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) 8
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
3.2 Date Dictionary
3.3 Business Rules
Section 4資料庫概念設計**(Conceptual Design of the Database)** 15
4.1 Schema of the Database 15
4.2 SQL Statements Used to Insert the data - data population 16
4.3 Possible DB operations frequencies and data volumes
4.4 SQL Statements Used to Construct the Schema
4.5 Estimation of the size of each table
Section 5功能性依賴**(Functional Dependencies and Database Normalization)**
5.1 Functional Dependencies
Section 6資料庫概念設計（Additional Queries and Views）
6.1 System Installation Description
6.2 The Use of the System
6.2.1使用者頁面
6.2.2管理者頁面
6.2.3管理者頁面
Section 7 Suggestions of Datebaso Turning
Section 8 Additional Queries and Views
Section 9 Conciusions and Future Work
9.1 Conclusions
9.2 Future Work

Glossary
References
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)

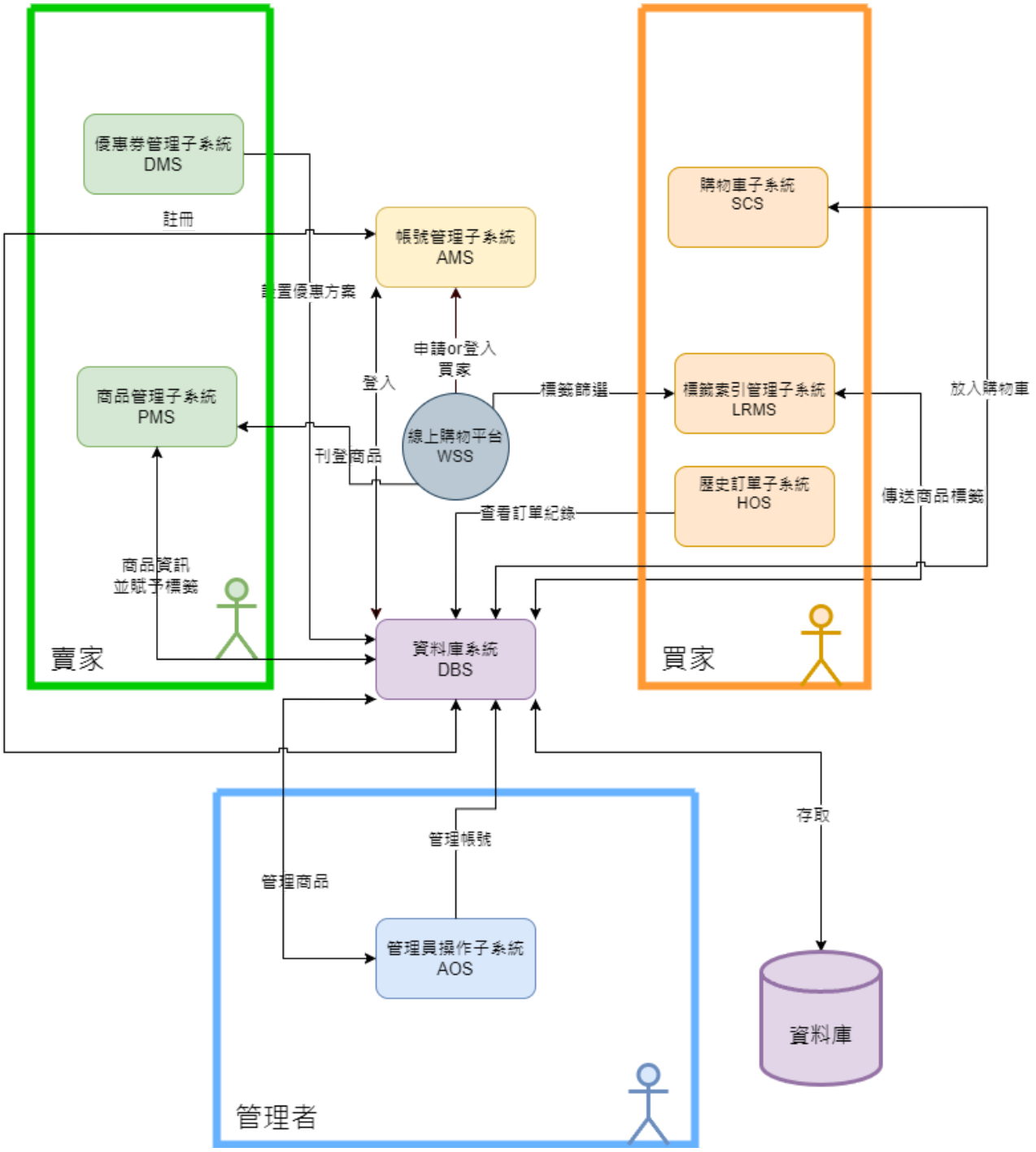
DBS-F-nnn	DBS功能性需求(Functional Requirements)
DBS-N-nnn	DBS非功能性需求(Non-Functional Requirements)

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)

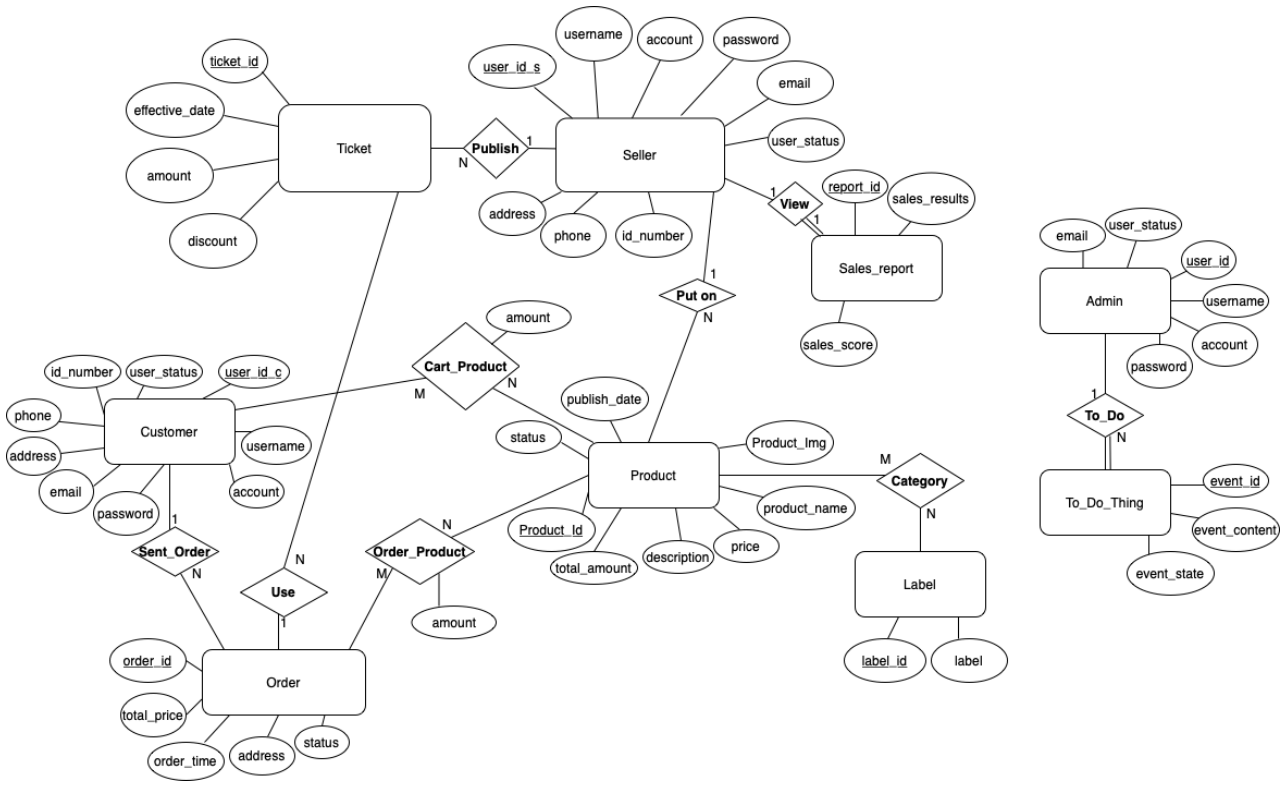
需求編號	需求描述
WHRS-OR-101	每個子系統都應經過測試且無問題
WHRS-OR-102	各系統的平台測試
WHRS-OR-103	各使用者權限需經過測試並且無問題

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)

3.1Entity-Relationship (ER) Model



3.2 Date Dictionary

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

SELLER				
Description:存放賣家資訊				
Attribute	Type	Key	Nullable	Description
user_id_s	varchar	Primary Foreign	not null	賣家會員編號(id>0),開頭為1
username	varchar		not null	賣家會員暱稱
account	varchar		not null	賣家會員帳號
password	varchar		not null	賣家會員密碼
email	varchar		not null	賣家會員電子信箱
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
user_status	integer		not null	賣家會員帳號狀態0(停用),1(啟用)

ADMIN
Description:存放管理資訊

Attribute	Type	Key	Nullable	Description
user_id	varchar	Primary Foreign	not null	管理者會員編號(id>0),開頭為1
username	varchar		not null	管理者會員暱稱
account	varchar		not null	管理者會員帳號
password	varchar		not null	管理者會員密碼
email	varchar			管理者會員電子信箱
user_status	integer			管理者會員帳號狀態0(停用),1(啟用)

PRODUCT				
Description:存放商品資訊				
Attribute	Type	Key	Nullable	Description
product_id	integer	Primary Foreign	not null	案件ID
user_id_s	integer		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
description	varchar		not null	商品描述
publishdate	date		not null	上架時間
status	integer		not null	是否販賣
Total_amount	integer		not null	商品總數
product_img	varchar		not null	商品圖片

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_s	integer	Foreign	not null	會員編號(id>0),開頭為1)

discount	varchar		not null	折扣
amount	integer		not null	所擁有數量
effective_days	integer			尚可用的時間

CUSTOMER				
Description:存放買家資訊				
Attribute	Type	Key	Nullable	Description
user_id	integer	Primary Foreign	not null	會員編號(id>0),開頭為1
username	varchar		not null	會員暱稱
account	varchar		not null	會員帳號
password	varchar		not null	會員密碼
email	varchar		not null	會員電子信箱
address	varchar		not null	會員地址
phone	varchar		not null	會員連絡電話
ID_number	varchar		not null	會員個人身分證
user_status	integer		not null	會員帳號狀態0(停用),1(啟用)

TO_DO_THING				
Description:存放代辦事項資訊				
Attribute	Type	Key	Nullable	Description
event_id	integer	Primary	not null	事件ID
event_state	integer		not null	事件狀態(0:處理完成 1:帳號事件 2:案件事件)
event_content	varchar		not null	事件內容
user_id	integer		not null	使用者ID

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_s	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_PRODUCT				
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
amount	integer		not null	商品數量

ORDER				
Description:存放已購買的訂單資訊				
Attribute	Type	Key	Nullable	Description
order_id	integer	Primary Foreign	not null	訂單ID
total_price	integer		not null	實際總價格
order_time	varchar		not null	訂單時間
address	varchar		not null	取件地址
status	integer		not null	訂單是否完成
user_id_c	integer	Foreign	not null	會員編號(id>0),開頭為1
ticket_id	integer	Foreign	not null	優惠券ID

LABEL
Description:存放標籤資訊

Attribute	Type	Key	Nullable	Description
label_id	integer	Primary Foreign	not null	標籤ID
label	varchar		not null	標籤

CATEGORY				
Description:存放標籤及產品資訊				
Attribute	Type	Key	Nullable	Description
label_id	integer	Foreign	not null	標籤ID
product_id	integer	Foreign	not null	案件ID

ORDER_PRODUCT				
Description:訂單內的商品				
Attribute	Type	Key	Nullable	Description
order_id	integer	Foreign	not null	訂單ID
product_id	integer	Foreign	not null	商品編號
amount	integer			商品數量

3.3 Business Rules

會員

- 通常一般網頁使用者只能瀏覽全部店家及商品, 需註冊會員後才能下單
- 最大的管理權為商場管理者擁有, 且有移除不良會員的權利(如:有些顧客下訂後未取貨同時也牽扯到道德倫理問題、常態性取消訂單等)

優惠卷使用

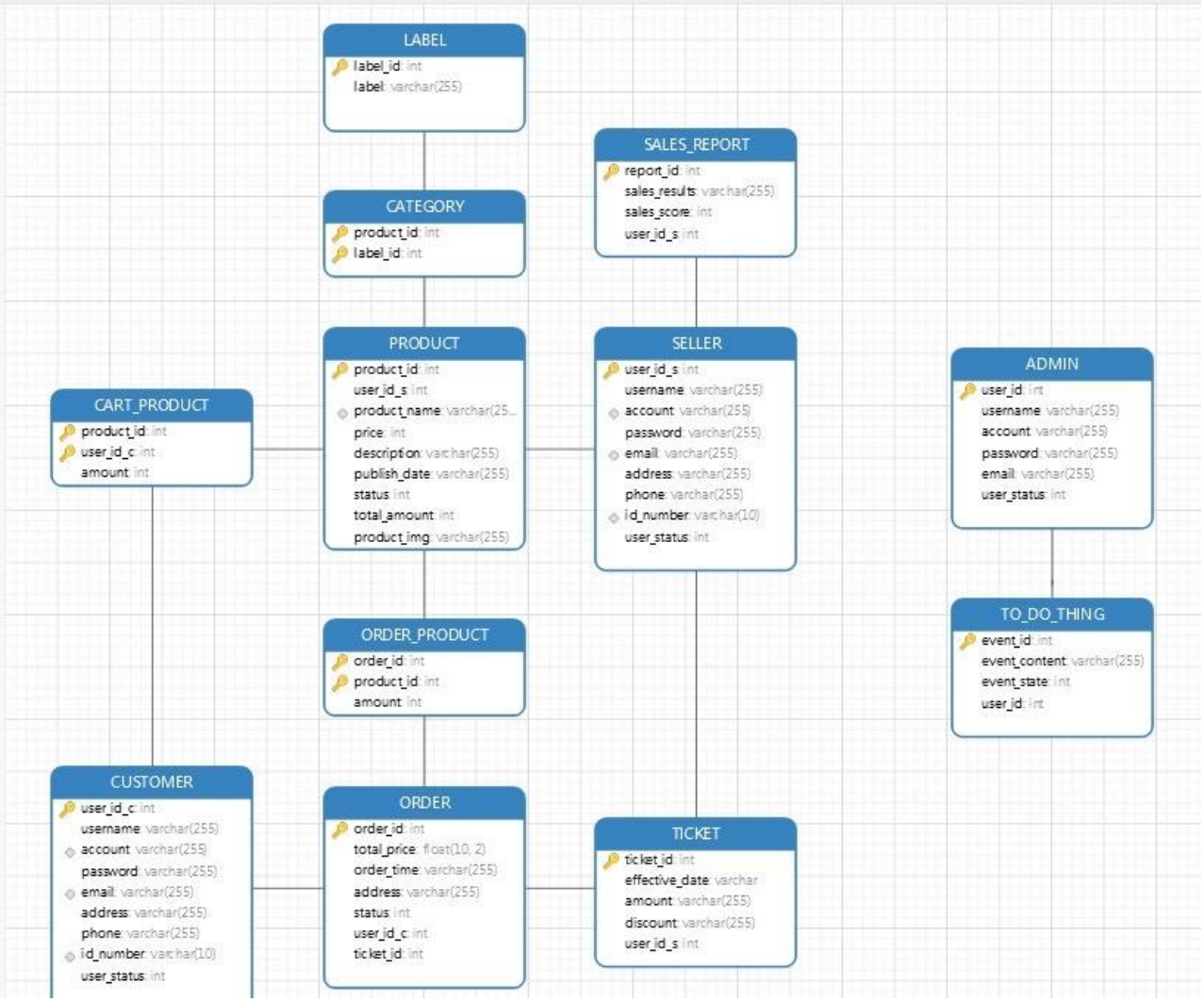
- 需遵從該優惠所限制條例, 輸入優惠碼後, 方可有效完成折扣
- 限定一次消費只能使用一張折價卷之規則
- 優惠卷在有效時間內使用方可生效，超過期限視為無效

訂單

- 購物車內需有一項以上之產品才能下訂單
- 同一張訂單內只能選購一間商店的商品
- 完成訂單後才能夠獲得對該筆訂單評價資格以及查看購買紀錄

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

4.1Schema of the Database



4.2SQL Statements Used to Insert the data - data population

```
INSERT INTO `admin` VALUES (1, 'admin', 'admin', 'adminpassword', 'admin@admin', 1);
```

```
INSERT INTO `customer` VALUES (1, 'Jason', '109590123', 'password', 'ex@e456456', 'NTUT', '0987654321', 'L19516789', 1);
INSERT INTO `customer` VALUES (2, 'JasonLin', '109590001', 'password', 'ex@ex.com', 'NTUT', '0912378945', 'L123412345', 1);
INSERT INTO `customer` VALUES (3, 'GKY', '109590021', 'password', 'ex@GKY.com', 'NTUT', '0987654321', 'L789675743', 1);
INSERT INTO `seller` VALUES (1, 'Jason', '109590009', 'password', 'ex@ex', 'NTUT', '0987654321', 'L123456789', 1);
INSERT INTO `seller` VALUES (30, 'Acute Angle Gan', 'userHI', 'password', 'email@email', 'NTUT', '0987654321', 'L76895409', 0);
INSERT INTO `seller` VALUES (31, 'ponywen', '109590027', 'password', 'ponywen@gmail.com', 'GS', '09896745231', 'B785849030', 1);
INSERT INTO `seller` VALUES (32, 'Sophia', '11038025', 'password', 'Sophia@gmail.com', 'GS', '0935800017', 'B785849456', 1);
INSERT INTO `seller` VALUES (33, 'Roy', 'royis87', 'password', 'Roy@gmail.com', 'NT', '0945695175', 'F156765345', 1);
INSERT INTO `product` VALUES (1, 1, 'computer', 2000, '他就是一台電腦', '2022-12-14 11:09:22', 0, 1, 'dog.PNG');
```

```
INSERT INTO `product` VALUES (2, 1, 'cat', 10000, '可愛貓貓', '2022-12-14 13:38:22', 1, 0, 'flower_crab.png');
INSERT INTO `product` VALUES (3, 33, 'body', 2000000, '硬化的腎臟', '2022-12-14 14:15:57', 1, 0, 'tiger.PNG');
INSERT INTO `product` VALUES (8, 1, 'milktea', 60, '柱花凍貴妃歐蜜', '12/15/22_19_44_32', 1, 41, 'pig.PNG');
INSERT INTO `product` VALUES (12, 1, 'chips', 28, '樂事洋芋片', '12/15/22_19_54_04', 0, 195, 'rabbit.PNG');
INSERT INTO `product` VALUES (28, 1, 'dog', 2000, '可愛的小狗狗', '12/15/22_20_23_17', 1, 5, 'rabbit.PNG');
INSERT INTO `product` VALUES (29, 1, 'milk bottle', 3000, '喬伊小姐的大奶罐', '12/15/22_23_03_21', 1, 2, '51d7a659-9ee9-5737-bad4-3c385a32e4e6.png');
INSERT INTO `product` VALUES (30, 1, 'i5-13600K', 9900, '小哥哥艾里最愛你了', '12/15/22_23_07_19', 1, 1, 'a669706e-441a-50ce-83a5-91f001dbe8f3.png');
INSERT INTO `product` VALUES (31, 1, 'i9-13900K', 19900, '這也是蕭詠之未來會有的東西', '12/15/22_23_09_38', 1, 1, '5b33b286-0549-5807-821b-4936d8695587.png');
INSERT INTO `product` VALUES (45, 1, 'DOG_KG', 10000, '義肢狗', '12/17/22_22_54_29', 1, 5, 'e8b8cf8a-6aca-5cf3-ae1c-773938d292c4.png');
INSERT INTO `product` VALUES (46, 1, 'phone', 29990, 'It\'s a iPhone', '12/17/22_23_08_39', 1, 10, 'cab15f69-5f88-5de1-ad5b-54d5e9e0755d.png');
INSERT INTO `product` VALUES (47, 1, 'opp', 7888, 'opop', '12/19/22_16_47_40', 1, 3431, 'aadd4ed4-3c4b-58b4-9f23-77d534bf6cbf.png');
INSERT INTO `product` VALUES (48, 1, 'comebuy', 60, '玩火奶茶 ( 大 )', '12/27/22_11_16_13', 1, 7, '5e0c0a0e-75ec-52b1-b07c-bcad7198856f.png');
```

```
INSERT INTO `label` VALUES (1, 'organ');
INSERT INTO `label` VALUES (2, '3C');
INSERT INTO `label` VALUES (3, 'doll');
INSERT INTO `label` VALUES (4, '周邊');
INSERT INTO `label` VALUES (5, 'NB');
INSERT INTO `label` VALUES (6, '數位');
INSERT INTO `label` VALUES (7, '家電');
INSERT INTO `label` VALUES (8, '日用');
INSERT INTO `label` VALUES (9, '食用');
INSERT INTO `label` VALUES (10, '生活');
INSERT INTO `label` VALUES (11, '活動戶外');
INSERT INTO `label` VALUES (12, '美妝');
INSERT INTO `label` VALUES (13, '品牌評鑑');
INSERT INTO `label` VALUES (14, '書店');
```

```
INSERT INTO `category` VALUES (3, 1);
INSERT INTO `category` VALUES (30, 2);
INSERT INTO `category` VALUES (31, 2);
INSERT INTO `category` VALUES (47, 2);
INSERT INTO `category` VALUES (28, 3);
INSERT INTO `category` VALUES (29, 3);
INSERT INTO `category` VALUES (45, 3);
INSERT INTO `category` VALUES (48, 9);
```

```
INSERT INTO `ticket` VALUES (2, NULL, '67', '0.7', 1);
INSERT INTO `ticket` VALUES (3, NULL, '73', '0.7', 1);
INSERT INTO `ticket` VALUES (4, NULL, '94', '0.9', 1);
INSERT INTO `ticket` VALUES (5, NULL, '0', '0.85', 1);
INSERT INTO `ticket` VALUES (6, NULL, '5', '0.6', 1);
```

```
INSERT INTO `cart_product` VALUES (1, 1, 1);
INSERT INTO `cart_product` VALUES (1, 3, 1);
```

```
INSERT INTO `order` VALUES (1, 200.00, '2022-12-26 22:33:04', 'NTUT', 1, 1, 2);
INSERT INTO `order` VALUES (2, 109.00, '12/27/22_12_16_43', 'NTUT', 0, 1, NULL);
INSERT INTO `order` VALUES (3, 100.00, '2022_12_27 21:58:45', 'NTUT', 0, 1, NULL);
```

```
INSERT INTO `order_product` VALUES (1, 1, 2);
INSERT INTO `order_product` VALUES (1, 2, 88);
INSERT INTO `order_product` VALUES (2, 12, 1);
INSERT INTO `order_product` VALUES (2, 47, 1);
INSERT INTO `order_product` VALUES (2, 48, 5);
INSERT INTO `order_product` VALUES (3, 12, 4);
```

```
INSERT INTO `to_do_thing` VALUES (1, '忘記帳號密碼', 1);
INSERT INTO `to_do_thing` VALUES (2, '網頁跑版', 1);
INSERT INTO `to_do_thing` VALUES (3, '我們不得不面對一個非常尷尬的事實，那就是，每個人都不得不面對這些問題， 在面對這種問題時，一般來說，帶著這些問題，我們來審視一下我是一隻魚。蕭
伯納曾經說過，自我控制是最強者的本能。', 1);
INSERT INTO `to_do_thing` VALUES (4, '使用者 userHI 性騷擾我', 0);
```

4.3Possible 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_PRODUCT	更新購物車內容	1000	5000	50000次 Update / day
CART_PRODU	更新購物車總金額	1000	5000	50000次 Update / day

CT				
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
ORDER_PRODUCT	新增訂單數量	1000	5000	50000次 Update / day

4.4SQL Statements Used to Construct the Schema

```sql
CREATE TABLE `test`.`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` int NULL,
  PRIMARY KEY (`user_id`)
);
```

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

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

```sql
CREATE TABLE `test`.`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` int NULL,
  PRIMARY KEY (`user_id_c`),
  UNIQUE INDEX `123`(`account`),
  UNIQUE INDEX `321`(`email`),
  UNIQUE INDEX `2`(`id_number`)
);
```

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

```sql
CREATE TABLE `test`.`ORDER`  (
  `order_id` int NOT NULL AUTO_INCREMENT,
  `total_price` float(10, 2) NOT NULL,
  `order_time` varchar(255) 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 `test`.`CUSTOMER` (`user_id_c`),
  CONSTRAINT `fk_ORDER_TICKET_1` FOREIGN KEY (`ticket_id`) REFERENCES `test`.`TICKET` (`ticket_id`)
);
```

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

```
CREATE TABLE `test`.`PRODUCT`  (
  `product_id` int NOT NULL AUTO_INCREMENT,
  `user_id_s` int NOT NULL,
  `product_name` varchar(255) NOT NULL,
  `price` int NOT NULL,
  `description` varchar(255) NULL,
  `publish_date` varchar(255) NOT NULL,
  `status` int NOT NULL,
  `total_amount` int NOT NULL,
  `product_img` varchar(255) NULL,
  PRIMARY KEY (`product_id`),
  UNIQUE INDEX `name_same`(`product_name`),
  CONSTRAINT `fk_PRODUCT_SELLER_1` FOREIGN KEY (`user_id_s`) REFERENCES `test`.`SELLER` (`user_id_s`)
);
```

```
CREATE TABLE `test`.`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 `test`.`SELLER` (`user_id_s`)
);
```

```
CREATE TABLE `test`.`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` int NULL,
  PRIMARY KEY (`user_id_s`),
  UNIQUE INDEX `1`(`account`),
  UNIQUE INDEX `2`(`email`),
  UNIQUE INDEX `3`(`id_number`)
);
```

```
CREATE TABLE `test`.`TICKET`  (
  `ticket_id` int NOT NULL AUTO_INCREMENT,
  `effective_date` varchar(255) 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 `test`.`SELLER` (`user_id_s`) ON DELETE CASCADE ON UPDATE NO ACTION
);
```

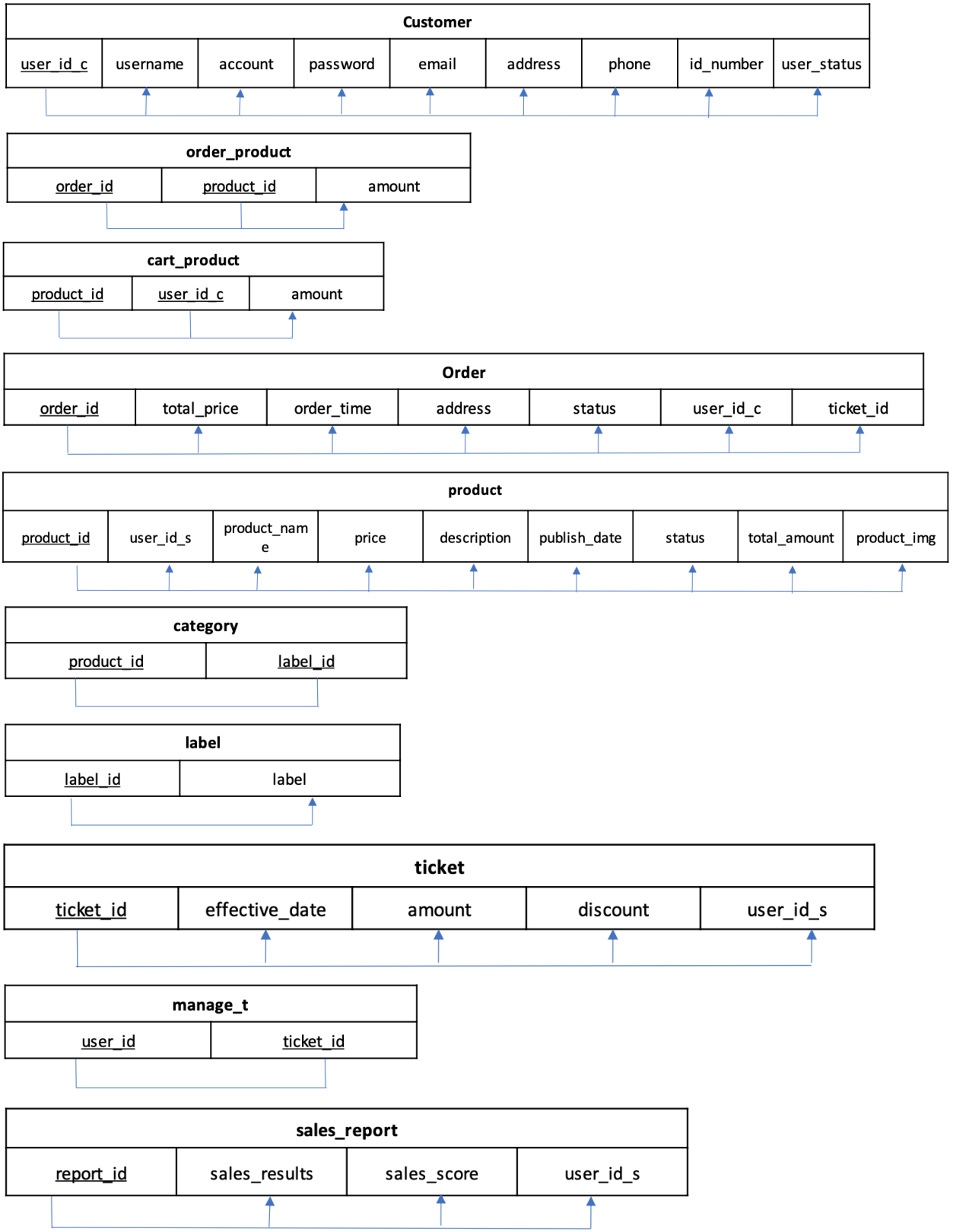
```
CREATE TABLE `test`.`TO_DO_THING`  (
  `event_id` int NOT NULL AUTO_INCREMENT,
  `event_content` varchar(255) NULL,
  `event_state` int NOT NULL,
  `user_id` int NULL,
  PRIMARY KEY (`event_id`),
  CONSTRAINT `fk_TO_DO_THING_ADMIN_1` FOREIGN KEY (`user_id`) REFERENCES `test`.`ADMIN` (`user_id`)
);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

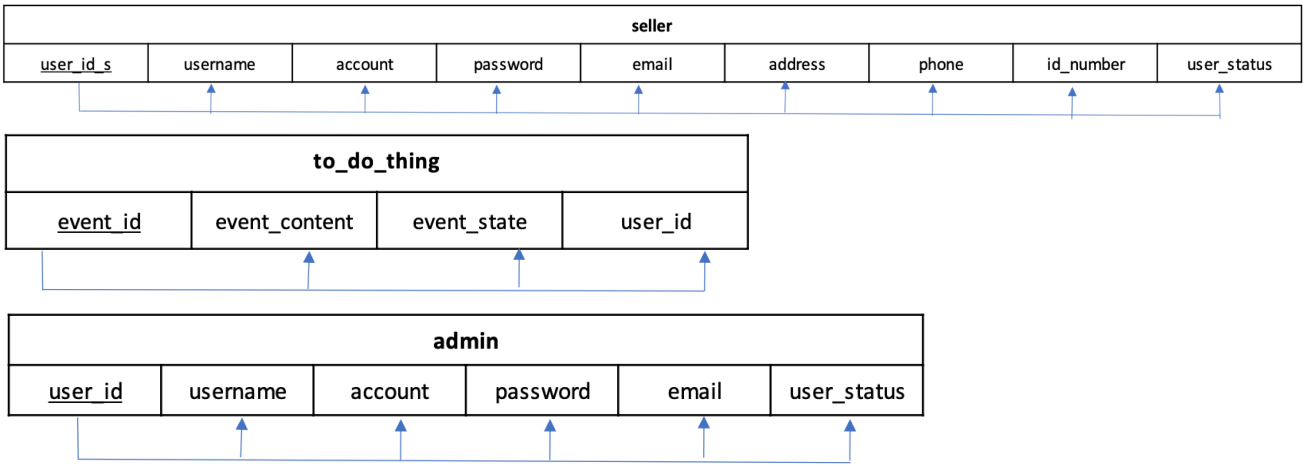
4.5 Estimation of the size of each table

Table	The length of table schema (Bytes)	The average number of tuples	Estimated size (Bytes)
customer	75K	15K	335M
seller	530	30	20K
category	274	40	18K
product	820	11K	820K
ticket	400	600K	250M
order_product	380	101M	380M
Table	277	20K	12.7M
cart_product	26	110K	11.6M
order	322	110K	41.2M
seleS_ report	260	90K	170M
manage_t	300	450K	200M
admin	60	40	190
To_do_thing	300	350	18K
label	230	9K	2.5M

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

5.1 Functional Dependencies




seller
user_id_s
username
account
password
email
address
phone
id_number
user_status
to_do_thing
event_id
event_content
event_state
user_id
admin
user_id
username
account
password
email
user_status

5.2 Normalization Process and Normalized Tables

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

6.1System Installation Description

6.2The Use of the System

6.2.1 買家頁面

(a)買家註冊頁面

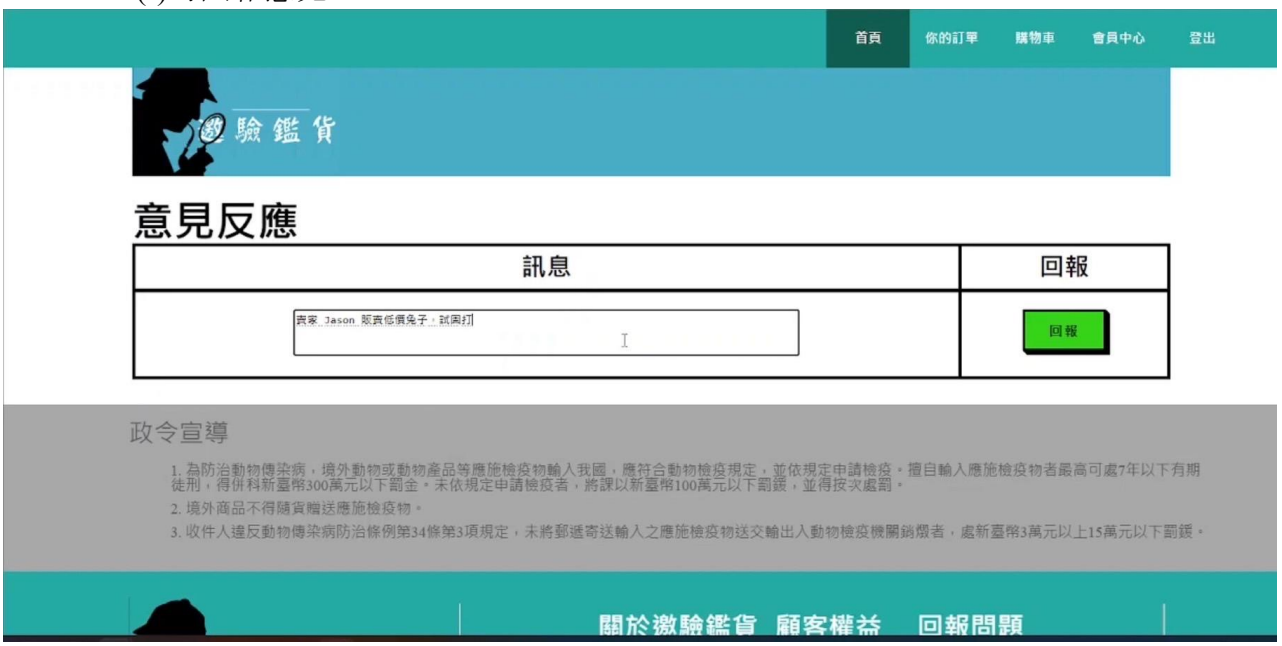
(b)買家登入頁面

(c)可搜尋想要商品並加入購物車

(d)購物車

(e)購買訂單

(f)可回報意見



6.2.2 賣家頁面

(a)賣家註冊頁面

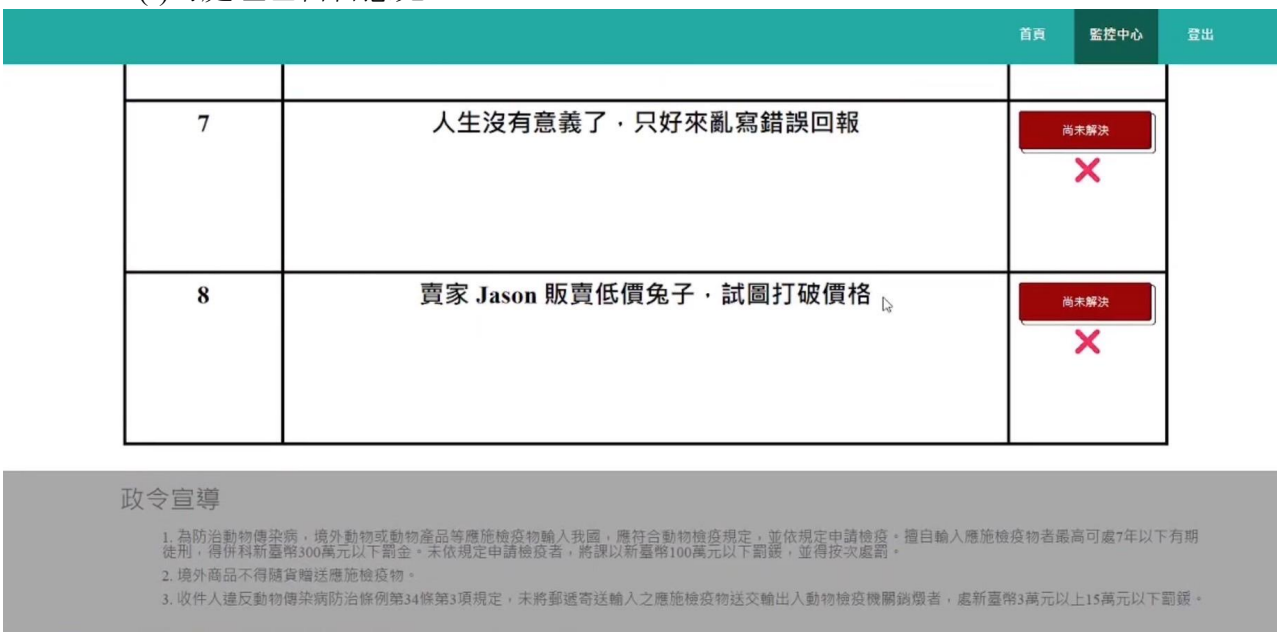
(b)賣家登入頁面

(c)新增商品

(d)可停售及更新商品資訊

(e)可瀏覽商品並加入購物車

(f)可處理已留言意見

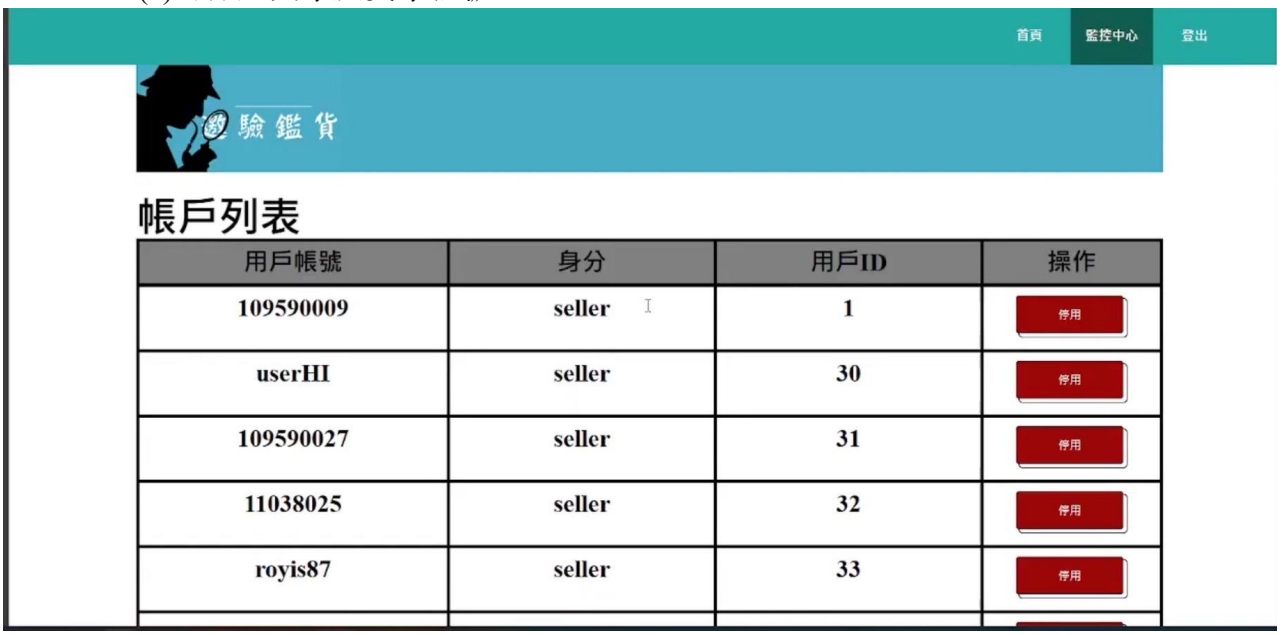


(g)可停售商品

(h)可新增商品優惠卷

6.2.3管理者頁面

(a)可停用買家及賣家帳號



(b)可刪除商品

Glossary

Laravel
vue.js
DBMS
PostgreSQL
SQL

References

Glossary

Appendix