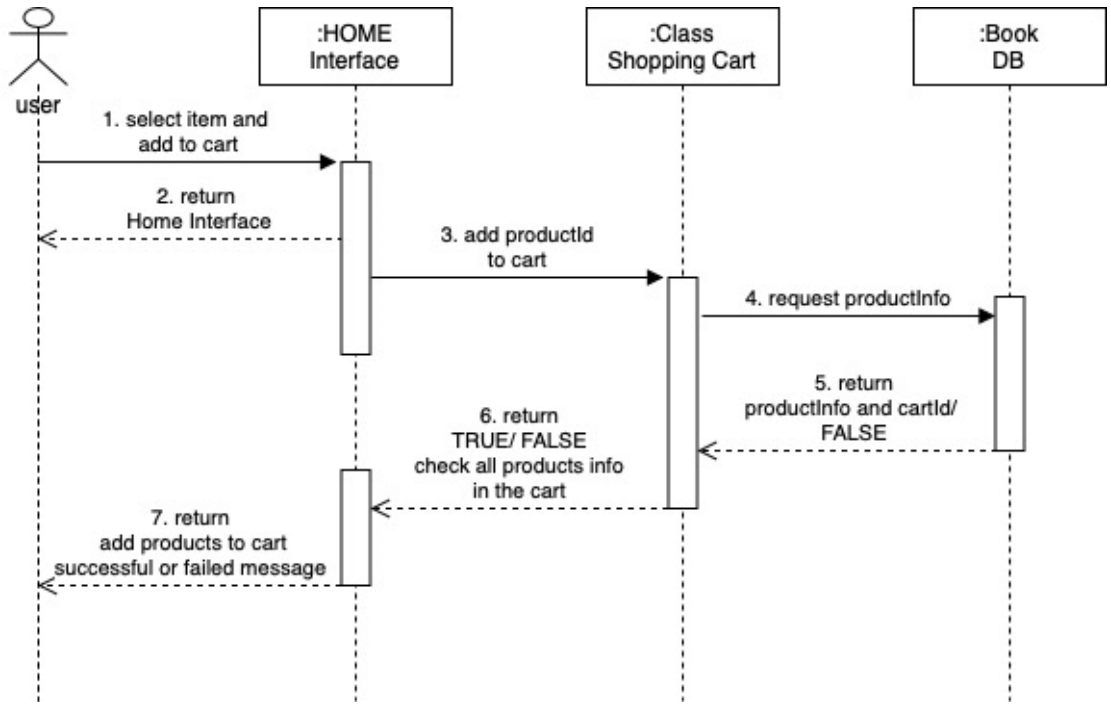
圖四、註冊時序圖

➢ 新增書籍



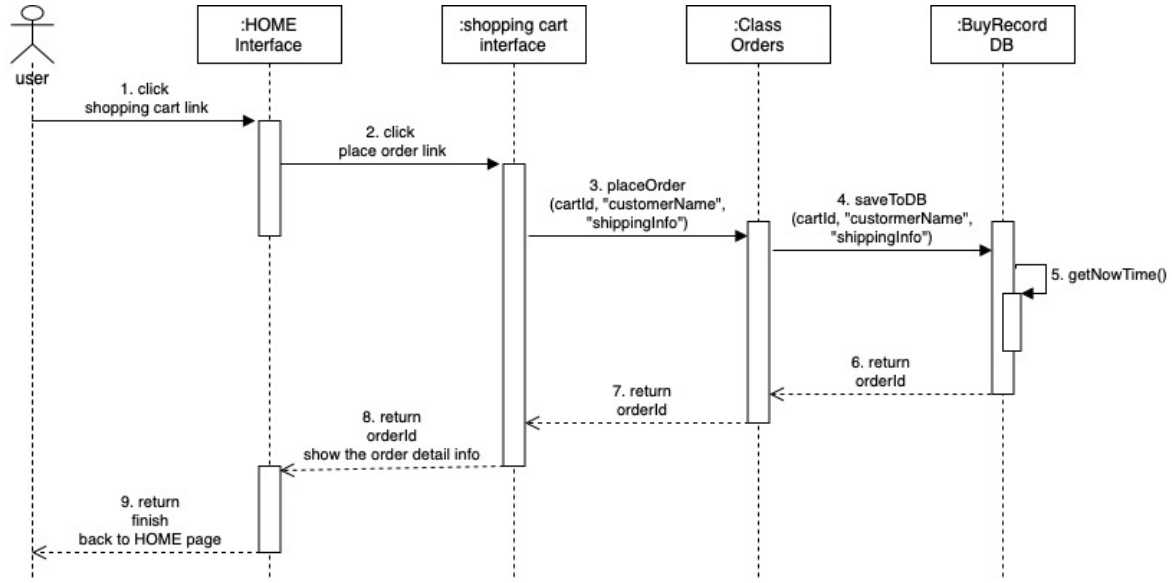
圖五、新增書籍時序圖

➢ 放入購物車

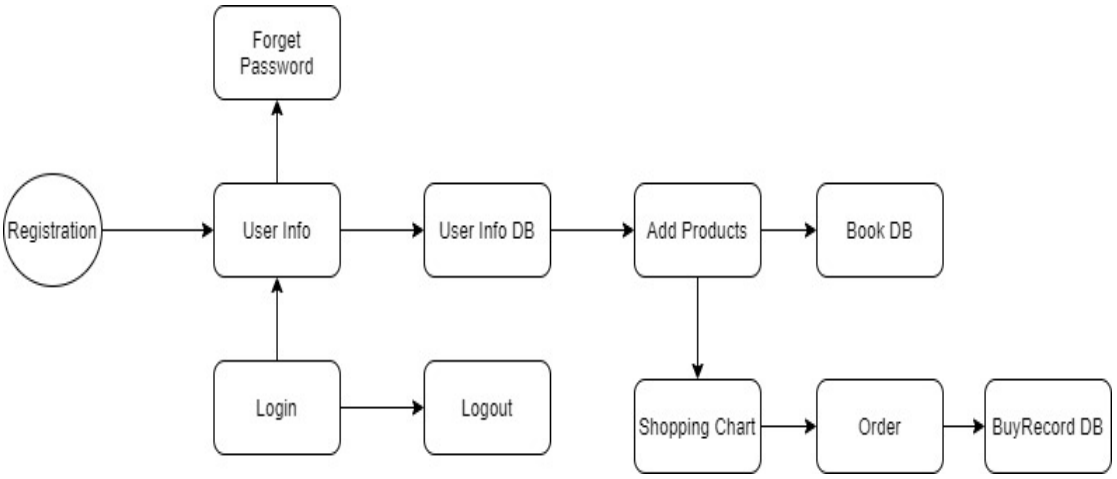


圖六、放入購物車時序圖

➢ 購買



圖七、購買時序圖



圖八、Client-Object Diagram

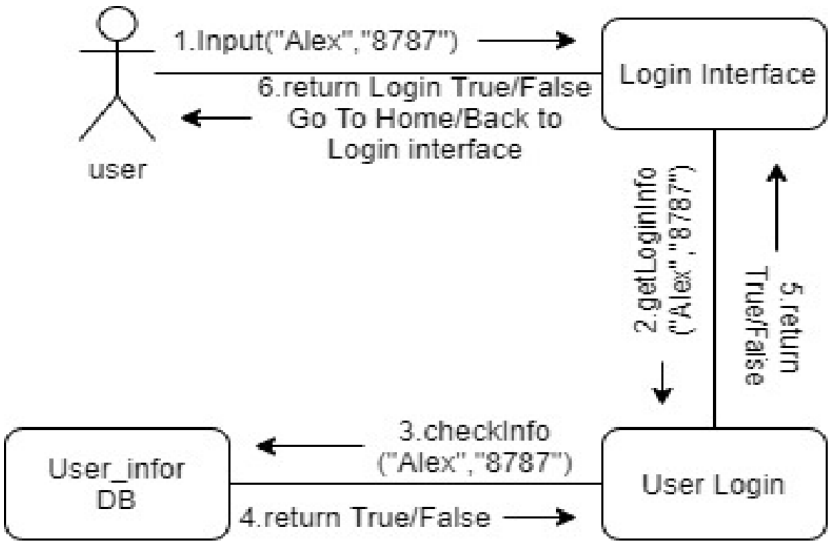
三、 軟體設計描述

3.1 設計概述

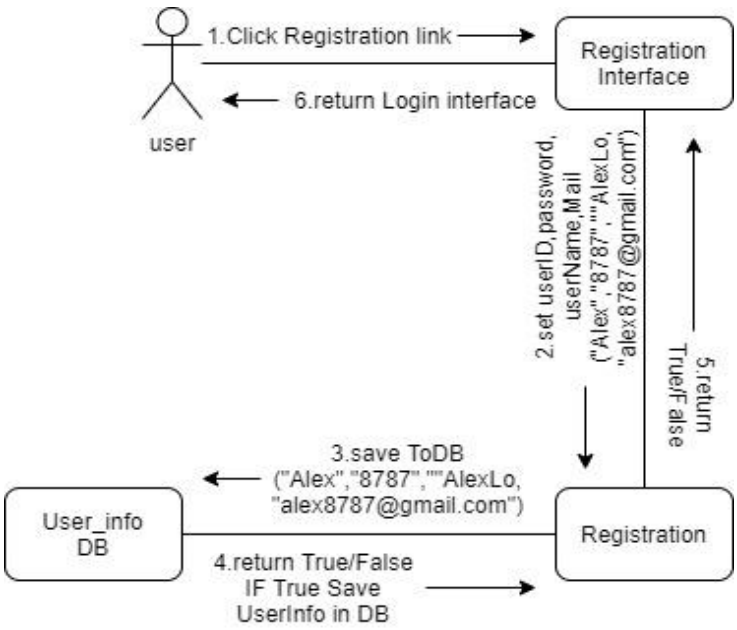
本節內容主要是系統之間的合作關係、各個子系統內部之類別合作關係及其類別相對應之虛擬碼等系統元件設計相關描述。

3.2 軟體類別設計描述

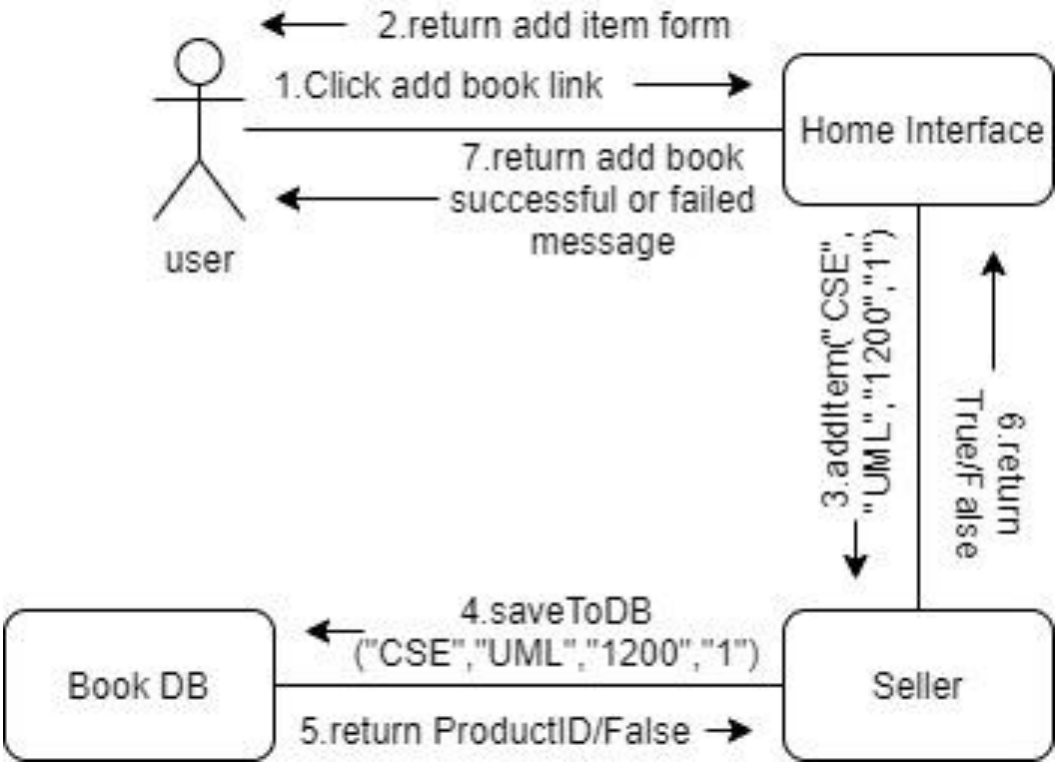
功能包括人員名單管理、會員帳號密碼管理、購買紀錄查詢、書籍上傳管理、購物車價錢計算、配送合作物流。



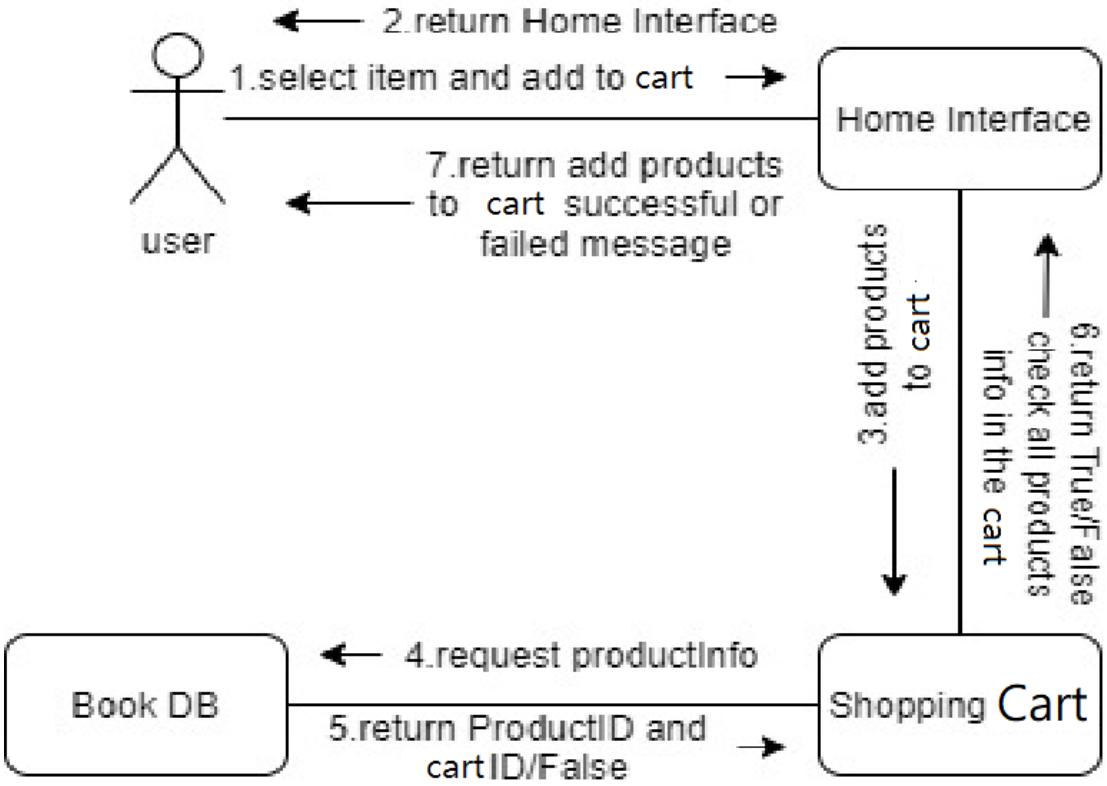
圖九、正常登入合作圖



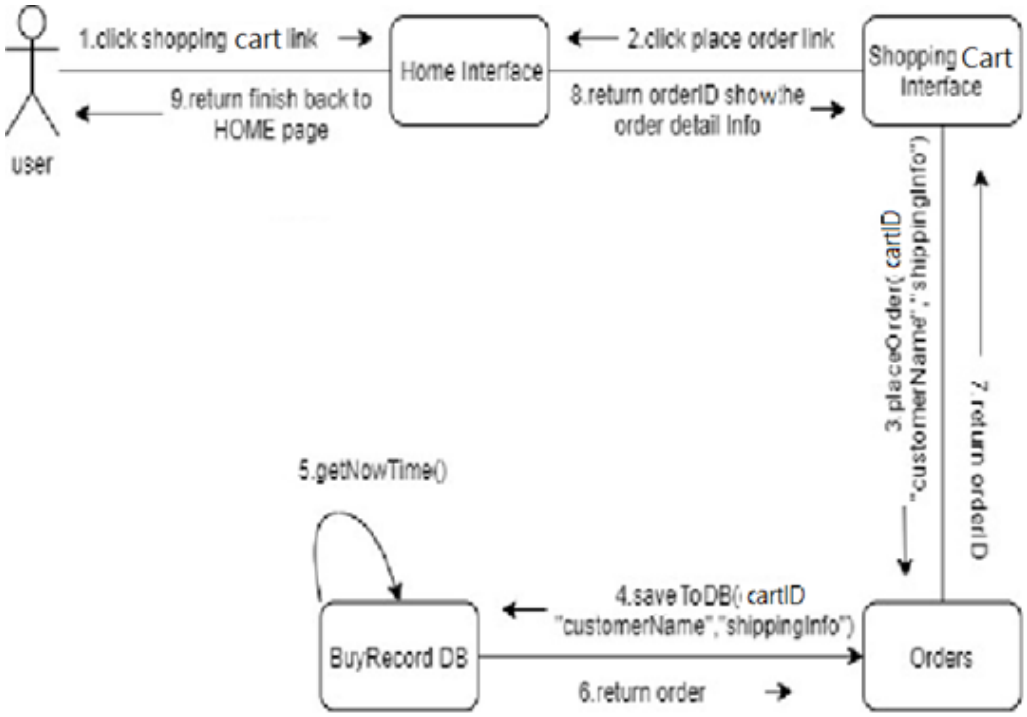
圖十、註冊合作圖



圖十一、新增書籍合作圖



圖十二、放入購物車合作圖



圖十三、購買合作圖

圖十四、Login system

- 帳號註冊
  - 說明：使用者註冊一組帳號，以使用此系統
  - 輸入：帳號、密碼、信箱
  - 處理：驗證帳號是否合法，若合法則將帳號資訊寫入資料庫中，反之則重新輸入
  - 輸出：註冊成功／失敗

■ 類別函式：

public void setUserId(string newUserId)

函式說明：設定使用者帳號並檢查格式

輸入參數：新使用者的 ID

輸出參數：無

虛擬碼：

```
public void setUserId(string newUserId)
{
    IF the UserId is already in database
        Show error message
    Else
        POST the newUserId in the database
        Show successive adding message
}
```

public void setPassword(string newPassword)

函式說明：設定使用者密碼並檢查格式

輸入參數：新使用者的密碼

輸出參數：無

虛擬碼：

```
public void setPassword(string newPassword)
{
    IF the newPassword is already in the right form
        Add the newPassword in the database
        Show successive adding message
    Else
        Show error message
}
```

public void setUserName(string newUserName)

函式說明：設定使用者名稱並檢查是否已被使用

輸入參數：新使用者的帳號名

輸出參數：無

虛擬碼：

```
public void setUserName(string newUserName)
{
    IF the newUserName is already in database
        Show error message
```

```
    Else
        POST the newUserName in the database
        Show successive adding message
}

public void setUserMail(string mail)
函式說明：設定使用者電子郵件地址並檢查是否合法
輸入參數：新使用者的 mail
輸出參數：無
虛擬碼：
public void setUserMail(string mail)
{
    IF mail is not in the right form
        Show error message
    ELSE IF mail is already in database
        Show error message
    ELSE
        Add mail to the database
        Show successive adding message
}
```

- 操作：

  在“登入介面”中選擇“帳號註冊”，使用者填入欲設定的個人資料後送出，系統確認資料合法後，將新的使用者資料存入資料庫中。

- 帳號登入
  - 說明：輸入帳號密碼後登入系統
  - 輸入：帳號、密碼
  - 處理：驗證帳號密碼是否正確，正確則跳轉至系統主頁，反之則重新輸入
  - 輸出：登入成功 / 失敗
  - 類別函式：

```
public bool verifyStatus(string userId, string password)
函式說明: 驗證登入狀態
輸入參數：使用者 ID 密碼
輸出參數：登入狀態
虛擬碼：
public bool verifyStatus(string userId, string password)
```

```
{
    IF getLoginInfo(string userId, string password) ==
    TRUE
        RETURN TRUE
    ELSE
        RETURN FALSE
}

public void getLoginInfo(string userId, string password)
```

函式說明: 取得輸入的帳號密碼訊息

輸入參數：使用者 ID 密碼

輸出參數：無

虛擬碼：

```
public void getLoginInfo(string userId, string password)
{
    IF userId is in the database
    IF checkInfo(userId,password ) == TRUE
        Show successive adding message
    ELSE
        Show error message
    ELSE
    Show error message
}
```

- 操作：

  已註冊過的使用者，在“登入介面”中輸入帳號及密碼，系統驗證正確後，進入銷售主畫面。

- 忘記密碼
  - 說明：忘記密碼
  - 輸入：申請帳號時綁定的電子信箱帳號
  - 處理：管理者接收到訊息，回傳使用者密碼至使用者信箱
  - 輸出：密碼已傳送
  - 類別函式：

```
public void askPassword(userID)
```

函式說明: 要求取得密碼

輸入參數：使用者 ID

輸出參數：無

虛擬碼：

```
Public void askPassword(string userId)
{
  Send verify email to the user
}

public void getPassword(string userId)
```
函式說明: 取得密碼
輸入參數：使用者 ID
輸出參數：無
虛擬碼：
```
public void getPassword(string userId)
{
  Send password email to the user
}
```

- 操作：

  在“登入介面”中選擇“忘記密碼”，輸入帳號及電子郵件後，系統驗證正確即回傳該使用者密碼。

- 帳號登出
  - 說明：登出系統
  - 輸入：點擊登出按鈕
  - 處理：跳轉回登入頁面
  - 輸出：登出成功 / 失敗
  - 類別函式：

```
public void logout()
```
函式說明: 使用者登出
輸入參數：無
輸出參數：無
虛擬碼：
```
public void logout()
{
    UserStatus = “LOGOUT”
}

public bool verifyStatus()
```
函式說明: 驗證登出狀態
輸入參數：無
輸出參數：登出狀態

虛擬碼：

```
public bool verifyStatus()
{
    IF UserStatus ==  “LOGOUT”
        Return TRUE
    ELSE
        Show error message
        Return FALSE
}
```

- 操作：

  已登入的使用者，在“銷售主畫面”中點選“登出按鈕”，即跳轉回登入介面。

圖十五、Product Control System

- 商品查詢
  - 說明：快速查詢欲尋找的商品或是在列表中選擇想要的分類
  - 輸入：商品品名 / 選取分類
  - 處理：於資料庫中提取出目標商品資訊
  - 輸出：跳轉至目標商品頁面
  - 類別函式：

    public string searchInId(int productid)

    函式說明：輸入商品編號

    輸入參數：商品編號

    輸出參數：商品 ID

    虛擬碼：

    public string searchInId(int productid)

```
{
    FOR i in ProductDocument
        IF i[ID] == productid
            RETURN i
}
```

```
public string searchInType(string productType)
函式說明：輸入商品相關分類
輸入參數：品相關分類
輸出參數：商品 ID
虛擬碼：

public string searchInType(string productType)
{
    FOR i in ProductDocument
        IF i[TYPE] == productType
            RETURN i
}

public string searchInName(string productName)
函式說明：輸入商品名稱關鍵字
輸入參數：商品名稱關鍵字
輸出參數：商品 ID
虛擬碼：
public string searchInName(string productName)
{
    FOR i in ProductDocument
        IF i[name] == productName
            RETURN i
}
```

- 操作：

  在 Search 欄中輸入自己想要的書名，即跳轉至與關鍵字相關聯之商品。

- 商品管理(上架 / 下架)
  - 說明：將需要販賣的商品上架 / 下架不再販賣的商品
  - 輸入：商品相關資訊 ex: 品名、價錢、數量等等 / 選取下架商品
  - 處理：將商品資訊存入資料庫中 / 將商品資訊從資料庫中移除

- 輸出：將商品資訊刊登在商品頁上 / 將商品從商品頁上移除
- 類別函式：

  public void addItem(string productType, string productName, int price, int quantity)

  函式說明：上架商品

  輸入參數：商品名稱、價錢、數量

  輸出參數：無

  虛擬碼：

```
public void addItem(string productType, string productName, int price, int quantity)
{
    Add Data to DataBase
}

public void deleteItem(int productid)
```

  函式說明：下架商品

  輸入參數：商品 ID

  輸出參數：無

  虛擬碼：

```
public void deleteItem(int productid)
{
    Delet Data in DataBase
}
```

- 操作：

  在“賣家專區”中“商品管理”處，點選“上架 button”並輸入商品資訊後，點選“新增 button”，即可上架商品。賣家於現有商品中點選“下架 button”，選取“確認下架 button”後，商品即下架。

➢ 商品編輯

- 說明：商品狀態發生改變時，用以編輯最新資訊
- 輸入：修改需更改的商品資訊
- 處理：將新的資訊取代舊的，放進資料庫中
- 輸出：更新商品資訊於商品頁上
- 類別函式：

  public string updateType(string productType)

  函式說明：賣家更新商品分類

  輸入參數：商品分類

輸出參數：新商品分類
虛擬碼：

```
public string updateType(string productType)
{
    RETURN productType
}

public string updateName(string productName)
```

函式說明：賣家更新商品名稱
輸入參數：商品名稱
輸出參數：新商品名稱
虛擬碼：

```
public string updateName (string productName)
{
    RETURN productName
}

public int updatePrice(int price)
```

函式說明：賣家更新商品價錢
輸入參數：商品價錢
輸出參數：新商品價錢
虛擬碼：

```
public int updatePrice(int price)
{
    RETURN price
}

public int updateQuantity(int quantity)
```

函式說明：賣家更新商品數量
輸入參數：商品數量
輸出參數：新商品數量
虛擬碼：

```
public int updateQuantity(int quantity)
{
    RETURN quantity
}
```

- 操作：

在“賣家專區”中“商品管理”處，於現有的架上商品中點選

“編輯商品 button”，即可編輯商品資訊。

圖十六、Product Trading System

- 購物車
  - 說明：存放買家所選取的商品，可編輯商品資訊及送出訂單
  - 輸入：點擊放入購物車，亦可於此修改選取的商品資訊，送出訂單
  - 處理：根據使用者選項，改變數量或從購物車中移除商品，將訂單資訊存入資料庫中
  - 輸出：更改選取的商品資訊
  - 類別函式：

```
public bool deleteCartItem(int productId)
函式說明：移除已存於購物車的商品
輸入參數：購物車的商品 ID
輸出參數：是否 delete 成功
虛擬碼：
public bool deleteCartItem(int productId)
{
    FOR i in CartitemList
        IF i[ID] == productId
            Delete i[ID]
            Return TRUE
}

public void updateQuantity(int productId, int quantity)
函式說明：更新已存於購物車中欲購買之商品數量
輸入參數：商品數量
輸出參數：無
```

虛擬碼：

```
public void updateQuantity(int productId, int quantity)
{
    FOR i in CartitemList
        IF i[ID] == productId
            I[QUANITY] = quantity
}

public int viewCartDetails()
```

函式說明：確認購物車中已選取的商品之商品資訊（價錢、數量）

輸入參數：無

輸出參數：總額

虛擬碼：

```
public int viewCartDetails()
{
    Int SUM = 0
    FOR i in CartitemList
        PRINT i
        SUM = SUM + i[PRICE]
    RETURN SUM
}

public bool checkout()
```

函式說明：確認訂單內容後，送出訂單，進入結帳頁面

輸入參數：無

輸出參數：訂單狀態

虛擬碼：

```
public bool checkout()
{
    RETURN TRUE
}
```

- 操作：
  - “數量 button”透過加減選擇下單數量。
  - 按下加入“購物車 button”，將產品加入購物車。
  - 購物車中也可增減商品數量，確認商品資訊。

- 送出訂單
  - 說明：確認購物車的商品資訊後，送出訂單
  - 輸入：於購物車下方點選送出訂單
  - 處理：將最終訂單資訊存入資料庫中
  - 輸出：訂單成立
  - 類別函式：

    public string placeOrder(int cartId, string customerName, string shippingInfo)

    函式說明：確認收件人資訊及寄件資訊後，接收訂單，下訂單後紀錄訂單資訊。

    輸入參數：收件人資訊及寄件資訊

    輸出參數：訂單編號

    虛擬碼：

```
public string placeOrder(int cartId, string customerName, string shippingInfo)
{
    RETURN orderId
}
```

    public int calcPrice(int unitCost)

    函式說明：計算商品總額

    輸入參數：運費

    輸出參數：總額

    虛擬碼：

```
public int calcPrice(int unitCost)
{
    Int SUM
    SUM = unitCost+viewCartDetails()
    Return SUM
}
```

    public void updateShippingInfo(int shippingId, string shippingType, int shippingCost)

    函式說明：更新寄送資訊

    輸入參數：新寄送資訊

    輸出參數：無

    虛擬碼：

    public void updateShippingInfo(int shippingId, string

```
shippingType, int shippingCost)
{
    SHIPPINGID = shippingId
    SHIPPINGTYPE = shippingType
    SHIPPINGCOST = shippingCost
}

public void pay(int credictCardID,int safeNumber, int
subtotal)
```

函式說明：付款

輸入參數：信用卡 ID,安全碼,交易金額

輸出參數：無

虛擬碼：

```
public void pay(int credictCardID, int safeNumber, int
subtotal)
{
    IF credictCardID and safeNumber is available
        Pay subtotal
    ELSE
        Show error message
}
```

- 操作：

  於購物車中，點選“送出訂單 button“即進入結帳畫面。

- 取消訂單
  - 說明：取消已送出但尚未出貨的商品訂單
  - 輸入：至訂單查詢頁面，選取取消訂單
  - 處理：將訂單狀態更改為取消，停止出貨流程
  - 輸出：取消訂單成功
  - 類別函式：

```
public bool cancel(int orderId)
```

函式說明：取消訂單

輸入參數：訂單 ID

輸出參數：訂單取消狀態

虛擬碼：

```
public bool cancel(int orderId)
{
    FOR i in ORDERLIST
```

```
        If i[ID] == orderId
            DELETE I
            RETURN TRUE
    }
```

- 操作：

  在“銷售主畫面”中“會員中心”的“訂單查詢”中，選擇已成立但尚未送出的訂單，點選“取消訂單 button”後，點選“確認取消訂單 button”即可取消此筆訂單。

- 商品退貨
  - 說明：退回已領取的商品
  - 輸入：至訂單查詢頁面，選取退貨
  - 處理：將訂單狀態更改為退還
  - 輸出：退還商品的流程資訊 ex: 退費流程、退還地址
  - 類別函式：

```
public bool return(int orderId)
說明：退貨
輸入參數：訂單 ID
輸出參數：訂單退還狀態
虛擬碼：
public bool return(int orderId)
{
    Find orderId in DataBase
    OrderId[STATUS] =  'RETURN'
    RETURN TRUE
}
```

  - 操作：

    在“銷售主畫面”中“會員中心”的“訂單查詢”中，選擇已成立並已送達的訂單，點選“商品退貨 button”後，點選“確認退貨 button”即可取消此筆訂單。

圖十七、Database Control System

- 使用者資訊
  - 說明：所有使用者的數據資料
  - 輸入：使用者名稱
  - 處理：於資料庫中提取出目標使用者資訊
  - 輸出：跳轉至使用者個人資料
  - 類別函式：

    public bool checkInfo(string userId, string password)

    函式說明: 確認帳號密碼是否合法

    輸入參數：使用者 ID 密碼

    輸出參數：是否合法

    虛擬碼：

```
Public bool checkInfo(string userId, string password)
{
    IF userId and password is same as the data in the
database
        Show successive message
        Return TRUE
    ELSE
        Show error message
        Return FALSE
}
```

  - 操作：

    使用者註冊或編輯個人資訊後，系統驗證正確後將數據存於使用者資料庫中。

- 書籍資訊
  - 說明：所有書籍商品的資訊
  - 輸入：商品資訊
  - 處理：於資料庫中提取出目標商品資訊
  - 輸出：印出及更新商品資訊
  - 類別函式：

    public string updateStatus(int quantity)

    函式說明：當商品賣出時，更新商品資訊

    輸入參數：商品資訊

    輸出參數：新商品資訊

    虛擬碼：

```
public string updateStatus(int quantity)
{
```

```
    RETURN quantity
  }
```

- 操作：

  使用者為賣家進行商品編輯動作後，系統驗證正確後將數據存於使用者資料庫中。

- 購買資訊
  - 說明：每筆訂單的購買資訊
  - 輸入：訂單資訊
  - 處理：於資料庫中提取出目標購買資訊
  - 輸出：印出購買資訊
  - 類別函式：

    public void addRecord(string userName, int cartId)

    函式說明：送出訂單時，新增此筆訂單記錄

    輸入參數：訂單資訊

    輸出參數：無

    虛擬碼：

```
public void addRecord(string userName, int cartId)
{
    USERNAME = username
    CARTID = cartId
}
```

  - 操作：

    使用者送出訂單後，系統驗證正確後將數據存於使用者資料庫中。

3.3 執行概念

如圖三~圖七(軟體行為概述)

3.4 需求追朔

OOA 需求	OOD 相關類別
操作介面便利及簡單	- Class user_Login - Class Shopping Cart
系統穩定	- Class User_Login - Class User_Logout - Class Payment
功能操作及介紹	Login_system: - Class Registration

	- Class Forget_Pass Product_Control_system: - Class Search - Class Edit Product_Trading_system - Class Orders - Class Return Order - Class Cancel Order
系統功能	DataBase_Control_system: - Class Book DB - Class User_Info DB - Class BuyRecord DB

3.5 重複使用性

每個子系統是各自獨立，只有相依性的關連，故將其子系統個別套用於其他系統上仍然可以運作，故其重複使用性較高。

四、軟體介面設計描述

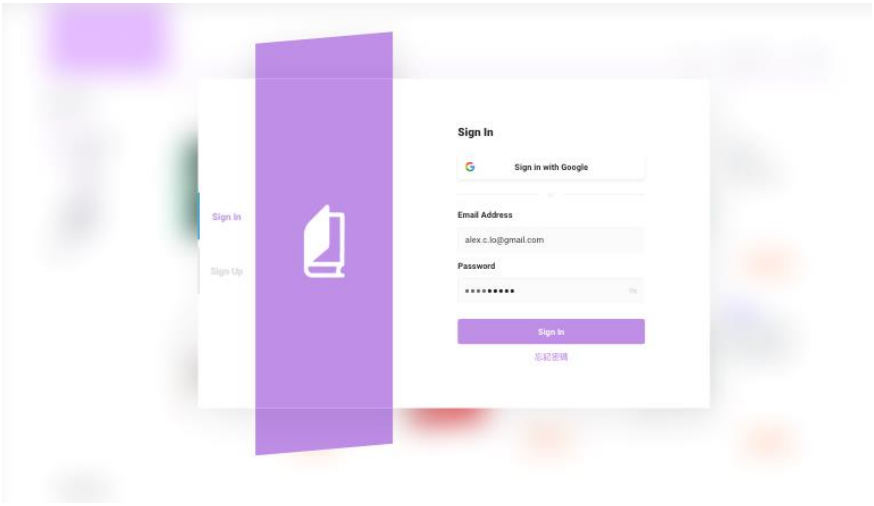
4.1 介面設計概述

本節主要描述主要控管軟體之外部介面、系統之函式介面與系統之物件介面等三部分。

4.2 外部介面定義

- 系統平台
  - 硬體平台 : 本軟體設定於支援 html 之硬體平台上執行
  - 作業系統 : 支援 Microsoft Windows 系列與 Mac OS
- 主畫面
  - 硬體平台 : 本軟體設定於 PC 之硬體平台上執行
  - 本軟體之視窗規劃如下圖所示 <使用者介面 與 開發者介面>
  - 視窗標題名稱設為 BookBook
  - 本軟體執行於 PC 環境 : 螢幕 1600x1200 解析度下，本軟體之視窗範圍不可超出螢幕顯示範圍

軟體視窗規劃如下圖



圖十八、登入介面

登入會員、註冊新會員、忘記密碼按鈕。

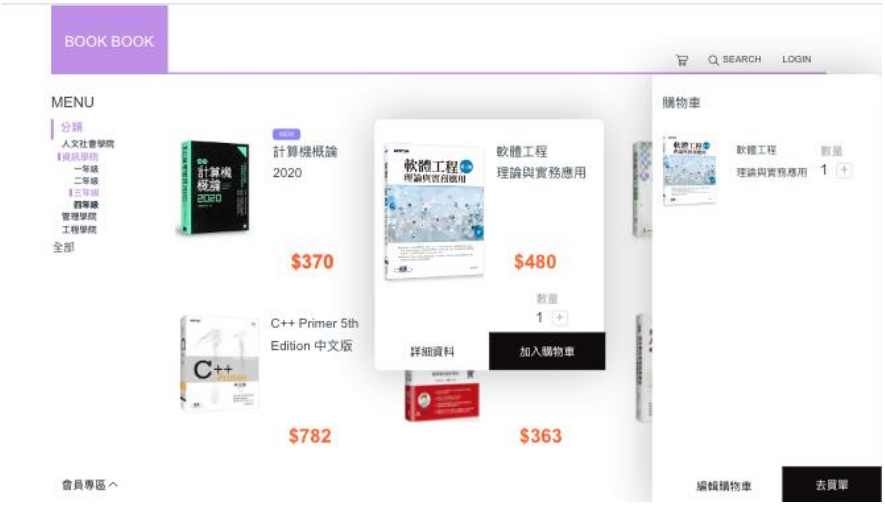
圖十九、銷售主介面

滑鼠移至書本上方，可檢視商品概略資訊，可快速選擇數量後加入購物車，也可點選詳細資料 button，檢視更詳細的商品資訊。

也有搜尋書本、檢視購物車的功能

圖二十、商品詳細內容介面

顯示該商品詳細資訊介面，可於此選擇商品數量後加入購物車。



圖二十一、購物車介面

將商品計算總數量、金額，並進行下單、結帳。

4.3 函示介面定義表

Class (User_Login)
- public bool verifyStatus(string userId, string password) - public void getLoginInfo(string userId, string password)
Class (User_Logout)
- public bool verifyStatus() - public void logout()
Class (Registration)
- public void setUserId(string newUserId) - public void setPassword(string newPassword) - public void setUserName(string newUserName)

- public void setMail(string mail)
Class (Forget_Pass)
- public void askPassword(string userId) - public void getPassword(string userId)
Class (Seller)
- public void addItem(string productType, string productName, int price, int quantity) - public void deleteItem(int productid)
Class (Search)
- public string searchInId(int productid) - public string searchInType(string productType) - public string searchInName(string productName)
Class (Edit)
- public string updateType(string productType) - public string updateName(string productName) - public int updatePrice(int price) - public int updateQuantity(int quantity)
Class (Shopping Cart)
- public bool deleteCartItem(int productId) - public void updateQuantity(int productId, int quantity) - public int viewCartDetails() - public bool checkout()
Class (Orders)
- public string placeOrder(int cartId, string customerName, string shippingInfo)
Class (Return Order)
- public bool return(int orderId)
Class (Order Details)
- public int calcPrice(int unitCost)
Class (Cancel Order)
- public bool cancel(int orderId)
Class (Shipping Info)
- public void updateShippingInfo(int shippingId, string shippingType, int shippingCost)
Class (Payment)
- public void pay(int credictCardID, int safeNumber, int subtotal)

Class (Book DB)
- public string updateStatus(int quantity) - public saveToDB()
Class (User_Info DB)
- public bool checkInfo(string userId, string password) - public saveToDB()
Class (BuyRecord DB)
- public void addRecord(string userName, int cartId) - public saveToDB()

4.4 類別介面定義

如圖十四～十七所示