

# 一、簡介

## 1.1 一般描述資訊

- a.議題日期和狀態：此版於 2023/05/03 尚在討論階段未完成。
- b.範圍：本規格書的內容以遊戲帳號交易系統為主
- c.議題機構：各遊戲玩家或對其遊戲有興趣的人所建立之系統
- d.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。
- e.規格書修改歷史：此版於 2023/05/10 為第 2 版

# 二、規格書內容敘述

## 2.1 系統概述

### 2.1.1 系統目標

本系統的主要目標是建立一個遊戲帳號交易和估價的線上平台，使用者可以在此平台上買賣遊戲帳號，也可以使用我們的估價服務來估計帳號的價值。

### 2.1.2 系統範圍

系統將涵蓋帳號註冊、登錄、買賣遊戲帳號、帳號估價（包括系統估價和公眾估價）等功能。目前，我們專注於處理遊戲帳號的買賣交易和估價，但未來可能擴展至其他相關服務。

### 2.1.3 系統架構

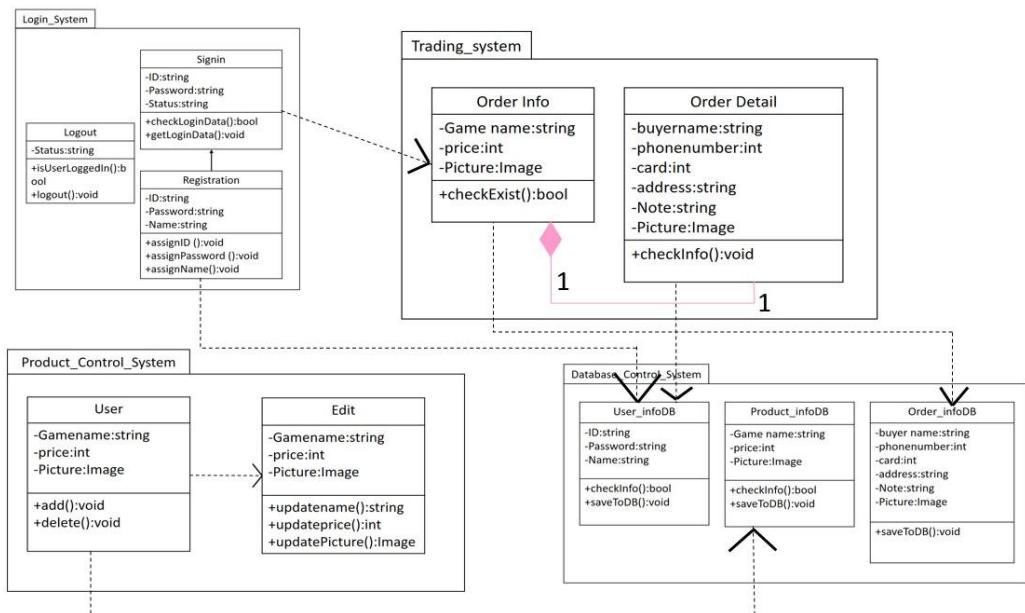
系統將採用客戶端-伺服器架構。伺服器端將負責處理用戶的請求和資料儲存，而客戶端則將提供用戶介面，讓使用者與系統互動。

## 2.2 軟體架構



圖一、系統功能架構圖

## 2.3 軟體類別別定義



圖二、系統功能架構圖

## 2.4 軟體類別描述

Class SignIn
類別描述: 此類別為處理使用者登入流程，只有成功登入後使用者才能進入並使用本系統。
類別屬性: <b>private string ID:</b> 儲存使用者的帳號資訊 <b>private string Password:</b> 儲存使用者的密碼資訊 <b>private string Status:</b> 用以追蹤使用者當前的狀態 (已登入/已登出)
類別函式: <b>public bool checkLoginData(string ID, string password):</b> 檢查使用者的帳號密碼是否正確 <b>public bool getLoginData(string ID, string password):</b> 獲取並確認使用者輸入的帳號密碼資訊

Class Logout
類別描述: 該類別負責處理使用者登出的流程。
類別屬性: <b>private string status:</b> 用以追蹤使用者當前的狀態 (已登入/已登出)
類別函式: <b>public bool isUserLoggedIn():</b> 用以確認使用者是否仍處於登入狀態 <b>public void Logout():</b> 負責進行使用者的登出程序
Class Registration
類別描述: 此類別負責管理首次使用本系統的使用者的註冊流程。
類別屬性: <b>private string ID:</b> 欲使用的帳號 <b>private string password:</b> 欲使用的密碼 <b>private string Name:</b> 欲使用的名稱
類別函式:

```
public void assignID(string ID): 設定帳號並檢查其格式  
public void assignPassword(string password): 設定密碼並檢查其格式  
public void assignName(string Name): 設定新的使用者名稱並檢查其是否已被  
其他使用者使用
```

### Class User

#### 類別描述:

此類別適用於希望出售/購買遊戲帳號的使用者，負責管理遊戲帳號的上架和下架以及檢視想購買的商品。

#### 類別屬性:

private string Gamename: 遊戲帳號的名稱

private int Price: 遊戲帳號的價格

private img Picture: 遊戲帳號的細圖

#### 類別函式:

public void add(string Gamename, string Price, int accountPrice, img Picture): 上架遊戲帳號

public void delete(string Gamename, string Price, int accountPrice, img Picture): 下架遊戲帳號

### Class Edit

#### 類別描述:

此類別適用於希望出售/購買遊戲帳號的使用者，負責更新已上架更改價格或其他資訊。

#### 類別屬性:

private string Gamename: 遊戲帳號的名稱

private int Price: 遊戲帳號的價格 private img

Picture: 遊戲帳號的細圖

#### 類別函式:

public void updatename(string Gamename): 更改遊戲帳號的名稱

public void updatePrice(int Price): 更改遊戲帳號的價格

public void updatePicture(img Picture): 更改遊戲帳號的細圖

### Class OrderInfo

#### 類別描述:

此類別適用於遊戲帳號的訂單資訊

類別屬性:

`private string Gamename`: 遊戲帳號的名稱

`private int Price`: 遊戲帳號的價格

`private img Picture`: 遊戲帳號的細圖

類別函式:

`public bool checkExist(string Gamename)`: 確認帳號是否尚未被下架或購買

### Class OrderDetail

類別描述:

此類別為買家訂單資訊

類別屬性:

`private string buyername`: 買家名稱

`private int phonenum`: 買家手機號

`private int card`: 買家信用卡卡號 `private`

`string Note`: 買家對賣家的備註 `private`

`string address`: 買家地址 `private img`

`Picture`: 遊戲帳號的細圖

類別函式:

`public bool checkInfo(string buyername)`: 檢查買家資訊是否完整正確

### Class User\_infoDB

類別描述:

此類別為管理用戶資訊的資料庫

類別屬性:

`private string ID`: 用戶的帳號 `private`

`string password`: 用戶的密碼 `private`

`string Name`: 用戶的名稱

類別函式:

`public bool checkInfo(string ID, string password, string Name)`: 檢查用戶資訊是否完整正確

### Class Product\_infoDB

類別描述:

此類別為管理用戶資訊的資料庫

類別屬性:

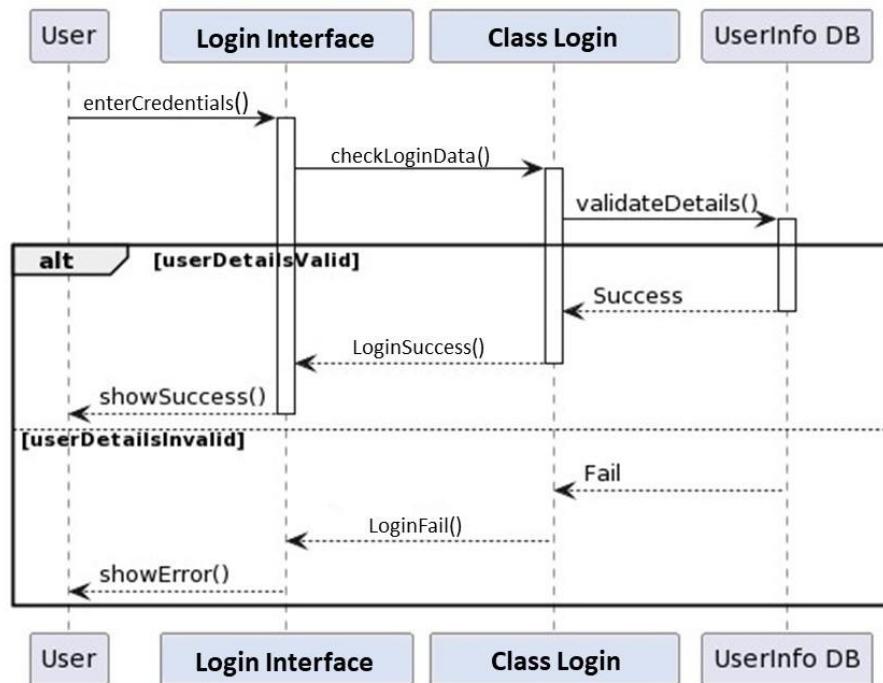
```
private string Gamename: 遊戲帳號的名稱  
private int Price: 遊戲帳號的價格  
private img Picture: 遊戲帳號的細圖
```

類別函式：

```
public bool checkInfo(string Gamename, int Price, img Picture):檢查帳號  
資訊是否完整正確
```

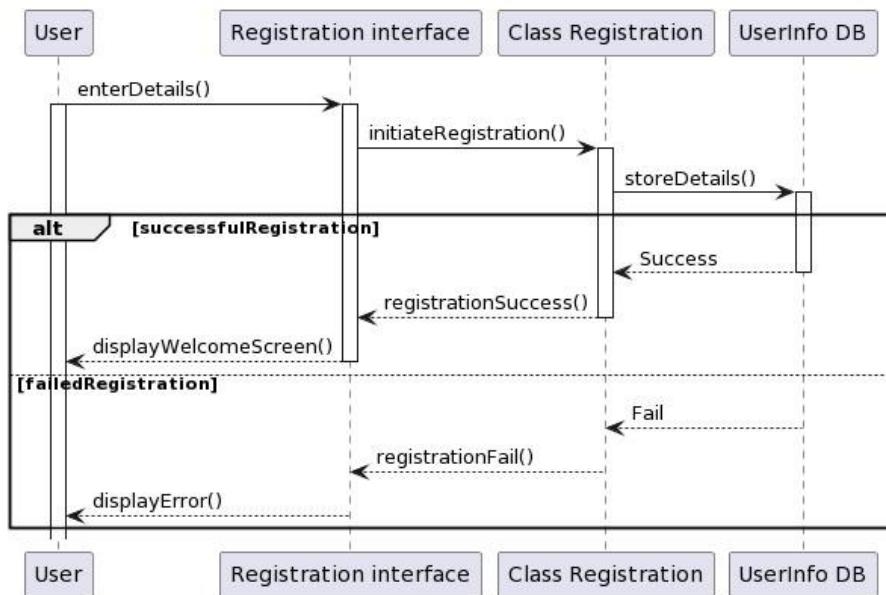
## 2.5 軟體行為概述

- 正常登入



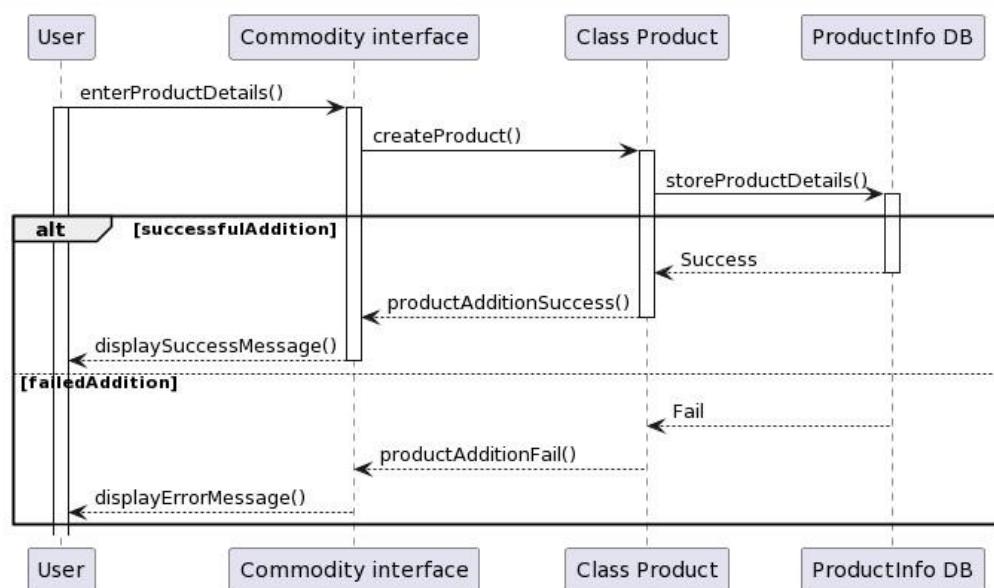
圖三、正常登入時序圖

- 註冊



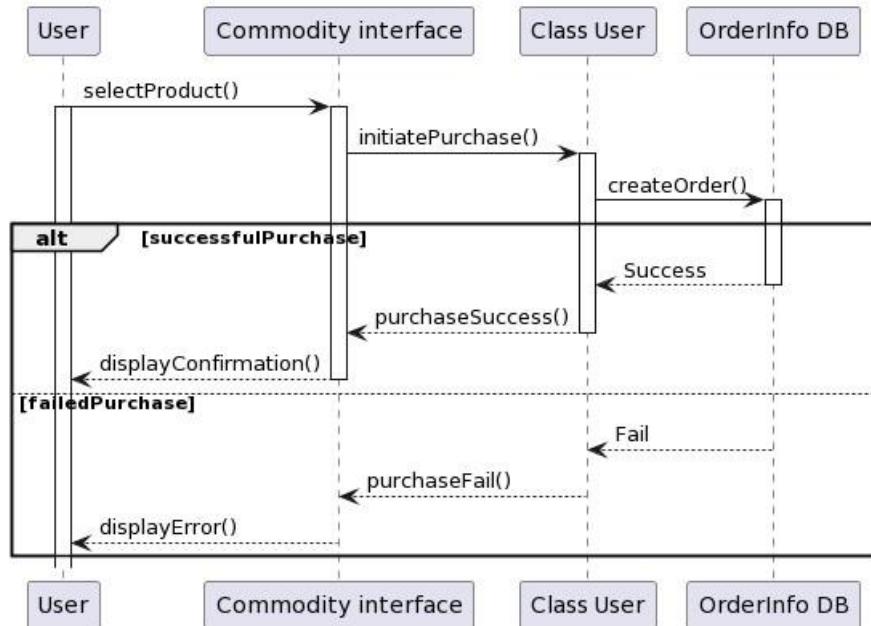
圖四、註冊時序圖

- 新增商品



圖五、新增遊戲帳號時序圖

- 購買



圖六、購買時序圖

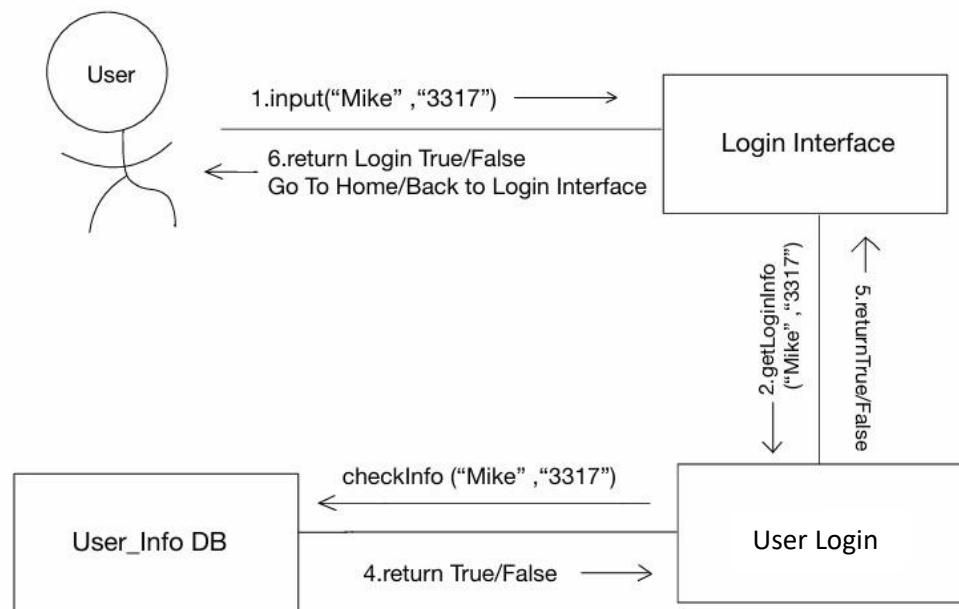
### 三、軟體設計描述

#### 3.1 設計概述

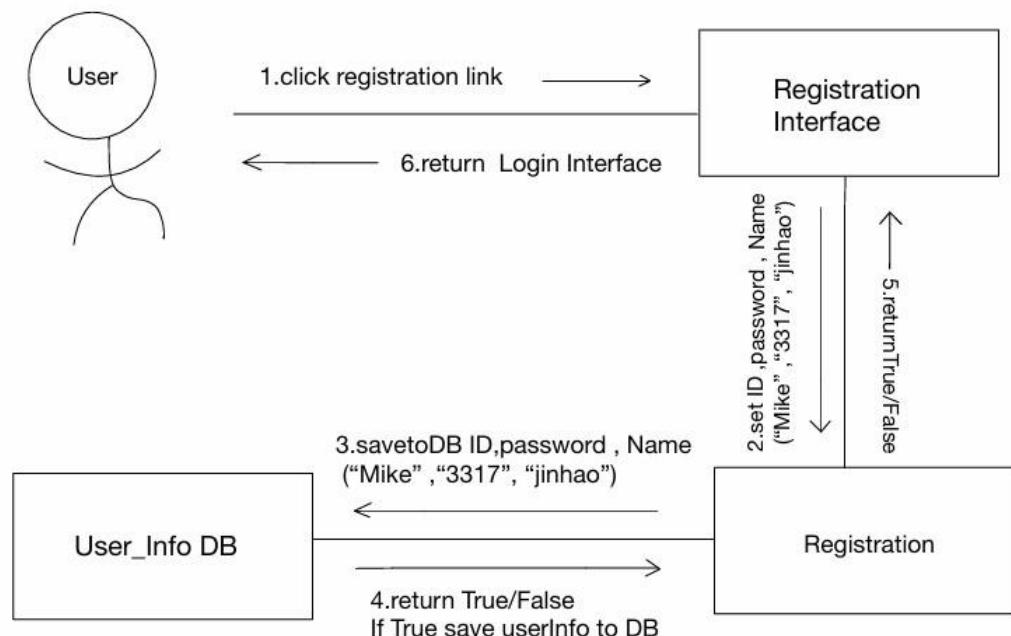
本節內容主要是系統之間的合作關係、各個子系統內部之類別合作關係及其類別相對應之虛擬碼等系統元件設計相關描述。

#### 3.2 軟體類別設計描述

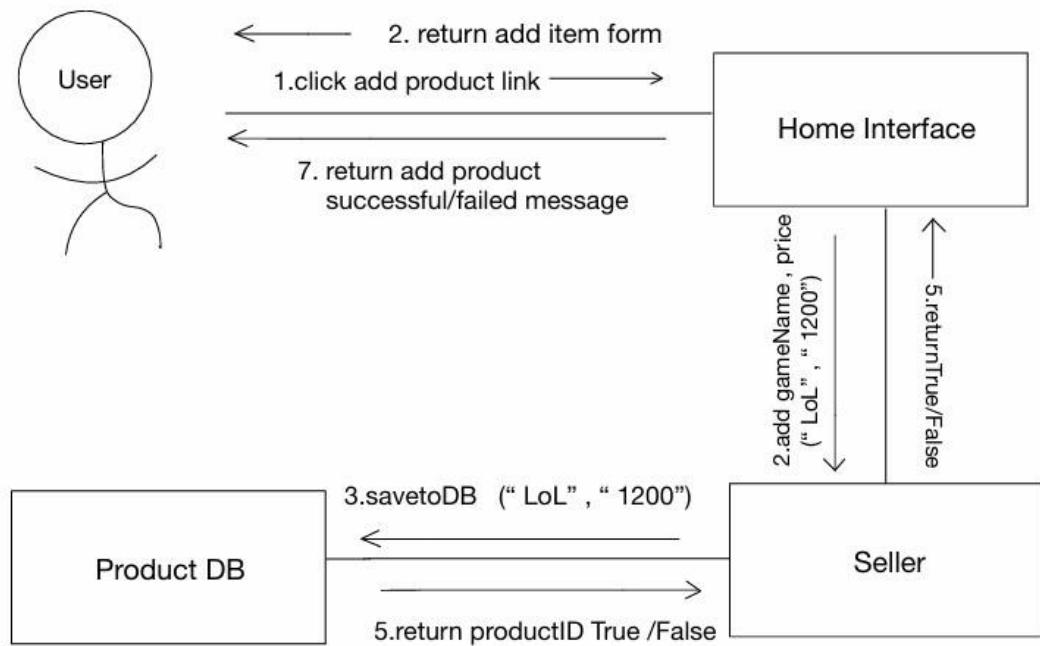
功能包括人員名單管理、會員帳號密碼管理、商品上傳管理、訂單購買管理



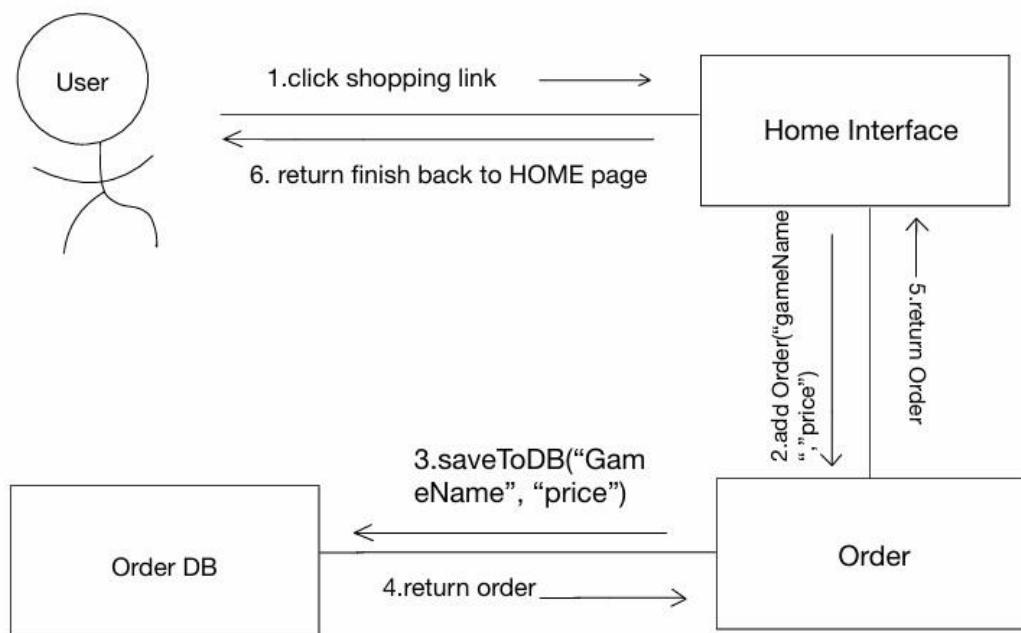
圖七、正常登入合作圖



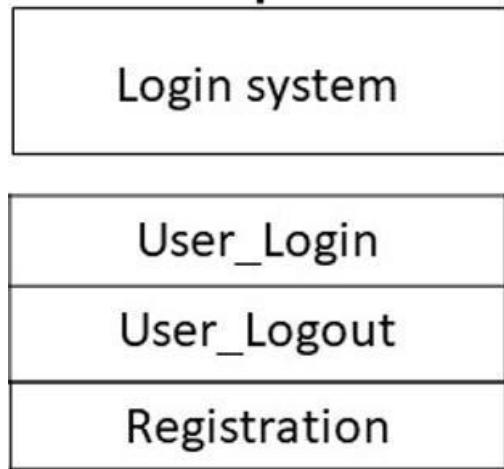
圖八、註冊合作圖



圖九、新增商品合作圖



圖十、下單合作圖



圖十一、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

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 assignID(string ID)

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 assignID(string ID) {  
    IF the ID is already in database Show  
        error message  
    Else  
        POST the ID in the database  
        Show successive adding message  
}
```

public void assignPassword(string Password)

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 assignPassword(string Password){  
    IF the Password is already in the right form  
        Add the Password in the database  
        Show successive adding message  
    Else  
        Show error message  
}
```

### public void assignName(string Name)

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 assignName(string Name)

{

    IF the Name is already in database Show

        error message

    Else

        POST the Name in the database

        Show successive adding message

}

## 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 checkLoginData(string ID, 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 checkLoginData (string ID, string password)

{

    IF getLoginData(string ID, string password) == TRUE

        RETURN TRUE

    ELSE

        RETURN FALSE

}

### public void getLoginData (string ID, 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 getLoginData(string ID, string password)
```

```
{
```

```
    IF ID is in the database
```

```
        IF checkInfo(ID,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。

### 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()
```

```
{
```

```
    Status = "LOGOUT"
```

```
}
```

public bool IsUserLoggedIn()

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 IsUserLoggedIn()

{

    IF Status == "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。



圖十二、Product Control System

## 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 add(string Gamename ,Price, img Picture)

函式說明：上架商品

輸入參數：商品名稱、價錢、商品圖

輸出參數：無

虛擬碼：

`public void add(string Gamename ,int Price, img Picture)`

{

Add Data to DataBase

}

public void delete(string Gamename)

函式說明：下架商品

輸入參數：商品名稱

輸出參數：無

虛擬碼：

`public void delete(string productID)`

{

Delete Data in DataBase

}

操作：在“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。

## 編輯價格

說明：商品價格發生改變時，用以編輯最新資訊

輸入：修改需更改的商品價格

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 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

}

操作：在更改價格處，打上欲更改的遊戲名，即可編輯商品價格。



圖十三、Trading System

## 訂單資訊

說明：將需要販賣的商品上架／下架不再販賣的商品

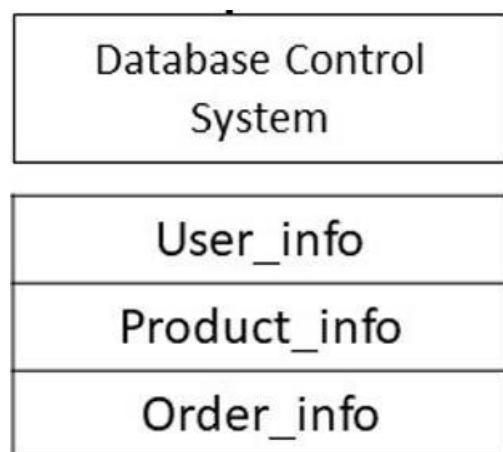
輸入：商品相關資訊 ex: 品名、價錢、數量等等／選取下架商品

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：無

## 訂單細節



圖十四、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 ID, 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 ID, string password)

{

    IF ID 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。

## 商品資訊

說明：所有遊戲帳號商品的資訊

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 bool checkInfo(string ID, 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 ID, string password)

{

    IF ID 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。

## 訂單資訊

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~圖六(軟體行為概述)

### 3.4 需求追溯

OOA 需求	OOD 相關類別
操作介面便利及簡單	- Class user_Login
系統穩定	- Class User_Login - Class User_Logout
功能操作及介紹	Login_system: - Class Registration Product_Control_system: - Class Delete - Class Add - Class Edit Product_Trading_system - Class Orders_Info - Class Order_Detail
系統功能	DataBase_Control_system: - Class Product_Info_DB - Class User_Info_DB - Class Order_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

本軟體執行於 PC 環境：

螢幕 1920x1080 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



圖十五、登入介面

## 熱門遊戲帳號



圖十六、銷售主介面

## 管理你的商品

商品名稱

預計售價

上傳您的圖片:

沒有選擇檔案

圖十七、商品管理介面

## 訂購人資訊

## 付款資訊

## 選擇付款方式

- 信用卡
- 行動支付
- 銀行轉帳

送出訂單

圖十八、商品購買介面

### 4.3 函示介面定義表

Class SignIn

```
public bool checkLoginData(string ID, string password)  
public void getLoginData(string ID, string password)
```

Class Logout

```
public bool isUserLoggedIn()  
public void Logout()
```

Class Registration

```
public void assignID(string ID)  
public void assignPassword(string password)  
public void assignName(string Name)
```

Class User

```
public void add(string Gamename, string Price, int accountPrice, img Picture)  
public void delete(string Gamename, string Price, int accountPrice, img Picture)
```

Class Edit

```
public void updatename(string Gamename)  
public void updatePrice(int Price)  
public void updatePicture(img Picture)
```

Class OrderInfo

```
public bool checkExist(string Gamename)
```

Class OrderDetail

```
public bool checkInfo(string buyername)
```

Class User\_infoDB

```
public bool checkInfo(string ID, string password, string Name)
```

Class Product\_infoDB

```
public bool checkInfo(string Gamename, int Price, img Picture)
```

#### 4.4 類別介面定義

如圖十一～十四 所示