

國立臺灣科技大學資訊工程系

一百零六學年度第一學期專題研究 總報告

點餐系統搭配電子菜單製作

研究組員

李垣峻

B10315016

黃湧鼎

B10315027

黃若望

B10315033

林煜宸

B10315045

指導教授：_____

中 華 民 國 106 年 1 月 4 日

目錄

目錄.....	1
致謝.....	2
一、研究動機.....	3
二、網站介紹.....	4
三、實作過程.....	33
四、學習心得.....	40
五、測試結果.....	42
六、結論.....	47
七、未來發展.....	48
八、參考文獻.....	49

致謝

一路走來需要感謝的人太多了，首先想感謝我們的專題教授——吳怡樂教授，每次與他開會之後，都會受到許多啟發，教授會給我們很多想法上的啟迪，對於我們提出的想法，他也可以告訴我們實作上需要注意的地方，讓我們少走了很多冤枉路，此外教授說：「什麼東西都可以做，只要是創新的就可以。」我們很感謝教授給我們自由發揮的空間。

另外，除了組員之外，還有其他同學提供給我們技術上的支援，還有一些使用者回饋，讓我們更能知道，這專題還有什麼不夠好的地方可以加強，也感謝網路上各種開源的程式碼，跟一些程式問題交流的平台，沒有他們我們的專題一定無法如期完成，真的很感謝願意分享的人如此之多。

也想感謝從大一到大四的教授，如果不是你們的辛苦指導，我們也不會有這些知識可以實踐我們的想法，感謝一路上支持我們的任何朋友，你們的支持是我們堅持完成的動力，此專題結果獻給這些支持、幫助過我們的人們。

專題小組 誌於
台灣科技大學資訊工程系
中華民國一百零六年一月

一、研究動機

會想要開發一個半自動的點餐系統，是因為在摩斯(MOS)漢堡用餐時，察覺到一個現象，MOS 的菜單擺在架子上，但是幾乎沒有人會去拿，都是排隊排到輪到自己時，才看桌上的菜單，然後決定很久，服務生也只能等待客人決定，也因此每次 MOS 都大排長龍，我們想要解決這個現象。

此外，有些說明可能無法寫在菜單上，因為實體菜單的大小有限，但如果使用電子菜單就沒有這問題了，可以用一個彈出的視窗，上面有詳細的餐點成分，如此一來顧客吃得更加安心，服務人員也省了回答顧客的時間，還有些餐廳的餐點並沒有圖片，如果是電子菜單可以每份餐點都搭配圖片，這樣可以讓顧客挑選餐點時，有更多參考的因素。

最後一個原因，是有一次去餐廳用餐時，揮手揮很久服務生都沒有注意到，那時候覺得如果可以有服務鈴，那一定很方便，但不是每家餐廳都有，這些原因讓我們想要做一個「可以節省服務生時間、不用排隊可以坐著看圖片點餐、有服務鈴功能」的點餐系統。

二、網站介紹

網站功能

此點餐系統主要有以下功能。

1. **自製菜單**：餐廳端註冊並登入之後，就可以使用我們的自製菜單功能，我們有提供編輯菜單的頁面，完成編輯之後按下儲存按鈕，即可完成菜單設定。
2. **點餐 SOP**：我們的點餐系統，將掃 QR code 帶位、點餐、確認點餐、餐點明細、服務鈴一手包辦。
 - (1) **QR code 帶位**: 獨一無二的 token，結帳完成之後該 token 就會被移除，目的是為了只讓在餐廳中的顧客點餐，防止惡意騷擾。
 - (2) **點餐**：每一道菜都有圖片、價錢、介紹，決定想要的份數之後，就可以將喜歡的餐點加入購物車。
 - (3) **確認點餐**：會列出顧客點的餐點以及數量，如果都正確，就按下確認。
 - (4) **餐點明細**：確認點餐之後會跑出餐點明細，上面有個欄位是「待上」，代表還有幾道菜還未送到，該欄位的數值會隨著餐點的到來減少。
 - (5) **服務鈴**：當有需要服務生之時，就可以按下服務鈴，但是在服務生過來之前，不能重複按。

系統流程圖：

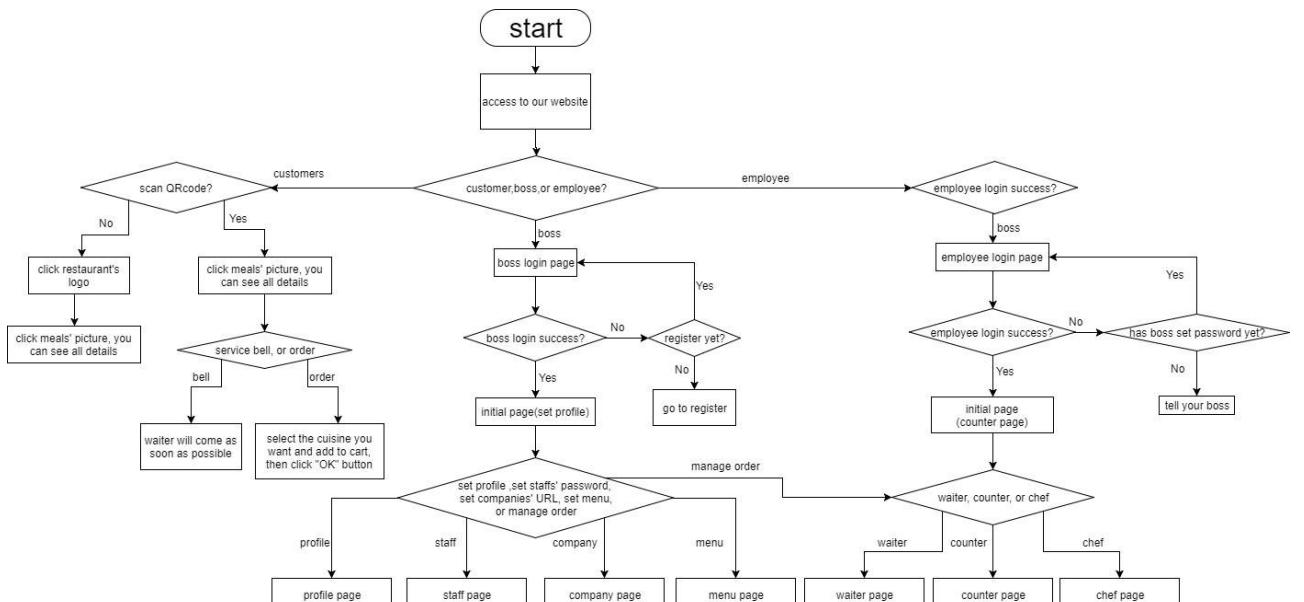


圖 2-1 系統流程圖

循序圖：

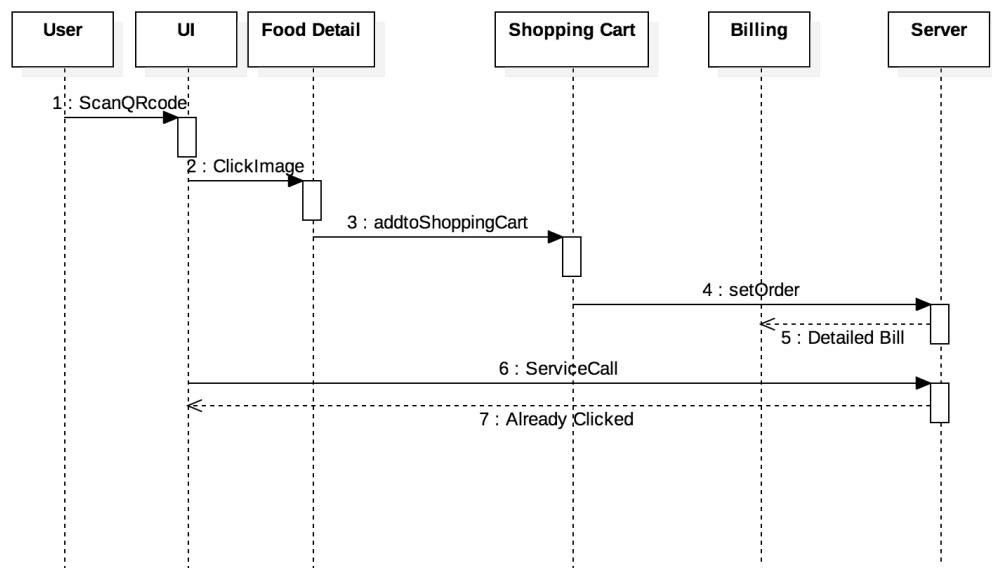


圖 2-2 客戶端循序圖

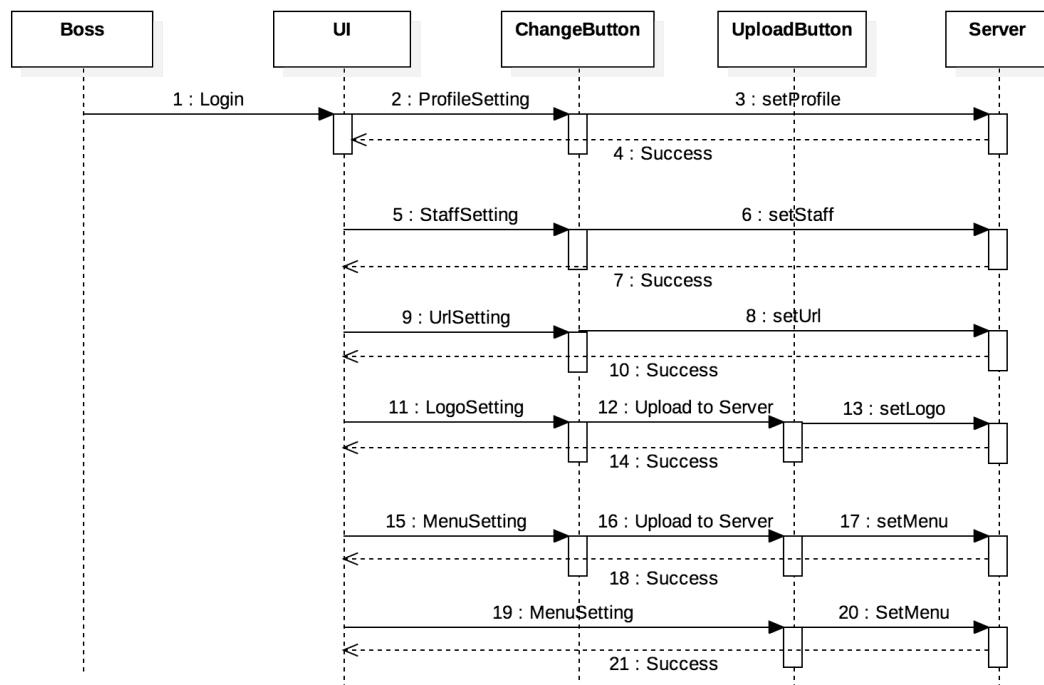


圖 2-3 最高權限(老闆)循序圖

Smart Menu Special Project

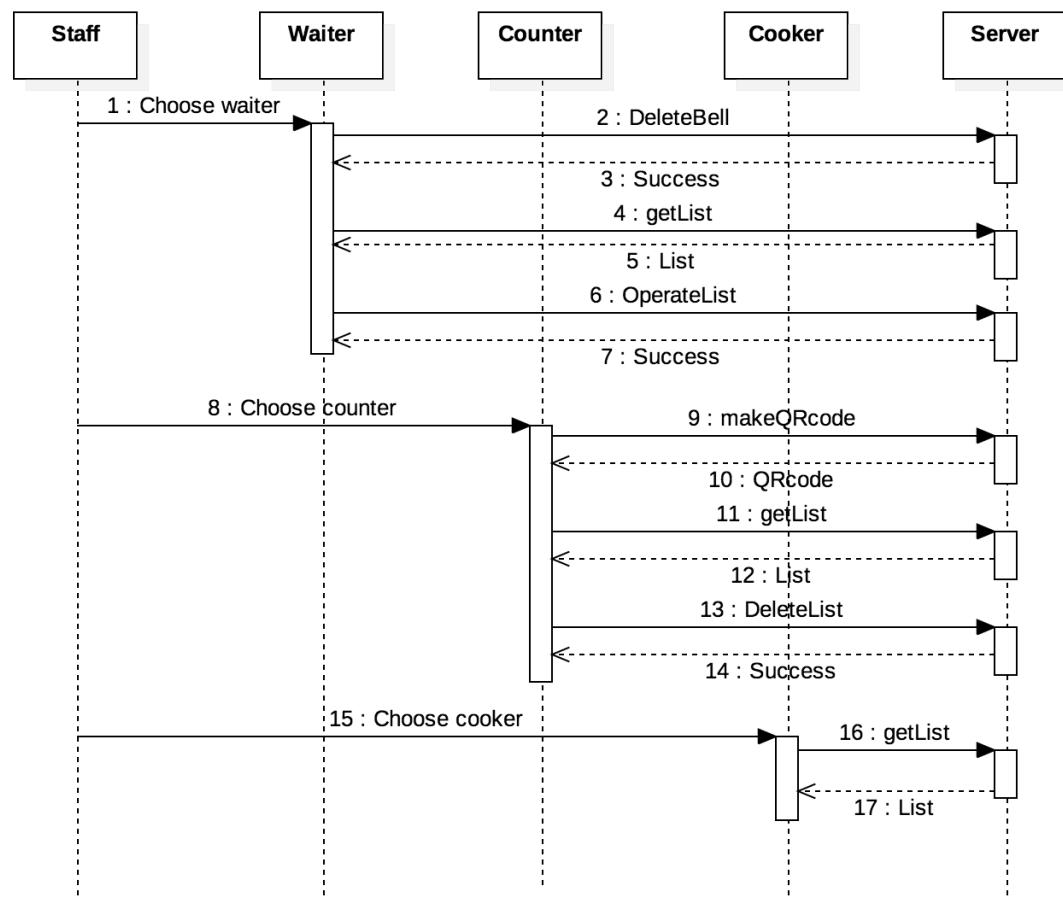


圖 2-4 員工循序圖

實際操作流程：

先以老闆的角度進行解釋。

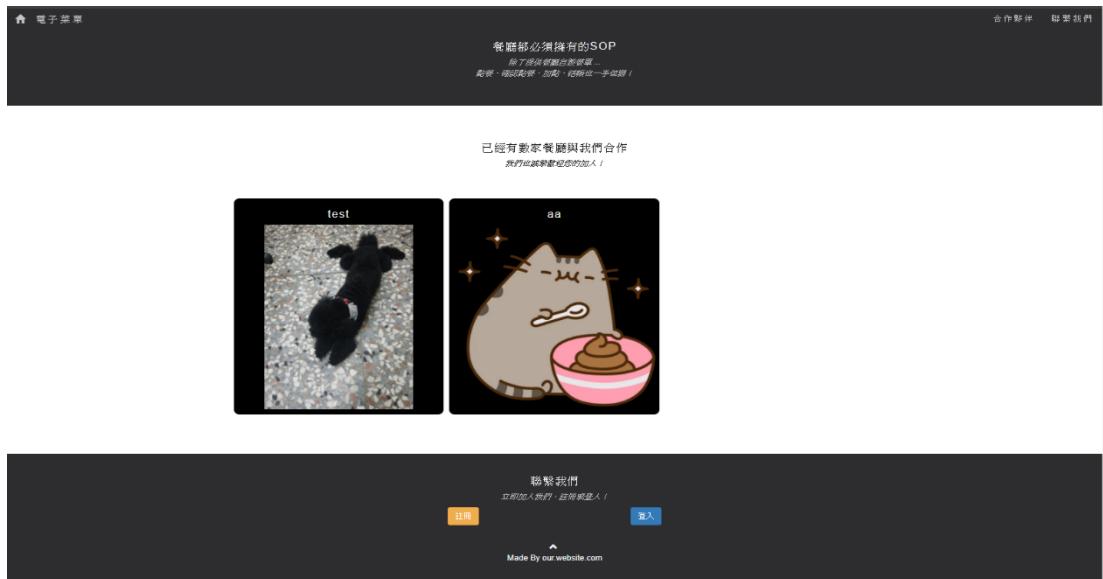


圖 2-5 連上網站

輸入網址後會進到此頁。網址為：<https://www.huuuuuding.me/> 其中 test 、 aa 為餐廳名稱，點下圖片可以連到該餐廳的菜單，但是沒有點餐功能。

The screenshot shows a registration page with a dark background. It includes sections for '帳號資訊' (Account Information), '餐廳資訊' (Restaurant Information), and '申請人資訊' (Applicant Information). The '帳號資訊' section has fields for '帳號' (Account) containing 'NTUST', '密碼' (Password) containing '.....', and '確認密碼' (Confirm Password) containing '.....'. The '餐廳資訊' section has fields for '餐廳名稱' (Restaurant Name) containing '台科大餐廳', '地址' (Address) containing '臺北市大安區基隆路 4 段 43 號', and '電話' (Phone) containing '886-2-27333141'. The '申請人資訊' section has fields for '姓名' (Name) containing '老闆', '電話' (Phone) containing '0987-654-321', and '電子郵件' (Email) containing 'NTUST@delicious.com.tw'. A large blue '註冊' (Register) button is at the bottom.

圖 2-6 註冊畫面

網址為：<https://www.huuuuuding.me/auth/register/>，填妥所有資料後，按下註冊。

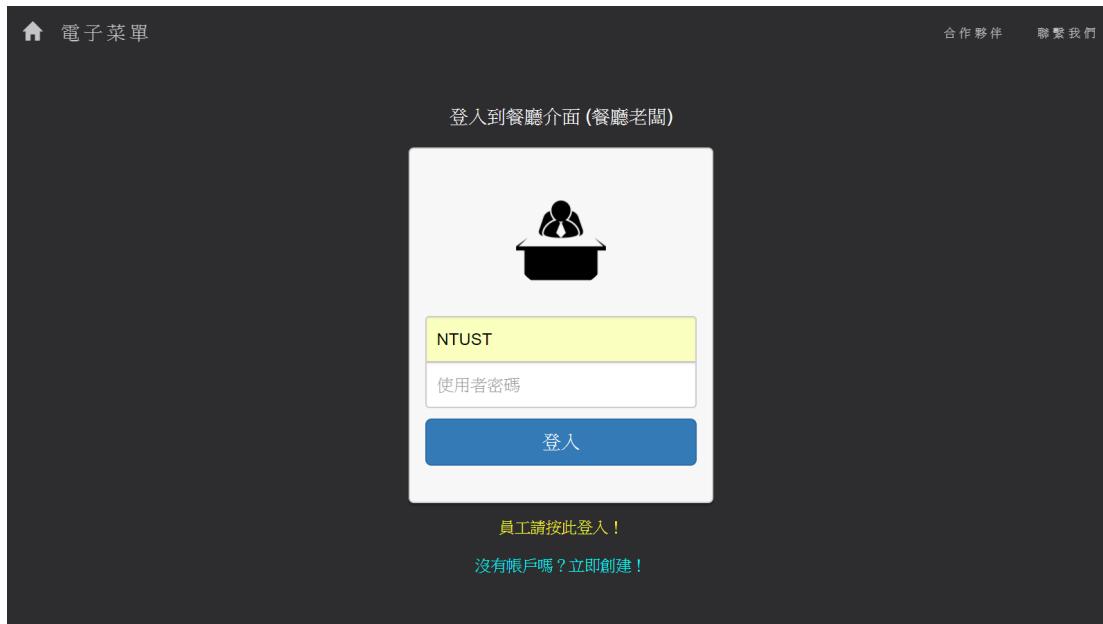


圖 2-7 老闆登入頁面

網址為: <https://www.huuuuuding.me/auth/login/>，按下註冊後若成功，系統會自動跳到老闆登入頁面，輸入剛剛註冊的帳號、密碼，正確即可登入。

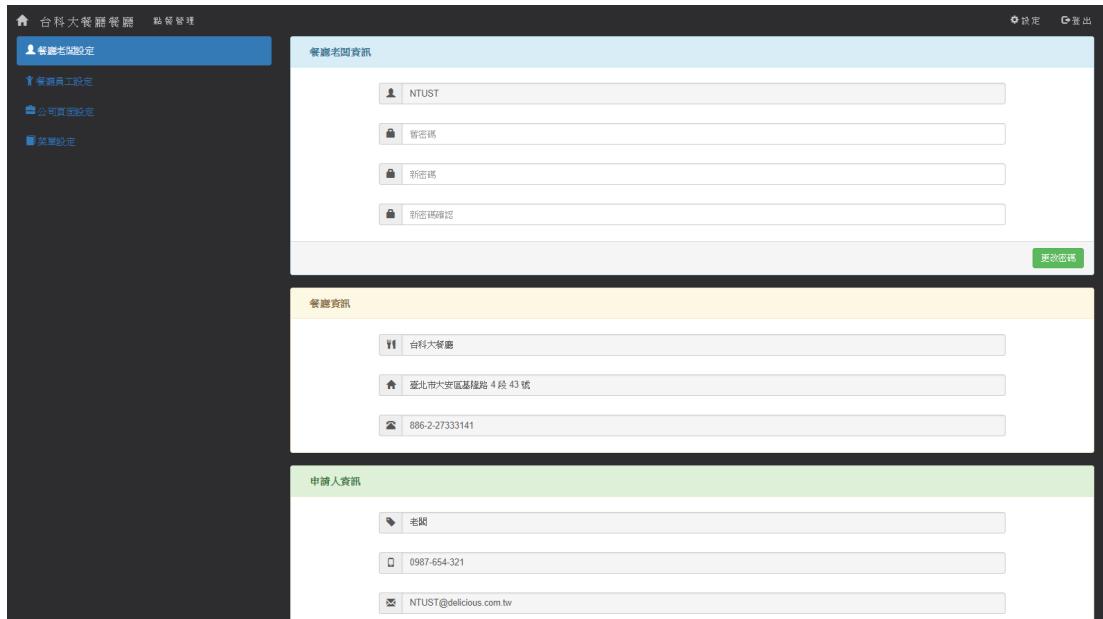


圖 2-8 更改密碼頁面

網址為: <https://www.huuuuuding.me/setting/profile/>，登入後，第一個頁面是更改密碼的頁面，而只有密碼可以更改，其餘的資料不可更動。



圖 2-9 按下餐廳員工設定

網址為: <https://www.huuuuuding.me/setting/staff/>，老闆要先設定員工的帳號密碼，員工的登入口才會有效。

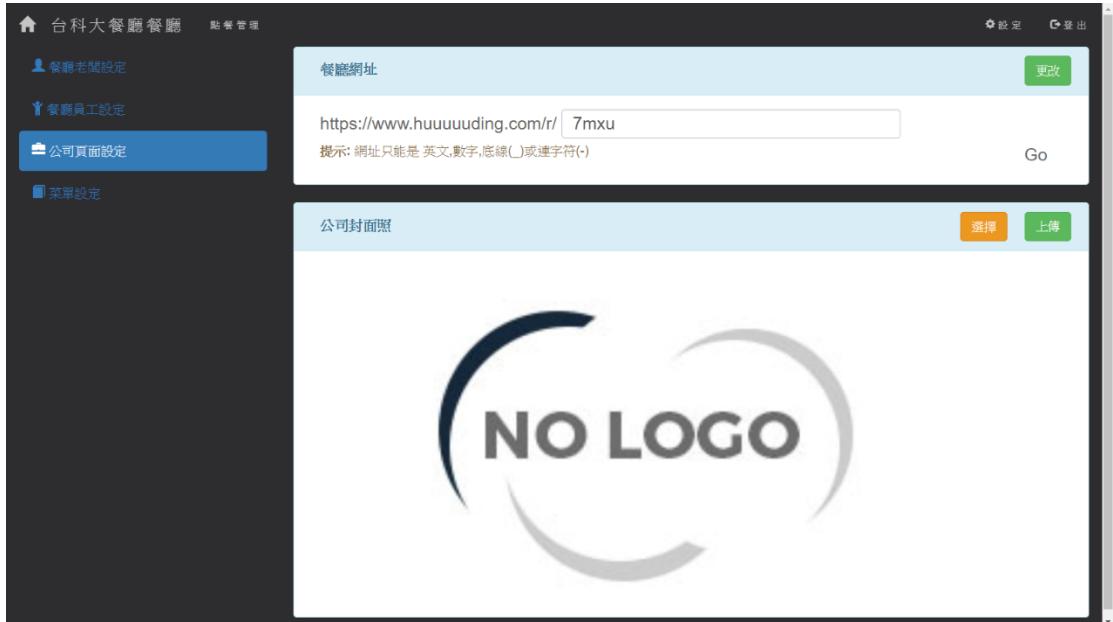


圖 2-10 按下公司頁面設定

網址為: <https://www.huuuuuding.me/setting/company/>，這個頁面可以讓餐廳自訂自己的網址、以及上傳餐廳的封面照片。



圖 2-11 按下 GO 畫面會跳到餐廳頁面

系統初始的 <https://www.huuuuuding.me/r/7mxu/> 就是餐廳頁面。

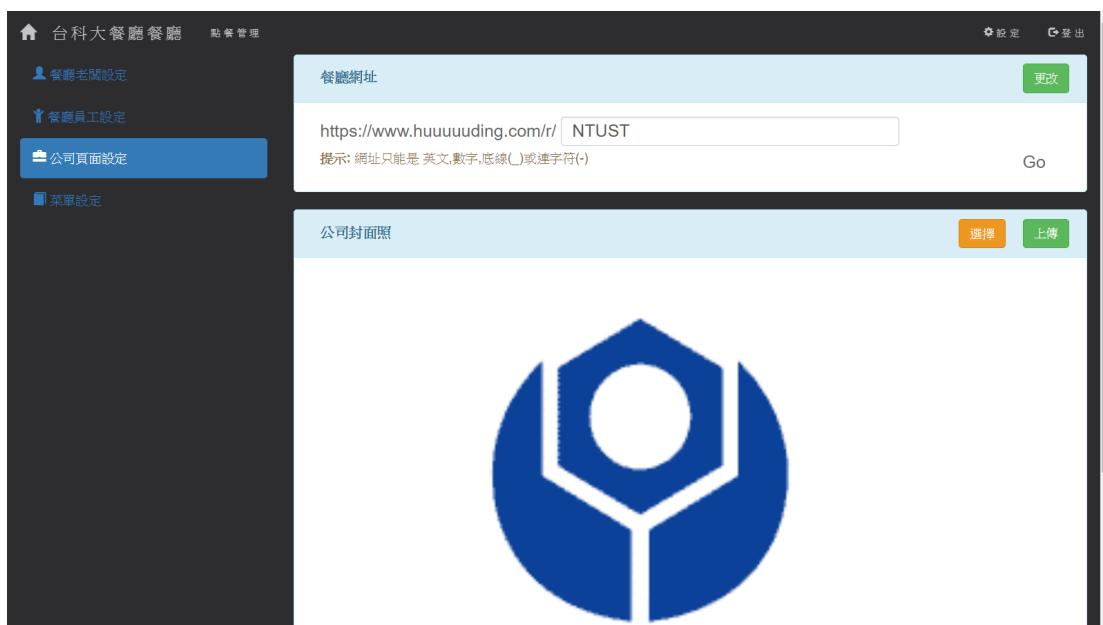


圖 2-12 改好網址、上傳封面

設定餐廳的 URL 為 <https://www.huuuuuding.me/r/NTUST/>，並上傳圖片，此圖片會顯示在 <https://www.huuuuuding.me/> 代表餐廳的封面，點下後會連到 <https://www.huuuuuding.me/r/NTUST/>。

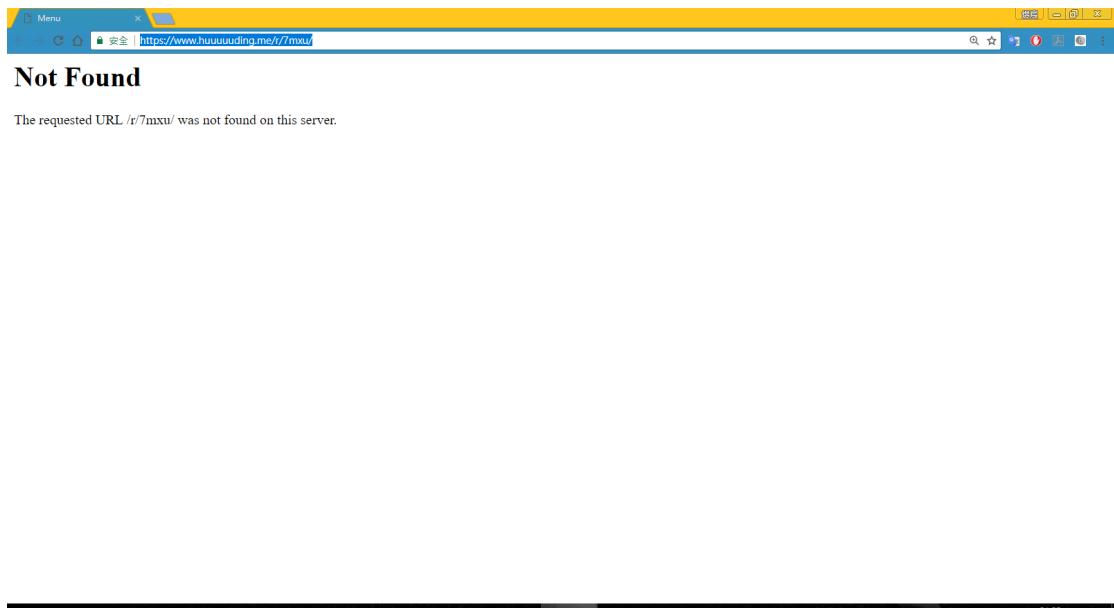


圖 2-13 原本的網址會失效

原本的 <https://www.huuuuuding.me/r/7mxu/> 就會失效。



圖 2-14 新的網址可以正常運作

這個 <https://www.huuuuuding.me/r/NTUST/> 就會是餐廳的網址。

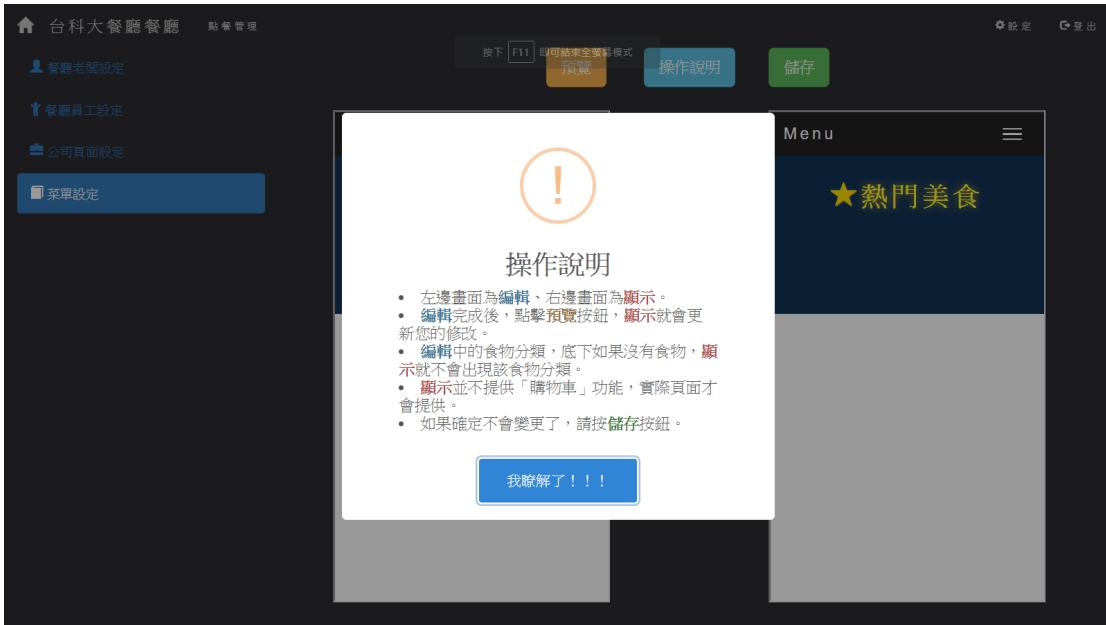


圖 2-15 菜單設定

網址為: <https://www.huuuuuding.me/setting/menu/>，第一次進入會有操作說明，此頁面可以讓餐廳自訂的菜單內容，會顯示在 <https://www.huuuuuding.me/r/NTUST/>。

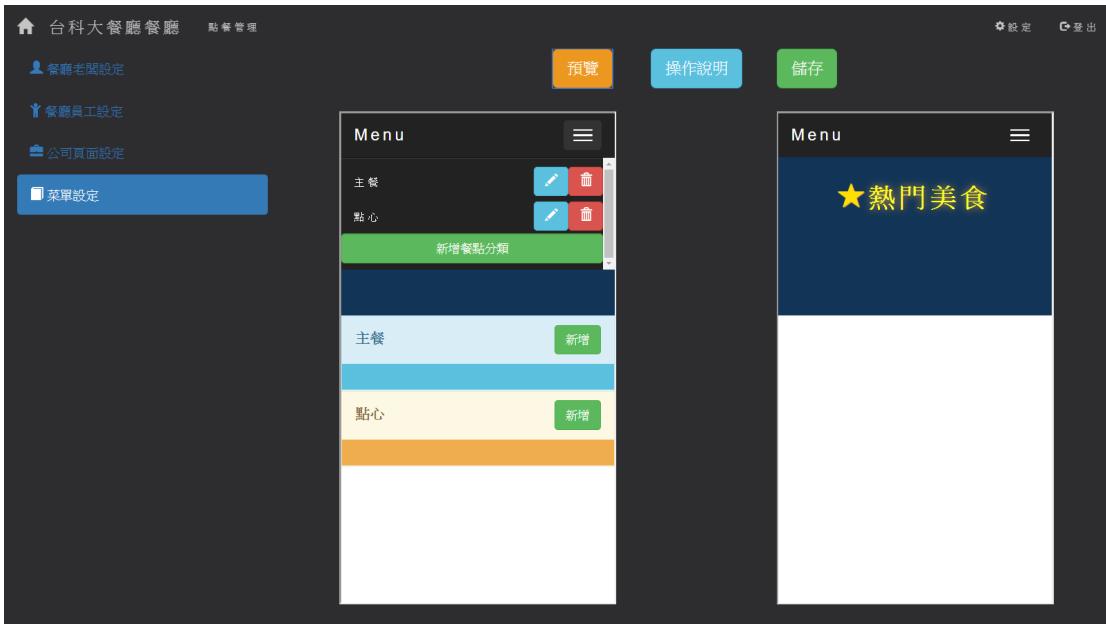


圖 2-16 菜單設定分類

必須設定分類才能在往下新增餐點的詳細資料，若只有分類在此時按下預覽並不會顯示任何東西。

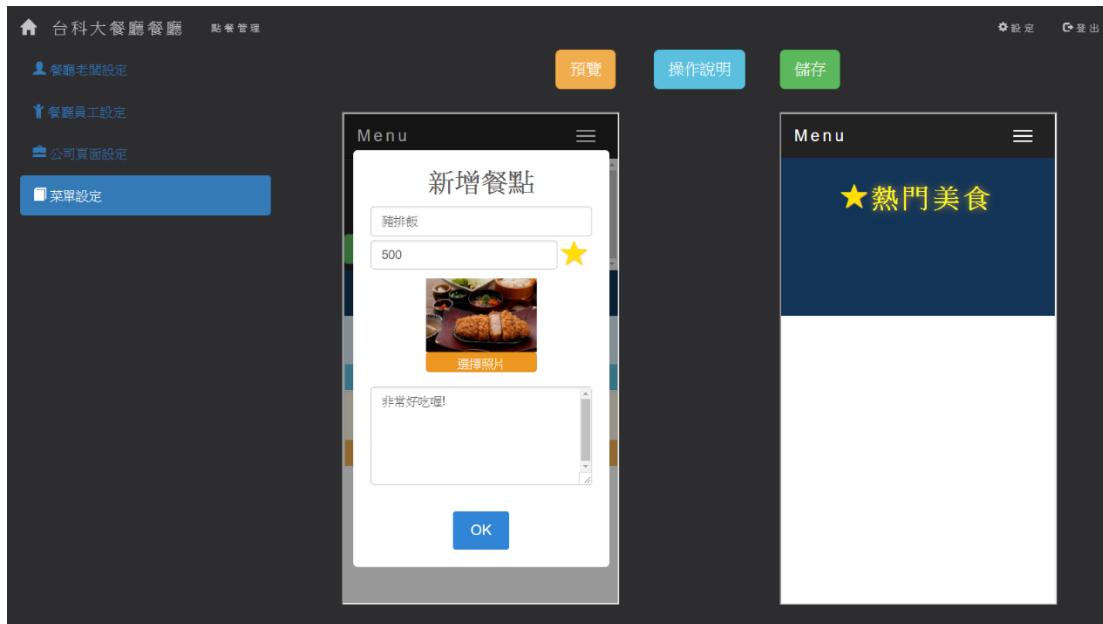


圖 2-17 菜單設定主餐

依序填入菜名、價錢、照片、內容，接著按下 OK；在價錢的旁邊有一個黃色星號，點一下可以啟動它，當它填滿時代表熱門美食，會讓它顯示在菜單的最上方。

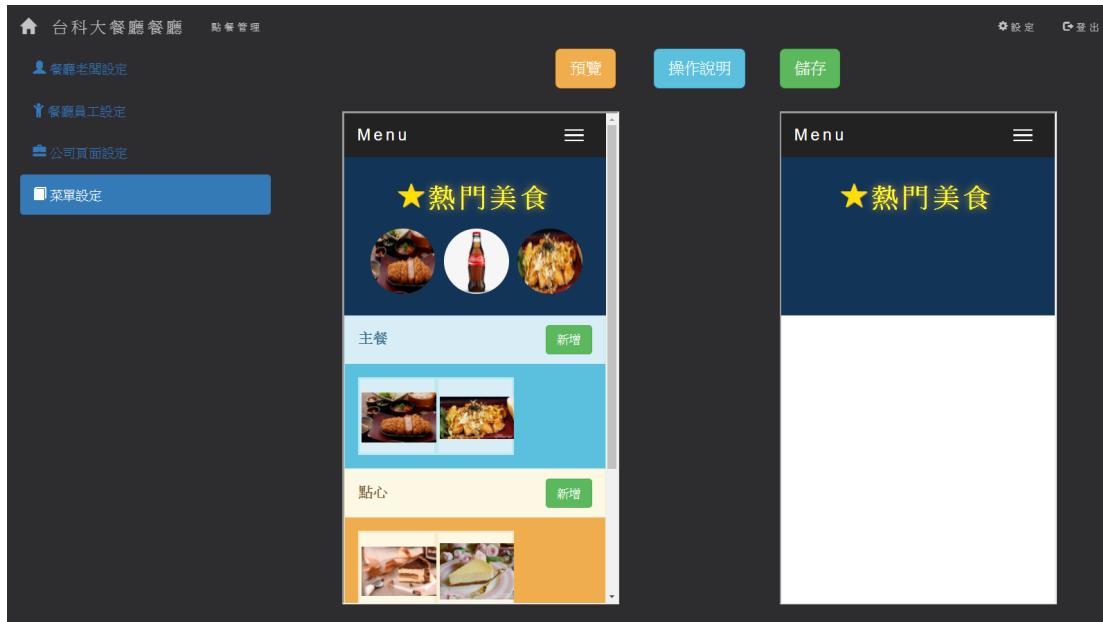


圖 2-18 新增完所有的餐點

按照上述所說一步一步把所有的餐點新增完，熱門美食的地方是可以滑動的，以三個餐點為一個頁面；若要更改餐點，點一下餐點即可修改餐點資訊。

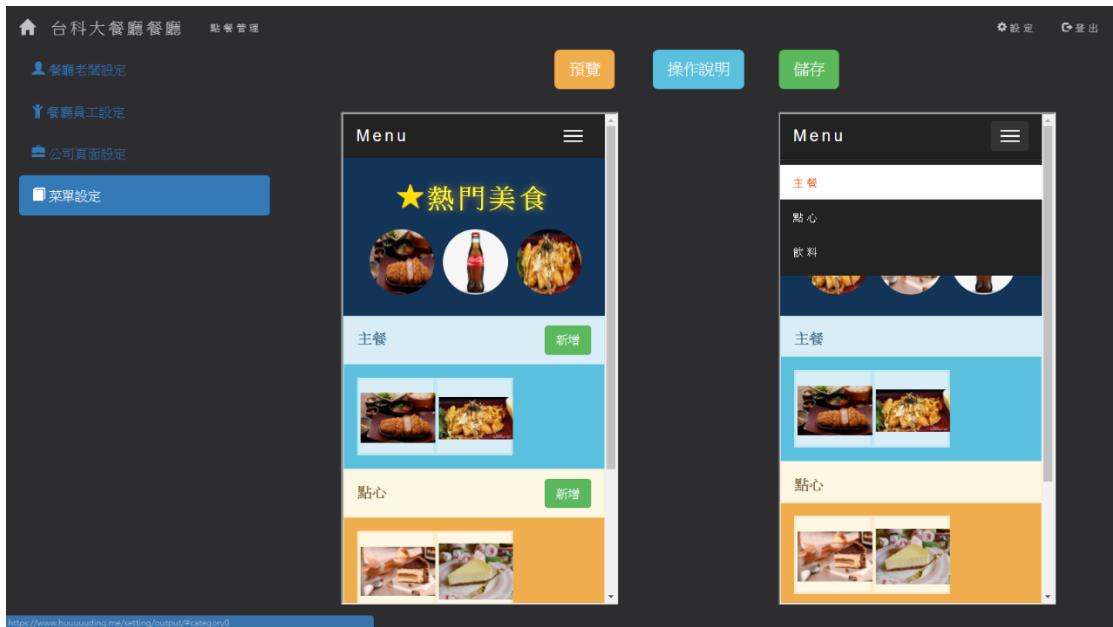


圖 2-19 按下預覽後

左上方的選單可以看有哪些分類，你可以在預覽確定你設定的東西，格式有沒有跑掉。

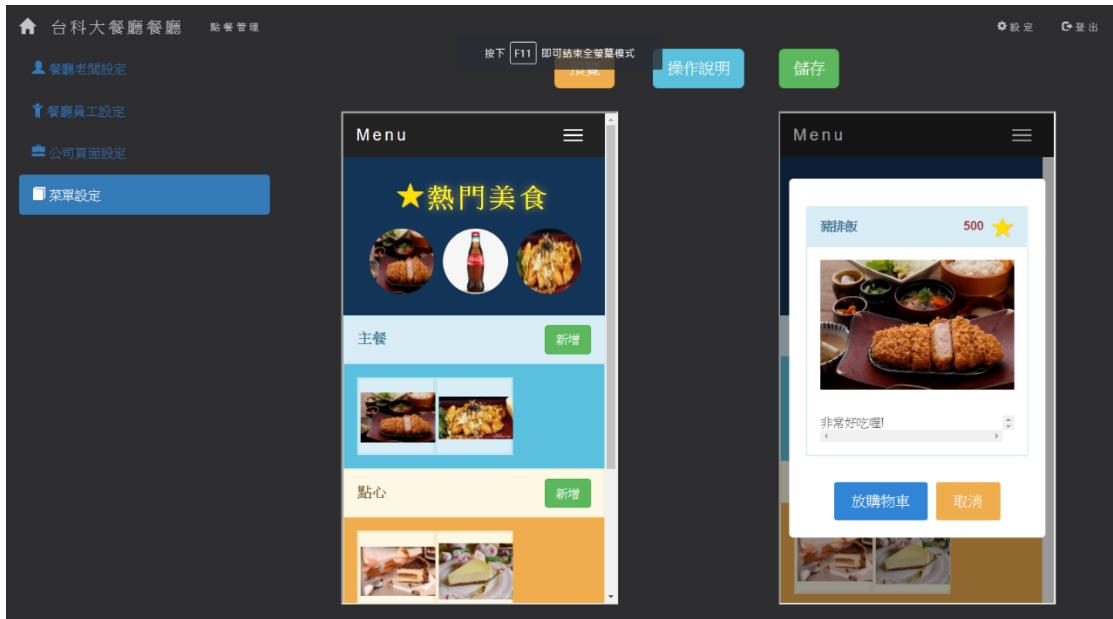


圖 2-20 在預覽的地方按下圖片

你可以在預覽的地方按下圖片，這就是儲存後在
<https://www.huuuuuding.me/r/NTUST/> 顯示的樣子。

老闆的實際操作流程到這邊為止，由於管理點餐這部分，員工也有(老闆只需按下“點餐管理”就會跳到一樣的頁面)，所以以員工舉例，這邊就不再贅述，接下來是員工的部分。

員工的部分：

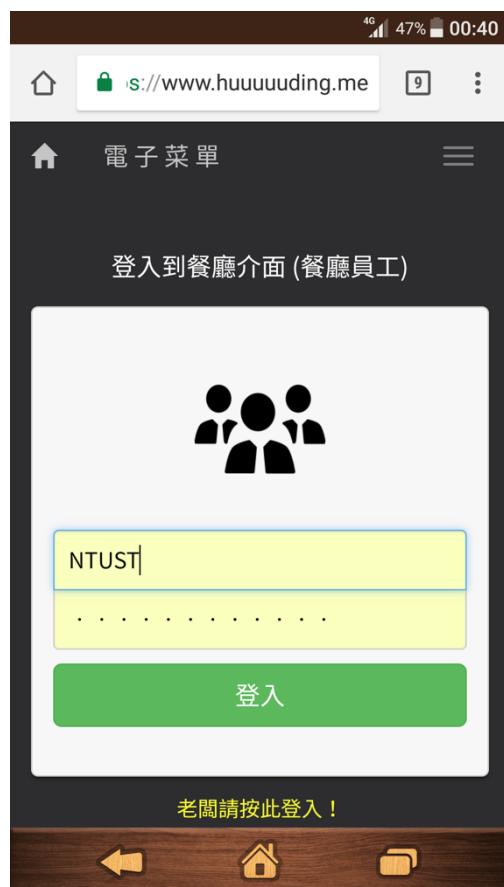


圖 2-21 員工登入畫面

網址為: <https://www.huuuuuding.me/auth/staffLogin/>，密碼為老闆在圖
2-6 設定的密碼。



圖 2-22 員工登入後畫面(櫃台)

網址為: <https://www.huuuuuding.me/staff/counter/>，登入後，第一個頁面是櫃檯的頁面，負責產生點餐的 QR code。



圖 2-23 產生 QR code

輸入人數、桌號，即可按下按鈕產生 QR code，顧客必須要掃描此 QR code 或輸入下面的縮網址，才可以開始點餐。



圖 2-24 櫃台的訂單列表

櫃檯的訂單列表部分，這裡顯示的是所有點餐完畢的桌號，該桌的價錢、餐點、份數。



圖 2-25 服務生頁面

網址為: <https://www.huuuuuding.me/staff/waiter/>，服務鈴列表會顯示按下服務的桌號，訂單列表可以讓服務生，送達餐點後按下減號待上就會減一，紀錄餐點送達情況。



圖 2-26 刪除服務鈴

當服務完該桌後，服務生可以再次點擊服務鈴刪除服務鈴。



圖 2-27 內場人員頁面

網址為: <https://www.huuuuuding.me/staff/chef/>，內場人員沒有操作的必要，因此僅顯示幾桌要做幾份餐點。

以上就是員工的所有操作，老闆可以透過點擊點餐管理，進入上述這些畫面，接著是顧客的部分。

顧客的部分：



圖 2-28 顧客掃描 QR code 之畫面

網址為: <QR code 之網址>，掃描櫃台人員提供的 QR code 即可進入點餐系統。



圖 2-29 無掃描 QR code 之畫面

網址為: <各家餐廳之網址>，不是由 QR code 進入就無法點餐，僅供瀏覽。

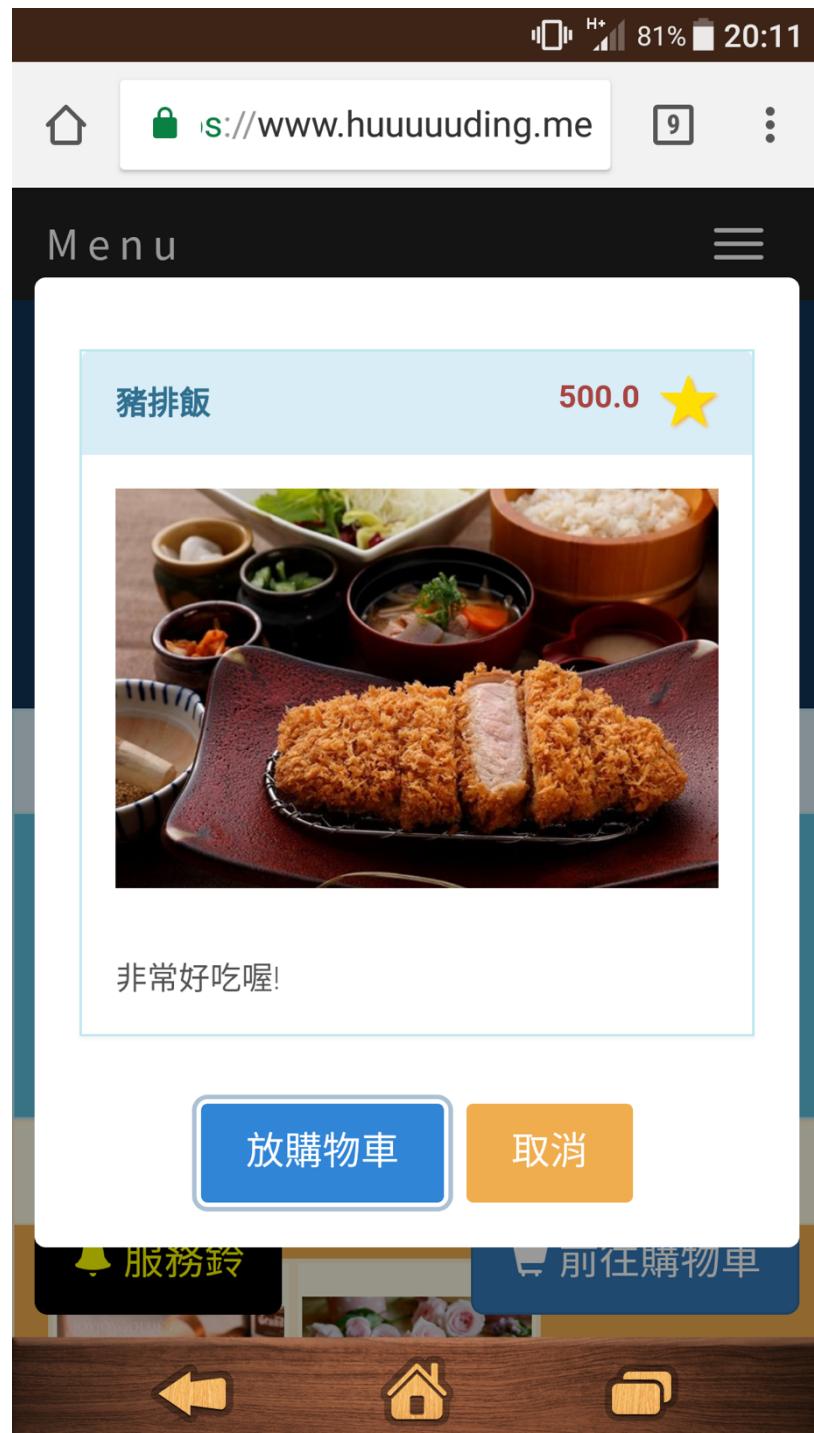


圖 2-30 按下你喜歡餐點的圖片

網址為: <QR code 之網址>，按下圖片後，可以看到品名、價錢、圖片、描述、是否為熱門商品。



圖 2-31 按下放購物車後之畫面

按下後，可以輸入你要的份數，必須是數字，之後畫面會跳回菜單。



圖 2-32 按下前往購物車後之畫面

依序新增你想要的餐點，之後按下前往購物車，即會看到此頁，顧客可以在這裏修改訂單。



圖 3-33 按下確定後之畫面

此畫面讓顧客再次確認菜單內容。

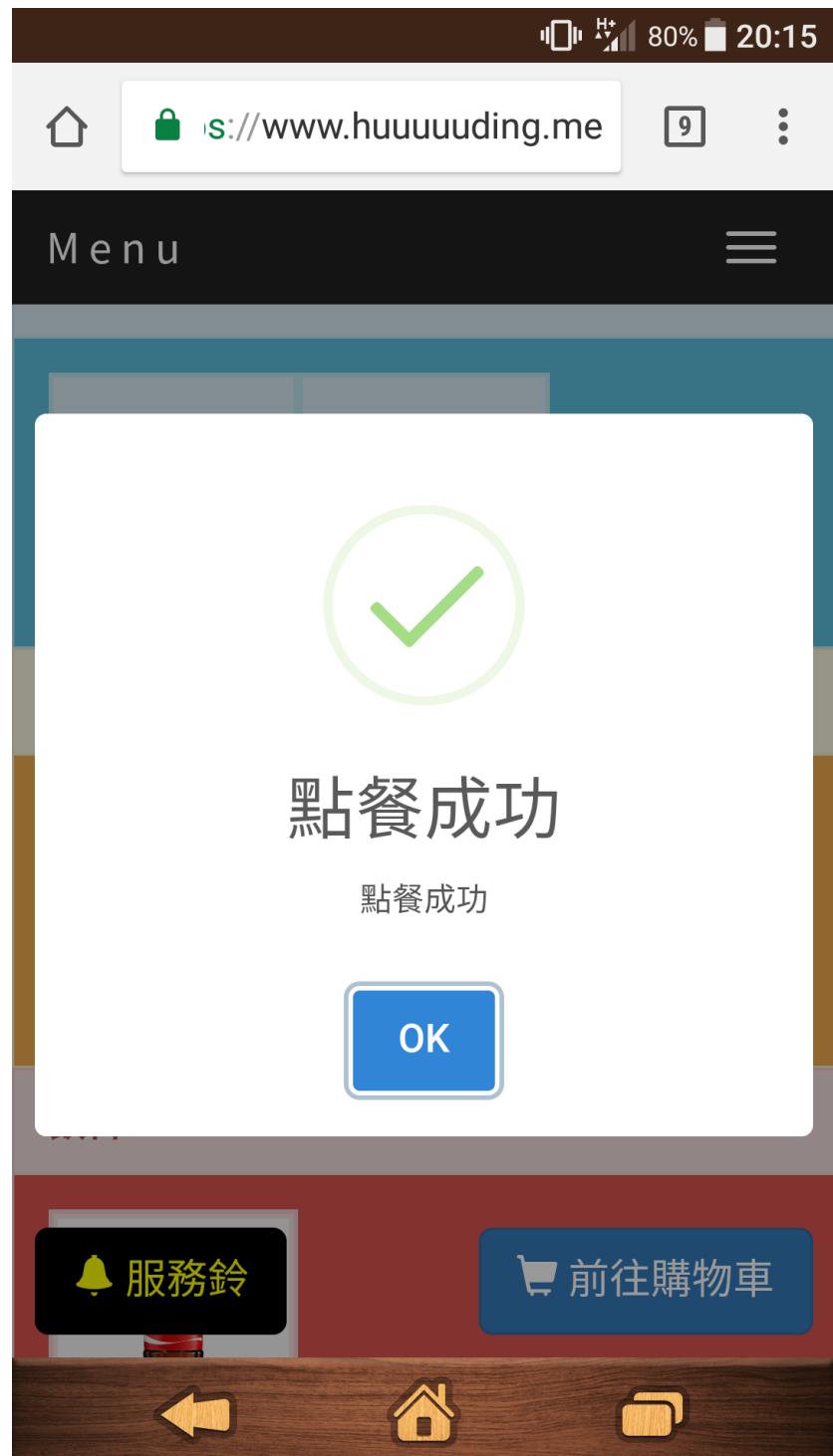


圖 3-34 按下點餐後之畫面

點餐成功。



圖 3-35 點餐成功後之畫面

點餐成功後，即會跳至消費明細，可以看到訂單的詳細資料，待上會隨著服務生上菜的進度減少。



圖 3-36 點完餐回到菜單之畫面

點完餐後，再回到菜單就無法再點餐，僅供預覽，前往購物車會變成消費明細。

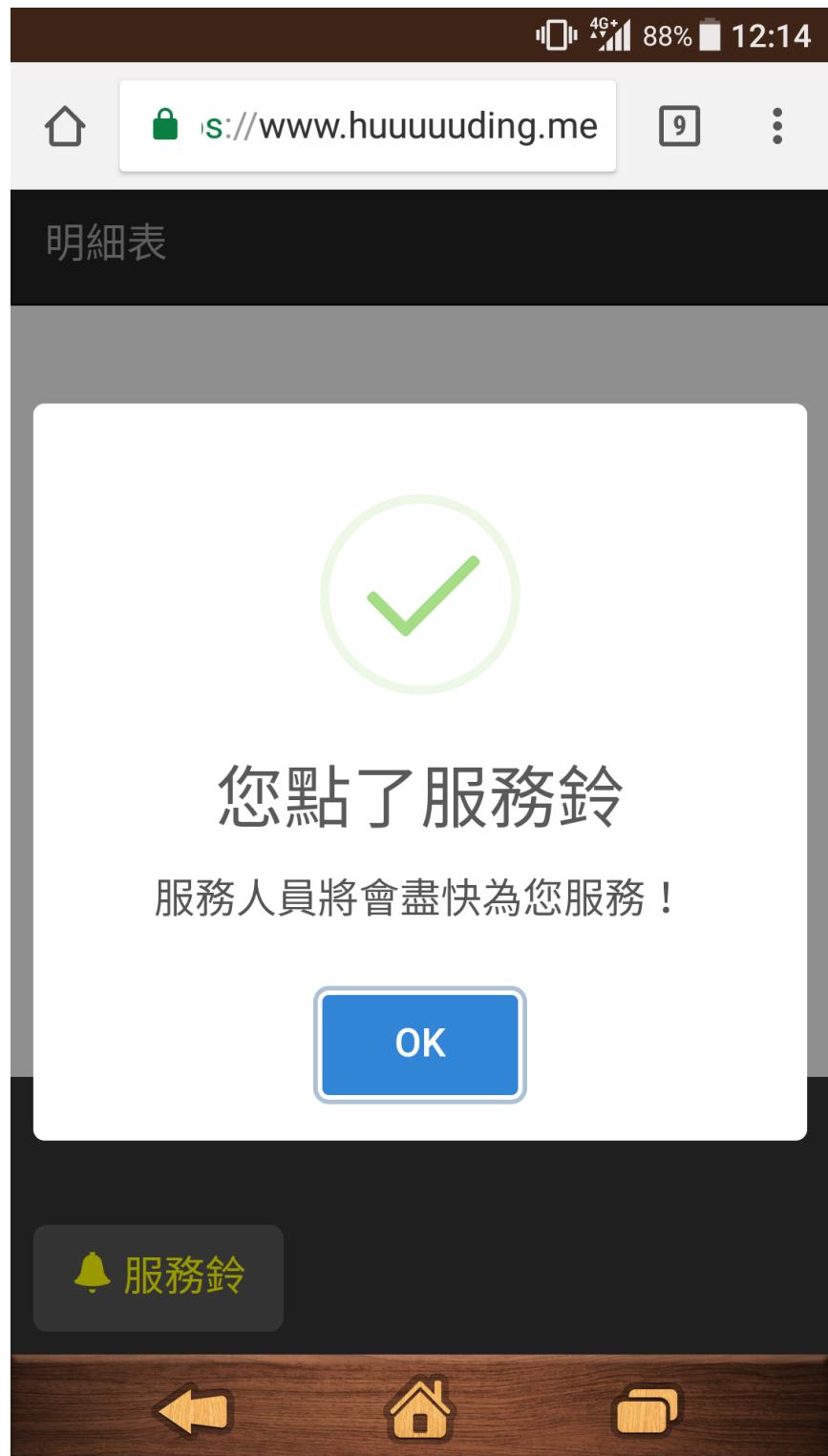


圖 3-37 按下服務鈴後之畫面

按下服務鈴，服務生會盡快服務該桌的顧客。

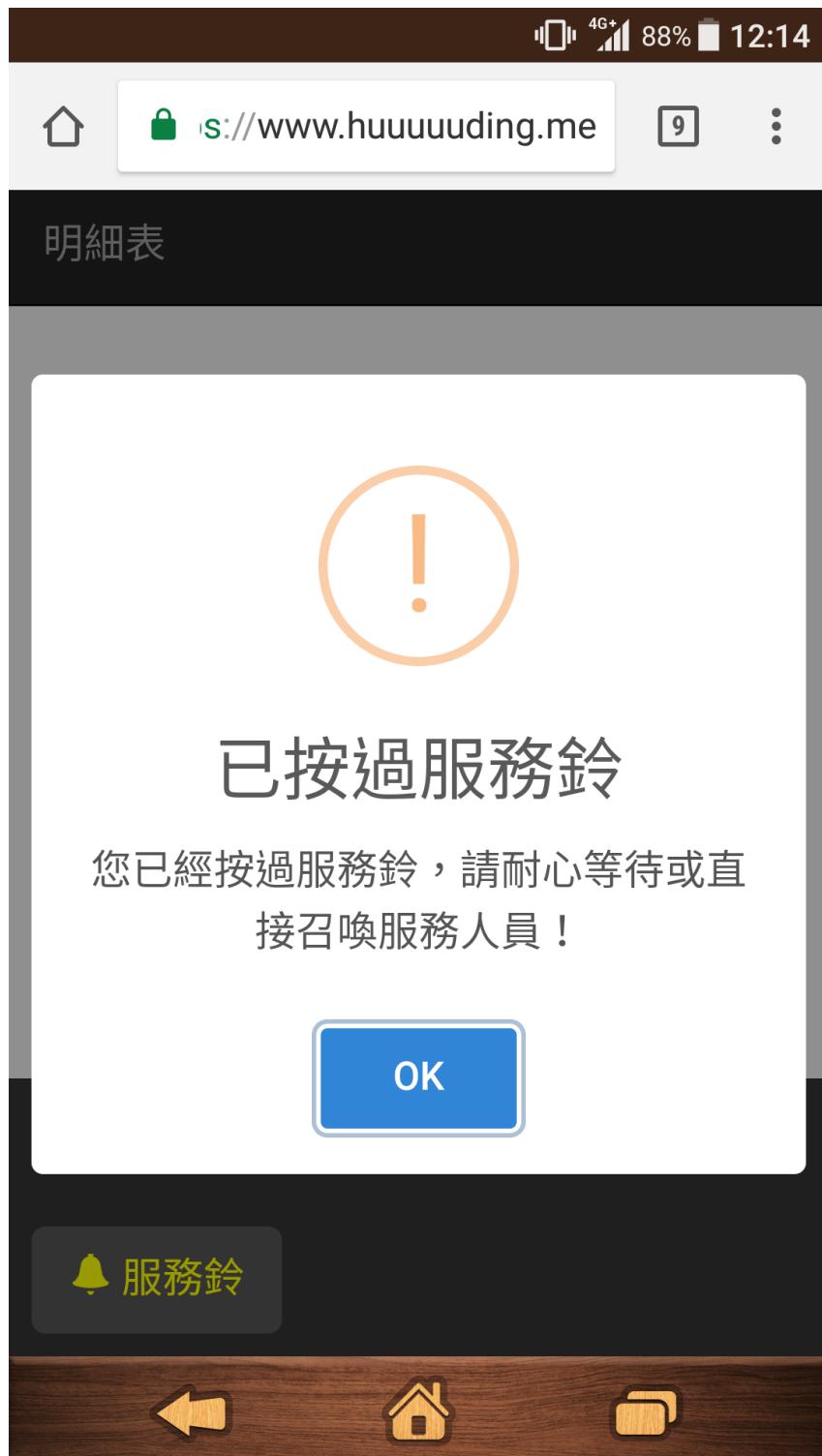


圖 3-38 再次按下服務鈴之畫面

在服務生還未到該桌服務，重複按下服務鈴不會有效果。

三、實作過程

組員：黃淯鼎

實作內容：server 架設、前後端資料傳輸、資料庫架構、後端程式

server 架設：

因為我們是一個網站，希望能越多人瀏覽越好。
所以我們需要一台主機、固定的 ip 和 domain name
因此就在 linode 租了一台 ubuntu 的 server，並且有一個固定的 ip，
而 domain 是使用 namecheap 所提供的服務，
後來因為考量到我們有登入的功能，
希望在傳輸過程中，帳密等重要資料有被加密傳輸，
所以還有申請 SSL 憑證。
後端的 HTTP server 我們選擇使用 apache，
後端的程式語言使用 python 的 django 這個框架來實作，
將上面那些全部整合在一起後，
就能透過 <https://www.huuuuuding.me>，來瀏覽到我們的網頁。

前後端資料傳輸：

很多地方使用者都需要讀取後端的資料來顯示，
也要將資料寫入後端來儲存某些資訊，
某些比較少的資料就使用 html 的 form 來傳遞，
但有時需要傳輸大筆資料時則是使用 ajax 來傳遞 json 格式的資料。

資料庫架構：

User:使用者

username:使用者帳號
password:使用者密碼

Restaurant:餐廳

name:餐廳名稱
phone:餐廳電話

address:餐廳地址

applicant:餐廳聯絡人(OneToOne User)

applicant_name:餐廳聯絡人姓名

applicant_phone:餐廳聯絡人手機

applicant_email:餐廳聯絡人 email

url:餐廳網址

logo:餐廳封面照片

employee:餐廳員工(OneToOne User)

Food:食物

name:食物名稱

price:食物價錢

intro:食物介紹

src:食物照片

category:食物分類

top10:是否為熱門

restaurant:此食物屬於哪間餐廳(ForeignKey Restaurant)

Order:訂單

orderId:訂單編號

creation_date:訂單日期

peopleNum:人數

tableNum:桌號

bell:是否有呼叫服務鈴

bellTime:呼叫服務鈴的時間

ordered:是否已點餐

restaurant:此訂單屬於哪間餐廳(ForeignKey Restaurant)

OrderItem:訂單內容

order:此訂單內容屬於哪個訂單(ForeignKey Order)

food:關聯到哪個食物(ForeignKey Food)

count:食物的數量

scount:食物待上數量

後端程式:

我們是按照下列的順序將頁面和功能一點一點的加上去，

1. 網站首頁
2. 餐廳老闆註冊、登入
3. 餐廳老闆登入後的設定頁面
 - 3.1 基本資料
 - 3.2 頁面設定
 - 3.3 製作菜單
 - 3.4 員工帳密
4. 每家餐廳的菜單頁面
 - 4.1 餐廳客戶點餐
 - 4.2 點餐後的明細資料
5. 員工登入後的頁面
 - 5.1 內場人員
 - 5.2 服務生
 - 5.3 櫃檯

組員：黃若望

實作內容：前端服務生介面設計

在一連串的發想與討論過程後，我負責的部分主要為服務生端的網頁前端設計。服務生端的網頁我起初規劃為三大功能區塊。
由上到下分別為：

一、帶位區：可供服務生選擇好人數以及桌號後，跳出一個 QR code 供客人可以進入我們的點餐頁面，這個 QR code 是獨特的，會讓我們知道是哪一桌的客人進行的點餐。（圖 3-1）

二、訂單列表：顯示所有桌號的客人訂單及其餐點的詳細資料，供服務生確認哪個餐點要送到哪一桌，上好菜後，可以按下“出餐”按鈕，讓整個系統同步知道此餐點已經解決。（圖 3-2）

三、服務鈴列表：會依順時間序列出所有按下服務鈴呼叫服務生的桌號，供服務生確認該去哪一桌進行服務，前往服務客人時可以按下“OK!”按鈕，讓整個系統同步知道此問題已經被某位服務員經手了。（圖 3-3）

為了實作以上的前uffix面我學習了 Bootstrap 3 以及 jQuery，最後所有的功能在我的階段，只使用到了 Bootstrap 3 的部分功能來實作，有些不合我們需求的樣式則自己 Override 自訂成為想要的樣子。最後由統整組員統整後，介面功能會依需求而被有所增減，此實作過程所寫的是我設計時的初衷。

顧客入座

人數

桌號

1

確認

圖 3-1 顧客入座

桌號 : 5 訂單	3
青醬義大利麵	出餐
紅醬義大利麵	出餐
白醬義大利麵	出餐
桌號 : 6 訂單	3
桌號 : 7 訂單	3

圖 3-2 訂單列表



圖 3-3 服務鈴呼叫

組員：李垣峻

實作內容：點餐系統的所有介面(首頁、註冊、登入、四個設定頁面、三種員工頁面、點餐頁面)

這次的專案，我負責前端，所有畫面跟組員討論過後，除了「三種員工頁面」有組員有幫忙之外，其他都是由我獨自完成，跟前面有一位組員所說的一樣，我是按照以下順序，去實作前端的。

1. 網站首頁
2. 餐廳老闆註冊、登入
3. 餐廳老闆登入後的設定頁面
 - 3.1 基本資料
 - 3.2 頁面設定
 - 3.3 製作菜單
 - 3.4 員工帳密
4. 每家餐廳的菜單頁面
 - 4.1 餐廳客戶點餐
 - 4.2 點餐後的明細資料
5. 員工登入後的頁面
 - 5.1 內場人員
 - 5.2 服務生
 - 5.3 櫃檯

其中最花時間的是 3.3 製作菜單這一項，幾乎花了我們整個專案 50% 的時間來製作，我們希望可以提供一個給餐廳端使用的介面，讓他們可以自己製作菜單，所以我在畫面(圖 3-4)上放上了兩個 iframe，左邊的是給餐廳製作菜單，右邊的是按下「預覽」之後，就可以看到製作後的樣子，由於要跟後端搭配，所以按下儲存的時候，我們發送一個 json 給後端，這些都是我以前沒碰過的，所以在實作上花了很多時間去學習。

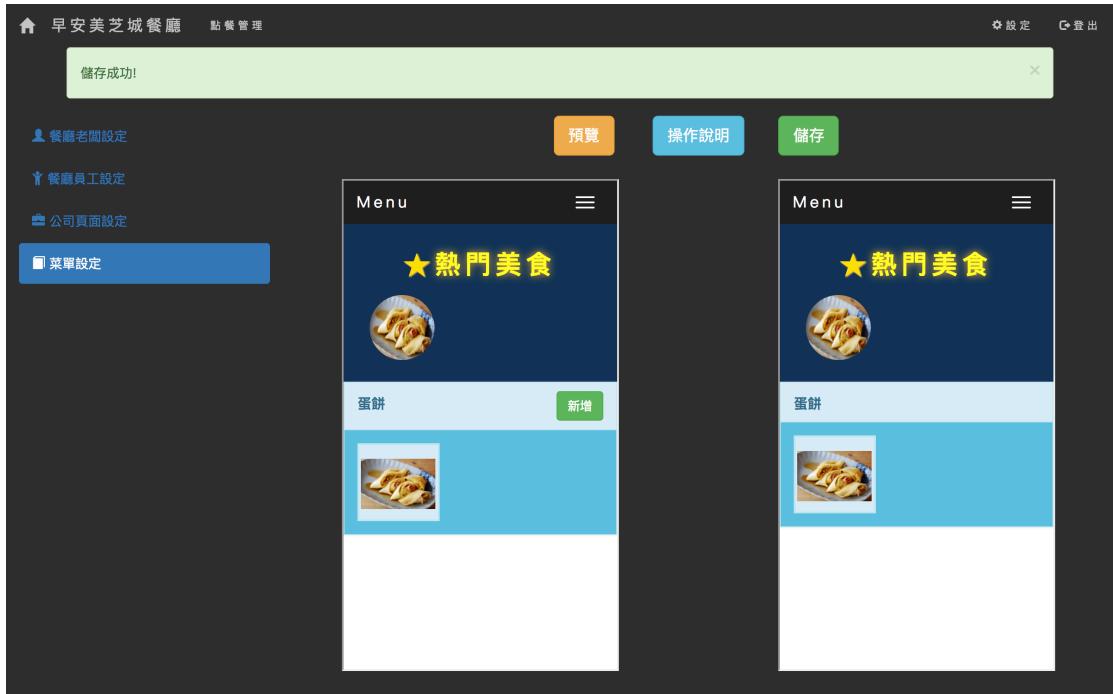


圖 3-4 菜單設定介面

4、5 兩大項也很花時間，我跟負責後端的組員，一直在想如何即時的更新頁面，如果顧客點餐之後，內場人員、服務生、跟櫃台應該都要收到一筆訂單明細，如果還要自己手動重新整理就太不方便了，但是我們又怕伺服器負擔過大，所以最後我們決定某些地方 5 秒重整一次，某些地方 10 秒重整一次，如此一來顧客跟餐廳都可以知道，還有幾份餐點還沒上，我覺得思考這些事情非常有趣，也很享受這過程。

5.1~5.3 這三個介面，我有參考另一個組員的設計，然後搭配上我自己的想法，原本是打算做在同一個介面，後來我們分成了三個介面，這樣對員工來說比較方便，不會受到其中不相干的資訊干擾，其中比較有趣的是 QRcode 的產生，我們是直接透過 google 的 api 完成的，只要在後面加 .qr，google 就會給我們一張 QRcode，非常的方便。

另外我將服務生頁面中的服務鈴，改成只有桌號沒有時間，服務鈴會按照顧客按的先後進行排列，我認為服務生只要知道哪一桌先按的即可，不太需要知道確切的時間，另外服務生頁面中的訂單列表，我也加了三個按鈕(新增、減少、刪除)，如果服務生送完一道餐點，就可以按減少，這樣顧客跟內場人員都會知道還有幾份餐點還沒做，新增按鈕是怕服務生誤觸減少按鈕，可以再加回來，刪除按鈕則是如果客人突然不要某道菜了，就可以按刪除按鈕。

組員：林煜宸

實作內容：協助前/後端、測試、報告撰寫

多次的開會後，我是負責幫忙前/後端，協助後端同學完成比較繁瑣的功能，以及幫忙後端的同學跟前端溝通，後端的部分，我們考慮到了許多細節，像是防範常見的攻擊手法，有些可能後端的同學沒有注意到的地方，我就會稍微提醒他。

前端的部分，像是版面的編排，畫面的美觀，介面的實用性...等，老闆登入後的餐廳管理的頁面都有參與版面討論，其中菜單設定的頁面的發想是因為在某個教學程式語言的平台，有著類似的頁面，覺得用的蠻順手的，就跟組員一起套論這個頁面的製作，最後的成品我們也很滿意，美觀又有實用性。

測試的部分，也需要想一些可能發生錯誤的輸入，測試一些常見的攻擊手法，或是有沒有顯示不合理的地方，另外畫面的美觀以及實用性，也透過我們反覆的討論才有現今的頁面，當初在預設哪些畫面要用手機瀏覽哪些畫面要用電腦瀏覽也是花了我們一番苦心討論，各個組員各司其職，最後才有這麼完整的作品。

四、學習心得

組員：黃清鼎

這次的專題我主要是負責後端程式的開發，所以也發現跟前端的合作非常重要，前端只要先將頁面的顯示方式先做好，我就只需要專心處理後端的功能就好，在前後端資料傳輸的部分，也只要先將資料的格式先定好，前後端就能平行的開發，也加快了我們專題的開發速度。

做這個專題用到了大學三年來學習到的各種知識，雖然過程中遇到了很多沒學過的東西，也遇到很多一開始沒有想到的問題，但後來也都一一的被我們解決了，因此也因為做專題學到了更多其他的知識。

組員：黃若望

網站的設計以及實作一直是我甚少去觸及的領域，由於這次的專題，我試著去了解它的架構、運作原理以及設計方針，學到了非常多以往完全不了解的東西。我相信將來網頁的發展會是很重要的領域，隨著網路速度以及電腦效能的提升，越來越多的東西會移植到網頁上，趁著這次專題有機會提升自己在網頁上的能力我獲益良多。

專題初期的發想階段，我認為非常的有趣，組員跟組員間的意見常常會擦撞出意想不到的火花，有些東西你有想到我沒想到，有些東西你想的不完備我來幫你補足，人類之可貴果然在於合作，與組員的良性溝通才能造就一人無法獨自創造的厲害專案。

撰寫網頁程式過程，我也確實遇到了好幾個程式上的問題，不過多虧在大學這四年的訓練，大多都可以經由網路上的專業論壇找到解答，提升自己的寫程式效率，站在廣大網友的肩膀上，創造出屬於自己的程式，這種遇到問題經過探索，而後解決問題我認為也是程式設計最迷人的地方，非常的有成就感，希望這次專題的經驗，可以讓我往後實現各種專案想法時更加的得心應手。

組員：李垣峻

這次的專題是我從以前到現在，寫過最大的一個專案了，平常教授出的小作業，跟這個專題比起來，簡直是小巫見大巫，從這次的專案我學到非常多關於前端的知識，原本就對於網站的前端有很深的熱愛，但一直只做過一些小作品，如

今其他組員替我圓夢了，讓我有這個機會在網站前端的領域，學得更深更廣，我很謝謝他們。

在這次的專案中，我們一開始先用 google 文件討論好整個網站的架構，以前的小作業都是直接寫，但是像這種大規模的專案，如果不事先討論，會走很多冤枉路，討論好架構之後，我負責前端，另一個組員負責後端，而其他兩個組員幫忙我們，我特別感謝做後端的那位組員，他總是可以很精準地告訴我，他需要的東西是什麼，這樣我前端就可以先做，然後有更多時間可以做得更好，因為有規格了，所以之後也不用怕做出來的東西不是他要的，我很喜歡這種團隊溝通。

最後想感謝組員配合我，選擇這個主題，有了這次的經驗，我覺得我對前端更加有興趣了，這次的專案我覺得比任何一堂課收穫的都更多，有種自己是教授同時也是學生的感覺，自己出作業給自己做，學得非常踏實跟滿足。

組員：林煜宸

網站一直是我非常感興趣的主題，一開始我們討論了很久，在偶然一次吃中餐時，發想了這個主題，起初我們討論了很久，一直沒辦法取得細節的共識，溝通的過程中，也有因為表達的不清楚或是描述的不夠具體，曾經有誤會有和爭執，在多次的開會後，才終於可以著手實作。我想除了程式能力外，語言的表達也是很重要，學習如何清楚的表達自己的想法讓別人知道，也是很重要的一環，才不會有不必要的誤會。

實作時，因為我的能力不如其他組員，所以我負責傳遞資訊以及前後端溝通的橋樑，和後端的幫忙，不懂的地方就問負責後端的組員，也讓我學習到很多，也因為這個專題碰了很多之前沒遇過的東西，但縱使遇到這些困難我們也都沒有輕易放棄，依然完成此次的專題，在此要謝謝所有組員的幫忙以及指導教授的提點，沒有他們，這個專題就不會完成。

五、測試結果

測試方法：

測試分為老闆、員工、顧客三個部分。

1. 老闆：註冊一個新帳號，登入之後是否有設定頁面、設定頁面功能是否正常、是否可以選擇三種員工身份。
2. 員工：使用老闆提供的密碼，登入之後是否可選擇三種身份，三種身份的功能是否有異常。
3. 顧客：是否能正常點餐、確認點餐、餐點明細是否有誤、服務鈴有無異常。

結果測試表：

餐廳端：

(1) 註冊：

使用設備	測試項目	測試結果
電腦	尚有欄位空缺	註冊失敗
電腦	欄位填寫正確	註冊成功
手機	尚有欄位空缺	註冊失敗
手機	欄位填寫正確	註冊成功

(2) 登入：

使用設備	登入身份	測試項目	測試結果
電腦	老闆	帳密正確	登入成功
電腦	老闆	帳密錯誤	登入失敗
電腦	員工	帳密正確	登入成功
電腦	員工	帳密錯誤	登入失敗
手機	老闆	帳密正確	登入成功
手機	老闆	帳密錯誤	登入失敗
手機	員工	帳密正確	登入成功
手機	員工	帳密錯誤	登入失敗

(3) 設定頁面

使用設備	身份	測試項目	測試結果
電腦	老闆	是否可以看到此頁面	是
手機	老闆	是否可以看到此頁面	是，但是畫面跑版。
電腦	員工	是否可以看到此頁面	否
手機	員工	是否可以看到此頁面	否

(4) 設定頁面——餐廳老闆設定：

使用設備	身份	測試項目	測試結果
電腦	老闆	欄位填寫正確	更改密碼成功
電腦	老闆	欄位填寫錯誤	更改密碼失敗
手機	老闆	欄位填寫正確	更改密碼成功，但是畫面跑版。
手機	老闆	欄位填寫錯誤	更改密碼失敗，畫面跑版。

(5) 設定頁面——餐廳員工設定：

使用設備	身份	測試項目	測試結果
電腦	老闆	欄位填寫正確	設定員工密碼成功
電腦	老闆	欄位填寫錯誤	設定員工密碼失敗
手機	老闆	欄位填寫正確	設定員工密碼成功，但是畫面跑版。
手機	老闆	欄位填寫錯誤	設定員工密碼失敗，畫面跑版。

(6) 設定頁面——公司頁面設定：

使用設備	身份	測試項目	測試結果
電腦	老闆	餐廳網址填寫正確 (英文、數字、底線 或連字符)且不重複	設定餐廳網址成功
電腦	老闆	餐廳網址填寫錯誤	設定餐廳網址失敗
電腦	老闆	公司封面照格式正 確(jpg/png/gif 且 大小<10MB)	設定公司封面照成功
電腦	老闆	公司封面照格式錯 誤	設定公司封面照失敗
手機	老闆	餐廳網址填寫正確 (英文、數字、底線 或連字符)且不重複	設定餐廳網址成功，但 跑版。
手機	老闆	餐廳網址填寫錯誤	設定餐廳網址失敗，且 跑版。
手機	老闆	公司封面照格式正 確(jpg/png/gif 且 大小<10MB)	設定公司封面照成功， 但跑版。
手機	老闆	公司封面照格式錯 誤	設定公司封面照失敗， 且跑版。

(7) 設定頁面——菜單設定：

使用設備	身份	測試項目	測試結果
電腦	老闆	預覽	成功預覽
電腦	老闆	儲存	成功儲存
電腦	老闆	操作說明	彈出操作說明窗口
電腦	老闆	新增、編輯、刪除餐點分類	成功新增、編輯、刪除
電腦	老闆	新增、編輯、刪除餐點	成功新增、編輯、刪除
電腦	老闆	設定、移除熱門美食	成功設定、移除
手機	老闆	預覽	成功預覽，但跑版。
手機	老闆	儲存	成功儲存，但跑版。
手機	老闆	操作說明	彈出操作說明窗口，但跑版。
手機	老闆	新增、編輯、刪除餐點分類	成功新增、編輯、刪除，但跑版。
手機	老闆	新增、編輯、刪除餐點	成功新增、編輯、刪除，但跑版。
手機	老闆	設定、移除熱門美食	成功設定、移除，但跑版。

(8) 點餐管理——服務生：

使用設備	身份	測試項目	測試結果
電腦、手機	老闆、員工	顧客按服務鈴，服務鈴列表是否有新增	有新增
電腦、手機	老闆、員工	顧客點餐，訂單列表是否有新增	有新增
電腦、手機	老闆、員工	增加、減少、刪除餐點	成功增加、減少、刪除

(9) 點餐管理——櫃檯：

使用設備	身份	測試項目	測試結果
電腦、手機	老闆、員工	人數桌號正確填寫，是否能產生 QR code	產生成功
電腦、手機	老闆、員工	人數桌號填寫不完全，是否能產生 QR code	產生失敗
電腦、手機	老闆、員工	顧客點餐，訂單列表是否有新增	有新增
電腦、手機	老闆、員工	刪除訂單列表	成功刪除

(10) 點餐管理——內場人員：

使用設備	身份	測試項目	測試結果
電腦、手機	老闆、員工	顧客點餐，訂單列表是否有新增	有新增

顧客：

使用設備	身份	測試項目	測試結果
電腦、手機	顧客	掃 QR code 之後，能否進入點餐頁面	成功進入
電腦、手機	顧客	服務鈴是否可以使用	可以使用
電腦、手機	顧客	餐點是否可以加入購物車	可以加入
電腦、手機	顧客	購物車裡的餐點是否可以增加、減少、刪除	可以增加、減少、刪除。
電腦、手機	顧客	餐點明細是否正確	完全正確
電腦、手機	顧客	同一個 QR code 是否可以重複點餐	否，只會看到瀏覽頁面。

六、結論

經過數個月的努力，前端跟後端之間不斷地溝通協調，點餐系統終於順利地按照我們希望的模樣產生出來了，除了研究動機所提到的，我們還加了讓餐廳製作菜單的功能，因為我們認為，每個店家所希望的菜單模樣可能不盡相同，另外我們也多了掃描 QR code 的功能，只有掃 QR code 的點餐會被認可，如此一來即可防範有意人士的惡意點餐。

我們也測試過許多情況，除了基本使用者可能會犯的操作錯誤之外，我們後端使用 django，它本身就有很多防止被攻擊的機制，操作的流程我們盡量簡化，也盡量多給一些使用者提示，期望可以讓使用者操作的方便又安心。

七、未來發展

之後想以全自動的點餐系統作為目標，因為我們跟硬體不太熟，所以這次只有先挑戰軟體的部分，如果結合 post 機，就可以做到顧客點完餐，post 機就把印出來了，這樣就不需要櫃檯了，點餐時服務生會直接收錢，然後把發票給顧客。

另外也想把介面做得更加漂亮，並且提供更多的模板，這樣餐廳才可以選到最適合自己的菜單，資安部分也有不完善的地方，希望可以防得更仔細一點。

此外，大數據分析的部分，也是我們之後想完成的，想要分析使用者提供的數據，然後預測使用者可能會喜歡的餐點為何，期望可以增加餐廳的銷售量。

八、參考文獻

[1] linode document

網址：<https://linode.com/docs/>

[2] namecheap doamin

網址：

<https://www.namecheap.com/support/knowledgebase/category.aspx/34/domains>

[3] namecheap ssl

網址：

<https://www.namecheap.com/support/knowledgebase/category.aspx/14/ssl-certificates>

[4] django document

網址：<https://docs.djangoproject.com/en/1.11/>

[5] JQuery AJAX

網址：<http://api.jquery.com/jquery.ajax/>

[6] 專案架構

網址：

<https://stackoverflow.com/questions/22841764/best-practice-for-django-project-working-directory-structure>

[7] django 實例網站

網址：<https://github.com/vitorfs/parsifal>

[8] google 縮址 API

網址：https://developers.google.com/url-shortener/v1/getting_started

[9] 偵測 iframe 中的 click 事件

網址：

<https://stackoverflow.com/questions/13439303/detect-click-event-inside-iframe>

[10] 讀取 iframe 裡的變數

網址：

<https://stackoverflow.com/questions/13757943/access-a-variable-of-iframe-from-parent>

[11] 設置 iframe 裡的變數

網址：

<https://stackoverflow.com/questions/1301540/set-variable-in-parent-window-from-iframe>

[12] SweetAlert2

網址：<https://limonte.github.io/sweetalert2/>

[13] CSS Tutorial

網址：<https://getbootstrap.com/docs/3.3/css/>

[14] CSS 關於 position

網址：<http://zh-tw.learnlayout.com/position.html>

[15] Bootsnipp

網址：<https://bootsnipp.com/tags/registration?page=1>

[16] JavaScript 取得畫面大小指令

網址：https://dotblogs.com.tw/jenny_ming/2013/06/06/105333

[17] JQuery 获得绝对、相对位置的坐标方法

網址：<http://www.jb51.net/article/22081.htm>

[18] Bootstrap3 Tutorial

網址：<https://www.w3schools.com/bootstrap/default.asp>

[19] Bootstrap Document

網址：<https://v4-alpha.getbootstrap.com/getting-started/introduction/>

[20] Smooth touch scroll

網址：<http://smoothtouchscroll.com/>