

邀驗鑑貨

系統需求規格書

Software Requirements Specification (SRS)

Version: 1.0

姓名	學號	E-mail
林昕昱	109590001	t109590001@ntut.org.tw
陳右儒	109590005	t109590005@ntut.org.tw
楊明哲	109590009	t109590009@ntut.org.tw
楊挺煜	109590015	t109590015@ntut.org.tw
王郁鈞	109590020	t109590020@ntut.org.tw
古佳豫	109590021	t109590021@ntut.org.tw

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

11/01/2022

目錄 (Table of Contents)

Section 1 簡介 (Introduction)	1
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) (if any)	2
Section 2 系統(System)	
2.1 系統描述 (System Description)	4
2.1.1 系統架構圖 (System Architecture Diagram)	4
2.2 操作概念 (Operational Concepts or User Stories)	5
2.3 功能性需求 (Functional Requirements)	5
2.4 資料需求 (Data Requirements)	6
2.5 非功能性需求 (Non-Functional Requirements)	6
2.5.1 效能需求 (Performance Requirements)	6
2.5.2 資安需求 (Security Requirements) (if any)	6
2.6 介面需求 (Interface Requirements)	6
2.6.1 使用者介面需求 (User Interfaces Requirements)	6
2.6.2 外部介面需求 (External Interface Requirements) (if any)	7
2.6.3 內部介面需求 (Internal Interface Requirements) (if any)	7
2.7 其他需求 (Other Requirements)	8
2.7.1 環境需求 (Environmental Requirement)	8
2.7.2 安裝需求 (Installation Requirement)	9
2.7.3 測試需求 (Test Requirements) (if any)	8
2.8 商業規則與限制 (Business Rules and Integrity Constraints)	8
Section 3 資料庫概念設計(Conceptual Design of the Database)	9
3.1 Entity-Relationship (ER) Model	9
Section 4 資料庫概念設計(Conceptual Design of the Database)	10
4.1 Schema of the Database	10
4.2 Domain of the DataBase	11
4.3 Possible DB operations frequencies and data volumes	
4.4 SQL Statements Used to Construct the Schema	
4.5 The implementation of tables in target DBMS	
Section 5 功能性依賴(Functional Dependencies and Database Normalization)	
Section 6 資料庫概念設計 (Additional Queries and Views)	
6.1 首頁畫面	
6.2 登入畫面	
References	14
Appendix	15

Section 1 簡介 (Introduction)

1.1 目的 (Purpose)

此系統我們要做的是線上購物平台，在平台上可以使賣家發布商品資訊、介紹商品內容等等，例如：各類生活用品、家具、3c產品...，用標籤化進行篩選，使買家可以輕鬆找到需要的商品，這個世界有供也有需，透過我們的平台，讓每個人都可以促進經濟發展。

並根據使用者身分將擁有以下功能：

- 買家：
 - 搜尋商品
 - 與賣家私訊
 - 購買商品
- 賣家：
 - 刊登商品資訊
 - 檢視買家私訊
 - 對商品制定折扣
- 管理員：
 - 審核買家與賣家資料
 - 刪除假帳號、不實刊登
 - 進行商品內容驗證
- 訪客：
 - 查看賣家商品（僅檢視功能 若要購買 需要註冊）

1.2 系統名稱 (Identification)

主系統：線上購物系統(Web-based Shopping System,WSS)

- 標籤索引管理子系統(Label Retrieval Management Subsystem,LRMS)
- 帳號管理子系統(Account Management Subsystem,AMS)
- 商品管理子系統(Product Management Subsystem,PMS)
- 歷史訂單子系統(Historical Order Subsystem,HOS)
- 優惠券管理子系統(Discount Management Subsystem,DMS)
- 購物車子系統(Shopping Cart Subsystem,SCS)
- 管理員操作子系統(Admin Operation Subsystem,AOS)
- 資料庫系統(Database System,DBS)

1.3 概觀 (Overview)

資料(Data)是通過觀測得到的數字性的特徵或數據，以線上購物平台為例，資料包含了賣家所刊登的商品內容資訊，其中有商品的名稱、價錢、介紹、各種款式、商品照片等等的數據、買家的購物車，而將這些數據集合儲存、管理的地方即為「資料庫」，我們利用 **MySQL** 這款開源軟體來實現資料庫的系統。建立完資料庫系統之後，還需要有各種「資訊系統」來提供賣家刊登產品內容，買家的購物車存取。使用者介面我們使用 **Vue.js** 等語言來實現。

1.4符號描述 (Notation Description) (if any)

WSS 1.0.0	The WSS system will be labeled with the number 1.0.0.
LRMS 1.1.n	The LRMS components will be labeled with the number 1.1.n.
AMS 1.2.n	The AMS components will be labeled with the number 1.2.n.
PMS 1.3.n	The PMS components will be labeled with the number 1.3.n.
HOS 1.4.n	The HOS components will be labeled with the number 1.4.n.
DMS 1.5.n	The DMS components will be labeled with the number 1.5.n.
SCS 1.6.n	The SCS components will be labeled with the number 1.6.n.
AOS 1.7.n	The AOS components will be labeled with the number 1.7.n.
DBS 1.8.n	The DBS components will be labeled with the number 1.8.n.

WSS-F-nnn	WSS功能性需求(Functional Requirements)
WSS-N-nnn	WSS非功能性需求(Non-Functional Requirements)
LRMS-F-nnn	LRMS功能性需求(Functional Requirements)
LRMS-N-nnn	LRMS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)
PMS-F-nnn	PMS功能性需求(Functional Requirements)
PMS-N-nnn	PMS非功能性需求(Non-Functional Requirements)
HOS-F-nnn	HOS功能性需求(Functional Requirements)
HOS-N-nnn	HOS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)
SCS-F-nnn	SCS功能性需求(Functional Requirements)
SCS-N-nnn	SCS非功能性需求(Non-Functional Requirements)
AOS-F-nnn	AOS功能性需求(Functional Requirements)
AOS-N-nnn	AOS非功能性需求(Non-Functional Requirements)

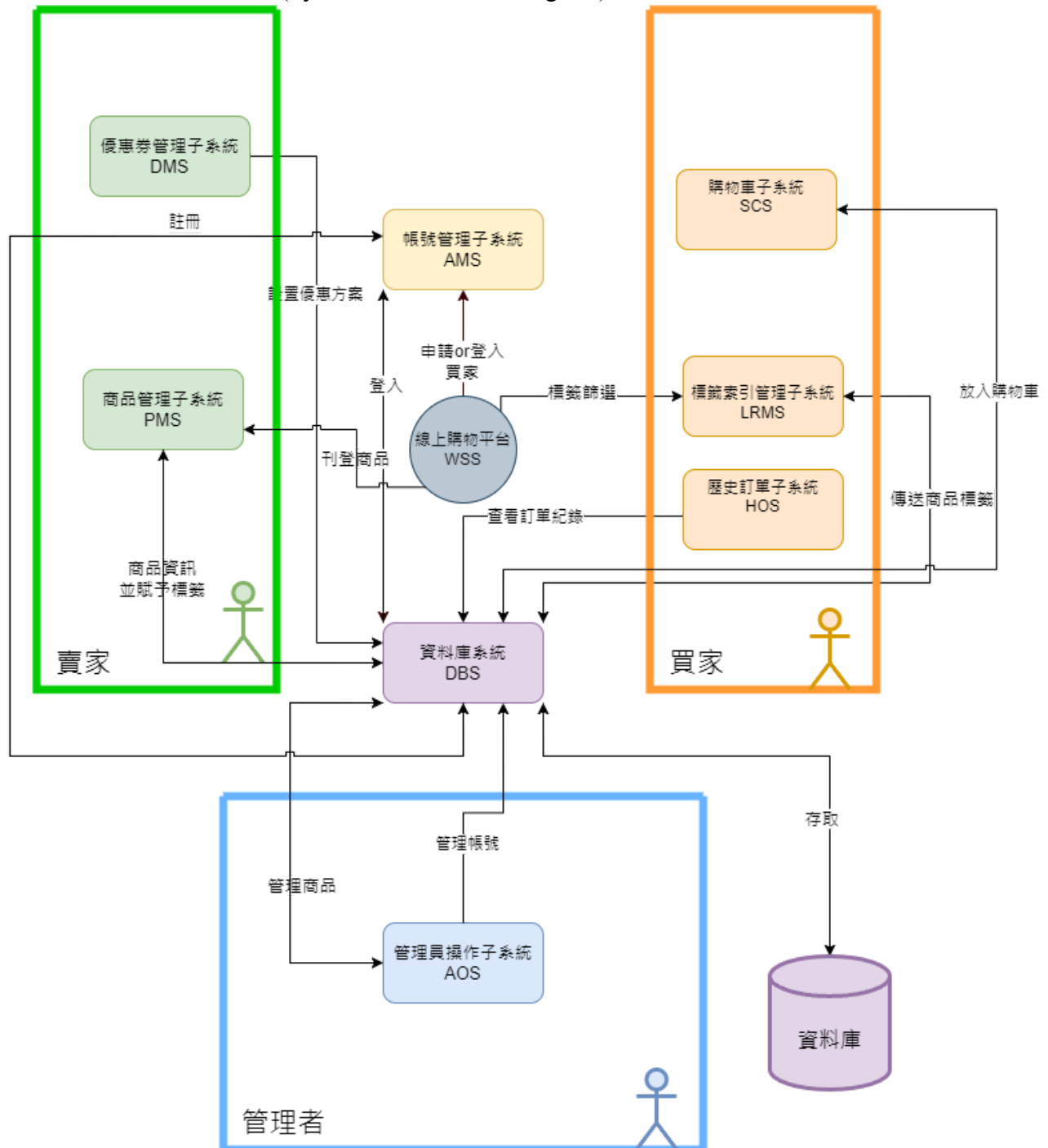
DBS-F-nnn	DBS功能性需求(Functional Requirements)
DBS-N-nnn	DBS非功能性需求(Non-Functional Requirements)

Section 2 系統(System)

2.1系統描述 (System Description)

本系統主要分為8個部分，分別為標籤索引管理子系統(Label Retrieval Management Subsystem,LRMS)、帳號管理子系統(Account Management Subsystem,AMS)、商品管理子系統(Product Management Subsystem,PMS)、歷史訂單子系統(Historical Order Subsystem,HOS)、優惠券管理子系統(Discount Management Subsystem,DMS)、購物車子系統(Shopping Cart Subsystem,SCS)、管理員操作子系統(Admin Operation Subsystem,AOS)、資料庫系統(Database System,DBS)。

2.1.1系統架構圖 (System Architecture Diagram)



2.2 操作概念 (Operational Concepts or User Stories)

Scenario 1. 首頁導覽概念

買家可以透過資訊索引子系統(IRS)輸入關鍵字, 或用標籤管理子系統(LMS)篩選商品條件, 來找到需要的商品等等。賣家則可以用商品發布資訊管理子系統(CPMS)刊登商品、發布商品, 而買家和賣家之間可以用業者與使用者資訊交流子系統(IES)來進行交流。

Scenario 2. 買家操作概念

買家可以透過帳號管理子系統(AMS)進行註冊, 並經由審核資料安全子系統(ADSS)審核後成為會員, 並用資訊索引子系統(IRS)和標籤管理子系統(LMS)來尋找自己想要的商品, 並放入購物車, 等待賣家出貨。

Scenario 3. 賣家操作概念

賣家可以透過帳號管理子系統(AMS)進行註冊, 用商品發布資訊管理子系統(CPMS)發布商品, 使買家搜尋, 也可以對商品進行管理, 編輯商品資訊、商品價格等等, 如果有買家想購買或詢問問題, 便可以用業者與使用者資訊交流子系統(IES)來確認交流。

Scenario 4. 管理員操作概念

管理員能用審核資料安全子系統(ADSS)對想註冊會員的使用者進行資料審核, 確定此使用者是不是一個安全帳戶, 並且有權限操作會員管理子系統(UMS)和案件管理子系統(AMS), 對網站上的使用者或案件進行管理, 以防止違規使用者或不良刊登者, 並可以對資料庫系統(DBS)進行維護。

2.3 功能性需求 (Functional Requirements)

需求編號	需求描述
WHRS-F-001	訪客可註冊會員
WHRS-F-002	訪客與買家可以篩選標籤
WHRS-F-003	管理員可進行帳號審核
WHRS-F-004	管理員可進行商品刪除與審核
WHRS-F-005	賣家可以刊登商品、下架與編輯商品資訊
WHRS-F-006	可以使用聊天室讓買家與賣家進行資訊交換
WHRS-F-007	賣家可以賦予商品內容的相關標籤
WHRS-F-008	買家可以製作購物車
WHRS-F-009	訪客與買家可以使用索引搜尋
WHRS-F-010	可以登入系統

2.4資料需求 (Data Requirements)

需求編號	需求描述
DR-001	帳號資訊
DR-002	商品資訊

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
WHRS-N-001	瀏覽讀取逾時不超過5秒
WHRS-N-002	篩選標籤逾時不超過3秒
WHRS-N-003	資料表設計正規化、節省空間
WHRS-N-004	系統維護時，應跳出通知告知所有使用者

2.5.2資安需求 (Security Requirements) (if any)

需求編號	需求描述
WHRS-N-101	註冊密碼必須符合複雜度需求
WHRS-N-102	應定期維護網站與資訊完整性檢查
WHRS-N-103	管理員應遵守隱私權保護政策，不可隨意檢視個人資料

2.6介面需求 (Interface Requirements)

2.6.1買家介面需求 (User Interfaces Requirements)

需求編號	需求描述
IR-001	搜尋商品介面
IR-002	標籤索引介面
IR-003	製作商品購物車介面

2.6.2賣家介面需求 (Publisher Interface Requirements)

需求編號	需求描述
IR-101	刊登商品資訊介面
IR-102	檢視買家訂單
IR-103	管理已刊登商品介面
IR-104	折扣管理介面
IR-105	檢視銷售報表介面

2.6.3內部介面需求 (Internal Interface Requirements) (if any)

需求編號	需求描述
IR-201	WSS向AMS傳送進行帳號申請或登入需求
IR-202	WSS能向LRMS傳送標籤索引需求
IR-203	WSS能向PMS傳送刊登商品需求
IR-204	PMS能向DBS傳送商品資訊並賦予標籤
IR-205	AMS與DBS需有傳送與接收帳號資訊需求
IR-206	LRMS與DBS需有傳送與接收商品標籤需求
IR-207	DMS與DBS需有傳送與接收優惠資訊需求
IR-208	SCS與DBS需有傳送與接收購物車資訊需求
IR-209	AOS能向DBS傳送與接受管理商品及帳號需求
IR-210	HOS能向DBS傳送查看訂單紀錄需求
IR-211	PMS能向DBS傳送商品資訊並賦予標籤
IR-212	DSB能向資料庫傳送存取需求

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
WHRS-OR-001	需要在有網路的環境
WHRS-OR-002	需要在可以開啟網頁瀏覽器的環境

2.7.2測試需求 (Test Requirements) (if any)

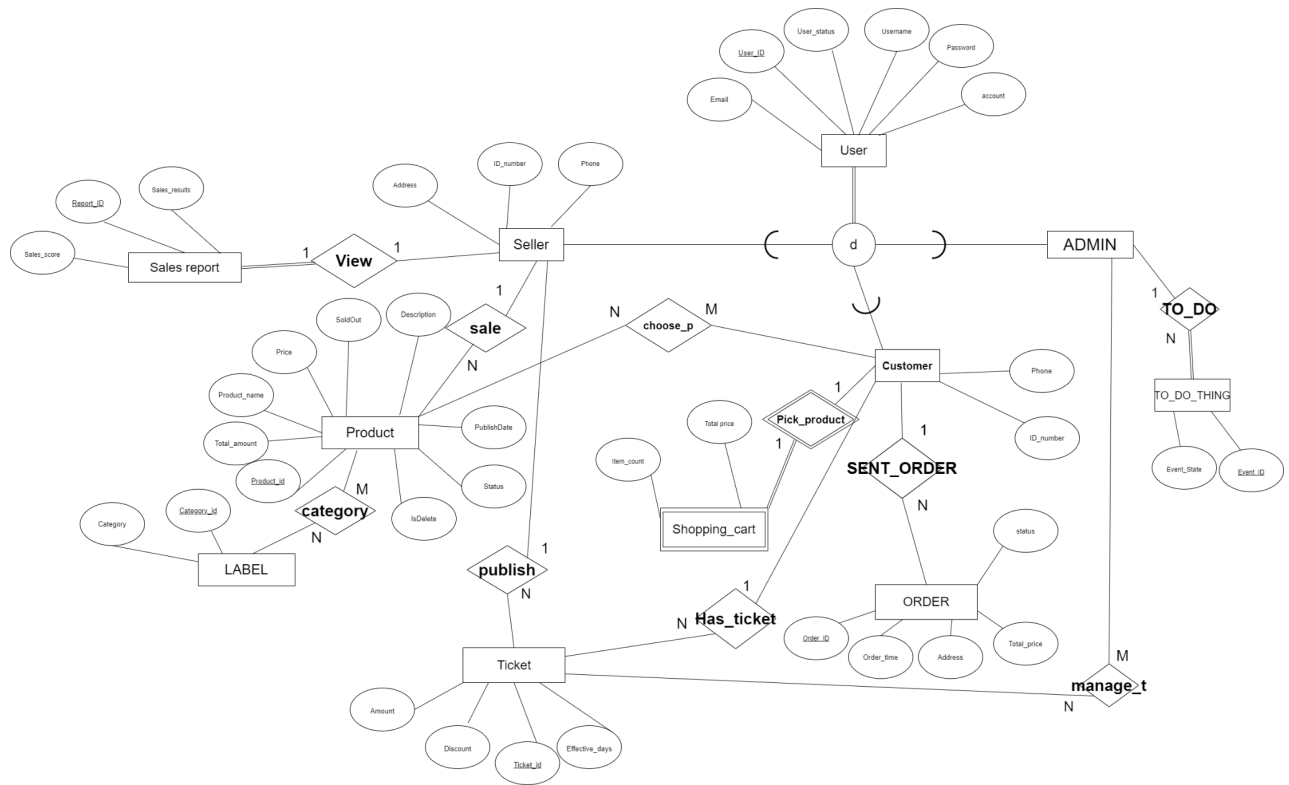
需求編號	需求描述
WHRS-OR-101	每個子系統都應經過測試且無問題
WHRS-OR-102	各系統的平台測試
WHRS-OR-103	各使用者權限需經過測試並且無問題

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

需求編號	需求描述
BBIC-001	不可刊登非法商品
BBIC-002	售價以台幣為單位

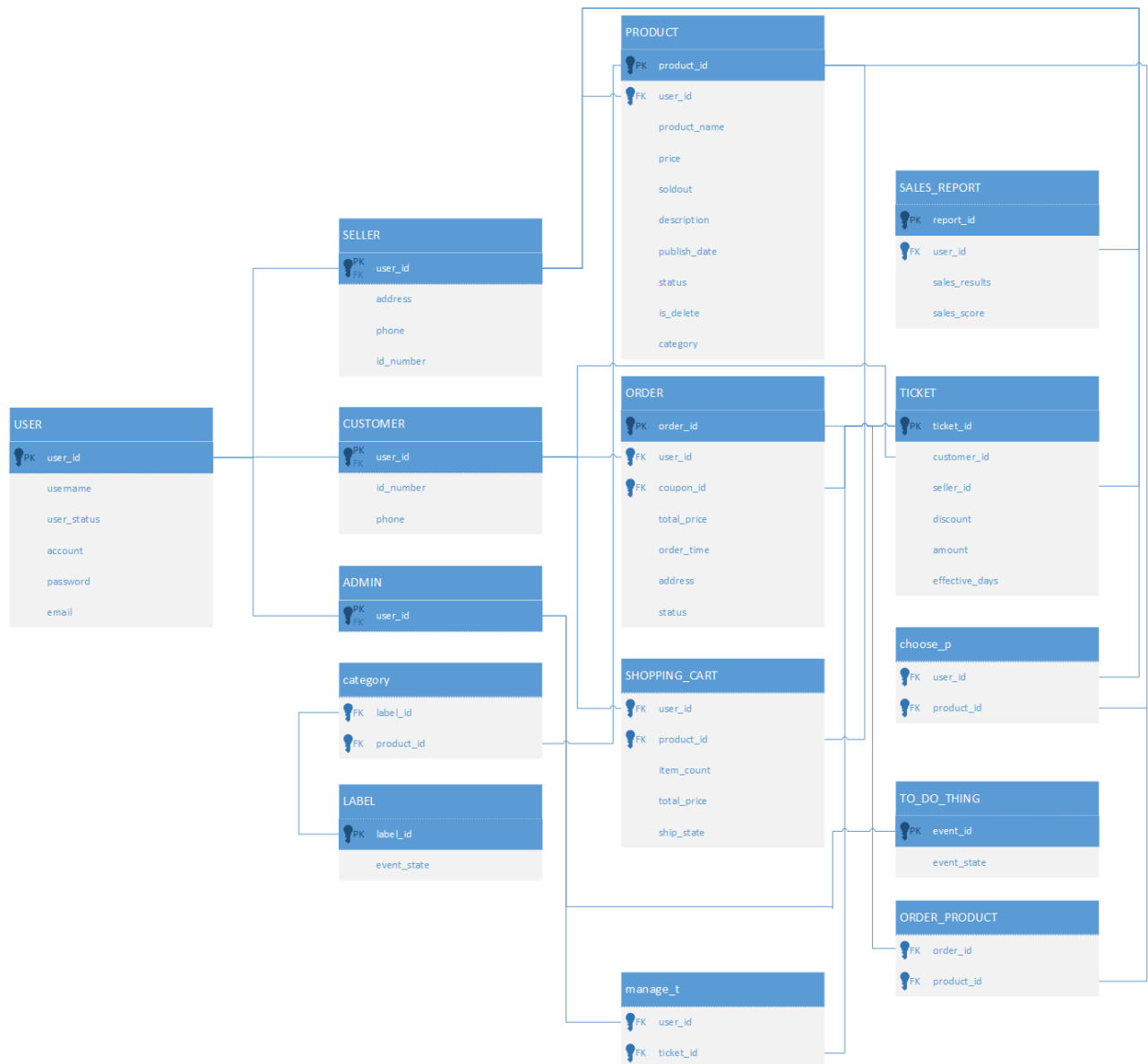
Section 3 資料庫概念設計(Conceptual Design of the Database)

3.1 Entity-Relationship (ER) Model



Section 4 資料庫概念設計(Conceptual Design of the Database)

4.1 Schema of the Database



4.2 Domain of the DataBase

USER				
Description: 存放使用者資訊				
Attribute	Type	Key	Nullable	Description
user_id	varchar	Primary	not null	會員編號(id>0),開頭為1
username	varchar			會員暱稱
account	varchar		not null	會員帳號
user_status	integer		not null	會員帳號狀態0(停用),1(啟用)
password	varchar		not null	會員密碼
email	varchar		not null	會員電子信箱

SELLER				
Description: 存放賣家資訊				
Attribute	Type	Key	Nullable	Description
user_id	varchar	Primary Foreign	not null	會員編號(id>0),開頭為1
address	varchar		not null	賣家地址
phone	varchar		not null	賣家電話
ID_number	varchar		not null	賣家個人身分證

ADMIN				
Description: 存放管理資訊				
Attribute	Type	Key	Nullable	Description
user_id	varchar	Primary Foreign	not null	會員編號(id>0),開頭為1

PRODUCT				
Description: 存放商品資訊				

Attribute	Type	Key	Nullable	Description
product_id	integer	Primary	not null	案件ID
user_id	integer	Foreign	not null	會員編號(id>0),開頭為1
product name	varchar		not null	商品名稱
price	integer		not null	商品價格
soldout	boolean		not null	庫存狀態
description	varchar		not null	商品描述
publishdate	date		not null	上架時間
status	boolean		not null	是否販賣
isDelete	boolean		not null	已刪除
Total_amount	integer		not null	商品總數
category	varchar			商品分類

TICKET				
Description:存放履歷資訊				
Attribute	Type	Key	Nullable	Description
ticket_id	integer	Primary	not null	折價券ID
user_id	integer	Foreign	not null	會員編號(id>0),開頭為1)
discount	integer		not null	折扣
amount	integer		not null	所擁有數量
effective_days	integer		not null	尚可用的時間

CUSTOMER				
Description:存放買家資訊				
Attribute	Type	Key	Nullable	Description
user_id	varchar	Primary Foreign	not null	會員編號(id>0),開頭為1
ID_number	varchar		not null	名字

phone	varchar			買家連絡電話
-------	---------	--	--	--------

TO_DO_THING				
Description: 存放代辦事項資訊				
Attribute	Type	Key	Nullable	Description
event_id	varchar	Primary	not null	事件ID
event	varchar		not null	事件狀態(0:處理完成 1:帳號事件 2:案件事件)

SALES_REPORT				
Description: 存放銷售統計資訊				
Attribute	Type	Key	Nullable	Description
report_id	integer	Primary	not null	報表ID
user_id	integer	Foreign	not null	會員編號(id>0),開頭為1
sales_results	varchar		not null	銷售業績
sales_score	float		not null	評分

CART				
Description: 存放購物車資訊				
Attribute	Type	Key	Nullable	Description
product_id	integer	Foreign	not null	商品編號
user_id	integer	Foreign	not null	使用者ID
item_count	integer		not null	商品數量
total_price	integer		not null	總金額

ORDER				
Description: 存放已購買的訂單資訊				
Attribute	Type	Key	Nullable	Description

order_id	integer	Primary	not null	訂單ID
user_id	integer	Foreign	not null	會員編號(id>0),開頭為1
ticket_id	integer	Foreign	not null	優惠券ID
product_id	integer	Foreign	not null	產品ID
total_price	integer		not null	實際總價格
order_time	date		not null	訂單時間
address	varchar		not null	取件地址
status	boolean		not null	訂單是否完成

LABEL				
Description:存放標籤資訊				
Attribute	Type	Key	Nullable	Description
label_id	integer	Primary	not null	標籤ID
product_id	integer	Foreign	not null	產品ID
label	varchar		not null	標籤

4.3 Possible DB operations frequencies and data volumes

表格	可能操作	頻率 (per day)	資料量	系統負擔
USER	user登入身分驗證	300	500	5000次 Query / day
USER	新增user身分	10	don't care	500次 Insert / day
USER	停用帳號	3	don't care	500次 Update / day
SELLER	讀取商店資料	100	don't care	10000次 Query / day
SELLER	修改商店資料	2	100	50次 Update / day
SELLER	新增賣家資料	30	don't care	500次 Insert / day
SELLER	修改賣家資料	2	100	500次 Update / day
PRODUCT	新增商品資料	10	1000	1000次 Insert / day
PRODUCT	查詢商品	200	10000	60000次 Query/Day
PRODUCT	更新商品資料	5	1000	300次 Update / day
PRODUCT	讀取商品資料	2000	1000	2000次 Query / day
PRODUCT	讀取商品圖片	2000	1000	2000次 Query / day
TICKET	新增優惠券資料	10	100	1000次 Insert / day
TICKET	修改優惠券資料	10	100	1000次 Update / day
TICKET	使用優惠券	100	1000	100次 Query / day
CUSTOMER	新增會員資料(顧客)	50	don't care	500次 Insert / day
CUSTOMER	修改帳號資料(顧客)	2	100	500次 Update / day
CUSTOMER	停用帳號	2	don't care	100次 Update / day
TO_DO_THING	錯誤回報	20	1000	500次 Insert / day
SALES_REPORT	更新銷售業績	80	1000	5000次 Update / day
SALES_REPORT	更新銷售評分	not often	don't care	don't care
CART	更新購物車內容	1000	5000	50000次 Update / day
CART	更新購物車總金額	1000	5000	50000次 Update / day
ORDER	新增訂單資料	800	don't care	50000次 Insert / day

ORDER	編輯訂單資料	300	don't care	50000次 Update / day
LABEL	讀取標籤資料	200	1000	2000次 Query / day
LABEL	新增標籤	not often	don't care	don't care

4.4SQL Statements Used to Construct the Schema

```
CREATE TABLE `world`.`ADMIN` (
  `user_id` int NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `account` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NULL,
  `user_status` bit(1) NULL,
  PRIMARY KEY (`user_id`)
);

CREATE TABLE `world`.`CART_PRODUCT` (
  `product_id` int NOT NULL,
  `user_id_c` int NOT NULL,
  `amount` int NULL,
  CONSTRAINT `fk_CART_PRODUCT_PRODUCT_1` FOREIGN KEY (`product_id`) REFERENCES `world`.`PRODUCT` (`product_id`),
  CONSTRAINT `fk_CART_PRODUCT_CUSTOMER_1` FOREIGN KEY (`user_id_c`) REFERENCES `world`.`CUSTOMER` (`user_id_c`)
);

CREATE TABLE `world`.`CATEGORY` (
  `product_id` int NOT NULL,
  `label_id` int NOT NULL,
  CONSTRAINT `fk_CATEGORY_LABEL_1` FOREIGN KEY (`label_id`) REFERENCES `world`.`LABEL` (`label_id`),
  CONSTRAINT `fk_CATEGORY_PRODUCT_1` FOREIGN KEY (`product_id`) REFERENCES `world`.`PRODUCT` (`product_id`)
);

CREATE TABLE `world`.`CUSTOMER` (
  `user_id_c` int NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `account` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  `phone` varchar(255) NOT NULL,
  `id_number` varchar(10) NOT NULL DEFAULT '',
  `user_status` bit(1) NOT NULL,
  PRIMARY KEY (`user_id_c`),
  UNIQUE INDEX `123` (`account`),
  UNIQUE INDEX `321` (`email`),
  UNIQUE INDEX `2` (`id_number`)
);

CREATE TABLE `world`.`LABEL` (
  `label_id` int NOT NULL AUTO_INCREMENT,
  `label` varchar(255) NOT NULL,
  PRIMARY KEY (`label_id`)
);

CREATE TABLE `world`.`manage_t` (
  `user_id` int NOT NULL,
  `ticket_id` int NOT NULL,
  CONSTRAINT `fk_manage_t_ADMIN_1` FOREIGN KEY (`user_id`) REFERENCES `world`.`ADMIN` (`user_id`),
  CONSTRAINT `fk_manage_t_TICKET_1` FOREIGN KEY (`ticket_id`) REFERENCES `world`.`TICKET` (`ticket_id`)
);

CREATE TABLE `world`.`ORDER` (
  `order_id` int NOT NULL AUTO_INCREMENT,
  `total_price` float(10, 2) NOT NULL,
  `order_time` datetime NOT NULL,
  `address` varchar(255) NOT NULL,
  `status` int NOT NULL,
  `user_id_c` int NOT NULL,
  `ticket_id` int NULL,
  PRIMARY KEY (`order_id`),
  CONSTRAINT `fk_ORDER_CUSTOMER_1` FOREIGN KEY (`user_id_c`) REFERENCES `world`.`CUSTOMER` (`user_id_c`),
  CONSTRAINT `fk_ORDER_TICKET_1` FOREIGN KEY (`ticket_id`) REFERENCES `world`.`TICKET` (`ticket_id`)
);

CREATE TABLE `world`.`ORDER_PRODUCT` (
  `order_id` int NOT NULL,
  `product_id` int NOT NULL,
  `amount` int NULL,
  CONSTRAINT `fk_ORDER_PRODUCT_ORDER_1` FOREIGN KEY (`order_id`) REFERENCES `world`.`ORDER` (`order_id`),
  CONSTRAINT `fk_ORDER_PRODUCT_PRODUCT_1` FOREIGN KEY (`product_id`) REFERENCES `world`.`PRODUCT` (`product_id`)
);
```

```
CREATE TABLE `world`.`PRODUCT` (
  `user_id_s` int NOT NULL,
  `product_id` int NOT NULL,
  `product_name` varchar(255) NOT NULL,
  `price` int NOT NULL,
  `description` varchar(255) NULL,
  `publish_date` datetime NOT NULL,
  `status` bit(1) NOT NULL,
  `total_amount` int NOT NULL,
  PRIMARY KEY (`product_id`),
  CONSTRAINT `fk_PRODUCT_SELLER_1` FOREIGN KEY (`user_id_s`) REFERENCES `world`.`SELLER` (`user_id_s`)
);
```

```
CREATE TABLE `world`.`SALES_REPORT` (
  `report_id` int NOT NULL AUTO_INCREMENT,
  `sales_results` varchar(255) NOT NULL,
  `sales_score` int NOT NULL,
  `user_id_s` int NOT NULL,
  PRIMARY KEY (`report_id`),
  CONSTRAINT `fk_SALES_REPORT_SELLER_1` FOREIGN KEY (`user_id_s`) REFERENCES `world`.`SELLER` (`user_id_s`)
);
```

```
CREATE TABLE `world`.`SELLER` (
  `user_id_s` int NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `account` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  `phone` varchar(255) NOT NULL,
  `id_number` varchar(10) NOT NULL,
  `user_status` bit(1) NOT NULL,
  PRIMARY KEY (`user_id_s`)
);
```

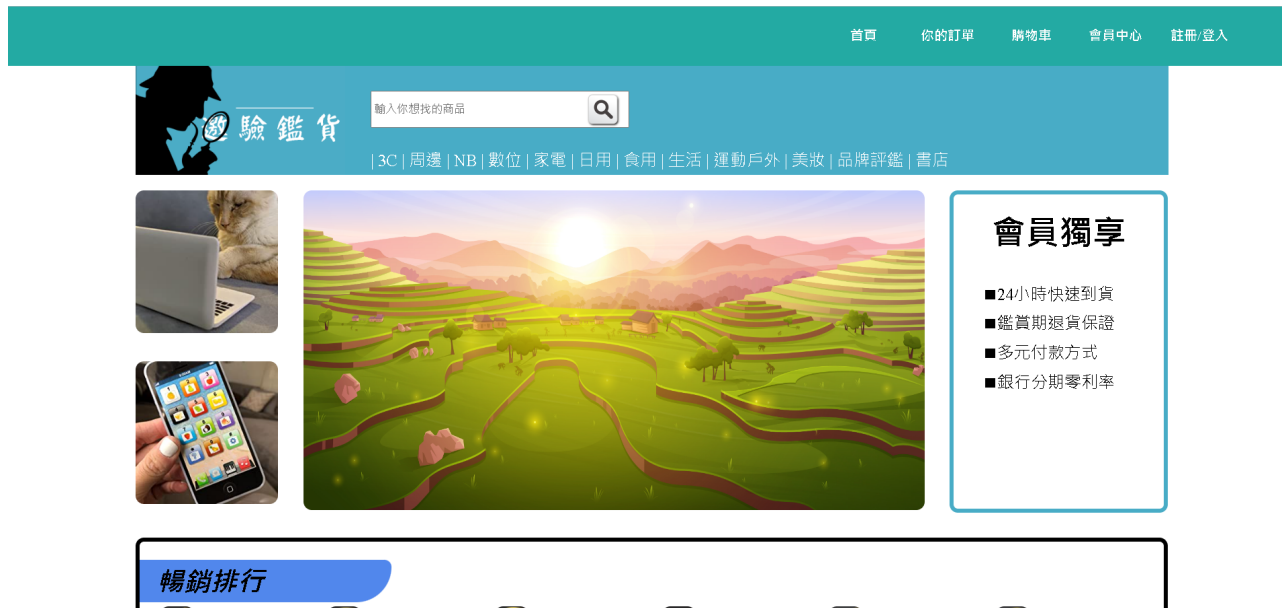
```
CREATE TABLE `world`.`TICKET` (
  `ticket_id` int NOT NULL AUTO_INCREMENT,
  `effective_date` datetime NOT NULL,
  `amount` varchar(255) NOT NULL,
  `discount` varchar(255) NOT NULL,
  `user_id_s` int NOT NULL,
  PRIMARY KEY (`ticket_id`),
  CONSTRAINT `fk_TICKET_SELLER_1` FOREIGN KEY (`user_id_s`) REFERENCES `world`.`SELLER` (`user_id_s`)
);
```

```
CREATE TABLE `world`.`TO_DO_THING` (
  `event_id` int NOT NULL AUTO_INCREMENT,
  `event_content` varchar(255) NULL,
  `event_state` bit(1) NOT NULL,
  `user_id` int NOT NULL,
  PRIMARY KEY (`event_id`),
  CONSTRAINT `fk_TO_DO_THING_ADMIN_1` FOREIGN KEY (`user_id`) REFERENCES `world`.`ADMIN` (`user_id`)
);
```

Section 5 功能性依賴(Functional Dependencies and Database Normalization)

Section 6 資料庫概念設計(Additional Queries and Views)

6.1 首頁畫面



6.2 登入畫面



Appendix