

路邊攤行政助理

Administrative Assistant Of Stall

指導老師：韓欽銓 教授

學生：朱宏文、楊仁允、莊家瑋、林顥圃、蔡紘睿

國立聯合大學 資訊工程學系

苗栗市恭敬里聯大一號

nuu.csie.404@gmail.com

摘要

一個讓老闆專注美食料理、消費者輕鬆訂餐的環境，從訂餐、訂單傳送、料理餐點排程到領取餐點，「路邊攤行政助理」完全搞定。老闆節省行政人力成本、減少負擔，建立消費者快速且安心的點餐環境。

消費者利用 APP 選餐後，送出訂單，立即用手機支付金額，系統自動將取餐的 QR 碼和取餐編號，一併傳送給消費者，而訂單內容則會顯示在餐廳的觸控螢幕上，老闆依據螢幕上的訂單順序料理餐點，完成後放入系統提示之取餐櫃。

當餐點完成後，系統亦會送訊息通知消費者並取餐櫃號，消費者到現場後，掃描 QR 碼，系統進行身分比對，並打開對應的電子鎖櫃，讓消費者順利將餐點取走。

關鍵詞：行政助理、Raspberry Pi、APP、傳產再造。

Abstract

Using the APP costumer can order and pay meal with smart phone, than the system will automatic send QR code and order number back to costumer, while the order showed on stall screen. Shopkeeper will cook base on the order screen, and put inside cabinet after finish. The system will send notification and cabinet number to costumer, so costumer can go to stall, and scan out QR code. After that system will identify and

open the corresponding stall cabinet number, so costumer can take out and enjoy the delicious meal

Key word: Administrative Assistant、Raspberry Pi、APP、Reform Traditional Industry

一、前言

現代人生活步調快速，凡事講求快、狠、準，藉此，我們設計了一個方便快速的環境，讓消費者可迅速訂餐、取餐，老闆則只需要提供它本身的專業，製作餐點。

顧客可以在公司、工作崗位立即下訂單，並依據系統提示的取餐時間，至現場領取餐點，簡單明瞭的步驟提供顧客便利及快速的服務。

對於老闆來說，並不需要再大量依賴紙本菜單，一旦收到訂單訊息，即可立即下廚。

二、架構與流程

本系統為跨平台設計並整合物聯網的相關硬體設備，如圖 1 所示，其中包含客戶端的裝置，提供顧客透過手邊的裝置即可做到即時點餐與取餐通知，在這之間系統也包含電子櫃與樹梅派作為核心物流控制，讓使用者可以即時的在餐點完成時接收到通知，並且每項訂單都有各自的 QR code，避免取餐出現遺漏或誤取，此外這系統也使店家可以用減低人力花費，提升點餐效率。

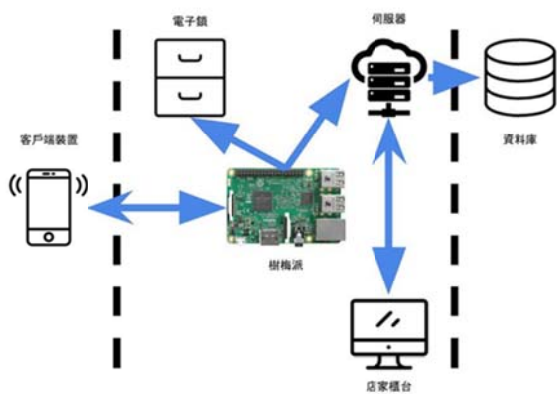


圖 1、系統架構圖

1. 首先，在訂餐方面顧客可以選擇使用手機端或是網頁來訂餐，於系統中選擇餐點及數量，並輸入電話號碼即可完成訂單。
2. 送出訂單後會將餐點資訊和電話號碼傳送給伺服器，伺服器配置一組亂碼給此訂單，供取餐時能準確打開對應電子櫃及核對顧客，以預防誤取餐點狀況發生。
3. 伺服器會將取餐資訊(取餐編號、訂單內容、QR code)發送至顧客手機端，且將訂餐資訊傳送至老闆的平板。
4. 老闆餐點完成後會點擊平板上的狀態，將狀態由未完成變更為已完成，並發送資訊通知顧客可以前來取餐，同時會對伺服器發出櫃號請求，伺服器會將閒置的櫃號回傳至老闆的平板，老闆即可將餐點放入提示的電子櫃中等待顧客取餐。
5. 顧客只需要帶著手機到店家，於 APP 中打開訂單資訊點選訂單並對著 QR code 掃描器進行掃描。伺服器讀取條碼後在資料庫搜尋相應櫃

號，通過 Node MCU 對電子櫃相應櫃號解鎖，顧客即可領取餐點，在餐點領取完畢將櫃子闔上時，餐點狀態會由已完成變更為已領走，以便老闆了解取餐狀況。

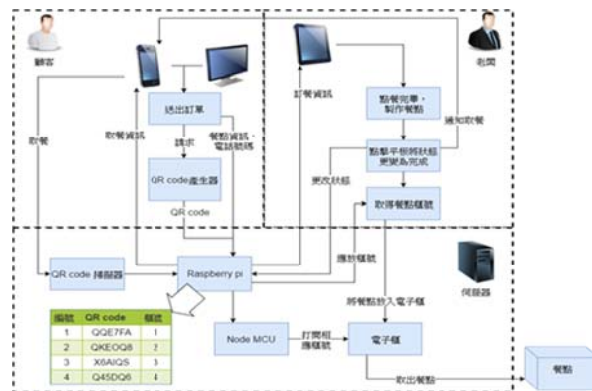


圖 2、流程圖

三、應用情境

3.1 消費者端

消費者選擇想要購買的餐點之後，將菜單送出，手機則立即支付餐點金額，系統藉由 QR code 產生器獨立生成一個 QR code，並將此 QR code 外加取餐時間和取餐編號，一併傳送給消費者，而菜單內容則傳送給老闆的觸控板上，提供老闆製作餐點。

消費者抵達現場後，使用收到的 QR code 於掃描區進行掃描，核對身分後，符合則依據取餐編號，領取餐點。

3.1.1 APP 導覽

為了避免消費者初次接觸此系統，產生的不適應、不熟悉，進而造成 APP 使用度下滑，我們製作了流程導覽，介紹 APP 內部功能、流程順序，也讓消費者省時省力，不須額外花費更多時間融入這個 APP。

3.1.2 線上預購

消費者選擇線上預購功能後，輸入手機號碼、姓氏，即可觀看菜單內容，填好數量後，按下確認鍵完成訂餐。

(1) APP



(a) (b)

圖 3、APP 線上預購畫面

(2) Web

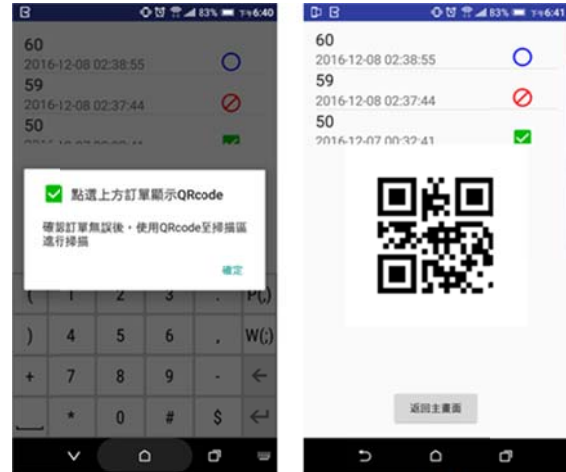


圖 4、網頁預購畫面

3.1.3 查詢訂單

當消費者欲解鎖電子櫃時，按下訂單資訊功能，輸入手機號碼後，即可觀看消費者的訂單清單，點擊訂單後，確認訂單內容無誤，按下確認便可取得 QR code。

(1) APP



(a) (b)

圖 5、APP 查詢訂單畫面

(2) Web



圖 6、網頁查詢訂單畫面

3.1.4 公告資訊

為了推廣此系統，以及讓使用者持續使用，推出活動、方案，促使消費者們回流率提高，也利用推播功能，主動與消費者互動。

3.1.5 意見回饋

依據店家、顧客所回饋的意見，提升整體系統的穩定性，甚至解決錯誤、危障，提高軟體使用者的滿意度。

3.2 供應者端

一間餐館最重要的莫過於老闆，而在需求鏈上，老闆必然是擔任供應者的部分。為了讓老闆專心製作美味佳餚，我們將有關於料理之外的事物一手包辦，像是結帳、訂餐，儼然是一個全能的行政助理，不只節省了人力，也省下紙本訂單上的成本。在消費者送出訂單之後，系統會將菜單自動傳送給老闆，老闆則依據觸控板上的菜

單依序製作餐點，完成後放入取餐區中所對應之取餐編號位置。

3.2.1 新增菜色

當老闆研發新菜色後，想要新增到菜單上只需要輸入菜名及價錢，若已經拍好圖片可以選擇檔案上傳，也可以使用 Webcam 在網頁上直接進行拍攝，完成上述動作後按下儲存，即可將這道菜存入資料庫。

3.2.2 編輯菜單

當老闆需要調整價錢，或者想要移除菜單中的菜色，只要點擊清單上每到菜後面的修改或刪除按鈕，方可進行上述動作。

3.2.3 查看訂單

此頁面利用 javascript 執行 ajax 即時更新訂單清單，當老闆完成訂單後，點擊按鈕便可改變訂單狀態，並傳送推播提醒顧客取餐與告知取餐櫃號。



時間	取餐編號	內容	電話	狀態
2016/10/12 12:10:00	6	肉羹麵*2	0987123512	取餐中
2016/10/12 12:05:12	5	水晶餃湯*1, 燙青菜*1	0912545311	取餐中
2016/10/12 11:58:11	4	豬腳麵*1, 肉羹*1	0912344674	取餐中
2016/10/12 11:56:33	3	飽飯*2, 紅豆湯*1	0912342221	取餐中
2016/10/12 11:52:43	2	魯肉飯*1, 粟米湯*1	0912354323	取餐中
2016/10/12 11:50:00	1	排骨麵*2, 水晶餃湯*1, 肉羹*1	0912345678	取餐中

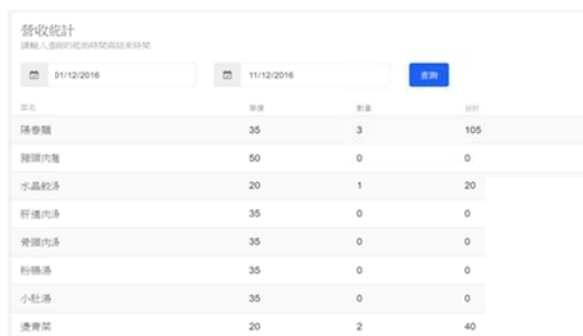
圖 7、網頁查看訂單畫面

3.2.4 公告資訊

當老闆想要增加顧客舉辦活動，增加顧客的回流率，輸入公告標題、公告內容、及截止日期，按下送出便可傳送推播通知到 APP。

3.2.5 營收統計

老闆選擇起始日期與結束日期後，按下確認鍵，系統便會幫老闆統計出每道菜在選擇時間內所賣出的數量，及算出時間內的營業額。



品名	數量	單價	合計
排骨麵	35	3	105
豬腳肉羹	50	0	0
水晶餃湯	20	1	20
野蔬肉湯	35	0	0
魯肉肉湯	35	0	0
粉麵湯	35	0	0
小肚湯	35	0	0
燙青菜	20	2	40

圖 8、網頁營收統計畫面

3.3 伺服器端

透過此助理系統，老闆可以大幅度降低人力成本，而且老闆只需要製作餐點，也大量減少老闆的負擔；消費者可以加速取餐，節省許多等待餐點完成的時間。伺服器負責接收所有的資料，消費者的部分，像是菜單內容、問題回報；老闆的部分，例如：新增、刪減或編輯菜單，接收到資料後，依據資料內容做出相符動作，像是：接收到菜單後，將菜單傳給老闆，消費者則是收到二維條碼、取餐時間以及編號，而在老闆按下餐點完成時，推播一則訊息給消費者，提醒其該餐點已完成。

3.3.1 分配櫃號

先取得所有櫃子總數，利用 for 迴圈由小到大至資料庫查詢最新狀態，如果沒有被使用過，或者狀態是以領走，則回傳該櫃號，如果 for 迴圈跑完依然沒有回傳代表沒有空的櫃子，則回傳 null。

3.3.2 打開電子櫃

當伺服器接收 QR code 內的資訊後，至資料庫搜尋該訂單資料，當訂單狀態為已完成時，將櫃號藉由 GET 傳送到 NodeMCU，NodeMCU 發送正訊號給繼電器將們打開

3.3.3 傳送推播

伺服器將推播的標題、內容及對象傳送至 Google API “Firebase Cloud Messaging”，透過 google 伺服器傳送訊息，手機可以即時收到通知，以預防遺漏店家

傳來的訊息。

四、結論

由於我們在本次專題中選擇了一般民眾平常就會使用到的應用程式，所以最複雜的地方在於使用流程的討論以及 UI 介面整體的流暢度，讓使用者因為便利性而下意識的熟悉新操作方式，進而推廣給更多其他還未接觸或是還在使用傳統方式的使用者，因為我們希望本系統能夠成為一般小吃店店家與購買民眾交易方式中的一項核心，達到方便迅速有效率且能將爭議減至最低的交易方式。

我們不希望這套系統只是用於專題、比賽，我們更希望我們所做出來的東西，可以用得長久、安心，因應社會的變遷，生活步調的轉變，我們盡了一點心力，終究希望能為這個社會付出，雖然我們擁有整理程式碼並使其完成任務的能力，但若要是給其他使用者使用的話就必須想到更多的方面，我想這將是我們在未來也必須面對的非常大的一個挑戰吧。

五、參考文獻

- [1] Raspberry Pi 最佳入門與實戰應用(第二版) 作者：柯博文
出版社：碁峰資訊
- [2] Android 程式設計入門、應用到精通-第四版 作者：孫宏明
出版社：碁峰資訊
- [3] Java Programming for Android Developers for Dummies
作者：Burd, Barry 出版社：For Dummies
- [4] Java Performance Companion
作者：Hunt, Charlie , Beckwith, Monica/ Parhar, Poonam/ Rutisson, Bengt 出版社：Addison-Wesley Professional
- [5] PHP 程式設計 第三版 作者：Kevin Tatroe 譯者：林耕溥
出版社：歐萊禮