

Online Shopping System

第 12 組：相見恨晚

	系級	學號	姓名
組長	資科三	108703019	賴冠瑜
組員	資科三	108703021	張誠宗
組員	資科三	108703030	田詠恩
組員	資科三	108703034	蔡博超
組員	資科三	108703037	林宏睿
組員	資科三	108703043	倪靖揚
組員	資科三	108703050	吳尚恩

● 任務分工

姓名	分工	貢獻百分比
賴冠瑜	PM 前後端 ER Model、系統架構製作 書面報告製作、報告	1/7
張誠宗	前端製作 ER Model	1/7
田詠恩	後端 API 製作資料庫、Relational Model 投影片製作	1/7
蔡博超	前端負責人 前端製作、ER Model、報告	1/7
林宏睿	前端製作、ER Model 報告整理	1/7
倪靖揚	製造資料庫、後端製作 需求分析製作	1/7
吳尚恩	前端製作、ER Model 系統功能製作	1/7

● 需求分析(系統功能需求分析)

買家：

1. 可以在賣場搜尋商品種類，商家名稱，並把商品加入購物車
2. 可從購物車刪除已選擇的商品
3. 在購物車裡可以選擇商品數量並購買
4. 購買後的商品可在歷史紀錄裡查看

賣家：

1. 可訂商品到指定倉庫
2. 可從倉庫選擇商品上架販賣
3. 可確認倉庫和架上貨品狀態
4. 可查看出售紀錄
5. 可以查看時間內營業額

● 需求分析(資料需求分析)

Customer API:

UserExist (Read):

確認註冊的帳號是否已經存在

Register (Create):

輸入使用者資料及設定密碼並註冊帳號

Search_Product (Read):

前端傳入ShopID,產品種類和頁數，後端會傳出所有此種產品的所有資訊。

Get_All_Type (Read):

後端會傳回所有type放入選單。

Get_All_ShopName (Read):

後端會傳回所有ShopName放入選單。

Get_Product_MaxPage (Read):

前端輸入ShopID和產品種類，後端會傳回產品的最大頁數。

Add_Product_In_Cart (Create):

前端輸入ProductID,ProductSupplierID,CustomerID和ShopID，後端會將此產品加入到這位顧客的購物車裡，如果已經在購物車了則會回傳error。

Delete_Product_In_Cart (Delete):

前端傳入使用者id、產品資訊，後端會從購物車裡刪除商品

Get_Products_IN_Cart (Read):

前端傳入CustomerID，後端則會傳出此顧客他的購物車裡有什麼內容。

Buy_All_In_Cart (Create, Update, Delete):

前端傳入CustomerID，後端會幫此顧客購買他所有在購物車裡的產品。

AddProductNum (Update):

前端傳入CustomerID,ProductSupplierID,ShopID和ProductID，後端會幫此顧客增加一個這個商品在他購物車的數量。

SubtractProductNum (Update):

前端傳入CustomerID,ProductSupplierID,ShopID和ProductID，後端會幫此顧客減少一個這個商品在他購物車的数量。

Customer_TradeHistory (Read):

前端傳入CustomerID跟頁數，後端會回傳此顧客的交易歷史。

GetTradeHistoryMaxPage (Read):

前端傳入CustomerID，後端傳回此顧客交易歷史最大頁數。

Manager API:

UserExist (Read):

確認註冊的帳號是否已經存在

Register (Create):

輸入使用者資料及密碼並註冊成為賣家

GetRevenue (Read):

前端傳入ManagerID，後端會回傳此管理者的營業額。

GetOrderHistory (Read):

前端傳入ManagerID。

GetTradeHistory (Read):

前端傳入ManagerID，後端會回傳此管理者的交易歷史。包括time, product name, quantity, price, historyID, total price, customerID。

GetStoreHouseID (Read):

後端會回傳所有倉庫的ID。

OrderProduct (Create, Update):

前端輸入StoreHouseID, ShopManagerID, ProductSupplierID, ProductID, Num，後端會幫此管理者下單他要的產品以及數量到倉庫裡。

GetForSale (Read):

前端輸入ManagerID，後端會回傳此管理者在架上的所有產品。包括ProductID,SupplierID,Productname, Productprice, Quantity。

GetHave (Read):

前端輸入ManagerID，後端會傳回此管理者擁有的所有產品以及數量。包括 ProductID,SupplierID,Productname, Quantity, StoreHouseID。

ForSell (Create, Update):

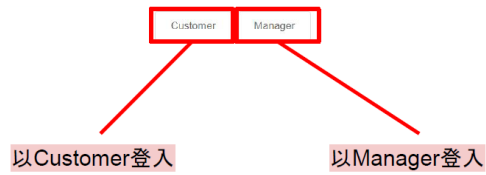
前端輸入ManagerID,ProductID,SupplierID, Quantity,ProductPrice, StoreHouseID，後端會把此管理者這個產品在架上的數量增加，並且減少庫存數量，如果不在架上則會幫忙新增。

PriceChange (Update):

前端輸入ManagerID,ProductID,SupplierID,ProductPrice，後端會幫忙把此產品的價格改成輸入的價格。

● 系統功能

首頁：決定登入的身分
Customer 或是 Manager



Customer Login

Customer 登入畫面
輸入帳號密碼登入

Login

Don't have an account?

Register

logout Products Cart History

登出、進入商店頁面、進入購物車、進入購物紀錄

透過商品種類篩選

Select Type:

all

Select Shop

all

透過商店篩選

Product	Shop	Remain	Price	Type	
product81	shop90	105	100	meat	Add
product82	shop90	111	150	drink	Add
product80	shop90	177	99	fruit	Add
product80	shop90	166	199	fruit	Add
product81	shop91	164	200	meat	Add
product83	shop91	168	300	clothes	Add
product84	shop91	112	400	shoes	Add
product84	shop92	135	310	shoes	Add

Customer首頁(商城)

可以看到所有可購買的商品，也可以透過特定條件篩選，按下add將其加入購物車

新增商品至購物車

logout Products Cart History

Customer購物車

Name	Shop	Remain	Price	Quantity
product80	90	177	99	- 5 +
product81	90	105	100	- 2 +

Total Price: 695

Submit

從商店新增的商品會進入購物車，在購物車確認商品與數量後即可購買

購入購物車中的商品

商品最終購買數量確認與調整

logout	Products	Cart	History
--------	----------	------	---------

Order history			
2021-05-04	ID:1120	\$4000	
Name	Shop	Number	Price
product86	shop64	4	1000

Customer購物紀錄

將購物紀錄以一筆交易為單位列出

#一筆交易：按一次Submit為一筆交易

Last Page 1 Next Page

上/下一頁

Manager Login

Manager 登入畫面

輸入帳號密碼登入

login

password

Login

Don't have an account?

Register

logout	Order History	Trade History	Order	Shop	Revenue
--------	---------------	---------------	-------	------	---------

登出、進貨紀錄、交易紀錄、進貨、商店頁面、收益紀錄

Manager首頁(商店管理)

確認及管理上架商品(ForSale)的種類以及售價

ForSale			
Product	Supplier	Number	Price
product80	70	177	99
product81	70	105	100
product82	70	111	150
product80	71	166	199

上架中的商品

設定價錢與上架數量後上架

在倉庫中的商品

Product	Supplier	StoreHouse	Remaining	Number			Price			
product82	70	102	1112	-	1	+	-	150	+	Add
product80	70	103	8	-	1	+	-	99	+	Add
product80	71	101	1660	-	1	+	-	199	+	Add
product81	70	101	3002	-	1	+	-	100	+	Add

Order History

Manager進貨紀錄
向供應商進貨的紀錄

Order ID	Product	Supplier	Number	Time
10	product81	supplier70	12	2021-7-20 05:12:20
11	product82	supplier70	3	2020-12-31 13:15:55
12	product80	supplier71	1314	2019-4-28 23:24:12
20	product81	supplier70	1000	2021-12-24 07:17:17
21	product80	supplier70	2	2022-01-18 03:42:54
22	product80	supplier70	2	2022-01-18 03:43:22
23	product80	supplier70	2	2022-01-18 03:43:49
24	product80	supplier70		2022-01-18 03:44:12

Manager交易紀錄
與Customer的交易紀錄

Date	HID	Customer	
2021-05-05 05:23:00	1111	108703043@nccu.edu.tw	297
2021-04-20 06:46:00	1113	108703030@nccu.edu.tw	750
2021-12-28 07:31:30	1121	108703030@nccu.edu.tw	90

顯示細節

選擇存放倉庫

Select Store House

Manager進貨畫面
向Supplier進貨

Product	Supplier	Type	Number	Order
product80	70	fruit	1	Order
product81	70	meat	1	Order
product82	70	drink	1	Order
product80	71	fruit	1	Order
product81	71	meat	1	Order

選擇數量

進貨

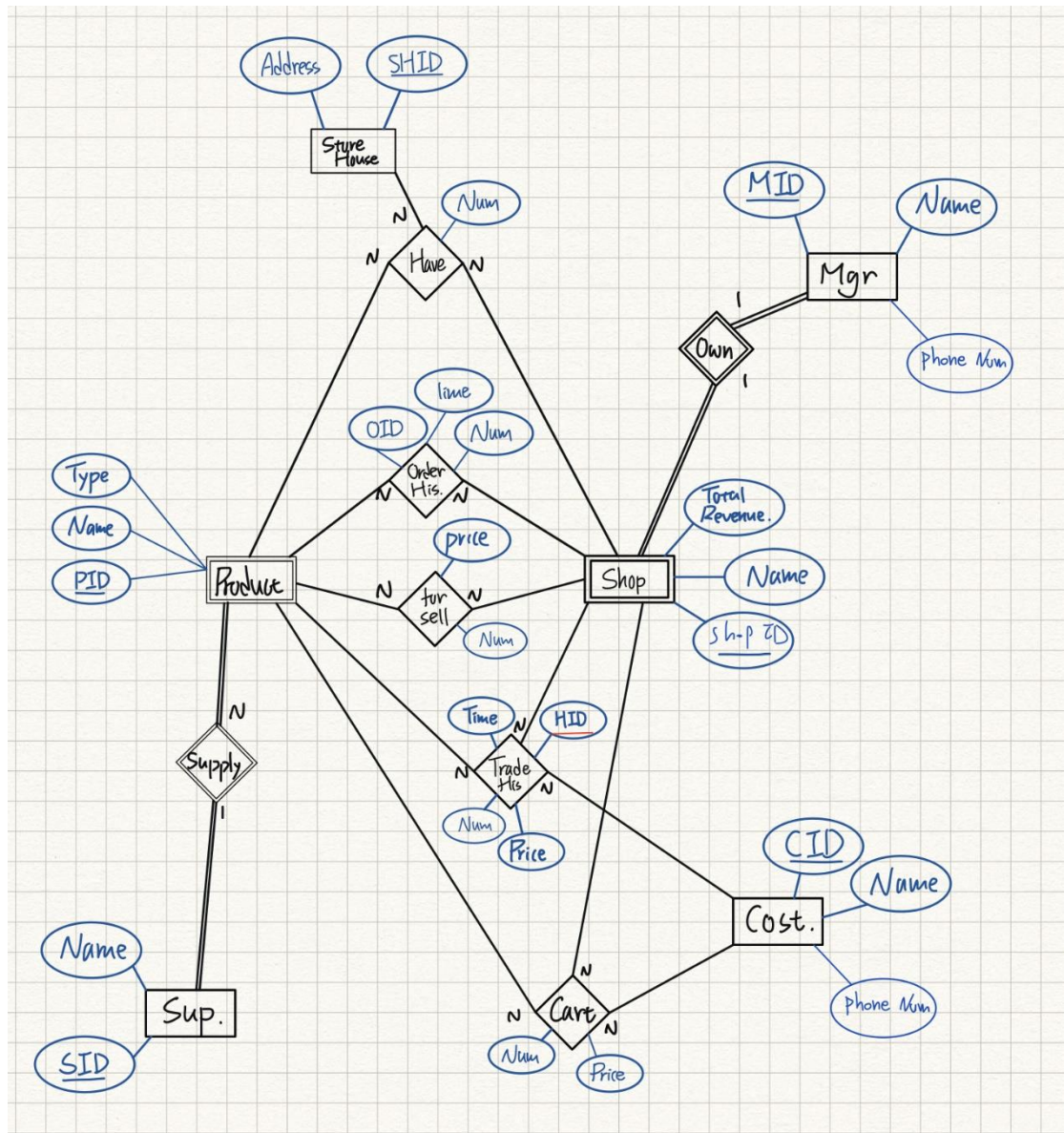
選擇時間區間

ALL

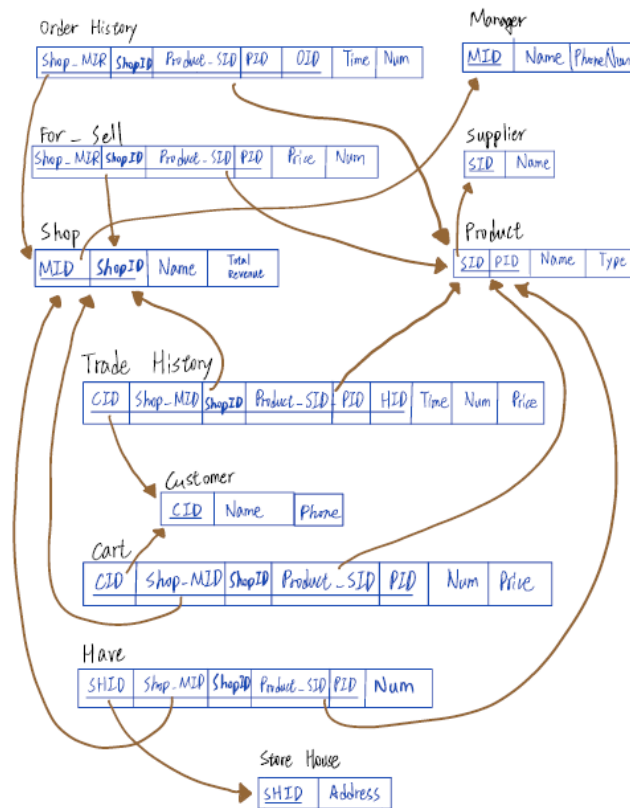
Manager收益紀錄
收益折線圖



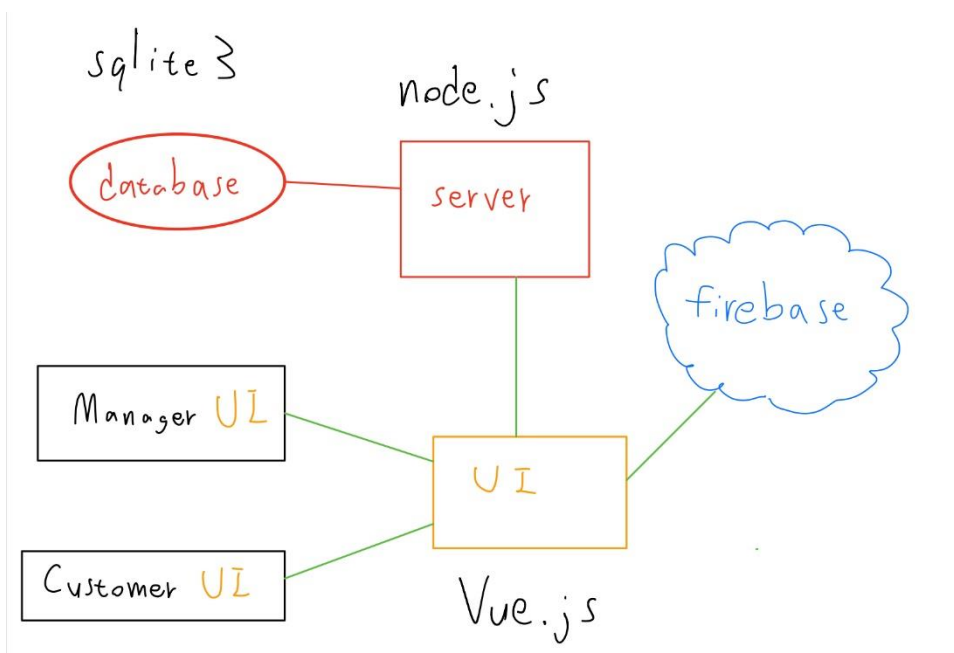
● ER Model



● Relational Schema



● 系統架構



● 心得

賴冠瑜：這次專題的專題，主要挑戰是在前後端的溝通，以及取得系統功能和可行性之間的平衡，因為不是所有組員都做過系統，寫過前後端，全組只有我同時接觸過這兩項 node.js 和 vue.js，所以專案一開始，主要是帮大家解決學習新工具時，遇到的一些問題，在系統功能設計時，也不能設計的太複雜，不然大家最後可能做不出來。等大家都學得差不多後，就開始實作，本來覺得不需要做測試 API，前端直接用我們寫好功能的 API，但後來發現這樣反而會沒效率，因為前端要等後端寫完，才能根據回傳值去改後端的 code，這在下次專題時會特別注意，但做到後來，大家都熟悉這個模式後，實做起來也算快。開發系統時，有些功能也會臨時更改，也會需要考慮到對整個系統的影響。以上都是很好的經驗，在下次遇到類似問題時，可以更好的解決。

田詠恩：我覺得這學期的資料庫系統非常的充實，上半學期有三個作業，讓我們更熟悉 ER-model, Relation model, SQL。下半學期則是專題實作，又剛好我是負責後端的部分，所以對資料庫的建造、schema 有實作上的認識。很感謝跟我同組的組員，每個人都很負責任，使我們這次專題完美結束。這個專題是上大學來第一次有實作的機會，也是第一次這麼多人共同和合作，不僅能學到程式開發，更能學到分工、溝通的重要。最後要謝謝老師跟助教們這學期辛苦的付出，才能有這麼完整、有系統的教學體驗。

蔡博超：這次的資料庫專題的工作量非常的龐大，幾乎每個禮拜都要花上至少一整天來做這個專題。我之前有稍微接觸過一些前端，所以當了前端的負責人。在過程當中難免遇到一些 bug，有時候不知道是後端出了問題，還是自己前端的部分出了問題，在這其中學習到了如何跟別人合作寫程式，並 debug 這種由多人一起寫的程式。除此之外，我還對 git 的用法更加的熟悉，例如：merge conflict 怎麼解決等等。這次的專題讓我學到了很多東西，最後做出了一整套的系統也是一件很有成就感的事。

張誠宗：這次 DataBase project，讓我學到了很多東西，像是 javascript、vue、網頁設計環境設置和 github 指令等等，能順利完成都是多虧了強力的隊友領導，給予我很多幫助，在做前端的時
候，遇到的最大困難是 vue 環境的建構，因為是第一次使用，所以對於一些語法和工具相當不熟悉，尤其是在安裝圖表模組時，因為使用的人較少，所以很難找到使用方法。再來就是這次 project 花了很多時間討論，感覺團隊合作很重要。

倪靖揚：這次專題報告我學到了非常多，一開始聽教授上課覺得非常抽象，但真正實作起來真的是很有趣，而且自己實際做過以後就發現上課有聽沒有懂的內容其實滿好上手的。像我在這次報告負責的是撰寫後端 api 跟建 db 檔，真正操作一次那些程式碼後才完全能理解老師在上課講解的東西，而且建 db 檔讓我也對資料庫這個東西有更深入的理解，從抽象的概念變成很具體的畫面。這次專案我還學到了很多上課沒教的東西，大部分都是需要用到時上網查的，也在實作過程中想起老師教的一些觀念，在很多卡關的時候都是想起 ppt 裡的內容讓我可以突破難關，才讓我負責的地方順利圓滿完成。這堂課讓我受益良多，也真的非常喜歡實作的這個課程，學以致用的感覺真的很好。

林宏睿：這次專題我負責前端的工作，在這之前有簡單碰過 HTML、CSS 和 JavaScript，不過這是我第一次使用 Vue 框架去架構網頁。從透過 Component 去區分不同功能的區塊，到運用 element plus 去實現網頁內容、第一次用 axios 去 get 和 post 後端的資料、使用 table plus 去觀察實際資料庫的變化等等…。過程中也遇到一些問題，比如如何針對一個 row 裡的商品數量做變化、如何顯示不只有一項商品的訂單、如何在下拉表格內再建立一個 table 等…。撇除程式實作的部分，在剛始設計 ER Model 和 Relational Schema 也是不容易的工作，不僅需要縝密設計 Entity、Attribute、Relation，更要分析實際可行性和可能衍生出的問題。接著要去針對想要實現的功能去區分做哪些 Component，並個別去設計它的 UI，並對每個 Component 的需求整理 API 給後端實現。無論是在 ER Model、Relational Schema、Component 或是 API 需求的設計上，都時常需要作出調整，也必須定時跟組員聯絡以更新需求，相比以往沒有那麼多資料庫概念的專案，每個部份的緊密度不但更加提升、實作難度也上升不少且工作量和考量因素都多上許多，雖然這次專案麻煩了不少，卻是一個有全新、具有挑戰性的體驗。

吳尚恩：這是我第一次和一個小組一起建立一個包含前後端的小系統，真的從中學到了非常多東西。我在這次專案中主要負責的部分是需求發想、ER-Model 繪製、前端，為了希望可以對計畫中的各項細節做出明確的定義，我們在前面兩者花了許多時間討論以及修正，試圖想出各種可能的情況；在我們好不容易完成概念設計，進入實作階段後，才發現"沒想到的問題遠比想到的多"，一路上需求被更(闕)改(割)過許多次，前後端之間也要不斷對 api 的功能溝通(多半是前端許願後端生產)，最終才把系統建立出來。另外，因為我這次也有負責前端，我們用的框架是 Vue3，對一個以前沒有寫過網頁的我來說，這學習曲線真的是蠻陡峭的，花了好大把力氣才入門這些語言的使用方法和語言邏輯；但我想這就是業界真正的學習情景，不像在學校會有人以系統性的方式整理好，除了學習本身之外，規劃學習的路線也是相當重要的一環。

總之，整體來說，經過這次的專案，我對程式語言、學習心態、開發流程等等，都有了新的一些見解和想法，經驗值大飆升！