

遊戲搜尋系統(Game Search System)

系統需求規格書

Software Requirements Specification (SRS)

Version: 2.0

姓名	學號	E-mail
葉凱毅	106590047	khaiyi5893@gmail.com
尹秉豪	106590014	zx99519567@gmail.com
劉政威	106590039	wei.141227@gmail.com
陳峻暉	106590017	will84568035@gmail.com
黃宥勛	106590002	z59141483@gmail.com
陳茗洋	106590038	a24911920@gmail.com

**Department of Computer Science & Information Engineering
National Taipei University of Technology**

1/1/2020

目錄 (Table of Contents)

1.1 目的 (Purpose).....	1
1.2 系統名稱 (Identification).....	1
1.3 概觀 (Overview).....	1
1.4 符號描述 (Notation Description).....	2
Section 1 系統(System).....	3
2.1 系統描述 (System Description).....	3
2.1.1 系統架構圖 (System Context Diagram).....	3
2.2 操作概念 (Operational Concepts).....	4
2.3 設計限制 (Design, Data, and Implementation Constrains).....	4
2.4 技術限制 (Technological Limitations).....	4
2.5 介面需求 (Interface Requirements).....	4
2.5.1 使用者介面需求 (User Interfaces Requirements).....	4
2.5.2 外部介面需求 (External Interface Requirements).....	4
2.5.3 內部介面需求 (Internal Interface Requirements).....	5
2.6 功能性需求 (Functional Requirements).....	5
2.7 非功能性需求 (Non-Functional Requirements).....	5
2.7.1 效能需求 (Performance Requirements).....	5
2.7.2 測試需求 (Test Requirements).....	6
2.8 其他需求 (Other Requirements).....	6
2.8.1 環境需求 (Environmental Requirement).....	6
2.8.2 安裝需求 (Installation Requirement).....	6
Section 2 資料庫概念設計(Conceptual Design of the Database).....	7
3.1 Entity-Relationship (ER) Model.....	7
3.2 Data Dictionary and Business Rules.....	8
Section 3 邏輯資料庫綱要(Logical Database Schema).....	9
4.1 Schema of the Database.....	9
4.2 Expectation of the possible DB operations, frequencies and data volumes.....	12
4.3 SQL Statements Used to Construct the Schema.....	13
4.4 The implementation of tables in target DBMS.....	16
Section 4 Functional Dependencies and Database Normalization.....	17
5.1 Functional Dependencies.....	17
Section 5 The use of the Database System.....	18
6.1 System Installation Description.....	18
6.1.1 MySQL.....	18
6.1.2 Heroku 設定.....	19
6.2 The Use of the System.....	21

6.2.1 使用者頁面.....	21
6.2.2 管理者頁面.....	28
Section 6 Suggestions of Database Tuning.....	33
Section 7 Additional Queries and Views.....	34
8.1 Database Queries.....	34
8.2 Database Views.....	35
Section 8 Conclusions and Future Work.....	36
9.1 Conclusions.....	36
9.2 Future Work.....	38
Glossary.....	39
References.....	40
Appendix.....	41
Appendix A.....	41

簡介 (Introduction)

1.1 目的 (Purpose)

想要更深入了解資料庫設計方式、技術運用、需求分析和效能評估等相關知識，此次實作遊戲搜尋系統（GSS）是個很好的學習機會。

為了能讓使用者能更輕鬆地找到不同遊戲種類，像是遊戲玩家想要尋找特定種類的遊戲，或是家長要讓孩子游玩適合年齡段的遊戲，因此開發了一套大家簡單搜尋的遊戲搜尋系統。

本系統主要目標：

- 顧客可以瀏覽、搜尋、購買遊戲
- 管理者可以管理遊戲的上下架與價格狀況
- 顧客可以查看目前的購買狀況
- 顧客可以為遊戲評分

1.2 系統名稱 (Identification)

主系統為：

遊戲搜尋系統（Game Search System，**GSS**）

各子系統分別為：

會員管理子系統（Account and Member Management Subsystem，**AMMS**）

購物車管理子系統（Shopping Cart Management Subsystem，**SHMS**）

遊戲管理子系統（Games Management Subsystem, **GMS**）

財務管理子系統（Financial Management Subsystem，**FMS**）

遊戲評分、評語管理子系統（Score and Comment Management Subsystem，**SCMS**）

結賬管理子系統（Bill Management Subsystem，**BMS**）

資料庫子系統（Database Subsystem，**DBS**）

1.3 概觀 (Overview)

資料庫系統是資訊工程當中相當重要的一個環節，尤其在未來雲端的發展中很重要，而本專案利用 MySQL 作為 DBMS，它具有以下優點：便宜、高度最佳化、好學且操作簡單，基於以上幾個優點，我們在專案的設計上採用 MySQL 做為資料庫的系統，而透過這次的專案，讓我們能夠更加了解資料庫的設計與運用。

1.4 符號描述 (Notation Description)

GSS 1.0.0	The GSS system will be labeled with the number 1.0.0.
AMMS 1.1.n	The AMMS components will be labeled with the number 1.1.n.
SHMS 1.2.n	The SHMS components will be labeled with the number 1.2.n.
GMS 1.3.n	The GMS components will be labeled with the number 1.3.n.
FMS 1.4.n	The FMS components will be labeled with the number 1.4.n.
SCMS 1.5.n	The SCMS components will be labeled with the number 1.5.n.
BMS 1.6.n	The BMS components will be labeled with the number 1.6.n.
DBS 1.7.n	The DBS components will be labeled with the number 1.7.n.

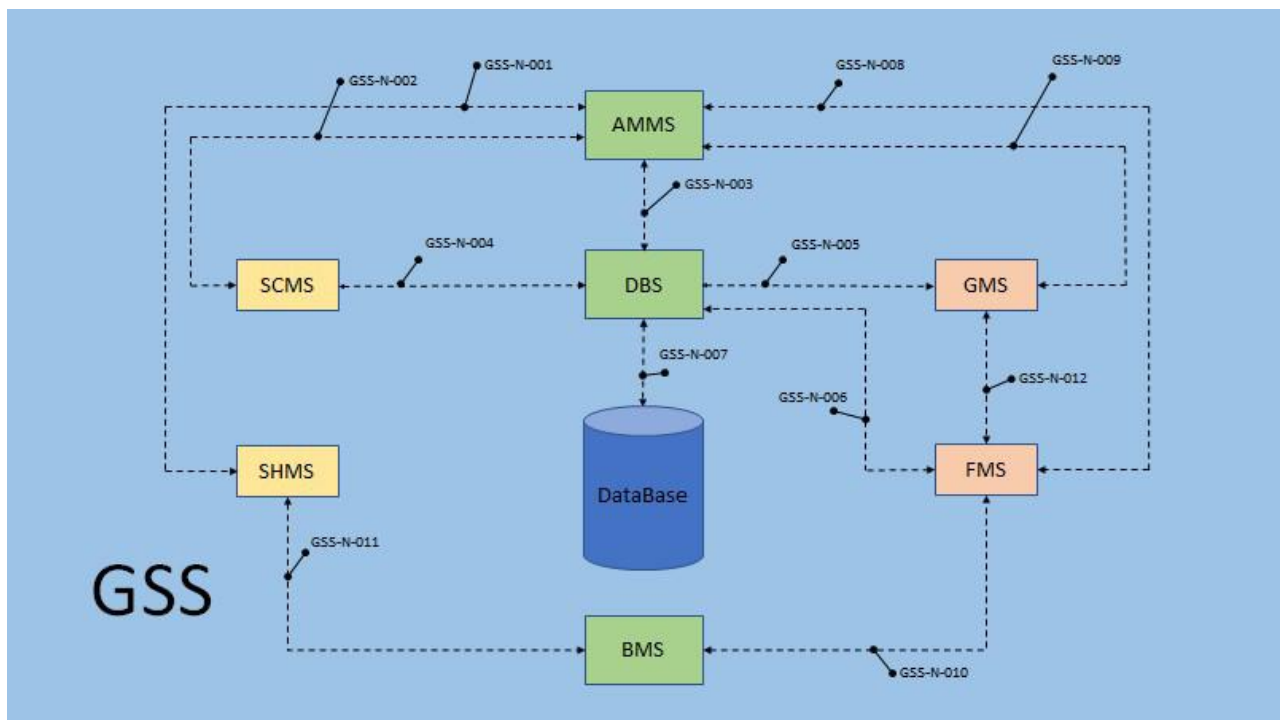
GSS-F-nnn	GSS 功能性需求(Functional Requirements)
GSS-N-nnn	GSS 非功能性需求(Non-Functional Requirements)
AMMS-F-nnn	AMMS 功能性需求(Functional Requirements)
AMMS-N-nnn	AMMS 非功能性需求(Non-Functional Requirements)
SHMS-F-nnn	SHMS 功能性需求(Functional Requirements)
SHMS-N-nnn	SHMS 非功能性需求(Non-Functional Requirements)
GMS-F-nnn	GMS 功能性需求(Functional Requirements)
GMS-N-nnn	GMS 非功能性需求(Non-Functional Requirements)
FMS-F-nnn	FMS 功能性需求(Functional Requirements)
FMS-N-nnn	FMS 非功能性需求(Non-Functional Requirements)
SCMS-F-nnn	SCMS 功能性需求(Functional Requirements)
SCMS-N-nnn	SCMS 非功能性需求(Non-Functional Requirements)
BMS-F-nnn	BMS 功能性需求(Functional Requirements)
BMS-N-nnn	BLMS 非功能性需求(Non-Functional Requirements)
DBS-F-nnn	DBS 功能性需求(Functional Requirements)
DBS-N-nnn	DBS 非功能性需求(Non-Functional Requirements)

Section 1 系統(System)

2.1 系統描述 (System Description)

本系統主要分為六個部份，分別為會員管理子系统（Account and Member Management Subsystem，**AMMS 1.1.0**）、購物車管理子系统（Shopping Cart Management Subsystem，**SHMS 1.2.0**）、遊戲管理子系统（Games Management Subsystem，**GMS 1.3.0**）、財務管理子系统（Financial Management Subsystem，**FMS 1.4.0**）、遊戲評分、評語管理子系统（Score and Comment Management Subsystem，**SCMS 1.5.0**）、結賬管理子系统（Bill Management Subsystem，**BMS 1.6.0**）、資料庫子系统（Database Subsystem，**DBS 1.7.0**）。

2.1.1 系統架構圖 (System Context Diagram)



2.2 操作概念 (Operational Concepts)

Scenario 1：網站首頁導覽概念(Anonymous Operational Concepts)

任何使用者可以觀看各種遊戲商品，並能將商品放入**購物車(SHMS)**的功能。使用者可以使用**結賬管理子系統 (BMS)**來購買遊戲。

Scenario 2：顧客使用者操作概念(Customer Operational Concepts)

使用者經由**會員管理系統(AMMS)**登入為**顧客(Customer)**身分，可以使用**遊戲評分、評語管理系統(SCMS)**進行評價遊戲，使用**財務管理系統(FMS)**查看自己的購買記錄。

Scenario 3：管理者操作概念(Manager Operational Concepts)

使用者經由**會員管理系統(AMMS)**登入為**一般管理者(Manager)**身分，可以使用**遊戲管理系統(GMS)**上下架遊戲與管理遊戲評語。

2.3 設計限制 (Design, Data, and Implementation Constrains)

需求編號	優先順序	需求描述
GSS-N-024	1	本系統使用的 DBMS 為 MySQL
GSS-N-025	1	本系統網頁設計使用 React + JavaScript

2.4 技術限制 (Technological Limitations)

需求編號	優先順序	需求描述
GSS-N-026	1	金流系統（使用線上刷卡、PayPal 或 ATM 轉賬）
GSS-N-027	1	會員認證（手機簡訊認證或實名身份證認證）
GSS-N-028	1	庫存管理（實體遊戲的渠道）

2.5 介面需求 (Interface Requirements)

2.5.1 使用者介面需求 (User Interfaces Requirements)

需求編號	優先順序	需求描述
-	1	

2.5.2 外部介面需求 (External Interface Requirements)

需求編號	優先順序	需求描述
GSS-N-029	1	使用者使用瀏覽器透過 HTTP 通訊協定與 GSS Web 主機通訊
GSS-N-030	1	Web 主機透過 Apache 鏈接 Tomcat，進入 GSS 系統

2.5.3 內部介面需求 (Internal Interface Requirements)

需求編號	優先順序	需求描述
GSS-N-001	1	AMMS 和 SHMS 間必須能傳送與接收使用者所點選之商品資訊。
GSS-N-002	1	AMMS 和 SCMS 間必須能傳送與接收使用者對商品的評價資訊。
GSS-N-003	1	AMMS 和 DBS 間必須能傳送與接收各種使用者登入資訊。
GSS-N-004	1	SCMS 和 DBS 間必須能傳送與接收遊戲商品評價的相關資訊。
GSS-N-005	1	GMS 和 DBS 間必須能傳送與接收遊戲商品的相關資訊。
GSS-N-006	1	FMS 和 DBS 間必須能傳送與接收財務收支的資訊。
GSS-N-007	1	DBS 和資料庫之間必須能正確的傳送、接收資料。
GSS-N-008	1	AMMS 和 FMS 間必須能傳送與接收使用者金錢儲匯和紅利點數的資訊。
GSS-N-009	1	AMMS 和 GMS 間必須能傳送與接收使用者欲新增、修改、刪除之遊戲資訊。
GSS-N-010	1	FMS 和 BMS 間必須能傳送與接收商品結帳和所得利潤的資訊。
GSS-N-011	1	SHMS 和 BMS 間必須能傳送與接收購買商品的資訊。
GSS-N-012	1	GMS 和 FMS 間必須能傳送與接收遊戲進貨的金錢和遊戲資訊。

2.6 功能性需求 (Functional Requirements)

需求編號	優先順序	需求描述
-	1	

2.7 非功能性需求 (Non-Functional Requirements)

2.7.1 效能需求 (Performance Requirements)

需求編號	優先順序	需求描述
GSS-N-031	1	資料表的設計應正規化、節省空間
GSS-N-032	1	使用者瀏覽時，頁面讀取應不大於 5 秒
GSS-N-033	1	使用者搜尋時，搜尋時間應不大於 3 秒
GSS-N-034	1	系統維護時，應有故障頁面通知使用者

2.7.2 測試需求 (Test Requirements)

需求編號	優先順序	需求描述
GSS-N-035	1	每個子系統功能都應被測試過
GSS-N-036	1	提供使用者 GUI 介面的測試
GSS-N-037	1	提供各式平台使用系統的測試

2.8 其他需求 (Other Requirements)

2.8.1 環境需求 (Environmental Requirement)

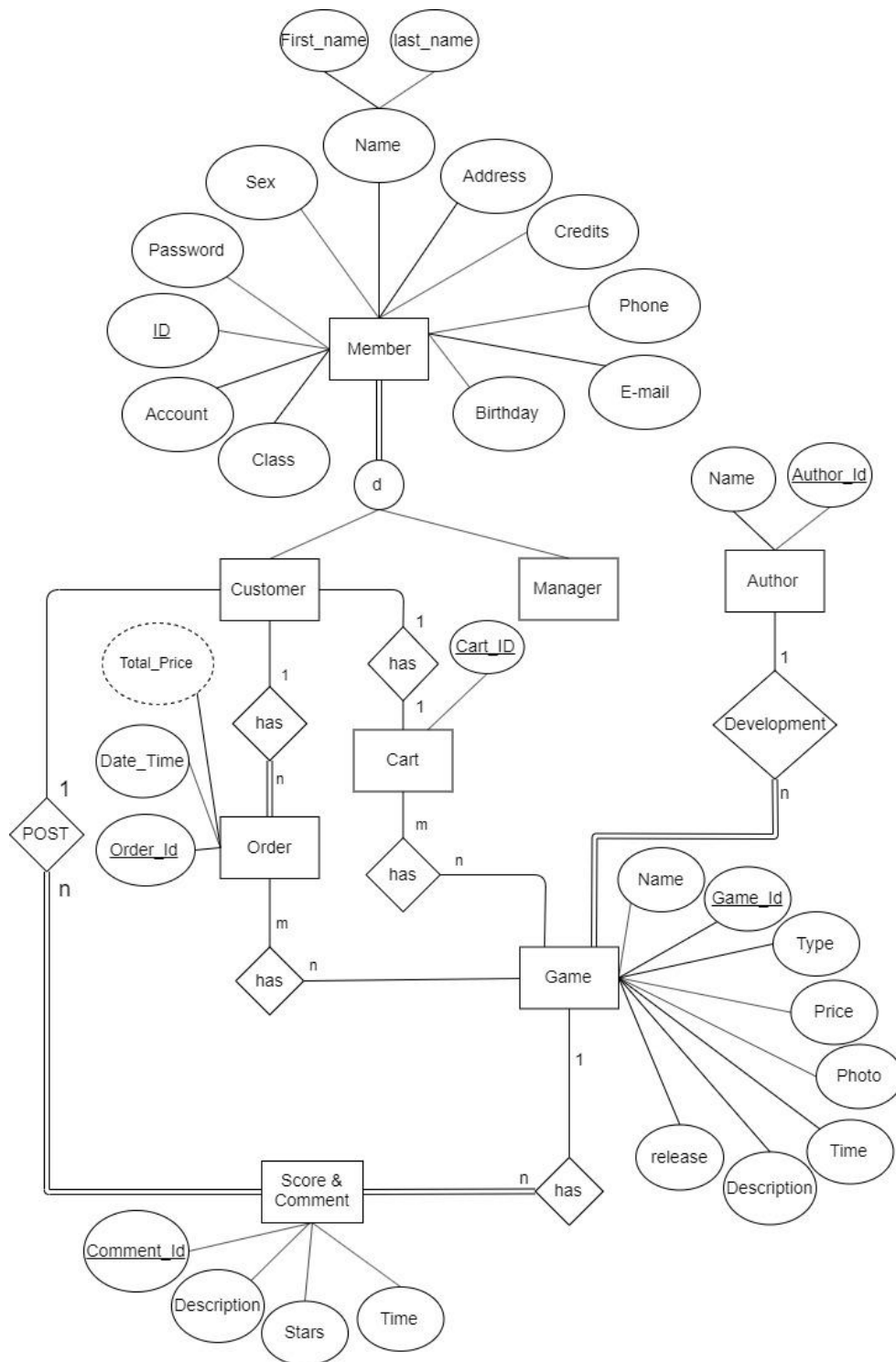
需求編號	優先順序	需求描述
-	1	

2.8.2 安裝需求 (Installation Requirement)

需求編號	優先順序	需求描述
-	1	

Section 2 資料庫概念設計(Conceptual Design of the Database)

3.1 Entity-Relationship (ER) Model

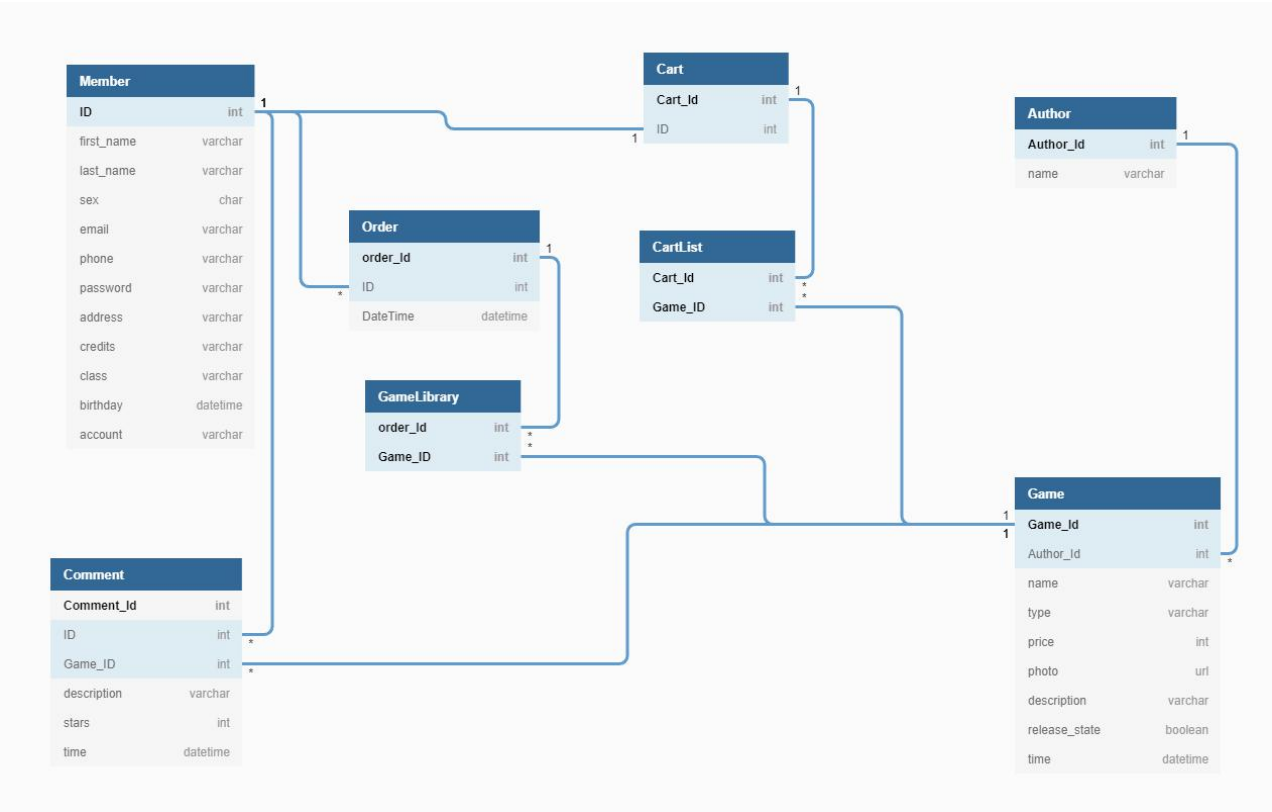


3.2 Data Dictionary and Business Rules

- 1.本商店商品售價皆已包含營業稅價格，結賬時不需再加收營業稅金錢。
- 2.本商店只提供線上刷卡付費，不接受任何現金交易。
- 3.本商店商品不提供任何退貨方式，所以買了就不能退款。
- 4.本商店的會員注冊是免費的，但是遊戲商品需要付費。

Section 3 邏輯資料庫網要(Logical Database Schema)

4.1 Schema of the Database



MEMBER				
Description: 存放各種權限會員個人資料及賬號密碼				
Attribute	Domain Type	Key	Nullable	Description
ID	int	Primary	Not null	會員編碼(ID >=0)
First_name	varchar		Not null	會員 First Name
Last_name	varchar		Not null	會員 Last Name
Sex	varchar			會員性別
Email	varchar		Not null	會員電子郵件
Phone	varchar			會員電話號碼
Password	varchar		Not null	會員密碼
Address	varchar		Not null	會員地址
Credit	varchar		Not null	會員行用卡號碼
Class	int		Not null	會員權限階級(0 <= Class <= 2) 0 為訪客 1 為普通會員 2 為管理員
Birthday	datetime			會員生日
Account	varchar	Unique	Not null	會員賬號

GAME_LIBRARY				
Description: 多對多生成的 game 表格				
Attribute	Domain Type	Key	Nullable	Description
Order_id	int	Primary	Not null	賬單編號(ID >=0)
Game_id	int	Foreign	Not null	遊戲編號 Reference (MEMBER)

GAME				
Description: 存放遊戲基本資訊				
Attribute	Domain Type	Key	Nullable	Description
Game_id	int	Primary	Not null	遊戲編碼(ID >=0)
Author_id	int	Foreign	Not null	作者編號 Reference(AUTHOR)
Name	varchar		Not null	遊戲名稱
Type	varchar		Not null	遊戲類型
Price	int		Not null	遊戲價格(price >= 0)
Photo	varbinary(max)			遊戲封面
Time	datetime		Not null	遊戲上架時間
Description	varchar			遊戲內容介紹
Release_state	bool		Not null	遊戲上下架狀況

CART				
Description: 會員購物車				
Attribute	Domain Type	Key	Nullable	Description
Cart_ID	int	Primary	Not null	購物車編號(ID >=0)
ID	int	Foreign	Not null	會員編號 Reference (MEMBER)

CART_LIST				
Description: 多對多生成的 cart 表格				
Attribute	Domain Type	Key	Nullable	Description
Cart_ID	int	Primary	Not null	購物車編號(ID >=0)
Game_ID	int	Foreign	Not null	遊戲編號

COMMENT				
Description: 遊戲的評分				
Attribute	Domain Type	Key	Nullable	Description
Comment_id	int	Primary	Not null	評論的編號(ID >=0)
ID	int	Foreign	Not null	會員編號 Reference(MEMBER)
Game_ID	int	Foreign	Not null	遊戲編號 Reference(GAME)
Description	varchar		Not null	遊戲的評論內容
Stars	int		Not null	評分的星級(0 <= Star <= 10)
Time	datetime		Not null	遊戲評論時間

ORDER				
Description: 購物車的賬單				
Attribute	Domain Type	Key	Nullable	Description
Order_id	int	Primary	Not null	賬單的編號(ID >=0)
ID	int	Foreign	Not null	會員編號 Reference(CART)
DateTime	datetime		Not null	賬單時間

AUTHOR				
Description: 遊戲開發者資訊				
Attribute	Domain Type	Key	Nullable	Description
Author_id	int	Primary	Not null	開發者的編號(ID >=0)
Name	varchar		Not null	開發者的名字

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

這裡先針對各表格做各別的預估，會跨表格影響的操作接在各別表格之後，操作基本上不離新增查詢修改刪除四項。

Table	可能操作	預估使用頻率 (per day)	表格資料量 (tuples)	系統負擔 (worst case)
MEMBER	user 登入時驗證身份	100	200	2000 次 Query/day
MEMBER	下訂單后查詢個資	50	200	1000 次 Query/day
MEMBER	新增會員資料	10	Don't care	10 次 Insert/day
MEMBER	更新會員資料	50	200	10000 次 Query/day 50 次 Update/day
GAME	新增遊戲資料	5	Don't care	5 次 Insert/day
GAME	更新遊戲資料	2	Don't care	2 次 Update/day
GAME	查詢遊戲資料	100	5000	500000 次 Query/day
AUTHOR	新增作者資料	7	Don't care	7 次 Insert/day
AUTHOR	查詢作者資料	50	3000	150000 次 Query/day
GAME_LIBRARY	查詢遊戲資料	100	5000	500000 次 Query/day
CART	新增購物車資訊	100	Don't care	100 次 Insert/day
CART	更新購物車資訊	100	2000	200000 次 Query/day 100 次 Update/day
CART	刪除購物車資訊	10	5000	50000 次 Query/day 10 次 Delete/day
CART_LIST	查詢購物車內遊戲資料	100	5000	500000 次 Query/day
COMMENT	更新評論資訊	10	5000	50000 次 Query/day 10 次 Update/day
COMMENT	新增評論資訊	50	Don't care	50 次 Insert/day

ORDER	新增訂單資料	50	Don't care	50 次 Insert/day
ORDER	查詢訂單資料	50	5000	250000 次 Query/day
ORDER	查看訂單內遊戲資訊	100	15000	1500000 次 Query/day

4.3 SQL Statements Used to Construct the Schema

```
DROP DATABASE IF EXISTS Project;
CREATE DATABASE IF NOT EXISTS Project CHARACTER SET utf8;
USE Project;
```

```
CREATE TABLE IF NOT EXISTS `Member` (
  ID INT NOT NULL,
  First_Name VARCHAR(20),
  Last_Name VARCHAR(20),
  Sex CHAR NOT NULL,
  Email VARCHAR(50) NOT NULL,
  Phone VARCHAR(10),
  `Password` VARCHAR(30) NOT NULL,
  Credits VARCHAR(20),
  Class VARCHAR(20),
  Birthday DATE,
  Account VARCHAR(20) NOT NULL,
  PRIMARY KEY (ID)
)ENGINE=InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `Author` (
  Author_ID INT NOT NULL,
  name VARCHAR(50),
  PRIMARY KEY(Author_ID)
)ENGINE=InnoDB;
```



```

CREATE TABLE IF NOT EXISTS `Game` (
  Game_ID INT NOT NULL,
  Author_ID INT ,
  name VARCHAR(50) NOT NULL,
  type VARCHAR(50) NOT NULL,
  price INT NOT NULL,
  photo LONGBLOB ,
  description TEXT NOT NULL,
  release_state BOOLEAN NOT NULL,
  `time` DATE NOT NULL,
  PRIMARY KEY(Game_ID),
  FOREIGN KEY(Author_ID) REFERENCES `Author`(Author_ID) ON DELETE CASCADE
)ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Cart` (
  Cart_ID INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (Cart_ID),
  FOREIGN KEY(ID) REFERENCES `Member`(ID) ON DELETE CASCADE
)ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Cart_List` (
  Cart_ID INT NOT NULL,
  Game_ID INT NOT NULL,
  PRIMARY KEY (Cart_ID, Game_ID),
  FOREIGN KEY(Cart_ID) REFERENCES Cart(Cart_ID) ON DELETE CASCADE,
  FOREIGN KEY(Game_ID) REFERENCES Game(Game_ID) ON DELETE CASCADE
)ENGINE=InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `Order` (
  Order_ID INT NOT NULL,
  ID INT,
  DateTime DateTime,
  PRIMARY KEY (Order_ID, ID),
  FOREIGN KEY(ID) REFERENCES `Member`(ID) ON DELETE CASCADE
)ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `GameLibrary` (
  Order_ID INT NOT NULL,
  Game_ID INT NOT NULL,
  PRIMARY KEY (Order_ID, Game_ID),
  FOREIGN KEY(Order_ID) REFERENCES `Order`(Order_ID) ON DELETE CASCADE,
  FOREIGN KEY(Game_ID) REFERENCES Game(Game_ID) ON DELETE CASCADE
)ENGINE=InnoDB;

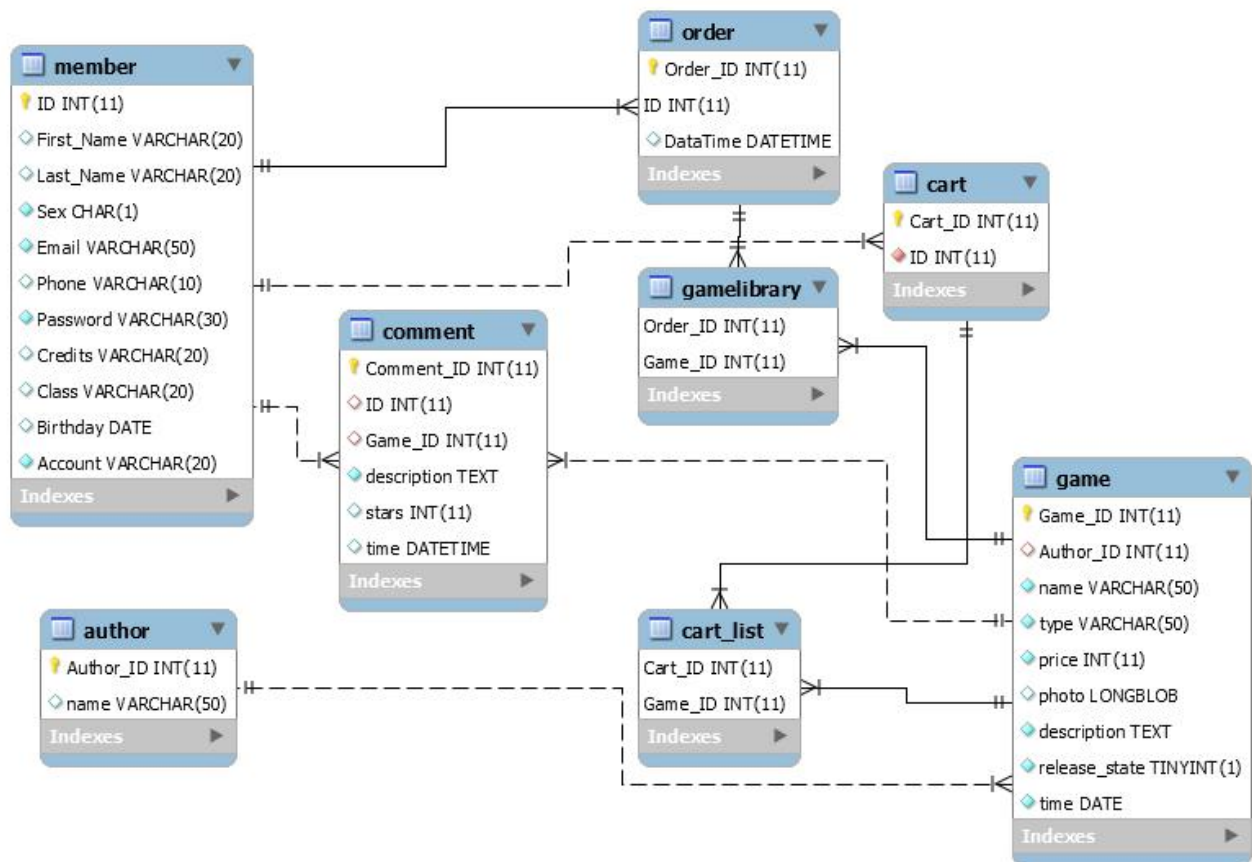
CREATE TABLE IF NOT EXISTS `Comment` (
  Comment_ID INT NOT NULL,
  ID INT,
  Game_ID INT,
  description TEXT NOT NULL,
  stars INT,
  `time` DateTime,
  PRIMARY KEY(Comment_ID),
  FOREIGN KEY(ID) REFERENCES `Member`(ID) ON DELETE CASCADE,
  FOREIGN KEY(Game_ID) REFERENCES `Game`(Game_ID) ON DELETE CASCADE
)ENGINE=InnoDB;

INSERT INTO `Member` VALUES (123, 'ASD', 'FF', 0, 'E3FIOWEF@JIOA', '0956481235', 'fewgdq', 'creditddd', '123', '1999-01-12', 'wd');
INSERT INTO `Author` VALUES (456, '張育瑞');
INSERT INTO `Game` VALUES (123, 456, '越南大戰', '射擊', 100, NULL, '很好玩 吧', TRUE, '2087-05-11');
INSERT INTO `Order` VALUES (123, 123, '1999-01-12');
INSERT INTO `GameLibrary` VALUES (123, 123);
INSERT INTO `Cart` VALUES (123, 123);
INSERT INTO `Cart_List` VALUES (123, 123);

SELECT * FROM `Member`;
SELECT * FROM `GameLibrary`;
SELECT * FROM `Game`;
SELECT * FROM `Author`;
SELECT * FROM `Order`;
SELECT * FROM `Cart`;
SELECT * FROM `Cart_List`;

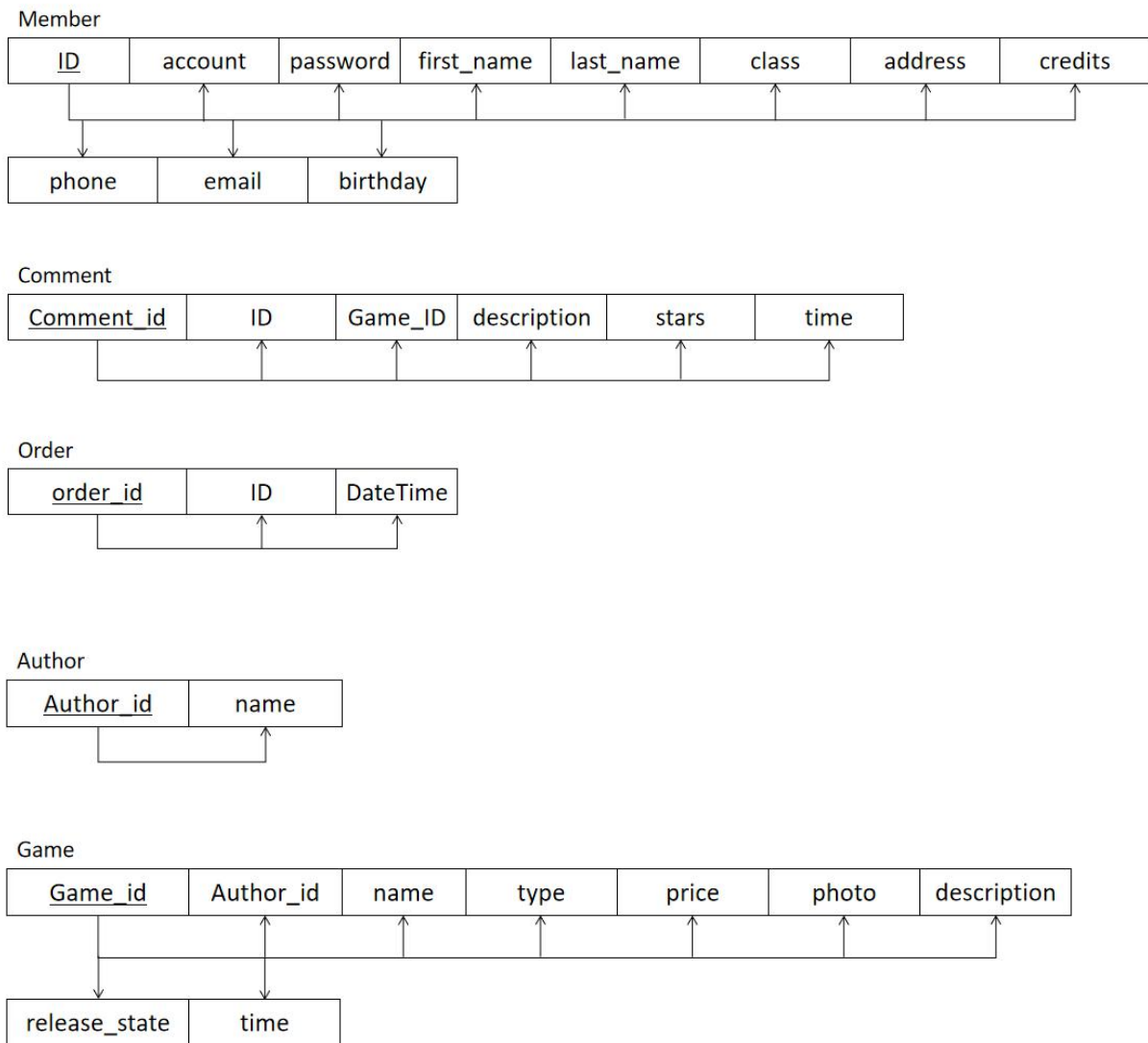
```

4.4 The implementation of tables in target DBMS



Section 4 Functional Dependencies and Database Normalization

5.1 Functional Dependencies



Section 5 The use of the Database System

6.1 System Installation Description

6.1.1 MySQL



Figure 1 安裝程式畫面

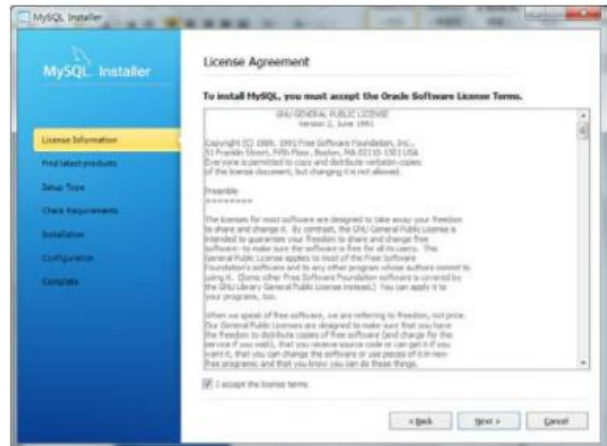


Figure 2 授權文件

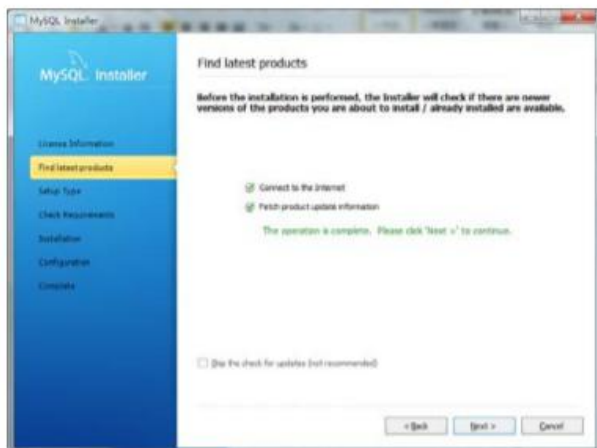


Figure 3 檢查最新版本

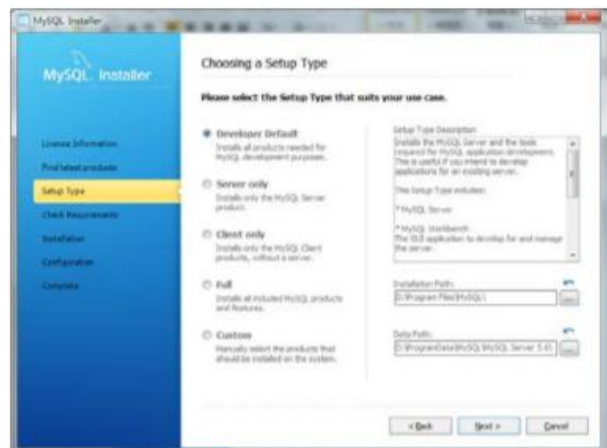


Figure 4 安裝類型

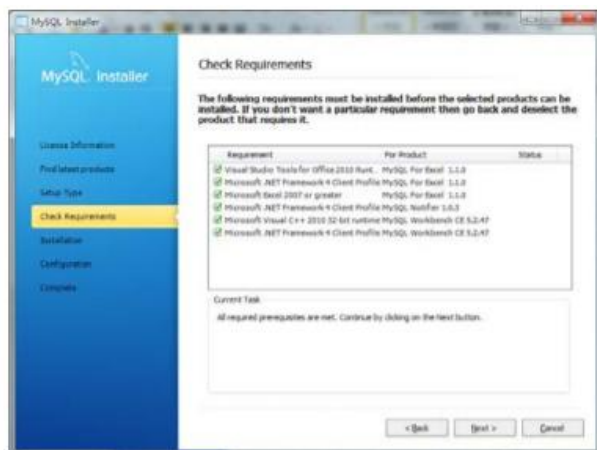


Figure 5 檢查安裝前必要軟體

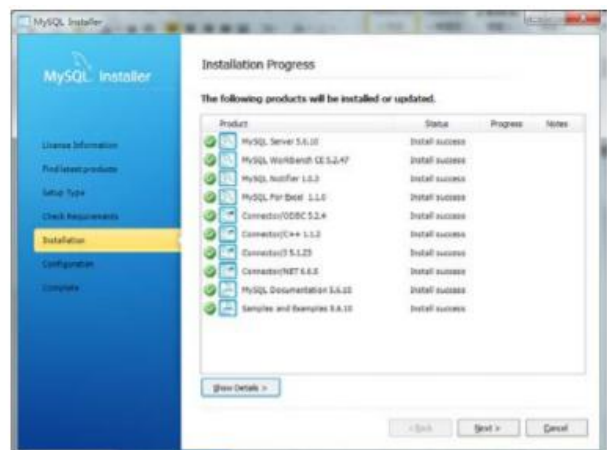


Figure 6 安裝所有軟體

6.1.2 Heroku 設定

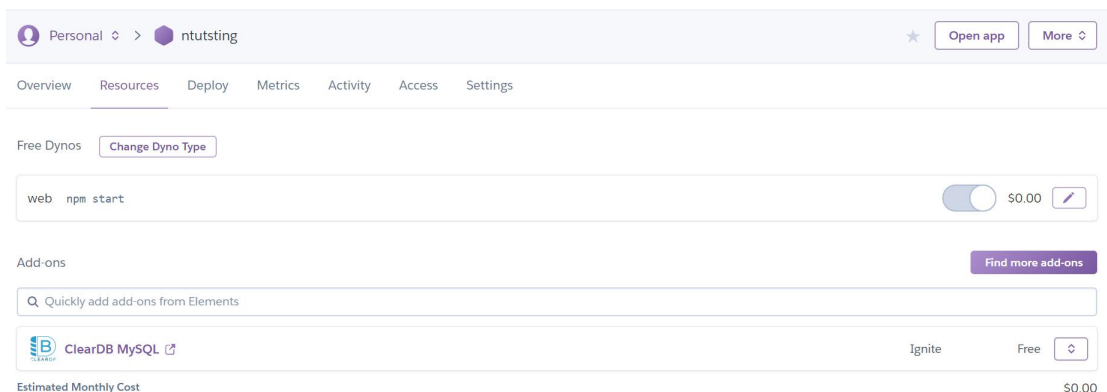
1. 申請 heroku 帳號
2. ClearDB 這項服務是免費的，但是必須要先填寫信用卡資料，才能完成啟用
3. heroku addons:create cleardb:ignite (啟用 ClearDB)

```
Creating cleardb:ignite on ● ntutsting... free
Created cleardb-infinite- as CLEARDB_BRONZE_URL
Use heroku addons:docs cleardb to view documentation
```

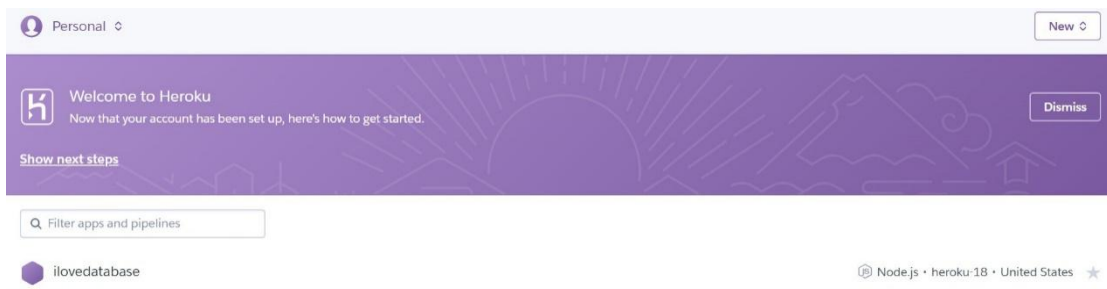
4. heroku config | grep CLEARDB_DATABASE_URL
(取得 DB 資訊 mysql://User:Pasword@host/databasename)

```
== ntutsting Config Vars
CLEARDB_DATABASE_URL: mysql://b1l f274:35 16@us-cdbr-iron-east-05.cleardb.net/heroku_4099 892?reconnect=true
```

5. 將本地先前建好的資料庫遷移至 heroku CleanDB
6. 匯出本機 database
mysqldump -u username -p databasename > backup.sql
7. 匯入 CleanDB
mysql -u username -p databasename < backup.sql
8. 完成



9. 一樣使用 heroku 部屬前端和後端 api(以下以前端為例)
10. 新增 app



11. 將網站部署到 heroku

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Clone the repository

Use Git to clone ilovedatabase's source code to your local machine.

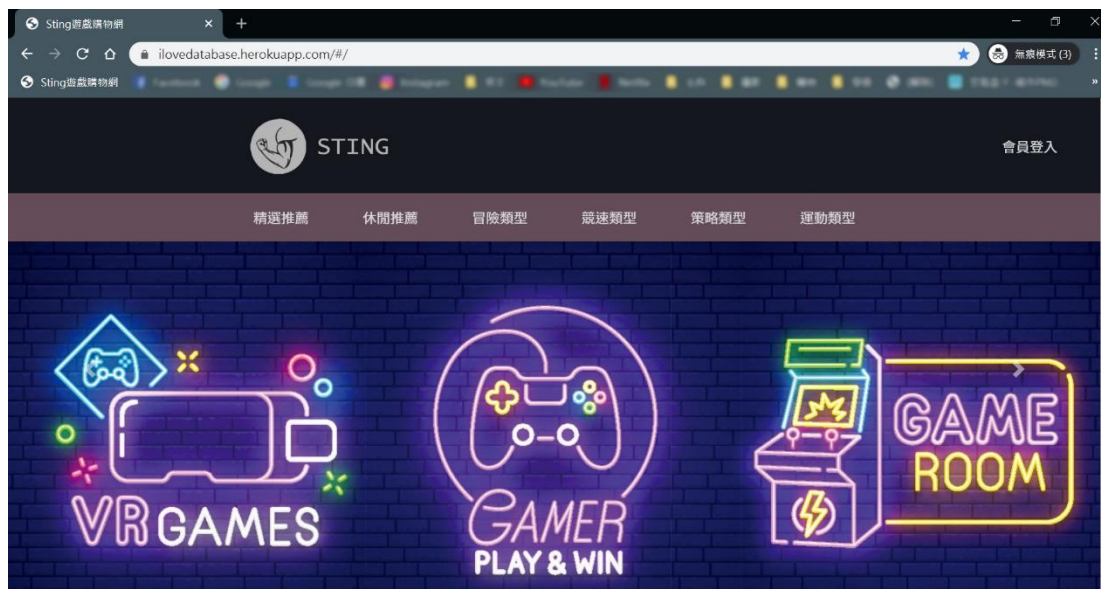
```
$ heroku git:clone -a ilovedatabase  
$ cd ilovedatabase
```

Deploy your changes

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

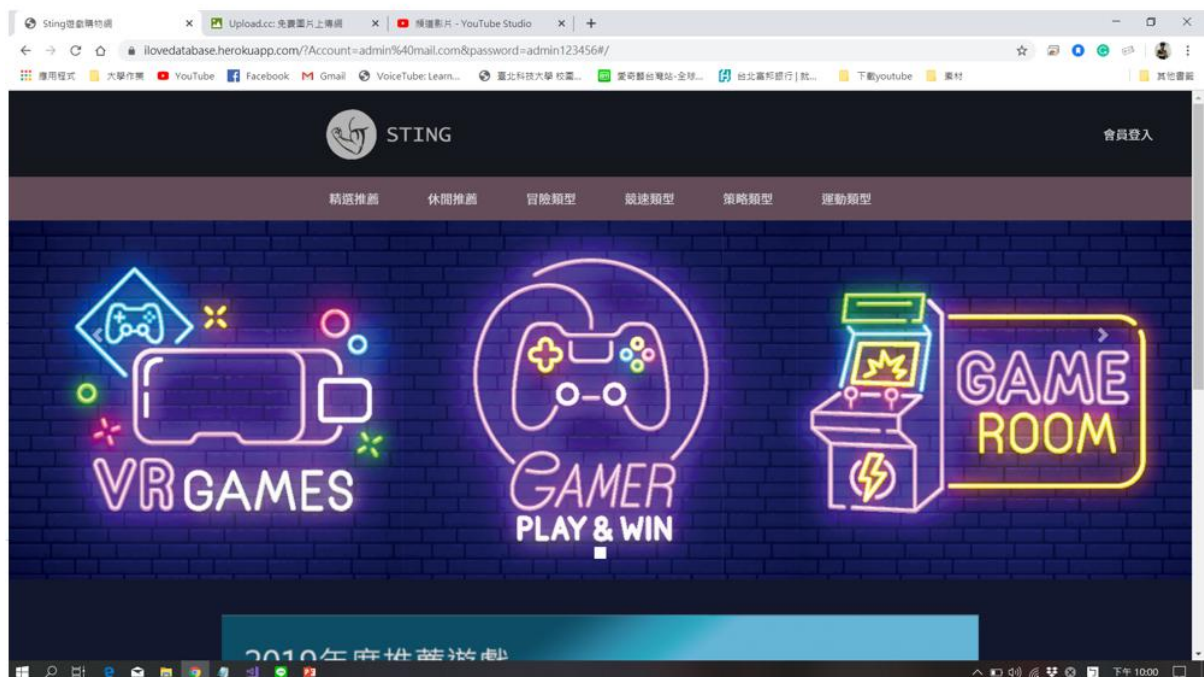
12. 完成



6.2 The Use of the System

6.2.1 使用者頁面

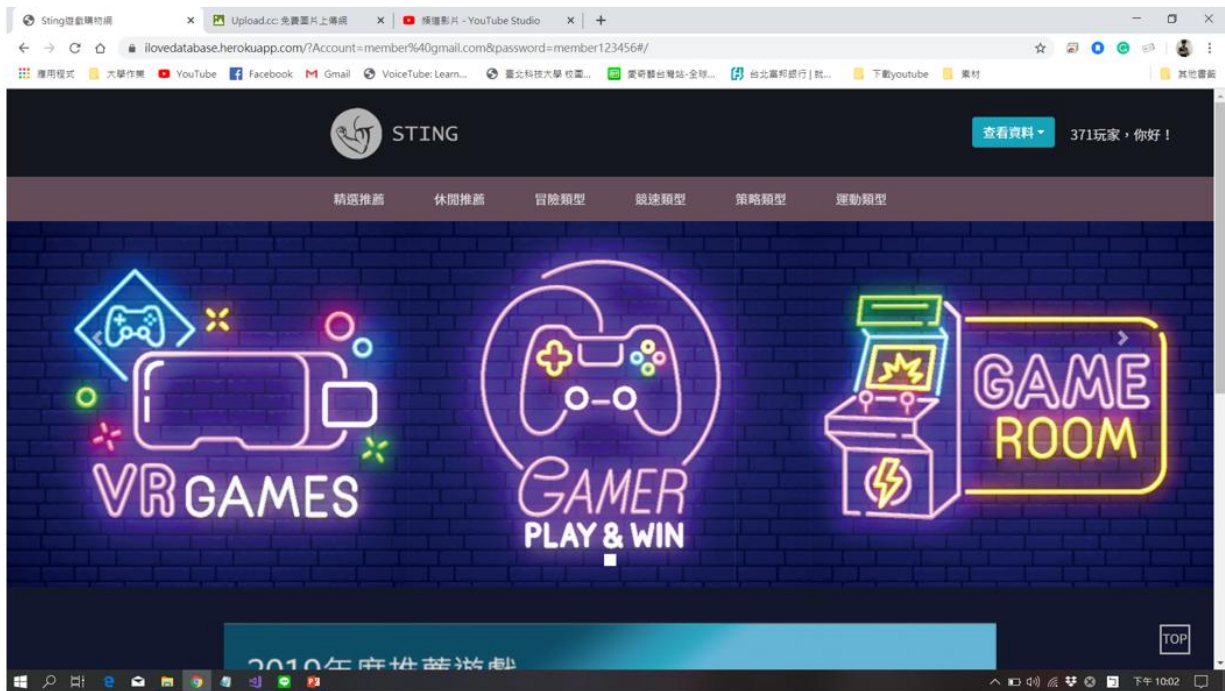
首頁



帳號註冊

A screenshot of the '帳號註冊' (Account Registration) page on the STING website. The page has a dark background with white text and input fields. The registration form includes the following fields: '輸入帳號' (Account) with a placeholder 'Account', '輸入密碼' (Password) with a placeholder 'password', and a '性別' (Gender) dropdown menu with a placeholder '--性別--'. Below these, there are fields for '姓' (Last Name) with a placeholder 'First Name', '名' (First Name) with a placeholder 'last Name', and 'Phone' with a placeholder '09XXXXXXXX'. There are also fields for 'Email' with a placeholder 'Ex:123456789@gmail.com' and '地址' (Address) with a placeholder 'address'. A '生日' (Birthday) section includes dropdowns for year, month, and day. A '信用卡' (Credit Card) section includes a dropdown for card type, and fields for '卡號' (Card Number), 'CVC', 'YY/MM' (Expiration Date), and 'UserName'. A blue '送出' (Submit) button is located at the bottom of the form.

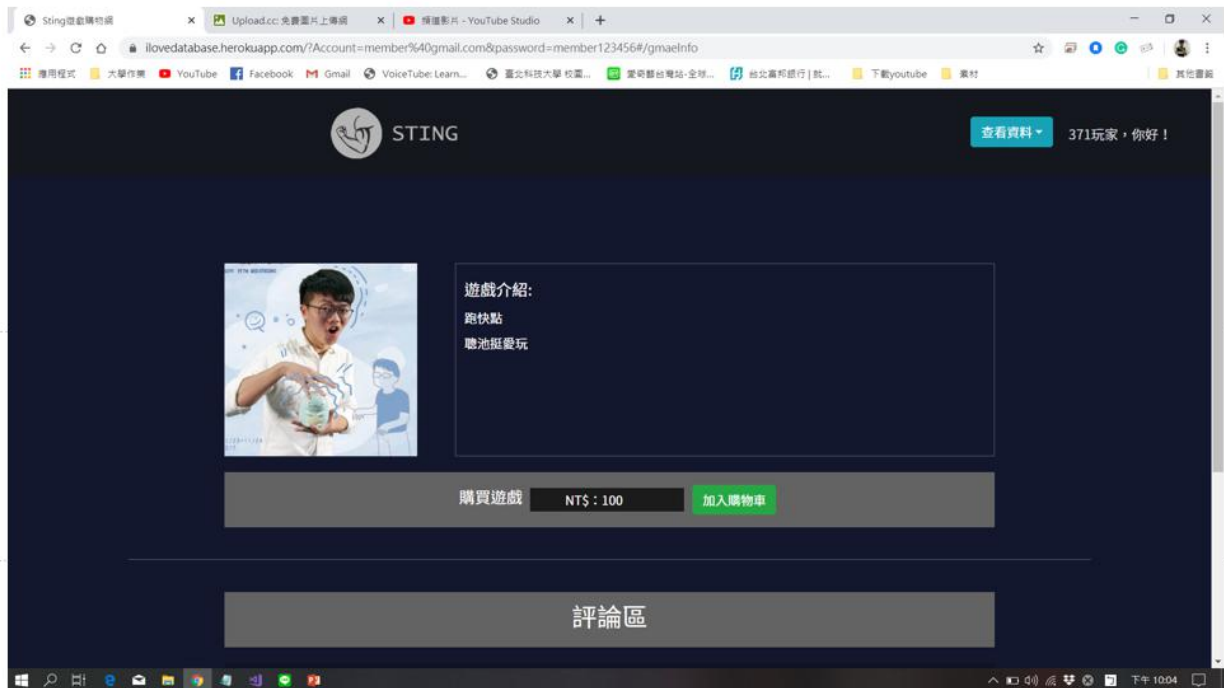
玩家登錄



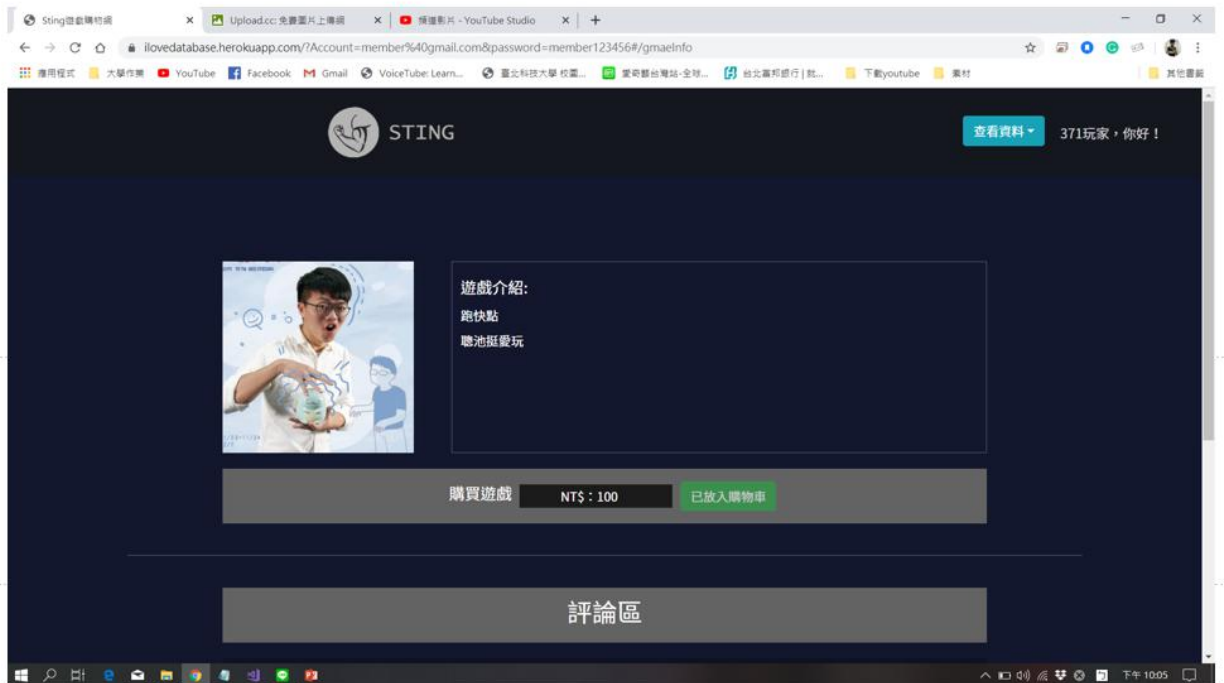
點選休閒推薦分類



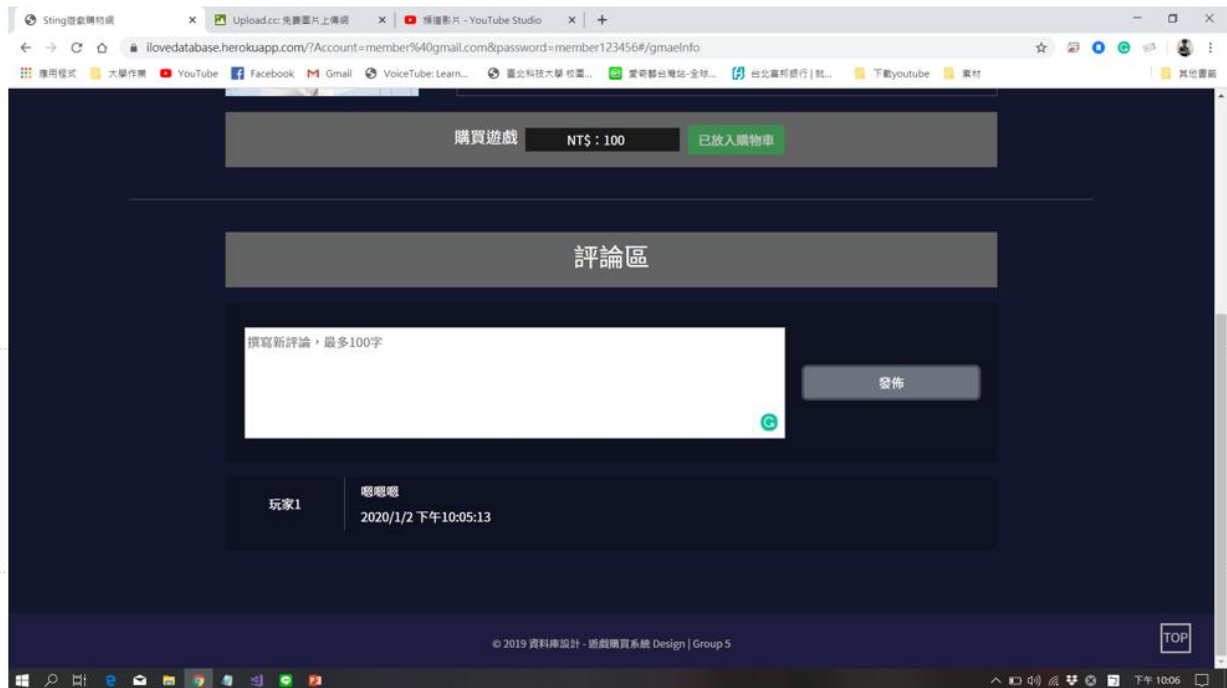
點選進入商品頁面



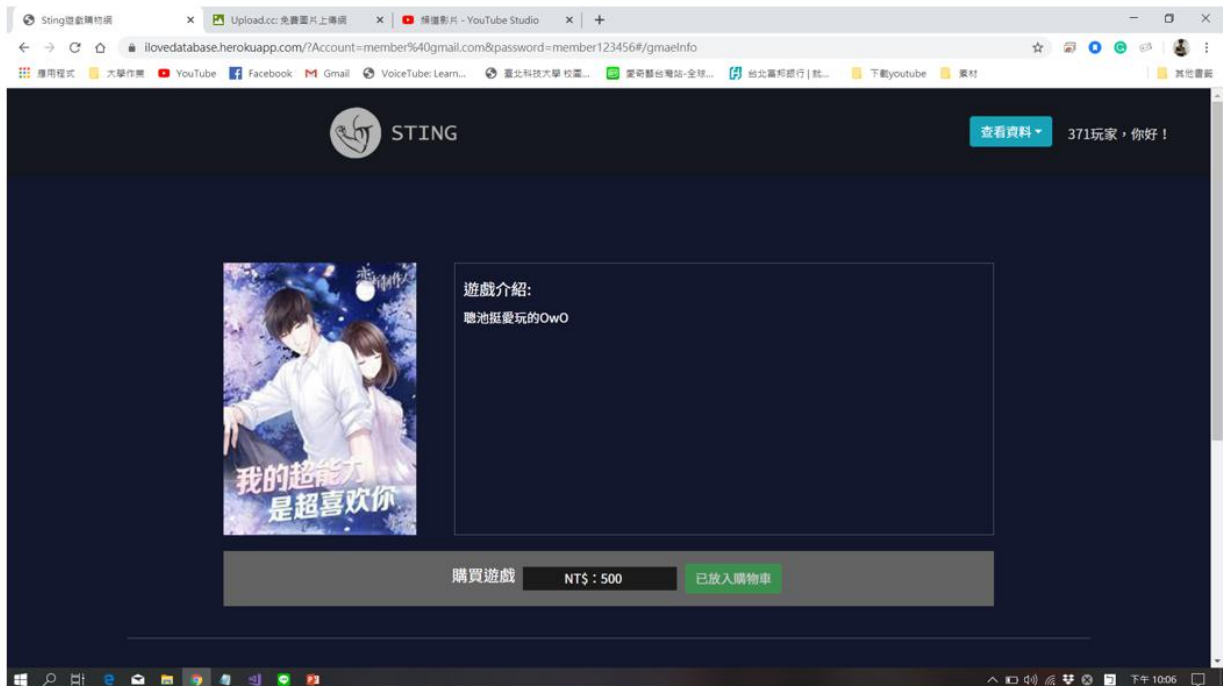
加入購物車



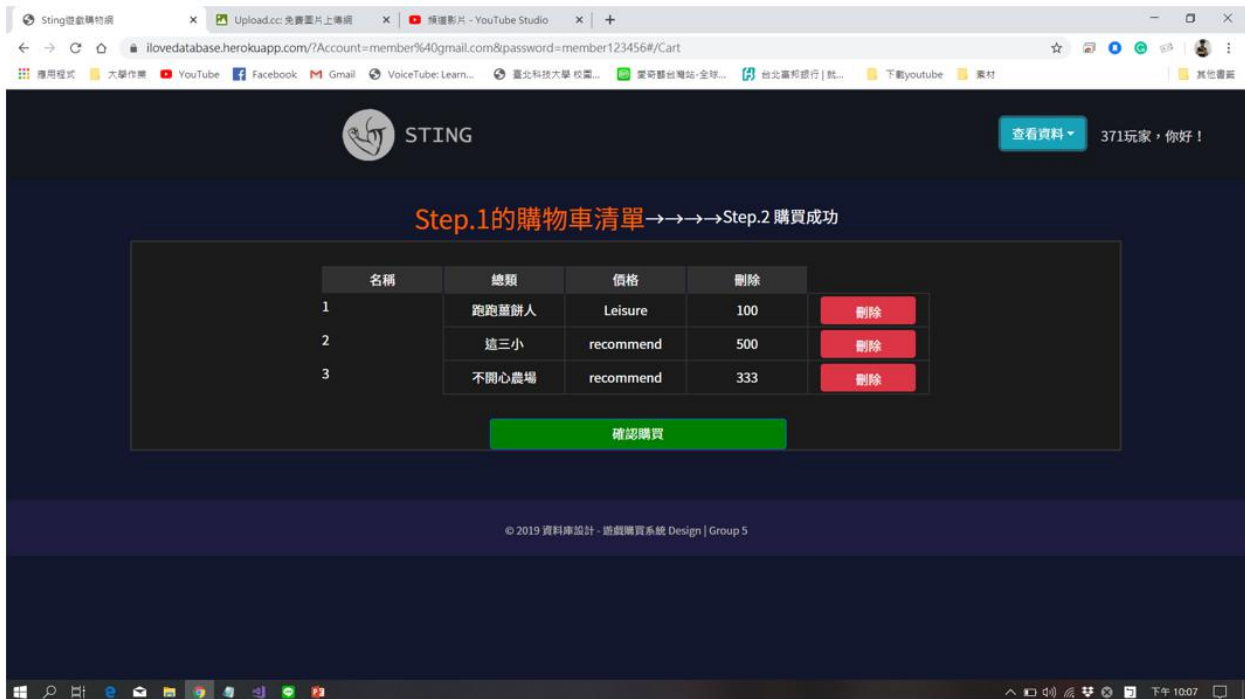
對遊戲發表評價



購物車中在加入一個遊戲



到購物車的頁面確認訂單



STING 查看資料 371玩家，你好！

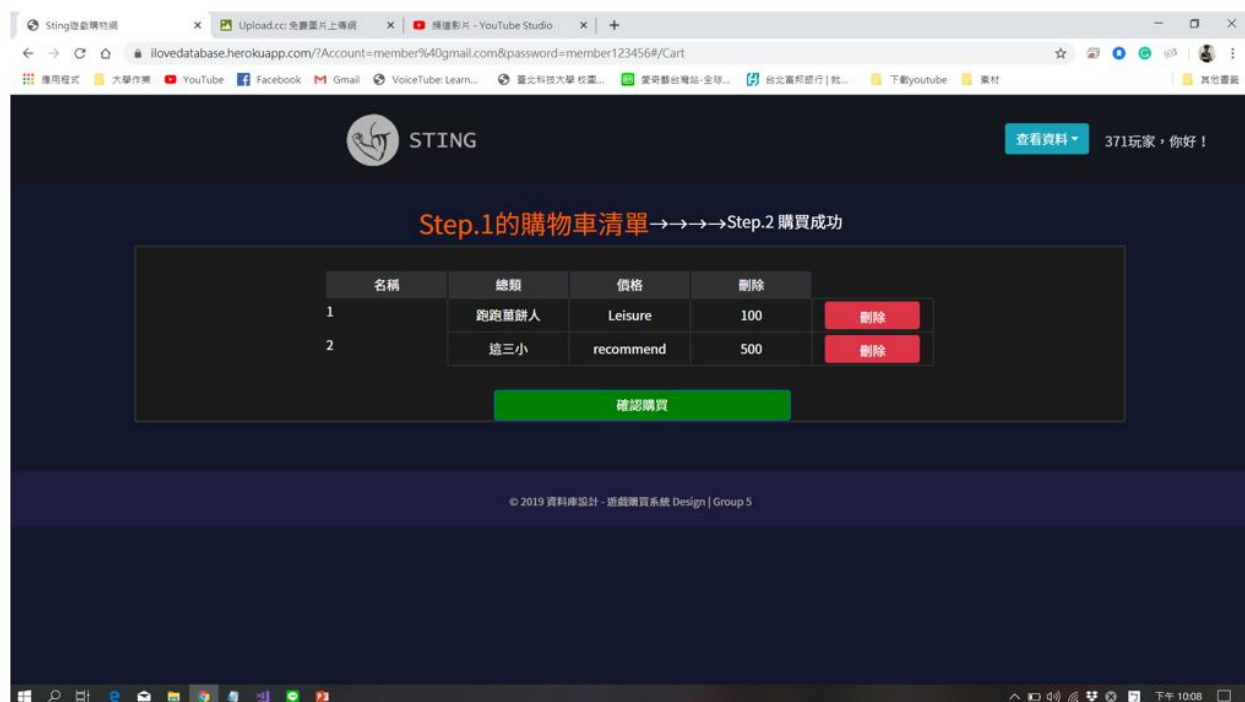
Step.1的購物車清單→→→Step.2 購買成功

名稱	總類	價格	刪除	
1	跑跑薑餅人	Leisure	100	刪除
2	這三小	recommend	500	刪除
3	不開心農場	recommend	333	刪除

確認購買

© 2019 資料庫設計 - 遊戲購買系統 Design | Group 5

刪除一筆



STING 查看資料 371玩家，你好！

Step.1的購物車清單→→→Step.2 購買成功

名稱	總類	價格	刪除	
1	跑跑薑餅人	Leisure	100	刪除
2	這三小	recommend	500	刪除

確認購買

© 2019 資料庫設計 - 遊戲購買系統 Design | Group 5

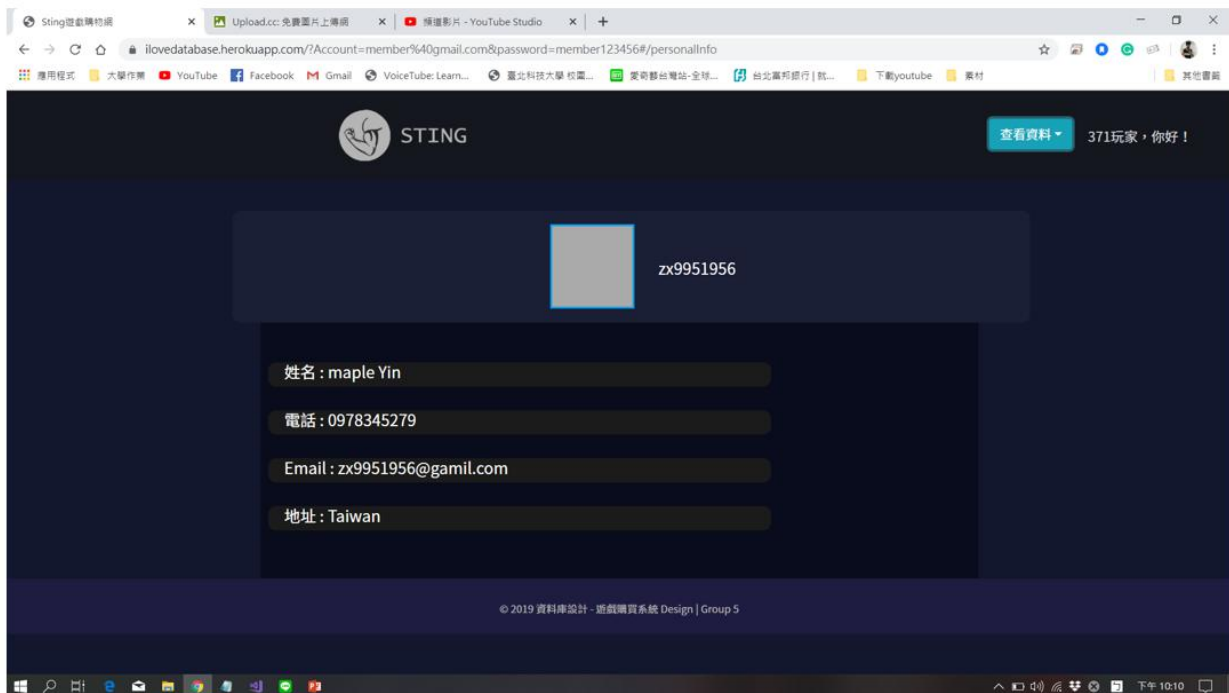
按下確認購買



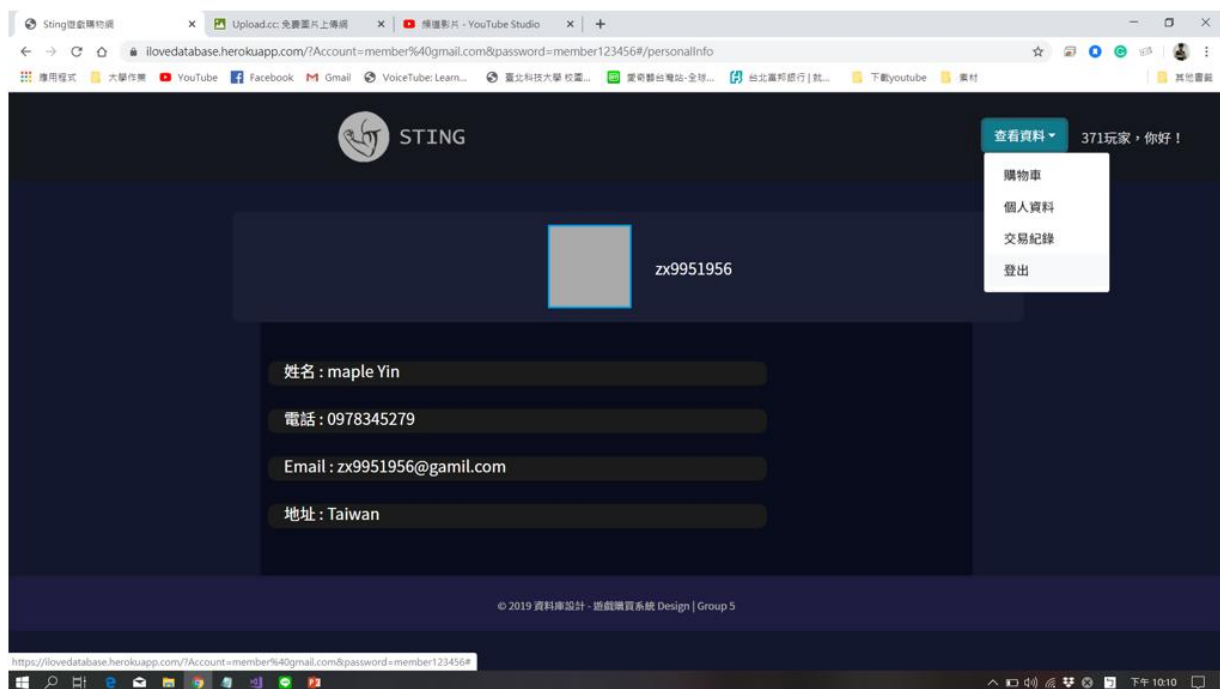
查看交易紀錄



查看個人資料

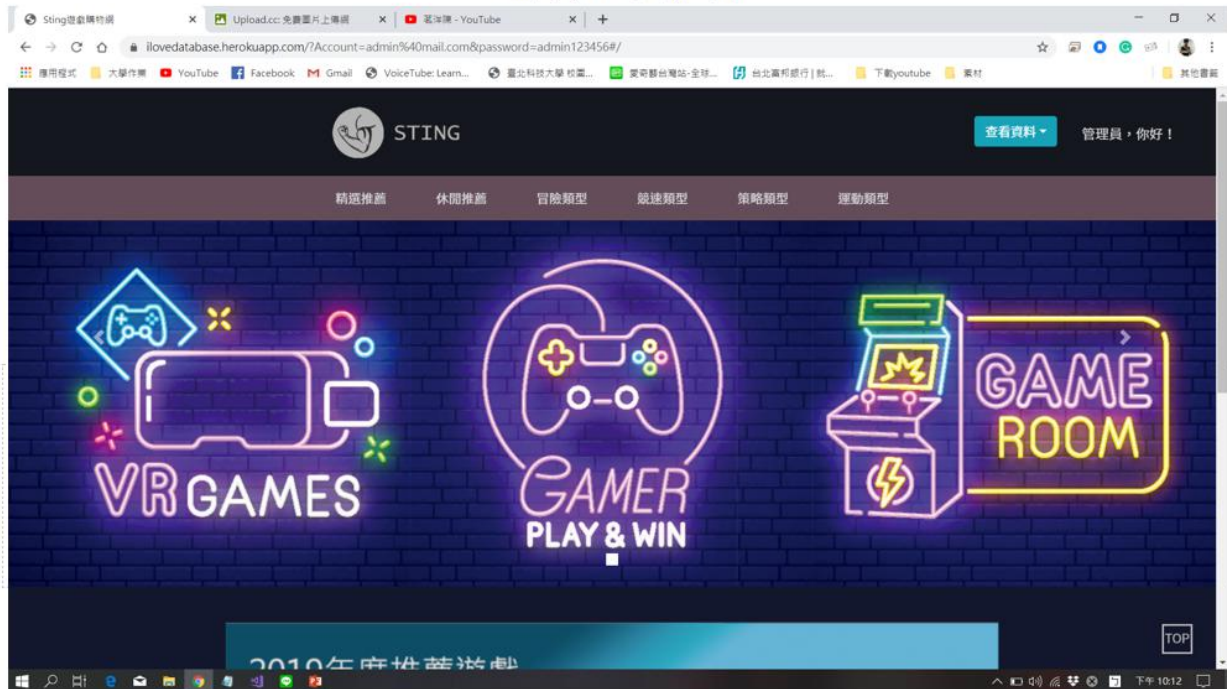


一般使用者登出



6.2.2 管理者頁面

登錄管理員帳號



點選遊戲管理



點及下架



STING 查看資料 管理員，你好！

遊戲管理：管理架上遊戲

名稱	種類	價格	廠商	管理
跑跑薑餅人	Leisure	100	張育瑞	下架
fight	adventure	200	劉政威	下架
chicken horse	Strategy	400	劉政威	下架
Grand Thert Auto 5	Leisure	66	陳峻峰	下架
NBA 2K20	sports	66	葉凱毅	下架
overcooked	Strategy	66	劉政威	下架
overcooked 2	Strategy	166	張育瑞	下架
ASSASSIN'S	adventure	399	陳峻峰	下架
JUST DANCE	sports	399	葉凱毅	下架
急速領域	Speed	1000	陳峻峰	下架

遊戲被下架了



STING 查看資料 管理員，你好！

遊戲管理：管理架上遊戲

名稱	種類	價格	廠商	管理
fight	adventure	200	劉政威	下架
chicken horse	Strategy	400	劉政威	下架
Grand Thert Auto 5	Leisure	66	陳峻峰	下架
NBA 2K20	sports	66	葉凱毅	下架
overcooked	Strategy	66	劉政威	下架
overcooked 2	Strategy	166	張育瑞	下架
ASSASSIN'S	adventure	399	陳峻峰	下架
JUST DANCE	sports	399	葉凱毅	下架
急速領域	Speed	1000	陳峻峰	下架
跑跑跑	recommend	100000000	張育瑞	下架

點及查看遊戲庫

Sting遊戲購物網

Upload.cc: 免費圖片上傳網

Sting - YouTube

ilovedatabase.herokuapp.com/?Account=admin%40mail.com&password=admin123456#/gameManage

名稱	種類	價格	遊戲情況	管理
NBA 2K20	sports	66	上架中	下架
overcooked	Strategy	66	上架中	下架
overcooked 2	Strategy	166	上架中	下架
ASSASSIN'S	adventure	399	上架中	下架
JUST DANCE	sports	399	上架中	下架
急速領域	Speed	1000	上架中	下架
跑跑跑	recommend	10000000	上架中	下架
這三小	recommend	500	上架中	下架
不開心農場	recommend	333	上架中	下架
R6	Leisure	123	上架中	下架
神來也麻將	Leisure	1234	上架中	下架
海賊無雙	sports	0	上架中	下架

查看遊戲庫

© 2019 資料庫設計 - 遊戲購買系統 Design | Group 5

TOP

未上架的商品會在這裡看到

Sting遊戲購物網

Upload.cc: 免費圖片上傳網

Sting - YouTube

ilovedatabase.herokuapp.com/?Account=admin%40mail.com&password=admin123456#/gameBox

查看資料

管理員，你好！

遊戲庫：管理所有遊戲發布狀態

名稱	種類	價格	遊戲情況	管理
跑跑薑餅人	Leisure	100	未上架	編輯
fight	adventure	200	上架中	編輯
chicken horse	Strategy	400	上架中	編輯
Grand Thert Auto 5	Leisure	66	上架中	編輯
NBA 2K20	sports	66	上架中	編輯
overcooked	Strategy	66	上架中	編輯
overcooked 2	Strategy	166	上架中	編輯
ASSASSIN'S	adventure	399	上架中	編輯
JUST DANCE	sports	399	上架中	編輯
急速領域	Speed	1000	上架中	編輯

點及新增並新增遊戲

新增遊戲：加入新遊戲到遊戲庫

遊戲名稱: Young 遊戲類型: 精選推薦

製造商名稱: 遊戲橘子

遊戲價格(NT): 100 遊戲圖片(url): https://upload.cc/i1/2020/01/02/79RGt

遊戲介紹: ...

取消 新增

TOP

就會出現了

ilovedatabase.herokuapp.com/?Account=admin%40mail.com&password=admin123456#/gameBox

急速領域	Speed	1000	上架中	編輯
跑跑跑	recommend	100000000	上架中	編輯
這三小	recommend	500	上架中	編輯
不開心農場	recommend	333	上架中	編輯
R6	Leisure	123	上架中	編輯
神來也麻將	Leisure	1234	上架中	編輯
海賊無雙	sports	0	上架中	編輯
三國沙	Leisure	111111111	未上架	編輯
紫色恐怖	recommend	0	未上架	編輯
三國傳奇	Leisure	100	未上架	編輯
Young	recommend	100	上架中	編輯

返回 新增

© 2019 資料庫設計 - 遊戲購買系統 Design | Group 5

TOP

點選編輯進行編輯遊戲

編輯遊戲：更改遊戲資料 上架中

遊戲名稱

遊戲類型

製造商名稱

遊戲價格(NT)

遊戲圖片(url)

遊戲介紹

TOP

編輯後的價格

chicken horse	Strategy	400	上架中	編輯
Grand Thert Auto 5	Leisure	66	上架中	編輯
NBA 2K20	sports	66	上架中	編輯
overcooked	Strategy	66	上架中	編輯
overcooked 2	Strategy	166	上架中	編輯
ASSASSIN'S	adventure	399	上架中	編輯
JUST DANCE	sports	399	上架中	編輯
急速領域	Speed	1000	上架中	編輯
跑跑跑	recommend	100000000	上架中	編輯
這三小	recommend	500	上架中	編輯
不開心農場	recommend	333	上架中	編輯
R6	Leisure	123	上架中	編輯
神來也麻將	Leisure	2147483647	上架中	編輯
海賊無雙	sports	0	上架中	編輯
三國沙	Leisure	111111111	未上架	編輯
紫色恐怖	recommend	0	未上架	編輯
三國傳奇	Leisure	100	未上架	編輯
Young	recommend	100	上架中	編輯

TOP

Section 6 Suggestions of Database Tuning

在這次 database 的設計與實作當中，我們發現了使用者如果要查找遊戲需要使用分類來查找會很慢。所以我們可以使用關鍵字搜尋的方法來查找遊戲。我們可以參考老師在課堂上的教學，利用建立 index 的方式。將關鍵字所要搜尋到的部分，如遊戲名、出版商、作者等等建立 secondary index 再去做 query 的動作，這樣不僅可以大幅減低因大量重複的 linear search 所造成的效能損失，還能夠快速使用關鍵字搜尋遊戲。

在查詢過程中因為有些多對多關係的表格會需要分出多一個與他們之間的關聯表格。因此事，導致許多查詢方面需要跨過很多表格來搜尋會很浪費時間與精神，所以我們使用了 view 把三個表格合成一個表格來進行查詢。這樣不僅節省時間，在後端寫程式的時候也能夠一目瞭然清楚目前對應的關係。

在創建 view 有時候會遇到有兩個同名的欄位，這個是時候可以使用老師上課教的宣告別名 AS 來解決同名的狀況。

還有關於整體系統的交易部分，由於很難對於系統進行整合性的多使用者測試，因此目前看不出來在大量使用者的情況，交易之間的 Transaction 的處理，我想這個關於 Transaction 的部分，也是我們可以改善的部分。

Section 7 Additional Queries and Views

8.1 Database Queries

1. 會員管理子系統 (AMMS)

modify

“UPDATE Member SET value = value WHERE ID = value”

Login

“SELECT password, ID FROM Member WHERE Email = value”

2. 財務管理子系統 (FMS)

showOrder

“SELECT Order_ID, DateTime, SUM(price) AS price FROM order_view WHERE ID = value GROUP BY Order_ID”

showOrderGame

“SELECT name, price FROM order_view WHERE Order_ID = value”

showPersonGame

“SELECT Game_ID, name, type, description, photo, price, FROM order_view WHERE ID = value”

getOrderID

“SELECT Order_ID FROM Order WHERE ID = value AND DateTime = value”

3. 遊戲管理子系統 (GMS)

Shelves_takeOff

“UPDATE Game SET release_state = value WHERE Game_ID = value AND release_state = value”

Modify

“SET SQL_SAFE_UPDATES = 0; UPDATE Game SET value = value WHERE Game_ID = value; SET SQL_SAFE_UPDATES = 1;”

Gamelist

“SELECT type, price, Game_ID, description, photo, release_state, Aname, Gname FROM game_view WHERE release_state = value AND type = value”

4. 遊戲評分、評語管理子系統 (SCMS)

updateComment

“UPDATE comment SET description = description, stars = stars, time = time WHERE ID = value AND Game_ID = value”

getPersonComment

“SELECT ID, description, stars, time FROM comment WHERE Game_ID = value AND NOT ID = value”

getCommentID

“SELECT ID FROM comment WHERE Game_ID = value”

deleteGameComment

“DELETE FROM comment WHERE Game_ID = value AND ID = value”

5. 購物車管理子系統 (SHMS)

getCartID

“SELECT Cart_ID FROM Cart WHERE ID = value”

Remove(nest queries)

“DELETE FROM Cart_List WHERE Game_ID = value AND Cart_ID = (SELECT Cart_ID FROM Cart WHERE ID = value)”

List

“SELECT Game_ID, name, type, price FROM cart_view WHERE ID = value”

Clear(nest queries)

“DELETE FROM Cart_List WHERE Cart_ID = (SELECT Cart_ID FROM Cart WHERE ID = value)”

6. 結賬管理子系統 (BMS)

Payment

“INSERT into gamelibrary VALUE (Order_ID, Game_ID)”

“DELETE FROM cart_list WHERE Cart_ID = value AND Game_ID = value”

8.2 Database Views

1. 新增 order_view(ORDER BY)

“CREATE VIEW order_view AS SELECT O.Order_ID, O.ID, GL.Game_ID, G.name, G.price, O.DataTime FROM `order` O, `gamelibrary` GL, `game` G WHERE O.Order_ID = GL.Order_ID AND GL.Game_ID = G.Game_ID ORDER BY O.Order_ID;”

2. 新增 cart_view(ORDER BY)

“CREATE VIEW cart_view AS SELECT C.ID, CL.Game_ID, G.name, G.type, G.price FROM cart C, cart_list CL, game G WHERE C.Cart_ID = CL.Cart_ID AND CL.Game_ID = G.Game_ID ORDER BY C.ID;”

3. 新增 game_view(ORDER BY)

“CREATE VIEW game_view AS SELECT G.Game_ID, G.Author_ID, A.name as Aname, G.name as Gname, G.type, G.price, G.photo, G.description, G.release_state, G.time FROM `game` G, `author` A WHERE G.Author_ID = A.Author_ID ORDER BY G.Game_ID;”

Section 8 Conclusions and Future Work

9.1 Conclusions

葉凱毅心得：

我覺得這次的實作學習到很多東西，因為在人與人直接要一起互相討論之前，本身必須要有一些底子才能夠跟別人一起探討該怎麼設計該怎麼去寫資料庫。在這次的實作裏面我都有遇到很多的困難都是上網去找解決方法不然就是查看歷屆他們報告遇到的問題，還有參考許多的教材。而且課業壓力也蠻大的，很多東西都有期末專題期末考。突然發現都是一些心得...那這邊再補充一下關於學術性的討論，就是使用 view 可以減少 queries 的次數能夠更快搜尋那些多對多關係的表格，這樣搜尋會比較有效。

陳茗洋心得：

這次資料庫的實作專案中學到許多知識，首先就是學到資料庫表格的建立，修到正確後又發現沒有針對使用者作設計，這次專案有別於平常的作業繳交，是讓我們發揮創意，更讓我明白我們除了掌握知識技術以外也需要多想一些使用者者情境，這樣做出來的東西才能發威一定的價值，這次謝謝我的隊友們，我是負責寫前端介面，在學習 react 時也沒有少吃苦頭，雖然最後能幫上的忙不多，但我的隊友們也都很用的指導我，讓我更明白團隊合作的重要性。

黃宥勛心得：

剛開始用資料庫作業的時候，因為我們用 node js 連接資料庫，而我對 java 也不熟，所以中間也摸索了很久，最後做完之後我覺得要弄好一個完整的資料庫，所需要用到的東西遠遠不只是 mysql，還有各種需要用到的東西，感覺資料庫這門課就像個壓縮檔，點開之後才發現裡面有各式各樣的知識需要去搞懂。

劉政威心得：

這次資料庫專題對我來說算是真的學到蠻多東西的，因為雖然除了只有主要的 database 要撰寫，更需要熟悉前後端分離的 mvc 架構才能對於整個團隊更有效率的分工。一開始後端寫出來時，還摸不著頭緒如何與前端進行交流，後來一步步學習到如何使用 node js 的 express 套件做出自己的 api 並佈署到 heroku 上，算是蠻有趣得體驗。另外，雖然這次的 database 功能雖然不多，不過也是花了我們蠻多時間的，像是購物車到結帳的中間流程，就必須以 transaction 的操作去思考，來確保交易的完整性，又或是在 nodejs 裡 mysql query 的異步處理對於整個專案來說都是很重要的，總而言之，算是學到了相當多的課外技術，受益良多。

尹秉豪心得：

這次的期末專案，可能是我這四年來時間用的最長的或是心力搞最久的，我負責的是前端以及一些協助後段的銜接，這兩個東西都是我第一次的接觸，相當的不熟悉，也因為一些組員的個人關係，而造成其他的工作加大。

前端呢~我們使用的是 react.js 來撰寫，雖然相當的方便，有許多插件可以使用，但換來的是環境架設的困難、語言的限制和不同以及 bug 滿天飛的後果，為了學習這程式真的下了許多功夫跟時間在測試跟查資料，我一開始以為我們只要把頁面寫好就好，結果因為後端的需求而要大改特改、大加特加，搞得我一個頭兩個大，還好其他頁面我的前端夥伴可以先把其他頁面做好，我再去修那些細節，到了期末許多的期末專題都蹦了出來，每天都在熬夜搞這些東西，讓我身體都變得很虛。

我覺得最忙得是阿威，他負責的是大部分的前後端銜接跟一些後段，他因為我們前端不給力而造成他越到後面越忙，真的實在對不起他，連其他組 demo 時都還在搞 debug，對於沒辦法做到事而感到羞愧。

陳峻暉心得：

這次的期末專題之前，就蠻想接觸資料庫的，因為以前想要寫一些互動式網頁，但因為不懂資料庫而只能做出簡單的靜態網頁，實用性非常有限，所以就對這次專題蠻期待的。

但剛開始的時候還蠻沒有頭緒的，因為對於後端運作原理的認識挺模糊，所以在分工的時候還是先選了自己比較熟悉的前端，也在跟後端溝通時花了一些時間來釐清跟了解問題。

雖然這次大部分是做前端，不過對於後端跟資料庫有了一定的認識，雖然最後因為時間的關係，沒有辦理把網頁做的很完善，但還是很開心能做出第一個連結前後端的東西，只是這次的課程跟專題還只是帶我們入門而已，以後想要做出更好的東西勢必還要再自己去多多學習，希望自己能越來越掌握資料庫的技術

9.2 Future Work

在未來展望的部分，這次的專案是模仿知名網站 Stream 的系統，也簡化了許多功能，畢竟時間及能力有限，雖然已做到玩家的評分回饋系統，但系統還尚未整理出整體評分，以及呈獻給玩家推薦內容，未來希望能做出更貼近使用者的介面，像是新增遊戲規格，推薦內容及客服系統，未來希望能更吸引使用者來使用我們的系統，還有既然是遊戲購物系統，應該也要有活動特惠，以吸引玩家，但活動的部分尚需討論其架構內容應該有哪些，希望未來能做的更加完善。

這學期學習的 Database system，未來接觸的領域又更廣了，在這個資訊的時代，各行各業針對其未來發展都會做很多數據或整理出更有效率的系統，在各大中小型企業裡都一定擁有網路與資料庫方面的部門，這些部門著重於資訊網路系統與資料庫技術與應用，這些都是有可能成為未來出路之一。

如果我們掌握這些知識技術，可在企業界具體擔任許多職位，像是 DBA、資料庫設計師、資料庫工程師、資訊系統管理師、資訊網路管理師、系統分析師、資料庫系統顧問等。除了上述列舉之外凡是有類似性質的工作也是屬於資料庫系統相關工作職位。為了能夠掌握資訊科技領域發展的基礎，必預將所學應用到未來工作中，以代替傳統比較繁瑣且效率差的工作，而提升公司競爭力。

Glossary

DBMS	Database Management System，是一種針對物件資料庫，為管理資料庫而設計的大型電腦軟體管理系統。
GUI	Graphical User Interface，是指採用圖形方式顯示的電腦操作用戶介面
HTTP	是全球資訊網（WWW）上進行檔案交換的一套規則，用來界定首頁（Home Page）與 HTTP 伺服器之間的互動。相對於 TCP/IP 協定，HTTP 是一種應用協定。HTTP 最主要的特性之一在於，它是一個跨平台標準，存放在不同電腦系統的資料，都可以經由網際網路（Internet）互連。通訊時，一端必須執行 HTTP 用戶端程式，另一端必須執行 HTTP 伺服器軟體。
MySQL	一個開放原始碼的關聯式資料庫管理系統，原開發者為瑞典的 MySQL AB 公司，該公司於 2008 年被昇陽微系統（Sun Microsystems）收購。2009 年，甲骨文公司（Oracle）收購昇陽微系統公司，MySQL 成為 Oracle 旗下產品。
Heroku	是一個支援多種程式語言的雲平台即服務。在 2010 年被 Salesforce.com 收購。Heroku 作為最元祖的雲平台之一，從 2007 年 6 月起開發，當時它僅支援 Ruby，但後來增加了對 Java、Node.js、Scala、Clojure、Python 以及（未記錄在正式檔案上）PHP 和 Perl 的支援。基礎作業系統是 Debian，在最新的技術堆疊則是基於 Debian 的 Ubuntu。
React	<p>（有時叫 React.js 或 ReactJS）是一個為資料提供彩現為 HTML 視圖的開源 JavaScript 庫。React 視圖通常採用包含以自訂 HTML 標記規定的其他組件的組件彩現。React 為程式設計師提供了一種子組件不能直接影響外層組件（"data flows down"）的模型，資料改變時對 HTML 文件的有效更新，和現代單頁應用中組件之間乾淨的分離。</p> <p>它由 Facebook、Instagram 和一個由個人開發者和企業組成的社群維護。根據 JavaScript 分析服務 Libscore，React 目前正在被 Netflix、Imgur、Bleacher Report、Feedly、Airbnb、SeatGeek、HelloSign 等很多網站的首頁使用。</p>
Node.js	是能夠在伺服器端運行 JavaScript 的開放原始碼、跨平台 JavaScript 執行環境。Node.js 由 Node.js Foundation（已與 JS Foundation 合併為 OpenJS Foundation）持有和維護，亦為 Linux 基金會的專案。Node.js 採用 Google 開發的 V8 執行程式碼，使用事件驅動、非阻塞和非同步輸入輸出模型等技術來提高效能，可最佳化應用程式的傳輸量和規模。這些技術通常用於資料密集的即時應用程式。

References

網頁架構

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

圖片素材

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

React

https://www.w3schools.com/REACT/react_jsx.asp

React npm

<https://www.npmjs.com/package/react>

VSCode live share 使用

<https://xiaosean.github.io/vscode/2018-05-19-VSCode-Live-share/>

使用 vscode 添加 mysql 的時候報錯問題解決

<https://www.jianshu.com/p/73ab5f6b7478>

Visual Studio Code 鏈接 MySQL 數據庫並進行查詢

<https://blog.csdn.net/sigmarising/article/details/80665252>

MySQL 環境變數配置

<https://www.itread01.com/content/1544253495.html>

Bootstrap 中文網

<https://www.bootcss.com/>

Heroku 架設教學

<https://dwatow.github.io/2018/01-13-heroku-node-mvp/>

Upload.cc 網站用來儲存線上圖片

<https://upload.cc/>

Appendix

Appendix A

SQL statements for populating database.

新增賬戶 資料

INSERT INTO Member VALUE (First_Name, Last_Name, Sex, Email, Phone, Password, Credits, Class, Birthday)

新增購物車

INSERT INTO Cart VALUE (ID)

新增遊戲去訂單

INSERT INTO gamelibrary VALUE (Order_ID, Game_ID)

新增訂單

INSERT INTO Order VALUE (ID, DateTime)

新增遊戲

INSERT INTO Game VALUE (Author_ID, name, type, price, photo, description, release_state, time)

新增評論

INSERT INTO comment VALUE (ID, Game_ID, description, stars, time)

新增賬戶對應的購物車關聯

INSERT INTO Cart_List VALUE (CartID, gameID)

DEMO 影片鏈接

<https://www.youtube.com/watch?v=QbAbWs0GGmE>

網站鏈接

<https://ilovedatabase.herokuapp.com/#/>