

# 珍室鞋樓

## 系統需求規格書

### Software Requirements Specification (SRS)

Version: 4.0

姓名	學號	E-mail
何柏憲	108590044	t108590044@ntut.org.tw
黃品宥	108590021	t108590021@ntut.org.tw
蔡承修	108590030	t108590030@ntut.org.tw
林奕杰	108590032	t108590032@ntut.org.tw
許晉緯	108590034	t108590034@ntut.org.tw
張君瑋	108590035	t108590035@ntut.org.tw
沈宗毅	108590036	t108590036@ntut.org.tw

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

10/05/2021

# 目錄 (Table of Contents)

<b>Section 1</b>	<b>簡介 (Introduction)</b>	<b>1</b>
1.1	目的 (Purpose)	1
1.2	系統名稱 (Identification)	2
1.3	概觀 (Overview)	2
1.4	符號描述 (Notation Description)	3
<b>Section 2</b>	<b>系統(System)</b>	<b>5</b>
2.1	系統描述 (System Description)	5
2.1.1	系統架構圖 (System Architecture Diagram)	5
2.2	操作概念 (Operational Concepts or User Stories)	6
2.3	功能性需求 (Functional Requirements)	7
2.4	資料需求 (Data Requirements)	8
2.5	非功能性需求 (Non-Functional Requirements)	9
2.5.1	效能需求 (Performance Requirements)	9
2.6	介面需求 (Interface Requirements)	9
2.6.1	使用者介面需求 (User Interfaces Requirements)	9
2.7	其他需求 (Other Requirements)	10
2.7.1	環境需求 (Environmental Requirement)	10

2.8 商業規則與限制 (Business Rules and Integrity Constrains).....	10
Section 3 資料庫概念設計(Conceptual Design of Datebase) .....	11
Section 4 邏輯資料庫綱要(Logic Database Schema) .....	12
4.1 Schema of the Database .....	12
4.2 Domain of the Database .....	13
4.3 Expectation of the possible DB operations, frequencies and data volumes	
18	
4.4 SQL Statements Used to Construct the Schema .....	19
Section 5 Functional Dependencies and Database Normalization.....	22
5.1 Funtional Dependencies .....	22
Section 6 資料庫系統的使用(The Use of the Database System) .....	23
6.1 系統安裝 (System Installation Description) .....	23
6.2 系統使用 (The Use of the System) .....	23
Section 7 Glossary.....	錯誤! 尚未定義書籤。
References .....	錯誤! 尚未定義書籤。
Appendix.....	錯誤! 尚未定義書籤。

## Section 1 簡介 (Introduction)

### 1.1 目的 (Purpose)

在本學期的資料庫課程中，學到了如何設計資料庫。為了提升實務上的能力，我們小組將協力完成一項大專案，除了資料庫架構，還會加入網頁設計等等的基礎能力。該專案旨在完成一個專有品牌購物網站「珍室鞋樓」，該系統讓顧客能在網路上買鞋、查詢等，以精緻簡單的介面，讓顧客享有難以忘懷的買鞋體驗。

並根據使用者身分將擁有以下功能：

- 管理員：
  1. 創建和管理用戶帳戶
  2. 啟用或關閉折扣活動
- 員工：
  1. 處理產品存貨
  2. 更新產品與訂單資訊
  3. 產品數據統計
- 客戶：
  1. 瀏覽、搜索、訂購商品
  2. 查詢目前購物車、訂單狀況

## 1.2 系統名稱 (Identification)

主系統：

線上商場系統(Web-based Mall Business System, WMBS)

子系統：

優惠管理子系統(Discount Management Subsystem, DMS)

帳戶管理子系統(Account Management Subsystem, AMS)

購物車子系統(Shopping Cart Subsystem,SCS)

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

存貨資料子系統(Inventory Data Subsystem, IDS)

商品管理子系統(Commodity Management Subsystem, CMS)

商品查詢子系統(Commodity Inquiry Subsystem, CIS)

訂單管理子系統(Order Management Subsystem, OMS)

資料庫子系統(Database Subsystem, DBS)

## 1.3 概觀 (Overview)

依現代網站設計趨勢，選擇前後端分離，讓成員能更有效的分工。這次專案主要是學習資料庫設計相關，因此前端選擇 React，並套用現成模板；後端使用 PHP 語言+Laravel 框架，資料庫選擇 PostgreSQL 作為 DBMS，PostgreSQL 在高併發下能正常工作，也是相容 SQL 語言，相較於 MySQL、MariaDB，PostgreSQL 擁有優勢。

#### 1.4 符號描述 (Notation Description)

WMBS 1.0.0	The WMBS system will be labeled with the number 1.0.0.
DMS 1.1.n	The DMS system will be labeled with the number 1.1.n.
AMS 1.2.n	The AMS system will be labeled with the number 1.2.n.
SCS 1.3.n	The SCS system will be labeled with the number 1.3.n.
FMS 1.4.n	The FMS system will be labeled with the number 1.4.n.
IDS 1.5.n	The IDS system will be labeled with the number 1.5.n.
CMS 1.6.n	The CMS system will be labeled with the number 1.6.n.
CIS 1.7.n	The CIS system will be labeled with the number 1.7.n.
OMS 1.8.n	The OMS system will be labeled with the number 1.8.n.
DBS 1.9.n	The DBS system will be labeled with the number 1.9.n.

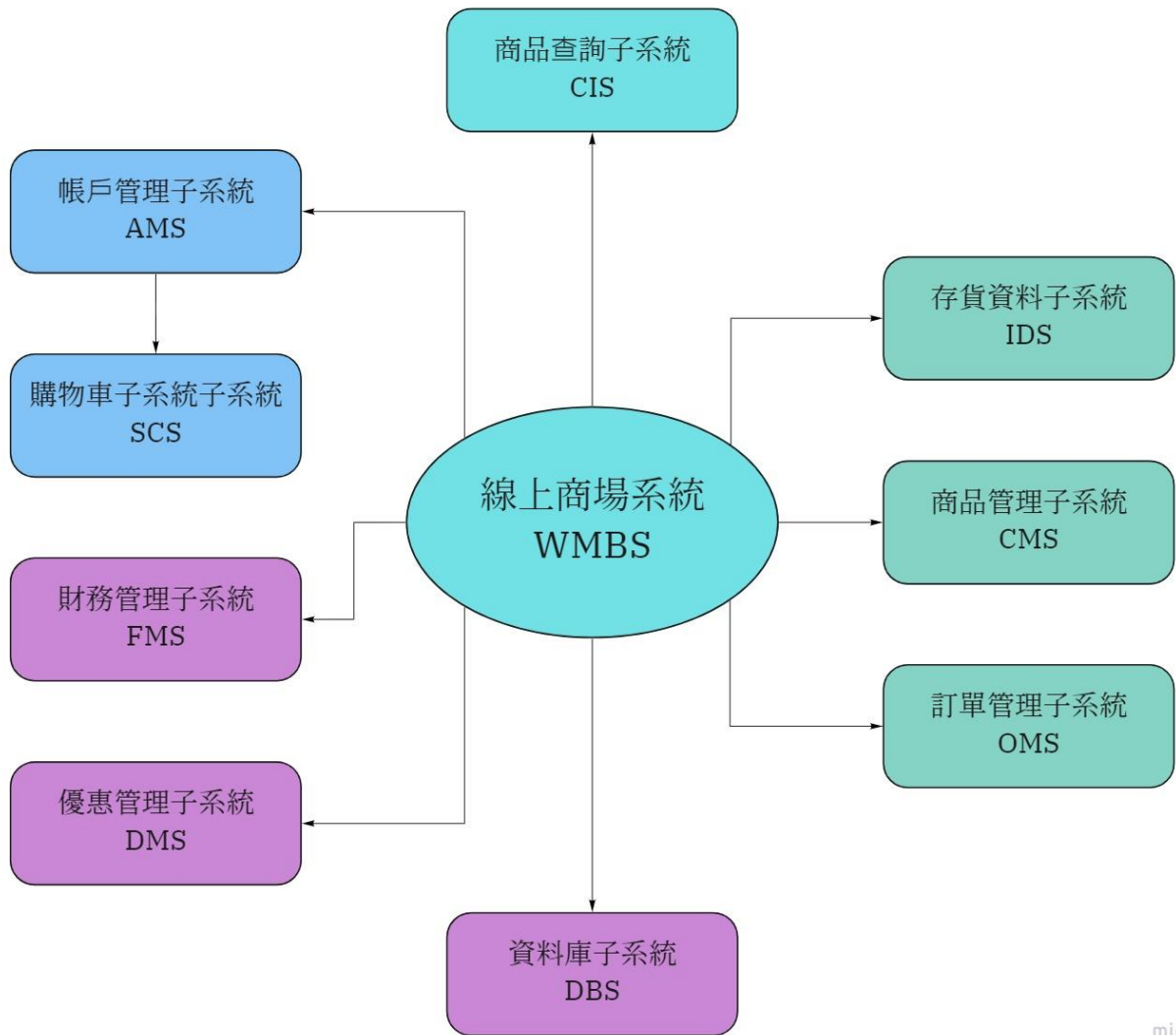
WMBS -F-nnn	WMBS 功能性需求(Functional Requirements)
WMBS -N-nnn	WMBS 非功能性需求(Non-Functional Requirements)
DMS -F-nnn	DMS 功能性需求(Functional Requirements)
DMS -N-nnn	DMS 非功能性需求(Non-Functional Requirements)
AMS -F-nnn	AMS 功能性需求(Functional Requirements)
AMS -N-nnn	AMS 非功能性需求(Non-Functional Requirements)
SCS -F-nnn	SCS 功能性需求(Functional Requirements)

SCS -N-nnn	SCS 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)
IDS -F-nnn	IDS 功能性需求(Functional Requirements)
IDS -N-nnn	IDS 非功能性需求(Non-Functional Requirements)
CMS -F-nnn	CMS 功能性需求(Functional Requirements)
CMS -N-nnn	CMS 非功能性需求(Non-Functional Requirements)
CIS -F-nnn	CIS 功能性需求(Functional Requirements)
CIS -N-nnn	CIS 非功能性需求(Non-Functional Requirements)
OMS -F-nnn	OMS 功能性需求(Functional Requirements)
OMS -N-nnn	OMS 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 2 系統(System)

### 2.1 系統描述 (System Description)

#### 2.1.1 系統架構圖 (System Architecture Diagram)



miro



## 2.2 操作概念 (Operational Concepts or User Stories)

### Scenario 1：客戶操作概念

透過帳戶管理子系統 ( AMS ) 確認是否有登入，若確認已登入，則可經由商品查詢子系統系統 ( CIS ) 瀏覽商品，選購商品並加入購物車子系統 ( SCS )；若未登入，則只能瀏覽商品，不能使用購物車。

### Scenario 2：員工操作概念

利用存貨資料子系統 ( IDS ) 保證商品庫存充足，並透過訂單管理子系統 ( OMS ) 確定每筆交易狀態，無論是上游或是顧客；最後使用商品管理子系統 ( CMS )，控制商品上下架，以及價格等等。

### Scenario 3：管理員操作概念

以資料庫子系統 ( DBS ) 記錄所有帳戶的操作，防止被鑽漏洞；透過帳戶管理子系統 ( AMS ) 以控管所有帳戶，像是 VIP 或黑名單；利用財務管理子系統 ( FMS ) 確保金流無誤，同時計算盈虧；經由優惠管理子系統 ( DMS ) 記錄折扣函式，同時啟用或關閉活動優惠。

### 2.3 功能性需求 (Functional Requirements)

需求編號	需求描述
FR 001	註冊帳戶
FR 002	帳戶 新增/修改/刪除/查詢/啟用/關閉，以及修改權限
FR 003	商品 新增/修改/刪除/查詢
FR 004	商品 加入/移除 購物車
FR 005	折扣優惠 新增/修改/刪除/查詢/啟用/關閉
FR 006	結帳系統
FR 007	訂單 新增/修改/刪除/查詢
FR 008	商品庫存確認與補充
FR 009	記錄金錢流向
FR 010	記錄所有帳戶的任何操作

## 2.4 資料需求 (Data Requirements)

需求編號	需求描述
DR 001	使用者姓名
DR 002	身份證字號
DR 003	出生年月日
DR 004	電話號碼
DR 005	電子郵件
DR 006	使用者地址

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

### 2.5.1 效能需求 (Performance Requirements)

需求編號	需求描述
NFR 001	網頁載入速度不應大於 5 秒

## 2.6 介面需求 (Interface Requirements)

### 2.6.1 使用者介面需求 (User Interfaces Requirements)

#### 2.6.1.1 客戶介面需求

需求編號	需求描述
CUI 001	會員登入及註冊介面
CUI 002	商品瀏覽介面
CUI 003	購物車介面
CUI 004	結帳介面

#### 2.6.1.2 員工介面需求

需求編號	需求描述
SUI 001	商品庫存介面
SUI 002	訂單管理介面
SUI 003	商品管理介面

### 2.6.1.3 管理員介面需求

需求編號	需求描述
AUI 001	帳戶管理介面
AUI 002	折扣優惠管理介面
AUI 003	金流管理介面
AUI 004	帳戶操作記錄介面

## 2.7 其他需求 (Other Requirements)

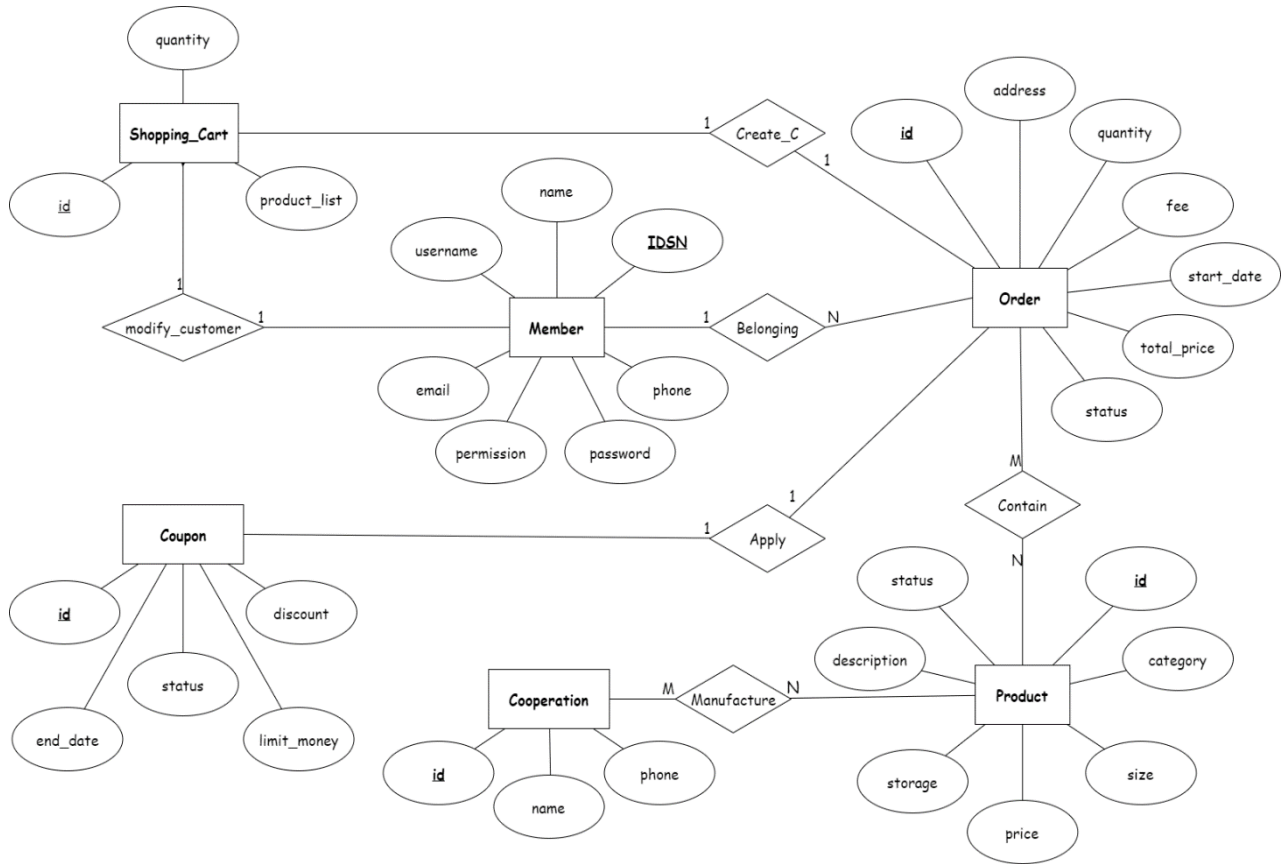
### 2.7.1 環境需求 (Environmental Requirement)

需求編號	需求描述
ER 001	需要在有網路的環境

## 2.8 商業規則與限制 (Business Rules and Integrity Constrains)

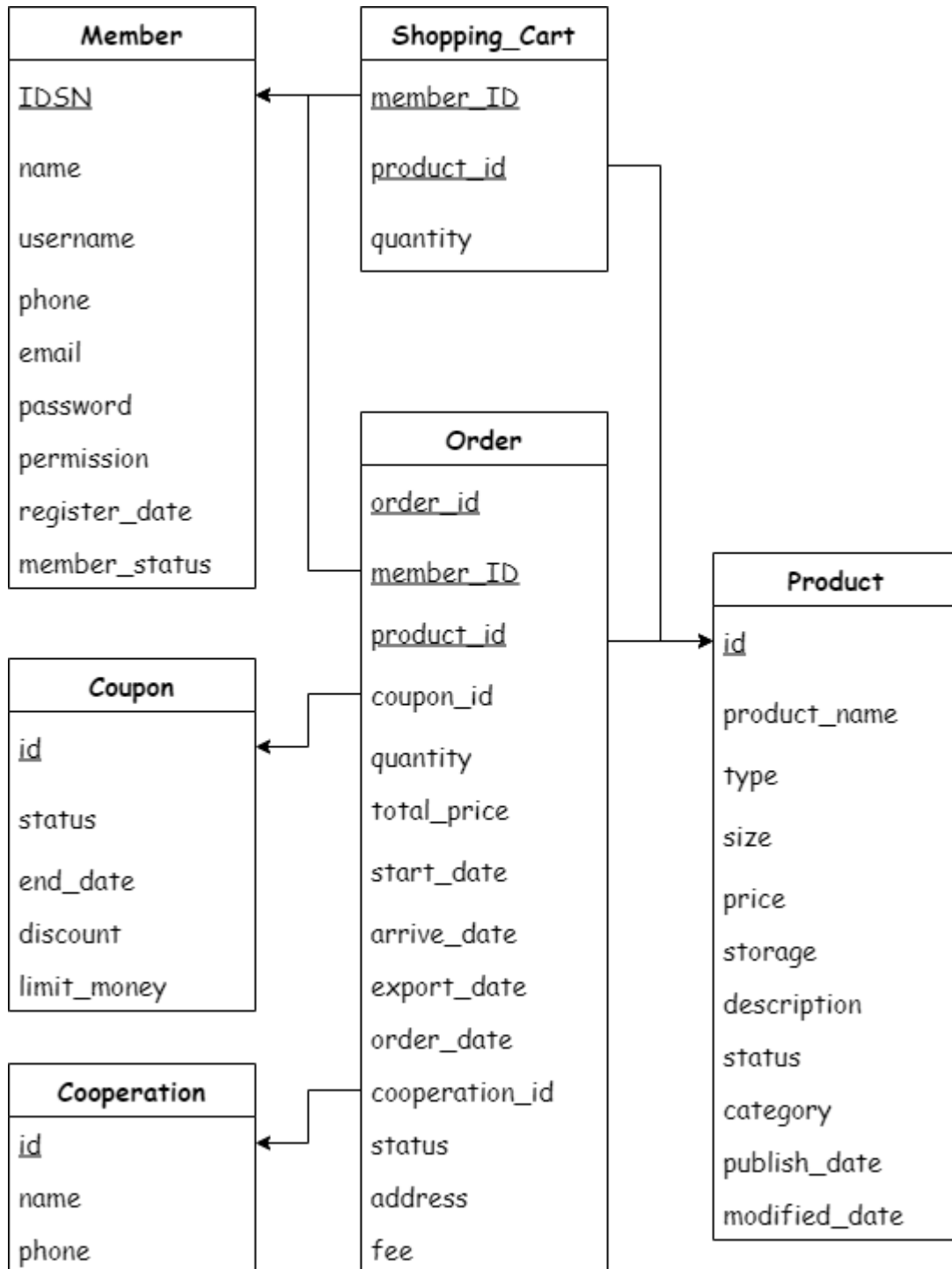
需求編號	需求描述
RC 001	員工與管理員不可擁有購物功能

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



## Section 4 邏輯資料庫綱要(Logic Database Schema)

### 4.1 Schema of the Database



## 4.2 Domain of the Database

Member				
Description : 存放各個與使用者有關的資料				
Attribute	Domain Type	Key	Nullable	Decription
IDSN	integer	Primary	not null	會員身分證號碼
name	varchar			會員暱稱
username	varchar		not null	會員帳號
phone	Integer	unique	not null	會員電話
email	varchar	unique	not null	會員信箱
Password	varchar		not null	會員密碼
permission	integer		not null	權限等級 0：管理員 1：員工 2：客戶
register_date	timestamp		not null	會員註冊時間
member_status	integer		not null	會員帳號狀態 0：停用 1：正常



Shopping_Cart				
Description : 存放購物車的資料				
Attribute	Domain Type	Key	Nullable	Decription
member_id	integer	primary foreign	not null	顧客 id Reference(Member)
product_id	integer	foreign	not null	商品 id Reference(Product)
quantity	interger		not null	數量

Order				
Description : 存放訂單的資料				
Attribute	Domain Type	Key	Nullable	Decription
id	integer	primary	not null	訂單編號
member_id	integer	foreign	not null	購買人 id reference(member)
coupon_id	integer	foreign		優惠券 id reference(coupon)
product_id	integer	foreign	not null	商品 id

				reference(product)
cooperation_id	integer	foreign		廠商 id
quantity	integer		not null	商品數量
order_date	datetime		not null	下訂時間
export_date	datetime		not null	出貨時間
arrive_date	datetime		not null	抵達時間
total_price	integer		not null	總金額
status	integer		not null	訂單狀態  0 : 待處理  1 : 已出貨  2 : 訂單完成
address	varchar		not null	訂單目的地
fee	integer		not null	運費

Cooperation				
Description：存放廠商的資料				
Attribute	Domain Type	Key	Nullable	Decription
id	integer	primary	not null	廠商編號
name	varchar		not null	廠商名稱
phone	varchar		not null	廠商電話

Product				
Description：存放商品的資料				
Attribute	Domain Type	Key	Nullable	Decription
id	integer	primary	not null	商品編號
product_name	varchar	unique	not null	商品名稱
type	varchar		not null	商品類型
size	integer		not null	商品尺寸
price	integer		not null	商品價格>0
storage	integer		not null	庫存數量>=0
description	varchar		not null	商品描述
status	integer		not null	商品狀態

				0 : 下架 1 : 上架
category	varchar		not null	商品分類
publish_date	timestamp		not null	商品創建日期
modified_date	timestamp		not null	商品修改日期

Coupon				
Description : 存放優惠券的資料				
Attribute	Domain Type	Key	Nullable	Decription
id	integer	primary	not null	優惠券編號
status	integer		not null	優惠券狀態 0 : 不可使用 1 : 可使用
end_date	timestamp		not null	優惠券結束日期
discount	double			優惠券折扣
limit_money	integer			最低金額>0

#### 4.3 Expectation of the possible DB operations, frequencies and data volumes

表格	可能操作	頻率(per day)	資料量	系統負擔
Member	使用者登入驗證	200	400	80000(Query/day)
Member	新增會員資料	100	500	50000(Query/day)
Member	更新會員資料	5	X	5(Insert/day)
Coupon	查詢優惠券資料	50	200	10000(Query/day)
Coupon	新增優惠券	5	X	5(Insert/day)
Coupon	更新優惠券	10	200	2000(Query/day)
Shopping_Cart	會員新增購物車	50	200	10000(Query/day)
Shopping_Cart	會員確認訂單	10	200	2000(Query/day)
Order	訂單狀態查詢	100	400	40000(Query/day)
Order	新增訂單	10	X	10(Insert/day)
Order	訂單更新	20	400	8000(Query/day)
Cooperation	新增公司資料	2	X	2(Insert/day)
Cooperation	公司資料更新	1	100	100(Query/day) 1(update/day)
Product	查詢商品資料	300	300	90000(Qurey/day)
Product	新增商品	10	X	10(Insert/day)
Product	更新商品資料	600	300	180000(Query/day)

## 4.4 SQL Statements Used to Construct the Schema

### product

```
1 CREATE TABLE `product`(  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `product_name` varchar(45) NOT NULL UNIQUE,  
4   `size` int(11) NOT NULL,  
5   `price` int(11) NOT NULL,  
6   `storage` int(11) NOT NULL,  
7   `description` varchar(45) NOT NULL,  
8   `status` int(11) NOT NULL,  
9   `category` varchar(45) NOT NULL,  
10  `publish_date` DATETIME NOT NULL,  
11  `modified_date` DATETIME NOT NULL,  
12  PRIMARY KEY(id)  
13 ) ENGINE = InnoDB;
```

### order

```
1 CREATE TABLE `order`(  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `member_id` int(11) NOT NULL,  
4   `coupon_id` int(11),  
5   `product_id` int(11) NOT NULL,  
6   `cooperation_id` int(11),  
7   `quantity` int(11) NOT NULL,  
8   `order_date` DATETIME NOT NULL,  
9   `export_date` DATETIME NOT NULL,  
10  `arrive_date` DATETIME NOT NULL,  
11  `payment_method` int(11) NOT NULL,  
12  `status` int(11) NOT NULL,  
13  `address` varchar(45) NOT NULL,  
14  `fee` int(11) NOT NULL,  
15  PRIMARY KEY(id),  
16  FOREIGN KEY(member_id) REFERENCES member(IDSN) ON UPDATE CASCADE ON DELETE CASCADE,  
17  FOREIGN KEY(coupon_id) REFERENCES coupon(id) ON UPDATE CASCADE ON DELETE CASCADE,  
18  FOREIGN KEY(product_id) REFERENCES product(id) ON UPDATE CASCADE ON DELETE CASCADE,  
19  FOREIGN KEY(cooperation_id) REFERENCES cooperation(id) ON UPDATE CASCADE ON DELETE CASCADE  
20 ) ENGINE = InnoDB;
```

### member

```

1 CREATE TABLE `member`(
2     `idsn` int(11) NOT NULL UNIQUE AUTO_INCREMENT,
3     `name` varchar(45) NOT NULL,
4     `username` varchar(45) NOT NULL UNIQUE,
5     `phone` varchar(45) NOT NULL UNIQUE,
6     `email` varchar(45) NOT NULL UNIQUE,
7     `password` varchar(45) NOT NULL,
8     `permission` int(11) NOT NULL,
9     `register_date` DATETIME NOT NULL,
10    `member_status` int(11) NOT NULL,
11    PRIMARY KEY(idsn)
12 ) ENGINE = InnoDB;

```

## coupon

```

1 CREATE TABLE `coupon`(
2     `id` int(11) NOT NULL AUTO_INCREMENT,
3     `status` int(11) NOT NULL,
4     `start_date` DATETIME NOT NULL,
5     `end_date` DATETIME NOT NULL,
6     `type` int(11) NOT NULL,
7     `discount` double,
8     `limit_money` int(11),
9     PRIMARY KEY(id)
10 ) ENGINE = InnoDB;

```

## cooperation

```

1 CREATE TABLE `cooperation`(
2     `id` int(11) NOT NULL AUTO_INCREMENT,
3     `name` varchar(48) NOT NULL,
4     `phone` varchar(45) NOT NULL,
5     PRIMARY KEY(id)
6 ) ENGINE = InnoDB;

```

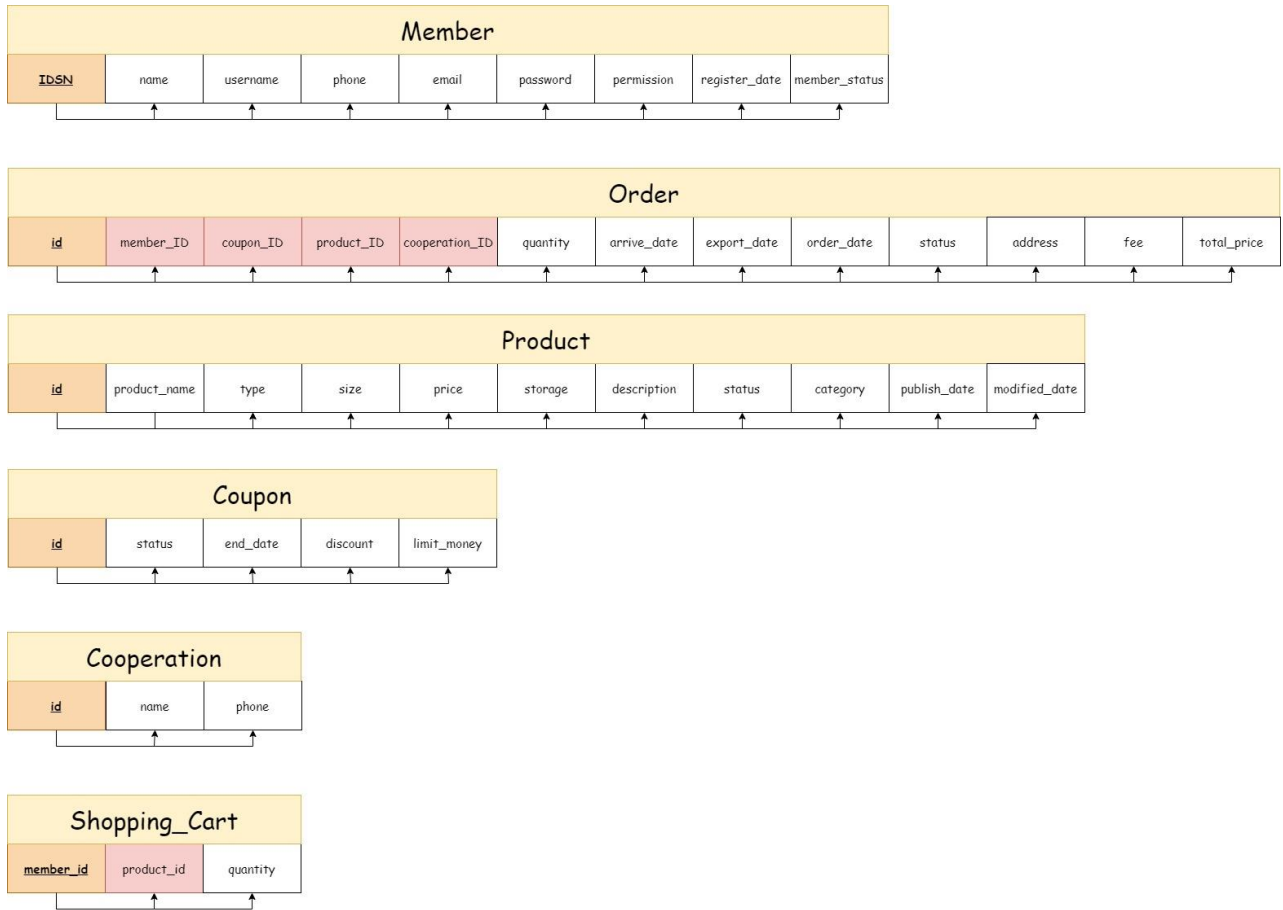
## shopping\_cart

```
1 CREATE TABLE `shopping_cart`(  
2     `member_id` int(11) NOT NULL,  
3     `product_id` int(11) NOT NULL,  
4     `quantity` int(11) NOT NULL,  
5     PRIMARY KEY(member_id),  
6     FOREIGN KEY(member_id) REFERENCES member(IDSN) ON UPDATE CASCADE ON DELETE CAS  
7     FOREIGN KEY(product_id) REFERENCES product(id) ON UPDATE CASCADE ON DELETE CAS  
8 ) ENGINE = InnoDB;
```



## Section 5 Functional Dependencies and Database Normalization

### 5.1 Funtional Dependencies

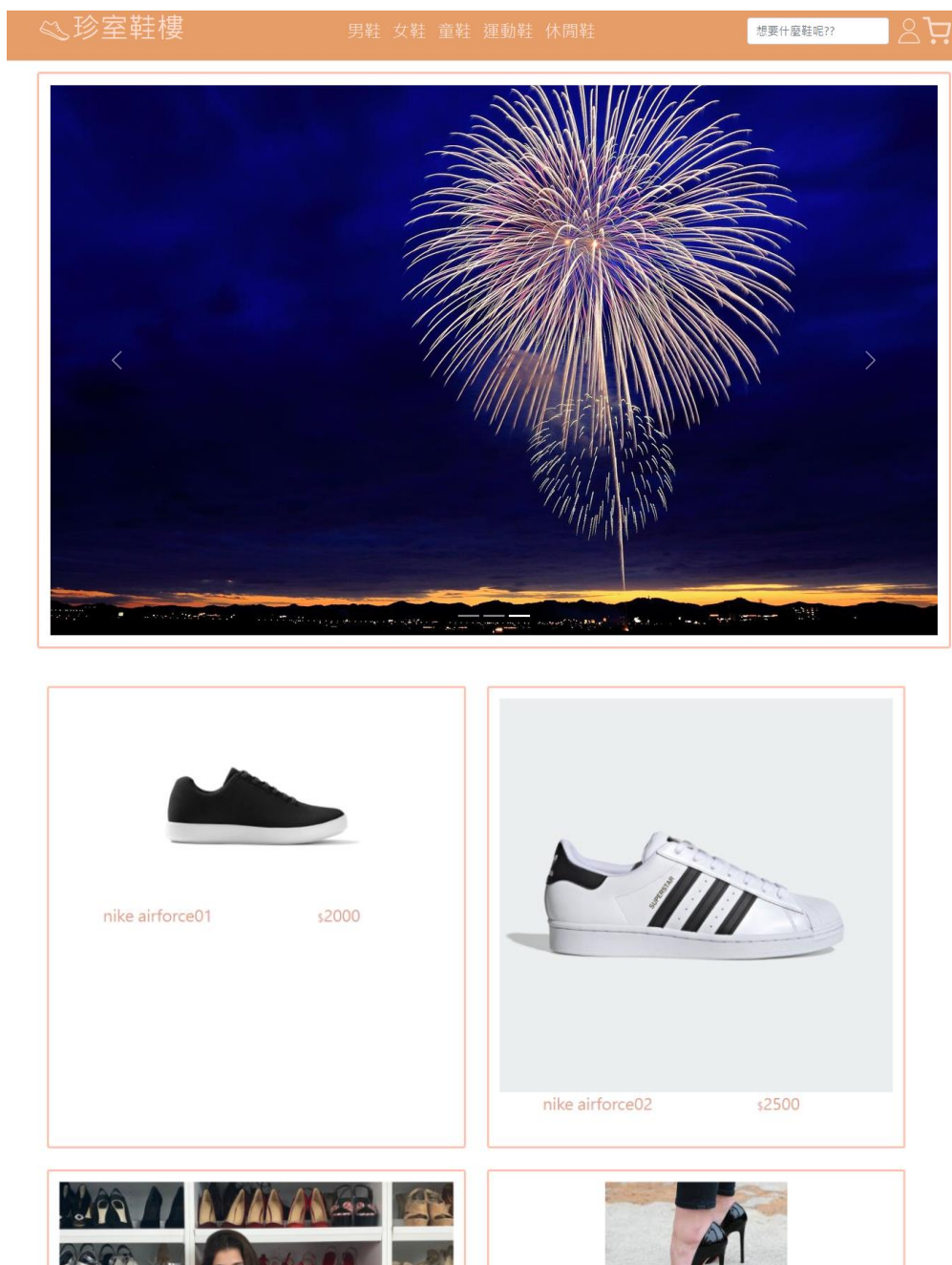


## Section 6 資料庫系統的使用(The Use of the Database System)

### 6.1 系統安裝 (System Installation Description)

### 6.2 系統使用 (The Use of the System)

#### ✧ 首頁畫面



## ✧ 會員登入介面

 珍室鞋樓

會員

密碼

登入

註冊

## ✧ 註冊畫面

 珍室鞋樓

電子郵件

姓名

電話

帳號

密碼

完成註冊

## ✧ 會員更改密碼

修改會員資料

OldPassword

NewPassword

修改密碼

登出

## ✧ 商品列表


珍室鞋樓

男鞋 女鞋 童鞋 運動鞋 休閒鞋

想要什麼鞋呢??


人 車

商品列表



nike airforce01

\$2000



nike airforce02

\$2500

## 管理資料庫介面

The screenshot displays the MySQL Workbench interface for a local instance named 'MySQL10055'. The left sidebar contains navigation panels for 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), 'INSTANCE' (Startup/Shutdown, Server Logs, Options File), 'PERFORMANCE' (Dashboard, Performance Reports, Performance Schema Setup), and 'Administration' (Schemas, Information). The main query editor shows a series of SQL commands for creating and modifying tables, and inserting data. The results pane at the bottom displays a table with columns: id, product\_name, size, price, storage, description, status, category, address\_date, and modified\_date. The table contains 5 rows of data. The bottom status bar shows the current action: 'select \* from product LIMIT 0, 1000'.

Query 1

```

CREATE TABLE `order` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `product_name` VARCHAR(255) NOT NULL,
  `size` INT(11) NOT NULL,
  `price` DECIMAL(10,2) NOT NULL,
  `storage` INT(11) NOT NULL,
  `description` VARCHAR(255) NOT NULL,
  `status` VARCHAR(255) NOT NULL,
  `category` VARCHAR(255) NOT NULL,
  `address_date` DATETIME NOT NULL,
  `modified_date` DATETIME NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

ALTER TABLE `product`
MODIFY `id` INT(11) NOT NULL AUTO_INCREMENT;

select * from product;

```

Results Grid

#	id	product_name	size	price	storage	description	status	category	address_date	modified_date
1	1	nike air force 27	20	2000	100	all white	1	man	2021-12-21 12:12:12	2021-12-21 12:12:12
2	2	nike air force 27	25	2500	100	all blue	1	man	2022-01-01 12:12:12	2022-01-01 12:12:12
3	3	nike air force 27	20	2000	100	all yellow	1	women	2022-01-01 12:12:12	2022-01-01 12:12:12
4	4	nike air force 27	25	2500	100	all purple	1	women	2022-01-01 12:12:12	2022-01-01 12:12:12
5	5	nike air force 27	20	2000	100	all red	1	child	2022-01-11 20:15:43	2022-01-11 20:15:43

product 1 x

Action Output

#	Time	Action	Message	Duration / Fetch
210	20:10:26	delete from product where id = 5 OR id = 7	2 row(s) affected	0.000 sec
211	21:04:38	select * from member LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
212	22:58:55	select * from member LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
213	23:36:37	show tables	6 row(s) returned	0.015 sec / 0.000 sec
214	23:38:04	select * from product LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

## Section 7 結論與未來規劃(Conclusions and Future Work)

### ◆ 蔡承修

資料庫系統這堂課使我獲益良多，尤其是小組專案的部分，讓我們這些新手知道一個專案從 0 到有應該要進行怎麼樣的分工合作，雖然我們這次做的沒有到很好，但是有了這次經驗，相信日後遇到了需要團隊合作完成的專案，我們都能比較得心應手。我們這次的資料庫前後端使用的分別為 vue.js、express.js 和 node.js，只有 node.js 有在上學期接觸過，稍微知道一點語法而已，其他都是靠大家不斷的摸索才能夠完成專案，感謝各位的努力。

### ◆ 何柏憲

這次做資料庫的專題，我遇到許多的困難，但也學到了許多的東西。首先就是分工分配不均，因為是第一次接觸到這種大專案造成，我們組內完全沒有人會一開始都毫無頭緒，導致上分工上相當粗糙，而且我們大部分成員都是系學會的成員，我們要一邊舉辦活動一邊兼顧課業導致時間非常不足，不過後來還是有生出了一些東西。想到在報告前一個禮拜每天都在熬夜加班，只為了多寫幾個功能跟 A P I，想想都覺得好累，不過還是要感謝大家的幫忙。

## ◆ 張君瑋

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，因為我們用 node js 連接資料庫，然後我是寫前端的，當初在連接前後端發生了很多問題花了許多時間才解決，這次的專題不只是單單使用 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，謝謝老師這學期的教學。

## ◆ 許晉緯

這次的課程，我學習到了非常多的關鍵，從應該如何建立一個 ER Model，到最後的，將冗餘去除的正規化，這些都讓我對於資料庫有更深的理解，也理解到一個優秀的資料庫，是需要仔細規劃的。而本次收穫最大的，就是負責前端，雖然起初的進度不是相當理想，每個人都不知道如何開始，但經過討論，以及不斷測試，最後還是做出了不錯的項目。技術方面學習到了，使用 Vue 去渲染頁面，且用 Bootstrap 5 簡單的切版，並跟後端的人溝通，確定所需要的資料，藉由 API，對資料庫進行取得、搜索、更新、刪除等操作，讓我瞭解到小組內溝通的重要性，也讓我了解前端工程師，在一個專案應該要負責什麼，收穫滿滿。

## ◆ 林奕杰

在這學期的資料庫系統中，我學習到許多跟資料庫有關的知識，知道了資料庫並不只是跟表面看起來一樣簡單，有許許多多更進階的概念，像是一般化、功能性相依還有讓我們很頭痛的 ER-Diagram，總而言之，這學期真的學到了非常多的東西，還有專案的部分也是接觸到了有關於前後端的設計，讓我受益良多。

## ◆ 黃品宥

上完了這門課，我了解到原來資料庫並不是想像中的那麼簡單，他有很多複雜的關係要等你去釐清，原本的我以為資料庫就是單純的儲存資料，並且可以拿來應用的系統，但實際上他必須要懂非常多的資料關係，像是 primary key 、foreign key 之類的，懂這些並不是讓整個系統可以運作，而是讓整個系統可以運作的更加順暢並且消耗最少的資源達到我們需要的目的。

## ◆ 沈宗毅

這學期的資料庫學了不少東西，特別是在小組專案的時候，不僅要主動設計一些資料庫的東西，還有了解一些前後端的架構，連結方式還有一些東西是我之前沒有經驗的，這次我做的很多部分是後端，學了很多後端的技術，從一開始什麼架構都不知道，雖然不敢說到最後出來的結果多好，但對我來說是一大突破。



