

國立彰化師範大學電子工程學系

實務專題報告

智能冰箱之食品管控雲端資料庫應用

Refrigerator Foods Management Based On Cloud Database

指導教授：黃其泮

作者：柯竣鑫

民國一百一十年六月

## **Abstract**

As technology advances, some people became aware the problem of power consumption in the refrigerator. The longer we open the refrigerator door, the more power we consume. If we can reduce the time to find foods, it will improve the problem of power consumption. Therefore, we have proposed a design to control the storage of foods through the application of labelled foods and cloud database. This can directly find the foods on the panel, reduce energy waste significantly.

## 中文摘要

近年來隨著資源的消耗增加，人們對能源的意識也逐漸重視。

在家庭用電方面，冰箱與冷氣的耗電量往往占了極大部分在每個人的月平均用電量上，而冰箱會有如此龐大的耗電量，往往是由於人為的疏失所導致。每當我們在開啟冰箱時，若是開啟時間過長冷氣容易流失到空氣中，造成不必要的耗能。若能在開啟冰箱前預先知道冰箱內的食物，可以大幅度減少冰箱門開啟時間，從而達到節能減碳的效果。

本專題秉持節約能源的精神，從而使用無線射頻技術(radio frequency identification, RFID)和物聯網技術(internet of things, IoT)設計出一套完整系統，結合資料庫與互聯網的應用，從而達到智能管控冰箱中食品。

# 目錄

Abstract.....	i
中文摘要.....	ii
目錄.....	iii
圖目錄.....	iv
第一章 介紹.....	1
1.1 研究動機與目的.....	1
1.2 研究背景.....	1
1.3 研究方法.....	2
1.4 全文概述.....	2
第二章 技術簡介.....	3
2.1 RFID .....	3
2.2 NodeMCU-V3.....	3
2.3 XAMPP .....	4
2.4 MySQL .....	4
2.5 phpMyAdmin.....	5
2.6 html .....	5
2.7 php.....	5
2.8 Bootstrap .....	5
第三章 研究過程.....	6
3.1 實體架構.....	6
3.1.1 讀取標籤的UID .....	7
3.1.2 上傳UID至終端 .....	8
3.2 網頁架構.....	9
3.2.1 接收UID .....	10
3.2.2 建立網頁基本架構.....	10
3.2.3 建立MySQL資料庫 .....	11
3.2.4 建立資料庫頁面.....	12
3.2.5 建立新增頁面.....	14
3.2.6 建立讀取頁面.....	15
第四章 成果展示.....	16
4.1 實體展示.....	16
4.2 網頁展示.....	17
第五章 結論.....	21
參考文獻.....	22

## 圖目錄

圖(1) rfid原理圖。	3
圖(2) Xampp Control Panel。	4
圖(3) MySQL資料庫。	4
圖(4) 實體架構圖。	6
圖(5) SPI架構。	7
圖(6) Arduino IDE流程圖。	8
圖(7) 網頁架構圖。	9
圖(8) example.php。	10
圖(9) MySQL資料庫欄位設定。	11
圖(10) MySQL資料庫內容。	11
圖(11) database.php。	12
圖(12) user data.php。	13
圖(13) user data edit page.php。	13
圖(14) registrantion.php。	14
圖(15) 狀態圖。	15
圖(16) red tag.php。	15
圖(17) 路由器 & 實體裝置。	16
圖(18) 資訊面板。	16
圖(19) 主畫面。	17
圖(20) 資料庫。	17
圖(21) 編輯資料庫資料。	18
圖(22) 刪除資料庫資料。	18
圖(23) 新增標籤內容。	19
圖(24) 讀取標籤資料。	19
圖(25) 關於(上半)。	20
圖(26) 關於(下半)。	20

# 第一章 介紹

在現今的生活中，冰箱是我們生活的必需品，但若未規劃好冰箱內食品的儲藏，往往會造成資源的浪費，甚至會因為忘記了食品的保存期限，進而影響到自身健康。隨著科技的變遷，越來越多人開始對冰箱進行更多的開發，使冰箱不單單只是存放食品的地方，而能有更高效的管理方式，透過對食品的監控，達到保持食品的新鮮度及數量的控管，也是本專題主要研究的方向。

此章節主要介紹專題的研究動機與目的及相關系統之概述，方便了解本專題整體實驗架構。

## 1.1 研究動機與目的

一般人在採買食品時並未作好有效的規劃，很容易會忘記自己曾購入的種類及數量，一旦食品在冰箱放置過久就容易造成食品的腐敗及冰箱內清潔等問題。而隨意的堆積往往會造成在找尋食品的不便，這也會導致冰箱門打開的時間過長，造成冰箱需要花費更多的電能去回復到原先冷藏/冷凍的溫度，也是冰箱的耗電量龐大的主因，若是能設計出一套介面能讓使用者在外置面板上能清楚地得知食品的種類及數量，就能夠有效的解決此問題。

## 1.2 研究背景

隨著科技的進步，市面上的冰箱也有許多功能，有些冰箱皆設有感知功能，透過偵測冰箱內的亮度及容量，藉此調節冰箱壓縮機的功率，從而達到自動省電的效果。由於現今提倡綠色能源的發展，許多廠家開始重視這方面的研究，因此本專題提出一套方式，藉由無線射頻技術(radio frequency identification, RFID)和物聯網技術(internet of things, IoT)，將食品貼上專屬標籤，並將該 ID 記錄在雲端資料庫中，透過區域網路的連接，能夠讀取到主機端資料，從而達到即時監控冰箱食品的效果。

## 1.3 研究方法

本專題設計的裝置主要分成兩個部分，分為**實體裝置**部分及**雲端網頁**部分。

### **實體裝置：**

1. 在冰箱上方建立讀卡裝置，將標籤黏貼於保鮮盒上，通過讀卡機掃描，若此標籤未被新增置資料庫中，會提醒用戶註冊新資料，也提供建立資料的功能。
2. 能夠與雲端網頁進行連接的裝置，將讀卡機讀取的數據上傳至雲端上。

### **雲端網頁：**

1. 當有新的標籤讀取時，可輸入食品的名稱、種類、購買日期、保存期限。
2. 當用戶想知道先前購買的日期及相關資訊，可藉由介面讀取到此標籤中所含的資訊，方便用戶即時得知食品資訊。
3. 食品若被食用或丟棄可由介面進行刪除或編輯。

## 1.4 全文概述

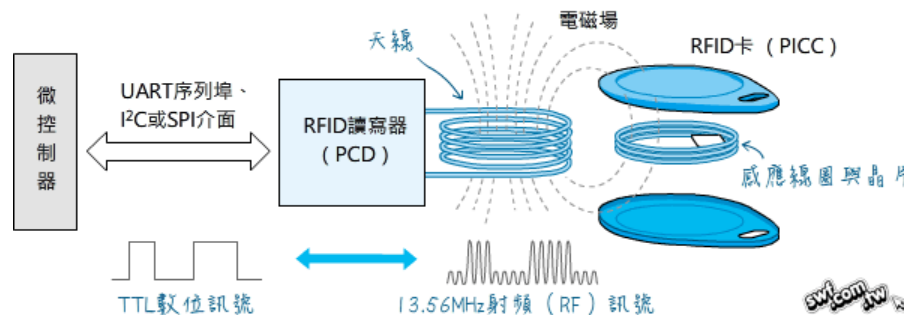
本專題第二章節的部分會提及相關硬體介紹及本次所使用到的軟體、語法等，並探討此技術的特色及同類型比較差異。第三章節的部分為本專題重點設計，由硬體層面的控制程式與雲端層面的網頁設計進行解說，第四章節則為實際運行的過程及系統網頁的展示。

## 第二章 技術簡介

此章節主要介紹本專題裝置所應用到的相關技術及軟體，第一、二小節為 RFID 與 nodemc-v3 屬於實體層面的技術，第三小節以後為 XAMPP、MySQL、php 等，屬於網頁技術層面，應用相關軟體進行網頁開發。

### 2.1 RFID

無線射頻辨識(radio frequency identification, RFID)是一種無線通訊技術,其利用電磁感應的方式，由讀卡機形成特定頻率的電磁場，使得標籤內天線產生電流驅動晶片工作，進而發出訊號溝通讀卡機，通過無線電訊識別特定目標並讀寫相關數據。與傳統條形碼(QR code)差異點在於無需建立機械或光學接觸，可在黑暗中進行讀取，大大增加使用的方便性。本專題採用的 RFID 讀卡機模組為 Mifare 公司所推出的 RFID-RC522，載波頻率為 13.56MHz，標籤類型為接近式 IC 卡 (Proximity IC Card, PIC)，屬於非接觸式 IC 卡。



圖(1) rfid 原理圖。

(圖源:《超圖解 Arduino 互動設計入門》第三版 第十七章)

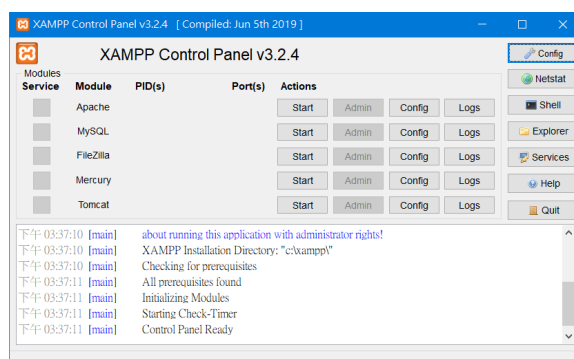
### 2.2 NodeMCU-V3

NodeMCU Lua 是一款原型開發板，集成 ESP8266 WIFI 芯片，擁有 GPIO、PWM、I2C、1-Wire、ADC 等等硬件 IO 功能，能夠像 Arduino 一樣操作硬件輸入輸出，用交互式 Lua 腳本編寫硬件代碼，能夠用 Node.js 類似的語法編寫網絡應用程式，豐富的事件驅動型 API，方便地讓用戶進行網絡應用開發，並運行於 5cm 大小的 MCU 上，是開發物聯網的極佳原型開發板。



## 2.3 XAMPP

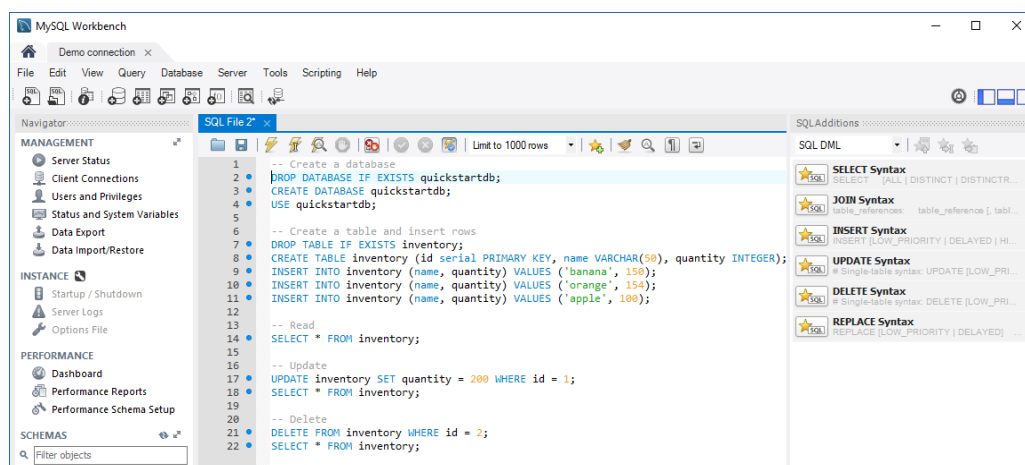
XAMPP(Apache+MySQL+PHP+PERL)是一個把 Apache 網頁伺服器與 PHP、Perl 及 MariaDB 集合在一起的安裝包，允許用戶可以在自己的電腦上輕易的建立網頁伺服器，可在 Windows、Linux、Solaris 等操作系統下安裝使用。



圖(2) Xampp Control Panel。

## 2.4 MySQL

MySQL 是一個開程式原始碼的小型關聯式資料庫管理系統，開發者為瑞典 MySQL AB 公司。目前 MySQL 被廣泛地應用在 Internet 上的中小型網站中。由於其體積小、速度快、總體成本低，尤其在開放開源碼方面，許多中小型網站為了降低網站總體成本，選擇使用 MySQL 作為網站資料庫，也是本專題選用的主因之一。



圖(3) MySQL 資料庫。

## 2.5 phpMyAdmin

phpMyAdmin 是一個以 PHP 為基礎,Web-base 方式架構在網站主機上的 MySQL 的資料管理工具,讓管理者可用 Web 面管理 MySQL 資料。藉此 Web 介面可以成為一個簡易方式輸入省略繁雜的 SQL 語法,尤其要處理大量資料的匯入及匯出更為方便。

## 2.6 html

HTML 是一種用於建立網頁的標準標示語言,常與 CSS、JavaScript 一起被眾多網站用於設計應用,而 HTML 描述了一個網站的結構語意隨著線索的呈現使之成為一種標記語言而非程式語言。而它跟一般的文書處理器不同的地方在於,它具有超文字(HyperText)、超連結(HyperLink)、超媒體(HyperMedia)的特性,透過 HTTP(HyperText Transfer Protocol)網路通訊協定,便能夠在世界各地透過 WWW(word wide web)的架構做跨平台的交流。

## 2.7 php

PHP 即超文字預處理器,是一種開源的通用電腦手稿語言,尤其適用於網路開發並可嵌入 HTML 中使用而 PHP 的語法吸收 C 語言、Java 和 Perl 等流行電腦語言的特點,也易於一般程式設計師學習 PHP 在 windows 上有專門的官方移植編譯專案,並且分多個 VC 編譯器版本和執行序安全特性來提供不同的版本支援,而目前大多開發使用者皆採用 PHP 來開發。

## 2.8 Bootstrap

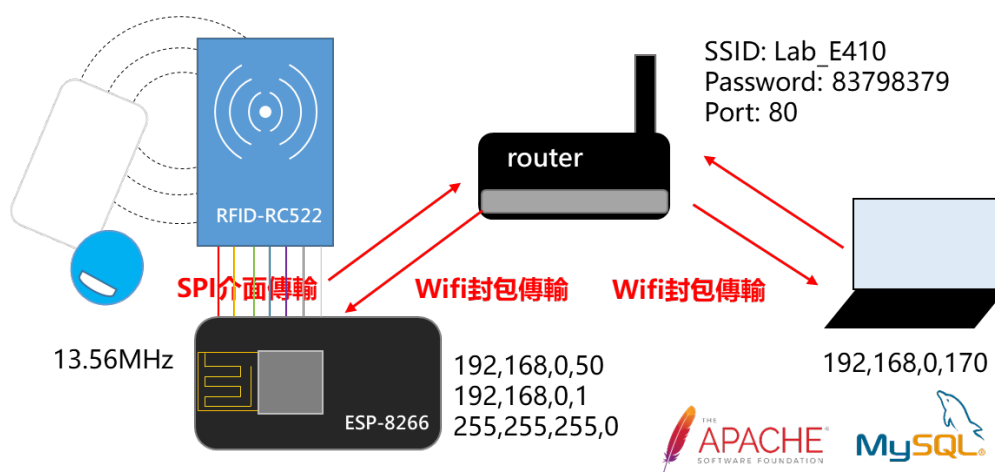
Bootstrap 是一組用於網站和網路應用程式開發的開源前端框架,包括 HTML、CSS 及 JavaScript 的框架,提供字體排印、表單、按鈕、導航及其他各種元件及 JavaScript 擴充套件,本專題使用此框架方便設計、排版網頁和 Web 應用的開發。

## 第三章 研究過程

此章節主要介紹本專題整體架構，分成實體架構與網頁架構進行探討，實體架構部分由 RFID-RC522 及 ESP8266 組成，網頁架構由設計功能、設定資料庫及最後的編寫 php 以實現網頁之功能。

### 3.1 實體架構

本專題會先利用 RFID-RC522 進行讀取標籤上 block 0 區塊上的 UID，此 UID 由 4 個 Byte 構成，由 Hex 方式存入標籤晶片中的記憶體(Ex:0x12 0x34 0x56 0x78)，並利用 ESP-8266 連接至區域網路。終端部分連接此區域網路，形成互聯網架構，一旦有新資料更新就會透過此裝置將 UID 傳送至架設在 apache HTTP sever 的 MySQL 資料庫，資料庫架設於終端主機上，整體實體架構如圖(4)。



圖(4) 實體架構圖。

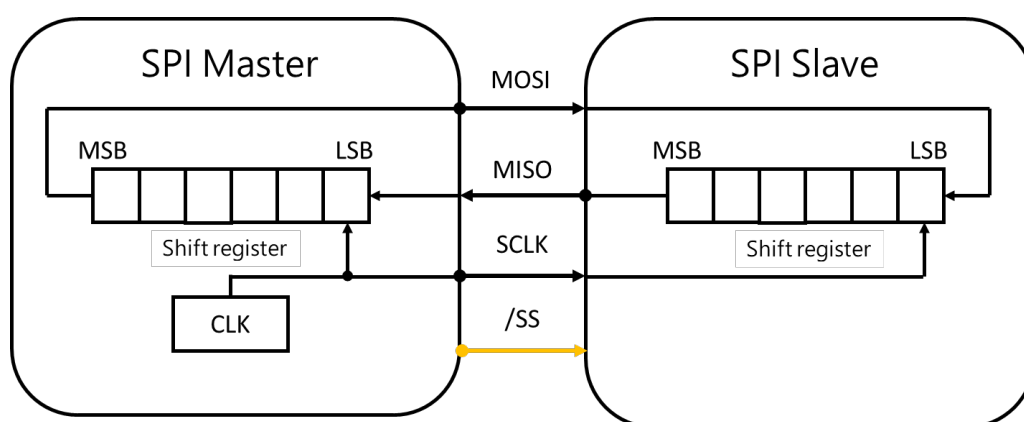
左半邊的裝置至於冰箱上，一旦有食品通過讀卡機，食品上的標籤就會經由讀卡機進行讀取資料的動作。由於現今大多數家庭中皆有家庭用的 Wi-Fi 訊號，只要能夠連接此區域網路，就能以客戶端的方式進行讀取存放在終端網頁的資料，使用這樣的連接方式比較不用擔心資料外洩的問題。

由於資料傳輸皆是在區域網路內進行，並非公開至廣域網路上，若往後有增加在外讀取資料的需求(如門市採購方面確認)，可新增客戶端登入介面，增加資料保密機制及用戶管理功能。

### 3.1.1 讀取標籤的 UID

本專題使用的是由 Mifare 公司所推出的 RFID-RC522，可利用 Arduino IDE 進行編譯及燒錄，而 Arduino 方面有提供基本的範例庫，可直接引用 RFID-RC522 Library 進行開發。傳輸介面部分本專題使用的是 SPI 介面，屬於同步傳輸介面，其優點在於可以有效的減少 CPU 和周邊晶片之間的接腳。

SPI 為一主從式架構，通常有一個 Master 和多個 Slave（從設備），如圖(5)所示。本次使用的 Master 為 ESP8266，Slave 則為 RFID-RC522。



圖(5) SPI 架構。

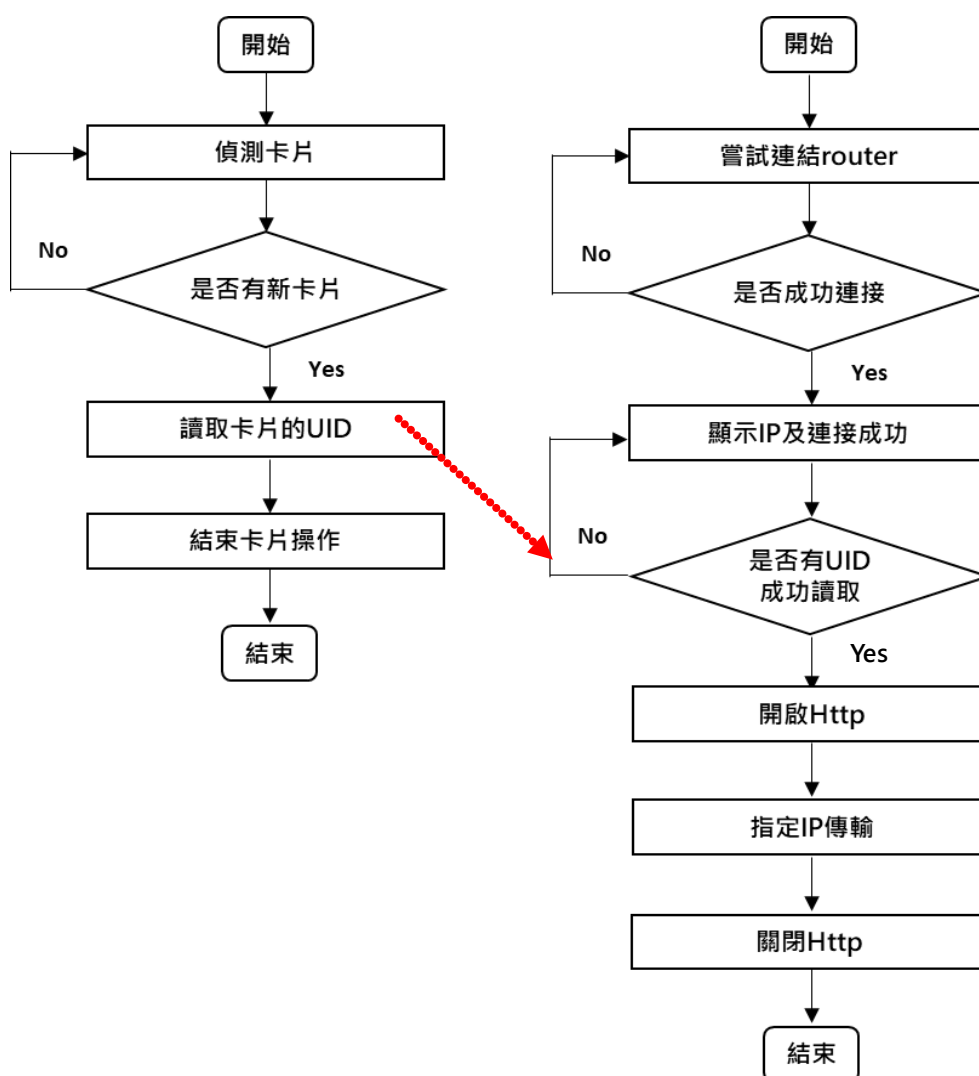
而程式方面，由於需做讀寫的動作，利用 MFRC522.h 函式庫的函式，`mfr522.PICC_IsNewCardPresent()`及 `mfr522.PICC_ReadCardSerial()`，就能做到判斷是否有新卡片接近及卡片格式。當有新卡片接近時，就能用 `mfr522.uid.uidByte[ ]`將 block 0 中的 UID 依序存入 `readcard[ ]`陣列中。

由於 ESP8266 傳輸是以 String 方式傳輸，必須做格式的轉換。轉成 String 的方式是先將 array 的 byte 依序讀出後(遍歷)，判斷這些 byte 是否小於 Hexadecimal "A"，若是小於 A，代表值為 0~9，由 ASCII "0"的位置開始增加；若是大於等於 A，代表值為 A~F，則由 ASCII "A"的位置開始增加，會做這些動作的原因是由於 ASCII 在 0 跟 A 間並不連續，需以判斷式去轉換出正確的 ASCII。

## 3.1.2 上傳 UID 至終端

在傳輸方面，ESP8266 必須先連接上特定的 Router，給定此 Router 的 SSID 及 Password 後，ESP8266 就會自動嘗試連結此 Router，連結成功後會在監控視窗顯示相關資訊，並等待 RFID-RC522 傳輸新字串給自己。一旦新字串輸入，會先將字串新增上 header (postData = "UIDresult=" + UIDresultSend;)，方便進行辨識，再將整個字串傳輸至終端上 (<http://192.168.0.170/.../getUID.php>)。

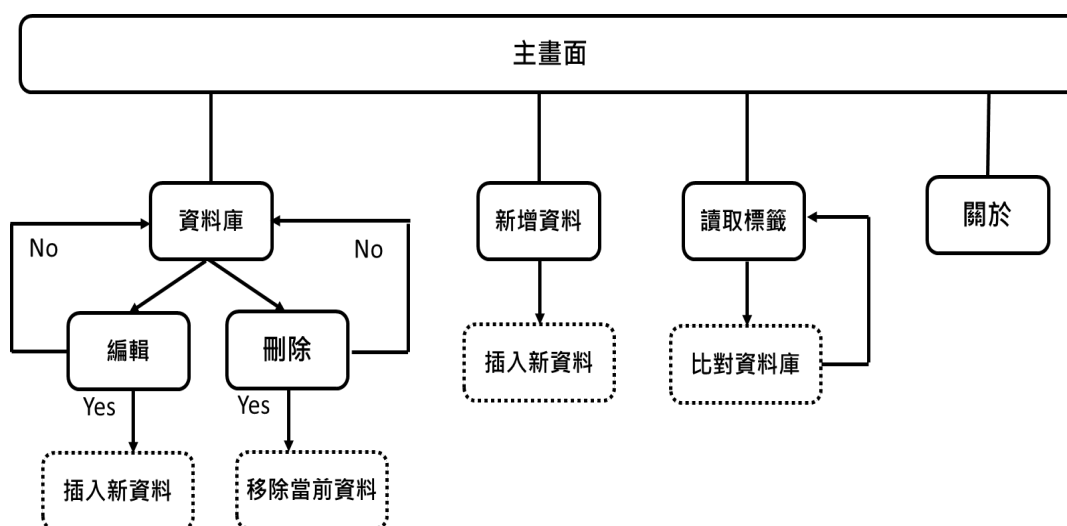
以下為 Arduino IDE code 的流程圖，圖(6)。



圖(6) Arduino IDE 流程圖。

## 3.2 網頁架構

網頁架構的部分分成以下幾類，分別是主畫面、資料庫顯示、新增資料、讀取標籤和關於，如圖(7)。



圖(7) 網頁架構圖。

主畫面的部分為網頁首頁，可點擊連結到其他功能網頁。

資料庫顯示的部分，顯示目前 MySQL 資料庫內的資料，並具有刪除及編輯資料的動作，方便使用者做即時修正。

新增資料的部分，當使用者將新的卡片或標籤接近讀卡機時，可立即將標籤內 UID 上傳至此畫面，使用者即可在此畫面做資料輸入的動作，將食品の種類名稱等相關資訊，做進一步的儲存。

讀取標籤的部分，在使用者將食品拿出冰箱時，若想確認食品的保存狀態，可將食品上的標籤接近讀卡機，讀卡機會根據標籤內的 UID 搜尋資料庫內與之相符的 UID，並讀取此欄位資料。若是標籤沒有做新增的動作，也會提醒使用者此標籤未被新增進 MySQL 資料庫中。

關於的部分，顯示本專題的相關資訊，如器材、軟體名稱等，也有附上終端連結網址，方便展示本作品。

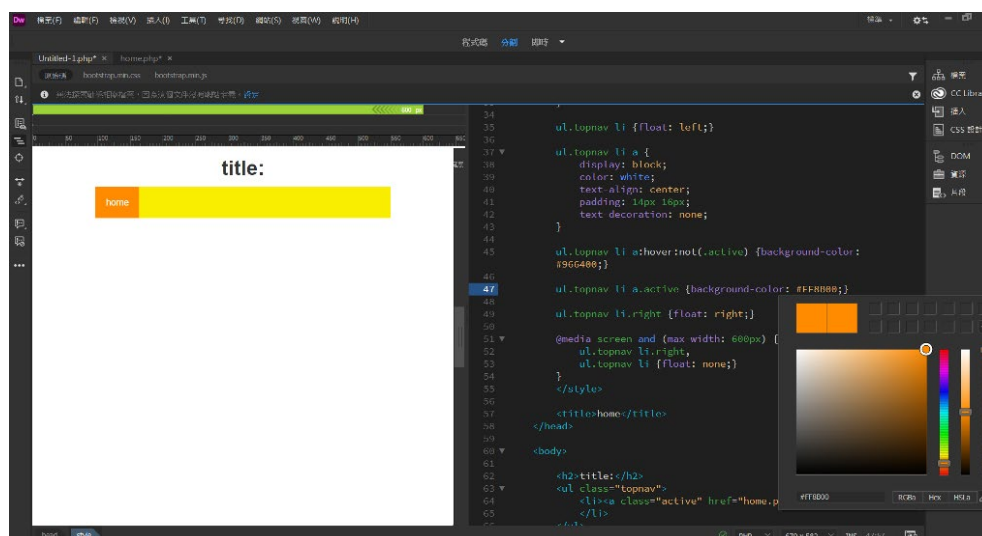
以上五點為本專題的網頁架構，主要的網頁應用在於資料庫、新增資料、讀取標籤三點：資料庫的部分可以在冰箱上做一個液晶面板(本專題使用平板代替)，連接到終端的資料庫，可隨時在冰箱上監控內部食品的狀態，方便使用者規劃食品的購買及存放；新增資料的部分可以選擇透過智慧型手機的輸入，提高整體的方便性。讀取標籤的頁面是為了讓用戶快速得知食品的資訊，在這部分未來能夠與食譜做結合，提供使用者做更多選擇性。

### 3.2.1 接收 UID

網頁端接收 UID 的部分則是建立 getUID 及 UIDContainer 的 php，getUID 是利用 POST 接收 UID，再進行字串切割，將"UIDresult="切割掉，接將切割好的 UID 利用 file\_put\_contents 丟入 UIDContainer，如此一來只要讀取 UIDContainer 就能得到當下讀卡機讀到的標籤所對應的 UID。

### 3.2.2 建立網頁基本架構

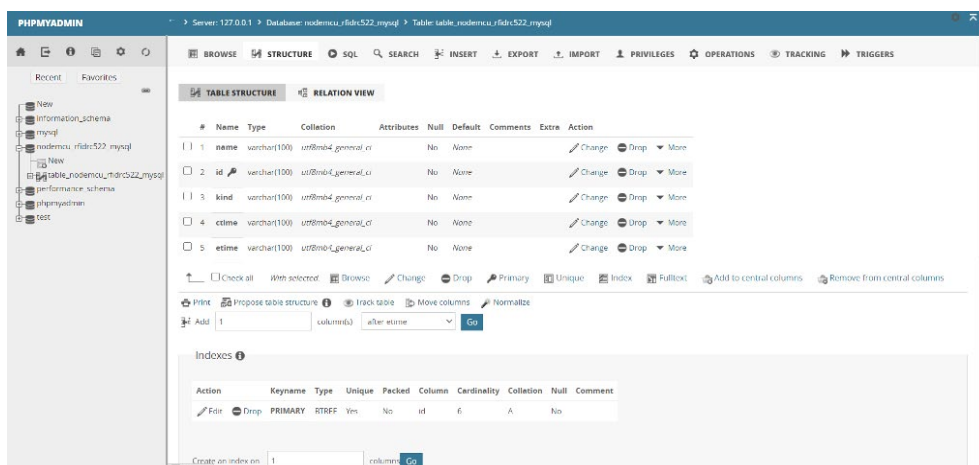
由於本專題使用的是 Bootstrap 4 框架，必須先了解相關設定，在 Bootstrap 的官方網站有提供範例可供參考，使用前必須先下載 CSS (階層式樣式表)及 JS (JavaScript)的 CDN(內容傳遞網路)和程式碼，接著就能透過 Bootstrap 優化過後的布局進行修改及創作，建立起網頁的基本架構，如圖(8)。



圖(8) example.php。

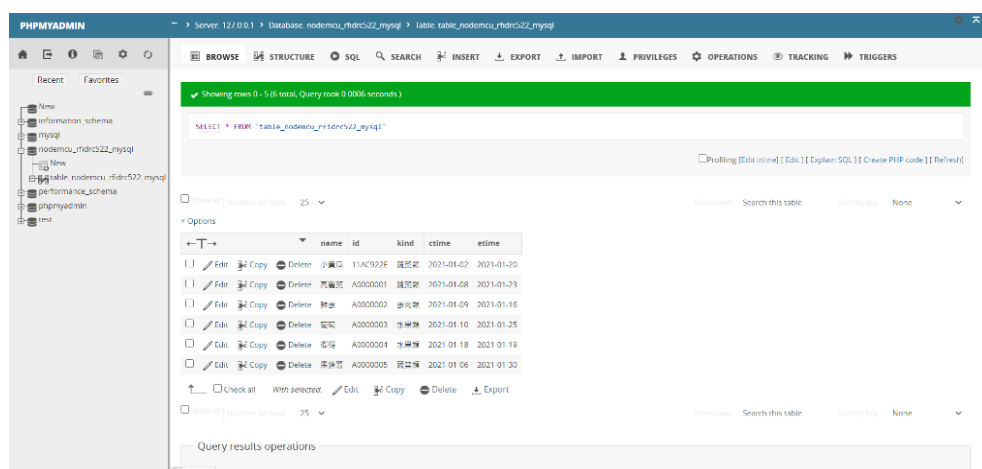
### 3.2.3 建立 MySQL 資料庫

在資料庫的部分，本專題使用 phpMyAdmin 進行設定，先新增一個 MySQL 的 Database，再建立一個專屬的 Table，接著創建出五個欄位，分別是 name、id、kind、ctime(create time)、etime(expiry date)，字節長度設定為 100 bites。



圖(9) MySQL 資料庫欄位設定。

其中 id 的部分由於是連接到我們標籤上的 UID，此欄位必需具有唯一性，否則將會造成資料重複輸入的問題，導致在標籤讀出時造成判斷錯誤。因此在這邊必須將 id 設定為 primary key，primary key 有兩個作用，一是約束作用（constraint），用來規範一個儲存主鍵和唯一性，二是在此 key 上建立了一個主鍵索引，給予 Table 進行索引的依據。



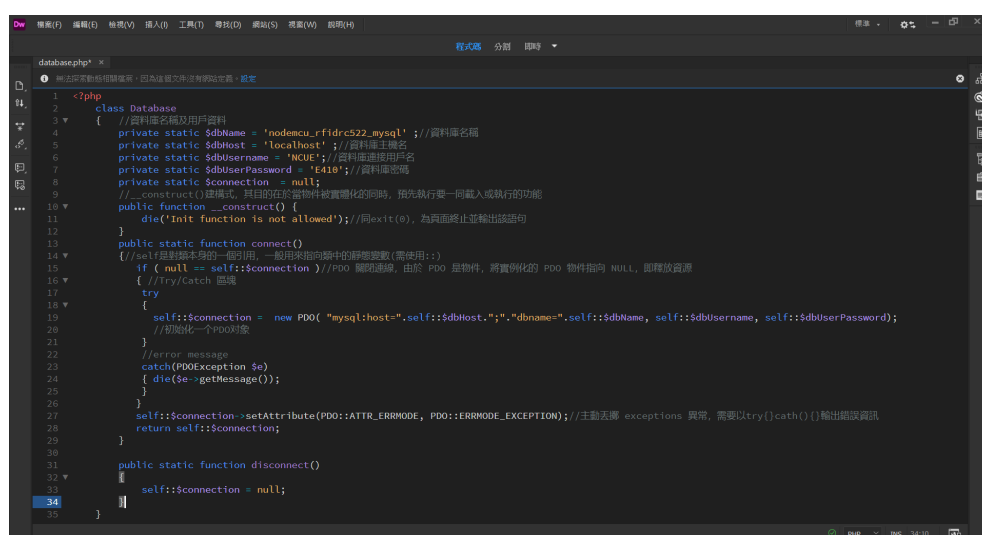
圖(10) MySQL 資料庫內容。



### 3.2.4 建立資料庫頁面

由於先前已經有建立基本的頁面設計，稍加修改即能建立新的 php 頁面，主要重點在於與 database 做結合，這邊必須先建立起 database.php，裡面包含與 database 的連線資料(Name、Host、Username、Password)。資料庫連接使用 PDO (PHP Data Object)方式，PDO 提供了一種名為預處理語句(prepared statement)的機制，它可以將整個 SQL 命令向資料庫伺服器傳送一次，以後只有在引數發生變化時，資料庫伺服器只需對命令的結構做一次分析就夠了，即編譯一次，就可以做多次執行。由於只傳輸改變值，會大幅降低伺服器傳輸資料間開銷。

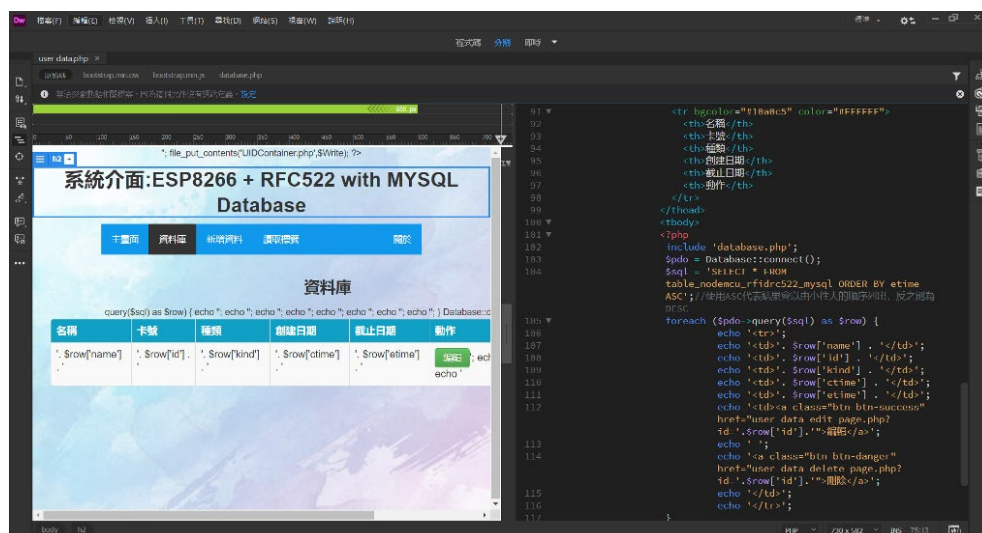
主要靜態函式部分則使用 try/catch 方式排除例外狀態，由於 PDO 屬於物件，需將實例化的 PDO 物件指向 Null，才可釋放資源並斷開與資料庫的連接。



圖(11) database.php。

頁面顯示的部分利用 SQL 之 ORDER BY 語法: SELECT "table\_column" FROM "table\_name" ORDER BY "column\_name" ASC|DESC;定義出 SQL 資料庫搜索函式，使用 ASC 代表結果會以由小到大的順序列出，反之則為 DESC。本專題以保存期限(etime)做 ASC 排序，方便使用者快速知道哪幾種食品即將過期。

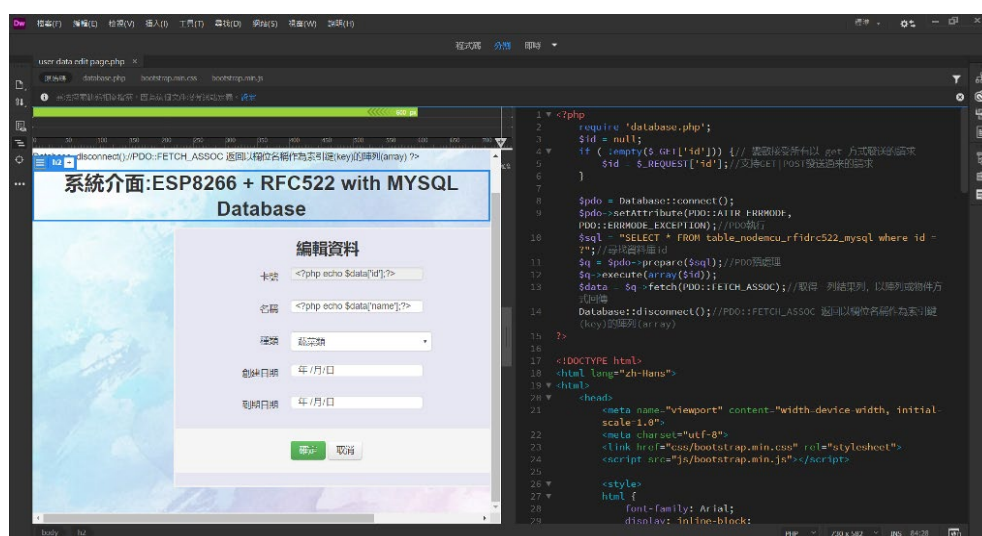
由於介面屬於 php 建構，mysql\_query() 函數在 PHP 中是常用的 MySQL 資料庫語法，用來判斷資料庫查詢是否成功，如果查詢成功則會回傳 true，反之則為 false，用於迴圈判斷資料庫查詢成功與否。



圖(12) user data.php。

在資料庫方面，本專題設計了能夠在此進行編輯或刪除的動作，這邊利用 `SELECT * FROM "table_name" WHERE "column_name" = 'something '` 去找尋特定 id 的相關欄位，在利用 `fetch(PDO::FETCH_ASSOC)` 取得一列結果列，以陣列或物件方式回傳，PDO::FETCH\_ASSOC 的意思則為返回以欄位名稱作為索引鍵 (key) 的陣列 (array)，如此就能完成以 id (UID) 為 key 傳回完整的陣列，並使用 `UPDATE "table_name" "column_name" = ?...`，能將更動的部分做資料庫更新。

刪除的部分有做兩段式驗證，必須在刪除頁面按下確認才能刪除，否則將跳回資料庫頁面。刪除的部分是將 id 清空後回傳，即可以刪除該欄位資料。



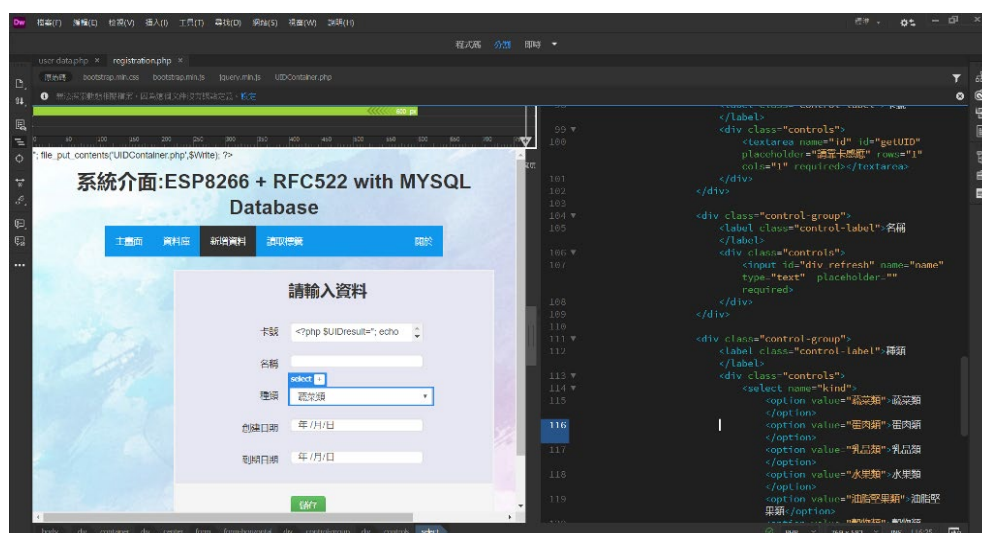
圖(13) user data edit page.php。

### 3.2.5 建立新增頁面

新增頁面的部分必須和讀卡機做結合，當標籤接近讀卡機時，只要有成功讀取到標籤內的 UID，就能立即顯示在此頁面上。想達到此目的必須使用先前 3.2.1 接收 UID 中所提到的 UIDContainer.php，利用 JavaScript 中的 setInterval 語法，每隔 1000 ms 就 loading 一次 UIDContainer，由於 UIDContainer 會因當下的 UID 做變動，如此一來就能欄位中的 UID 做不斷更新，能夠做出多筆資料輸入的動作，不需手動輸入該標籤的 UID。

當資料輸入完成時，按下儲存按鈕，就能夠利用 insertDB.php 進行儲存，若是填寫的資料不完整，會提醒使用者完成所有資料填寫。

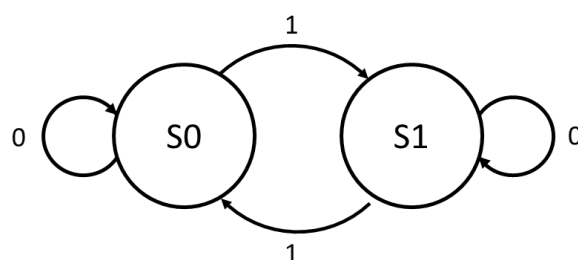
insertDB 的寫法與 3.2.4 中的編輯資料相似，不同點在於 3.2.4 中的資料欄位已經存在，而這邊的 SQL 語法需改寫成 INSERT INTO "table\_name" ("column\_name1", "column\_name2"... ) values(?, ?...)，就能將新得到的資料逐步輸入到 MySQL 資料庫的每個欄位中。



圖(14) registrantion.php。

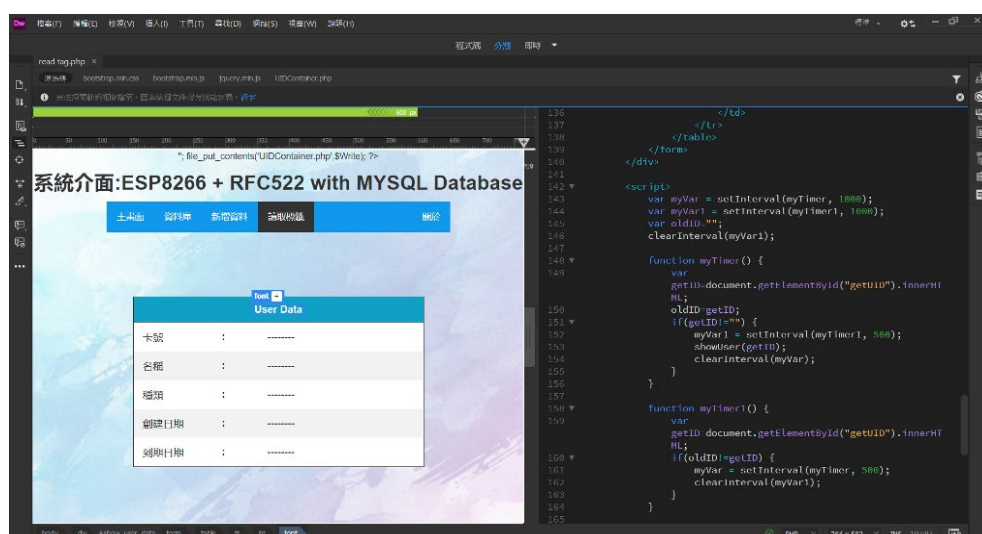
## 3.2.6 建立讀取頁面

讀取畫面的部份，設計一個有限狀態機去判斷是否有新的 UID 輸入，S0 (myTimer)為每隔 500ms 印出 UID 並將 UID 存入 oldUID，進入 S1 (myTimer1)。S1 為每隔 500ms 判斷 UID 是否等於 oldUID，若不等於代表有新的值輸入，則回到 S0 執行印出工作，如下圖(13)。



圖(15) 狀態圖。

接著引進 showUser 的 JavaScript 函式，其功能為檢查是否有用戶被選擇為 XMLHttpRequest 的對象，會創建在伺服器響應完成時執行的函式，並向伺服器上的文件發送請求。利用 showUser 可以找尋伺服器頁面名為"read tag user data.php" 的 PHP 文件，此文件會開啟到 MySQL 數據庫的連接，並找到選中的 UID 後，創建 HTML 表格和填充數據，發送回"User Data" 佔位符，完成資料庫讀取的動作。



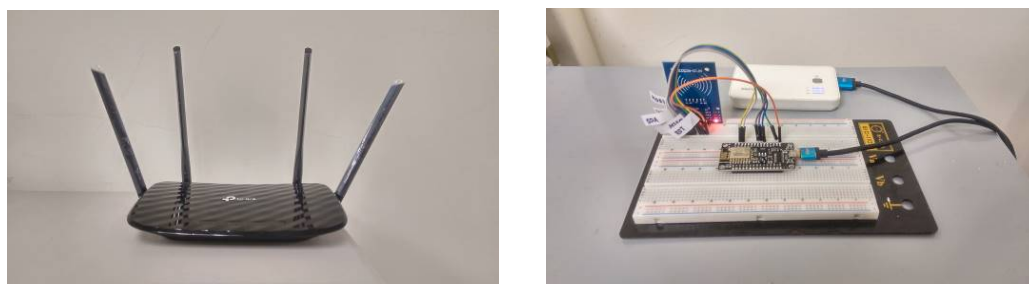
圖(16) red tag.php。

## 第四章 成果展示

### 4.1 實體展示

首先必須先建置區域網路，使用路由器設定好 WIFI 的 SSID 及 Password。

將 RFID 與 ESP8266 用杜邦線連接，確認是否遵循 SPI 連接方式(GPIO 腳位設定需與 Arduino IDE 設計之腳位相符)。以下為實體裝置圖，分別是 RFID-RC522、NodeMCU ESP-8266、電源供應器。



圖(17) 路由器 & 實體裝置。

若是需要做即時顯示，可在冰箱門上加裝面板，供用戶進行編輯、查詢資料的動作。(本專題以現有平板代替此面板設計)



圖(18) 資訊面板。



## 4.2 網頁展示

1.主畫面: 網頁首頁，可點擊連結到其他功能網頁。



圖(19) 主畫面。

2.資料庫: 顯示目前 MySQL 資料庫內的資料，並具有刪除及編輯資料的動作，方便使用者做即時修正。



圖(20) 資料庫。

3.編輯資料庫資料: 選定想要編輯的資料欄位，點擊編輯後跳出編輯頁面，可再次修正資料庫錯誤輸入的資料。



圖(21) 編輯資料庫資料。

4.刪除資料庫資料:選定想要刪除的資料欄位，點擊刪除後跳出刪除頁面，二次確認防止誤刪情況發生。



圖(22) 刪除資料庫資料。

5. 新增標籤內容：當使用者將新的卡片或標籤接近讀卡機時，可立即將標籤內 UID 上傳至此畫面，使用者即可在此畫面做資料輸入的動作，將食品的種類名稱等相關資訊，做進一步的儲存。

系統介面:ESP8266 + RFC522 with MYSQL Database

主畫面 資料庫 新增資料 讀取標籤 關於

請輸入資料

卡號

名稱

種類

創建日期

到期日期

儲存

圖(23) 新增標籤內容。

6. 讀取標籤資料：在使用者將食品拿出冰箱時，若想確認食品的保存狀態，可將食品上的標籤接近讀卡機，讀卡機會根據標籤內的 UID 搜尋資料庫內與之相符的 UID，並讀取此欄位資料。若是標籤沒有做新增的動作，也會提醒使用者此標籤位被新增進 MySQL 資料庫中。

系統介面:ESP8266 + RFC522 with MYSQL Database

主畫面 資料庫 新增資料 讀取標籤 關於

請將“卡片”或“標籤”放置於讀卡機前!

User Data	
卡號	:
名稱	:
種類	:
創建日期	:
到期日期	:

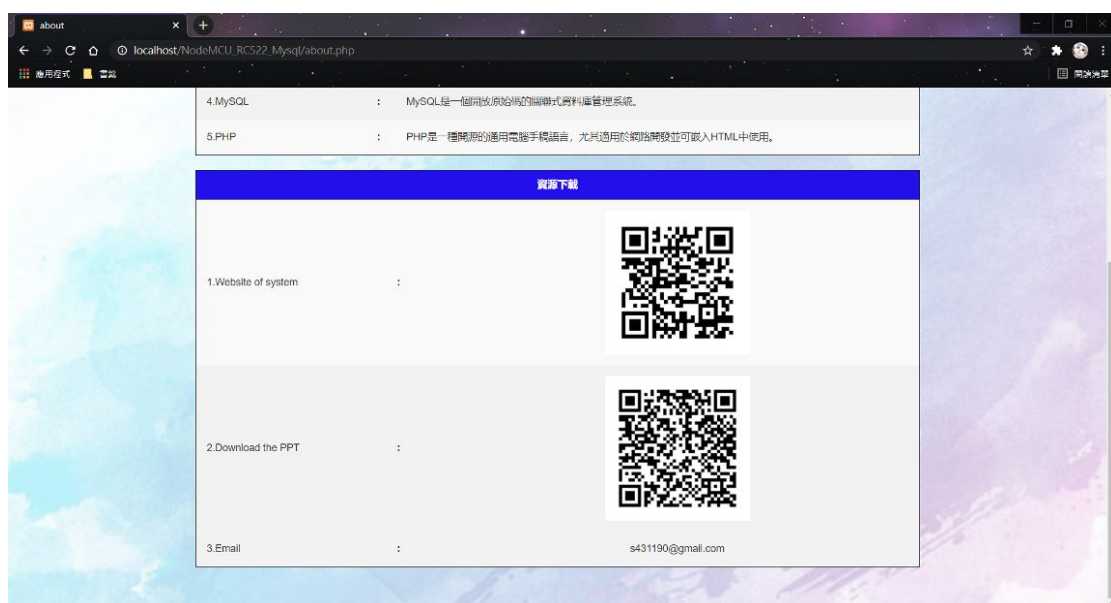
圖(24) 讀取標籤資料。



7. 關於:顯示本專題的相關資訊,如器材、軟體名稱等,也有附上終端連結網址,方便展示本作品。



圖(25) 關於(上半)。



圖(26) 關於(下半)。

## 第五章 結論

本專題提出一種方式去改善冰箱內儲物系統，藉由系統化的紀錄能避免掉不必要的能源浪費，對於家庭用戶而言也能更加方便規劃食材的選購。現今系統仍有進步空間，本次僅只是完成資料庫的建設及讀取，若能添加功能如：食譜系統、帳號登入系統、圖表化成呈現食品存放數量...等，會對使用者增加更多的便利性。

在製作本次專題的過程，由於有事先規劃出整體架構，在進行架構時會較具有系統性的規劃，但在實現理想設計的過程中，往往會遇到許多自己未曾想過的問題，需要經過不斷的嘗試修改、重新設定，或是上網找尋相關的範例教學，才能成功完成自己想要的功能。很多東西都是遇到問題後，才會真正學習到該如何去解決，或是了解到語法上的差異，在我一開始起步時，往往會到處碰壁，經過參看許多範例及他人的教學後，雖說用途有時與本專題並無相關，但透過理解他人的程式，能夠從中吸收並找尋新的點子，經過加以修改與編寫新的 function 後，就能架構出一套雲端資料庫系統，雖仍有些地方並非是最佳寫法或功能設計有所缺陷，但是能夠完成本系統還是十分開心的。

## 參考文獻

- [1] R. Want, "Enabling Ubiquitous Sensing with RFID", Computer, vol. 37, no. 4, pp. 84-86, 2004.
- [2] R. Want, "An introduction to RFID technology", Pervasive Computing, vol. 5, pp. 25-33, Feb. 2006.
- [3] Wang Yuan, Li Shuhua, Zhang Haifeng, "Design and realization of the LED management system based on PHP", Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Sep. 2011.
- [4] Jono Bacon, "Practical PHP and MySQL Building Eight Dynamic Web Applications", 2008.
- [5] Luke Welling, Laura Thomson, "PHP and MySQL Web Development", 5th Edition, Seq. 2017.
- [6] NXP company, "MFRC522 Standard performance MIFARE and NTAG frontend", Apr. 2016.
- [7] 趙英傑，《超圖解 Arduino 互動設計入門 第三版》第 17 章節，2016。
- [8] 王英英，《MySQL 最新版：最成功應用範例全書》，2020。
- [9] Arduino RFID Library for MFRC522  
<https://github.com/miguelbalboa/rfid>
- [10] PHP database connection class  
<https://gist.github.com/hexaholic/dcf92e903123887fbf30>
- [11] W3schools network of php  
<https://www.w3schools.com/php>
- [12] Connect RFID to PHP and MySQL Database with NodeMCU ESP8266  
<https://www.youtube.com/watch?v=AmK4AUwq1Ag>