

中華大學資訊工程學系

101學年度專題製作期末報告

NFC 行動盤點 APP

組長：黃群凱

組員：顏宜毫、郭承翰、陳裕承

指導老師：吳林全 老師

專題編號：PRJ2012-CSIE-10117

製作期間：2012/7/1 ~ 2013/6/21

目錄

一、 前言	4
二、 動機	4
三、 工作與時間分配.....	5
1、 時間分配表	5
四、 專題流程.....	6
1、 專題進行階段	6
2、 整體流程圖	7
五、 專題製作及資料.....	8
1、 <i>NFC(Near Field Communication)</i> 簡介.....	8
2、 <i>NFC</i> 工作模式	9
a. 卡片模式(Card emulation).....	9
b. 點對點模式(P2P mode)	9
c. 讀卡器模式(Reader/writer mode).....	9
3、 與藍牙的比較	10
4、 開發環境與工具.....	11

5、 碰到的問題與設定.....	11
a. 資料庫連接	11
b. 資料庫環境設定.....	12
c. 伺服器連線設定.....	13
d. 手機開發程式部分	16
e. 手機連線 MYSQL(資料庫系統).....	19
f. 手機網路功能.....	21
六、 專題製作成果之特色	24
1、 特色.....	24
2、 缺點.....	24
七、 未來展望.....	25
八、 結論	25
九、 銘謝	25
十、 參考文獻.....	26

一、前言

我們專題的主要目的在於用智慧型手機做盤點功能，而其中結合近場通訊 (Near Field Communication; NFC) 技術達到快速感應盤點物品的識別編碼，再透過無線網路 (WiFi) 連接主機端伺服器資料庫進行資料的比對，比對完之後在手機上顯示物品資訊，最後做認證動作完成整個盤點系統的運作，容易操作並減少盤點時間。另外使用者若是有想新增的物品，也可以感應物品的識別號碼，並輸入資料內容，將資料輸入資料庫完成新增。

二、動機

近場通訊 (Near Field Communication; NFC) 之技術為未來發展趨勢，尤其結合在智慧型手機上最為廣泛，現代化之社會人手一支智慧型手機，而將日常生活的動作藉由手機的結合，使生活更加方便迅速是未來的趨勢。

平常要盤點物品時，總是要拿著物品清單，查看物品編號，再查找物品是否在清單上，若是有個能自動判斷是否在清單上的 APP 就會方便許多，因此我們選擇研究此技術來實現這項功能作為專題。



圖 2-1 近場通訊 (NFC) 的發展趨勢

三、工作與時間分配

1、時間分配表

專題製作過程之工作與時間分配狀況。

表 3-1 工作與時間分配表

	第一個月	第二個月	第三個月	第四個月	第五個月	第六個月	第七個月	第八個月	第九個月	第十個月	第十一個月	第十二個月
研究方向	●	●										
架構分析			●									
程式研究				●	●	●						
程式實作					●	●	●	●				
程式除錯							●	●	●			
測試與實驗								●	●	●		
內容整合											●	●

四、專題流程

1、專題進行階段

本專題進行方式大致上分為五大階段。

第一階段:

文獻研究與探討、架構分析。包含蒐集相關 NFC 資訊，以及相關技術資料，系統可能需要使用的設備，包含準備擁有 NFC 智慧型手機、RFID 卡、伺服器等設備。

第二階段:

程式開發。包含開發手機 APP 去讀取 tag 卡號，手機使用者介面設計，伺服器架設，後端資料庫系統建立，網頁設計。

第三階段:

進行系統整合，將開發完成的程式進行結合。

第四階段:

進行測試與修正，實際跑過整個系統流程，測試效能是否達到預期以及修正不足之程式。

第五階段:

將整體系統進行包裝，並撰寫報告。

2、 整體流程圖

使用者(盤點人員)透過手機讀取欲盤點之設備或物品，並取得欲盤點設備與物品之資料開始進行盤點，盤點過程透過 WiFi 傳送至伺服器資料庫系統進行比對確認，並做資料的新增、修改與刪除，當完成盤點之後伺服器回傳訊息到使用者(盤點人員)手機。而管理者可透過伺服器之網頁進行進一步資料修改。

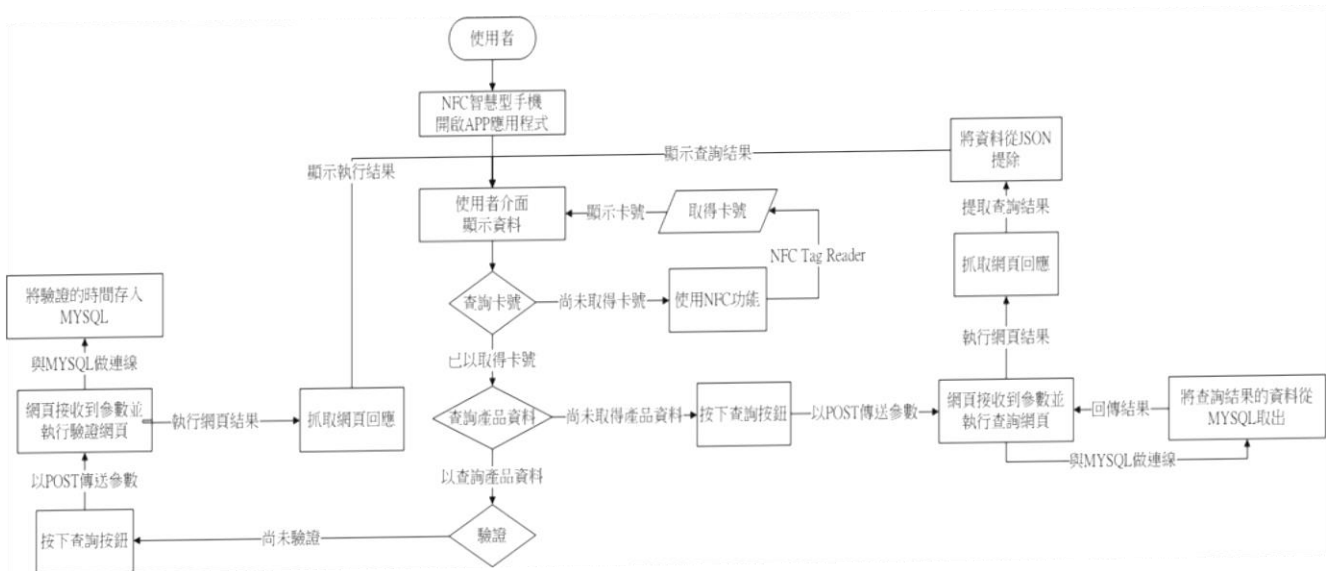


圖 4-1 系統流程圖

五、專題製作及資料

1、NFC(Near Field Communication)簡介

近場通訊(英語：Near Field Communication，NFC)，又稱近距離無線通訊，是一種短距離的高頻無線通訊技術，允許電子設備之間，使用 NFC Tag 裝置進行非接觸式點對點資料傳輸，在十公分（3.9 英吋）內交換資料。

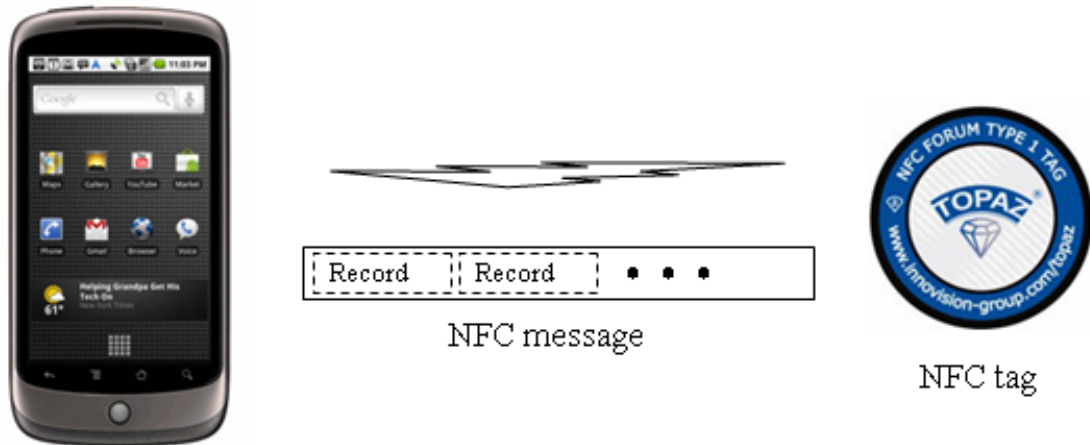


圖 5-1 近場通訊示意圖

這個技術由免接觸式射頻識別（RFID）演變而來，由飛利浦、諾基亞和索尼共同研製開發，其基礎是 RFID 及互連技術。近場通訊是一種短距高頻的無線電技術，在 13.56MHz 頻率執行於 20 公分距離內。其傳輸速度有 106 Kbit/秒、212 Kbit/秒或者 424 Kbit/秒三種。目前近場通訊已通過成為 ISO/IEC IS 18092 國際標準、EMCA-340 標準與 ETSI TS 102 190 標準。NFC 採用主動和被動兩種讀取模式。

2、NFC 工作模式

a. 卡片模式(Card emulation)

這個模式其實就是相當於一張採用 RFID 技術的 IC 卡。可以替代現在大量的 IC 卡(包括信用卡) 場合商場刷卡、悠遊卡、門禁管制，車票，門票等等。此種方式下，有一個極大的優點，那就是卡片透過非接觸讀卡器的 RF 域來供電，即便是寄主裝置(如手機)沒電也可以工作。

b. 點對點模式(P2P mode)

這個模式和紅外線差不多，可用於資料交換，只是傳輸距離較短，傳輸建立速度較快，傳輸速度也快些，功耗低(藍芽也類似)。將兩個具備 NFC 功能的裝置連結，能實作資料點對點傳輸，如下載音樂、交換圖片或者同步裝置位址簿。因此透過 NFC，多個裝置如數位相機、PDA、電腦和手機之間都可以交換資料或者服務。

c. 讀卡器模式(Reader/writer mode)

作為非接觸讀卡器使用，比如從海報或者展覽訊息電子標籤上讀取相關訊息。

3、與藍牙的比較

NFC 和藍牙 (Bluetooth) 都是短程通訊技術，而且都被整合到行動電話。但 NFC 不需要複雜的設定程式。NFC 也可以簡化藍芽連線。對於行動電話或是行動消費性電子產品來說，NFC 的使用比較方便。NFC 的短距離通訊特性正是其優點，由於耗電量低、一次只和一台機器連結，擁有較高的保密性與安全性，NFC 有利於信用卡交易時避免被盜用。NFC 的標的並非是取代藍芽等其他無線技術，而是在不同的場合、不同的領域起到相互補充的作用。

表 5-1 NFC 與 Bluetooth 各項比較

	NFC	Bluetooth
設定程式	< 0.1 秒	< 6 秒
功率耗損	< 15mA	< 30mA
標準化機構	ISO/IEC	Bluetooth SIG
網路標準	ISO 13157, etc.	IEEE 802.15.1
網路類型	Point-to-point	WPAN
範圍	< 0.2 m	~10 m
頻率	13.56 MHz	2.4-2.5 GHz
Bit rate	424 kbit/s	2.1 Mbit/s

4、開發環境與工具

作業系統：Windows 7 企業版 64 位元

網頁環境：WAMPSEVER (64 BITS & PHP 5.3) 2.2E

Android 手機開發環境：Android Developer Tools v21.1.0-569685

Android 手機測試機器：Google Nexus mobile phone

5、碰到的問題與設定

a. 資料庫連接

在當初要建立時，因為考慮到輸入的格式是中文，以 Big5 的方式存入的話，可能會造成一些中文上顯示的問題，所以後來以 Utf-8 的方式來建立資料庫。

一開始安裝完 WAMPSEVER，發現 mysql 環境語言是 latin1，所以後來將 my.ini 得設定檔裡面加入了下面幾項指令設定：

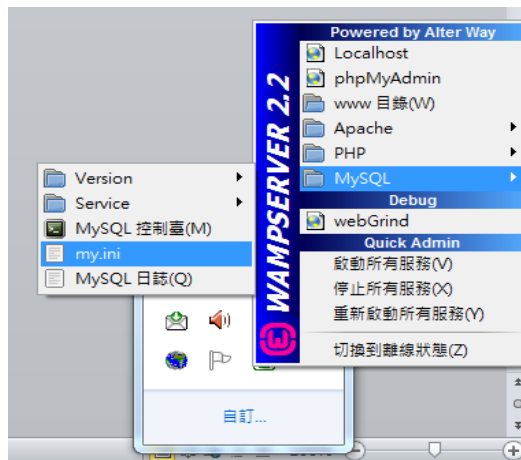


圖 5-1 開啟 WAMP 的工具列，進入到 MYSQL 設定檔

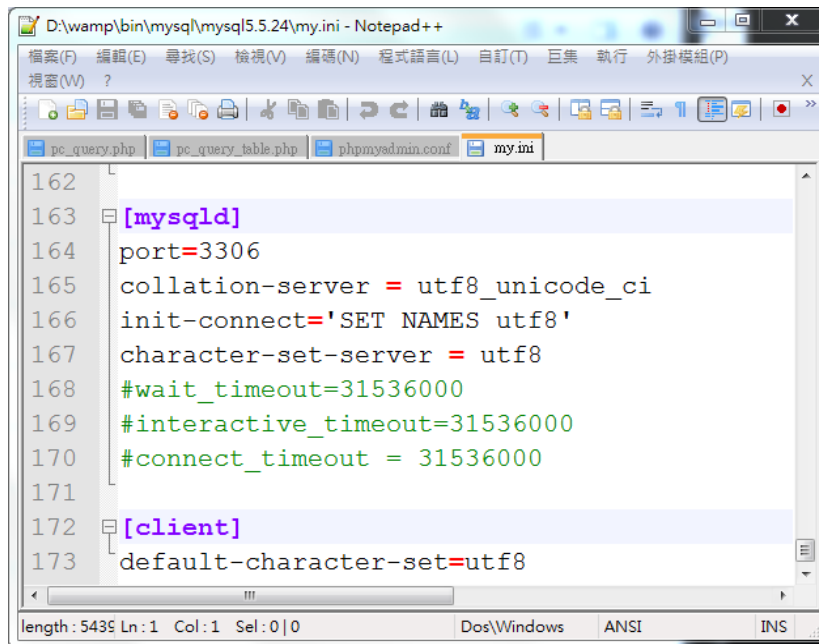


圖 5-2 將 UTF-8 的環境設定加入 MYSQL 設定檔

b. 資料庫環境設定

除了以上的設定檔案之外，在建立資料庫時除了建立的指令之外，還要下選擇語系參數的指令

```
DROP DATABASE IF EXISTS nfc_data ;
```

```
CREATE DATABASE nfc_data DEFAULT CHARACTER SET utf8 COLLATE  
utf8_unicode_ci ;
```

```
CREATE TABLE data(  
nfc_id char(8) NOT NULL,  
name varchar(50) default NULL,  
content varchar(255) default NULL,  
owner varchar(20) default NULL,  
user varchar(20) default NULL,  
price int(11) UNSIGNED default NULL,  
place varchar(255) default NULL,  
buy_date date default NULL,  
PRIMARY KEY (nfc_id)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci ;
```

```

D:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use nfc_data;
Database changed
mysql> DESCRIBE data;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nfc_id     | char(8)       | NO   | PRI | NULL    |      |
| name      | varchar(50)   | YES  |     | NULL    |      |
| content    | varchar(255)  | YES  |     | NULL    |      |
| owner     | varchar(20)   | YES  |     | NULL    |      |
| user      | varchar(20)   | YES  |     | NULL    |      |
| price     | int(11) unsigned | YES  |     | NULL    |      |
| place     | varchar(255)  | YES  |     | NULL    |      |
| buy_date  | date          | YES  |     | NULL    |      |
| check_date| datetime      | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.09 sec)

mysql>

```

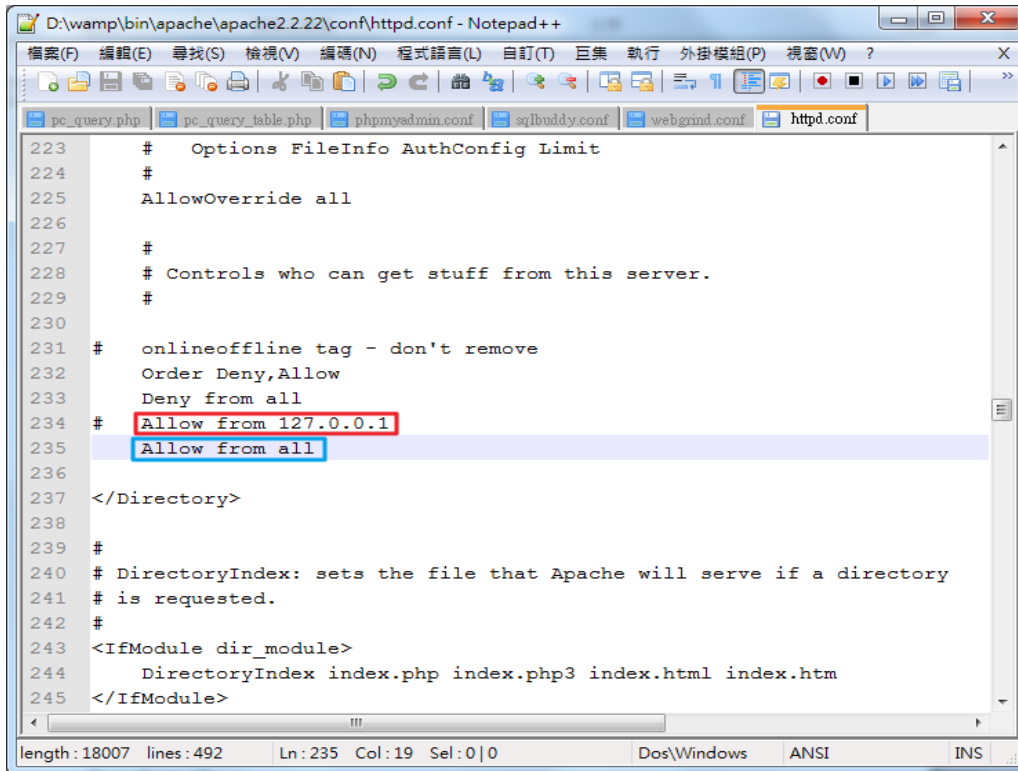
圖 5-3 資料表建立後的結構內容

c. 伺服器連線設定

還有一些 WAMP 連線設定要修改，否則無法連線



圖 5-4 開啟 WAMP 的工具列，進入到 Apache 設定檔



```
D:\wamp\bin\apache\apache2.2.22\conf\httpd.conf - Notepad++
檔案(F) 編輯(E) 尋找(S) 檢視(V) 編碼(N) 程式語言(L) 自訂(T) 巨集 執行 外接模組(P) 視窗(W) ?
pc_query.php pc_query_table.php phpmyadmin.conf sqlbuddy.conf webgrnd.conf httpd.conf
223 # Options FileInfo AuthConfig Limit
224 #
225 AllowOverride all
226
227 #
228 # Controls who can get stuff from this server.
229 #
230
231 # onlineoffline tag - don't remove
232 Order Deny,Allow
233 Deny from all
234 # Allow from 127.0.0.1
235 Allow from all
236
237 </Directory>
238
239 #
240 # DirectoryIndex: sets the file that Apache will serve if a directory
241 # is requested.
242 #
243 <IfModule dir_module>
244     DirectoryIndex index.php index.php3 index.html index.htm
245 </IfModule>
length: 18007 lines: 492 Ln: 235 Col: 19 Sel: 0|0 Dos\Windows ANSI INS
```

圖 5-5 將只有本機連線設定改成所有位置都可以連線，

更改完畢後還需要重新啟動伺服器即可完成。

PS：skype 會佔住 80 port 的問題

可以參考下圖的設定



圖 5-6 啟動 SKYPE → 功能 → 設定選項

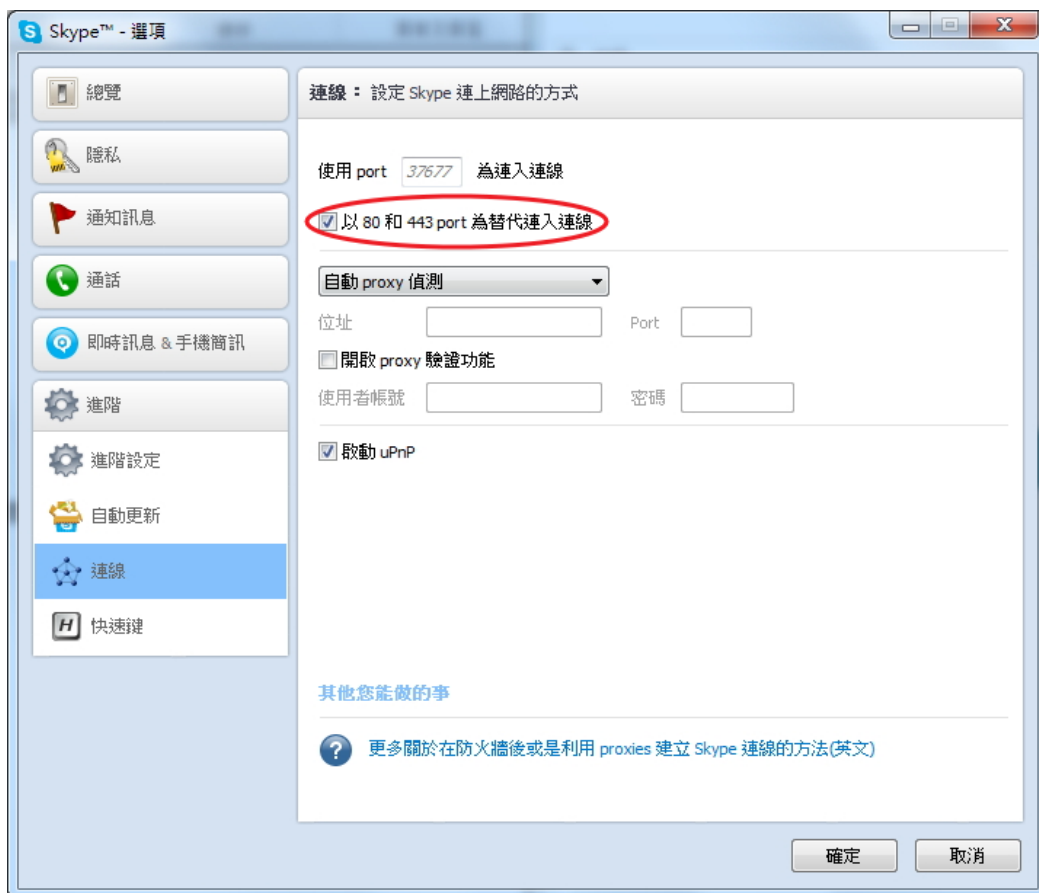


圖 5-7 將圖中的選項取消打勾

d. 手機開發程式部分

為了使用 NFC 的功能，除了要使用有 NFC 功能的手機之外，要讓程式能夠處理 NFC Tag 的資料必須先在程式功能描述檔 `AndroidManifest.xml` 中完成以下設定：

```
<uses-sdk
    android:minSdkVersion="10"
    android:targetSdkVersion="17" />
<uses-feature android:required="true" android:name="android.hardware.nfc" />
<uses-permission android:name="android.permission.NFC" />
```

圖 5-8 加入 NFC 硬體功能開啟指令

```
<activity
    android:name="chu.nfc.test2.MainActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.nfc.action.TAG_DISCOVERED" />
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

    <intent-filter>
        <action android:name="android.nfc.action.TECH_DISCOVERED"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>

    <meta-data android:name="android.nfc.action.TECH_DISCOVERED" android:resource="@xml/nfc_tech_filter"/>
</activity>
```

圖 5-9 加入 NFC 偵測功能開啟指令

如果要讓程式能夠處理 ACTION_TECH_DISCOVERED 型態的 Intent，必須先在程式專案的 res 資料夾中新增一個 xml 子資料夾，並且在裡頭建立一個 xml 檔，例如可以將它取名為 nfc_tech_list.xml，然後編輯它的內容如下：

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
3     <tech-list>
4         <tech>android.nfc.tech.MifareClassic</tech>
5         <tech>android.nfc.tech.IsoDep</tech>
6         <tech>android.nfc.tech.NfcA</tech>
7         <tech>android.nfc.tech.NfcB</tech>
8         <tech>android.nfc.tech.NfcF</tech>
9         <tech>android.nfc.tech.NfcV</tech>
10        <tech>android.nfc.tech.Ndef</tech>
11        <tech>android.nfc.tech.NdefFormatable</tech>
12        <tech>android.nfc.tech.MifareUltralight</tech>
13    </tech-list>
14 </resources>
```

圖 5-10 nfc_tech_list.xml 加入程式可以讀取卡片的規格

假如我不想在執行 APP 的時候又會跳出 NFC Menu，而是我偵測到 NFC 標籤(TAG) 時交給程式自動去處理呢？

這時就要另外一種操作 NFC 的方式 - Using the foreground dispatch system 透過這種方式，當 NFC 標籤被偵測到會傳遞 Intent 過來，程式可優先把 NFC Intent 抓下來處理而不交給 Android 的系統的 dispatch system。其使用方式如下：

要先在 onCreate()裡面定義和宣告這隻 Activity 所接收的 Intent 和標籤格式

```
try {
    NfcManager manager = (NfcManager) getSystemService(NFC_SERVICE);
    mAdapter = manager.getDefaultAdapter();
    mPendingIntent = PendingIntent.getActivity(
        this, 0, new Intent(this, getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);
    IntentFilter ndef = new IntentFilter(NfcAdapter.ACTION_TECH_DISCOVERED);
    mFilters = new IntentFilter[] {
        ndef,
    };
    mTechLists = new String[][] { new String[] { NfcA.class.getName() } };
} catch (Exception e) {
    e.printStackTrace();
}
```

圖 5-11 讓程式不透過系統去抓取 Intent

而在 `onResume()` 宣告如下：

```
@Override
protected void onResume() {
    super.onResume();
    mAdapter.enableForegroundDispatch(this, mPendingIntent, mFilters, mTechLists);
}
```

圖 5-12 設定 Activity 在前台時，才啟動抓取 Intent 的功能

在 `onPause()` 宣告如下：

```
@Override
protected void onPause() {
    super.onPause();
    mAdapter.disableForegroundDispatch(this);
}
```

圖 5-13 設定 Activity 在結束時，將抓取 Intent 的功能關閉

以上就算是配置完成了，以後這隻程式就有能力優先處理 NFC Intent 接下來會問收到的 NFC Intent 在 `onNewIntent` 這個地方，將程式所接收到的 NFC Intent 的資料讀出並將卡片上的卡號(byte)取出來

```
public void onNewIntent(Intent intent)
{
    /*
     * 將程式所接收到的NFC Intent的資料讀出
     * 並將卡片上的卡號(byte)取出來
     */
    Tag myTag = (Tag) intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
    byte[] tagId = intent.getByteArrayExtra(NfcAdapter.EXTRA_ID);

    hex.setText("");
    out_hex = "";

    try {
        out_hex = getHexString(tagId); // 將卡片上的卡號(byte)轉成16進位
        hex.setText(out_hex); // 顯示卡片上的卡號(16進位)
    } catch (Exception e) {
        hex.setText("讀取卡號失敗");
        e.printStackTrace();
    }
}
```

圖 5-14 使用 NFC Reader 模式，將卡號取出

但是 NFC Intent 的資料讀出並不是 16 進位，所以要將資料轉成 16 進位

```
public static String getHexString(byte[] b) throws Exception
{
    // 將卡號從 byte 轉成 Hexadecimal
    String result = "";
    for (int i=b.length - 1 ; i >=0 ; i--)
    {
        result += Integer.toString( ( b[i] & 0xff ) + 0x100, 16).substring(1);
    }
    return result;
}
```

圖 5-15 將卡號的 byte 資料轉成 16 進位

e. 手機連線 MYSQL(資料庫系統)

在當初一開始，本來是利用 JAVA 的 JDBC 方式來讓手機與 MYSQL 直接連線，但是後來實作過程中，發現一直無法利用 JDBC 方式連到 MYSQL。

所以就換個方式，利用 HttpClient 類處理網路訪問先連線到伺服端的 PHP 網頁，再利用 PHP 網頁與 MYSQL 連線來操作資料庫。

在伺服端部分先建立手機連線專用的 PHP 網頁

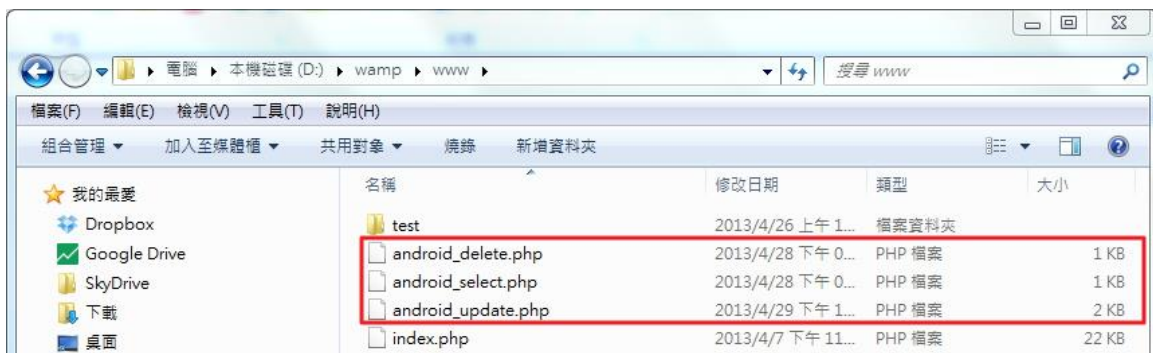


圖 5-16 建立要讓手機使用功能的網頁

由於 JAVA 在讀取 UTF-8 文件時，有的文件會添加 UTF-8 的 BOM 時進去，但是 JAVA 編譯器並不處理 BOM，如果你要將網頁存為 UTF-8，而且必須使用像是 Notepad++、UltraEdit 等文字編譯器，使文件可存為檔首無 BOM 的格式。參考圖 5-17。

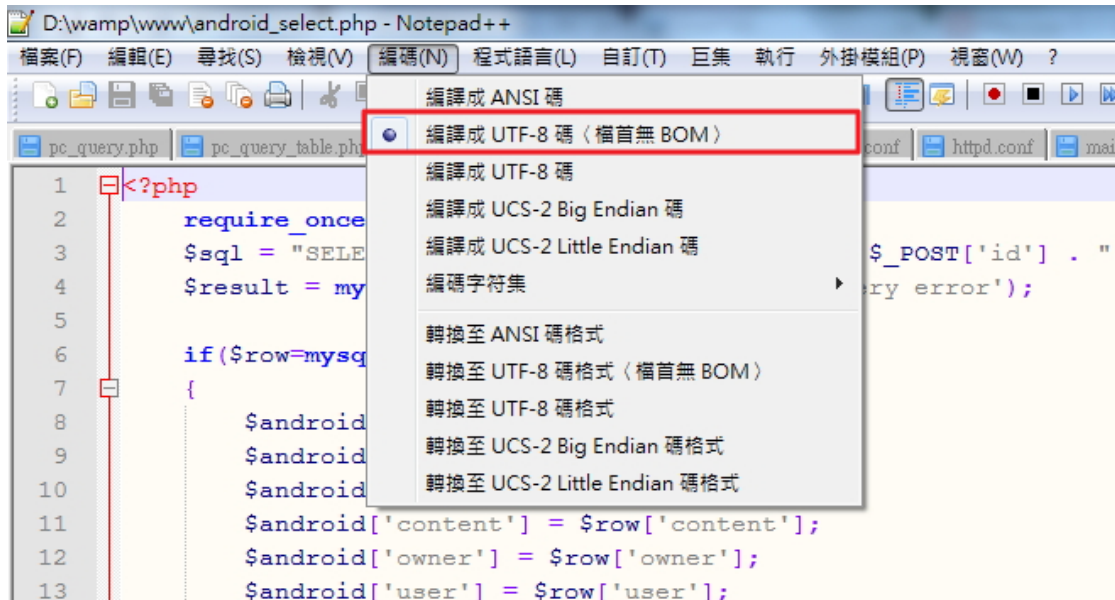


圖 5-17 將網頁以 UTF-8 (無 BOM) 方式建立在伺服器端

而網頁在連資料庫時，必須加入以下的指令，確保兩邊的編碼一致。

```
mysql_query("SET CHARACTER SET 'UTF8'");  
mysql_query('SET NAMES UTF8;');  
mysql_query('SET CHARACTER_SET_CLIENT=UTF8;');  
mysql_query('SET CHARACTER_SET_RESULTS=UTF8;');
```

圖 5-18 設定與資料庫連線使用 UTF-8 的 PHP 指令

f. 手機網路功能

手機部分，為了使用網路存取的功能，除了要使用有上網功能的手機之外，要讓程式能夠處理使用網路的存取，必須先在程式功能描述檔 `AndroidManifest.xml` 中完成以下設定：

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

圖 5-19 加入手機可以存取網路設定

必須使用網頁的 POST 方式傳到伺服器，所以在手機程式中必須使用 JAVA 裡模擬 HTTP 的 CLASS。參考圖 5-20。

```
HttpPost hp = new HttpPost(delete); // 使用HTTP_POST的物件與網頁連線(刪除)
String error = "成功連線!!";
List <NameValuePair> params = new ArrayList <NameValuePair>(); // 宣告要傳的POST參數的陣列
params.add(new BasicNameValuePair("id", id)); // 將POST參數與值加入到陣列

try {
    final HttpParams httpParams = new BasicHttpParams();
    HttpConnectionParams.setConnectionTimeout(httpParams, 3000);
    // 設定網頁連線的等待時間

    UrlEncodedFormEntity urf = new UrlEncodedFormEntity(params,HTTP.UTF_8);
    hp.setEntity(urf);
    // 設定傳送資料的編碼

    HttpResponse hr = new DefaultHttpClient(httpParams).execute(hp);
    // 將手機資料送出並取得網頁回應資料

    if(hr.getStatusLine().getStatusCode() == HttpStatus.SC_OK) { // 判斷網頁是否回應成功
        delete_str = EntityUtils.toString(hr.getEntity(), "UTF-8");
        // 網頁回應資料編碼設定
    }
    else {
        error = hr.getStatusLine().toString();
    }
}
```

圖 5-20 POST 方式連線伺服器

但是後來發現，Android 主程序主要處理 UI 的程序，一但執行到 HTTP POST 的程序時，會發現主程序資源全部都被占用，造成 UI 當機、沒有回應過久，使得程式自動關閉或是執行錯誤，必須在程式的 onCreate 裡加入以下指令

```
StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().
    detectDiskReads().detectDiskWrites().detectNetwork().penaltyLog().build());
StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder().
    detectLeakedSqlLiteObjects().penaltyLog().penaltyDeath().build());
```

圖 5-21 開啟網路上傳功能

或是使用 android 裡的多執行緒方式 HandlerThread、Handler、Runnable 方式處理，同一個時間做不同事情，否則在 Android 當中，如果做超過 5 秒被系統強制關閉(收到 Application not Responed 簡稱 ANR)；onCreate()如果做超過 10 秒就會跳 ANR。

而手機如何取得 PHP 網頁在 MYSQL 執行結果資料獲得回應，因為如果不使用一些方式的話，只有回傳單純文字，會造成手機程式或是伺服器端要花很多力氣去處理資料的組合或是辨識方式。後來使用了在網頁程式設計中(PHP、XML、AJAX)等一些工具書有提到的 JSON。

在伺服端的網頁中，宣告陣列，將查詢 MYSQL 結果放入陣列裡面，最後再只用 PHP 內建 json_encode 函式將陣列包裝成 JSON 格式並送出。參考圖 5-21。

```
$android = array();
$android['nfc_id'] = $row['nfc_id'];
$android['name'] = $row['name'];
$android['content'] = $row['content'];
$android['owner'] = $row['owner'];
$android['user'] = $row['user'];
$android['price'] = $row['price'];
$android['place'] = $row['place'];
$android['buy_date'] = $row['buy_date'];
$android['check_date'] = $row['check_date'];
echo json_encode($android);
```

圖 5-22 PHP 將資料以 JSON 方式送出

JSON(JavaScript Object Notation) 是一種羽量級的資料交換格式。易於人閱讀和編寫。同時也易於機器解析和生成。它基於 **JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999** 的一個子集，所以可以在 JavaScript 以 `eval()` 函式 (javascript 通過 `eval()` 調用解釋器) 讀入。

不過這並不代表 JSON 無法使用於其他語言，事實上幾乎所有與網頁開發相關的語言都有 JSON 函式庫。

JSON 採用完全獨立于語言的文本格式，但是也使用了類似於 C 語言家族的習慣 (包括 C、C++、C#、Java、JavaScript、Perl、Python 等)。這些特性使 JSON 成為理想的資料交換語言。

六、專題製作成果之特色

以下為本專題之特色與缺點：

1、特色

1. 比起原始盤點方法增加盤點速度
2. 使用者操作介面簡潔
3. NFC Tag 比起原始標籤不易損壞
4. 應用資料庫系統管理資料
5. 後端網頁管理介面方便查詢資料
6. 應用 NFC 技術達到快速盤點功能
7. 應用 WIFI 技術達到無線傳送

2、缺點

1. 應用上需要網路環境
2. 成本相對較高

七、未來展望

我們專題是在智慧型手機上實作一個 NFC 盤點 APP，由於網路環境需求度高，所以我們之後想使用智慧型手機上的資料庫，在無網路環境時進行盤點資料的儲存，當偵測到網路環境時再進行上傳，讓伺服器的資料庫更新最新狀況。也能將所要盤點的資料下載到手機的資料庫裡，在無網路環境時能進行盤點。

資料傳送的過程中，為了確保資料安全，打算加入 AES 加密演算法來應用。再來就是成本高的問題，雖然目前智慧型手機價格上比起一般手機還是偏高，但之後會越來越普遍，最後就是將 APP 設計得更加人性化，希望能有更多人能使用。

八、結論

這次的專題讓我們學到該怎麼分工合作、遇到問題時要如何解決，過程中的討論，就像是一個出社會工作的縮影，提前能感受到一個專案的流程。

另一方面，由於智慧型手機使用到的是 Android 環境，網路上的資源還尚未完整，增加了我們查找資料的經驗與面對新東西該有的心態，解決遇到的問題。

九、銘謝

首先要感謝專題指導教授吳林全教授，每週都抽出一點時間來跟我們討論，為我們講解人員規劃、工作分配該怎麼做，每個階段該怎麼計畫，要怎麼做，如何時間內完成，遇到錯誤時也會仔細的為我們講解。另外還要感謝小組成員之間的配合，一起解決問題。

十、參考文獻

- <http://zh.wikipedia.org/wiki/%E8%BF%91%E5%A0%B4%E9%80%9A%E8%A8%8A>
- <http://www.nfc-forum.org/home/>
- <http://developer.android.com/guide/topics/connectivity/nfc/index.html>