

線上購物系統

Online Shopping System

開發成員：第 22 組 - 唐于萍、黃子庭、吳芷芳

2019/5/1

目錄

一、 規格書內容描述.....	4
二、 系統架構描述.....	4
2.1、 系統概述.....	4
2.2、 軟體架構.....	5
2.3、 軟體類別定義與描述.....	6
2.4、 軟體行為描述.....	10
三、 軟體介面設計描述.....	10
3.1、 介面設計概述.....	10
3.2、 外部介面定義.....	10
3.3、 類別介面定義.....	16
四、 軟體設計描述.....	16
4.1、 設計概述.....	16
4.2、 軟體功能描述.....	16
4.3、 軟體行為描述.....	29
4.4、 軟體設計描述.....	34
4.5、 需求追溯.....	37
4.6、 重複使用性.....	37
五、 規格內容追溯.....	37

圖目錄

圖一、 軟體架構設計圖(package diagrams).....	5
圖二、 系統基本架構示意圖.....	5
圖三、 系統功能架構圖.....	6
圖四、 Class Diagram.....	6
圖五、 Detailed Class Diagram.....	6-10
圖六、 Client-object Diagram.....	10
圖七、 使用者介面示意圖.....	11-15
圖八、 類別介面定義示意圖.....	16
圖九、 Sequence Diagram.....	29-33
圖十、 Collaboration Diagram.....	34-36

一、規格書內容描述

1.1 一般描述資訊

發布日期和狀態：此版於 2019/05/01 發布

範圍：本規格書的內容以線上購物網站系統之操作介面等為主。

議題機構：欲發展網路購物之商家所需之系統。

參考：IEEE/EIA 12207.1-1997

內容：本線上購物系統之標準規格與介面設計。

主體：包含有系統架構描述、軟體介面設計描述、軟體設計描述，和規格內容追朔等。

二、系統架構描述

2.1 系統概述

2.1.1 系統目標

- ◆ 提供使用者線上購買商品之平台
- ◆ 人性化使用者介面
- ◆ 能在兩秒內搜尋並取得商品資訊
- ◆ 商品資訊更新與維護
- ◆ 資料庫管理維護

2.1.2 系統範圍

- ◆ 線上型錄瀏覽
- ◆ 商品搜尋
- ◆ 線上訂購商品
- ◆ 訂單狀態查詢訂貨及換貨處理

2.1.3 系統架構

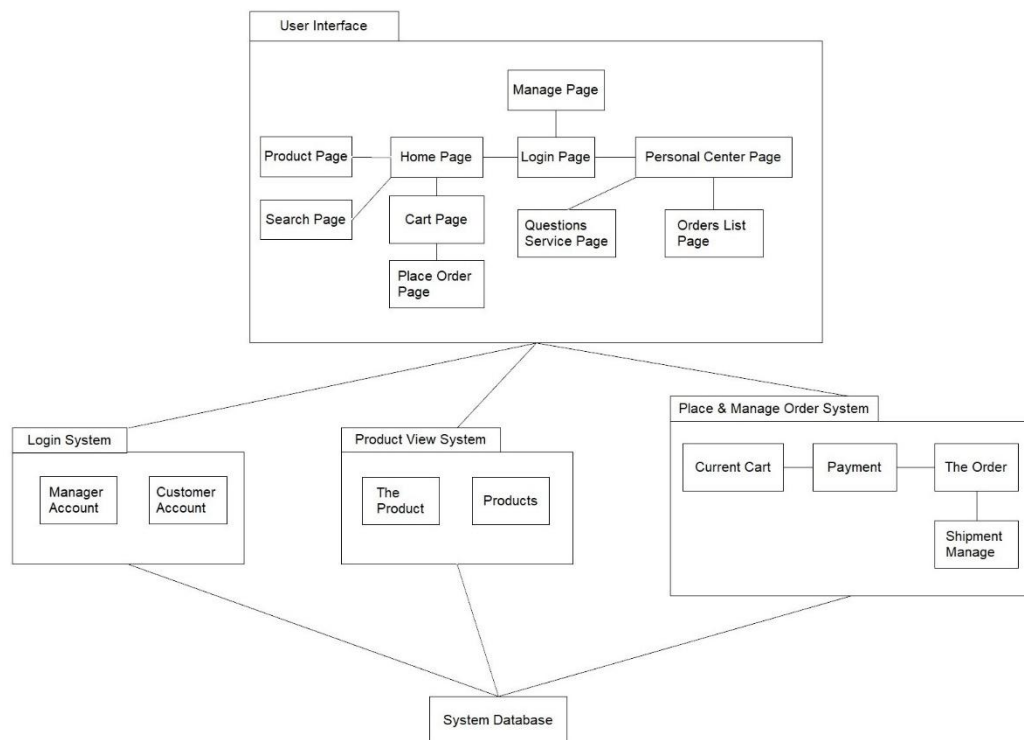
用戶透過使用者介面，可以瀏覽店家所販售之商品，並進行購買、退換貨等指令，使用者介面連接至系統後台資料庫，進行商品資料查詢、訂單查詢、訂單新增等操作，店家則依據訂單規格進行出貨之行為，完成交易。

2.1.4 環境需求

軟體：Google Chrome 版本 60 以上 (macOS / Windows)

硬體：網際網路連線

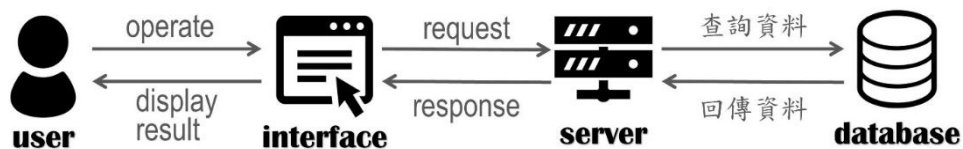
2.2 軟體架構



圖一：軟體架構設計圖(package diagrams)

2.2.1 基本架構

用戶透過使用者介面，可以瀏覽店家所販售之商品，並進行購買、退換貨等指令，使用者介面連接至系統後台資料庫，進行商品資料查詢、訂單查詢、訂單新增等操作，店家則依據訂單規格進行出貨之行為，完成交易。



圖二：系統基本架構示意圖

2.2.2 功能架構

User Interface：管理所有使用者介面

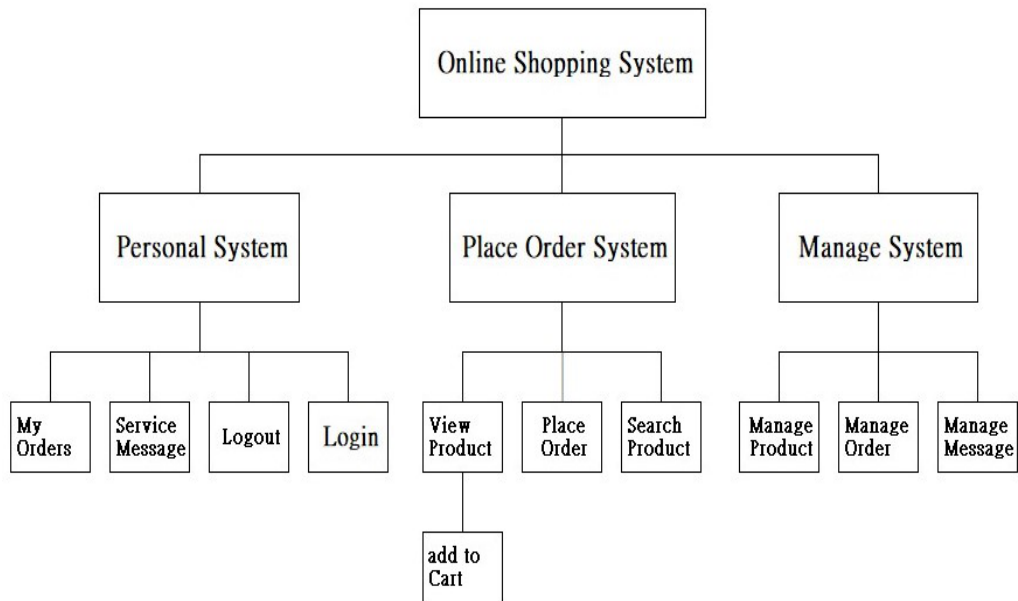
Login System：管理出入系統的所有元件

Product System：管理產品的所有元件

Place & Manage Order System：訂單建立與管理的所有元件

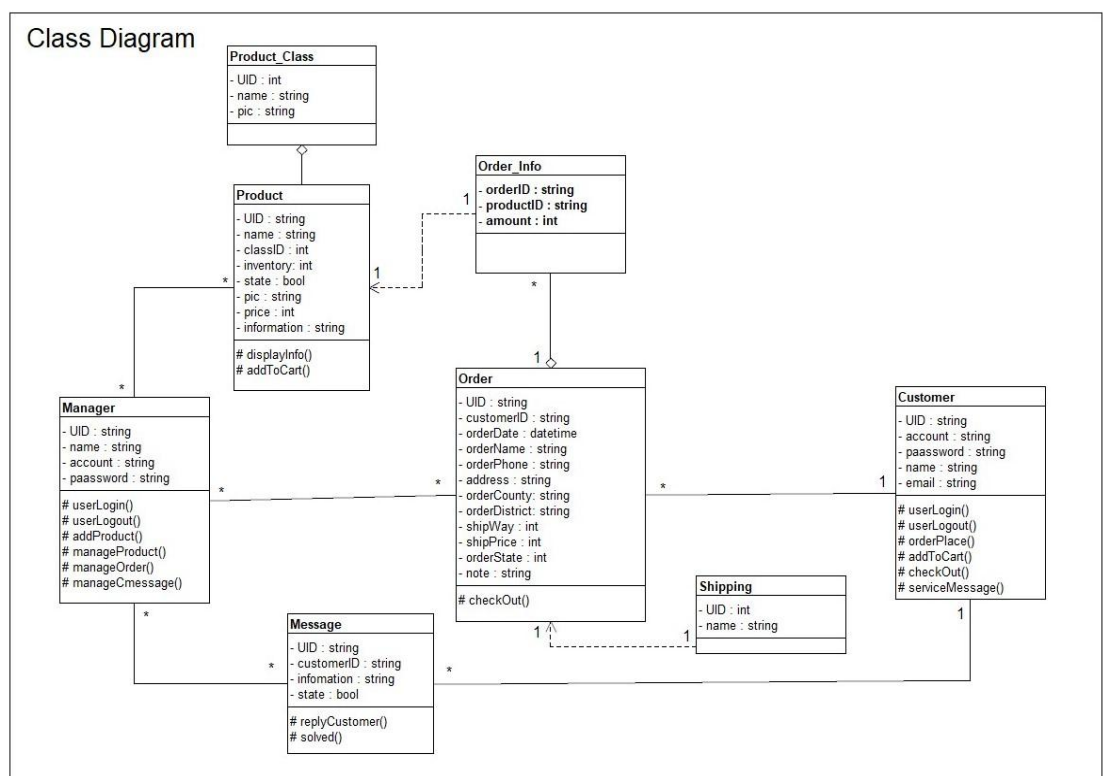
System Database：存放客戶、商品、訂單的資料

用戶透過操作 User Interface，跳轉至不同頁面，使用各種系統功能元件，間接訪問、建立、修改資料庫中的表內容



圖三：系統功能架構圖

2.3 軟體類別定義與描述



圖四：Class Diagram

Class Product_Class

類別描述：

網站上販售商品分類，讓客戶可以根據分類有條理的選購商品

類別屬性：
public int UID 作為資料庫內，商品類別的識別唯一碼 public string name 商品分類類別名稱 public string pic 該商品分類類別的代表圖片
類別函數：

圖五-1： Class Product_Class (Detailed Class Diagram)

Class Product
類別描述：
店家所販售的商品詳細資訊，提供客戶於網頁上瀏覽詳細資訊，包括商品價格、實拍照片、庫存量、詳細敘述，而網頁管理員則可以修改或新增商品資訊，管理販售狀態
類別屬性：
public string UID 作為資料庫內，商品的識別唯一碼 public string name 於網頁上展示給客戶看的商品名稱 public int classID 商品所屬分類別編號 public int inventory 商品庫存量 public string information 商品詳細敘述 public boolean state 商品販售狀態，若為非，則顧客無法再網頁上查看此商品
類別函數：
public void displayInfo() 將產品名稱、照片、庫存、詳細資訊顯示於網頁上 public void addToCart() 紀錄目前所選商品與數量做暫時的紀錄，存放至 Session

圖五-2： Class Product (Detailed Class Diagram)

Class Order_Info
類別描述：
訂單內部詳細商品資訊，關聯資料庫內訂單與商品
類別屬性：
public string orderID 訂單於資料庫中的識別唯一碼 public string productID 訂單中該商品於資料庫中的識別唯一碼 public int amount 訂單中該商品數量
類別函數：

圖五-3： Class Order_Info (Detailed Class Diagram)

Class Order
類別描述：
客戶每次購買交易，會形成一筆訂單，詳細記錄訂購商品及數量、送貨地址及方式、收件人資訊、付款狀態，是交易行為的重要憑據資料
類別屬性：
public string UID 作為資料庫內，訂單的識別唯一碼 public string customerUID 帳戶唯一碼，與帳戶資料表關聯 public datetime orderDate 訂單建立時間 public string orderName 訂單收件人姓名 public string orderPhone 訂單收件人聯絡電話 public string address 訂單收件地址 public string addressCounty 訂單收件縣市 public string addressDistrict 訂單收件行政區 public int shipWay 訂單運送方式 public int shipPrice 訂單運費 public int orderState 訂單狀態，紀錄出貨狀態 public string note 客戶備註的特別事項
類別函數：
public void checkOut(string orderID) 展示訂單資訊

圖五-4： Class Order (Detailed Class Diagram)

Class Customer
類別描述：
客戶帳號，欲在購物系統購買商品的用戶，需先建立帳號，登入系統，網頁中的個人中心可以查看自己的所有訂單紀錄。
類別屬性：
public string UID 作為資料庫內，帳戶的識別唯一碼 public string account 帳戶登入時，查詢資料表的值 public datetime password 用戶密碼，作為隱私安全憑證 public string name 帳戶姓名 public string email 帳戶電子信箱，作為聯繫客戶與宣傳的方式
類別函數：
public boolean userLogin(string ac, string pwd) 透過帳號與密碼驗證，帳戶登入系統 public boolean userLogout() 用戶登出系統 public void orderPlace() 商品結帳，填寫送貨資訊，建立訂單 public void addToCart(string pID, int quantity) 將目前頁面選取之商品及數量加入購物車暫存

<p>public void checkOut(string orderID) 展示訂單資訊</p> <p>public void serviceMessage() 請求客服，以文字訊息方式，傳送疑問至系統，尋求人工服務解決問題</p>
--

圖五-5：Class Customer (Detailed Class Diagram)

Class Manager
類別描述：
網站管理員帳號，可以維護商品、訂單資料，以及處理客服問題
類別屬性：
<p>public string UID 作為資料庫內，帳戶的識別唯一碼</p> <p>public string name 帳戶姓名</p> <p>public string account 帳戶登入時，查詢資料表的值</p> <p>public datetime password 用戶密碼，作為隱私安全憑證</p>
類別函數：
<p>public boolean userLogin(string ac, string pwd) 透過帳號與密碼驗證，帳戶登入系統</p> <p>public boolean userLogout() 用戶登出系統</p> <p>public void addProduct() 建立新商品資料</p> <p>public void manageProduct() 修改商品資料表內容</p> <p>public void manageOrder() 修改訂單資料表內容</p> <p>public void manageCmessage() 呼叫 message 類別，處理客服問題</p>

圖五-6：Class Manager (Detailed Class Diagram)

Class Message
類別描述：
客服訊息，管理員透過訪問查詢，人工解決客戶問題
類別屬性：
<p>public string UID 作為資料庫內，訊息的識別唯一碼</p> <p>public string customerID 帳戶識別唯一碼，關聯客戶資料表</p> <p>public string inforamtion 客戶填寫的問題描述</p> <p>public Boolean state 問題處理狀態</p>
類別函數：
<p>public void replayCustomer() 回覆用戶訊息</p> <p>public void sloved() 將問題狀態改為已解決的 True 狀態</p>

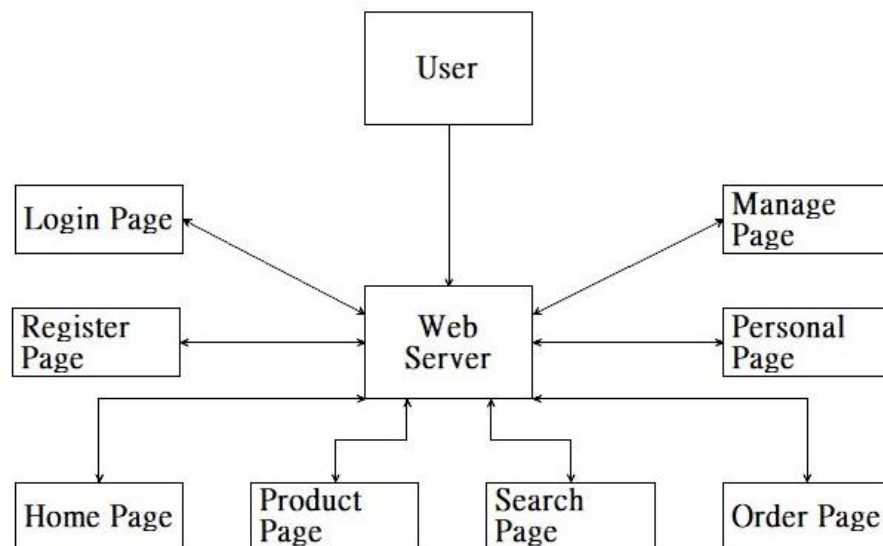
圖五-7：Class Message (Detailed Class Diagram)

Class Shipping
類別描述：
物流送貨方式列表
類別屬性：
public int UID 作為資料庫內，物流方的識別唯一碼 public string name 物流名稱
類別函數：

圖五-8：Class Shipping (Detailed Class Diagram)

2.4 軟體行為描述

◆ 如圖九所示



圖六：Client-object Diagram

三、 軟體介面設計描述

3.1 介面設計概述

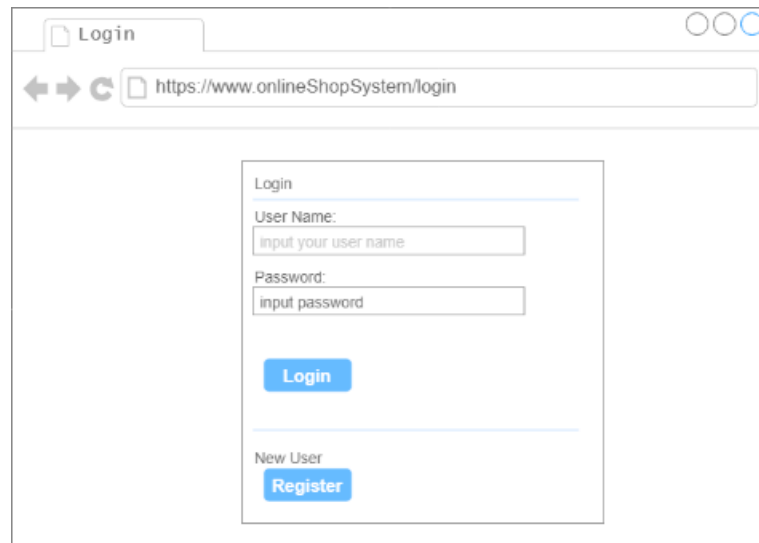
本章節主要描述購物系統之使用者外部介面

3.2 外部介面定義

3.2.1 系統目標

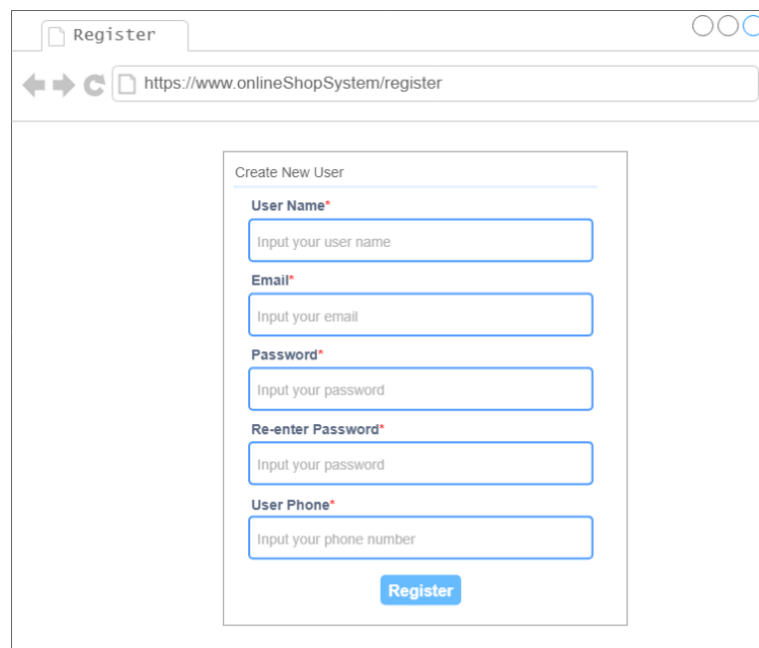
- ◆ 硬體平台：本系統設定於 PC 或智慧型行動裝置之硬體平台上執行
- ◆ 作業系統：支援 Google Chrome 版本 60 以上的 macOS / Windows 系列

3.2.2 登入頁面：提供使用者登入系統的功能，且連接到註冊頁面



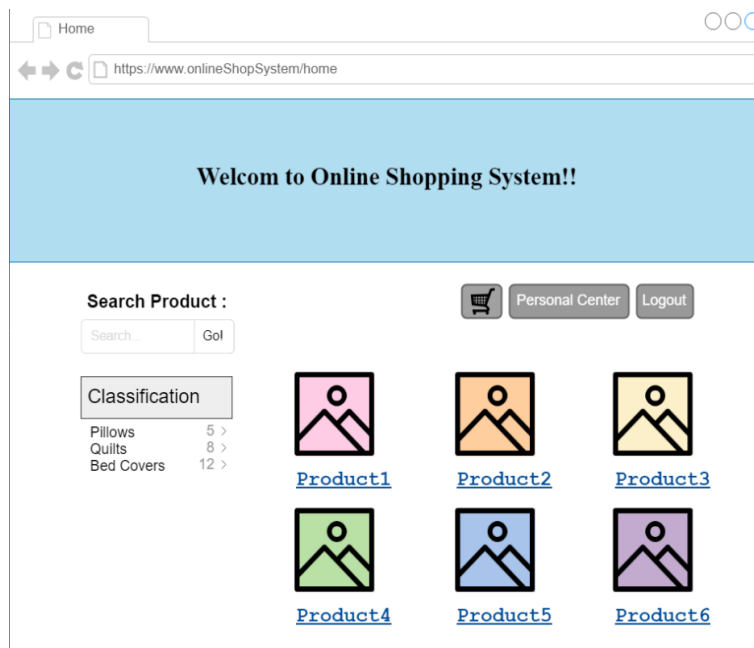
圖七-1：登入頁面(使用者介面示意圖)

3.2.3 註冊頁面：提供使用者註冊會員以使用本系統的功能，且可以連接到註冊頁面



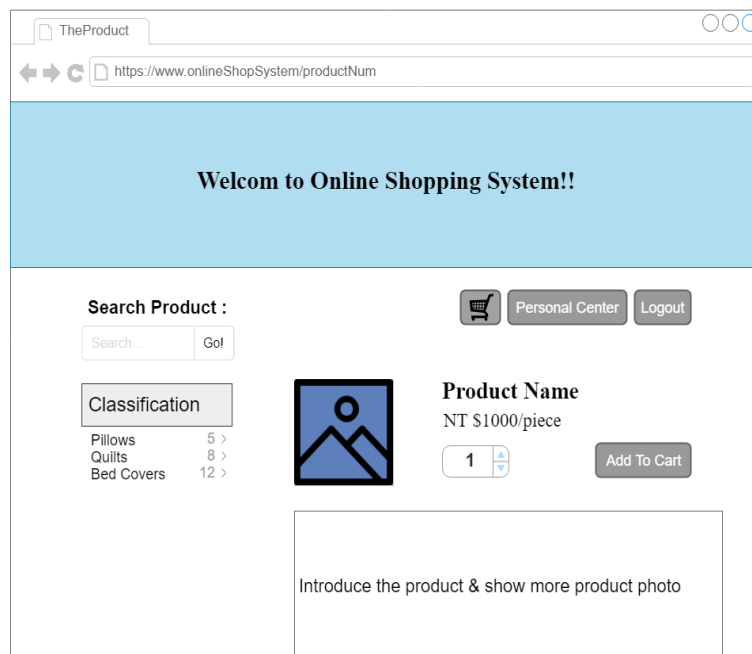
圖七-2：註冊頁面(使用者介面示意圖)

3.2.4 主頁面：當會員成功登入系統後，會導向主頁面，主頁面展示所有商品，點選商品圖片可以查看更詳細的圖文說明；頁面左側有商品分類，用戶能依據類別選購；歡迎字幕下方有三顆按鍵，由右至左分別為登出系統、前往個人中心、購物車；具有搜尋欄位，可以透過關鍵字查詢相關商品。



圖七-3：主頁面(使用者介面示意圖)

3.2.5 單一商品頁面：標示商品價格與目前庫存量，可以透過上下按鈕切換商品數量，點選”Add to Cart”按鍵可以將商品加入購物車中；有詳細的文字敘述與多張圖片可以作為選購參考。



圖七-4：單一商品頁面(使用者介面示意圖)

3.2.6 下單頁面：頁面顯示目前購物車中所有商品名稱與數量，透過每欄的上下按鍵可以修改數量，右側的紅色按鍵可以刪除該產品，系統會計算此筆訂單價格並顯示於運費下方，下方依序選取和填寫運送物流方式、收件人姓名、收件人電話、收件縣市、收件行政區、收件地址、付款方式，點擊最下面的按鍵會在資料庫中建立訂單資訊並通知店家準備出貨

PlaceOrder

https://www.onlineShopSystem/placeOrder

Welcom to Online Shopping System!!

Search Product :

Search... Go!

Classification

- Pillows 5 >
- Quilts 8 >
- Bed Covers 12 >

My Cart

#	Product ID	Quantity	Price	Delete
1	20190106csa	1	NT\$ 190	Delete
2	20190302adv	8	NT\$ 666	Delete
3	20190331pls	2	NT\$ 123	Delete

Shipment Type

Black Cat NTS 60

Total : 2494

Customer Name

Customer Phone

Customer Address County District

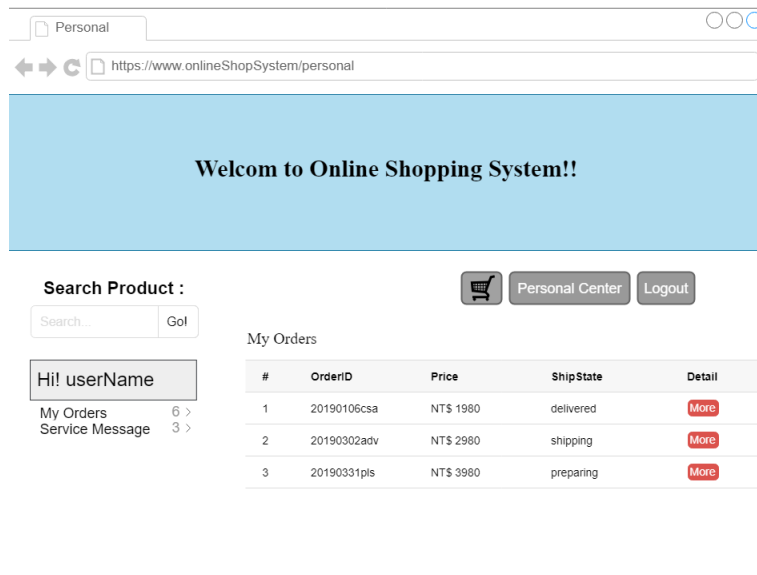
Payment Type

圖七-5：下單頁面(使用者介面示意圖)

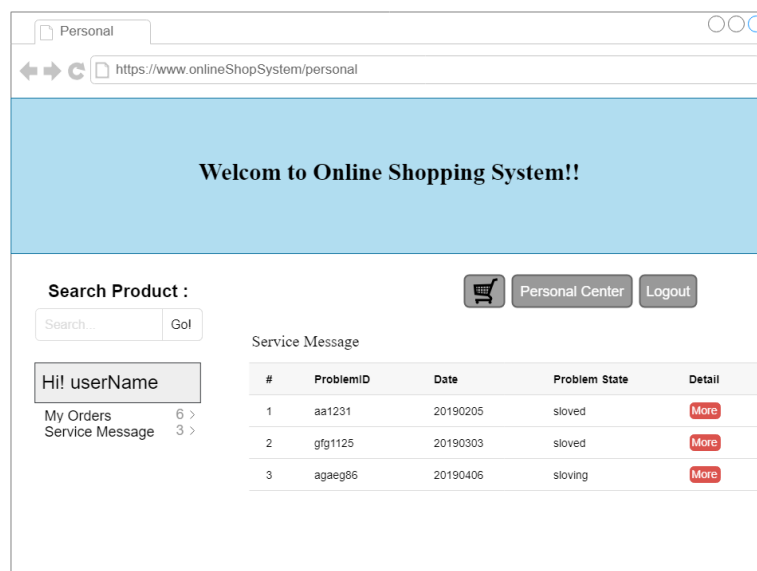
3.2.7 個人中心頁面：

圖七-6 中顯示 My Order 內容，為訂單列表，紀錄客戶於系統中每一筆交易內容與狀態，點選右側紅色按鍵可以看到訂單詳細內容，包含購買商品名稱、數量、送貨地點、付款方式等

圖七-7 中顯示 Service Message 內容，客戶服務訊息列表，客戶可以透過此撰寫訊息，已詢問店家問題，系統會盡速以人工方式回覆及處理問題，點選右側紅色按鍵可以看到針對該欄位問題的所有對話



圖七-6：個人中心頁面-訂單管理(使用者介面示意圖)



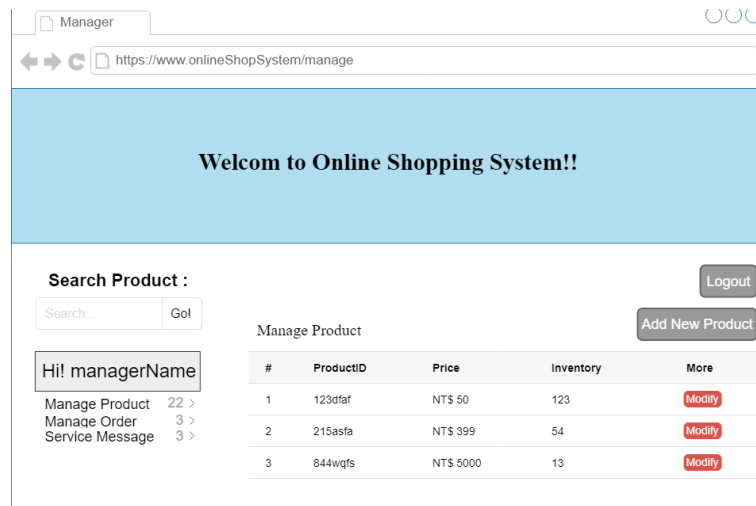
圖七-7：個人中心頁面-客服訊息(使用者介面示意圖)

3.2.8 管理人員頁面：

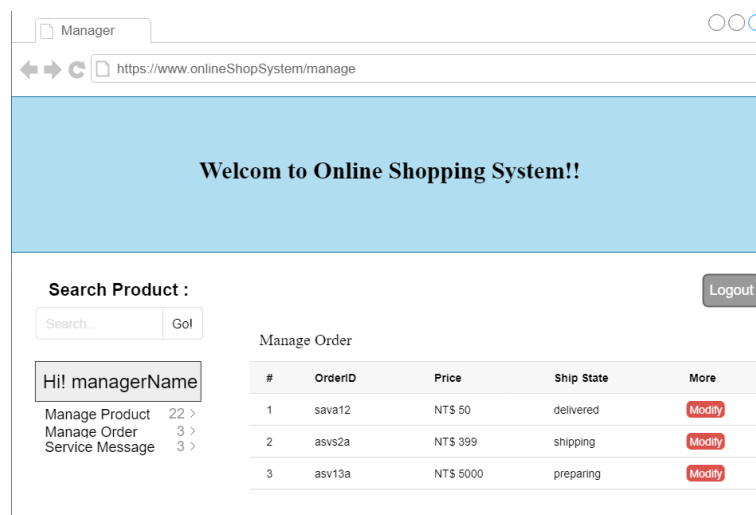
圖六-8 中顯示 Manage Product 內容，為商品列表，點選上方” Add new Product” 按鍵可以新增商品，點選右側紅色按鍵可以看到修改該欄位的商品內容及數量

圖六-9 中顯示 Manage Order 內容，為訂單列表，點選右側紅色按鍵可以看到修改該欄訂單的運送狀態

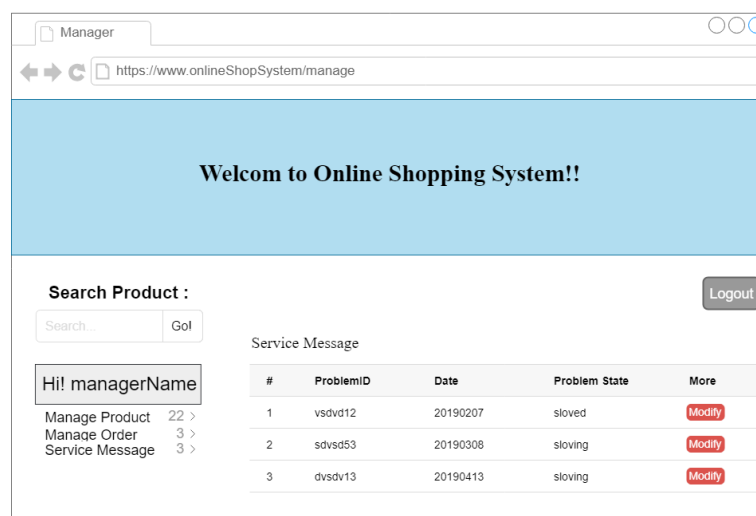
圖六-10 中顯示 Service Message 內容，客戶服務訊息列表，管理員可以透過此撰寫訊息，回覆客戶問題，點選右側紅色按鍵可以看到針對該欄位問題的所有對話



圖七-8：管理人員頁面(使用者介面示意圖)

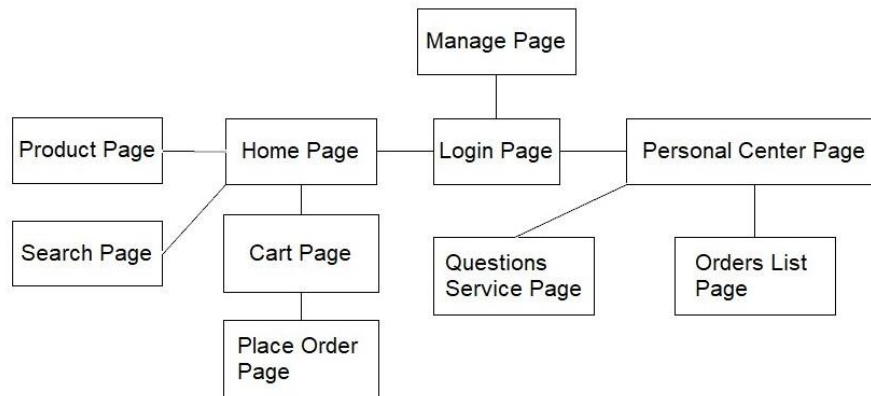


圖七-9：管理人員頁面(使用者介面示意圖)



圖七-10：管理人員頁面(使用者介面示意圖)

3.3 類別介面定義



圖八：類別介面定義示意圖

四、軟體設計描述

4.1 設計概述

本節內容主要是子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。

4.2 軟體功能描述

4.2.1 帳號申請

對象：一般用戶

說明：為了記錄用戶資訊及管理訂單，使用者需要一組帳號，當作識別碼

輸入：設定帳號(手機號碼)與密碼字串，並設定姓名及電子信箱

處理：檢驗輸入格式是否正確，比對資料庫中是否已經存在該帳號，決定是否可以進行註冊

輸出：註冊成功或失敗之訊息

Pseudo Code：

```
if(validateAccount(Account)){
    bool check = false;
    string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
    using (SqlConnection cnn = new SqlConnection(tcnn)){
        string tcmd = "SELECT COUNT(*) FROM 客戶 WHERE 手機號碼=@acct";
        using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
            cmd.Parameters.Add("@acct", SqlDbType.NVarChar).Value =
Account;

            cnn.Open();
            int cc = (int)cmd.ExecuteScalar();
            cmd.Dispose();
            cnn.Close();
            if (cc > 0)
                check = true;
```



```

    }
}
if(check){
    if(validateEmail(Email)){
        string tcnn =
"server=localhost;database=onlineShop;uid=admin;pwd=abc123";
        using (SqlConnection cnn = new SqlConnection(tcnn)){
            string tcmd = "INSERT INTO 客戶(客戶姓名,手機號碼,密碼,電子
信箱)"
                        + "VALUES(@Name,@Phone,@PW,@Email)";
            using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
                cmd.Parameters.Add("@Name", SqlDbType.NVarChar).Value = Name;
                cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = Phone;
                cmd.Parameters.Add("@PW", SqlDbType.NVarChar).Value = PW;
                cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = Email;
                cnn.Open();
                int cc = cmd.ExecuteNonQuery();
                cmd.Dispose();
                cnn.Close();
                if (cc > 0)
                    return "帳號申請成功";
                else
                    return "帳號申請失敗";
            }
        }
    }
    else
        return "電子信箱格式錯誤!";
}
else
    return "此帳號(手機號碼)已被註冊過!";
}
else
    return "帳號(手機號碼)格式錯誤!";
}

```

4.2.2 網站登入

對象：一般用戶、管理員

說明：使用者透過登入購物網站，獲得其所有訂單資訊，以及購買商品；管理員登入網站，以處理顧客問題和管理商品與訂單

輸入：輸入用戶於註冊時填寫之帳號密碼

處理：比對資料庫中用戶帳號密碼資料表，查看帳號是否存在，密碼一致性，決定是否完成登入，並登入資訊加入 Session 中

輸出：登入成功或失敗之訊息

Pseudo Code：

```

bool check = false;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";

```

```

using (SqlConnection cnn = new SqlConnection(tcnn)) {
    string tcmd = "SELECT 密碼 FROM 客戶 WHERE 手機號碼=@acct";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)) {
        cmd.Parameters.Add("@acct", SqlDbType.NVarChar).Value = Account;
        cnn.Open();
        SqlDataReader mydr = cmd.ExecuteReader();
        while (mydr.Read()){
            if(reader.GetValue(0).Equals>Password)) {
                check = true;
                Session["LoginName"] = Account;
            }
        }
        cnn.Close();
    }
}
if(check)
    return RedirectToAction("LoginSuccess");
else
    return RedirectToAction("LoginFail");

```

4.2.3 網站登出

對象：一般用戶、管理員

說明：基於安全性考量，為防止他人盜用帳號，離開本購物網站時，提供登出機制

輸入：點選登出選項

處理：清除 Session 中的登入資訊

輸出：登出成功訊息

Pseudo Code：

```

Session["LoginName"] = null;
return RedirectToAction("Home");

```

4.2.4 忘記密碼

對象：一般用戶

說明：當用戶忘記密碼時，提供重新設定密碼之機制，透過連絡方式驗證使用者，進行密碼重新設定

輸入：註冊時登記之電子信箱

處理：檢驗輸入格式是否正確，比對資料庫中是否存在該帳號，以判斷是否進行密碼重設

輸出：是否發送重設密碼之郵件至電子信箱

Pseudo Code：

```

string Email = "";
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)) {
    string tcmd = "SELECT 電子信箱 FROM 客戶 WHERE 手機號碼=@acct";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)) {

```

```

        cmd.Parameters.Add("@acct", SqlDbType.NVarChar).Value = Account;
        cnn.Open();
        SqlDataReader mydr = cmd.ExecuteReader();
        while (mydr.Read())
            Email = reader.GetValue(0);
        cnn.Close();
    }
}
if(Email == "")
    return "該帳號不存在";
string randomPW = CreateRandomPW();
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "UPDATE 客戶 SET 密碼=@password WHERE 手機號碼=@account";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)) {
        cmd.Parameters.Add("@acct", SqlDbType.NVarChar).Value = Account;
        cmd.Parameters.Add("@Password", SqlDbType.NVarChar).Value =
randomPW;
        cnn.Open();
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        cnn.Close();
    }
}
}
MailMessage msg = new MailMessage();
msg.To.Add(string.Join(",", MailList.ToArray()));
msg.From = new MailAddress(Email, "新密碼", System.Text.Encoding.UTF8);
msg.Subject = "新密碼";
msg.SubjectEncoding = System.Text.Encoding.UTF8;
msg.Body = "您的新密碼：" + randomPW;
msg.IsBodyHtml = true;
msg.BodyEncoding = System.Text.Encoding.UTF8;
msg.Priority = MailPriority.Normal;
SmtpClient MySmtp = new SmtpClient("smtp.gmail.com", 587);
MySmtp.Credentials = new System.Net.NetworkCredential("帳號", "密碼");
MySmtp.EnableSsl = true;
MySmtp.Send(msg);

```

4.2.5 商品搜尋

對象：一般用戶

說明：藉由輸入關鍵字，過濾所有產品資訊，幫助使用者快速找到商品，省去逐一查找的時間

輸入：輸入欲搜尋之產品名稱或關鍵字

處理：比對資料庫中商品資料表，提取所有相關商品資訊

輸出：顯示查找到的相關產品列表

Pseudo Code :

```
DataTable tt = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT * FROM 商品資料 WHERE FREETEXT(*,@text)";
    using (SqlCommand cmd=new SqlCommand(tcmd,cnn))    {
        cmd.Parameters.Add("@text", SqlDbType.NVarChar).Value = Text;
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if(mydr.HasRows){
                tt = new DataTable();
                tt.Load(mydr);
                mydr.Close();
            }
        }
        cnn.Close();
    }
}
return tt;
```

4.2.6 查看商品內容

對象：一般用戶

說明：查看商品資訊

輸入：點選欲察看之商品

處理：比對資料庫中商品資料表，提取該商品資訊

輸出：顯示該商品相關資訊

Pseudo Code :

```
DataTable tt = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT * FROM 商品資料 WHERE 商品編號 = @pid";
    using (SqlCommand cmd=new SqlCommand(tcmd,cnn)){
        cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value = productId;
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if(mydr.HasRows)    {
                tt = new DataTable();
                tt.Load(mydr);
                mydr.Close();
            }
        }
        cnn.Close();
    }
}
return tt;
```

4.2.7 購物車暫存

對象：一般用戶

說明：使用者瀏覽商品時，可以將欲購買之商品加入購物車清單暫時存放

輸入：於商品展示頁面選取數量後，點選加入購物車之按鈕

處理：查詢資料庫，確認商品型號與庫存是否足夠，將商品編號與數量等資訊存放在 Session 中

輸出：商品加入購物車成功或失敗之訊息

Pseudo Code：

```
HashSet<Models.CartItem> myCart = null;
if (Session["Cart"] == null)
    myCart = new HashSet<Models.CartItem>();
else
    myCart = Session["Cart"] as HashSet<Models.CartItem>;
Models.CartItem ci = new Models.CartItem();
ci.Id = pid.Value;
if (myCart.Contains(ci))
    return RedirectToAction("MyShoppingCart");
Dictionary<string, object> p = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT 商品編號, 商品名稱, 價錢 FROM 商品資料 WHERE 商品編號=@pid";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
        cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value = productId;
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if (mydr.HasRows){
                p = new Dictionary<string, object>();
                mydr.Read();
                for (int i = 0; i < mydr.FieldCount; i++)
                    p.Add(mydr.GetName(i), mydr[i]);
            }
        }
        conn.Close();
    }
}
if (p == null)
    return RedirectToAction("Main");
ci.Id = Convert.ToString(p["商品編號"]);
ci.Name = p["商品名稱"].ToString();
ci.Quantity = 1;
ci.Price = Convert.ToInt32(p["價錢"]);
myCart.Add(ci);
Session["Cart"] = myCart;
return RedirectToAction("MyShoppingCart");
```

4.2.8 購物車結帳

對象：一般用戶

說明：使用者透過登入購物網站，獲得其所有訂單資訊，以及購買商品

輸入：於購物車頁面點選結帳之按鍵

處理：取出 Session 中購物車資訊，並查詢資料庫，確認商品型號與庫存是否足夠

輸出：頁面導向訂單資料填寫頁面

Pseudo Code：

```
if (Session["Cart"] == null)
    return RedirectToAction("Main");
HashSet<Models.CartItem> myCart = Session["Cart"] as HashSet<Models.CartItem>;
int shipping = 0;
int amount = 0;
int total = 0;
foreach (Models.CartItem ii in myCart)
    amount += ii.Price * ii.Quantity;
if (amount < 500)
    shipping = 80;
else if (amount < 1000)
    shipping = 40;
total = amount + shipping;
ViewData["Shipping"] = shipping;
ViewData["Amount"] = amount;
ViewData["Total"] = total;
return RedirectToAction("Pay");
```

4.2.9 直接購買

對象：一般用戶

說明：用戶欲直接購買該項單一產品

輸入：選取商品數量後，點選直接購買之按鍵

處理：查詢資料庫，確認商品型號與庫存是否足夠，並放入 Session 中

輸出：頁面導向訂單資料填寫頁面或是缺貨訊息

Pseudo Code：

```
int shipping = 0;
int amount = 0;
int total = 0;
HashSet<Models.CartItem> myCart = new HashSet<Models.CartItem>();
Models.CartItem ci = new Models.CartItem();
Dictionary<string, object> p = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT 商品編號, 商品名稱, 價錢 FROM 商品資料 WHERE 商品編號=@pid";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
        cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value = productId;
        cnn.Open();
```

```

        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if (mydr.HasRows){
                p = new Dictionary<string, object>();
                mydr.Read();
                for (int i = 0; i < mydr.FieldCount; i++){
                    p.Add(mydr.GetName(i), mydr[i]);
                }
            }
        }
        conn.Close();
    }
}

ci.Id = Convert.ToString(p["商品編號"]);
ci.Name = p["商品名稱"].ToString();
ci.Quantity = 1;
ci.Price = Convert.ToInt32(p["價錢"]);
myCart.Add(ci);
Session["Cart"] = myCart;
amount += p.Price * p.Quantity;
if (amount < 500)
    shipping = 80;
else if (amount < 1000)
    shipping = 40;
total = amount + shipping;
ViewData["Shipping"] = shipping;
ViewData["Amount"] = amount;
ViewData["Total"] = total;
return RedirectToAction("Pay");

```

4.2.10 完成訂單

對象：一般用戶

說明：使用者填寫商品運送與付款相關細節，正式完成產品訂購之流程

輸入：根據系統提示與要求填寫完整運送與付款資訊，並同意相關規範

處理：確認填寫資料是否符合格式且完整，於資料庫中新增此筆訂單資訊，店家準備出貨事宜

輸出：訂單下訂成功或失敗訊息

Pseudo Code：

```

string orderId = AddNewOrder(OrderInfo);
HashSet<Models.CartItem> myCart = Session["Cart"] as HashSet<Models.CartItem>;
SqlTransaction tx;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    cnn.Open();
    tx = cnn.BeginTransaction(IsolationLevel.Serializable);
    foreach (Models.CartItem ii in myCart){

```

```

        if ((int)ii["Quantity"] > 0)
            this.AddNewOrderItem(tx, OrderId, (int)ii["Id"], (int)ii["Quantity"]);
    }
    tx.Commit();
    cnn.Close();
}

```

4.2.11 訂單追蹤

對象：一般用戶

說明：提供商品運送即時資訊，方便用戶追蹤與取貨

輸入：登入系統後，點選訂單管理之按鈕，再選取欲查詢之該筆訂單

處理：查詢資料庫，回傳該筆訂單運送狀態

輸出：頁面顯示訂單運送狀態訊息

Pseudo Code：

```

DataTable tt = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT * FROM 訂單 WHERE 訂單編號 = @oid";
    using (SqlCommand cmd=new SqlCommand(tcmd,cnn)){
        cmd.Parameters.Add("@oid", SqlDbType.NVarChar).Value = OrderId;
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if(mydr.HasRows) {
                tt = new DataTable();
                tt.Load(mydr);
                mydr.Close();
            }
        }
        cnn.Close();
    }
}
return tt;

```

4.2.12 聯繫客服解決問題

對象：一般用戶

說明：客戶收到商品後，與實際預期有落差，或者商品於運送過程中有毀損、瑕疵，想要申請退換貨和權益；或是有任何問題想要申訴，需要聯繫店家

輸入：登入系統後，進入用戶個人中心，填寫和送出客訴資訊

處理：新增該筆資訊於資料庫中，等待管理員查看訊息

輸出：客服信件是否成功送出之訊息

Pseudo Code：

```

string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";

```



```

using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "INSERT INTO 客服訊息(客戶編號, 訊息內容) VALUES(@cid,@text)";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
        cmd.Parameters.Add("@cid", SqlDbType.NVarChar).Value = CID;
        cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = Text;

        cnn.Open();
        int cc = cmd.ExecuteNonQuery();
        cmd.Dispose();
        cnn.Close();
        if (cc > 0)
            return "客服訊息儲存成功";
        else
            return "客服訊息儲存失敗";
    }
}

```

4.2.13 使用說明

使用對象：一般用戶

說明：使用者對於本購物系統有任何疑慮，可以藉由使用說明來理解與解決問題

輸入：於頁面點選使用說明之選項

處理：查找系統設定之說明文件訊息

輸出：於頁面顯示說明訊息

Pseudo Code：

```
return "使用說明內容";
```

4.2.14 新增商品

對象：管理員

說明：當商家要加入新產品時，管理者需要於商品展示頁面中加入新商品資訊

輸入：管理員登入管理網站後台，新增商品資訊

處理：比對該商品編號及名稱是否已經存在，更新資料庫中產品資料表

輸出：產品新增成功/失敗訊息

Pseudo Code：

```

bool check = false;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "INSERT INTO 商品資料(商品名稱,類別編號,價錢,庫存量,介紹內容,圖片)"
        + "VALUES(@name,@cid,@amount,@stock,@text,@pic)";
    using (SqlCommand cmd = new SqlCommand(tcmd, conn)){
        cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = Name;
        cmd.Parameters.Add("@cid", SqlDbType.Int).Value = ClassID;
    }
}

```

```

        cmd.Parameters.Add("@amount", SqlDbType.Int).Value = Amount;
        cmd.Parameters.Add("@stock", SqlDbType.Int).Value = Stock;
        cmd.Parameters.Add("@text", SqlDbType.NVarChar).Value = Text;
        cmd.Parameters.Add("@pic", SqlDbType.NVarChar).Value = Picture;
        conn.Open();
        int cc = cmd.ExecuteNonQuery();
        cmd.Dispose();
        if (cc > 0)
            check = true;
    }
    conn.Close();
}
if(check)
    return "新增商品成功";
else
    return "新增商品失敗";

```

4.2.15 刪除商品

對象：管理員

說明：當商家要下架或是停售某產品時，管理者需要於商品展示頁面中刪除該商品資訊

輸入：管理員登入管理網站後台，管理商品頁面，選取要刪除之商品

處理：於資料庫中產品資料表，將該商品狀態設為 False

輸出：產品刪除成功/失敗訊息

Pseudo Code：

```

bool check = false;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "UPDATE 商品資料 SET 販賣狀態=0 WHERE 商品編號=@pid";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)){
        cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value = ProductID;
        conn.Open();
        int cc = cmd.ExecuteNonQuery();
        cmd.Dispose();
        if (cc > 0)
            check = true;
    }
    conn.Close();
}
if(check)
    return "刪除商品成功";
else
    return "刪除商品失敗";

```

4.2.16 修改商品

對象：管理員

說明：當商家要修改指定產品資訊時，管理者需要於商品展示頁面中修改該商品資訊

輸入：管理員登入管理網站後台，管理商品頁面，選取欲修改之商品，並輸入新內容

處理：於資料庫中產品資料表，修改產品資訊

輸出：產品修改成功/失敗訊息

Pseudo Code：

```
bool check = false;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "UPDATE 商品資料 SET 商品名稱=@name,類別編號=@cid,價錢=@amount,"
        + "庫存量=@stock,介紹內容=@text,圖片=@pic WHERE 商品編號=@pid";
    using (SqlCommand cmd = new SqlCommand(tcmd, cnn)) {
        cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value = ProductID;
        cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = Name;
        cmd.Parameters.Add("@cid", SqlDbType.Int).Value = ClassID;
        cmd.Parameters.Add("@amount", SqlDbType.Int).Value = Amount;
        cmd.Parameters.Add("@stock", SqlDbType.Int).Value = Stock;
        cmd.Parameters.Add("@text", SqlDbType.NVarChar).Value = Text;
        cmd.Parameters.Add("@pic", SqlDbType.NVarChar).Value = Picture;
        conn.Open();
        int cc = cmd.ExecuteNonQuery();
        cmd.Dispose();
        if (cc > 0)
            check = true;
    }
    conn.Close();
}
if(check)
    return "修改商品成功";
else
    return "修改商品失敗";
```

4.2.17 管理訂單

對象：管理員

說明：當店家出貨以後，需要將訂單運送狀態改為已出貨，或是處理客戶要求取消訂單等問題

輸入：管理員登入管理網站後台，管理訂單頁面，選取欲修改或刪除之訂單，輸入新內容

處理：於資料庫中訂單資料表，修改和刪除訂單資訊

輸出：訂單修改成功/失敗訊息

Pseudo Code：

```

DataTable tt = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT * FROM 訂單 WHERE 訂單狀態=0";
    using (SqlCommand cmd=new SqlCommand(tcmd,cnn)){
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if(mydr.HasRows) {
                tt = new DataTable();
                tt.Load(mydr);
                mydr.Close();
            }
        }
        cnn.Close();
    }
}
return tt;
ModifyOrder(tt);

```

4.2.18 管理客服

對象：管理員

說明：供管理員查看並管理客服訊息

輸入：管理員登入管理網站後台，管理客服頁面，選取欲處理之項目，設定處理狀態，並輸入回覆訊息

處理：於資料庫中產品資料客訴表，修改狀態

輸出：將處理結果之訊息藉由電子信箱傳送給客戶

Pseudo Code：

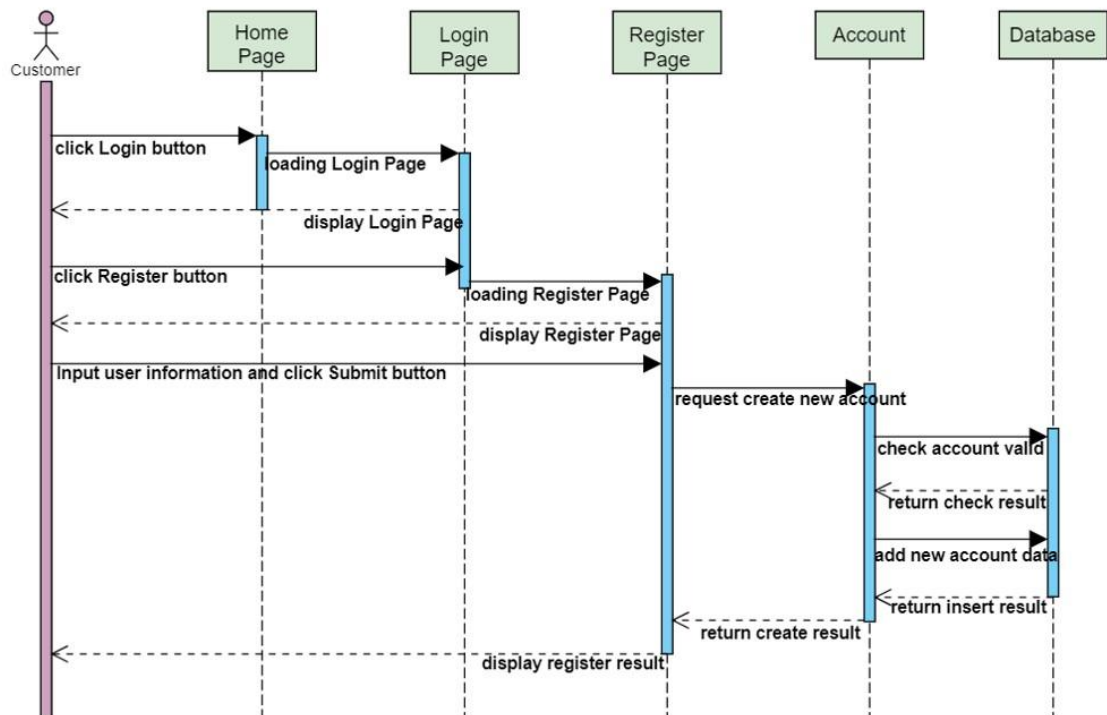
```

DataTable tt = null;
string tcnn = "server=localhost;database=onlineShop;uid=admin;pwd=abc123";
using (SqlConnection cnn = new SqlConnection(tcnn)){
    string tcmd = "SELECT * FROM 客服訊息 WHERE 處理狀態=0";
    using (SqlCommand cmd=new SqlCommand(tcmd,cnn)){
        cnn.Open();
        using (SqlDataReader mydr = cmd.ExecuteReader()){
            if(mydr.HasRows) {
                tt = new DataTable();
                tt.Load(mydr);
                mydr.Close();
            }
        }
        cnn.Close();
    }
}
return tt;
ModifyMessage(tt);

```

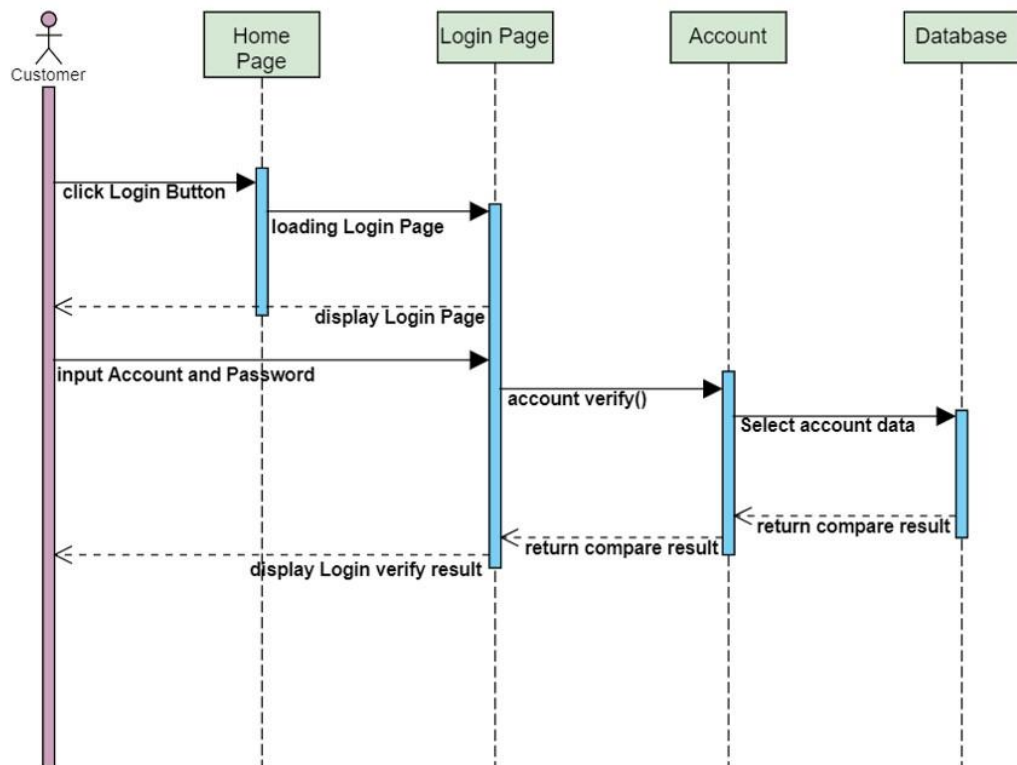
4.3 軟體行為描述

Sequence Diagram - Register



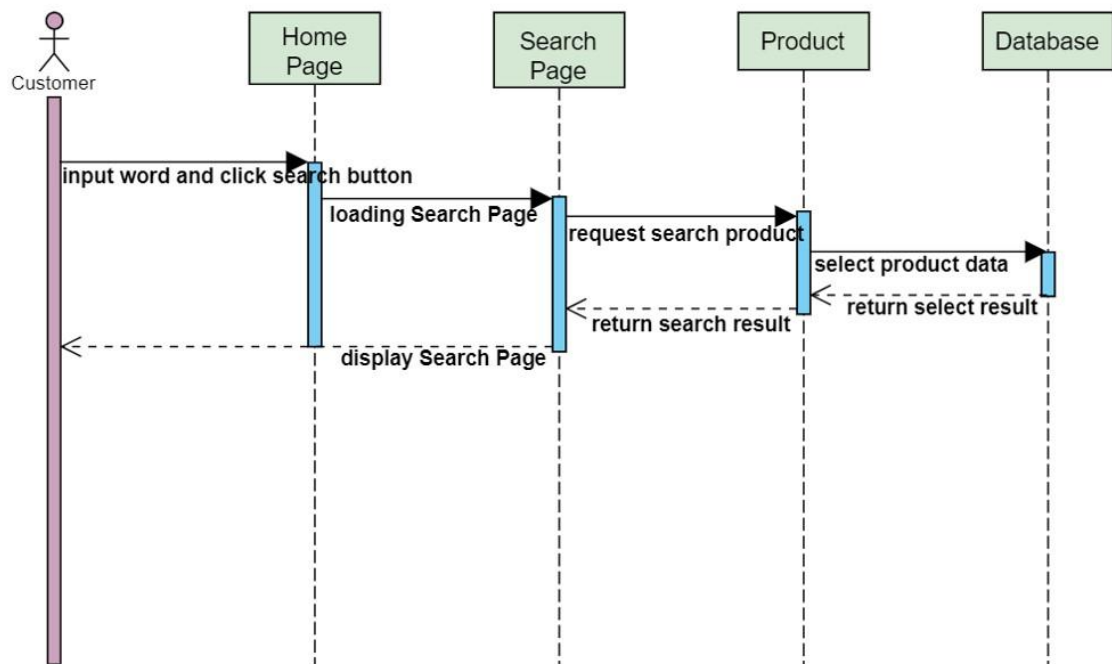
圖九-1：用戶註冊(sequence diagram)

Sequence Diagram - Login



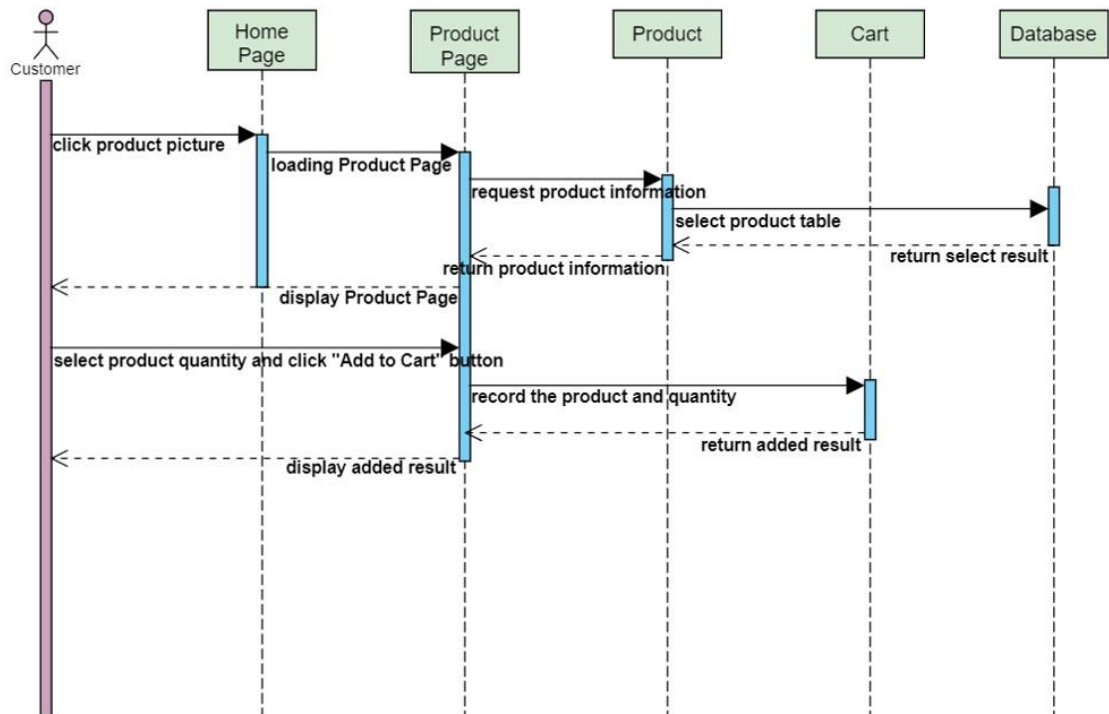
圖九-2：用戶登入(sequence diagram)

Sequence Diagram - Search Product



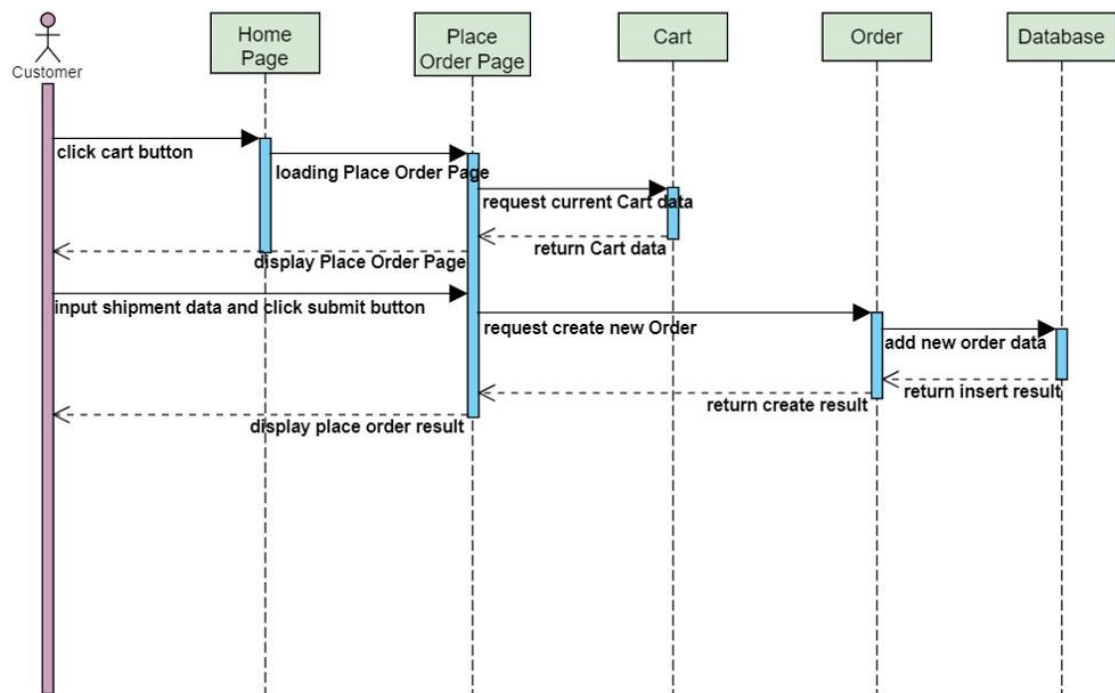
圖九-3：用戶搜尋商品(sequence diagram)

Sequence Diagram - View the product & Add to Cart



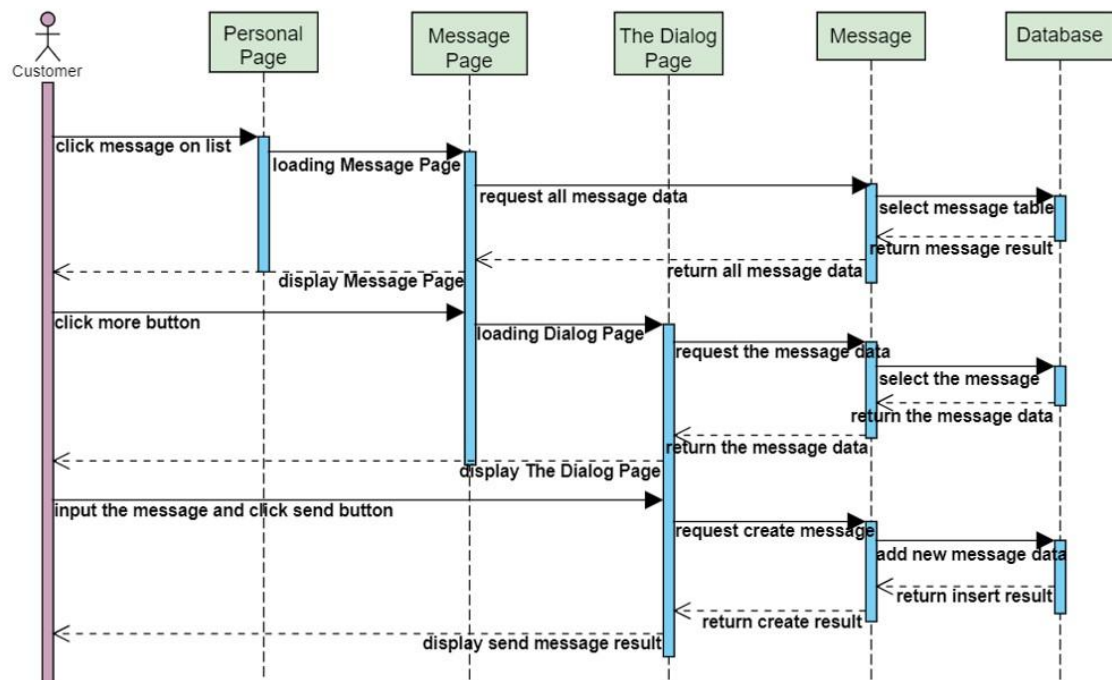
圖九-4：用戶選購商品加入購物車(sequence diagram)

Sequence Diagram - Place Order



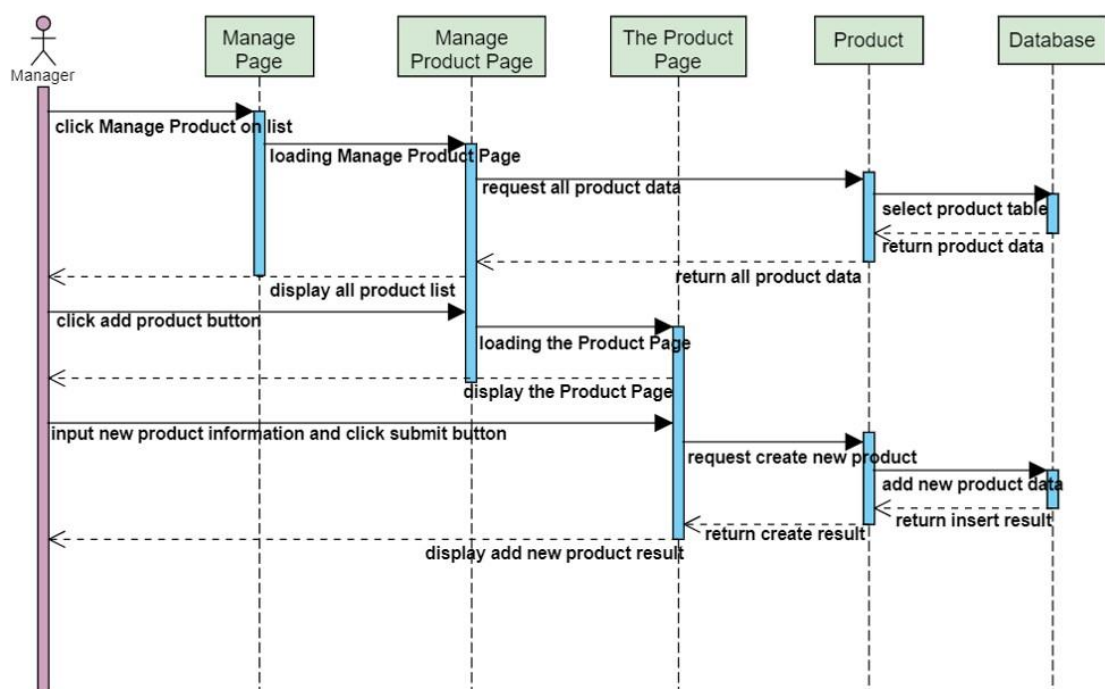
圖九-5：用戶下單(sequence diagram)

Sequence Diagram - Service Message



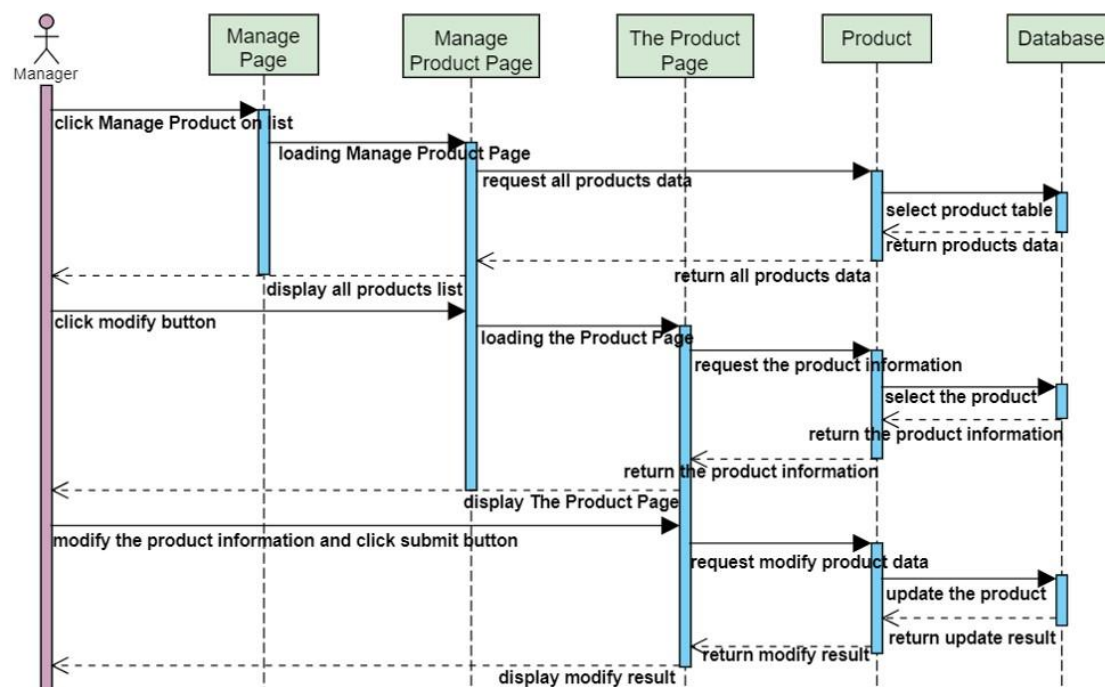
圖九-6：用戶申請客服(sequence diagram)

Sequence Diagram - Add new Product



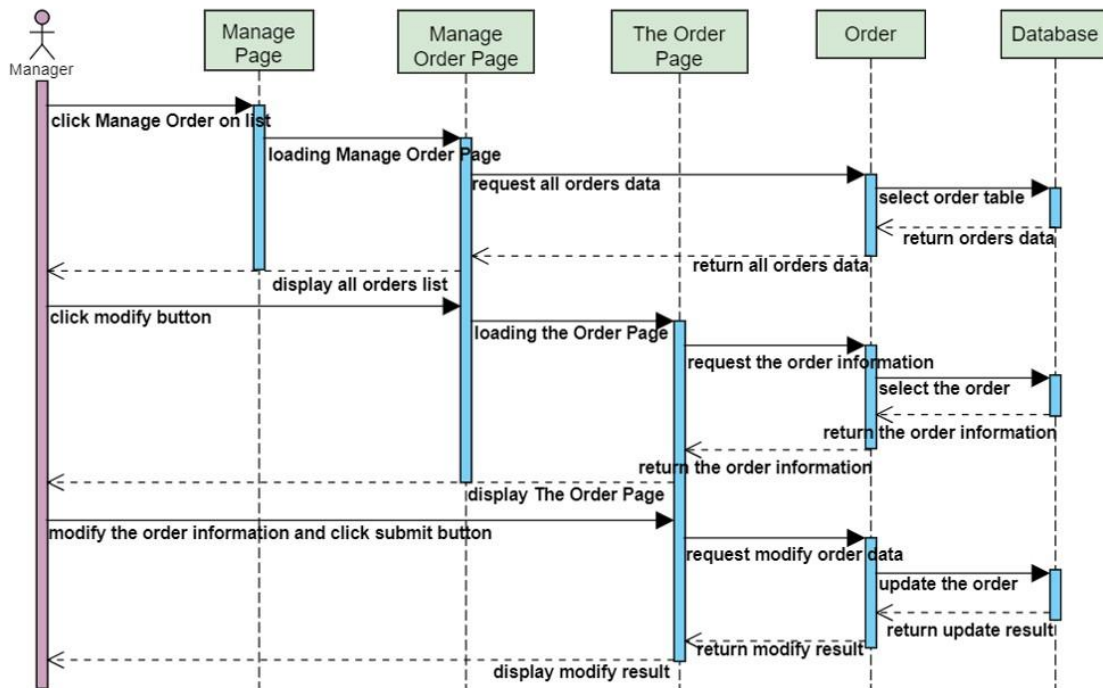
圖九-7：管理員新增商品(sequence diagram)

Sequence Diagram - Manage Product



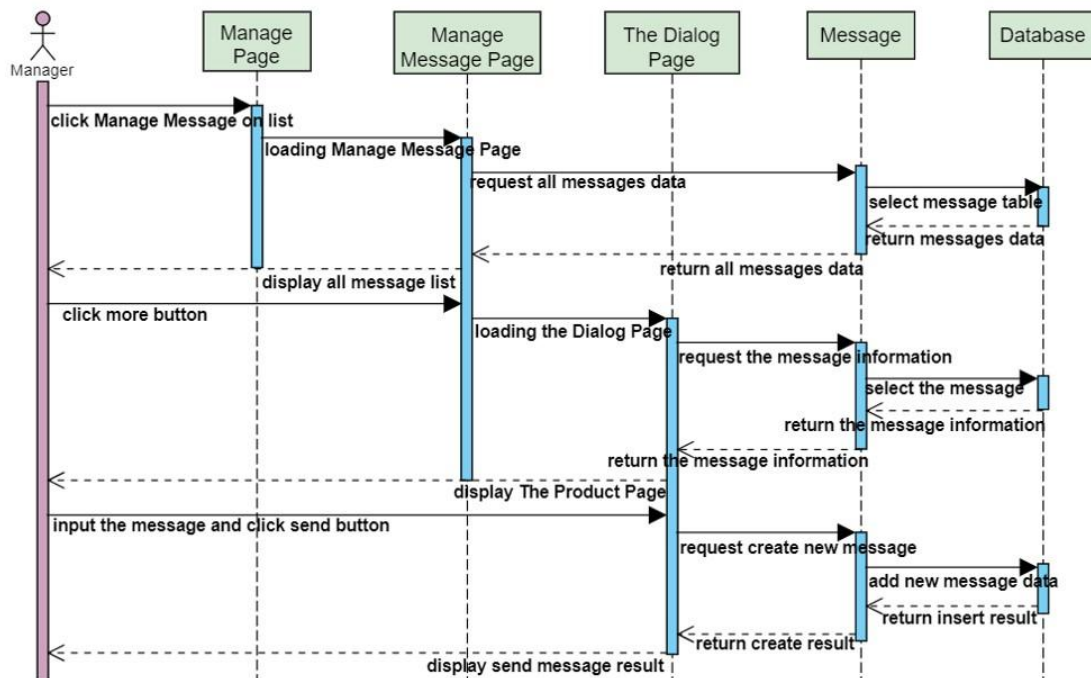
圖九-8：管理員管理商品(sequence diagram)

Sequence Diagram - Manage Order



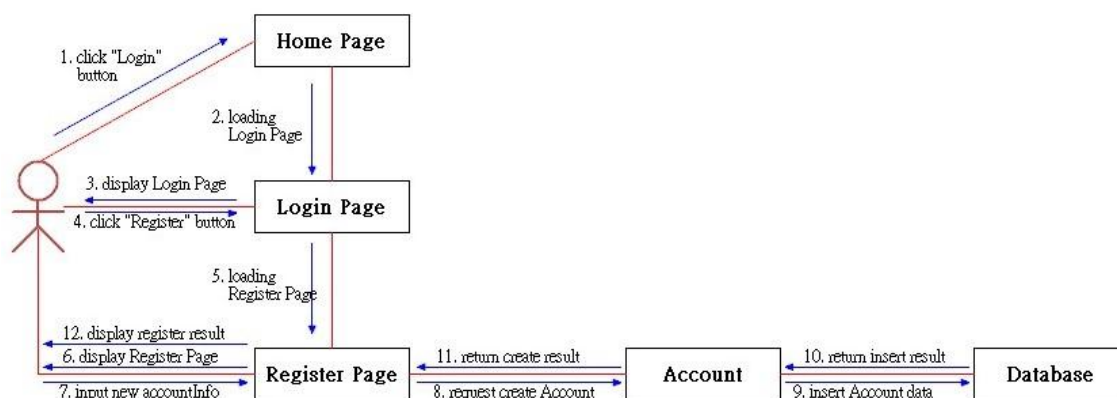
圖九-9：管理員管理訂單(sequence diagram)

Sequence Diagram - Manage Message

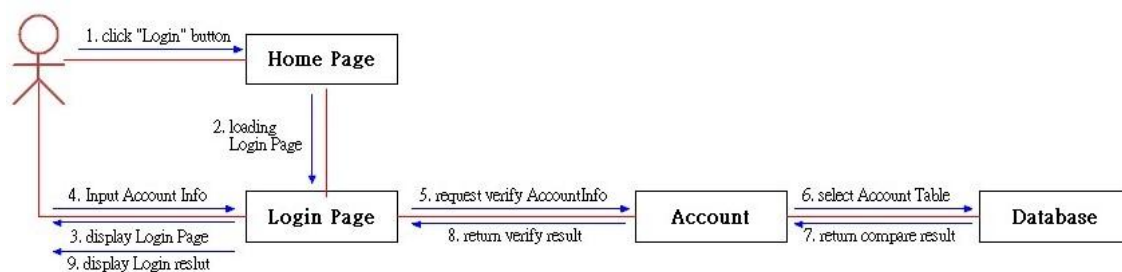


圖九-10：管理員管理訊息(sequence diagram)

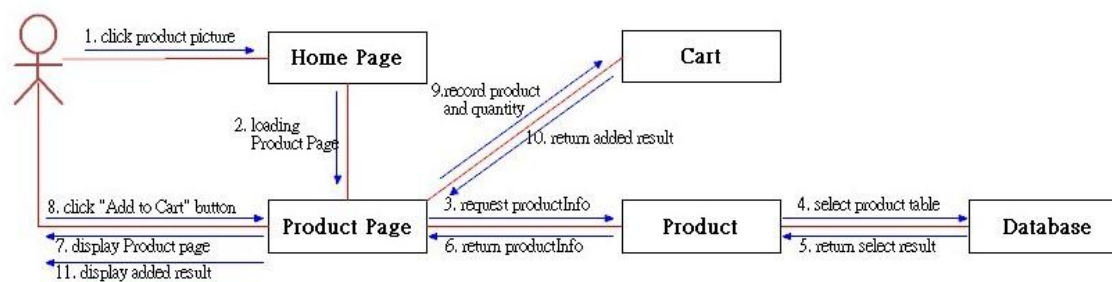
4.4 軟體設計描述



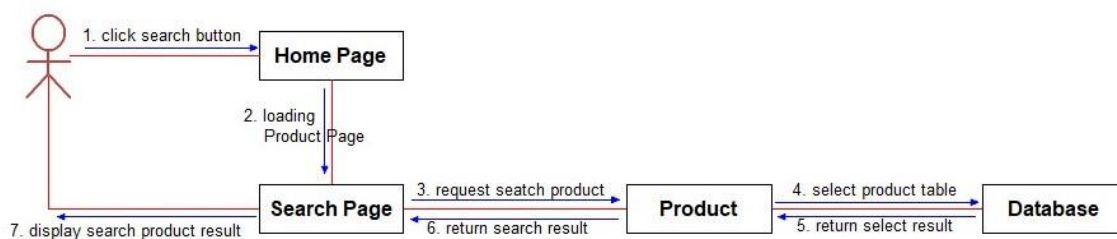
圖十-1：用戶註冊(Collaboration diagram)



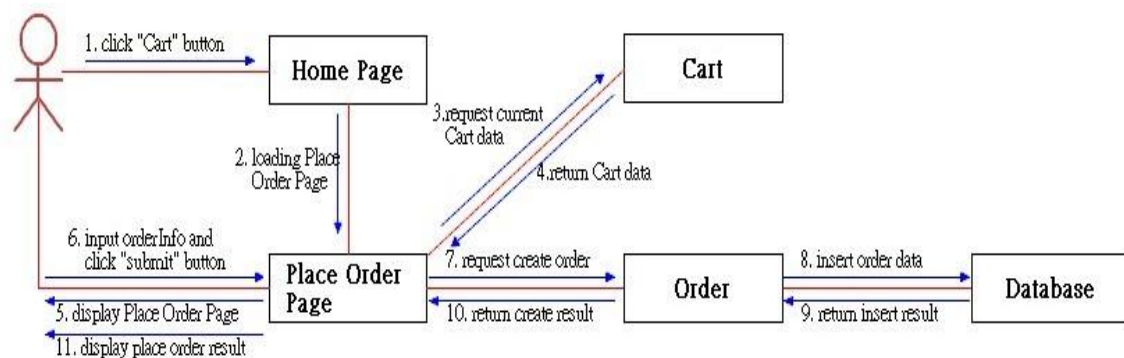
圖十-2：用戶登入(Collaboration diagram)



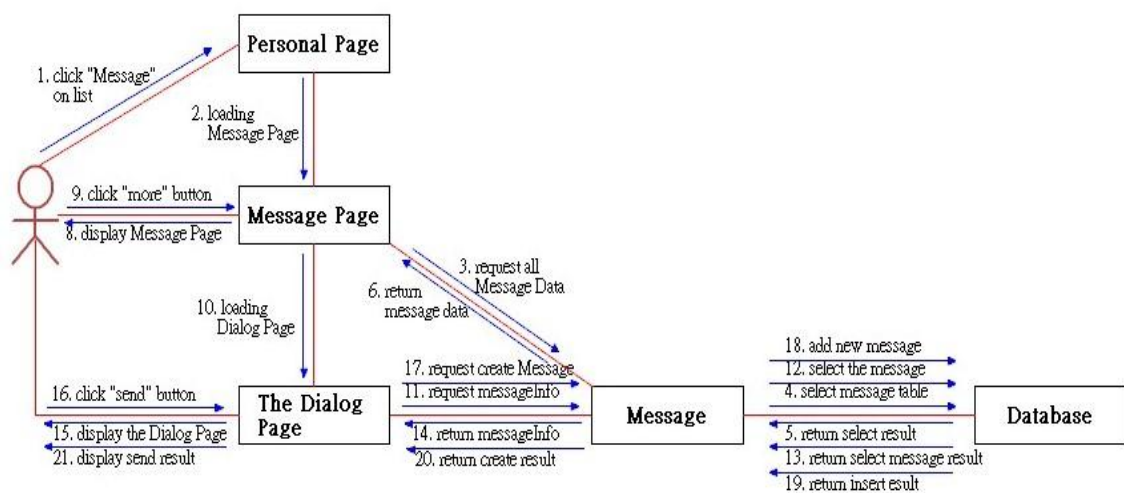
圖十-3：用戶選購商品加入購物車(Collaboration diagram)



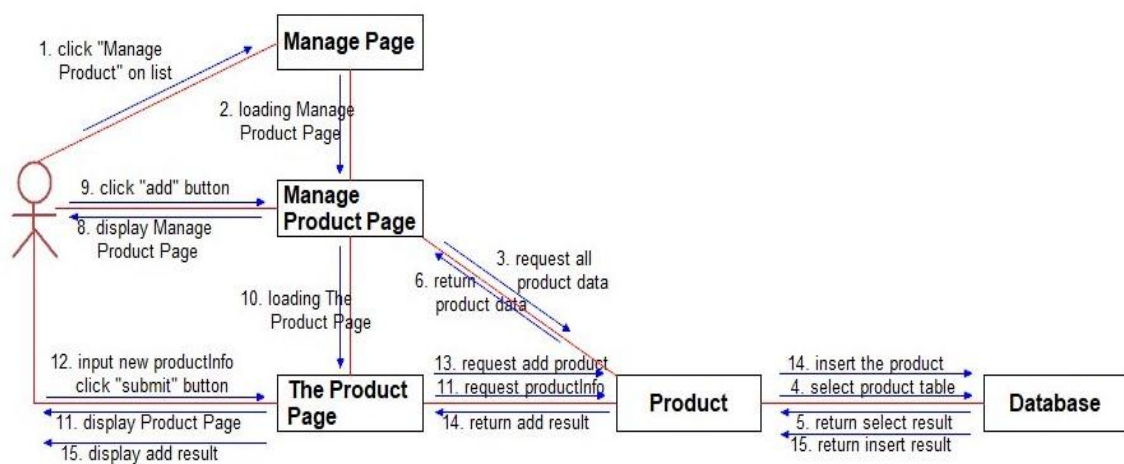
圖十-4：用戶搜尋商品(Collaboration diagram)



圖十-5：用戶下單(Collaboration diagram)



圖十-6：用戶申請客服(Collaboration diagram)



圖十-7：管理員新增商品(Collaboration diagram)

4.5 需求追溯

OOA 需求摘要	OOD 相關類別
帳號申請	Register Page
網站登入	Login Page
商品瀏覽	Home Page
搜尋商品	Home Page
商品選擇	Product Page
查閱商品資訊	Product Page
商品訂購	Order Place Page
聯繫客服	Personal Page
新增商品	Manage Product Page
管理商品	Manage Product Page
管理訂單	Manage Order Page
管理客服	Manage Message Page

4.6 重複使用性

本系統乃是由三個子系統組合而成。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。

五、規格內容追溯

SRS 規格內容需求摘要	對應 SDD 內容段落
SRS Sec. 2-3-A~D	SDD Sec. 2. 3. Fig 五-6
SRS Sec. 2-3-E~F、N	SDD Sec. 2. 3. Fig 五-1~2
SRS Sec. 2-3-G~K	SDD Sec. 2. 3. Fig 五-4
SRS Sec. 2-3-L、R	SDD Sec. 2. 3. Fig 五-8
SRS Sec. 2-3-O~Q	SDD Sec. 2. 3. Fig 五-7