**CSIE4109 Database Systems**

**遊戲銷售平台GAMLAB**

**Final Report**

|  |  |  |
| --- | --- | --- |
| **姓名** | **學號** | **E-mail** |
| **莊 永** | **107590005** | **t107590005@ntut.org.tw** |
| **劉恒育** | **107590007** | **robot881201@gmail.com** |
| **洪平彥**  **韓宗穎**  **康紘郡**  **林琨閔** | **107590010**  **107590018**  **107590031**  **107590033** | **t107590010@ntut.org.tw**  **f4f9115@gmail.com**  **t107590031@ntut.org.tw**  **sadq124329239@gmail.com** |

**Department of Computer Science & Information Engineering**

**National Taipei University of Technology**

**01/05/2021**

目錄 (Table of Contents)

[Section 1 簡介 (Introduction) 1](#_Toc60685422)

[1.1目的 (Purpose) 1](#_Toc60685423)

[1.2系統名稱 (Identification) 1](#_Toc60685424)

[1.3概觀 (Overview) 2](#_Toc60685425)

[1.4符號描述 (Notation Description) 2](#_Toc60685426)

[Section 2 系統 (System) 3](#_Toc60685427)

[2.1系統描述 (System Description) 3](#_Toc60685428)

[2.2操作概念 (Operational Concepts) 4](#_Toc60685429)

[2.3介面需求 (Interface Requirements) 4](#_Toc60685430)

[2.4功能性需求 (Functional Requirements) 5](#_Toc60685431)

[2.5非功能性需求 (Non-Functional Requirements) 5](#_Toc60685432)

[Section 3 資料庫概念設計 (Conceptual Design of the Database) 6](#_Toc60685433)

[3.1 Entity-Relationship (ER) Model 6](#_Toc60685434)

[Section 4 資料庫綱要 (Database Schema) 7](#_Toc60685435)

[4.1 Schema of the Database 7](#_Toc60685436)

[4.2 Expectation of the possible DB operations, frequencies and data volumes 10](#_Toc60685437)

[4.3 Expectation of the size of each DB table 11](#_Toc60685438)

[4.4 SQL Statements Used to Construct the Schema 12](#_Toc60685439)

[4.5 The implementation of tables in target DBMS 14](#_Toc60685440)

[Section 5 Functional Dependencies 15](#_Toc60685441)

[5.1 Functional Dependencies 15](#_Toc60685442)

[Section 6 The Use of the Database System 16](#_Toc60685443)

[6.1 使用者頁面 16](#_Toc60685444)

[Section 7 Suggestions of Database Turning 22](#_Toc60685445)

[Section 8 Additional Queries and Views 23](#_Toc60685446)

[8.1 Database Queries 23](#_Toc60685447)

[Section 9 Conclusions and Future Work 26](#_Toc60685448)

[9.1 Conclusions 26](#_Toc60685449)

[9.2 Future Work 26](#_Toc60685450)

[References 27](#_Toc60685451)

[Appendix 28](#_Toc60685452)

**Section 1** **簡介 (Introduction)**

**1.1目的 (Purpose)**

在這個時代，每個網站基本都會有一套資料庫系統，只要上網，就一定會使用到資料庫系統，但是目前我們都是以「使用者」這個身分去接觸它，很少會以「開發者」的角度去接觸到資料庫系統，在大三的上學期，有了資料庫系統的課程，也因為這次的課程，讓我們可以有機會以「開發者」的角度去建構資料庫與資料庫系統，藉由這次的專案，讓我們不只可以在課堂中學習，也可以透過實作讓我們更了解資料庫與資料庫系統的建構、運作和運用，因此我們經由討論後決定把目標設定為完成一個完整遊戲銷售平台，名稱為：「GAMLAB」，此平台能讓賣家在上架遊戲進行銷售，並能讓買家在平台購買遊戲。

本系統主要目標為：

* 賣家可以對遊戲進行上下架
* 當買家購買時，賣家可以去處理系統產生的訂單
* 買家可以新增遊戲至購物車，也可以從購物車移除遊戲
* 買家對購物車按購買後，將會產生訂單
* 買家可以查看歷史訂單紀錄
* 在買家購買遊戲後，系統會將訂單傳送給賣家

**1.2系統名稱 (Identification)**

本專案將會由一個主系統與多個子系統組成：

主系統：

GAMLAB系統(GAMLAB System, GLS)

子系統:

網頁系統(Webpage Subsystem, WS)

使用者系統(User Subsystem, US)

賣家系統(Seller Subsystem, SS)

買家系統(Buyer Subsystem, BS)

財務系統(Financial Subsystem, FS)

銷售管理系統(Sales Manage Subsystem,SMS)

購物系統(Purchase Subsystem, PCS)

商品系統(Product Subsystem, PS)

資料庫系統(Database Subsystem, DS)

**1.3概觀 (Overview)**

我們這次的專案將實作一個遊戲銷售平台GAMLAB，在這個商店內系統會將使用者分成兩種不同身分，分別為賣家以及買家，這兩種身分將會使用到商店內不同的功能，所以各個身分也將會有不同的資料表去儲存相對應的資訊，如果不同身分之間想要獲取資訊，將會經由關聯式資料庫系統去幫忙獲取，我們採用Apache作為網頁伺服器、MySQL作為關聯式資料庫並且使用PHP來處理伺服器與資料庫的邏輯。

**1.4符號描述 (Notation Description)**

|  |  |
| --- | --- |
| GLS 1.0.0 | GAMLAB System will be labeling with the number 1.0.0. |
| WS 1.1.x | Webpage Subsystem components will be labeling with the number 1.1.x. |
| US 1.2.x | User Subsystem components will be labeling with the number 1.2.x. |
| SS 1.3.x | Seller Subsystem components will be labeling with the number 1.3.x. |
| BS 1.4.x | Buyer Subsystem components will be labeling with the number 1.4.x. |
| FS 1.5.x | Financial Subsystem components will be labeling with the number 1.5.x. |
| SMS 1.6.x | Sales Manage Subsystem components will be labeling with the number 1.6.x. |
| PCS 1.7.x | Purchase Subsystem components will be labeling with the number 1.7.x. |
| PS 1.8.x | Product Subsystem components will be labeling with the number 1.8.x. |
| DS 1.9.x | Database Subsystem components will be labeling with the number 1.9.x. |
| GLS-I-xxx | GLS Interface Requirements |
| GLS-F-xxx | GLS Functional Requirements |
| GLS-N-xxx | GLS Non-Functional Requirements |

**Section 2 系統 (System)**

**2.1系統描述 (System Description)**

本專案將會由一個主系統與多個子系統組成，主系統為GAMLAB系統(GAMLAB System, GLS)，主系統內有許多的子系統，分別為網頁系統(Webpage Subsystem, WS)、使用者系統(User Subsystem, US)、賣家系統(Seller Subsystem, SS)、買家系統(Buyer Subsystem, BS)、財務系統(Financial Subsystem, FS)、銷售管理系統(Sales Manage Subsystem,SMS)、購物系統(Purchase Subsystem, PCS)、商品系統(Product Subsystem, PS)以及資料庫系統(Database Subsystem, DS)。

**2.1.1系統架構圖 (System Context Diagram)**



**2.2操作概念 (Operational Concepts)**

當使用者進入網站之後，網頁系統(Webpage Subsystem, WS)會透過使用者系統(User Subsystem, US)去辨別該使用者為何種身分，不同的身分將會顯示不同的頁面供使用者操作。若為賣家，網頁系統(Webpage Subsystem, WS)會透過賣家系統(Seller Subsystem, SS)顯示銷售管理頁面，若為買家，網頁系統(Webpage Subsystem, WS)會透過買家系統(Seller Subsystem, SS)獲取買家身分並顯示商店頁面並有購物車、訂單紀錄以及儲值等相關功能，若使用者未登入，網頁系統(Webpage Subsystem, WS)則會顯示商品頁面與登入按鈕。

**2.3介面需求 (Interface Requirements)**

**2.3.1使用者介面需求 (User Interfaces Requirements)**

|  |  |
| --- | --- |
| GLS-I-001 | 網站首頁提供使用者登入頁面、商品頁面瀏覽。 |
| GLS-I-002 | 使用者登入後，網站會依照不同身分的使用者提供不同的頁面。 |

**2.3.2外部介面需求 (External Interface Requirements)**

|  |  |
| --- | --- |
| GLS-I-003 | 使用者透過瀏覽器使用HTTP通訊瀏覽網頁。 |
| GLS-I-004 | 網頁透過jQuery傳送指令至PHP後，PHP再向資料庫發送請求。 |

**2.3.3內部介面需求 (Internal Interface Requirements)**

|  |  |
| --- | --- |
| GLS-I-005 | SS能透過PS獲取商品的各式資訊 |
| GLS-I-006 | BS、SS能透過FS進行財務檢視、錢包管理 |
| GLS-I-007 | BS、SS能透過US對其帳號進行管理的功能 |
| GLS-I-008 | WS可以透過US去獲取使用者身分顯示不同的頁面 |
| GLS-I-009 | WS透過PS獲取商品資訊後顯示在頁面上 |
| GLS-I-010 | BS可以透過PCS去購買商品 |
| GLS-I-011 | SS可以透過SMS對商品進行上下架、檢視商品銷售量及相關資訊 |
| GLS-I-012 | DS要能正確的從資料庫獲取資料 |

**2.4功能性需求 (Functional Requirements)**

|  |  |
| --- | --- |
| GLS-F-001 | 商品功能，提供商品上下架、新增商品、檢視銷售量。 |
| GLS-F-002 | 財務功能，提供賣家檢視獲利情況，對買家及賣家提供帳戶金額檢視、帳戶金額管理。 |
| GLS-F-003 | 網頁功能，提供各種身分所需要使用的頁面以及商品顯示、商品瀏覽、商品分類。 |
| GLS-F-005 | 購物功能，買家能將商品加入購物車後購買商品。 |
| GLS-F-006 | 銷售功能，賣家可以處理買家購買後產生的訂單。 |
| GLS-F-007 | 管理功能，訂單完成後，商品銷售量會自動更新。 |
| GLS-F-008 | 財務功能，買家可以檢視餘額並擁有儲值功能，賣家能檢視獲利狀況。 |
| GLS-F-006 | 資料庫功能，提供新增、刪除、修改資料庫等相關功能。 |

**2.5非功能性需求 (Non-Functional Requirements)**

**2.5.1****效能需求 (Performance Requirements)**

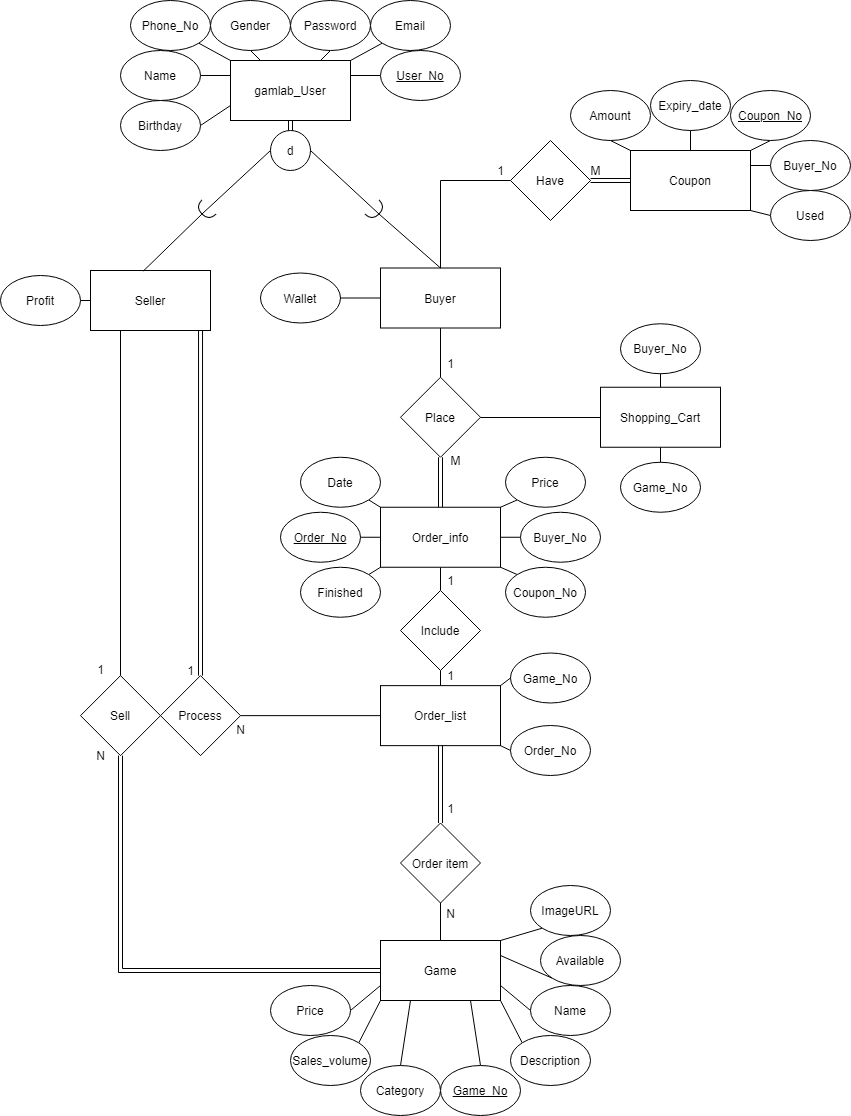
|  |  |
| --- | --- |
| GLS-N-001 | 頁面響應應小於3秒 |
| GLS-N-002 | 當系統出現在錯誤時，應顯示錯誤頁面並讓使用者返回上一頁 |

**2.5.2測試需求 (Test Requirements)**

|  |  |
| --- | --- |
| GLS-N-00 | 主系統與所有子系統需經過完善的測試才能讓使用者使用 |

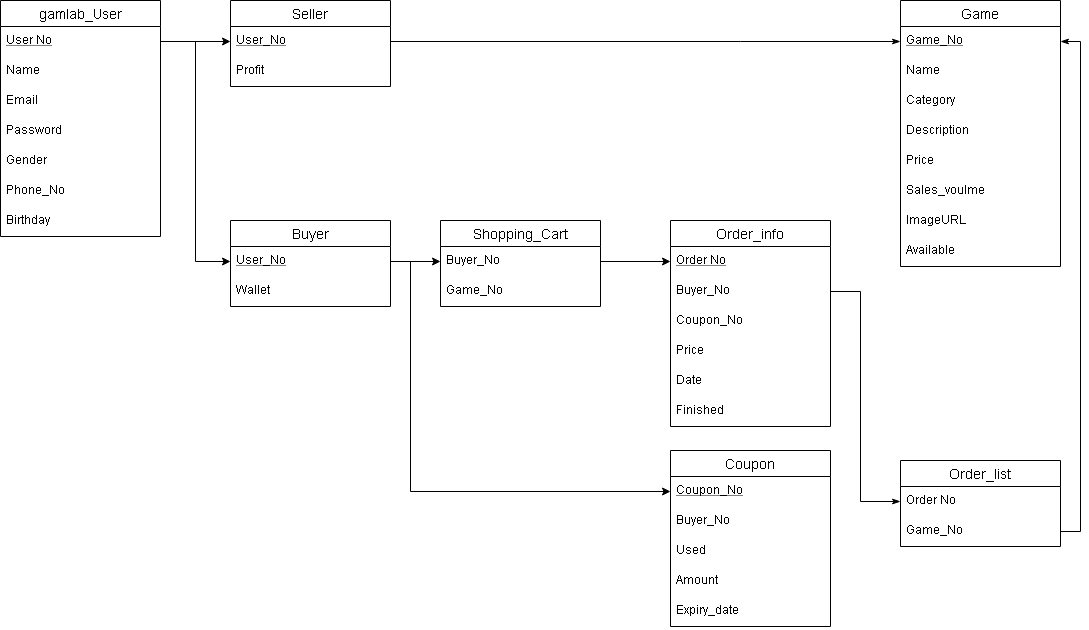
**Section 3 資料庫概念設計 (Conceptual Design of the Database)**

**3.1 Entity-Relationship (ER) Model**

****

**Section 4 資料庫綱要 (Database Schema)**

**4.1 Schema of the Database**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **gamlab\_User** | | | | |
| **Description:** 存放所有會員的基本資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| User\_No | varchar(5) | Primary | False | 會員編號 |
| Email | varchar(50) |  | False | 會員信箱 |
| Password | varchar(30) |  | False | 會員密碼 |
| Name | varchar(10) |  | False | 會員姓名 |
| Birthday | DATE |  | False | 會員生日 |
| Phone\_No | varchar(10) |  | False | 會員電話 |
| Gender | varchar(1) |  | True | 會員性別 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Buyer** | | | | |
| **Description:** 存放買家的基本資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| User\_No | varchar(5) | Primary | False | 買家編號 |
| Wallet | Integer |  | False | 錢包餘額 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Seller** | | | | |
| **Description:** 存放賣家的基本資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| User\_No | varchar(5) | Primary | False | 賣家編號 |
| Profit | Integer |  | False | 收益金額 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order\_info** | | | | |
| **Description:** 存放訂單資訊的資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| Order\_No | varchar(5) | Primary | False | 訂單編號 |
| Buyer\_No | varchar(5) | Foreign | False | 買家編號 |
| Coupon No | varchar(5) | Foreign | True | 優惠券編號 |
| Date | DATE |  | False | 購買日期 |
| Price | Integer |  | False | 消費金額 |
| Finished | Boolean |  | False | 訂單狀態 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order\_list** | | | | |
| **Description:** 存放訂單列表的資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| Order\_No | varchar(5) |  | False | 訂單編號 |
| Game\_No | varchar(5) |  | False | 遊戲編號 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Coupon** | | | | |
| **Description:** 存放優惠券的資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| Coupon\_No | varchar(5) | Primary | False | 優惠券編號 |
| Buyer\_No | varchar(5) | Foreign | False | 買家編號 |
| Used | Boolean |  | False | 使用狀態 |
| Amount | Integer |  | False | 折價金額 |
| Expiry\_date | DATE |  | False | 使用期限 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Game** | | | | |
| **Description:** 存放遊戲的資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| Game\_No | varchar(5) | Primary | False | 遊戲編號 |
| Price | Integer |  | False | 遊戲價錢 |
| Sales\_volume | Integer |  | False | 銷售數量 |
| Category | varchar(10) |  | False | 遊戲類別 |
| Name | varchar(50) |  | False | 遊戲名稱 |
| Description | varchar(500) |  | False | 遊戲簡介 |
| ImageURL | varchar(50) |  | False | 遊戲圖片路徑 |
| Available | Boolean |  | False | 上架狀態 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shopping\_Cart** | | | | |
| **Description:** 存放購物車的資料 | | | | |
| Attribute | Type | Key | Nullable | Description |
| Buyer\_No | varchar(5) |  | False | 買家編號 |
| Game\_No | varchar(5) |  | False | 遊戲編號 |

**4.2 Expectation of the possible DB operations, frequencies and data volumes**

|  |  |  |  |
| --- | --- | --- | --- |
| Table | 可能操作 | 使用頻率 | 系統負擔 |
| gamlab\_User | 驗證登入身分 | 500 | 2000次Query/day |
| gamlab\_User | 增加會員 | 20 | 50次Insert/day |
| gamlab\_User | 更改資料 | 50 | 100次Update/day |
| Buyer | 儲值或扣款 | 200 | 1000次Update/day |
| Buyer | 顯示錢包金額 | 200 | 2000次Query/day |
| Seller | 顯示收益金額 | 10 | 1000次Query/day |
| Coupon | 新增優惠券 | 5 | 10次Insert/day |
| Coupon | 使用優惠券 | 20 | 50次Update/day |
| Coupon | 顯示優惠券 | 20 | 50次Query/day |
| Game | 新增遊戲 | 10 | 50次Insert/day |
| Game | 刪除遊戲 | 2 | 10次Delete/day |
| Game | 上/下架遊戲 | 5 | 50次Update/day |
| Game | 顯示遊戲 | 1000 | 2000次Query/day |
| Order\_info | 顯示訂單 | 100 | 500次Query / day |
| Order\_info | 新增訂單 | 50 | 200次Insert/ day |
| Order\_info | 完成訂單 | 50 | 200次Update/ day |
| Order\_list | 新增訂單內遊戲 | 100 | 400次Insert / day |
| Order\_list | 顯示訂單內遊戲 | 200 | 1000次Query/day |
| Shopping\_Cart | 新增購物車遊戲 | 100 | 400次 Insert /day |
| Shopping\_Cart | 刪除購物車遊戲 | 75 | 150次 Delete/day |

**4.3 Expectation of the size of each DB table**

|  |  |
| --- | --- |
| Table | Size (MBytes) |
| User | 32 |
| Buyer | 29 |
| Seller | 3 |
| Order\_info | 78 |
| Order\_list | 391 |
| Coupon | 23 |
| Game | 16 |
| Shopping\_Cart | 156 |

**4.4 SQL Statements Used to Construct the Schema**

**User table**

CREATE TABLE gamlab\_User (

User\_No varchar(5) NOT NULL,

Email varchar(50) NOT NULL,

Password varchar(30) NOT NULL,

Name varchar(10) NOT NULL,

Birthday DATE NOT NULL,

Phone\_No varchar(10) NOT NULL,

Gender varchar(1) NULL,

PRIMARY KEY(User\_No)

);

**Buyer table**

CREATE TABLE Buyer (

User\_No varchar(5) NOT NULL,

Wallet INT NOT NULL,

PRIMARY KEY(User\_No)

);

**Seller table**

CREATE TABLE Seller (

User\_No varchar(5) NOT NULL,

Profit INT NOT NULL,

PRIMARY KEY(User\_No)

);

**Coupon table**

CREATE TABLE Coupon (

Coupon\_No varchar(5) NOT NULL,

Buyer\_No varchar(5) NOT NULL,

Amount INT NOT NULL,

Used Boolean NOT NULL,

Expiry\_date DATE NOT NULL,

PRIMARY KEY(Coupon\_No)

);

**Game table**

CREATE TABLE Game (

Game\_No varchar(5) NOT NULL,

Price INT NOT NULL,

Sales\_volume INT NOT NULL,

Category varchar(10) NOT NULL,

Name varchar(50) NOT NULL,

Description varchar(500) NOT NULL,

ImageURL varchar(50) NOT NULL,

Available Boolean NOT NULL,

PRIMARY KEY(Game\_No)

);

**Order\_info table**

CREATE TABLE Order\_info (

Order\_No varchar(5) NOT NULL,

Buyer\_No varchar(5) NOT NULL,

Coupon\_No varchar(5) NULL,

Date DATE NOT NULL,

Price INT NOT NULL,

Finished Boolean NOT NULL,

PRIMARY KEY(Order\_No)

);

**Order\_list table**

CREATE TABLE Order\_list (

Order\_No varchar(5) NOT NULL,

Game\_No varchar(5) NOT NULL

);

**Shopping\_Cart table**

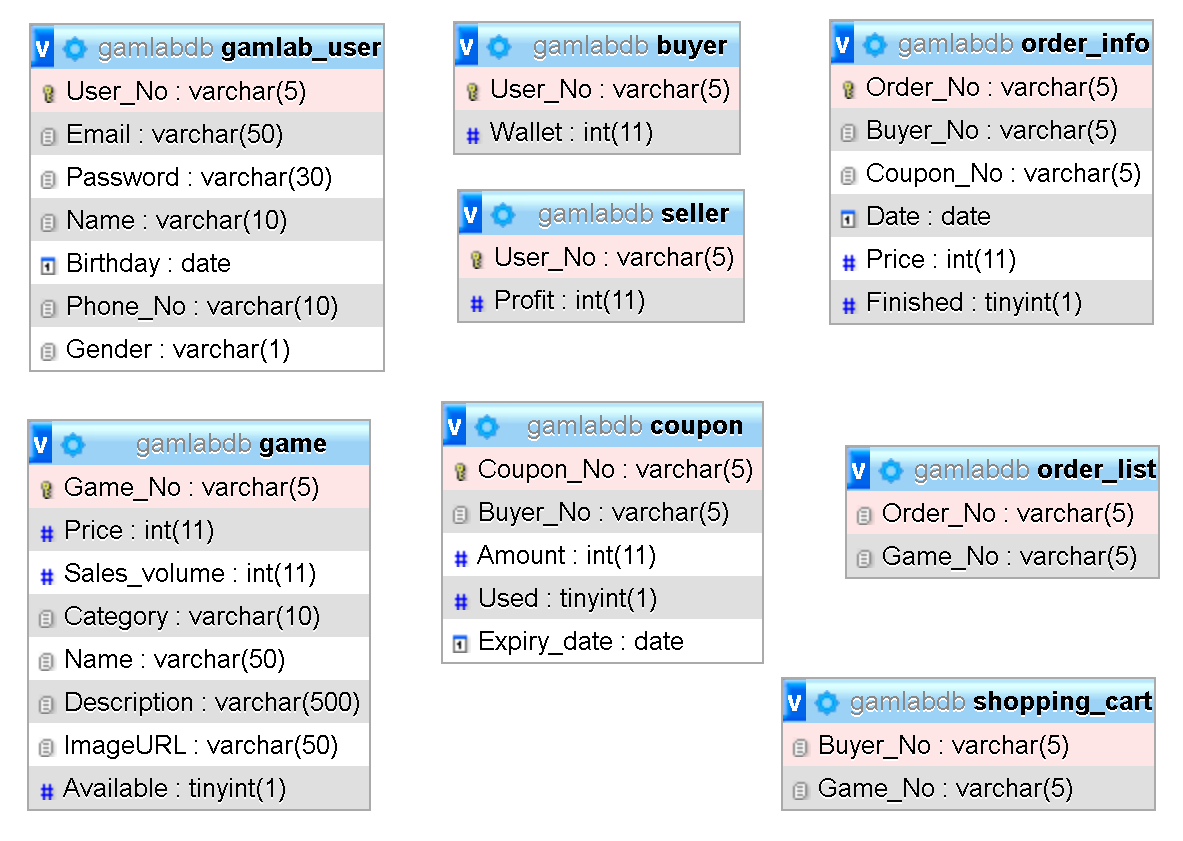
CREATE TABLE Shopping\_Cart (

Buyer\_No varchar(5) NOT NULL,

Game\_No varchar(5) NOT NULL

);

**4.5 The implementation of tables in target DBMS**



**Section 5 Functional Dependencies**

**5.1 Functional Dependencies**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User** | | | | | | |
| User\_No | Email | Password | Name | Birthday | Phone\_No | Gender |

|  |  |
| --- | --- |
| **Buyer** | |
| User\_No | Wallet |

|  |  |
| --- | --- |
| **Seller** | |
| User\_No | Profit |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order\_info** | | | | | |
| Buyer\_No | Order\_No | Coupon\_No | Date | Price | Finished |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Coupon** | | | | |
| Coupon\_No | Buyer\_No | Used | Amount | Expiry\_date |

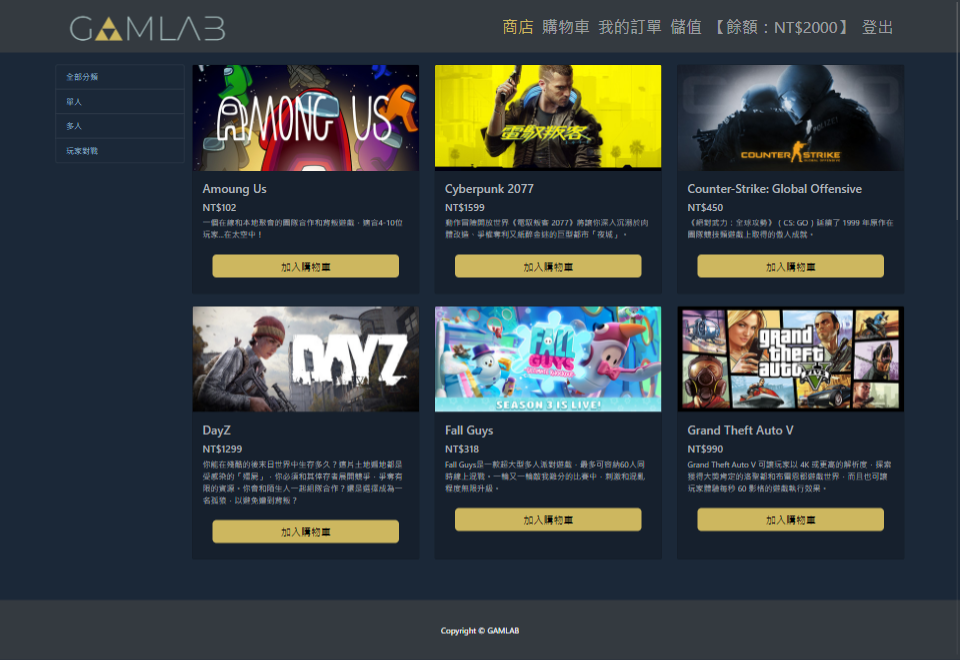
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Game** | | | | | | |
| Game\_No | Price | Sales\_volume | Category | Name | Description | ImageURL |

|  |  |
| --- | --- |
| **Game** | |
| Game\_No | Available |

**Section 6 The Use of the Database System**

**6.1 使用者頁面**

1.登入畫面

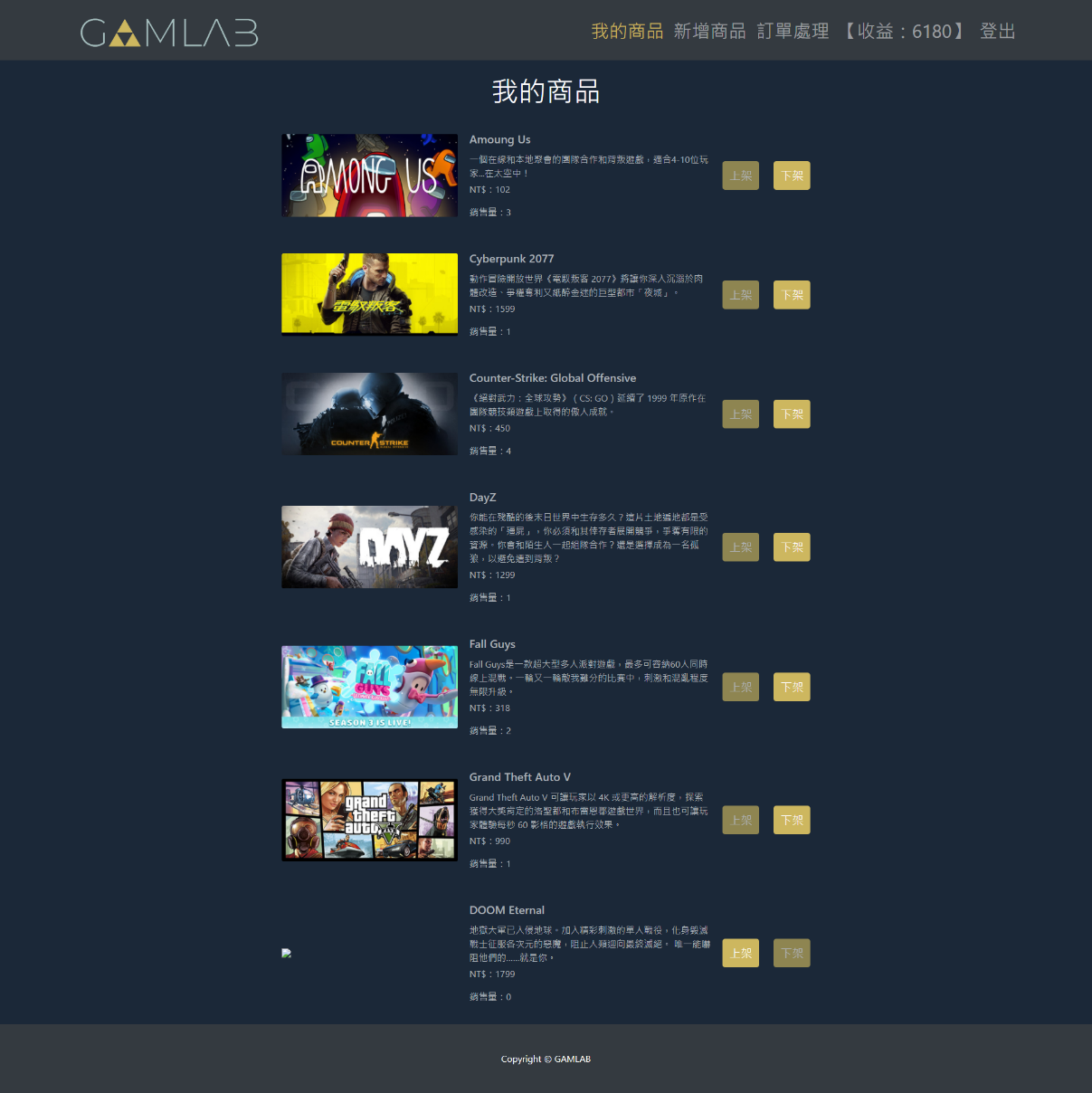
2.買家首頁

****3.買家購物車

4.買家訂單

5.買家儲值



6.賣家商品

7.賣家新增商品



8.賣家訂單處理

一張含有 文字, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

**Section 7** **Suggestions of Database Turning**

在本次Database實作當中，有遇到一些技術層面的問題，例如：由於在Apache平台上運行會有Cache的問題，因此當HTML在引入CSS與Javascript的連結時，我們採用在連結後方加上目前時間作為隨機亂數來避免Cache；在PHP偵錯回傳值的部分，一開始還不知道要如何print出來查看變數或是錯誤訊息，後來研究發現會顯示在Network中.php檔案的Preview標籤頁內；而在使用者登入狀態保存的部分，我們採用PHP Session的技術，來保存使用者登入的角色與帳號，並依據當前角色來產生對應的頁面；在前端的部分，會出現頁面在沒有全展開的情況下，Footer的位置會位移，後來加入判斷在頁面高度未達100%時，加入style:height 100%的設定來解決；最後在按鈕特效的部分，由於要達到點擊出現光暈漸淡的效果，我們會在<button>的標籤旁，加入<span>的標籤，將之設定z-index:-1來重疊於按鈕物件下，並設定光暈漸淡的動畫，否則直接在<button>上設定，會造成按鈕字樣隨著漸淡而消失。

**Section 8 Additional Queries and Views**

**8.1 Database Queries**

1. 網頁系統

顯示當前在架上的所有遊戲

SELECT \* FROM game WHERE game.Available=1

顯示使用者所選擇的分類的遊戲

SELECT \* FROM game WHERE game.Category='$category' and game.Available=1

2. 使用者系統

登入驗證

SELECT \* FROM gamlab\_user   
WHERE Email='$email' AND Password='$password

辨別會員身分

SELECT \* FROM seller WHERE User\_No={$\_SESSION['userno']}

3. 購物系統

取得購物車內遊戲數量

SELECT Count(Game\_No) FROM shopping\_cart WHERE Buyer\_No=$userno

新增遊戲至購物車

INSERT INTO shopping\_cart values('$userId', '$gameId')

顯示購物車內遊戲

SELECT game.Name, game.Price, game.Description, game.ImageURL

FROM game, shopping\_cart

WHERE shopping\_cart.Buyer\_No = '$userno'

AND shopping\_cart.Game\_No = game.Game\_No

顯示可使用的優惠券

SELECT \* FROM coupon

WHERE coupon.Buyer\_No = '$userno' AND coupon.used = 0

取得優惠券折抵金額

SELECT Amount FROM coupon WHERE Coupon\_No = $couponid

更新優惠券使用狀態

UPDATE coupon SET used=1 WHERE Coupon\_NO='$couponid'

刪除購物車

DELETE FROM shopping\_cart WHERE Buyer\_No='$userno'

4. 銷售管理系統

取得訂單

SELECT \* From order\_info WHERE Buyer\_No='$userno'

取得優惠券折抵金額

SELECT Amount From coupon

WHERE Coupon\_No={$row\_result['Coupon\_No']}

取得訂單內的遊戲

SELECT game.Game\_No, Name, Price, Description, ImageURL

FROM order\_list, game

WHERE order\_list.Order\_No = '$orderno'

AND order\_list.Game\_No = game.Game\_No

產生訂單編號

SELECT Order\_No FROM order\_info

WHERE

Order\_No=(SELECT MAX(CAST(Order\_No as SIGNED)) FROM order\_info)

產生訂單

INSERT INTO

order\_info VALUES('$orderno', '$userno', '$couponid', '$date', '$total', False)

將購物車內遊戲新增至order\_list

INSERT INTO order\_list VALUES('$orderno', '$gameno')

產生遊戲編號

SELECT Game\_No From game

WHERE

Game\_No = (SELECT MAX(CAST(Game\_No as SIGNED)) FROM game)

獲取賣家販售的商品

SELECT game.Name, game.Price, game.Description, game.Sales\_volume, game.ImageURL, game.Available, game.Game\_No FROM game

取得所有訂單編號

SELECT order\_info.Finished,order\_list.Order\_No

FROM order\_list,order\_info

WHERE order\_list.Order\_No=order\_info.Order\_No

GROUP by order\_list.Order\_No

取得所有訂單資訊

SELECT \* FROM order\_info WHERE order\_info.Order\_No='$orderno'

取得待處理訂單內商品資訊

SELECT order\_info.Date, order\_info.Price as Order\_Price, order\_list.Game\_No, game.Name,game.Description,game.ImageURL,game.Price

FROM order\_info, order\_list, game

WHERE order\_list.Order\_No='$orderno'

AND order\_info.Order\_No=order\_list.Order\_No

AND order\_list.Game\_No=game.Game\_No

GROUP by order\_list.Game\_No

更改訂單狀態

UPDATE order\_info SET Finished = true WHERE order\_info.Order\_No = '$orderno'

5. 商品系統

新增遊戲

INSERT INTO game

VALUES

('$gameno', '$userno', '$price', '0', '$category', '$name', '$description', '$link', TRUE)

賣家下架商品

UPDATE game SET Available = False WHERE game.Game\_No = '$gameno'

賣家上架商品

UPDATE game SET Available = True WHERE game.Game\_No = '$gameno'

取得商品銷量

SELECT order\_list.Game\_No,COUNT(\*) as Amount FROM order\_list

WHERE order\_list.Order\_No='$orderno'

GROUP BY order\_list.Game\_No

更新商品銷量

UPDATE game SET Sales\_volume = Sales\_volume+$amount

WHERE game.Game\_No = '$gameno'

6. 財務系統

獲取買家錢包餘額

SELECT Wallet FROM buyer WHERE User\_No={$\_SESSION['userno']}

獲取賣家當前收益

SELECT Profit FROM seller WHERE User\_No={$\_SESSION['userno']}

取得利潤

SELECT order\_info.Price FROM order\_info WHERE order\_info.Order\_No='$orderno'

更新總收入

UPDATE seller SET Profit = Profit + $profit WHERE seller.User\_No = '$userno'

買家儲值後更新餘額

UPDATE buyer SET Wallet = $wallet+$value

WHERE User\_No={$\_SESSION['userno']}

買家消費後更新餘額

UPDATE buyer SET Wallet = $wallet WHERE User\_No={$\_SESSION['userno']}

**Section 9 Conclusions and Future Work**

**9.1** **Conclusions**

透過完成這次Database的專題,讓我們理解道從後端到前端一整套系統的設計:原本在老師的課堂中無法理解和使用的理論和知識,在這個專題中都我們基本上都將他了解,例如:ER Model,系統架構圖。

在課堂上我會想說我們為何要畫出ER Model圖才能進行實作，應該只須要想好需求後直接去進行實作，但在實作時會發現這個想法在簡單的Project中能順利完成，但在比較大的Project中，會因為關係複雜，使你在實作時會發生一直搞不懂哪個需求是需要優先進行的，哪個需求是要透過另一個需求來完成的，導致無法專心在寫程式，會造成一邊寫一邊設計，甚至有可能一個地方想錯,整個程式或架構就要去進行修改，這樣會使我們浪費非常多的時間在進行偵錯，而畫出ER Model圖可以讓我們很清楚的知道各個Entity之間的關係，可以讓我們更快速的進行進行理解實作；而系統架構圖的部分，我們一開始設計時也搞不太清楚各個系統之間的關係，以致於我們在實作時架構圖有更改過許多次，幸好大部分的架構都是OK的，一些小地方或者不足之處在經過大家的討論後都順利的解決。

最後，我認為在這次實作中我們因為第一次接觸這類專案，我們對使用方式的不熟悉，使我們做出來的Project沒有很完整，裡面還是缺少了一些功能，而且也因為時間的關係我們沒有那麼多的時間去測試系統的效能和完整度，造成我們的系統會有一些小小的瑕疵，這是我們需要去測試和改進的地方；但也因為這些錯誤的地方，讓我們學習到如何成功地建立整個資料庫系統，也讓我們對老師課堂上所教的資料庫系統相關的知識有更深入的了解，相信在下次再碰到資料庫時我們會更加有效率的去應用這些東西，使我們的Project更加的完整與完美。

**9.2** **Future Work**

我們這次的遊戲分類，因為對前端操作沒有那麼熟悉，在一開始的排版和設計沒有考慮那麼詳細，所以賣家新增商品時只有三個分類可以選，若有許多遊戲時，會使買家在購買遊戲時無法迅速找到想要購買的遊戲，而我們未來也希望在首頁中加入熱銷排行榜，讓買家更清楚的了解到現在最熱門的是甚麼遊戲，讓買家更方便的去挑選喜愛的遊戲。

我們原本在賣家的部分原本也希望可以去計算個每月收入，讓賣家能清楚了解到一年中大概那些月份收入會最高，哪些是淡季，進而調整銷售策略，但是後來因為時間的關係，只計算了賣家當前所有遊戲銷售的總收入，並沒有依照月份進行統計，希望未來能將這些缺點一一的去改善。

**References**

* Steam

<https://store.steampowered.com/>

* Bootstrap

<https://getbootstrap.com/docs/5.0/>

* XAMPP

<https://www.apachefriends.org/zh_tw/index.html>

* Draw.io

<https://app.diagrams.net/>

**Appendix**

**Appendix A**

SQL statements for populating database

新增使用者

INSERT INTO gamlab\_user

VALUES('00001', 'buyer@gamlab.com', 'buyer', 'buyer', '2021-01-01', '0912345678', 'M');

INSERT INTO gamlab\_user

VALUES('00002', 'seller@gamlab.com', 'seller', 'seller', '2021-01-01', '0912345678', 'M');

新增買家

INSERT INTO buyer VALUES('00001', 2000);

新增賣家

INSERT INTO seller VALUES('00002', 852);

新增折價券

INSERT INTO coupon VALUES('00001', '00001',150, True, '2021-02-01');

INSERT INTO coupon VALUES('00002', '00001',300, False, '2021-03-01');

INSERT INTO coupon VALUES('00003', '00001',250, False, '2021-03-01');

新增訂單資訊

INSERT INTO order\_info VALUES('00001', '00001', NULL, '2020-12-24', 552, True);

INSERT INTO order\_info VALUES('00002', '00001', '00001', '2020-12-25', 300, True);

INSERT INTO order\_info VALUES('00003', '00001', NULL, '2020-12-31', 4758, False);

新增訂單內遊戲

INSERT INTO order\_list VALUES('00001', '00001');

INSERT INTO order\_list VALUES('00001', '00003');

INSERT INTO order\_list VALUES('00002', '00003');

INSERT INTO order\_list VALUES('00003', '00001');

INSERT INTO order\_list VALUES('00003', '00002');

INSERT INTO order\_list VALUES('00003', '00003');

INSERT INTO order\_list VALUES('00003', '00004');

INSERT INTO order\_list VALUES('00003', '00005');

INSERT INTO order\_list VALUES('00003', '00006');

新增遊戲

INSERT INTO game VALUES('00001', 102, 1, 'multi', 'Amoung Us', '一個在線和本地聚會的團隊合作和背叛遊戲，適合4-10位玩家...在太空中！', './img/Among Us.jpg', True);

INSERT INTO game VALUES('00002', 1599, 1, 'single', 'Cyberpunk 2077', '動作冒險開放世界《電馭叛客 2077》將讓你深入沉溺於肉體改造、爭權奪利又紙醉金迷的巨型都市「夜城」。', './img/Cyberpunk 2077.jpg', True);

INSERT INTO game VALUES('00003', 450, 2, 'battle', 'Counter-Strike: Global Offensive', '《絕對武力：全球攻勢》（CS: GO）延續了 1999 年原作在團隊競技類遊戲上取得的傲人成就。', './img/Counter-Strike： Global Offensive.jpg', True);

INSERT INTO game VALUES('00004', 1299, 0, 'battle', 'DayZ', '你能在殘酷的後末日世界中生存多久？這片土地遍地都是受感染的「殭屍」，你必須和其倖存者展開競爭，爭奪有限的資源。你會和陌生人一起組隊合作？還是選擇成為一名孤狼，以避免遭到背叛？', './img/DayZ.jpg', True);

INSERT INTO game VALUES('00005', 318, 0, 'multi', 'Fall Guys', 'Fall Guys是一款超大型多人派對遊戲，最多可容納60人同時線上混戰。一輪又一輪敵我難分的比賽中，刺激和混亂程度無限升級。', './img/Fall Guys.jpg', True);

INSERT INTO game VALUES('00006', 990, 0, 'battle', 'Grand Theft Auto V', 'Grand Theft Auto V 可讓玩家以 4K 或更高的解析度，探索獲得大獎肯定的洛聖都和布雷恩郡遊戲世界，而且也可讓玩家體驗每秒 60 影格的遊戲執行效果。', './img/Grand Theft Auto V.jpg', True);