

致理技術學院

商務科技管理系 實務專題報告

APP 歌唱點播系統

指導老師：林正平 老師

學生：楊柏岳 19933109

張志銘 19933119

黃子龍 19933130

張晉嘉 19933149

林禹均 19933159

中華民國：102 年 12 月

致理技術學院

商務科技管理系
實務專題報告

APP 歌唱點播系統

學生：楊柏岳 19933109

張志銘 19933119

黃子龍 19933130

張晉嘉 19933149

林禹均 19933159

本成果報告書經審查及口試合格特此證明

指導老師：_____

中華民國：102 年 12 月

CTM 實務專題研究授權書

本授權書所授權之實務專題研究為_____

_____共_____

____人，在致理技 術學院商務科技管理系_____學年度第_____學期完

成商管實務專題。 商管實務專題名稱：_____

同意 不同意 本組同學共_____人，皆同意著作財產
權之論文全文資料，授予教育部指 定送繳之圖書館及本人畢業
學校圖書館，為學術研究之目的以各種方法重製， 或為上述目
的再授權他人以各種方法重製，不限地域與時間，惟每人以一
份為 限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行
權為非專屬性發 行權利。依本授權所為之收錄、重製、發行及
學術研發利用均為無償。上述同 意與不同意之欄位若未鉤選，
該組同學皆同意視同授權。

指導教授姓名：

專題生簽名：

學號：（親筆正楷）（務必填寫）

中華民國 年 月 日

誌 謝

關於這次專題研究開發此「APP 歌唱點播系統」，過程中很感謝老師的協助與幫忙，給予我們解答程式或資料庫上的問題，使我們更容易達成此系統的開發。還有感謝小組同學，分工合作努力不懈把此系統開發完成，並且藉由這次實務專題開發 APP 系統的應用程式，讓我們更深入學習認識到程式 C#語言、Java 語言上的應用。

摘要

隨著時代的進步，手機功能除了基本的接聽功能，拍照、錄音、3G 網路、體感裝置等功能可以說是一應俱全，手機的多媒體功能越來越強大，而現代人把手機用做娛樂用途的比例也越來越高。

由於目前台灣的消費生活習慣，對生活娛樂已經產生很大的重要性，也勢必對娛樂產業上帶來不少經濟發展，不像以前只重視於食、衣、住、行之上，娛樂也帶給我們紓解壓力、休閒娛樂、以及滿足消費者的需求渴望。

科技發達日新月異，現在智慧型手機已經成為人手一隻的必需品，APP 應用程式已經相當活絡，快速成為生活重要部分；因此專題研究將以 Android 開發系統架構做為一個雛形為基準。

本研究藉由撰寫「APP 歌唱點播系統」為主，此系統透過手機使用本應用程式，能達到完成歌唱點播的方便性及即時性，並認知「APP 歌唱點播系統」給與結論。

關鍵字：APP 歌唱點播系統、Android、Visual、Eclipse

目錄

授權書.....	I
誌謝.....	II
摘要.....	III
目錄.....	IV
圖目錄.....	V
表目錄.....	VI
第一章 緒論.....	1
第一節 研究背景與動機.....	1
第二節 研究目的(預期效益).....	1
第三節 研究流程.....	2
第四節 研究限制.....	2
第二章 系統分析.....	3
第一節 系統簡介.....	3
第二節 系統開發工具.....	3
第三節 系統功能.....	5
第四節 系統特色.....	5
第五節 系統使用環境.....	6
第三章 系統設計.....	7
第一節 研究方法.....	7
第二節 系統架構.....	7
第三節 系統設計.....	8
第四節 使用對象.....	17
第四章 系統成果.....	18
第一節 系統流程.....	18
第二節 系統畫面.....	19
第五章 結論與建議.....	34
第一節 研究結論.....	34
參考文獻.....	35
附錄一、主機端程式碼.....	36
附錄二、手機端程式碼.....	54

圖目錄

圖 1 研究流層.....	2
圖 2 系統架構圖	7
圖 3 主機端架構設計.....	8
圖 4 主機端(PC)讀取資料庫.....	8
圖 5 主機端(PC)點歌測試.....	9
圖 6 主機端(PC)搜尋功能測試.....	10
圖 7 主機端架構設計.....	11
圖 8 手機端架構設計.....	15
圖 9 手機端測試-顯示資料庫資料.....	16
圖 10 系統流程圖	18
圖 11 點歌機起始畫面.....	19
圖 12 點歌機使用點選搜尋台語類別.....	20
圖 13 使用類別搜尋點播歌曲至清單.....	21
圖 14 選擇點播的搜尋方式.....	22
圖 15 點歌機使用點播搜尋的注音功能(一).....	23
圖 16 點歌機使用點播搜尋的注音功能(二).....	24
圖 17 使用注音搜尋點播至清單.....	25
圖 18 開始播放清單內之歌曲	26
圖 19 APP 歌唱點播系統主畫面.....	27
圖 21 APP 歌手筆畫搜尋結果.....	29
圖 22 APP 男歌手搜尋結果.....	30
圖 23 APP 筆畫搜尋男歌手結果.....	31
圖 24 APP 點選歌曲至點播清單.....	32
圖 25 APP 輸入正確查詢條件.....	33

表目錄

表 1 使用環境開發程式設備.....	6
表 2 使用環境-手機設備.....	6

第一章 緒論

本研究以探討唱點播系統為主要參考依據指標，藉由台灣各家競爭業者，來探討改善使用更完善的 APP 歌唱點播系統。故於本章內容分別加以說明研究動機、研究目的、研究流程，透過本章各節論述提點本研究之必要性。

第一節 研究背景與動機

以 Android 智慧型手機的發展來看，越多人採用 Android 智慧型手機，代表有越多人對 Android 作業系統所開發的應用軟體的需求；再者 Google 公司有提供一個 Android Market 的軟體銷售平台，可以讓應用程式開發者所開發的應用程式軟體，有一個統一且全世界用戶皆可看到即進行購買的平台。

有鑑於目前安卓系統智慧型手機的市場逐步攀升，也促成使用率逐步增加，對於喜歡使用智慧型手機的消費者，在 APP 應用程式大部分已經暢行無阻。故本研究之 KTV 點歌僅受限於點歌機上點歌操控；因此藉由此實務專題以 APP 歌唱點播系統為主，以改善目前受限在點歌機上操控的單方操作或是遙控插播，將轉變成同時多方的即時性點歌。

科技發達日新月異，現在智慧型手機已經成為人手一機的必需品，APP 應用程式已經相當活絡，快速成為生活重要部分；因此專題研究將以一個 APP Android 的應用程式系統做為一個雛形為基準。

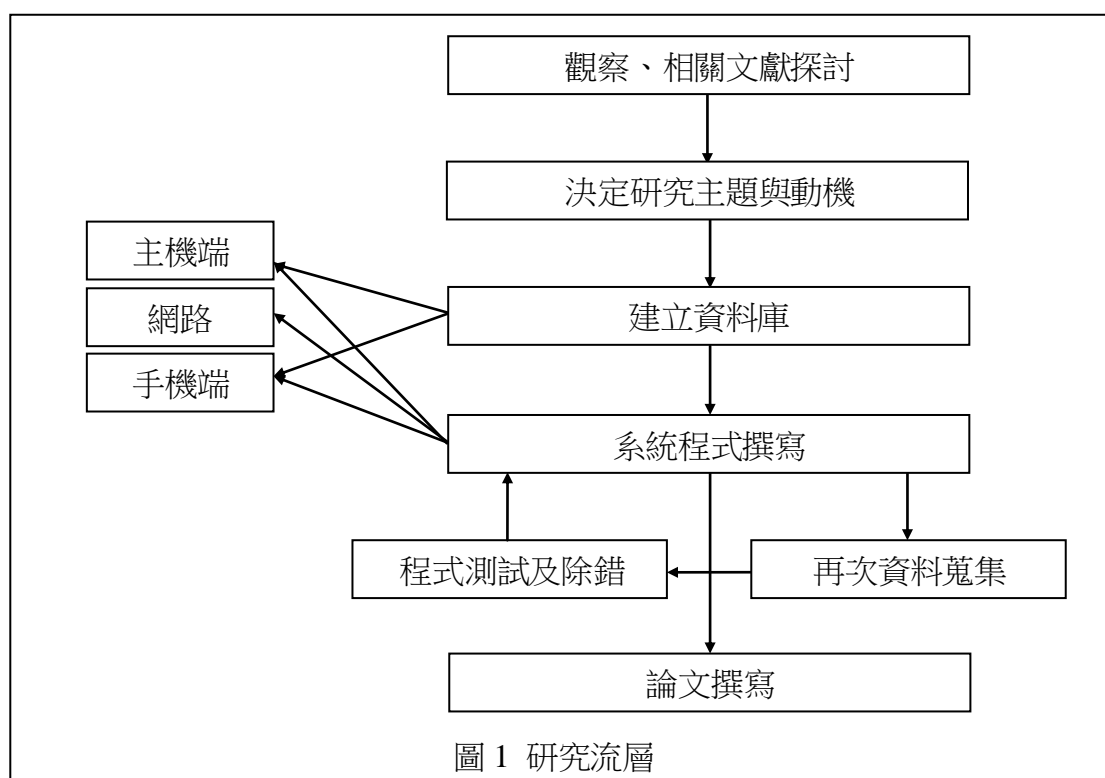
第二節 研究目的(預期效益)

本研究以探討 KTV 歌唱點播系統為主要參考依據指標，探討內容包含 APP 歌唱點播系統的各项內建配置及雛形建置。因此本研究依據上述研究動機，研究目的如下：

- 一、達到增加歌曲點播次數及便利性。
- 二、達到結合智慧型手機的實務應用。

第三節 研究流程

本研究流程如下圖(1-3-1)所示，首先從觀察以及相關文獻探討提出研究主題。第二部分進行相關文獻之蒐集與探討。第三部分進行研究程式資料庫建置設計，提出有效研究方法與研究架構。第四部分針對 APP 歌唱點播系統做完整基本建置，過程中有測試與除錯或是再次文獻探討達成本系統雛形的建置完成。



第四節 研究限制

專題研究開發此「APP 歌唱點播系統」，關於在硬體設備上無法使用到與 KTV 產業相同的點歌機；因此本研究開發將以電腦架設為點歌機的主機端，來替代點歌機開發此 APP 應用程式。

第二章 系統分析

第一節 系統簡介

開發此 APP 歌唱點播系統，主要提升在 KTV 消費歌唱點播的消費者，或是使用者能夠享有更便利的點播歌曲空間，達到智慧型手機轉變成遙控器的替代功能性，改變以前一對一的點歌機與遙控插播歌曲的傳統限制，享受更高科技的歌場點播服務，並讓智慧型手機達到替代功能。

第二節 系統開發工具

一、 Android 介紹

Android 是一個開放性作業系統，採用 Apache-2.0 授權條款，該授權條款允許廠商免費取得 Android 作業系統，如此可省去軟使用授權費，產品的成品可降低，並提升產品的競爭力；另外，廠商可以改寫原始程式碼，修改或自行開發的程式碼，廠商可以自己擁有，不需再開放出來，因此廠商的利益得以保障，如此也可以促成廠商願意在 Android 作業系統的基礎上繼續投入研發，提供自己產品的特有功能；再者 google 公司持續對 Android 作業系統的更新且快速，功能與日具增的提升，如此使得廠商更願意採用 Android 作業系統作為其手持設備之作業系統，到至今預估全球會有超過 50% 智慧型手持設備皆採用 Android 作業系統。

Android 以 Linux 為核心的 Android 行動平台，使用 Java 作為程式語言，使介面到功能，都有層出不窮的變化，其中 Activity 等同於 J2ME 的 MIDlet，一個 Activity 類別負責建立視窗，一個活動中的 Activity 就是在 foreground（前景）模式，背景執行的程式叫做 Service。兩者之間透過 ServiceConnection 和 AIDL 連結，達到複數程式同時執行的效果。如果執行中的 Activity 全部畫面被其他 Activity 取代時，該 Activity 便被停止，甚至被系統清除。

View 等同於 J2ME 的 Displayable，程式人員可以透過 View 類別與「XML layout」檔將 UI 放置在視窗上，並可以利用 View 打造出所謂的

Widgets，其實 Widget 只是 View 的一種，所以可以使用 xml 來設計 layout。至於 ViewGroup 是各種 layout 的基礎抽象類別，ViewGroup 之內還可以有 ViewGroup。View 的建構函式不需要在 Activity 中呼叫，但是 Displayable 的是必須的，在 Activity 中，要透過 findViewById() 來從 XML 中取得 View，Android 的 View 類的顯示很大程度上是從 XML 中讀取的。View 與事件息息相關，兩者之間透過 Listener 結合在一起，每一個 View 都可以註冊 event listener，例如：當 View 要處理使用者觸碰的事件時，就要向 Android 框架註冊 View.OnClickListener。另外還有 Image 等同於 J2ME 的 BitMap。

二、Visual studio

Microsoft Visual Studio (簡稱 VS) 是美國微軟公司的開發工具套件系列產品。VS 是一個基本完整的開發工具集，它包括了整個軟體生命週期中所需要的大部分工具，如 UML 工具、代碼管控工具、整合式開發環境(IDE) 等等。所寫的目的碼適用於微軟支援的所有平台，包括 Microsoft Windows、Windows Phone、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight。而 Visual Studio .NET 是用於快速生成企業級 ASP.NET Web 應用程式和高效能桌面應用程式的工具。Visual Studio 包含基於元件的開發工具(如 Visual C#、Visual J#、Visual Basic 和 Visual C++)，以及許多用於簡化基於小組的解決方案的設計、開發和部署的其他技術。

三、Eclipse

Eclipse 是著名的跨平台開源整合式開發環境(IDE)。最初主要用來 Java 語言開發，目前亦有人透過外掛模組使其作為 C++、Python、PHP 等其他語言的開發工具，Eclipse 的本身只是一個框架平台，但是眾多外掛程式的支援，使得 Eclipse 擁有較佳的靈活性。許多軟體開發商以 Eclipse 為框架開發自己的 IDE。

Eclipse 是開放原始碼的專案，你可以到 www.Eclipse.org 去免費下載 Eclipse 的最新版本，一般 Eclipse 提供幾個下載版本：Release、Stable Build、Integration Build 和 Nightly Build。Eclipse 本身是用 Java 語言撰寫，但下載的壓縮包中並不包含 Java 執行環境，需要用戶自己另行安裝 JRE，並且要在作業系統的環境變數中指明 JRE 中 bin 的路徑，安裝 Eclipse 的步驟非常簡單：只需將下載的壓縮包按原路徑直接解壓既可。需注意如果有了更新的版本，要先刪除老的版本重新安裝，不能直接解壓到原來的路徑覆蓋老版本，在解壓縮之後可以到相應的安裝路徑去找 Eclipse.exe 執行。Eclipse

可由額外的附加功能擴充，目前已有幾個 open source 的計劃案與公司，將額外的附加元件擴充至 Eclipse 中，例如：Android 開發。Eclipse 也可被用在一般性用途的應用程式開發上。

四、資料庫 SQLite

SQLite 是個輕量化的關聯式資料庫，它輕薄短小，無須設定或管理，沒有伺服器及組態檔，不需要資料庫管理員，它只是一個檔案，可以依須求四處移動，對移動裝置來說是非常好用的資料庫。使用 SQLite 時，你並不用先去建立資料庫檔案，而由 Android 幫你建立。你必須繼承 SQLiteOpenHelper 類別，由它來負責管理資料庫的建立和版本控制。你必須覆寫 onCreate 及 onUpgrade 方法，當你第一次使用資料庫時(查詢或新增)，若 Android 找不到資料庫檔案，就會觸發 onCreate 方法來建立，若已經有檔案了，則不再建立。當資料庫結構有更新時，即版本號有更改，就會觸發 onUpgrade 方法來更新資料庫，可以砍掉重建，也可直接更新結構。

第三節 系統功能

「APP 歌唱點播系統」主要透過手機使用本應用程式，就能達到點歌的效果。本應用程式 APP 歌唱點播系統，歌曲功能包括著：點播歌曲功能、歌曲搜尋功能、筆畫搜尋功能、歌曲語言搜尋功能、男歌手搜尋功能、女歌手搜尋功能、團體搜尋功能等。

第四節 系統特色

開發此手機安卓系統的 APP 歌唱點播系統，主要以提升改善在 KTV 歌唱娛樂的消費者，能夠享有更便利的點歌服務，來享受高歌一曲的娛樂。另外還包括主機端與網路共同與手機端達成一連貫的系統。本系統手機功能可以提供使用者達到透過手機本 APP 歌唱點播系統的應用程式，就可以達到輕鬆點歌的功能。

第五節 系統使用環境

主機端：電腦設備，開發此 APP 歌唱點播系統的點歌主機端。

項目	內容
電腦名稱	ST-15
作業系統	Windows7 旗艦版 64 位元(6.1，建組 7601)
語言	中文(繁體)(區域設定：中文(繁體))
系統製造商	System manufacturer
系統型號	System Product Name
BIOS	BIOS Date : 04/17/09 11:12:17 Ver :08.00.14
處理氣	Intel(R)Core(TM) 2QuadCPU Q9400 @2.66GHz(4 CPUs) ,~2.7GHz
記憶體	2048MB RAM
分頁檔案	使用了 1848MB，還有 2177MB 可用
Direct 版本	DirectX 11

表 1 使用環境開發程式設備

手機端：手機設備，開發 APP 歌唱點播系統之應用程式

頻率	GSM、WCDMA、HSDPA 21Mbps、HSUPA 5.76Mbps
處理器	NVIDIA Tegra 3、時脈 1.5GHz、4 核心
記憶體	1GB DRAM
儲存	內建 32GB、Dropbox 25GB 兩年免費、無記憶卡插槽
螢幕	4.7 吋、解析度 1280 × 720、IPS、電容式觸控
鏡頭	後鏡頭 800 萬像素 (光圈 F/2.0)、LED 補光燈、內建獨立影像晶片 (支援 HDR、1080p 影片拍攝); 前鏡頭 130 萬像素
無線	Wi-Fi 802.11 a/b/g/n、藍牙 4.0(支援 aptx)、GPS(支援 aGPS)
電池	1,800mAh
機身顏色	機身顏色：白
作業系統	Android 4.0.3 (Ice Cream Sandwich)
尺寸	134.36 × 69.9 × 8.9mm
重量	約 130g (含電池)

表 2 使用環境-手機設備

第三章 系統設計

第一節 研究方法

在「APP 歌唱點播系統」程式系統撰寫過程中，將以透過第一章節(圖 1)研究流程的順序來進行此系統建置。主要分成三個程式端，包括：主機端、網路、手機端。從使用手機本應用程式點歌透過網路傳送，抵達主機端系統回傳資料，再由主機端透過網路回傳至手機端通知點歌成功訊息。

第二節 系統架構

主要程式架構分為三個部分，有主機端、網路、手機端。使用者透過本 APP 歌唱點播系統操作，如下(圖 2)架構，經由使用手機應用程式，透過網路相互讓二系統端傳送資料，再給予使用者回饋成功訊息。而資料庫經由 Microsoft Access 撰寫，再由各程式端所需，再轉換成適當的資料庫。則資料庫經由 Microsoft Access 撰寫，由各程式端所需再轉換資料庫。

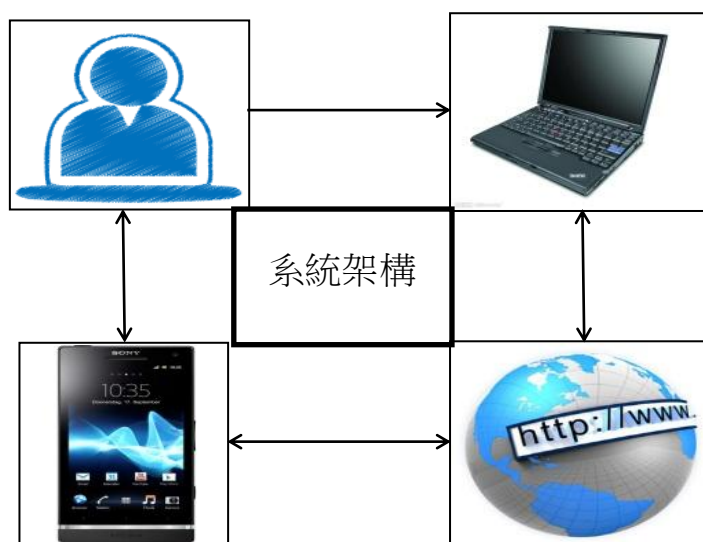


圖 2 系統架構圖

第三節 系統設計

「APP 歌唱點播系統」以分為以下三個型態，經由主機端的基礎，皆由網路架設資料互相回傳，最後透過安卓撰寫手機端達成一連串系統的應用程式。將分成以下三個程式端：

一、主機端

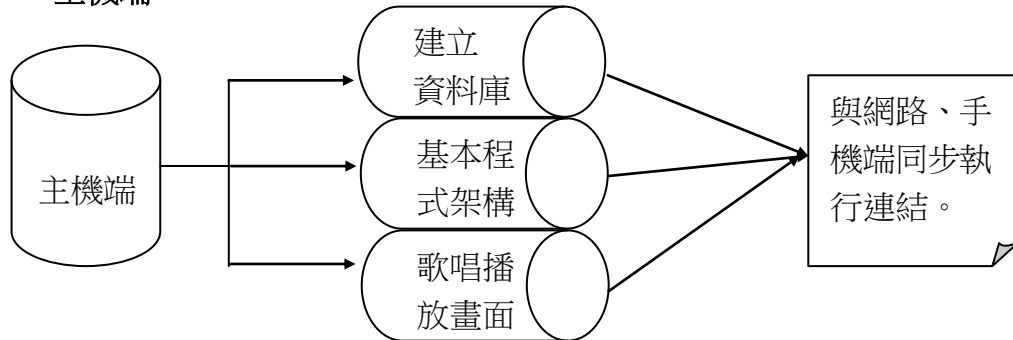


圖 3 主機端架構設計

主機端也就是電腦端。是三個程式的基本系統。主機端包含內部所有相關聯的重要作業，如同包括：資料庫、KTV 影片播放、已點歌排序、歌單查詢等，就如同 KTV 本身點歌機的相同功能型式。

(一)、從資料庫取出性別=男的資料顯示在畫面上

```
string sql = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱 FROM 歌單  
where 性別 ='1'";  
dataGridView2.DataSource = ds;  
dataGridView2.DataMember = "歌單";
```



圖 4 主機端(PC)讀取資料庫

(二)、按下按鍵可以把點選的歌曲顯示在點歌清單內



TabControl 控件顯示多個選項，這些選項類似於筆記本中的分隔卡和檔案具文件夾中的標籤。

```
private void button1_Click(object sender, EventArgs e){
    if (tabControl1.SelectedIndex == 0){
        Int32 selectedRowCount =
dataGridView2.Rows.GetRowCount(DataGridViewElementStates.Selected);
        if (selectedRowCount > 0){
            System.Text.StringBuilder sb = new
System.Text.StringBuilder();
            listBox1.Items.Add(sb.ToString())}}}
```



圖 5 主機端(PC)點歌測試

(三)、選取搜尋方式並輸入資料

```
string a = comboBox1.Text;  
string sql = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱 FROM 歌單  
where " + a + " like '%" + textBox1.Text + "%'";
```



圖 6 主機端(PC)搜尋功能測試

二、網路

(一)網路架構

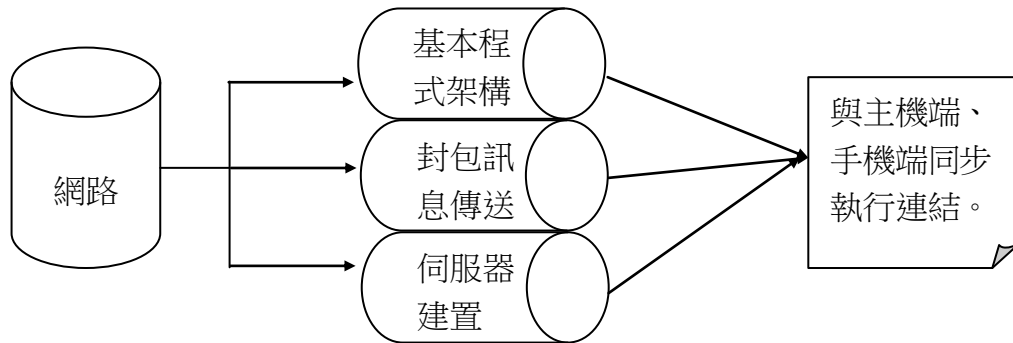


圖 7 主機端架構設計

網路亦是三個程式的歌曲點歌資料相互回傳。網路內含有其兩者內部所有相關有相關聯的重要作業，如同包括：手機已點過歌曲要給予點歌成功訊息、回傳給予主機端歌曲的點歌作業、通知主機端與手機端兩者之間的對傳功能，相關以(圖 8-12)可參見網路程式達到互相的回傳作業。

(二)網路測試

用以傳送訊息之 TCP Client 端輸入將傳送訊息的 IP:port , Connecting done 即連線成功

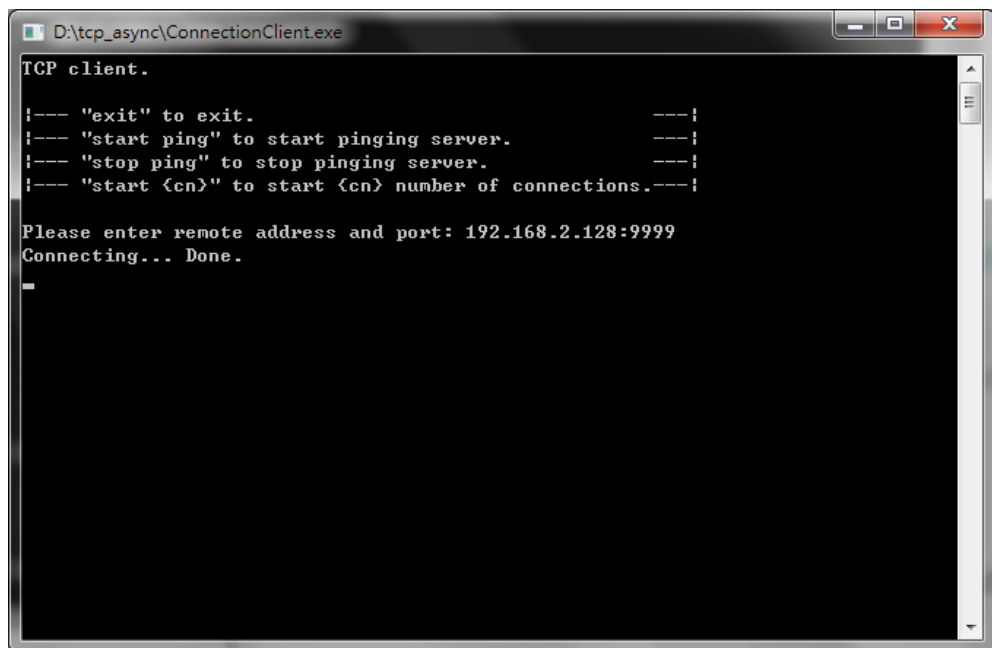
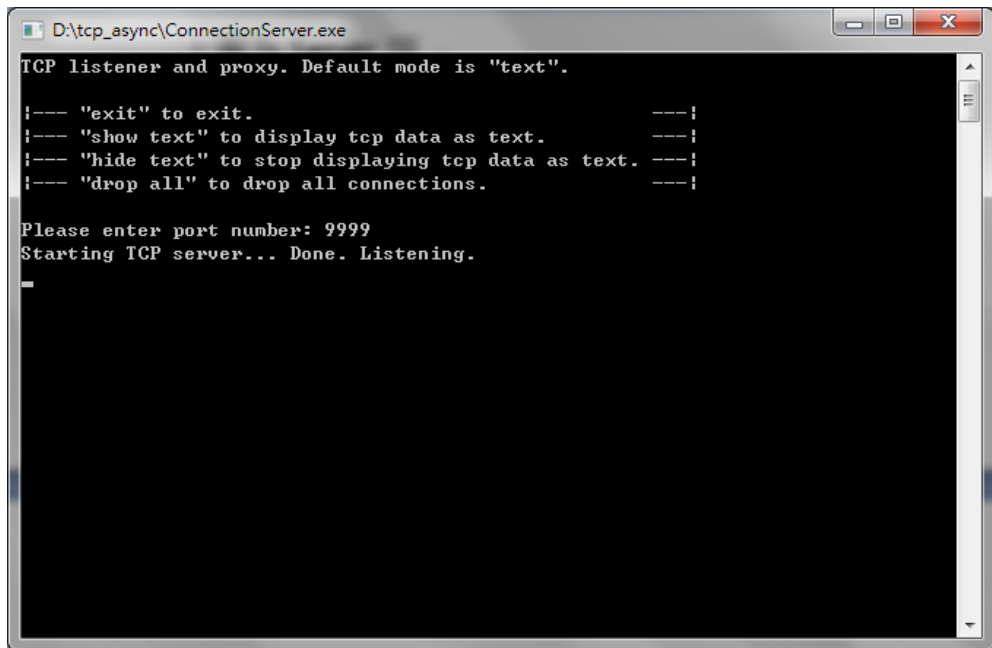


圖 8 網路測試起始畫面

下圖為 TCP Server 端輸入監聽 port , Server 開始監聽



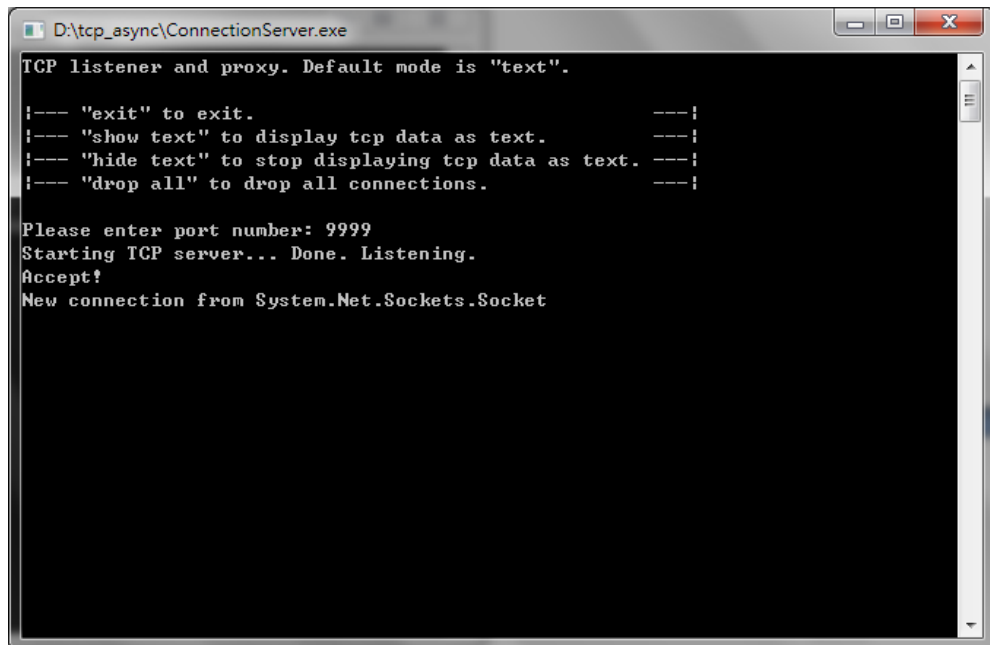
```
D:\tcp_async\ConnectionServer.exe
TCP listener and proxy. Default mode is "text".

!--- "exit" to exit. ---!
!--- "show text" to display tcp data as text. ---!
!--- "hide text" to stop displaying tcp data as text. ---!
!--- "drop all" to drop all connections. ---!

Please enter port number: 9999
Starting TCP server... Done. Listening.
```

圖 9 伺服器監聽(1)

Server 端監聽到 Client Socket , 可以開始接收訊息



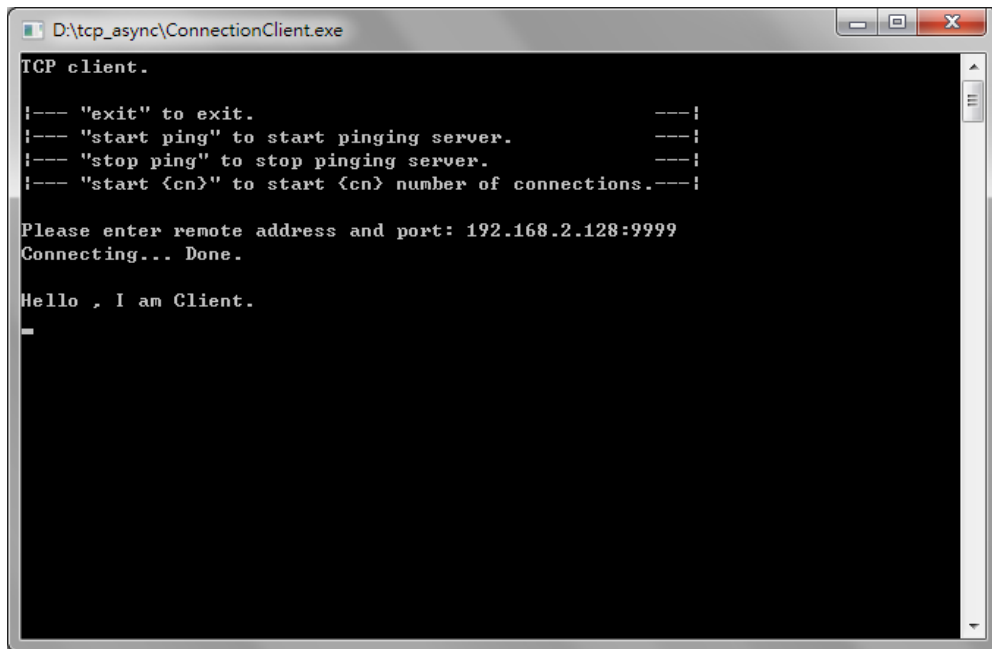
```
D:\tcp_async\ConnectionServer.exe
TCP listener and proxy. Default mode is "text".

!--- "exit" to exit. ---!
!--- "show text" to display tcp data as text. ---!
!--- "hide text" to stop displaying tcp data as text. ---!
!--- "drop all" to drop all connections. ---!

Please enter port number: 9999
Starting TCP server... Done. Listening.
Accept!
New connection from System.Net.Sockets.Socket
```

圖 10 伺服器監聽(2)

輸入訊息 Enter 送出給 Server 端



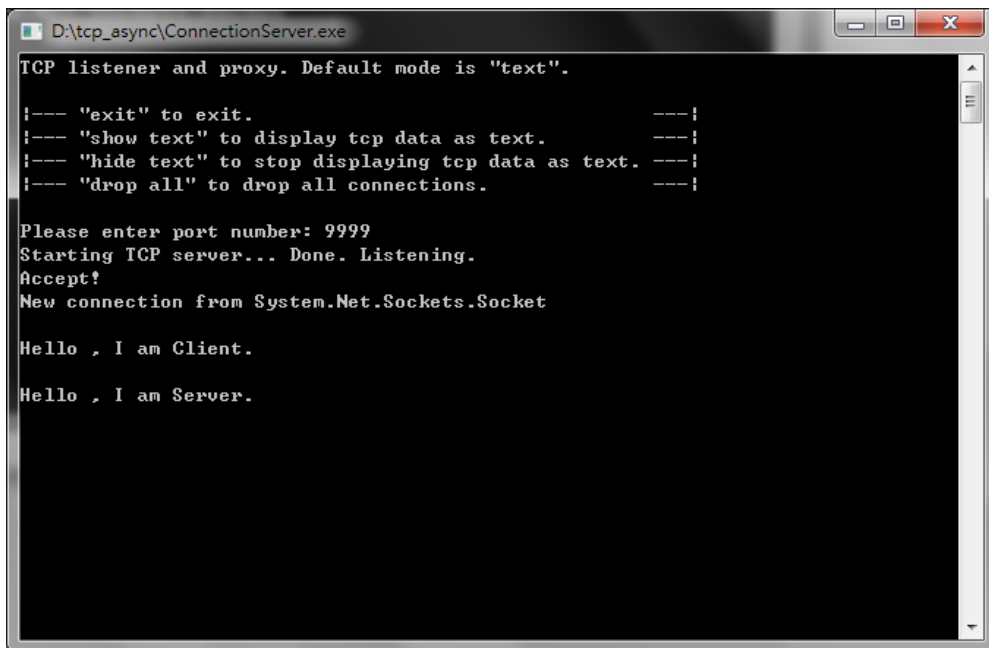
```
D:\tcp_async\ConnectionClient.exe
TCP client.
!--- "exit" to exit. ---!
!--- "start ping" to start pinging server. ---!
!--- "stop ping" to stop pinging server. ---!
!--- "start <cn>" to start <cn> number of connections.---!

Please enter remote address and port: 192.168.2.128:9999
Connecting... Done.

Hello , I am Client.
=
```

圖 11 Client 輸入資料

Server 端接收到 Client 端所傳送之訊息 Server 端也可把訊息送回給 Client 端。



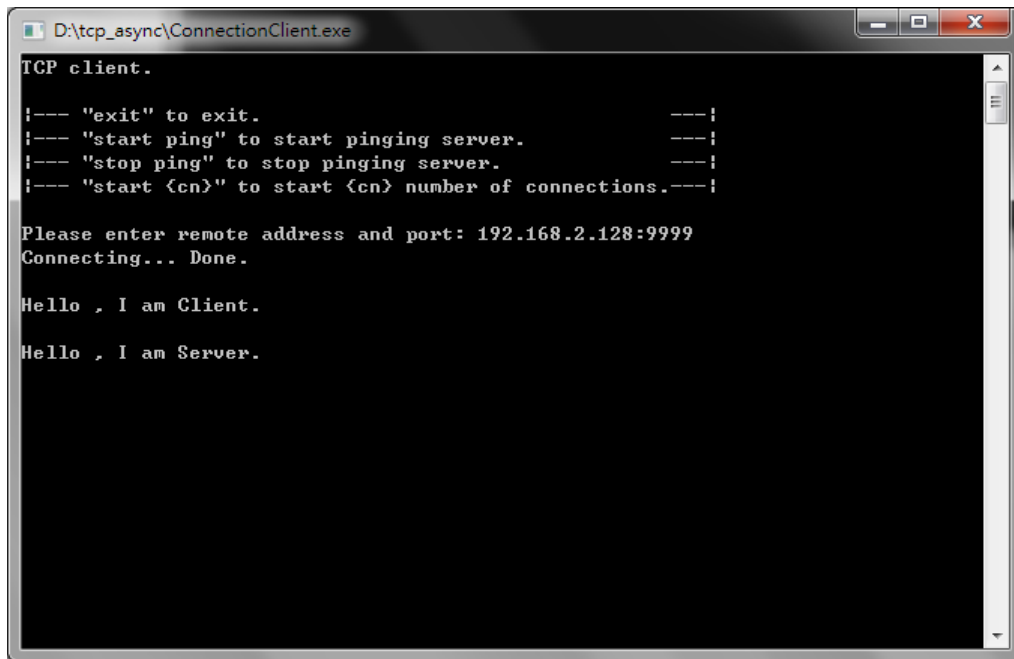
```
D:\tcp_async\ConnectionServer.exe
TCP listener and proxy. Default mode is "text".
!--- "exit" to exit. ---!
!--- "show text" to display tcp data as text. ---!
!--- "hide text" to stop displaying tcp data as text. ---!
!--- "drop all" to drop all connections. ---!

Please enter port number: 9999
Starting TCP server... Done. Listening.
Accept!
New connection from System.Net.Sockets.Socket

Hello , I am Client.
Hello , I am Server.
```

圖 11 Server 接收資料並回傳

此為收到 Server 回傳之訊息的 Client 端畫面。



```
D:\tcp_async\ConnectionClient.exe
TCP client.
|--- "exit" to exit. ---|
|--- "start ping" to start pinging server. ---|
|--- "stop ping" to stop pinging server. ---|
|--- "start <cn>" to start <cn> number of connections.---|

Please enter remote address and port: 192.168.2.128:9999
Connecting... Done.

Hello , I am Client.
Hello , I am Server.
```

圖 12 Client 接收 Server 回傳資料

三、手機架構

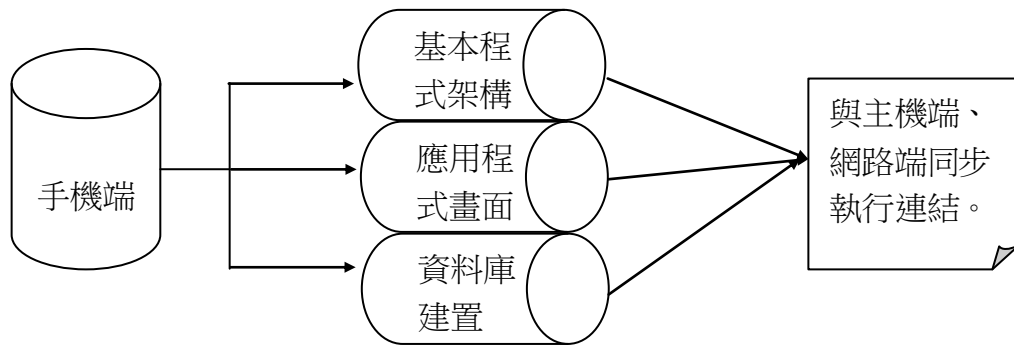


圖 8 手機端架構設計

手機端亦是三個程式的興新重要產品。手機端主要面對就是消費者，所以應用程式的風評好與壞，一切都交付於手機應用程式，主要包括：與主機端歌曲查詢的資料同步建置、歌單查詢功能、點歌成功訊息通知等，就如同是隨行方便的主機端。

以下程式可以達到搜尋的功能，假設使用者輸入周，可以找到所有姓「周」的歌手資料

```
if (v == btnCreateDB) {
    CreateDB();
    table.gvRemoveAll();
    InsertRecord(16);
    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME+ " where "+a+" like "+ "%"+s+"%",db);
    table.gvReadyTable("select musicnumber,name,song from "
+ TABLE_NAME+ " where "+a+" like "+ "%"+s+"%",db);
}
```

以下程式是可以在按下按鈕之後，顯示出所有的男歌手資料；團體、女歌手等等都是依此類推

```
else if (v == btnboy)
{
    table.gvRemoveAll();
    InsertRecord(16);
    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+1,db);
    table.gvReadyTable("select musicnumber,name,song from "
+ TABLE_NAME + " where sex="+1,db);
}
```

以下程式碼主要主要是用程式碼設置 GridView，以顯示所有的資料內容

```
gvPage=new GridView(context);
    gvPage.setColumnWidth(40);//設置每個分頁按鈕的寬度
    gvPage.setNumColumns(GridView.AUTO_FIT);//分頁按鈕數量自動設置
    addView(gvPage, new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
    LayoutParams.WRAP_CONTENT));
    srcPageID = new ArrayList<HashMap<String, String>>();
    saPageID = new SimpleAdapter(context,
        srcPageID, R.layout.items,
        new String[] { "ItemText" },
        new int[] { R.id.ItemText });
    gvPage.setAdapter(saPageID);
    gvPage.setOnItemClickListener(new OnItemClickListener(){
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
    long arg3) {LoadTable(arg2);//根據所選分頁讀取對應的數據
        if(switchListener!=null){//分頁切換時
            switchListener.onPageSwitchListener(arg2,srcPageID.size()); } } });
```

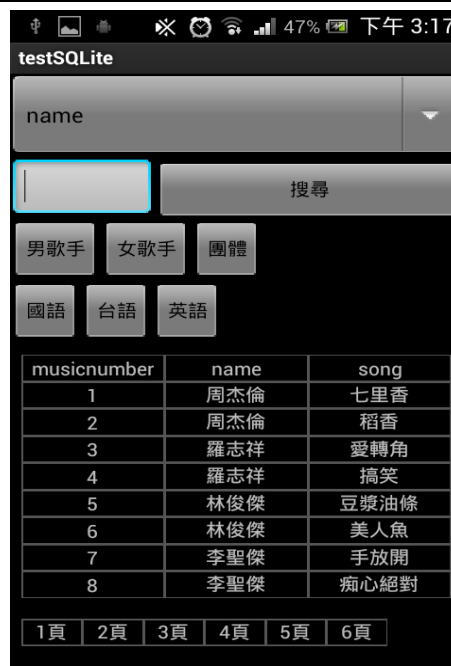


圖 9 手機端測試-顯示資料庫資料

第四節 使用對象

針對現在娛樂產業至於 KTV 歌唱的；好樂迪、星據點、錢櫃等。對消費者已經是生活上基本的生活娛樂消費。對於此開發應用程式之「APP 歌唱點播系統」，結合現在智慧型手機的實務應用，來提升歌唱點播的次數及便利性，打破傳統只能面對點歌機一對一操作，或是翻查歌本的遙控插播的單方操作，將轉變成與點歌機相同功能的「APP 歌唱點播系統」。

此「APP 歌唱點播系統」應用程式，主要替在於 KTV 歌唱消費之使用者，能夠享有更佳便利、更即時性的點歌空間，達到一對多的同時點歌傳送，不需再遲遲等待排隊點歌，就能立即的點取自我想要的歌曲。

第四章 系統成果

第一節 系統流程

下圖是本研究開發此「APP 歌唱點播系統」的系統流程圖。進入本 APP 應用程式系統，主要以提供以下多重查詢功能，包涵：男女歌手查詢、性別筆畫查詢、歌曲語言種類查詢，以方便快速尋想要點播的歌曲。

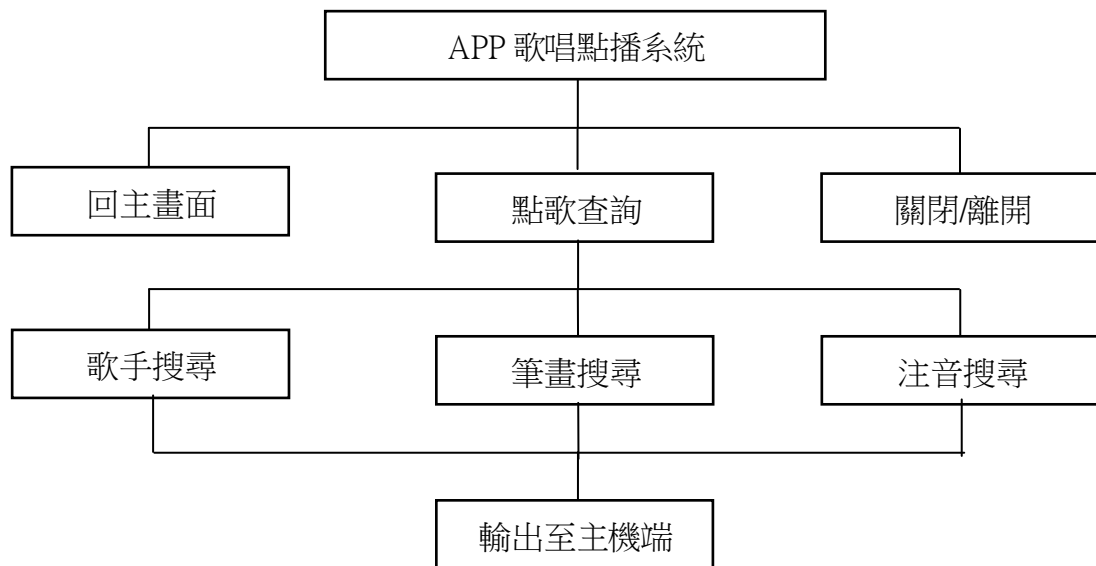


圖 10 系統流程圖

第二節 系統畫面

一、主機端執行畫面

下圖是：歌唱點播點歌機，主機端的起始畫面，共提供多項查詢功能服務。



圖 11 點歌機起始畫面

下圖是：選取歌曲種類搜尋，選擇台語之後所呈現的資料庫資料。



圖 12 點歌機使用點選搜尋台語類別

下圖是：選取歌曲種類搜尋後，將歌曲點播至清單的功能畫面。



圖 13 使用類別搜尋點播歌曲至清單

下圖是：操作使用下拉式選單，搜尋歌曲的搜尋方式提供包括：注音、筆畫歌手名稱、歌曲名稱。



圖 14 選擇點播的搜尋方式

下圖是：操作使用搜尋下拉式選單，選擇注音呈現注音小鍵盤。

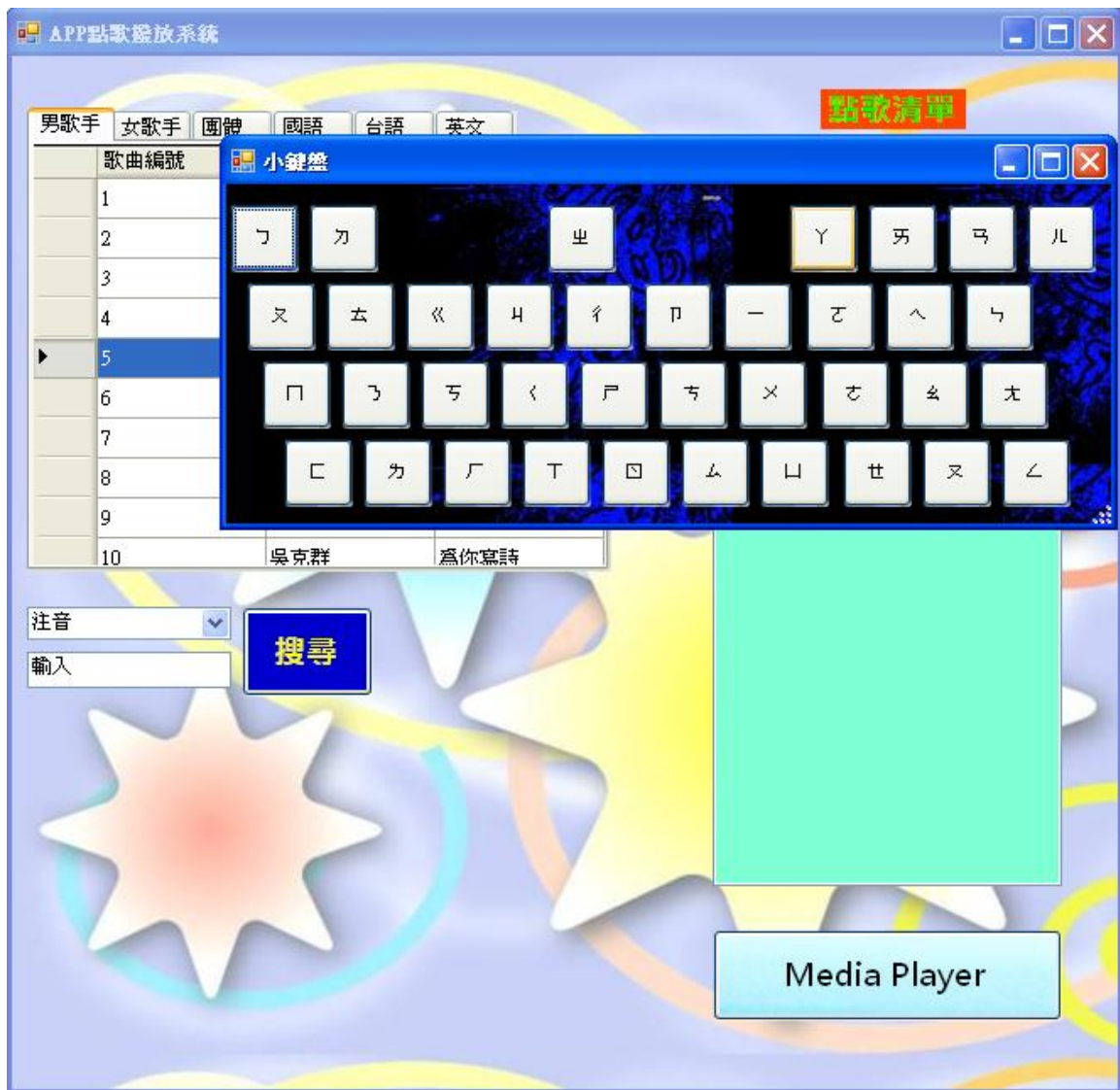


圖 15 點歌機使用點播搜尋的注音功能(一)

下圖是：搜尋選擇注音(ㄨ)之後的結果。



圖 16 點歌機使用點播搜尋的注音功能(二)

下圖是：搜尋選擇注音(ㄨ)之後的結果，點播至清單。



圖 17 使用注音搜尋點播至清單

下圖是：點播歌曲至清單，開始播放的 KTV 畫面。



圖 18 開始播放清單內之歌曲

二、手機端執行畫面

下圖是：進入程式的起始畫面，可以選擇直接關閉程式或者進入主頁



圖 19 APP 歌唱點播系統主畫面

下圖是：程式主畫面



下圖是：選擇並輸入搜尋條件之後，按下搜尋，就可以顯示搜尋結果，下圖搜尋比畫為 8 畫的資料。



圖 21 APP 歌手筆畫搜尋結果

下圖是：也可以直接在下面點擊按鈕搜尋，下圖點擊男歌手按鈕，就會顯示所有 男歌手資料。



圖 22 APP 男歌手搜尋結果

下圖是：搜尋筆畫 8 畫並且點擊男歌手，就會顯示所有歌手姓氏是 8 畫的男歌手。



圖 23 APP 筆畫搜尋男歌手結果

下圖是：點擊歌單內編號 5 歌曲，即可送出至主機端的點歌清單。



圖 24 APP 點選歌曲至點播清單

下圖是：若是什麼都沒輸入就點擊搜尋，就會顯示所有的資料，並且出現"請輸入查詢條件的"提示。



圖 25 APP 輸入正確查詢條件

第五章 結論與建議

第一節 研究結論

經由這次專題，研究開發「APP 歌唱點播系統」，主要以改善原本傳統「KTV 歌唱點播」的特性，達到同時多方的即時性點歌、與快速有效的時效性，及完成研究目的之標的。鑒於程式開發「APP 歌唱點播系統」，經由 Visual Studio 架設主機端及網路的程式基準，再由 Eclipse 編寫 APP Android 的手機應用程式，結合與智慧型手機的實務應用。

另外，評估智慧型手機替代遙控器點歌的使用效益上，在滿足此消費者之可行性結果，有相當顯著的需求。因此在未來希望會再增加更多方的點播功能，也讓此 APP 歌唱點播系統更多元化。

參考文獻

- 一、Android 程式設計實例入門，高橋麻奈 著/博碩文化 編譯
- 二、Google Android2.X 應用程式開發實戰，林城 著/碁峯資訊
- 三、Android 提高第九篇之 SQLite 分页表格
<http://blog.csdn.net/helloqv/article/details/6019301>
- 四、[Android]Sample Socket Server & Client
<http://blog.johnsonlu.org/androidsocket/>
- 五、網路端連線 c# udp client-server echo program
<http://stackoverflow.com/questions/15716566/c-sharp-udp-client-server-echo-program>

附錄一、主機端程式碼

FROM1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Net.Sockets;
using System.Net;
using System.Threading;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        DataSet ds = new DataSet();
        BindingSource bs = new BindingSource();
        Form2 f2 = new Form2();

        DataSet rcds = new DataSet();
        DataSet dss = new DataSet();
        int a;
        int b = 0;
        private static Socket mainSocket;
        int port;
        bool portready = false;
        const int MAX_CLIENTS = 20;
```

```

public class ConnectionInfo
{
    public Socket Socket;
    public byte[] Buffer;
}
private static List<ConnectionInfo> connections = new
List<ConnectionInfo>();

public Form1()
{
    InitializeComponent();

    /*
        string cstcall = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Database2.accdb";
        string sqlcall = "SELECT distinct 歌曲編號,歌手名稱,歌曲名
稱 FROM 歌單 where 歌曲編號="+ text;
        OleDbConnection connn = new OleDbConnection(cstcall);
        OleDbDataAdapter AdapterCallback = new
OleDbDataAdapter(sqlcall, connn);
        connn.Open();
        AdapterCallback.Fill(dss, "歌單");
        connn.Close();
        dataGridView8.DataSource = dss;
        dataGridView8.DataMember = "歌單";
    */
}
private void Form1_Load(object sender, EventArgs e)
{
    // System.Threading.Thread objWorkerThread = new
System.Threading.Thread(new ThreadStart(ReceiveCallback));
    // objWorkerThread.Start();

Form.CheckForIllegalCrossThreadCalls = false; //跨執行緒UI

//-----

```

```

// TODO: 這行程式碼會將資料載入 'databaseDataSet1.英語歌手' 資料表。
// 您可以視需要進行移動或移除。
//this.英語歌手TableAdapter.Fill(this.databaseDataSet1.英語歌手);
// TODO: 這行程式碼會將資料載入 'databaseDataSet1.台語女歌手' 資料
// 表。您可以視需要進行移動或移除。
//this.台語女歌手TableAdapter1.Fill(this.databaseDataSet1.台語女歌手);
// TODO: 這行程式碼會將資料載入 'databaseDataGriSet.台語女歌手' 資料
// 表。您可以視需要進行移動或移除。
//this.台語女歌手TableAdapter.Fill(this.databaseDataGriSet.台語女歌手);
    /* string sqlstr,connstr;
    //connstr = new OleDbConnection();
    connstr = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=|DataDirectory|\Database.accdb";
    //Conn.Open();
    OleDbConnection conn;
    OleDbDataAdapter da;
    conn = new OleDbConnection(connstr);*/

/*
    OleDbCommand command = new OleDbCommand();
    command.Connection = conn;
    command.CommandText = "Select * From 英語歌手";
    OleDbDataReader reader = command.ExecuteReader();
    while (reader.Read())
    {
        listBox1.Items.Add(reader[0].ToString());
    }
    */
//sqlstr = "select * from 英語歌手";
//da = new OleDbDataAdapter(sqlstr,conn);
// da.Fill(ds,"英語歌手");
/* sqlstr = "select * from 歌單";
    da=new OleDbDataAdapter(sqlstr,conn);
    if ("性別" = 1)
    {    textBox1.Text = "男";    }
    else
    {textBox1.Text="女";    }
    da.fill(ds, "歌單");*/
// sqlstr = "select * from 華語團體";

```

```

// da = new OleDbDataAdapter(sqlstr, conn);
// da.Fill(ds, "華語團體");
// sqlstr = "select * from 台語男歌手";
// da = new OleDbDataAdapter(sqlstr, conn);
// da.Fill(ds, "台語男歌手");
// sqlstr = "select * from 華語女歌手";
// da = new OleDbDataAdapter(sqlstr, conn);
// da.Fill(ds, "華語女歌手");
// sqlstr = "select * from 華語男歌手";
// da = new OleDbDataAdapter(sqlstr, conn);
// da.Fill(ds, "華語男歌手");
// sqlstr = "select * from 台語女歌手";
// da = new OleDbDataAdapter(sqlstr, conn);
// da.Fill(ds, "台語女歌手");
// for (int i = 0; i <= ds.Tables.Count - 1; i++)
//     comboBox1.Items.Add(ds.Tables[i].TableName);
//comboBox1.Text = ds.Tables[0].TableName;
// bs.DataSource=ds.Tables[0];
//dataGridView1.DataSource= bs.DataSource;
this.dataGridView2.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
//this.dataGridView2.MultiSelect = false;

this.dataGridView3.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
//this.dataGridView3.MultiSelect = false;

this.dataGridView4.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
//this.dataGridView4.MultiSelect = false;

this.dataGridView5.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
//this.dataGridView4.MultiSelect = false;

this.dataGridView6.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
//this.dataGridView4.MultiSelect = false;

```

```

        this.dataGridView7.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        //this.dataGridView4.MultiSelect = false;
        this.dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        //this.dataGridView4.MultiSelect = false;
        this.dataGridView8.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
        //this.dataGridView8.MultiSelect = false;

        this.dataGridView9.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;

        string connetionStringX =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sqlX = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單";
        OleDbConnection connback = new
OleDbConnection(connetionStringX);
        OleDbDataAdapter databack = new OleDbDataAdapter(sqlX,
connback);
        DataSet dsX = new DataSet();
        connback.Open();
        databack.Fill(dsX, "歌單");
        connback.Close();
        dataGridView8.DataSource = dsX;
        dataGridView8.DataMember = "歌單";

//=====
=====

        string connetionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sql = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單 where 性別 = '1' ";

        OleDbConnection connection = new
OleDbConnection(connetionString);
        OleDbDataAdapter dataadapter = new OleDbDataAdapter(sql,
connection);

```



```

        DataSet ds = new DataSet();
        connection.Open();
        dataadapter.Fill(ds, "歌單");
        connection.Close();
        dataGridView2.DataSource = ds;
        dataGridView2.DataMember = "歌單";

//-----

        string conectionString1 =
        "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sql1 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
        FROM 歌單 where 性別 = '2'";
        OleDbConnection connection1 = new
        OleDbConnection(conectionString1);
        OleDbDataAdapter dataadapter1 = new OleDbDataAdapter(sql1,
        connection1);
        DataSet ds1 = new DataSet();
        connection1.Open();
        dataadapter1.Fill(ds1, "歌單");
        connection1.Close();
        dataGridView3.DataSource = ds1;
        dataGridView3.DataMember = "歌單";

//-----

        string conectionString2 =
        "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sql2 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
        FROM 歌單 where 性別 = '3'";

        OleDbConnection connection2 = new
        OleDbConnection(conectionString2);
        OleDbDataAdapter dataadapter2 = new OleDbDataAdapter(sql2,
        connection2);
        DataSet ds2 = new DataSet();
        connection2.Open();

```

```

dataadapter2.Fill(ds2, "歌單");
connection2.Close();
dataGridView4.DataSource = ds2;
dataGridView4.DataMember = "歌單";

//-----

string conectionString3 =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
string sql3 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單 where 語言 = '國語';

OleDbConnection connection3 = new
OleDbConnection(conectionString3);
OleDbDataAdapter dataadapter3 = new OleDbDataAdapter(sql3,
connection3);
DataSet ds3 = new DataSet();
connection3.Open();
dataadapter3.Fill(ds3, "歌單");
connection3.Close();
dataGridView5.DataSource = ds3;
dataGridView5.DataMember = "歌單";

//-----

string conectionString4 =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
string sql4 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單 where 語言 = '台語';

OleDbConnection connection4 = new
OleDbConnection(conectionString4);
OleDbDataAdapter dataadapter4 = new OleDbDataAdapter(sql4,
connection4);
DataSet ds4 = new DataSet();
connection4.Open();
dataadapter4.Fill(ds4, "歌單");
connection4.Close();
dataGridView6.DataSource = ds4;
dataGridView6.DataMember = "歌單";

//-----

```

```

        string conectionString5 =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sql5 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單 where 語言 = '英語'";

        OleDbConnection connection5 = new
OleDbConnection(connetionString5);
        OleDbDataAdapter dataadapter5 = new OleDbDataAdapter(sql5,
connection5);
        DataSet ds5 = new DataSet();
        connection5.Open();
        dataadapter5.Fill(ds5, "歌單");
        connection5.Close();
        dataGridView7.DataSource = ds5;
        dataGridView7.DataMember = "歌單";
//-----
        string conectionString6 =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
        string sql6 = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱
FROM 歌單";
        OleDbConnection connection6 = new
OleDbConnection(connetionString6);
        OleDbDataAdapter dataadapter6 = new OleDbDataAdapter(sql6,
connection6);
        DataSet ds6 = new DataSet();
        connection.Open();
        dataadapter.Fill(ds6, "歌單");
        connection.Close();
        dataGridView9.DataSource = ds6;
        dataGridView9.DataMember = "歌單";
//-----
        comboBox1.Items.Add("注音");
        comboBox1.Items.Add("筆畫");
        comboBox1.Items.Add("歌手名稱");
        comboBox1.Items.Add("歌曲名稱");
//=====
=====

```

```

        port = 9999;
        mainSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        IPEndPoint ipLocal = new IPEndPoint(IPAddress.Any, port);
        mainSocket.Bind(ipLocal);
        mainSocket.Listen(10); //啟動監聽
        portready = true;
        mainSocket.BeginAccept(new AsyncCallback(AcceptCallback),
mainSocket); }

//=====
=====

private void button1_Click(object sender, EventArgs e) {
    if (tabControl1.SelectedIndex == 0)
        { // MessageBox.Show(dataGridView2[1,1].Value.ToString());
            Int32 selectedRowCount =
dataGridView9.Rows.GetRowCount(DataGridViewElementStates.Selected);
            if (selectedRowCount > 0)
                { System.Text.StringBuilder sb = new
System.Text.StringBuilder();
                    for (int i = 0; i < selectedRowCount; i++)
                        { for (int j = 0; j < 3; j++)
                            {

sb.Append(dataGridView9.CurrentRow.Cells[j].Value.ToString());
                                if (j < 2)
                                    { sb.Append(" "); }
                                    //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
                                    //sb.Append(Environment.NewLine);
                                } }
                                    //sb.Append("Total: " +
selectedRowCount.ToString());
                                    listBox1.Items.Add(sb.ToString());
                                    //MessageBox.Show(dataGridView2[a,
a].Value.ToString());
                                } }
        }
    }
}

```

```

        if (tabControl1.SelectedIndex == 1)
        {
            //
            MessageBox.Show(dataGridView2[1,1].Value.ToString());
            Int32 selectedRowCount =
dataGridView2.Rows.GetRowCount(DataGridViewElementStates.Selected);
            if (selectedRowCount > 0)
            {
                System.Text.StringBuilder sb = new
System.Text.StringBuilder();
                for (int i = 0; i < selectedRowCount; i++)
                {
                    for (int j = 0; j < 3; j++)
                    {
                        sb.Append(dataGridView2.CurrentRow.Cells[j].Value.ToString());
                            if (j < 2)
                            {
                                sb.Append(" ");
                            }
                            //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
                            //sb.Append(Environment.NewLine);
                        }
                    }
                    //sb.Append("Total: " +
selectedRowCount.ToString());
                    listBox1.Items.Add(sb.ToString());
                    //MessageBox.Show(dataGridView2[a,
a].Value.ToString());
                }
            }
        }
        if (tabControl1.SelectedIndex == 2)
        {
            // MessageBox.Show(dataGridView2[1,1].Value.ToString());
            Int32 selectedRowCount =
dataGridView3.Rows.GetRowCount(DataGridViewElementStates.Selected);
            if (selectedRowCount > 0)
            {
                System.Text.StringBuilder sb = new
System.Text.StringBuilder();
                for (int i = 0; i < selectedRowCount; i++)
                {
                    for (int j = 0; j < 3; j++)

```

```

{ sb.Append(dataGridView3.CurrentRow.Cells[j].Value.ToString());
    if (j < 2)
{ sb.Append(" "); }
    //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
    //sb.Append(Environment.NewLine);
} }
//sb.Append("Total: " +
selectedRowCount.ToString());
    listBox1.Items.Add(sb.ToString());
    //MessageBox.Show(dataGridView2[a,
a].Value.ToString());}
    if (tabControl1.SelectedIndex == 3)
    { // MessageBox.Show(dataGridView2[1,1].Value.ToString());
        Int32 selectedRowCount =
dataGridView4.Rows.GetRowCount(DataGridViewElementStates.Selected);
        if (selectedRowCount > 0)
        { System.Text.StringBuilder sb = new
System.Text.StringBuilder();
            for (int i = 0; i < selectedRowCount; i++)
            {for (int j = 0; j < 3; j++)

{ sb.Append(dataGridView4.CurrentRow.Cells[j].Value.ToString());
                if (j < 2)
                { sb.Append(" "); }
                //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
                //sb.Append(Environment.NewLine);
            } }
            //sb.Append("Total: " +
selectedRowCount.ToString());
            listBox1.Items.Add(sb.ToString());
            //MessageBox.Show(dataGridView2[a,
a].Value.ToString()); } }
            if (tabControl1.SelectedIndex == 4)
            {
                // MessageBox.Show(dataGridView2[1,1].Value.ToString());
                Int32 selectedRowCount =

```

```

dataGridView5.Rows.GetRowCount(DataGridViewElementStates.Selected);
    if (selectedRowCount > 0)
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
                for (int i = 0; i < selectedRowCount; i++)
                    {
                        for (int j = 0; j < 3; j++)
                            {
                                sb.Append(dataGridView5.CurrentRow.Cells[j].Value.ToString());
                                    if (j < 2)
                                        {
                                            sb.Append(" ");
                                        }
                                        //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
//sb.Append(Environment.NewLine);
                            }
                        }
                        //sb.Append("Total: " +
selectedRowCount.ToString());
                            listBox1.Items.Add(sb.ToString());
                                //MessageBox.Show(dataGridView2[a,
a].Value.ToString());
                                    }
                                        }
                                            if (tabControl1.SelectedIndex == 5)
                                                {
                                                    // MessageBox.Show(dataGridView2[1,1].Value.ToString());
//sb.Append("Total: " +
Int32 selectedRowCount =
dataGridView6.Rows.GetRowCount(DataGridViewElementStates.Selected);
    if (selectedRowCount > 0)
        {
            System.Text.StringBuilder sb = new System.Text.StringBuilder();
                for (int i = 0; i < selectedRowCount; i++)
                    {
                        for (int j = 0; j < 3; j++)
                            {
                                sb.Append(dataGridView6.CurrentRow.Cells[j].Value.ToString());
                                    if (j < 2)
                                        {
                                            sb.Append(" ");
                                        }
                                        //sb.Append("Row: ");
//sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
//sb.Append(Environment.NewLine);
                            }
                        }
                        //sb.Append("Total: " +
selectedRowCount.ToString());
                            listBox1.Items.Add(sb.ToString());
                                //MessageBox.Show(dataGridView2[a,
a].Value.ToString());
                                    }
                                        }

```

```

        if (tabControl1.SelectedIndex == 6)
            { // MessageBox.Show(dataGridView2[1,1].Value.ToString());
                Int32 selectedRowCount =
dataGridView7.Rows.GetRowCount(DataGridViewElementStates.Selected);
                if (selectedRowCount > 0)
                    { System.Text.StringBuilder sb = new System.Text.StringBuilder();
                        for (int i = 0; i < selectedRowCount; i++)
                            { for (int j = 0; j < 3; j++)
                                { sb.Append(dataGridView7.CurrentRow.Cells[j].Value.ToString());
                                    if (j < 2)
                                        { sb.Append(" "); }
                                    //sb.Append("Row: ");
                                //sb.Append(dataGridView2.SelectedRows[i].Index.ToString());
                                    //sb.Append(Environment.NewLine); } }
                                    //sb.Append("Total: " +
selectedRowCount.ToString());
                                listBox1.Items.Add(sb.ToString());
                                    //MessageBox.Show(dataGridView2[a,
a].Value.ToString()); } } }
private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
    { bs.DataSource = ds.Tables[comboBox1.Text];
      dataGridView1.DataSource = bs.DataSource;
      if (comboBox1.SelectedIndex == 0)
          {
              Form3 f3 = new Form3();
              f3.ShowDialog();
              textBox1.Text = f3.s; }
      if (comboBox1.SelectedIndex == 1)
          { textBox1.Text = "輸入筆畫";
              // MessageBox.Show("請輸入歌手筆畫"); }
      if (comboBox1.SelectedIndex == 2)
          { textBox1.Text = "輸入歌手名稱";
              //MessageBox.Show("請輸入國語、台語或英語"); }
      if (comboBox1.SelectedIndex == 3)
          { textBox1.Text = "輸入歌曲名稱";
              //MessageBox.Show("請輸入國語、台語或英語"); } }
private void button4_Click(object sender, EventArgs e)

```



```

        { // OleDbConnection cn = new OleDbConnection
          //("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Database2.accdb");
          // cn.Open();
          // OleDbCommand cmd = new OleDbCommand();
          //OleDbDataAdapter dataadapter = new OleDbDataAdapter(sql,
connection);
          //DataSet ds = new DataSet();
          // cmd.Connection = cn;
          //dataadapter.Fill(ds, "Authors_table");
          //connection.Close();
          //dataGridView1.DataSource = ds;
          //dataGridView1.DataMember = "Authors_table";
// cmd.CommandText = "select * from 歌單";
// cmd.ExecuteNonQuery();
//DataSet ds = new DataSet();
//dataGridView1.DataSource = a.DataSource;
//dataGridView1.DataSource = t;
//listBox1.Text = cmd;
//MessageBox.Show("Handsome");
//dataGridView1.DataSource = da.DataSource;
//for (int i = 0; i <= ds.Tables.Count - 1; i++)
//bs.DataSource=ds.Tables[0];
//dataGridView1.DataSource= bs.DataSource;
/*string a="性別";
sql = "select * from 歌單";
dataadapter = new OleDbDataAdapter(sql, connection);
if (b == 1)
{ textBox3.Text = "男"; }
else
{ textBox3.Text = "女"; }*/
//da.fill(ds, "歌單"); }
private void button2_Click(object sender, EventArgs e)
{if (textBox1.Text == null || textBox1.Text == "輸入" ||
comboBox1.Text == "選擇搜尋方式")
{ MessageBox.Show("請輸入搜尋條件", "條件錯誤");
textBox1.Text = "200";
comboBox1.Text = "歌手名稱"; }
}

```

```

string a = comboBox1.Text;
if (a == "筆畫")
{
    tabControl2.Visible = true;
    button20.Visible = true;
    string conectionString1 =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
    string sql1 = "SELECT distinct 歌曲編號,歌手名稱,歌曲
名稱 FROM 歌單 where 筆畫 = " + textBox1.Text + " ";
    //MessageBox.Show(textBox1.Text);
    OleDbConnection connection1 = new
OleDbConnection(conectionString1);
    OleDbDataAdapter dataadapter1 = new
OleDbDataAdapter(sql1, connection1);
    DataSet ds1 = new DataSet();
    connection1.Open();
    dataadapter1.Fill(ds1, "歌單");
    connection1.Close();
    dataGridView1.DataSource = ds1;
    dataGridView1.DataMember = "歌單";
}
else
{
    tabControl2.Visible = true;
    button20.Visible = true;
    //string a = comboBox1.Text;
    string conectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Database2.accdb";
    string sql = "SELECT distinct 歌曲編號,歌手名稱,歌曲名
稱 FROM 歌單 where " + a + " like '%" + textBox1.Text + "%'";
    OleDbConnection connection = new
OleDbConnection(conectionString);
    OleDbDataAdapter dataadapter = new
OleDbDataAdapter(sql, connection);
    DataSet ds = new DataSet();
    connection.Open();
    dataadapter.Fill(ds, "歌單");
    connection.Close();
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "歌單";
}
/* if (a=="語言" && (textBox1.Text != "國語" ||

```



```

else
{ //MessageBox.Show("歌曲播放完成");
  if (listBox1.Items.Count >= 1 )
  { listBox1.SetSelected(0, true);
    f2.axWindowsMediaPlayer1.URL = "D:\\GOOD\\" +
listBox1.Text.ToString() + ".MPG";
    //label1.Text=listBox1.Text.ToString();
    listBox1.Items.Remove(listBox1.SelectedItem); }
  //axWindowsMediaPlayer1.Text = "Windows Media Player is NOT
playing!"; } }
private void button9_Click(object sender, EventArgs e)
{ // Form2 f2 = new Form2();
  f2.Show(); }
private void timer2_Tick(object sender, EventArgs e)
{ if (listBox1.Items.Count >= 1)
{ timer1.Enabled = true;
  f2.axWindowsMediaPlayer1.URL = "D:\\GOOD\\" +
listBox1.Text.ToString() + ".MPG";
  timer2.Enabled = false; } }
private void textBox1_Click(object sender, EventArgs e)
{ textBox1.Text = ""; }

//=====
=====

private void AcceptCallback(IAsyncResult result)
{ ConnectionInfo connection = new ConnectionInfo();
  try
  { // Finish Accept
    Socket s = (Socket)result.AsyncState;
    connection.Socket = s.EndAccept(result);
    connection.Socket.Blocking = false;
    connection.Buffer = new byte[1024];
    lock (connections) connections.Add(connection);
    // Start Receive
    connection.Socket.BeginReceive(connection.Buffer, 0,
connection.Buffer.Length, SocketFlags.None, new
AsyncCallback(ReceiveCallback), connection);
    // Start new Accept
    mainSocket.BeginAccept(new AsyncCallback(AcceptCallback),

```

```

result.AsyncState); }
//=====
=====
private void ReceiveCallback(IAsyncResult result)
{   ConnectionInfo connection = (ConnectionInfo)result.AsyncState;
    // ThreadStart myRun = new ThreadStart(Sendthread);
    try
    {   int bytesRead = connection.Socket.EndReceive(result);
        string text = Encoding.UTF8.GetString(connection.Buffer, 0,
bytesRead);

        //string cst = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Database2.accdb";
        //string sql = "SELECT distinct 歌曲編號,歌手名稱,歌曲
名稱 FROM 歌單 where 歌曲編號=" + text;
string sqlcall = "SELECT distinct 歌曲編號,歌手名稱,歌曲名稱 FROM 歌
單 where 歌曲編號=" + text;
        //OleDbConnection connn = new OleDbConnection(cst);
        //OleDbDataAdapter dataadapter = new OleDbDataAdapter(sql, connn);
/*            connn.Open();
            dataadapter.Fill(ds, "歌單");
            connn.Close();
            dataGridView8.DataSource = ds;
            dataGridView8.DataMember = "歌單";    */
            a = Convert.ToInt32(text);
            a = a - 1;
            System.Text.StringBuilder sbb = new
System.Text.StringBuilder();
            try    {   for (int j = 0; j < 3; j++)
{   sbb.Append(dataGridView8.Rows[a].Cells[j].Value.ToString());
                if (j < 2)
                    { sbb.Append(" ");    }    }
                listBox1.Items.Add(sbb.ToString());    }
            catch (Exception) { }
            //    connection.Socket.BeginReceive(connection.Buffer, 0,
connection.Buffer.Length, SocketFlags.None, new
AsyncCallback(ReceiveCallback), connection);    }
//=====
=====

```

```

private void button3_Click_1(object sender, EventArgs e)
    {
        listBox1.SetSelected(0, true);
        f2.axWindowsMediaPlayer1.URL = "D:\\GOOD\\" +
listBox1.Text.ToString() + ".MPG";
        //label1.Text=listBox1.Text.ToString();
        listBox1.Items.Remove(listBox1.SelectedItem);
    }
private void button3_Click_2(object sender, EventArgs e)
{
    listBox1.Items.Remove(listBox1.SelectedItem);
}
private void button4_Click_1(object sender, EventArgs e)
{
    f2.axWindowsMediaPlayer1.Ctlcontrols.pause();
}

```

附錄二、手機端程式碼 **android textSQLite**

```

package com.testSQLite;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import android.R.string;
import android.app.Activity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;

```

```

import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
public class testSQLite extends Activity {
    GVTable table;
    Button
btnCreateDB,btnInsert,btnClose,btnboy,btngirl,btngro,btnlan1,btnlan2,btnlan3;
    ImageButton b1;
    SQLiteDatabase db;
    DatabaseHelper myDbHelper ;
    private static final int SOCKET_TIMEOUT = 1000;
    private static final int PORT=9999;
    private String ADDRESS="192.168.1.45";
    private Socket socket;
    BufferedReader buff;
    private EditText et1;
    //private Spinner spinner;
    private RadioGroup reg;
    private RadioButton n;
    private RadioButton mn;
    private RadioButton mnn;
    //private String[] list = {"歌手名稱","筆畫","歌曲名稱"};
    private ArrayAdapter<String> listAdapter;
    private Context mContext;
    String s,a;
    int d;
    int id;//添加記錄時的id 累加標記，必須全局
    private static final String TABLE_NAME = "music";
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        n = (RadioButton) findViewById(R.id.radio0);
        mn = (RadioButton) findViewById(R.id.radio1);

```

```

        nnn = (RadioButton) findViewById(R.id.radio2);
        reg = (RadioGroup) findViewById(R.id.radioGroup1);
        reg.setOnCheckedChangeListener(listener);
        et1 = (EditText) findViewById(R.id.editText1);
        s = et1.getText().toString();
        a = (String) n.getText();
        a = "number";
        mContext = this.getApplicationContext();
        //spinner = (Spinner) findViewById(R.id.spinner1);
        //listAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, list);
        //spinner.setAdapter(listAdapter);
        /* spinner.setOnItemSelectedListener(new OnItemSelectedListener(){
            @Override
            public void onItemSelected(AdapterView<?> arg0, View arg1, int
position, long arg3) {
                a = (String) spinner.getSelectedItem();
                if(a == "歌手名稱")
                { a = "name"; }
                if(a == "筆畫")
                { a = "number"; }
                if(a == "歌曲名稱")
                { a = "song"; }
                Toast.makeText(mContext, "你選的是"+a,
Toast.LENGTH_SHORT).show();}
            @Override
            public void onNothingSelected(AdapterView<?> arg0) {
                // TODO Auto-generated method stub
            }
        });*/
        et1.setOnTouchListener(new OnTouchListener(){ @Override
            public boolean onTouch(View v, MotionEvent event) {
                // TODO Auto-generated method stub
                et1.setText("");
                Toast.makeText(testSQLite.this, "請輸入查詢條件",
Toast.LENGTH_SHORT).show();
                return false; } });
        myDbHelper = new DatabaseHelper(this);
        try { myDbHelper.createDataBase(); }

```



```

        catch (IOException ioe) {
            throw new Error("Unable to create database");    }
        try { myDbHelper.openDataBase();
        }catch(SQLException sqle){throw sqle; }
        //Toast.makeText(this, "Success", Toast.LENGTH_SHORT).show();
        btnCreateDB = (Button) this.findViewById(R.id.btnCreateDB);
        btnCreateDB.setOnClickListener(new ClickEvent());
        btnboy = (Button) this.findViewById(R.id.button1);
        btnboy.setOnClickListener(new ClickEvent());
        btngirl = (Button) this.findViewById(R.id.button2);
        btngirl.setOnClickListener(new ClickEvent());
        btngro = (Button) this.findViewById(R.id.button3);
        btngro.setOnClickListener(new ClickEvent());
        btnlan1 = (Button) this.findViewById(R.id.button4);
        btnlan1.setOnClickListener(new ClickEvent());
        btnlan2 = (Button) this.findViewById(R.id.button5);
        btnlan2.setOnClickListener(new ClickEvent());
        btnlan3 = (Button) this.findViewById(R.id.button6);
        btnlan3.setOnClickListener(new ClickEvent());
        //btnClose = (Button) this.findViewById(R.id.btnClose);
        //btnClose.setOnClickListener(new ClickEvent());
        table=new GVTable(this);
        table.gvSetTableRowCount(8);//設置每個分頁的 ROW 總數
        LinearLayout ly = (LinearLayout)
findViewById(R.id.MainLinearLayout);
        btnboy.setOnTouchListener(new OnTouchListener(){
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                if(event.getAction() == MotionEvent.ACTION_DOWN){
                    v.setBackgroundResource(R.drawable.love);
                }else if(event.getAction() ==
MotionEvent.ACTION_UP){
                    v.setBackgroundResource(R.drawable.qwer); }
                return false; } });
        btngirl.setOnTouchListener(new OnTouchListener(){ @Override
            public boolean onTouch(View v, MotionEvent event) {
                if(event.getAction() ==
MotionEvent.ACTION_DOWN){

```

```

        v.setBackgroundResource(R.drawable.love);
        }else if(event.getAction() ==
MotionEvent.ACTION_UP){
        v.setBackgroundResource(R.drawable.qwer);
        } return false; } });
    btnGro.setOnTouchListener(new OnTouchListener(){
@Override
    public boolean onTouch(View v, MotionEvent event) {
        if(event.getAction() ==
MotionEvent.ACTION_DOWN){
v.setBackgroundResource(R.drawable.love);
        }else if(event.getAction() ==
MotionEvent.ACTION_UP){
        v.setBackgroundResource(R.drawable.qwer); }
        return false; } });
    btnlan1.setOnTouchListener(new OnTouchListener(){ @Override
    public boolean onTouch(View v, MotionEvent event) {
        if(event.getAction() ==
MotionEvent.ACTION_DOWN){
        v.setBackgroundResource(R.drawable.love);
        }else if(event.getAction() ==
MotionEvent.ACTION_UP){
v.setBackgroundResource(R.drawable.qwer); }
        return false; } });
    btnlan2.setOnTouchListener(new OnTouchListener(){ @Override
    public boolean onTouch(View v, MotionEvent event) {
        if(event.getAction() ==
MotionEvent.ACTION_DOWN){
        v.setBackgroundResource(R.drawable.love);
        }else if(event.getAction() ==
MotionEvent.ACTION_UP){
        v.setBackgroundResource(R.drawable.qwer); }
        return false; } });
    btnlan3.setOnTouchListener(new OnTouchListener(){ @Override
    public boolean onTouch(View v, MotionEvent event) {
        if(event.getAction() ==
MotionEvent.ACTION_DOWN){
        v.setBackgroundResource(R.drawable.love);

```

```

        }else if(event.getAction() ==
MotionEvent.ACTION_UP){
        v.setBackgroundResource(R.drawable.qwer); }
        return false;    }    });
table.setOnClickListener(new GVTable.OnTableClickListener() {
    @Override
        public void onTableClickListener(int x,int y,Cursor c) {
            c.moveToPosition(y);
            if(x==0) {
                socket=null;
                DataOutputStream os = null;
                String str="第"+c.getString(0)+"首，演唱
者："+c.getString(x+1)+"，歌曲名稱："+c.getString(x+2);
                Toast.makeText(testSQLite.this, str, 1000).show();
                String str2 = c.getString(0);
                try { InetAddress serverAddr =
InetAddress.getByName(ADDRESS);
                    socket = new Socket(serverAddr,PORT);
                    String st = c.getString(0);
                    socket.setSoTimeout(SOCKET_TIMEOUT);
                        os = new
DataOutputStream(socket.getOutputStream());
                            os.write(st.getBytes());
                                } catch (IOException e) {
                                    e.printStackTrace(); } }
                if(x==1) {
                    socket=null;
                        DataOutputStream os = null;
String str="第"+c.getString(0)+"首，演唱者："+c.getString(x)+"，歌曲名
稱："+c.getString(x+1);
                Toast.makeText(testSQLite.this, str, 1000).show();
                try{ InetAddress serverAddr = InetAddress.getByName(ADDRESS);
                    socket = new Socket(serverAddr,PORT);
                    String st = c.getString(0);
                    socket.setSoTimeout(SOCKET_TIMEOUT);
os = new DataOutputStream(socket.getOutputStream());
                        os.write(st.getBytes());
                            } catch (IOException e) {

```

```

        e.printStackTrace();}    }
    if(x==2)
    { socket=null;
      DataOutputStream os = null;
String str="第"+c.getString(0)+"首，演唱者:"+c.getString(x-1)+"，歌曲名
稱:"+c.getString(x);
      //String str=c.getString(x)+" 位
置:"+String.valueOf(x)+","+String.valueOf(y)+"";
      Toast.makeText(testSQLite.this, str, 1000).show();
      try {
          InetAddress serverAddr =
InetAddress.getByName(ADDRESS);
          socket = new Socket(serverAddr,PORT);
          String st = c.getString(0);
          socket.setSoTimeout(SOCKET_TIMEOUT);
          os = new DataOutputStream(socket.getOutputStream());
          os.write(st.getBytes());    }
catch (IOException e) { e.printStackTrace(); } }));
    table.setOnPageChangeListener(new GVTable.OnPageChangeListener()
{ @Override
    public void onPageChangeListener(int pageID,int pageCount) {
        String str="共有"+String.valueOf(pageCount)+
        " 當前第"+String.valueOf(pageID+1)+"頁";
        Toast.makeText(testSQLite.this, str, 1000).show() });
    ly.addView(table); }
    private RadioGroup.OnCheckedChangeListener listener = new
RadioGroup.OnCheckedChangeListener()
    { @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            // TODO Auto-generated method stub
            //int p = group.indexOfChild((RadioButton)
findViewById(checkedId));
            //int count = group.getChildCount();
            switch (checkedId)
            {
                case R.id.radio0:
                    a=(String) n.getText();
                    a="number";
                    //Toast.makeText(mContext, "你選的是"+n.getText(),

```

```

Toast.LENGTH_SHORT).show();
        break;
    case R.id.radio1:
        a=(String) m.getText();
        a="name";
        // Toast.makeText(mContext, "你選的是"+m.getText(),
Toast.LENGTH_SHORT).show();
        break;
    case R.id.radio2:
        a=(String) mn.getText();
        a="song";
//Toast.makeText(mContext, "你選的是"+mn.getText(),
Toast.LENGTH_SHORT).show();
        break;   } } };

class ClickEvent implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        if (v == btnCreateDB) {
            CreateDB();
            if(a=="number" ) {
                if(s.matches("")) {
//秀出所有資料(什麼都沒輸入的 NUMBER)
                    CreateDB();
                    table.gvRemoveAll();
                    InsertRecord(16);
                    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME,db);
                    table.gvReadyTable("select musicnumber,name,song from " +
TABLE_NAME,db);
                    table.LoadTable(0);  }
                else
            { CreateDB();
//InsertRecord(16)//插入 16 條記錄
                    table.gvRemoveAll();
                    InsertRecord(16);
                    //db.close();
                    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME+" where "+a+" = "+s+"",db);

```

```

        table.gvReadyTable("select musicnumber,name,song from " +
TABLE_NAME+" where "+a+" = "+"'+s+'",db);
        table.LoadTable(0);    }    }
        else
        { CreateDB();
          //InsertRecord(16);//插入 16 條記錄
          table.gvRemoveAll();
          InsertRecord(16);
          //db.close();
          table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME+" where "+a+" like "+"%'"+s+"%' ",db);
          table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME+" where "+a+" like "+"%'"+s+"%' ",db);
          table.LoadTable(0); } }
        if (v == btnboy)
        {   CreateDB();
          if(s.matches(""))
          {   table.gvRemoveAll();
            InsertRecord(16);
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+1,db);
            table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where sex="+1,db);
            table.LoadTable(0);    }
          else
          { if(a=="number")    {
            //InsertRecord(16);//插入 16 條記錄
            table.gvRemoveAll();
            InsertRecord(16);
            //db.close();
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+1+" and "+a+" = "+"'+s+'",db);
            table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+1+" and
"+a+" = "+"'+s+'",db);
            table.LoadTable(0); }
          else {   table.gvRemoveAll();
            InsertRecord(16);

```

```

        table.gvUpdatePageBar("select count(*) from " + TABLE_NAME + "
where sex="+1+" and "+" "+a+" like "+ "% "+s+"%",db);
            table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+1+" and
"+" "+a+" like "+ "% "+s+"%",db); table.LoadTable(0);    }}
        if (v == btngirl)
            { CreateDB();
                if(s.matches("")){
                    table.gvRemoveAll();
                    InsertRecord(16);
                    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+2,db);
                    table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where sex="+2+" and "+" language = '國語",db);
                    table.LoadTable(0);}
                else
                    {if(a=="number")  {
                        //InsertRecord(16);//插入 16 條記錄
                        table.gvRemoveAll();
                        InsertRecord(16);
                        //db.close();
                        table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+2+" and "+a+" = "+" "+s+"%",db);
                        table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+2+" and
"+a+" = "+" "+s+"%",db);
                            table.LoadTable(0);    }
                        else
                            {
                                table.gvRemoveAll();
                                InsertRecord(16);
                                table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+2+" and "+" "+a+" like "+ "% "+s+"%",db);
                                table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+2+" and
"+" "+a+" like "+ "% "+s+"%",db);
                                    table.LoadTable(0);    }}
                    }if (v == btngro)
                        {
                            CreateDB();

```

```

        if(s.matches(""))
        {
            table.gvRemoveAll();
            InsertRecord(16);
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+3,db);
            table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where sex="+3,db);
            table.LoadTable(0);}
        else
        {if(a=="number")
        {//InsertRecord(16);//插入 16 條記錄
            table.gvRemoveAll();
            InsertRecord(16);
            //db.close();
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+3+" and "+a+" = "+""+s+"",db);
            table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+3+" and
"+a+" = "+""+s+"",db);
            table.LoadTable(0);    }
        else
        {table.gvRemoveAll();
            InsertRecord(16);
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where sex="+3+" and "+a+" like "+"%"+s+"%",db);
            table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where sex="+3+" and
"+a+" like "+"%"+s+"%",db);
            table.LoadTable(0);}}
    } if (v == btnlan1)
    {CreateDB();
        if(s.matches("")){
            table.gvRemoveAll();
            InsertRecord(16);
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language = '國語' ,db);
            table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where language = '國語' ,db);

```



```

        table.LoadTable(0);}
else
{if(a=="number")
    { //InsertRecord(16);//插入 16 條記錄
      table.gvRemoveAll();
      InsertRecord(16);
      //db.close();
      table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '國語'" + " and "+a+" = "+"'+s+'",db);
      table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '國語'"
+" and "+a+" = "+"'+s+'",db);
      table.LoadTable(0);    }
    else
    { table.gvRemoveAll();
      InsertRecord(16);
      table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '國語' and "+" "+a+" like
"+"%'"+s+'%'",db);
      table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '國語'
and "+" "+a+" like "+"%'"+s+'%'",db);
      table.LoadTable(0);}    }
} if (v== btnlan2)
{ CreateDB();
  if(s.matches(""))
  { table.gvRemoveAll();
    InsertRecord(16);
    table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language = '台語'",db);
    table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where language = '台語'",db);
    table.LoadTable(0);}
  else
  { if(a=="number")
    {//InsertRecord(16);//插入 16 條記錄
      table.gvRemoveAll();
      InsertRecord(16);

```

```

        //db.close();
        table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '台語'" + " and "+a+" = "+"+s+",db);
        table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '台語'"
+" and "+a+" = "+"+s+",db);
        table.LoadTable(0);          }
    else
    {
        table.gvRemoveAll();
        InsertRecord(16);
        table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '台語' and "+a+" like
"+"%"+s+"%",db);
        table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '台語'"
and "+a+" like "+"%"+s+"%",db);
        table.LoadTable(0);}}
    } if (v == btnlan3)
    {
        CreateDB();
        if(s.matches(""))
        {
            table.gvRemoveAll();
            InsertRecord(16);
            table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language = '英語'",db);
            table.gvReadyTable("select musicnumber,name,song
from " + TABLE_NAME + " where language = '英語'",db);
            table.LoadTable(0);      }
        else
        {if(a=="number")
            {
                //InsertRecord(16);//插入 16 條記錄
                table.gvRemoveAll();
                InsertRecord(16);
                //db.close();
                table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '英語'" + " and "+a+" = "+"+s+",db);
                table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '英語'"
+" and "+a+" = "+"+s+",db);

```

```

        table.LoadTable(0);    }
    else
    {
        table.gvRemoveAll();
        InsertRecord(16);
        table.gvUpdatePageBar("select count(*) from " +
TABLE_NAME + " where language= '英語' and "+" "+a+" like
"+"%" +s+"%"',db);

        table.gvReadyTable("select
musicnumber,name,song from " + TABLE_NAME + " where language= '英語'
and "+" "+a+" like "+"%" +s+"%"',db);
        table.LoadTable(0);    }}}}
/**
 * 在內存創建數據庫和數據表*/
void CreateDB() {
    // 在內存創建數據庫
/*    db = SQLiteDatabase.create(null);
    Log.e("DB Path", db.getPath());
    String amount = String.valueOf(databaseList().length);
    Log.e("DB amount", amount);
    // 創建數據表
    String sql = "CREATE TABLE " + TABLE_NAME + "(" +
        ID + " text not null, " + NAME + " text not null," +
        ADDRESS + " text not null, " + PHONE + " text not
null," +
        AGE + " text not null "+");";
    try {db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        db.execSQL(sql);
    } catch (SQLException e) {}
*/
    s = et1.getText().toString(); }
/**
 * 插入 N 條數據
 */
void InsertRecord(int n) {
    db = myDbHelper.getReadableDatabase() ;
    /*    int total = id + n;
    for (; id < total; id++) {
        String sql = "insert into " + TABLE_NAME + "(" +

```

```

        ID + ", " + NAME + ", " + ADDRESS + ", " + PHONE + ", "+AGE
        + ") values('" + String.valueOf(id) + "',
'man','address','123456789','18');";
        try {
            db.execSQL(sql);
        } catch (SQLException e) {}
    }*/}}

```

GVTABLE

```

package com.testSQLite;
import java.sql.SQLXML;
import java.util.ArrayList;
import java.util.HashMap;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.GridView;
import android.widget.LinearLayout;
import android.widget.SimpleAdapter;
import android.widget.AdapterView.OnItemClickListener;
public class GVTable extends LinearLayout {
    protected GridView gvTable,gvPage;
    protected SimpleAdapter saPageID,saTable;// 適配器
    protected ArrayList<HashMap<String, String>> srcPageID,srcTable;// 數據源
    protected int TableRowCount=10;//分頁時，每頁的 Row 總數
    protected int TableColCount=0;//每頁 col 的數量
    protected SQLiteDatabase db;
    protected String rawSQL="";
    protected Cursor curTable;//分頁時使用的 Cursor
    protected OnTableClickListener clickListener;//整個分頁控件被點擊時的
    回調函數
    protected OnPageSwitchListener switchListener;//分頁切換時的回調函數
    public GVTable(Context context) {

```

```

super(context);
this.setOrientation(VERTICAL);//垂直
//-----
gvTable=new GridView(context);
addView(gvTable, new
LinearLayout.LayoutParams(LayoutParams.FILL_PARENT,
        LayoutParams.WRAP_CONTENT));//寬長式樣
//gvTable.setBackground(Red);
srcTable = new ArrayList<HashMap<String, String>>();
saTable = new SimpleAdapter(context,
        srcTable,// 數據來源
        R.layout.items,//XML 實現
new String[] { "ItemText" },// 動態數組與 ImageItem 對應的子項
        new int[] { R.id.ItemText });
// 添加並且顯示
gvTable.setAdapter(saTable);
gvTable.setOnItemClickListener(new OnItemClickListener(){
    @Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
        long arg3) {
        int y=arg2/curTable.getColumnCount()-1;//標題欄的不算
        int x=arg2 % curTable.getColumnCount();
        if (clickListener != null//分頁數據被點擊
            && y!=-1) {//點中的不是標題欄時
clickListener.onTableClickListener(x,y,curTable);}}  });
//-----
gvPage=new GridView(context);
gvPage.setColumnWidth(60);//設置每個分頁按鈕的寬度

gvPage.setNumColumns(GridView.AUTO_FIT);//分頁按鈕數量自動設置
addView(gvPage, new
LinearLayout.LayoutParams(LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT));//寬長式樣
srcPageID = new ArrayList<HashMap<String, String>>();
saPageID = new SimpleAdapter(context,
        srcPageID,// 數據來源
        R.layout.items,//XML 實現
new String[] { "ItemText" },// 動態數組與 ImageItem 對應的子項

```

```

        new int[] { R.id.ItemText });
// 添加並且顯示
gvPage.setAdapter(saPageID);
// 添加消息處理
gvPage.setOnItemClickListener(new OnItemClickListener(){
    @Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
        long arg3) {
        LoadTable(arg2);//根據所選分頁讀取對應的數據
        if(switchListener!=null){//分頁切換時
switchListener.onPageSwitchListener(arg2,srcPageID.size());}} }); }
/**
 * 清除所有數據 */
public void gvRemoveAll()
{   if(this.curTable!=null)
        curTable.close();
        srcTable.clear();
        saTable.notifyDataSetChanged();
        srcPageID.clear();
        saPageID.notifyDataSetChanged();    }
/**
 * 讀取指定 ID 的分頁數據,返回當前頁的總數據
 * SQL:Select * From TABLE_NAME Limit 9 Offset 10;
 * 表示從 TABLE_NAME 表獲取數據，跳過 10 行，取 9 行
 * @param pageID 指定的分頁 ID    */
protected void LoadTable(int pageID)
{   if(curTable!=null)//釋放上次的數據
        curTable.close();
        String sql= rawSQL+" Limit "+String.valueOf(TableRowCount)+ "
Offset "+String.valueOf(pageID*TableRowCount);
        curTable = db.rawQuery(sql, null);
        gvTable.setNumColumns(curTable.getColumnCount());//表現為表格
的關鍵點！
        TableColCount=curTable.getColumnCount();
        srcTable.clear();
        // 取得字段名稱
        int colCount = curTable.getColumnCount();
        for (int i = 0; i < colCount; i++) {

```

```

        HashMap<String, String> map = new HashMap<String,
String>());
        map.put("ItemText", curTable.getColumn(i));
        srcTable.add(map);
    } // 列舉出所有數據
    int recCount=curTable.getCount();
    for (int i = 0; i < recCount; i++) { //定位到一條數據
        curTable.moveToPosition(i);
        for(int ii=0;ii<colCount;ii++) //定位到一條數據中的每個字段
        {
            HashMap<String, String> map = new HashMap<String,
String>());
            map.put("ItemText", curTable.getString(ii));
            srcTable.add(map); }}
        saTable.notifyDataSetChanged(); }
/**
 * 設置表格的最多顯示的行數
 * @param row 表格的行數
 */
public void gvSetTableRowCount(int row)
{
    TableRowCount=row; }
/**
 * 取得表格的最大行數
 * @return 行數 */
public int gvGetTableRowCount()
{
    return TableRowCount; }
/**
 * 取得當前分頁的 Cursor
 * @return 當前分頁的 Cursor */
public Cursor gvGetCurrentTable()
{
    return curTable; }
/**
 * 準備分頁顯示數據
 * @param rawSQL sql 語句
 * @param db 數據庫 */
public void gvReadyTable(String rawSQL,SQLiteDatabase db)
{
    this.rawSQL=rawSQL;

```

```

        this.db=db;    }
/**
 * 刷新分頁欄，更新按鈕數量
 * @param sql SQL 語句
 * @param db 數據庫
 */
public void gvUpdatePageBar(String sql,SQLiteDatabase db)
{
    Cursor rec = db.rawQuery(sql, null);
    rec.moveToLast();
    long recSize=rec.getLong(0);//取得總數
    rec.close();
    int pageNum=(int)(recSize/TableRowCount) + 1;//取得分頁數
    srcPageID.clear();
    for (int i = 0; i < pageNum; i++) {
        HashMap<String, String> map = new HashMap<String, String>();
        map.put("ItemText",    String.valueOf(i+1)+"頁");// 添加圖像
        資源的 ID
        srcPageID.add(map);        }
        saPageID.notifyDataSetChanged();    }
//-----
/**
 * 表格被點擊時的回調函數
 */
public void setTableOnClickListener(OnClickListener click) {
    this.clickListener = click;    }

public interface OnClickListener {
    public void onClickListener(int x,int y,Cursor c); }
//-----
/**
 * 分頁欄被點擊時的回調函數
 */
public void setOnPageSwitchListener(OnPageSwitchListener pageSwitch)
{this.switchListener = pageSwitch;    }
public interface OnPageSwitchListener {
    public void onPageSwitchListener(int pageID,int pageCount);    }}

```