

# 逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

手 機 簡 訊 鈴 聲 圖 片 服 務 系 統

學

生：楊博偉（四乙）

曾琬瑁（四乙）

張峯銘（四乙）

巫承蒲（四乙）

指 導 教 授：黃秋煌 教授

中 華 民 國 九 十 三 年 十 二 月

資逢  
訊甲  
工甲  
程甲  
學大  
專學  
題學  
報學  
告學

圖手  
片機  
服簡  
務訊  
系鈴  
統聲

楊曾  
博琬  
偉瑁  
· ·  
張巫  
峯承  
銘蒲



# 目 錄

目錄	..
圖表目錄	..
摘要	..
第一章 導 論 .....	1
1.1 研究動機和目的 .....	1
1.2 開發工具與環境簡介 .....	1
1.2.1 .NET平臺介紹 .....	1
1.2.2 開發工具 .....	4
1.3 工作規劃 .....	4
第二章 簡訊服務的概念與應用 .....	5
2.1 簡訊服務的原理與架構 .....	5
2.2 簡訊服務的應用 .....	8
2.2.1 PDU數據格式分析 .....	8
2.2.2 PDU訊息編碼 .....	12
2.3 鈴聲與待機圖片的編碼 .....	12
2.4 簡訊發送元件.....	16
2.4.1計算機與手機模組的通信 .....	16
2.4.2程式實作 .....	19
第三章 系統實作.....	26
3.1 系統架構.....	26
3.2 資料庫設計 .....	27
3.2.1 SQL簡介.....	27
3.2.2 資料庫正規化.....	27
3.2.3 系統資料庫設計.....	29
3.3 網站系統架構分析.....	33
3.3.1 會員管理註冊系統.....	36
3.3.2 文字簡訊服務系統.....	40
3.3.3 鈴聲服務系統 .....	44
3.3.4 圖片服務系統.....	52
3.3.5 計費系統 .....	53
3.4 實作成果 .....	55
3.4.1 文字簡訊服務系統部分 .....	55
3.4.2 鈴聲服務系統部分 .....	55
3.4.3 圖片服務系統 .....	60

第四章 系統評估與討論.....	61
4.1 系統評估 .....	61
4.2 未來展望 .....	61
4.3 心得討論 .....	62
參考資料 .....	64



## 圖表目錄

(圖1-1)	伺服器平台	1
(圖1-2)	VB.NET的虛擬機器	2
(圖1-3)	.NET Framework類別庫	3
(圖2-1)	GSM網路原理	5
(圖2-2)	簡訊傳送架構	7
(圖2-3)	PDU發送數據格式分析	11
(圖2-4)	PDU接收數據格式分析	12
(圖2-5)	Norking數據格式分析	14
(圖2-6)	待機圖片數據格式分析	15
(圖2-7)	簡訊發送之硬體架構圖	16
(圖2-8)	SMS相關的AT指令集	17
(圖2-9)	選擇串列埠	17
(圖2-10)	設定串列埠參數	18
(圖2-11)	測試指令	19
(圖2-12)	簡訊發送程式分析	19
(圖3-1)	簡訊服務網路系統架構	26
(圖3-2)	3層式主從架構圖	26
(圖3-3)	正規化步驟圖	28
(圖3-4)	資料庫關聯圖	30
(圖3-5)	簡訊資料表	32
(圖3-6)	會員資料表	32
(圖3-7)	系統階層圖	33
(圖 3-8)	網站系統流程圖	35
(圖 3-9)	會員管理註冊系統功能	36
(圖 3-10)	會員管理註冊系統流程圖-註冊會員	37
(圖 3-11)	會員管理註冊系統流程圖-會員登入	38
(圖 3-12)	會員管理註冊系統流程圖-會員修改	39
(圖 3-13)	會員管理註冊系統流程圖-查詢發送記錄	40
(圖 3-14)	簡訊系統圖	41
(圖 3-15)	簡訊發送流程圖	42
(圖 3-16)	罐頭簡訊服務系統流程圖	43
(圖 3-17)	鈴聲系統圖	44
(圖 3-18)	鈴聲發聲流程圖	45
(圖 3-19)	鈴聲下載流程圖	51
(圖 3-20)	圖片服務系統流程圖	52
(圖 3-21)	計費系統流程圖	53

(圖 3-22)	系統主畫面	55
(圖 3-23)	文字簡訊編輯與發送畫面	55
(圖 3-24)	罐頭簡訊分類選取畫面	56
(圖 3-25)	罐頭簡訊發送畫面	57
(圖 3-26)	鈴聲編碼畫面圖	57
(圖 3-27)	鈴聲編碼傳送畫面圖	58
(圖 3-28)	自編鈴聲發送網頁畫面	58
(圖 3-29)	鈴聲下載分類選取畫面	59
(圖 3-30)	鈴聲發送畫面	59
(圖 3-31)	圖片下載分類選取畫面	60
(圖 3-32)	圖片發送畫面	60



## 摘要

近年來電腦與通訊科技日益精進，相關軟硬體蓬勃發展，手機已不再是通訊工具，而是生活的隨身必備品。在人手一機的情況下，通過GSM手機的簡訊服務(Short Message Service, SMS)也逐漸普及。SMS是一種使得無線設備可以發送和接收本文訊息的技術。如同電子郵件一樣，SMS訊息可以長時間被保存在手機終端或簡訊服務中心(SMS Center)。而且具有不同的訊息格式，包括文字、圖形、聲音、e-Mail和傳真等。

SMS因其經濟、便於群體通訊、可儲存等特點受到廣泛歡迎，用戶數量巨大，關於SMS的應用相繼而生；一開始有文字簡訊、鈴聲訊息，到後來利用SMS遙控家電與作為防盜工具。然而在SMS的便利下，手機的介面卻沒有相對改進，因此開始有利用PC端來發送接收簡訊的軟體出現。隨著網路人口逐漸擴增，利用網路來聯絡似乎已經成為必要的需求，因此將SMS移植為Web-based的方式已成為一種趨勢。有鑒於此，本專題嘗試實作手機簡訊網路服務系統，透過無線設備與SMSC連結傳輸訊息，利用瀏覽器介面的客戶端程式，使得SMS的服務變得更為可親。

本專題實驗主要為分兩部分：

1. 簡訊伺服器端與手機無線模組之資料傳輸。
2. 瀏覽器介面客戶端與簡訊伺服器端之資料傳輸。





# 第一章 導 論

## 1.1 研究動機和目的

近年來3G帶動手機多媒體風潮，促成手機使用者愈來愈普及的情況下，GSM所提供的簡訊服務相關應用也隨之應運而生，功能也越來越複雜與多元化，相對地必須提供一個簡便容易的介面來服務使用者，網路瀏覽器介面或許是一個適宜的解決方案。

由於簡訊服務在你我生活中的應用俯拾即是，消費者對於功能與操作介面的要求與日俱增，到目前為止，國內外已有許多電信業者頻頻發表簡訊網路服務的API模組，讓我們感受到簡訊服務未來的榮景。面對此具有發展潛力的領域，興起了對簡訊服務的研究興趣。

本專題中，企圖建構一個Web-Based的簡訊服務系統，進一步地探討簡訊服務在GSM網路中的協定運作方式，同時在系統架構上希望可以達到良好的擴充性。

## 1.2 開發工具與環境簡介

簡訊服務系統是架構在Windows Server系統下的IIS與SQL Server環境中，使用Visual Basic.Net語言完成的。系統主要可分為前端瀏覽器介面，與後端的Web伺服器。

### 1.2.1 .NET平臺介紹

VB.NET	C#	VB.NET
C++	ASP.NET	
Windows Form AP	Web Form AP	
ADO.NET+CLR+Class Library+XML+Web Service...		
.Net Framework		
Windows 2K Server		

Code Behind可用 VB.NET 或

圖 1-1 伺服器平臺

.Net架構相當適合開發新一代的Web-Based應用系統。在此架構下可以充分控制所用到的記憶體及系統資源，提供相當大的效能提升。

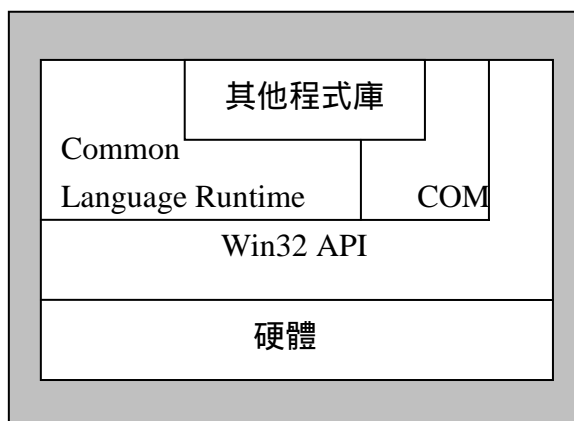


圖 1-2 VB.NET 的虛擬機器

VB.NET 撰寫程式所用的虛擬機器與過去 VB 程式完全不同，因為新一代的 VB.NET 使用的是 .NET 的 Common Language Runtime (CLR)。在此架構下，所有的 .NET 程式與元件均建立在以一個或多個 DLL 及 EXE 為主的組合檔 (Assembly) 之下，當將 .NET 組合檔編譯成為 DLL 或 EXE 檔時，編譯的程式其實尚未完成，因為絕大多數編譯器所需的資料與元件都儲存在組合檔中的清單 (Manifest)，這時所產生的程式碼並非原生碼 (Native Code)，而是仲介語言 (Intermediate Language, IL)，當系統第一次載入組合檔或在系統中安裝組合檔時，這種程式碼會透過 JIT (Just-in-Time) 編譯器轉成原生碼，此外 JIT 編譯器也會進行其他步驟：

1. JIT 編譯器會檢查程式碼以確定沒有記憶體存取的問題，如此程式才不會存取到物件或資料結構定義以外的記憶體。
2. JIT 編譯器會產生必要的程式碼及表格，以便讓 CLR 根據此一表格找到程式裡面的根階層變數 (Root Level Variables)。根階層變數即全域變數及儲存在堆疊或暫存在 CPU 暫存器之中的區域變數。

因此理論上，透過 CLR 所撰寫出來的程式可以不需重新編譯就可以在另一套有安裝 CLR 的作業系統上執行。

此外，.NET Framework 還具備了幾項特性：

1. 提供完整的類別程式庫(Class Library)：讓程式有更便利的物件可使用，而且物件導向的概念完整落實在VB.NET程式設計中。

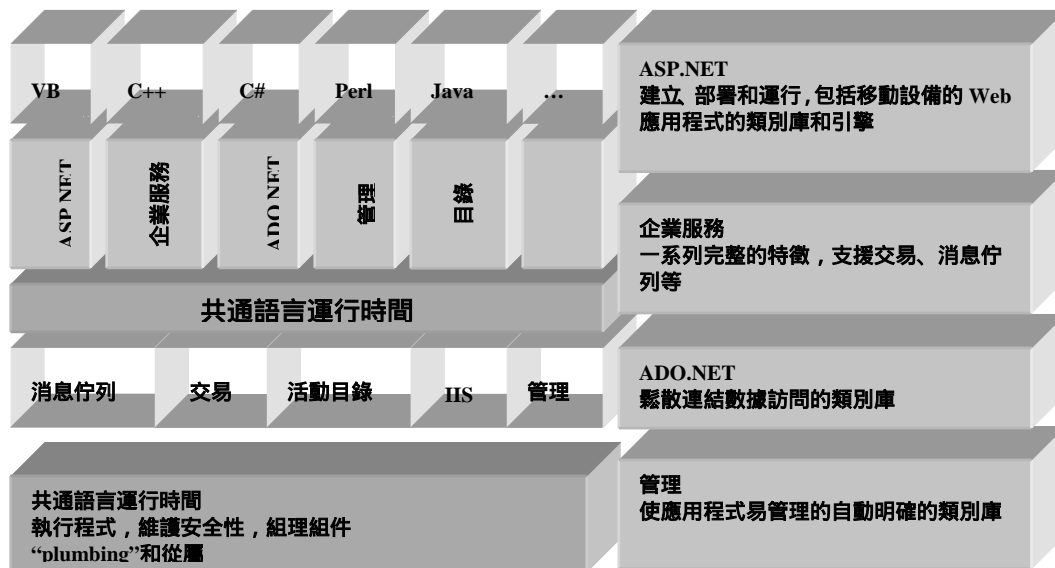


圖 1-3 .NET Framework 類別庫

2. 改良的ADO.NET、ASP與VB和JAVA(C#)，與WEB整合：.NET Framework徹底改善了WEB應用程式開發的流程，在架構上，傾向於撰寫一套程式，但是透過「網頁介面」和「網際網路環境」來執行。ASP.NET的核心是Web Form，本質上是一個類別物件，因此具有物件該有的特質，包括事件(Event)、方法(Method)、屬性(Property)，並將程式碼與網頁部份分離，實現程式碼隱藏(Code Behind)。

## 1.2.2 開發工具

整個專題中各個系統架構所需的各項軟硬體明細如下：

前端瀏覽器介面：

Microsoft Internet Explorer 6.0以上的版本  
.Net Framework

後端開發環境：

瀏覽器與Web伺服器端

Windows XP

Web Server ( IIS 5.1)

Microsoft SQL Server 2000

Visual Studio.Net

Web伺服器端與手機模組

Nokia 8210

Win32 API

## 1.3 工作規劃

工作項目\負責人	巫承蒲	張峯銘	曾琬瑀	楊博偉
系統分析	V	V	V	V
資料蒐集	V	V	V	V
系統流程圖	V	V	V	V
資料庫規劃	V	V		
會員管理系統		V	V	
鈴聲編輯系統				V
簡訊網路系統	V	V	V	V
計費系統	V			
簡訊發送元件			V	
系統測試	V	V	V	V
報告撰寫	V	V	V	V

## 第二章 簡訊服務的概念與應用

### 2.1 簡訊服務的原理與架構

簡訊服務(short message service)定義於ETSI(GSM 03.40)，最早於1991年應用於歐洲GSM數位無線通訊網路上。後來BellSouth Mobility與Nextel等先驅使其應用在北美的數位通訊網路上，1998年CDMA(Code Division Multiple Access)與TDMA(Time Division Multiple Access)技術完成，簡訊傳送技術才真正成熟。

簡訊是制定在無線通訊網路環境下，傳送有別於傳統語音(voice)的短訊息(Short Message)，主要包含兩項特點：

1. 將傳統通訊頻道中，劃分出140位元組的長度來傳送簡訊，因此簡訊最大傳送的長度限制即是140位元組，以7位元定義一個字元，最大可傳送簡訊長度為160字元。
2. 視通訊狀態選擇不同頻道傳輸簡訊，當手機處於通話狀態時，可以使用不同頻道傳送簡訊給處於通話狀況的手機。

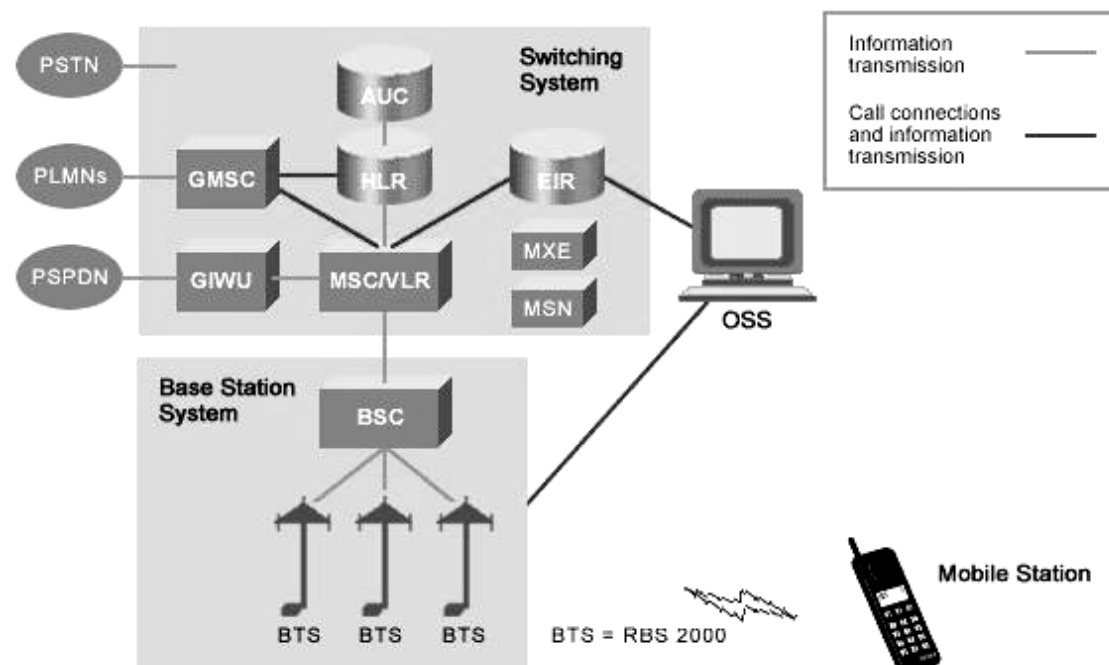


圖2-1 GSM網路原理

上圖所示為基本的GSM網路架構，具體實作根據實體在設備中分佈的不同而異。GSM網路分成三個主要系統。交換系統(Switching

System, SS)、基地台系統 (Base Station System, BSS) , 及操作和支援系統(OSS)。

### 1. 交換系統

交換系統(SS)負責執行通話處理及用戶相關功能。包括下列功能單元：

- (1) 家庭區域紀錄(HLR)：HLR是一個用來存儲和管理用戶的資料庫。包括用戶服務策略檔、本地訊息、以及活動狀態。
- (2) 行動服務交換中心(MSC)：負責系統的電話交換功能，控制發往或來自其他電話的數據系統呼叫。
- (3) 訪問者區域紀錄(VLR)：是一個包含行動服務中心(MSC)所需用戶的臨時訊息資料庫，為訪問用戶提供服務。VLR通常和MSC相結合。當一個行動終端設備漫遊到一個新的MSC時，連接到MSC的VLR將向HLR請求關於行動終端設備的資料。以後如果行動終端設備建立一個呼叫，VLR將具有建立呼叫所需的訊息，而不需要每次向HLR詢問。
- (4) 認證中心(AUC)：該單元提供認證和參數加密功能，以檢驗用戶身份及確保每次通話的機密性。
- (5) 設備訊息紀錄(EIR)：是一個包含行動設備身份訊息的資料庫，能使通話避免竊取行為或不合法的行動終端設備。AUC和EIR可作為單獨的節點執行或結合AUC/EIR的節點執行。

### 2. 基地台系統

所有無線通信相關的功能都是在BSS上實現的，它由基地台控制器(BSCS)和基本傳輸站組成(BTSs)。

- (1) BSC：基地台控制器提供一切控制功能及MSC和BTS的物理連接。它是高性能的交換器，提供移交、單元數據配置，以及控制無線電頻的功率(在基本傳輸站中)等功能，大量的BSCs服務於MSC。

(2)BTS：基本傳輸站處理無線電通訊和行動終端的接埠問題。BTS是無線電通訊設備(無線電收發器和天線)，它為網路的每個單元服務。一組BTSs由一個BSC控制。

### 3. 操作和支援系統(OSS)

操作和維設中心(OMC)與交換系統的所有設備及BSC相連接。執行OMC的系統即稱為操作和支援系統(OSS)。OSS是一個功能實體，網路工作者可以通過它對系統進行監控和控制。OSS的目的是為用戶提供一個網路概要並支援不同操作、維護機構的維護行為，這也正是GSM網路所需求的。

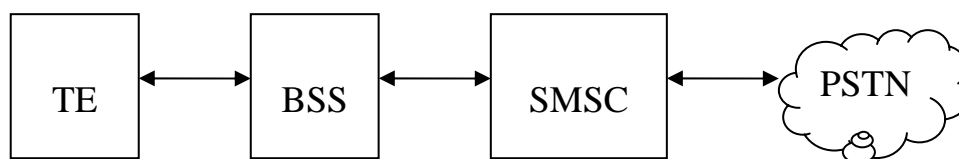
### 4. 附加功能

(1) 訊息中心(MXE)：訊息中心提供完整的語音、傳真和數據訊息。特別是MXE能處理簡訊服務、單元廣播、語音、傳真、電子郵件以及通知。

(2) 行動服務節點(MSN)：MSN是處理無線智慧網路服務的節點。

(3) 行動服務交換中心網關(GMSC)：網關是用於連接兩個網路的節點，該網關經常在MSC上工作，隨後MSC就被指定為GMSC。

(4) GSM相互作用單元(GIWU)：由軟硬體組成，提供各種網路的數據通訊接埠。通過GIWU，用戶能夠在同一個通話中交替使用語音和數據。GIWU硬體設備可以和MSC/VLR的硬體相結合。



PSTN: Public Switched Telephone Network

SMSC: Short Message Service Center

BSS: Base Station System

TE: Terminal Device

圖 2-2 簡訊傳送架構

上圖中的簡訊服務中心 (SMSC) 執行通用的SMS訊息中心功能。終端設備 (TE) 是語音和資料設備 (如電腦)，直接或間接的與簡訊服務中心相連。訊息中心包含於基地台中或與基地台同在一地。如此手機設備或終端發出的簡訊就可以在訊息中心指揮下，通過基地台或有線電話網路的幫助轉發到相對應的手機設備或終端。

執行簡訊服務時，透過SMSC的先儲存後傳送(Store-and-Forward)機制來完成，運作方式是指將來源端欲傳送的簡訊先儲存在SMSC，再經由Routing的動作傳送到目的地端。

## 2.2 簡訊服務的應用

SMS是由ETSI所制定的一個規範 (GSM 03.40 和 GSM 03.38)。根據7-bit default alphabet將每個字元以7-bits編碼的時候，最多可以發送160個字元。8-bit編碼最多140個字元，通常無法直接由手機顯示，被用來作為資料消息，例如：smart messaging中的圖片和鈴聲和OTA WAP設置。16-bit訊息最多70個字元，被用來顯示Unicode(UCS-2)本文訊息，支援大部分的手機。一個以class 0 開頭的16-bit本文訊息將在某些手機上以Flash SMS的方式顯示 (閃爍的SMS和警告SMS)。

有兩種方式來發送和接收SMS訊息：使用本文模式(Text Mode)或PDU(Protocol Description Unit)模式。文本模式 (某些手機不支援) 實際上也是一種由PDU模式展現的位元字串流編碼。在顯示SMS訊息時可能使用不同的字元集和不同的編碼方式，最常見的選擇是"PCCP437", "PCDN", "8859-1", "IRA" 和 "GSM"。如果要使用電腦應用程式來閱讀手機的資訊，須透過AT-Command中的AT+CSCS來設定。而使用手機來閱讀訊息，手機則會自行選擇合適的編碼。因此一個可以閱讀SMS訊息的應用程式，不是使用Text模式，就是使用PDU模式。如果使用Text模式，應用程式的編碼選擇將受到限制，因此本專題實作主要採用PDU模式，以便使用任何的編碼方式。

### 2.2.1 PDU數據格式分析

PDU字串表面上是一串ASCII碼，由'0'~'9'，'A'~'F'這些數字和字母組成。整個字串是以8位元的16進制數(hex decimal octets)或BCD碼的10進制數(decimal semi-octets)表示。PDU字串不僅包含了發送端的欲傳



送的訊息，還包含了很多其他訊息，如SMSC號碼、目的端號碼、回復號碼、編碼方式和服務時間等。發送和接收的PDU字串結構是不完全相同的。

例1發送：SMSC號碼是+886932400841，對方號碼是0932123456，訊息內容是”Hi”，PDU字串為：

079188962304801411000C918896232143650000AA02C834

	Octet(s)	敘述																																									
SMSC 號碼	SMSC 號碼自訂。																																										
	07	SMSC 號碼的長度。(91)+( 889623048014)。																																									
	91	SMSC 號碼類型。91 是 TON/NPI 遵守 International/E.164 標準，指在號碼前需加 '+' 號。																																									
	889623048014	SMSC 號碼。經過位移處理，實際號碼應為 886932400841 (此為中華電信 SMSC 號碼)。																																									
	SMSC 號碼直接參考手機模組裡儲存的號碼。																																										
	00	簡訊中心地址長度。"00"表示直接參考手機模組裡儲存的號碼。																																									
訊息 Header	11	<p>訊息 Header，11H=00010001B</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Bit no</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>TP-RP</td> <td>TP-UDH I</td> <td>TP-SRR</td> <td colspan="2">TP-VPF</td> <td>TP-RD</td> <td colspan="2">TP-MTI</td> </tr> <tr> <td>Value</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>欄位</th> <th>意義</th> </tr> </thead> <tbody> <tr> <td>TP-RP</td> <td>回應路徑(TP-Reply path).0-不設置;1-設置</td> </tr> <tr> <td>TP-UDH I</td> <td>用戶資料標頭標識(User data header indicator). 0-不含任何標頭訊息;1-含標頭訊息</td> </tr> <tr> <td>TP-SRR</td> <td>狀態報告要求(Status report request). 0-報告;1-不報告</td> </tr> <tr> <td>TP-VPF</td> <td>有效期限格式(Validity Period Format). 00: 提供 10: 標準 01: 預留 11: 提供半 8 位</td> </tr> <tr> <td>TP-RD</td> <td>拒絕複制(Reject duplicates). 0-接受複制;1-拒絕複制</td> </tr> <tr> <td>TP-MTI</td> <td>訊息類型提示(Message type indicator). 00-讀出(Deliver);01-提交(Submit)</td> </tr> </tbody> </table>	Bit no	7	6	5	4	3	2	1	0	Name	TP-RP	TP-UDH I	TP-SRR	TP-VPF		TP-RD	TP-MTI		Value	0	0	0	1	0	0	0	1	欄位	意義	TP-RP	回應路徑(TP-Reply path).0-不設置;1-設置	TP-UDH I	用戶資料標頭標識(User data header indicator). 0-不含任何標頭訊息;1-含標頭訊息	TP-SRR	狀態報告要求(Status report request). 0-報告;1-不報告	TP-VPF	有效期限格式(Validity Period Format). 00: 提供 10: 標準 01: 預留 11: 提供半 8 位	TP-RD	拒絕複制(Reject duplicates). 0-接受複制;1-拒絕複制	TP-MTI	訊息類型提示(Message type indicator). 00-讀出(Deliver);01-提交(Submit)
Bit no	7	6	5	4	3	2	1	0																																			
Name	TP-RP	TP-UDH I	TP-SRR	TP-VPF		TP-RD	TP-MTI																																				
Value	0	0	0	1	0	0	0	1																																			
欄位	意義																																										
TP-RP	回應路徑(TP-Reply path).0-不設置;1-設置																																										
TP-UDH I	用戶資料標頭標識(User data header indicator). 0-不含任何標頭訊息;1-含標頭訊息																																										
TP-SRR	狀態報告要求(Status report request). 0-報告;1-不報告																																										
TP-VPF	有效期限格式(Validity Period Format). 00: 提供 10: 標準 01: 預留 11: 提供半 8 位																																										
TP-RD	拒絕複制(Reject duplicates). 0-接受複制;1-拒絕複制																																										
TP-MTI	訊息類型提示(Message type indicator). 00-讀出(Deliver);01-提交(Submit)																																										
	00	訊息類型(TP-Message-Reference)。"00"表參考本身的號碼。																																									
目的端	0C	接收號碼長度。(889623214365)。																																									

號碼	91	接收號碼類型(同 SMSC 號碼類型)。																																																		
	889623214365	接收號碼，經過位移處理，實際號碼為 886932123456。																																																		
	00	<p>協議標識(TP-Protocol identifier):詳細定義參見 GSM03.40</p> <table border="1" data-bbox="667 389 1326 439"> <tr> <th>Bit no</th> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> </table> <p>一般將 Bit 7, Bit6 設為 00</p> <table border="1" data-bbox="667 472 1326 696"> <thead> <tr> <th>Bit 7</th> <th>Bit 6</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>表示 bit 0 5 如下面定義</td> </tr> <tr> <td>0</td> <td>1</td> <td>表示 bit 0 5 如下面定義</td> </tr> <tr> <td>1</td> <td>0</td> <td>保留</td> </tr> <tr> <td>1</td> <td>1</td> <td>特殊用途分配</td> </tr> </tbody> </table> <table border="1" data-bbox="667 730 1326 869"> <thead> <tr> <th>Bit 5</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>不使用 interworking，只使用 SME-to-SME 協議</td> </tr> <tr> <td>1</td> <td>使用 telematic interworking</td> </tr> </tbody> </table> <table border="1" data-bbox="667 902 1326 1375"> <thead> <tr> <th>Bits 4..0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00000</td> <td>隱含</td> </tr> <tr> <td>00001</td> <td>電傳 telex (or teletex reduced to telex format)</td> </tr> <tr> <td>00010</td> <td>group 3 telefax</td> </tr> <tr> <td>00011</td> <td>group 4 telefax</td> </tr> <tr> <td>00100</td> <td>語音 (i.e. conversion to speech)</td> </tr> <tr> <td>00101</td> <td>ERMES (European Radio Messaging System)</td> </tr> <tr> <td>00110</td> <td>National Paging System (known to the SC)</td> </tr> <tr> <td>10001</td> <td>any public X.400-based message handling system</td> </tr> <tr> <td>10010</td> <td>Internet Electronic Mail</td> </tr> </tbody> </table>	Bit no	7	6	5	4	3	2	1	0	Bit 7	Bit 6	Usage	0	0	表示 bit 0 5 如下面定義	0	1	表示 bit 0 5 如下面定義	1	0	保留	1	1	特殊用途分配	Bit 5	Description	0	不使用 interworking，只使用 SME-to-SME 協議	1	使用 telematic interworking	Bits 4..0	Description	00000	隱含	00001	電傳 telex (or teletex reduced to telex format)	00010	group 3 telefax	00011	group 4 telefax	00100	語音 (i.e. conversion to speech)	00101	ERMES (European Radio Messaging System)	00110	National Paging System (known to the SC)	10001	any public X.400-based message handling system	10010	Internet Electronic Mail
Bit no	7	6	5	4	3	2	1	0																																												
Bit 7	Bit 6	Usage																																																		
0	0	表示 bit 0 5 如下面定義																																																		
0	1	表示 bit 0 5 如下面定義																																																		
1	0	保留																																																		
1	1	特殊用途分配																																																		
Bit 5	Description																																																			
0	不使用 interworking，只使用 SME-to-SME 協議																																																			
1	使用 telematic interworking																																																			
Bits 4..0	Description																																																			
00000	隱含																																																			
00001	電傳 telex (or teletex reduced to telex format)																																																			
00010	group 3 telefax																																																			
00011	group 4 telefax																																																			
00100	語音 (i.e. conversion to speech)																																																			
00101	ERMES (European Radio Messaging System)																																																			
00110	National Paging System (known to the SC)																																																			
10001	any public X.400-based message handling system																																																			
10010	Internet Electronic Mail																																																			
	00	<p>資料編碼( Data coding scheme)</p> <table border="1" data-bbox="667 1458 1326 1507"> <tr> <th>Bit no</th> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> </table> <p>Bit No.7 與 Bit No.6 :一般設置為 00                      Bit No.5:0-本文未壓縮;                      1-本文用 GSM 標準壓縮演算法壓縮                      Bit No.4:0-表示 Bit No.1, Bit No.0 為保留位元，不含資訊類型資訊                      1-表示 Bit No.1、Bit No.0 含有資訊類型資訊                      Bit No.3, 2: 00-預設的 alphabet                      01-8bit                      10-USC2 ( 16bit )                      11-預留                      Bit No.1, 0: 00-Class 0;01-Class 1;10-Class 2(SIM 卡特定信息);11-Class 3</p>	Bit no	7	6	5	4	3	2	1	0																																									
Bit no	7	6	5	4	3	2	1	0																																												
	AA	<p>有效期(Validity-Period)。 AA 表 4 天</p> <table border="1" data-bbox="667 1966 1326 2018"> <thead> <tr> <th>VP value(&amp;h)</th> <th>對應的有效期</th> </tr> </thead> </table>	VP value(&h)	對應的有效期																																																
VP value(&h)	對應的有效期																																																			

		<table border="1"> <tr> <td>00 to 8F</td> <td>(VP+1)*5 分鐘</td> </tr> <tr> <td>90 to A7</td> <td>12 小時+(VP-143)*30 分鐘</td> </tr> <tr> <td>A8 to C4</td> <td>(VP-166)*1 天</td> </tr> <tr> <td>C5 to FF</td> <td>(VP-192)*1 周</td> </tr> </table>	00 to 8F	(VP+1)*5 分鐘	90 to A7	12 小時+(VP-143)*30 分鐘	A8 to C4	(VP-166)*1 天	C5 to FF	(VP-192)*1 周
00 to 8F	(VP+1)*5 分鐘									
90 to A7	12 小時+(VP-143)*30 分鐘									
A8 to C4	(VP-166)*1 天									
C5 to FF	(VP-192)*1 周									
	02	用戶資料長度(User-Data-Length. Length of message). 以 16 進制的 2 個字元表示。								
	C834	用戶資料(TP-User-Data).編碼後的"Hi"								

圖2-3 PDU發送數據格式分析

例2接收：SMSC號碼是+886932400841，對方號碼是0932123456，訊息內容是”你好”，PDU字串為：

0791889623048014840C91889623214365000830111290635480044F  
60597D

Octet(s)	敘述
07	SMSC 號碼資訊的長度，每八位元位元組(包括 91)
91	SMSC 地址格式(TON/NPI)，用國際格式號碼(在前面加'+')
889623048014	SMSC 地址。886932400841，補'F'湊成偶數個
84	基本參數(TP-MTI/MMS/RP)。接收，無更多消息，有回復位元元元地址
0C	回復位址數字個數，共 13 個十進位數字
91	回復地址格式(TON/NPI)，用國際格式號碼(在前面加'+')
889623214365	回復地址(TP-RA)，0932123456
00	協定標識(TP-PID)，是普通 GSM 類型，點到點方式
08	用戶資訊編碼方式(TP-DCS)，UCS2 編碼
30 11 12 90 63 54	時間戳(TP-SCTS)，2003-11-21 09:36:45 +8 時區

	80	
	04	用戶資訊長度(TP-UDL) , 實際長度 4 個位元組
	4F 60 59 7D	用戶資訊(TP-UD) , “你好”

圖2-4 PDU接收數據格式分析

### 2.2.2 PDU 訊息編碼

上一節兩例中已經出現了7-bit和UCS2編碼。現在詳細介紹這些編碼方式。在PDU模式中，可以採用三種編碼方式來對發送的內容進行編碼，分別是7-bit、8-bit和UCS-2編碼。7-bit編碼用於發送簡單的ASCII字元，它將一串7-bit的字元(最高位元為0)編碼成8-bit的數據，每8個字元可“壓縮”成7個；8-bit編碼通常用於傳送數據訊息，比如圖片和鈴聲等；而UCS2編碼用於發送Unicode字元。PDU字串的用戶訊息(TP-UD)最大是140個位元組，所以在這三種編碼方式下，可以發送的簡訊最大字元數分別是160、140和70。在此將每一個英文字母、中文字和一個位元組都視為一個字元。

而且，PDU字串的用戶訊息長度(TP-UDL)，在各種編碼方式下意義又有所不同。7-bit編碼時，指原始簡訊的字元個數，而不是編碼後的位元組數。8-bit編碼時，就是位元組數。UCS2編碼時，也是位元組數，等於原始簡訊的字元數的2倍。如果用戶訊息(TP-UD)中存在一個Header(基本參數的TP-UDHI=1)，在所有編碼方式下，用戶訊息長度都等於Header長度與編碼後位元組數之和。如果採用GSM 03.42的壓縮演算法(TP-DCS的高3位為001)，則該長度也是壓縮編碼後位元組數或Header長度與壓縮編碼後位元組數之和。

下面以訊息“hellohello”為例，其包含了10個字元，使用GSM字元集為7-bit編碼。

h	e	l	l	o	h	e	l	l	o
104	101	108	108	111	104	101	108	108	111
1101000	1100101	1101100	1101100	1101111	1101000	1100101	1101100	1101100	1101111
1101000	110010 1	11011 00	1101 100	110 1111	11 01000	1 100101	1101100	1101100	110111 1

首先將字元轉換為7個位元的二進位，然後將後面字元的位元調用到前面，補齊前面缺少的位元。例如：h翻譯成1101000，e翻譯成1100101，顯然h的二進位編碼不足八位元，那麼就將e的最後一位補足到h的前面。那麼就成了11101000(E8)，以此類推。

11101000	00110010	10011011	11111101	01000110	10010111	11011001		11101100	110111
E8	32	9B	FD	46	97	D9		EC	37

於是變成了9個8位元一組的16進制字串 E8 32 9B FD 46 97 D9 EC 37。

需要特別注意的是，7-bit的字元集和ANSI標準字元集不完全一致，在0x20以下也佈滿了一些可列印的字元，但英文字母、阿拉伯數字和常用符號的位置兩者是一樣的。因此使用7-bit編碼來收發純英文簡訊，一般情況下應該是足夠使用了，至於其他特殊字元，則要參閱GSM 03.38的規定。

8-bit編碼沒有規定具體的演算法，因此不介紹。

UCS2編碼是將每個字元按照ISO/IEC10646的規定，轉變為16位的Unicode寬字元。在.NET Framework中，命名空間system.text已提供了Unicode的編碼與解碼類別庫，因此只需調用現成的函數實現編碼與解碼即可。需注意的是，繁體中文是Big5，簡體中文則是GB2312。

### 2.3 鈴聲與待機圖片的編碼

手機鈴聲與待機圖片雖然是使用PDU模式中的8-bit編碼，然而由於各家手機廠商支援的鈴聲與圖片大不相同，因此根據不同的手機廠商便有不同的鈴聲與待機圖片編碼。本專題採用最多人使用的Nokring與SCKL格式。NOKRING(或稱為Ringing Tones Text Transfer Language, RTTTL)是Nokia Ringing tone的基本格式，最早定義在Nokia Smart Messaging 2.0 Specification，目前已成為鈴聲格式的標準。SCKL則定義在Nokia Smart Messaging 3.0 Specification中，它提供大部分的網路，如CDMA、iDEN和TDMA可以直接傳送二進位資料到行動電話。鈴聲和圖片都包含了二進位資料，透過Nokia SCKL編碼可以單單使用列印字元來傳送圖片和鈴聲。

Nokring的基本格式被”:”分成三段，分別是歌曲名稱、預設、音符。

songtitle:d=4,o=6,b=120:8g5,2c7.....

名稱區段	songtitle	歌曲的名稱，不得超過 11 個字元。														
預設區段		音符的音長。														
	d=4	<table border="1"> <tr> <td>d=</td> <td>1</td> <td>2</td> <td>4</td> <td>8</td> <td>16</td> <td>32</td> </tr> <tr> <td>音長</td> <td>全音符</td> <td>2分音符</td> <td>4分音符</td> <td>8分音符</td> <td>16分音符</td> <td>32分音符</td> </tr> </table>	d=	1	2	4	8	16	32	音長	全音符	2分音符	4分音符	8分音符	16分音符	32分音符
	d=	1	2	4	8	16	32									
音長	全音符	2分音符	4分音符	8分音符	16分音符	32分音符										
O=6	<p>音符的音階</p> <table border="1"> <tr> <td>0</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>音階</td> <td>低音</td> <td>中低音</td> <td>正常音域</td> <td>高音</td> </tr> </table>	0	4	5	6	7	音階	低音	中低音	正常音域	高音					
0	4	5	6	7												
音階	低音	中低音	正常音域	高音												
	B=120	全首鈴聲的節拍。在很多手機的 Ringing tone 中，都有節拍 (Tempo) 的出現，Nokring 也不例外。而在 Nokia PC Composer 中，Tempo 是在 95-266 之間。														
音符	8g5,2c7,....	最後一部分是音符。定義了整首鈴聲音符的音長、音階、頻率等。														

圖2-5 Norking數據格式分析

待機圖片是使用SCKL格式來編碼，主要支援的手機有Motorola，Nokia，Samsung，Siemens，Sony Ericsson等。下例為72\*13的OTA圖片在簡訊裡的傳輸編碼範例。

```
0605041582000042F45000480E01FFFFFFFFFFFFFFFF00000000
0000000000FFFFFFFFFFFFFFFF00000000000000000010F00000
00000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000
000000001
```

位元組序號	數值	描述
1	06	用戶資料頭的長度
2	05	IEI — 資訊元素標識 (應用埠位址表，16bit 埠位址)
3	04	IEDL — 資訊元素資料長度

4	15	資訊元素資料 (octets 4 & 5 --> 1582 - 目標埠)
5	82	資訊元素資料
6	00	資訊元素資料 (octets 6 & 7 --> 0000 - 初始埠)
7	00	資訊元素資料
8	64	MCC ( Mobile Country Code ) 台灣為 466
9	F6	注意:第 8 和第 9 位數值必須依據本地的行動提供商進行設置。
10	92	MNC ( Mobile Network Code ) 編碼。 (中華電信(Chunghwa)為 92 、台灣大哥大(TWN GSM 1800)為 97、遠傳電信(Far Eas Tone) 為 01、泛亞電信(TransAsia) 為 99)
11	00	資訊欄位，請參考 “ Smart Messaging Specification 3.0.0 ”。
12	48	點陣圖的寬度. 48H -> 72D
13	0E	點陣圖的高度. 0EH -> 14D
14	01	點陣圖的深度 ( 顏色灰度 )
15-140	<pre> FFF FFFFFFFFF000000000000000010F00000000000000000 00 00 00 0000000000000000                     </pre>	

圖2-6 待機圖片數據格式分析

## 2.4 簡訊發送元件

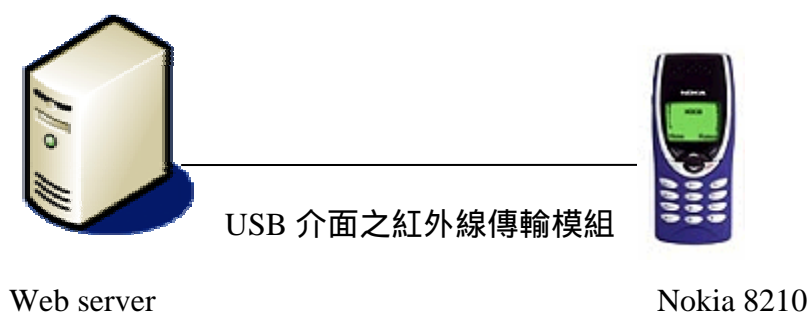


圖2-7 簡訊發送之硬體架構圖

整個簡訊發送系統是由一支Nokia 8210手機、以及一台PC所組成。PC是用來模擬Rs-232中控端，由PC端寫入手機，再經由手機傳訊息到SMSC。

### 2.4.1 計算機與手機模組的通信

現在來實現用AT-Command收發簡訊。目前在相關的系統中，不是由電信業者提供API模組，就是客戶自行利用手機模組(摩托羅拉、西門子等均有相關產品)來實現聯系簡訊服務中心。

AT 指令	功 能
AT+CMGC	Send an SMS command (發出一條簡訊命令)
AT+CMGD	Delete SMS message (刪除 SIM 卡記憶體的簡訊)
AT+CMGF	Select SMS message format(選擇簡訊資訊格式:0-PDU;1-本文)
AT+CMGL	List SMS message from preferred store (列出 SIM 卡中的簡訊 PDU/text: 0/"REC UNREAD"-未讀, 1/"REC READ"-已讀, 2/"STO UNSENT"-待發, 3/"STO SENT"-已發, 4/"ALL"-全部的)
AT+CMGR	Read SMS message (讀簡訊)
AT+CMGS	Send SMS message (發送簡訊)
AT+CMGW	Write SMS message to memory (向 SIM 記憶體中寫入待發的簡訊)
AT+CMSS	Send SMS message from storage (從 SIM M 記憶體中發送簡訊)
AT+CNMI	New SMS message indications (顯示新收到的簡訊)



AT+CPMS	Preferred SMS message storage ( 選擇簡訊記憶體 )
AT+CSCA	SMS service center address ( 簡訊中心位址 )
AT+CSCB	Select cell broadcast messages ( 選擇蜂窩廣播消息 )
AT+CSMP	Set SMS text mode parameters ( 設置簡訊本文模式參數 )
AT+CSMS	Select Message Service ( 選擇簡訊服務 )

圖2-8 SMS相關的AT指令集

計算機與手機模組是經由RS-232串列埠連結，故只需將其連接到計算機的串列埠即可。下圖是使用Windows內建的Hyper Terminal來測試計算機與手機模組的連接。

由程式集→附屬應用程式→通訊→超級終端機打開，在輸入連接的名稱後。選擇相對應的串列埠

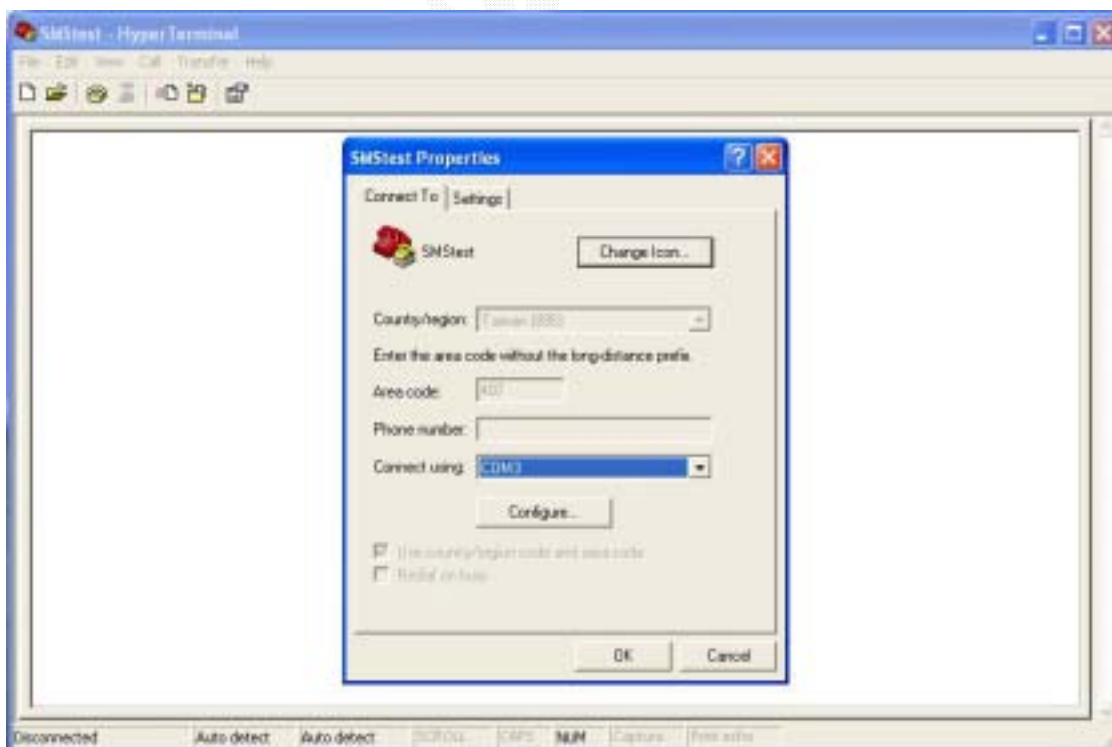


圖2-9 選擇串列埠

將埠口參數設置為：速率-115200、奇偶校驗位-none、數據位-8、停止位-1、流量控制-硬體。

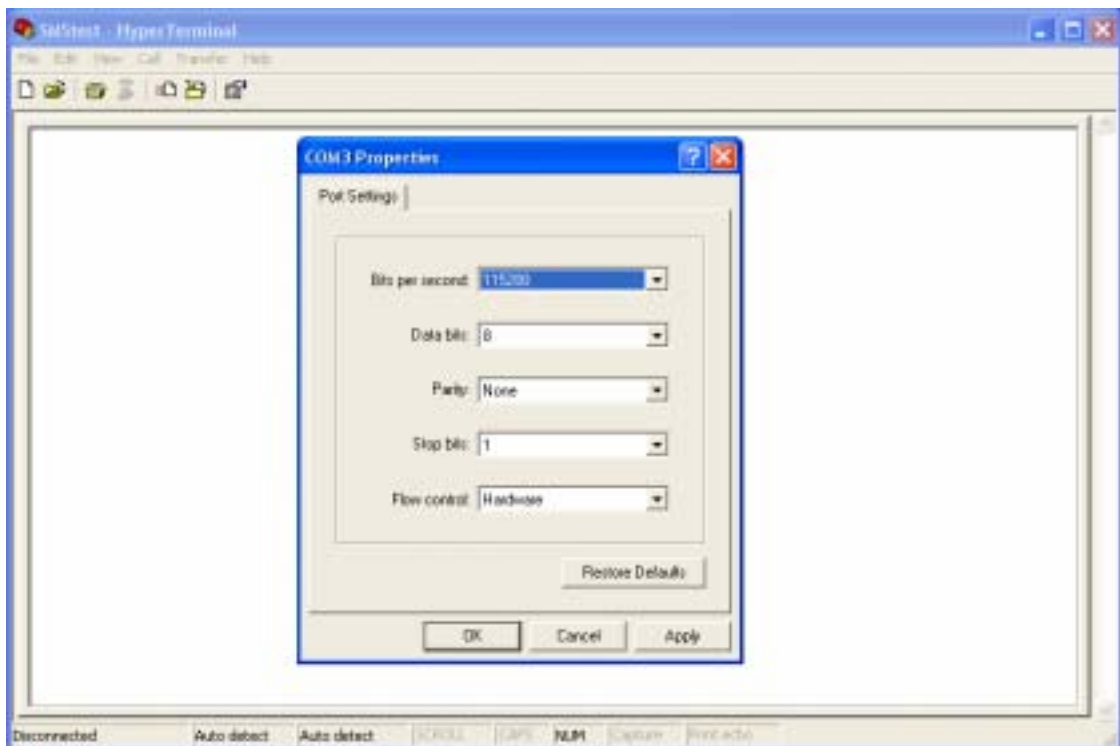


圖2-10 設定串列埠參數

輸入“AT”，螢幕上返回“OK”表明電腦與手機模組已連接成功，能夠正常工作。這時就可以測試各類AT命令。

當測試命令“AT+CMGF<CR>”時，如果返回“OK”標明手機模組支援該項指令。

如果此時手機模組處於PDU Mode（即“AT+CMGF=0<CR>”）

AT+CMGS=<length><CR>

>PDU is given<^Z/ESC>

如果短消息發送成功，則返回“OK”，並顯示錯誤訊息：

+CMGS: <mr>[,<ackpdu>]

如果短消息發送失敗，則返回如下錯誤訊息：

+CMS ERROR: <err>

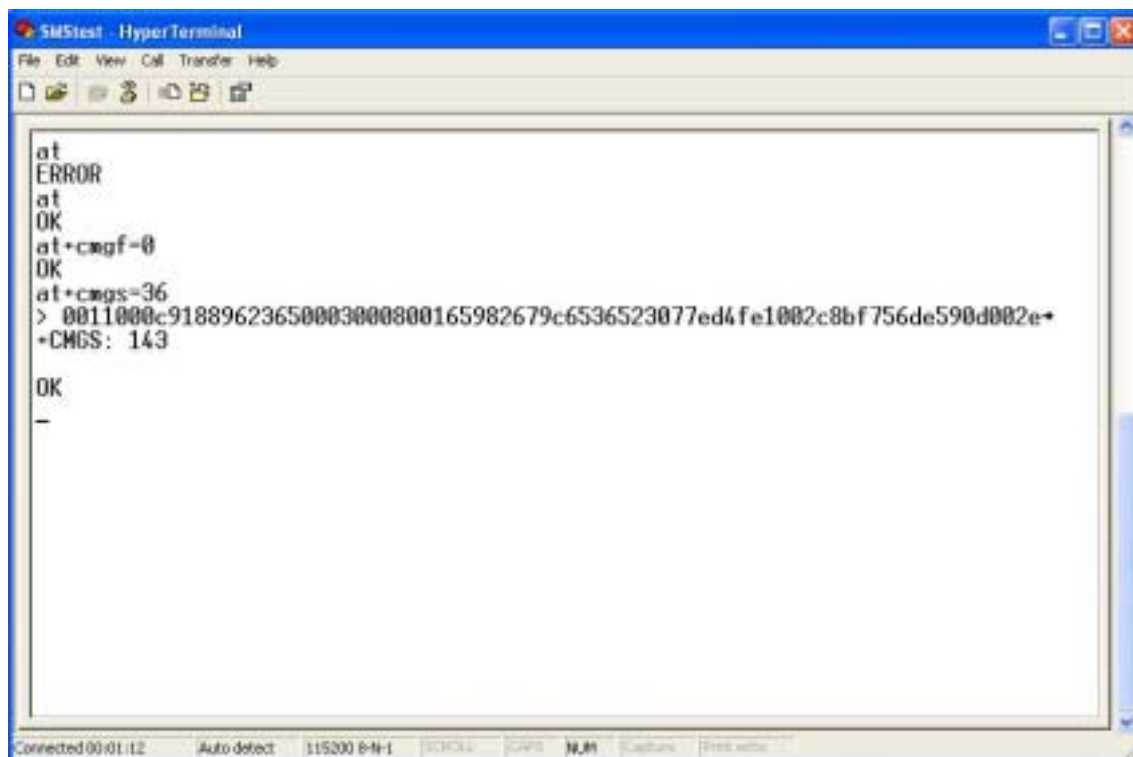


圖2-11 測試指令

## 2.4.2 程式實作

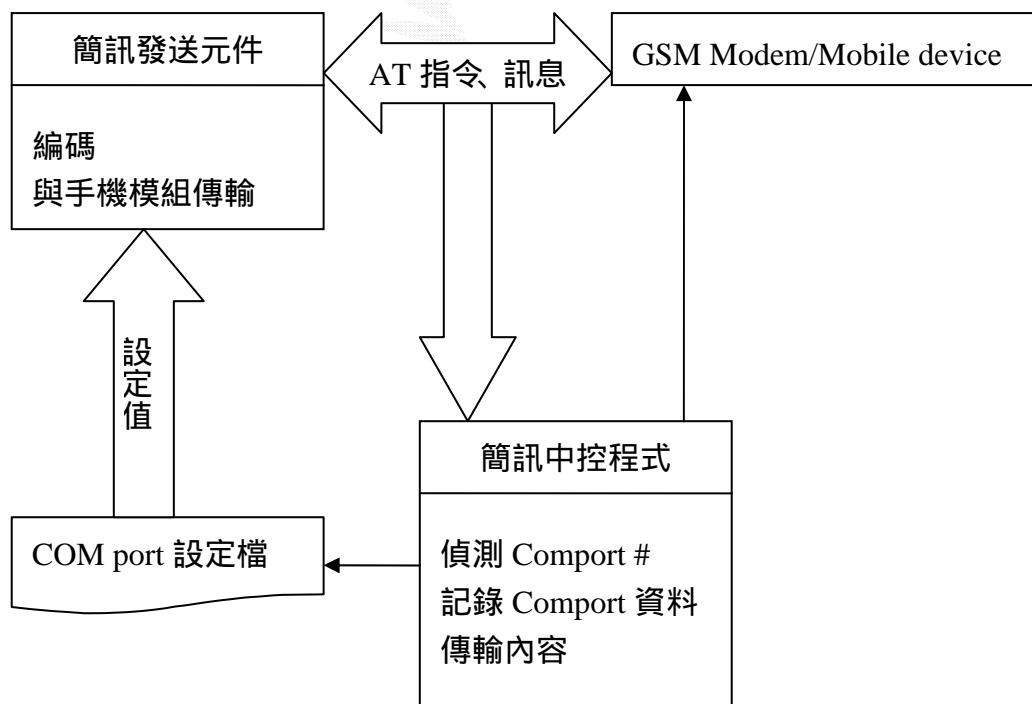


圖2-12 簡訊發送程式分析

雖然 VB.NET 中可匯入 VB 的 MSComm.ocx(Microsoft Communication Control) 控制項，控制項簡單易用，但由於必須在視窗下使用，在一些需要線程中實現通信的應用場合，控制項的使用顯示捉襟見肘。此時若能按不同需要制定靈活的串列埠通信類別將彌補控制項的不足，因此本專題採用在單線程中實現自定的串列埠通信類別。

## 1. 打開串列埠，獲取串列埠資源

通信程式從 CreateFile 指定串列埠設備及相關的操作屬性。CreateFile() 函數中有幾個必須注意的參數設置：串列埠共享方式應設為 0，串列埠為不可共享設備；創建方式必須為 OPEN\_EXISTING，即打開已有的串口。對於 dwFlagAndAttribute 參數，對串口有意義的值是 FILE\_FLAG\_OVERLAPPED，表明串列埠採用非同步通信模式，可進行重疊操作；若值為 NULL，則為同步通信方式。在同步方式下，應用程序將始終控制程序流，直到程序結束，若遭遇通信故障等因素，將導致應用程序的永久等待，所以一般多採用非同步通信。

## 2. 串列埠口設置

埠口打開後，其屬性被設置為預設值，根據具體需要，通過調用 GetCommState(mhRS, uDcb) 讀取目前埠口設備控制區塊 DCB(Device Control Block) 設置，修改後通過 SetCommstate(mhRS, uDCB) 將其寫入。再需注意非同步讀寫的超時控制設置，通過 COMMTIMEOUTS 結構設置超時，調用 SetCommTimeouts(mhRS, uCtm) 將結果寫入。以下是埠口初始成員函數：

```
Public Overloads Sub Open()
    Dim uDcb As DCB, iRc As Integer ' 取得目前DCB
    ' 設定工作模式
    Dim iMode As Integer = Convert.ToInt32(If(meMode = Mode.Overlapped, _
        FILE_FLAG_OVERLAPPED, 0))
    If miPort > 0 Then ' 初始化COM Port
        Try
            mhRS = CreateFile("COM" & miPort.ToString, _ ' 產生Com Port字串流控制
                GENERIC_READ Or GENERIC_WRITE, 0, 0, OPEN_EXISTING, iMode, 0)
            If mhRS <> -1 Then
                Dim lpErrCode As Integer
                iRc = ClearCommError(mhRS, lpErrCode, 0&) ' 清除錯誤
                iRc = PurgeComm(mhRS, PurgeBuffers.RXClear Or _ ' 清除 I/O buffers
```

```

        PurgeBuffers.TxClear)
iRc = GetCommState(mhRS, uDcb) ' 取得 COM Settings
    Dim sParity As String = "NOEM"
    sParity = sParity.Substring(meParity, 1) ' 更新 COM Settings
    Dim sDCBState As String = String.Format( _ ' 設定 DCB 狀態
        "baud={0} parity={1} data={2} stop={3}", _
        miBaudRate, sParity, miDataBit, CInt(meStopBit))
    iRc = BuildCommDCB(sDCBState, uDcb)
    iRc = SetCommState(mhRS, uDcb)
    If iRc = 0 Then
        Dim sErrTxt As String = pErr2Text(GetLastError())
        Throw New CIOChannelException( _
            "Unable to set COM state0" & sErrTxt)
    End If
        ' Setup Buffers (Rx,Tx)
    iRc = SetupComm(mhRS, miBufferSize, miBufferSize)
        ' 設定 Timeouts
Else
        ' Raise Initialization problems
    Throw New CIOChannelException( _
        "Unable to open COM" & miPort.ToString)
End If
Catch Ex As Exception
        ' Generica error
    Throw New CIOChannelException(Ex.Message, Ex)
End Try
Else
        ' Port not defined, cannot open
    Throw New ApplicationException("COM Port not defined, " + _
        "use Port property to set it before invoking InitPort")
End If
End Sub

```

### 3. 串列埠讀寫操作

主要運用 ReadFile()與 WriteFile()API 函數，若為非同步通信方式，兩函數中最後一個參數為指向 OVERLAPPED 結構的非空指標，在讀寫函式返回值為 FALSE 的情況下，調用 GetLastError()函數，返回值為 ERROR\_IO\_PENDING，表明

I/O 操作懸掛，即操作轉入後台繼續執行。此時，可以用 `WaitForSingleObject()` 來等待結束信號並設置最長等待時間。

```

If mbWaitOnRead = False Then
    ReDim mabtRxBuf(Bytes2Read - 1)
    iRc = ReadFile(mhRS, mabtRxBuf, Bytes2Read, _
        iReadChars, muOverlapped)
    If iRc = 0 Then
        iLastErr = GetLastError()
        If iLastErr <> ERROR_IO_PENDING Then
            Throw New ArgumentException("Overlapped Read Error: " & _
                pErr2Text(iLastErr))
        Else
            ' Set Flag
            mbWaitOnRead = True
        End If
    Else
        ' Read completed successfully
        RaiseEvent DataReceived(Me, mabtRxBuf)
    End If
End If

' Wait for operation to be completed
If mbWaitOnRead Then
    iRes = WaitForSingleObject(muOverlapped.hEvent, miTimeout)
    Select Case iRes
        Case WAIT_OBJECT_0
            ' Object signaled, operation completed
            If GetOverlappedResult(mhRS, muOverlapped, _
                iReadChars, 0) = 0 Then

                ' Operation error
                iLastErr = GetLastError()
                If iLastErr = ERROR_IO_INCOMPLETE Then ' 讀寫未完成
                    Throw New ApplicationException( _
                        "Read operation incomplete")
                Else
                    Throw New ApplicationException( _

```

```

        "Read operation error " & iLastErr.ToString)
    End If
Else
    ' Operation completed
    RaiseEvent DataReceived(Me, mabtRxBuf)
    mbWaitOnRead = False
End If
Case WAIT_TIMEOUT
    Throw New IOException("Timeout error")
Case Else
    Throw New ApplicationException("Overlapped read error")
End Select
End If

```

定義全域變數 `m_CommPort` 為 Rs232 的物件，通過調用類別的成員函數即可實現所需的串列埠通信功能。

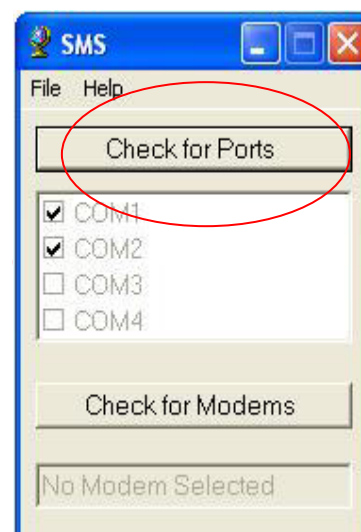
#### 4. 偵測 COM Port number

一開始先開啟簡訊中控程式，偵測目前可使用的 COM Port number。

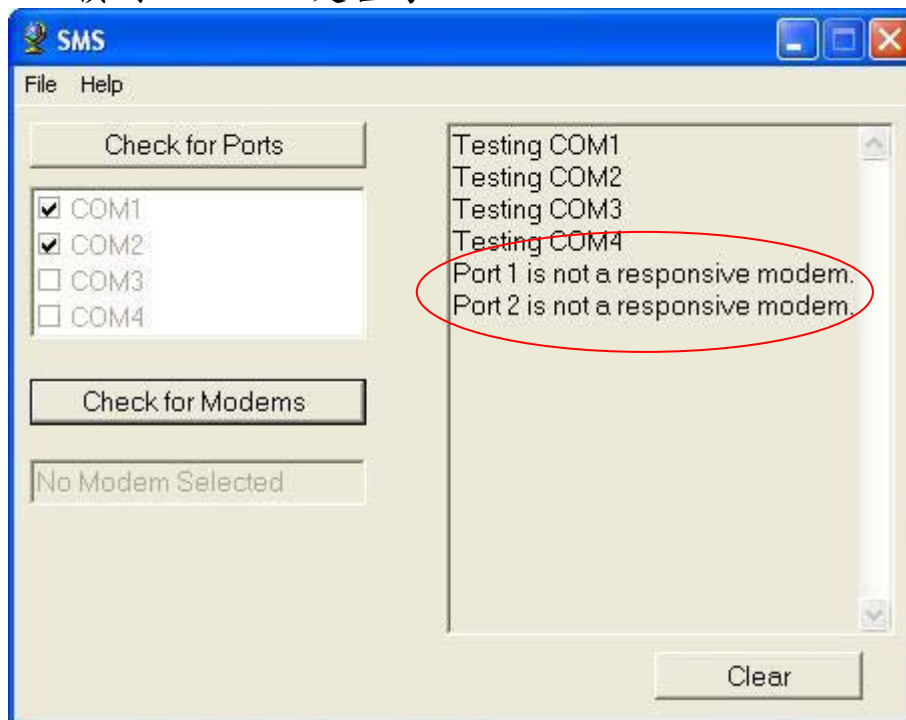
```

Private Function IsPortAvailable(ByVal ComPort As Integer) As _ Boolean
Try
    m_CommPort.Open(ComPort, 115200, 8, _
        rs232.Rs232.DataParity.Parity_None, _
        rs232.Rs232.DataStopBit.StopBit_1, 4096)
    m_CommPort.Close()
    Return True
Catch
    Return False
End Try
End Function

```



偵測 Com Port 是否為 Modem。



```
Private Function IsPortAModem(ByVal port As Integer) As Boolean
```

```
Try
```

```
    m_CommPort.Open(port, 115200, 8, rs232.Rs232.DataParity.Parity_None, _
        rs232.Rs232.DataStopBit.StopBit_1, 4096)
```

```
    m_CommPort.Write(Encoding.ASCII.GetBytes("AT" & Chr(13)))
```

```
    System.Threading.Thread.Sleep(200)
```

```
    Application.DoEvents()
```

```
Try
```

```
    Dim b As Byte
```

```
    m_CommPort.Read(1)
```

```
    m_CommPort.ClearInputBuffer()
```

```
    m_CommPort.Close()
```

```
Return True
```

```
Catch exc As Exception
```

```
    m_CommPort.Close()
```

```
Return False
```

```
End Try
```

```
Catch exc As Exception
```

```
Me.lstPorts.SetItemChecked(port - 1, False)
```

```
MsgBox("Could not open port.", MsgBoxStyle.OKOnly, Me.Text)
```



```
Return False  
End Try  
End Function
```

## 5. 發送簡訊函式

```
Public Function sendsms(ByVal msg As String) As String  
    m_CommPort.Open(4, 115200, 8, rs232.Rs232.DataParity.Parity_None, _  
        rs232.Rs232.DataStopBit.StopBit_1, 4096)  
    m_CommPort.Write(Encoding.ASCII.GetBytes("AT" & Chr(13)))  
    System.Threading.Thread.Sleep(200)  
    m_CommPort.Write(Encoding.ASCII.GetBytes("AT+CMGF=0" & Chr(13)))  
    System.Threading.Thread.Sleep(200)  
    m_CommPort.Write(Encoding.ASCII.GetBytes("AT+CMGS=" & "14" & Chr(13)))  
    System.Threading.Thread.Sleep(200)  
    m_CommPort.Write(Encoding.ASCII.GetBytes(msg & Chr(26)))  
    System.Threading.Thread.Sleep(200)  
    m_CommPort.Close()  
End Function
```



## 第三章 系統實作

### 3.1 系統架構

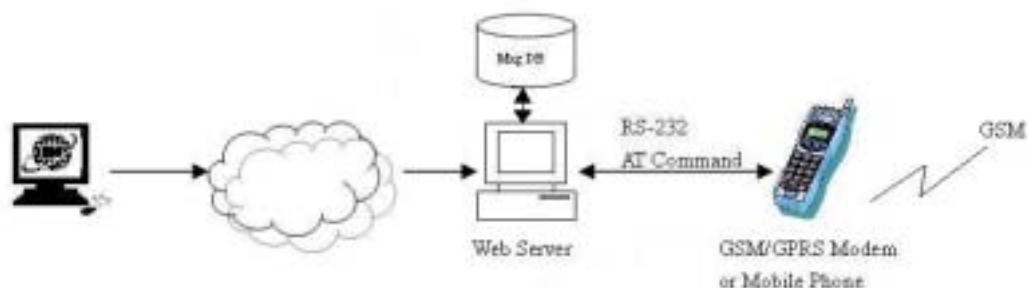


圖 3-1 簡訊服務網路系統架構

簡訊服務系統朝向結合全球資訊網(World Wide Web, WWW)方向規劃，採用了「三層式主從架構」的程式設計觀念，這種觀念包含使用者介面層、含有企業邏輯的中間層以及資料庫層。以三層式架構 Browser-WEB-Application 為主幹，並且以目前使用者最多之關聯式資料庫為整個系統之資料庫伺服器(Database Server)，使用者端朝向 Thin Client 於微軟視窗作業系統建置瀏覽器(Browser)，以減輕未來應用系統之維護管理事宜，網站伺服器結合應用程式伺服器，支援網路伺服器的工作、資料庫連結與手機無線模組之資料傳送。並以物件導向技術來分析設計整個應用軟體系統，然後透過網路系統來控管與傳送使用者端所需要的處理、傳送和接收之各式資料。

手機無線模組經由 Rs-232 與網站伺服器相連結，本專題中使用 Nokia 8210 手機取代 GSM/GPRS 數據機，並利用 USB 介面之紅外線傳輸線來傳送資料，以降低實作成本。

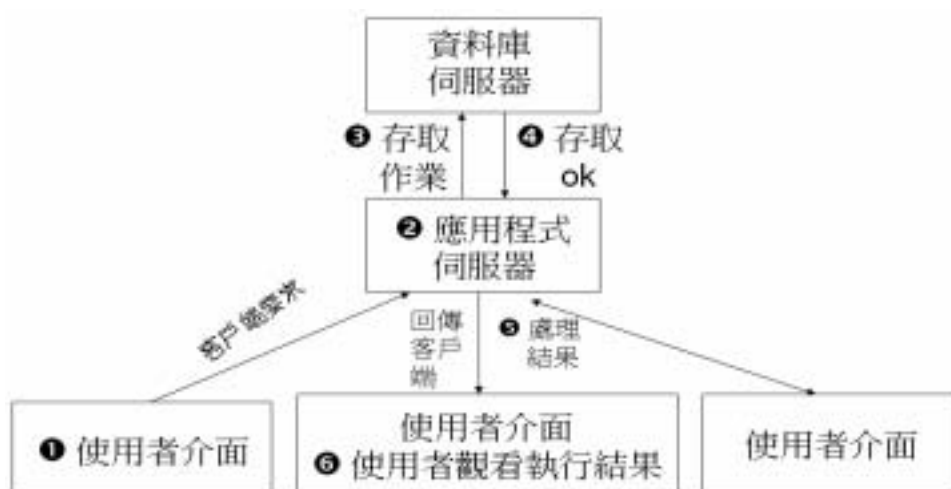


圖 3-2 3 層式主從架構圖

## 3.2 資料庫設計

### 3.2.1 SQL 簡介

結構化查詢語言(Structured Query Language)是一種對資料庫查詢、存取與管理的標準語言，SQL 是特別針對主從運算所設計的高伸縮性高效率的資料庫管理系統。他內建的資料複製功能，強有力的管理工具，以及開放的架構，賦予高階平台一個經濟又實惠的資訊解決方案。在企業致力追求有效率的流程與分散決策風險時，他們顯然越來越依賴那些能夠使用者與資訊整合的技術，其結果是企業組織逐漸以分散式運算作為資料與決策制訂的重要依據。

### 3.2.2 資料庫正規化：

#### 正規化的目的

正規化的目的就是要將資料的重複性降到最低，要是在不同的表格裡中都有會員的姓名時，一旦有個會員改了姓名，則必須同步更改更多個表格內的內容，修改過程中要是稍有遺漏有些資料沒更正，就會發生資料不一致的狀況。因此，避免資料重複是相當重要的。一般在設計資料庫時，正規化(Normalize)是必經過程，它可以盡可能減少資料重覆問題。當資料庫上線後，才可能發現某些查詢慢的離譜，這時可以“選擇性”將某些必要的欄位做反正規化，以期達到提高查詢的速度，但前提必須在不造成資料庫系統錯亂的原則下進行。合理的正規化可以有效提升效能。正規化有助於資料在新增、修改、刪除的正確性，避免出現孤兒資料。但過多的正規化會因要做許多 Join 才可獲得完整的資訊而造成查詢的效能不足。所以在本資料庫中只使用到第三正規化。

一般來說正規化對效能的好處有

1. 因為每一個資料表比較小，所以有利於單一資料表的排序、索引查詢等等。
2. 每個資料表的內容簡單、欄位少、索引少；而利於資料異動。

3. 因為資料表變小、變多，所以整個資料庫有更多的叢集索引 (clustered index)，且每個索引都可以更有效地涵蓋資料。

正規化的步驟：

STEP 1：將表格中的變動項目展開來，其結果即為一級正規化。

STEP 2：其次移去部分相關性(Partial dependency)得到二級正規化。

STEP 3：再來消除關聯間之遞移相關(Transitive dependency)最後得到三級正規化。

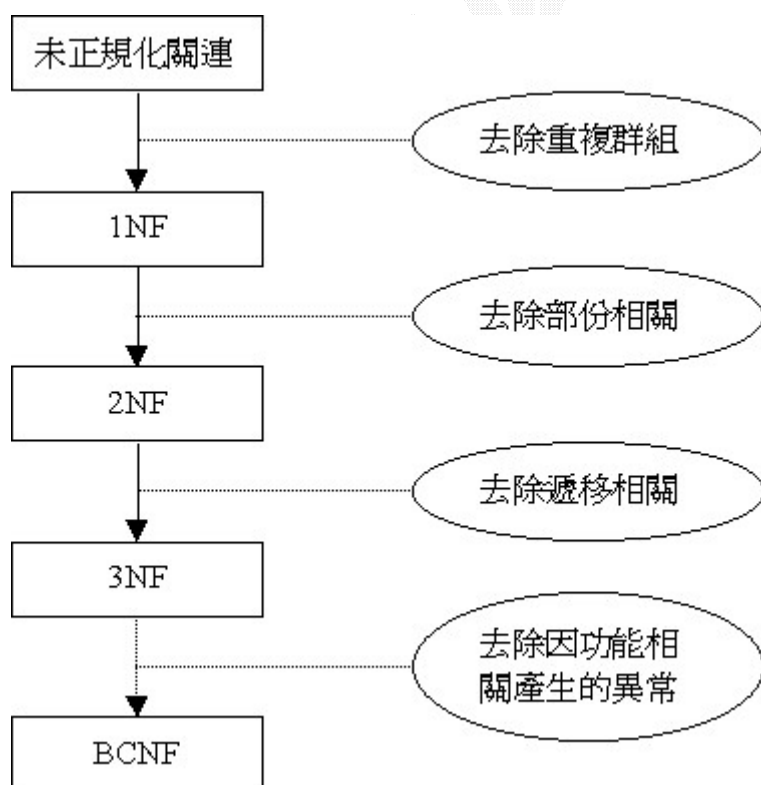


圖 3-3 正規化步驟圖

第一正規化的表格必須合以下條件：

1. 必須為 row-column 的二維式 table
2. table 的每一筆資料 (row) 只描述一件事情
3. 每一欄位只含有單一事物的特性 (欄位的唯一性)
4. 每一筆 row 的欄位內只允許存放單一值
5. 每個欄位名稱必須是獨一無二的
6. 沒有任何兩筆資料是相同的
7. row 或欄位的先後順序是無關緊要的

第二正規化的表格必須合下面條件：

所謂功能相依是指表格和表格之間的相互關係，若某個表格中有兩個欄位 A 及 B，當 A 欄位值可推導出 B 欄位值，稱功能相依性。即若有一關連 R，其屬性 Y 功能相對於屬性 X，記作  $R.X \rightarrow R.Y$ ；若且唯若 R 中有二個 X 值相同時，其 Y 值亦相同。移去部分相關性也就是要消除功能相依 (Functional Dependency)。

第三正規化的表格必須合下面條件：

消除遞移相依 (Transitive Dependency) 所謂遞移相依是指在一個表格中，如果某一欄位值可決定其他欄位值，但這些欄位中又存在某一欄位可以決定剩餘欄位值，稱為遞移相依性。若有上述情況存在，如果在刪除資料時，可能會造成其他資料損毀。一個 FD 若  $R.A \rightarrow R.B$  且  $R.B \rightarrow R.C$  則， $R.A \rightarrow R.C$  成立，此種相關性稱為遞移相關。

### 3.2.3 系統資料庫設計

本系統有 8 個資料表其中 tMember 是用來存放會員基本資料的並且有一份通訊錄 FriendList 存放會員的好友電話，在簡訊資料庫方面以 CanMessageIndex 來存放每一個罐頭簡訊的資料以及屬性，其他鈴聲圖片的資料表也是如此並且有一個外部 KEY 來說明他的類別。

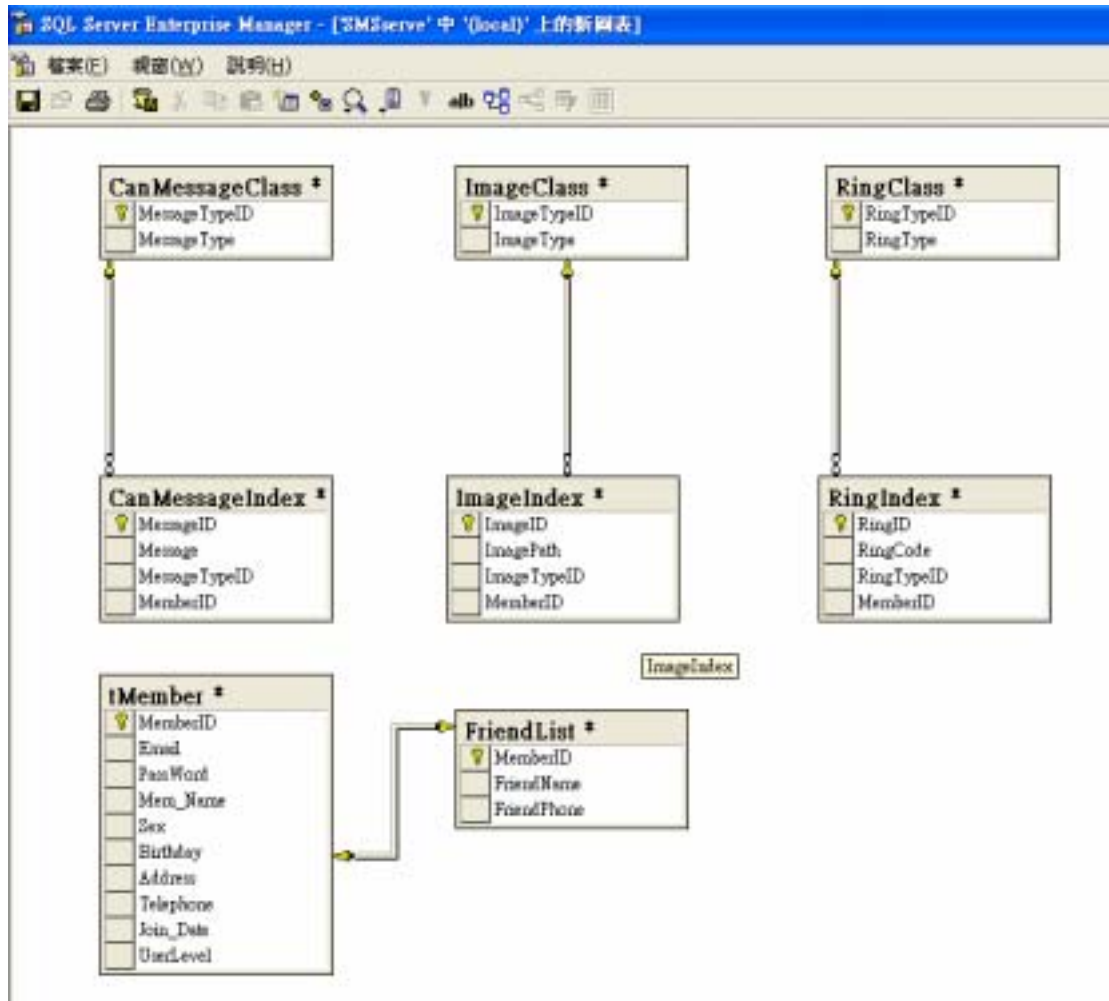


圖 3-4 資料庫關聯圖

### 資料庫關聯圖說明

ImageID:是圖片的編號是圖片資料表的主 KEY

ImagePath:是存放圖片所在位置的路徑

ImageTypeID:是用來表示圖片的類型，並且為各資料表的外部 KEY

MemberId:是會員的流水編號，並且為會員資料表的主 KEY，也是各資料表的外部 KEY，用來表示圖片的作者。

ImageType:是用來說明 ItypID 的屬性

MessageID:是表示簡訊的編號，並且為簡訊資料表的主 KEY

Message:用來存放簡訊的內容

MessageTypeID:是用數字來表示圖片的類型，並且為簡訊資料表的外部 KEY

RingID:是用來存放圖片的編號，並且為音樂資料表的主 KEY

RingCode:是存放鈴聲的編碼

RingTypeID:是用來表示音樂類型的代碼，也是音樂資料表的外部 KEY

Email:為會員的 E-mail，並且為登入帳號

Sex: 會員性別

Birthday: 會員生日

Join\_Date: 加入會員的日期

Password:為會員登入的密碼

Mem\_Name:為會員的真實姓名

UserLevel:為使用者權限

Telephone:為會員的電話

FriendName:是用來存放會員的朋友姓名

FriendPhone:是用來放會員朋友的電話，朋友的資料表仍然用 MbrID 會員資料表來查詢朋友資料表的外部 KEY

### 罐頭簡訊系統資料表

罐頭簡訊資料表主要的屬性有罐頭簡訊的流水編號、簡訊內容、簡訊的類別編號、由於簡訊的類別有太多重複的所以加入一個類別編號並設其外 foreign key 來表示次簡訊的類別至於其他的鈴聲圖片都是以此法來表示他的類別，用來避免資料重複。



圖 3-5 簡訊資料表

### 會員資料表

會員資料表是由會員流水編號、會員的 E-MAIL、密碼、會員姓名、性別、生日、地址、電話、加入日期、使用者等級所組成並且使用一個 FriendList 作為通訊錄、再以會員流水編號作為 Foreign key 來跟會員基本資料表作一對一關係。

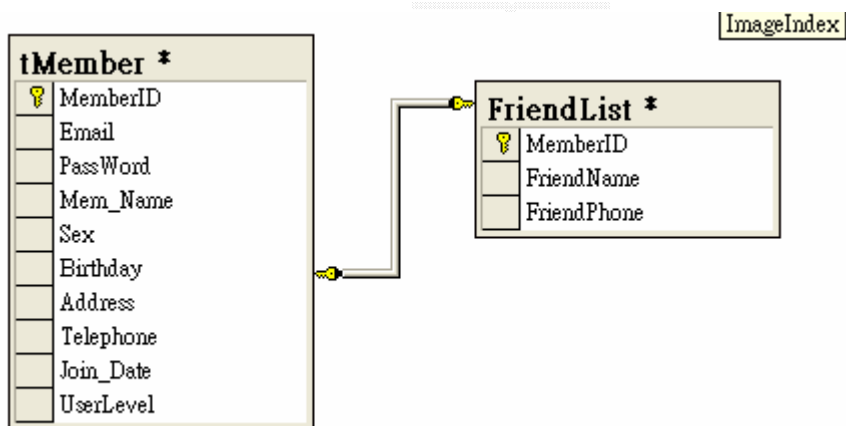
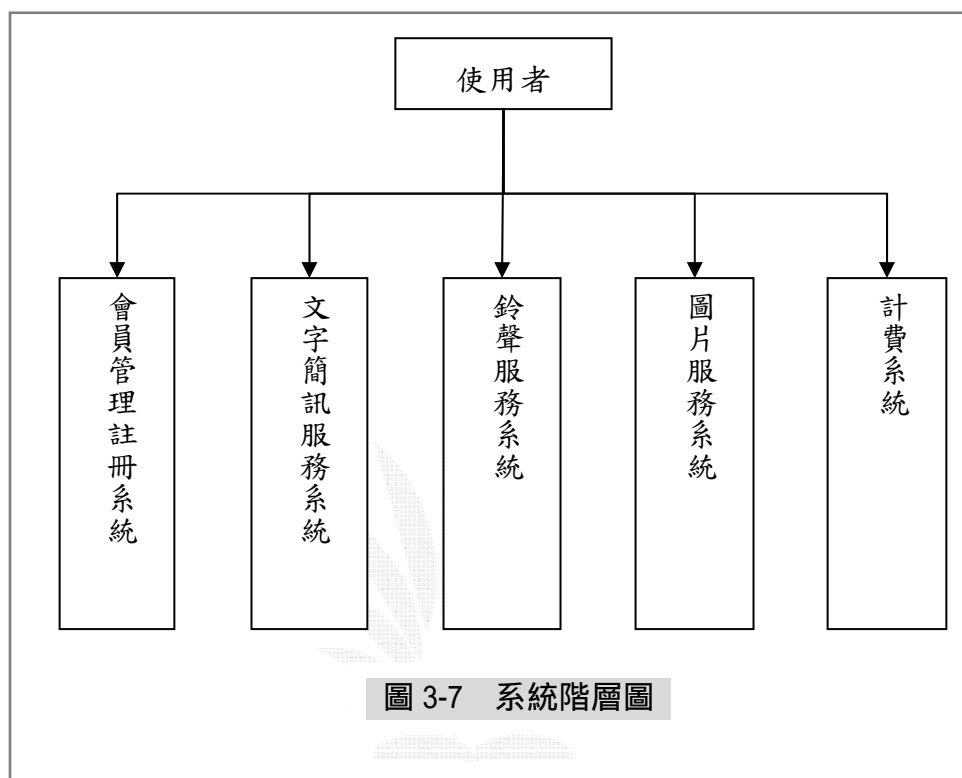


圖 3-6 會員資料表



### 3.3 網站系統架構分析

本系統以使用者觀點來看，主要分成五大子系統，1. 會員管理註冊系統、2. 文字簡訊服務系統、3. 鈴聲服務系統、4. 圖片服務系統、5 計費系統，並且使用網路作為相互連結與溝通的管道。



#### 1. 會員管理註冊系統

使用者在進入子系統前，需先註冊並登入會員方可使用。會員管理註冊系統提供會員註冊、登入、修改及查詢發送記錄等功能。

#### 2. 文字簡訊服務系統

使用者使用時需先登入網站會員，文字簡訊服務系統分為文字簡訊與罐頭簡訊。

- A. 文字簡訊在輸入接收端號碼及發送內容確認後，傳送成功後將費用記入其會員帳號中。
- B. 罐頭簡訊在用戶進入後，直接點選喜愛的簡訊，並輸入接收端號碼及發送內容，傳送成功後將費用記入其會員帳號中。

### 3. 鈴聲服務系統

使用者使用時需先登入網站會員，鈴聲服務系統分為自編鈴聲與鈴聲下載。

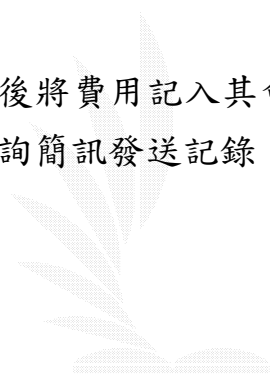
- A. 自編鈴聲在編輯好鈴聲並按下確認後，會將鈴聲格式傳送到發送網頁，輸入接收端號碼及發送內容，傳送成功後將費用記入其會員帳號中。
- B. 鈴聲下載在用戶進入後，直接點選喜愛的鈴聲，並輸入接收端號碼及發送內容，傳送成功後將費用記入其會員帳號中。

### 4. 圖片服務系統

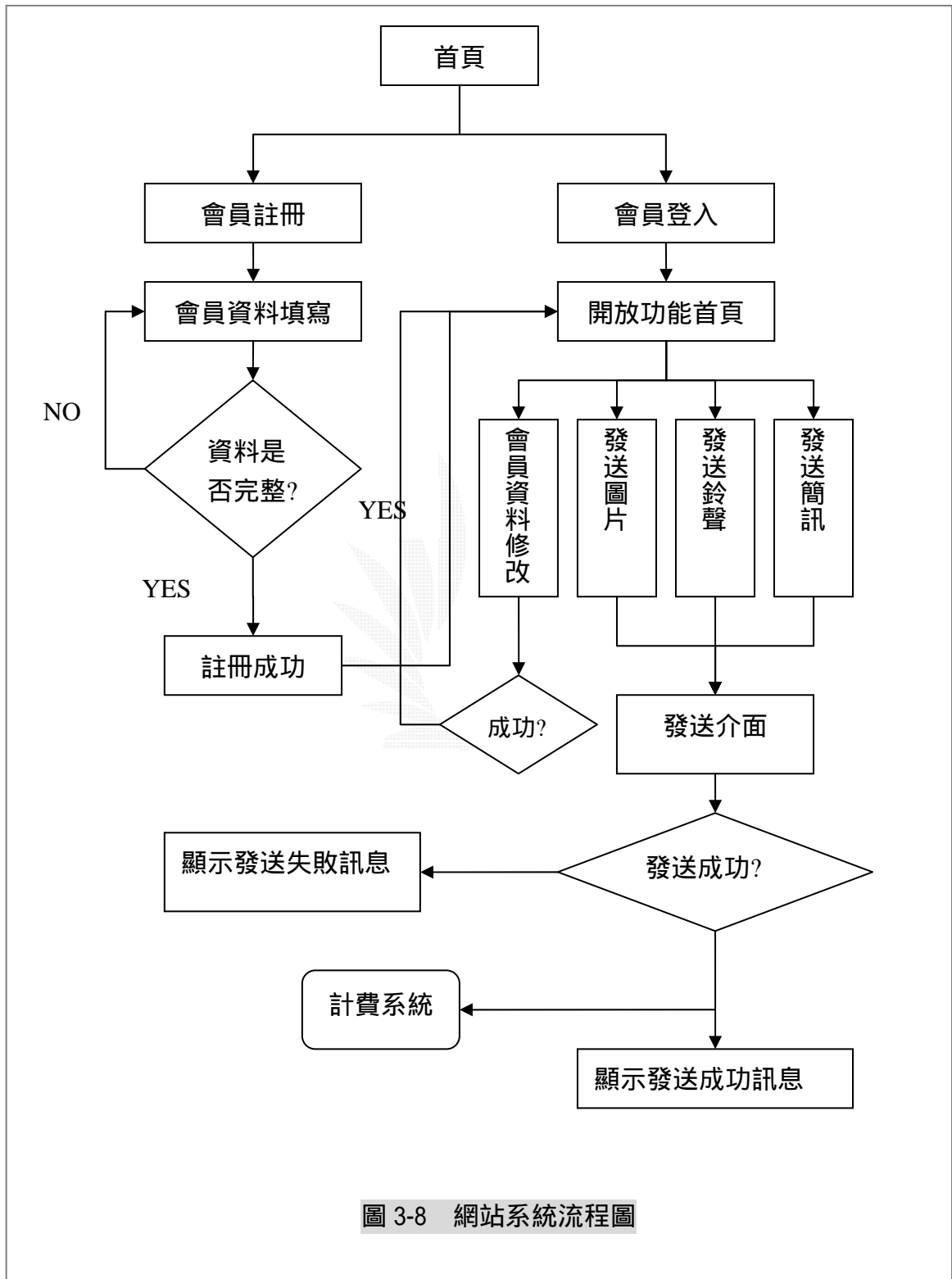
用戶進入「圖案簡訊」後，直接點選喜愛的圖片，並輸入接收端號碼及發送內容，傳送成功後將費用記入其會員帳號中。

### 5. 計費系統

用戶在簡訊傳送成功後將費用記入其會員帳號中，在點選進入「我的簡訊」後，可以查詢簡訊發送記錄。



使用者必須先進行會員註冊而後登入網站才能使用子系統所提供的功能。



### 3.3.1 會員管理註冊系統

#### 系統功能

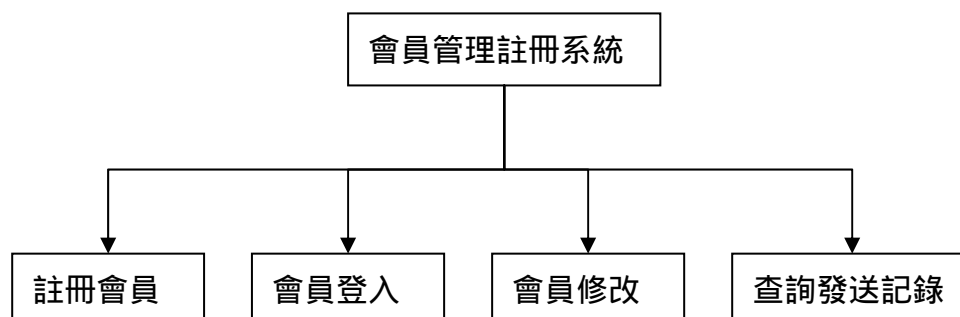


圖 3-9 會員管理註冊系統功能

會員管理註冊系統提供了 4 個功能：

1. 註冊會員：使用者可以註冊自己的會員資料。包括會員帳號、會員密碼、姓名、電話、生日、住址…等基本資料。
2. 會員登入：使用者在進入登入頁面後，當使用者輸入正確的帳號密碼，會由 session 檢查登入狀態。以避免使用者在沒有權限的情況下，企圖跳過登入程序。
3. 會員修改：使用者在登入後，可以由「我的簡訊」頁面自行修改會員密碼、電話、住址…等基本資料。
4. 查詢發送記錄：使用者在登入後，可以由「我的簡訊」頁面查詢發送記錄，包括發送日期、簡訊內容。

## 系統流程

### 1. 註冊會員：

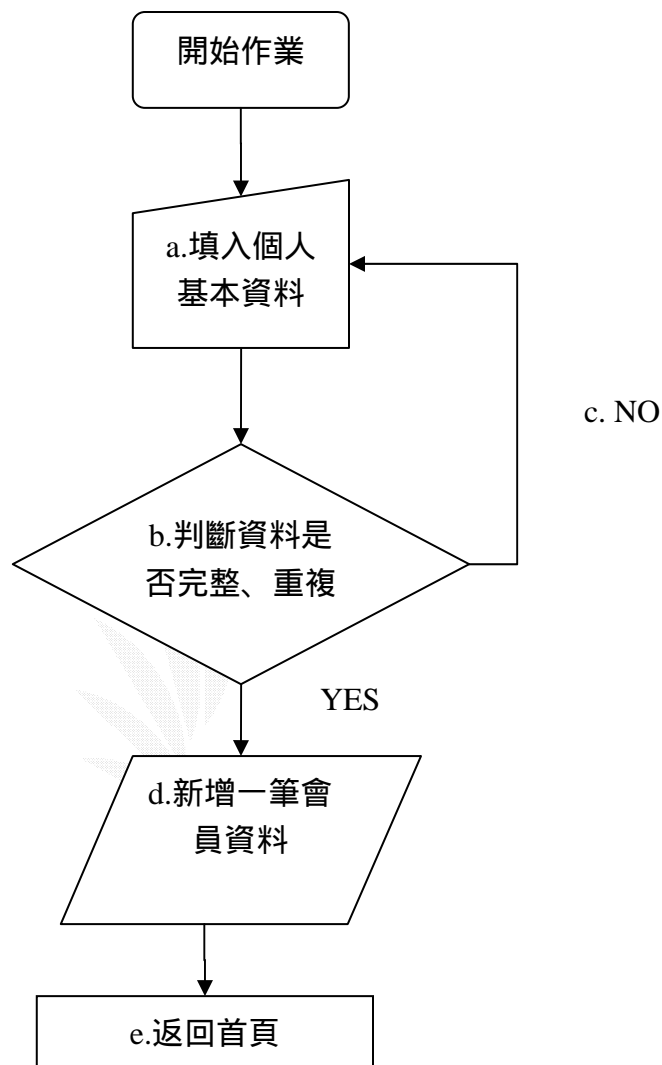


圖 3-10 會員管理註冊系統流程圖-註冊會員

- a. 需依照網頁上之欄位填寫基本資料。
- b. 系統檢查欄位資料填寫是否正確，資料庫資料是否重複。
- c. 若所填入的資料不正確、不完整或是資料重複，則會跳回 a.。
- d. 如資料正確則會由系統新增一筆資料到資料庫。
- e. 返回首頁。

## 2. 會員登入：

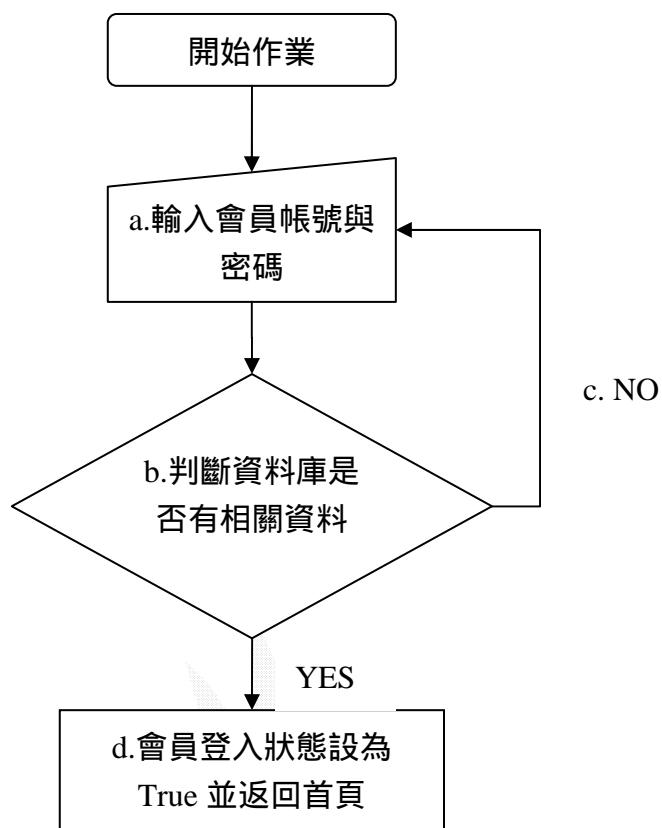


圖 3-11 會員管理註冊系統流程圖-會員登入

- a. 輸入欲登入會員之帳號與密碼。
- b. 系統會到資料庫搜尋帳號與密碼。
- c. 若無相關資料或不正確，則會跳回 a. 重新登入。
- d. 如資料正確則 session 登入狀態會設為 True 並返回首頁。

### 3. 會員修改

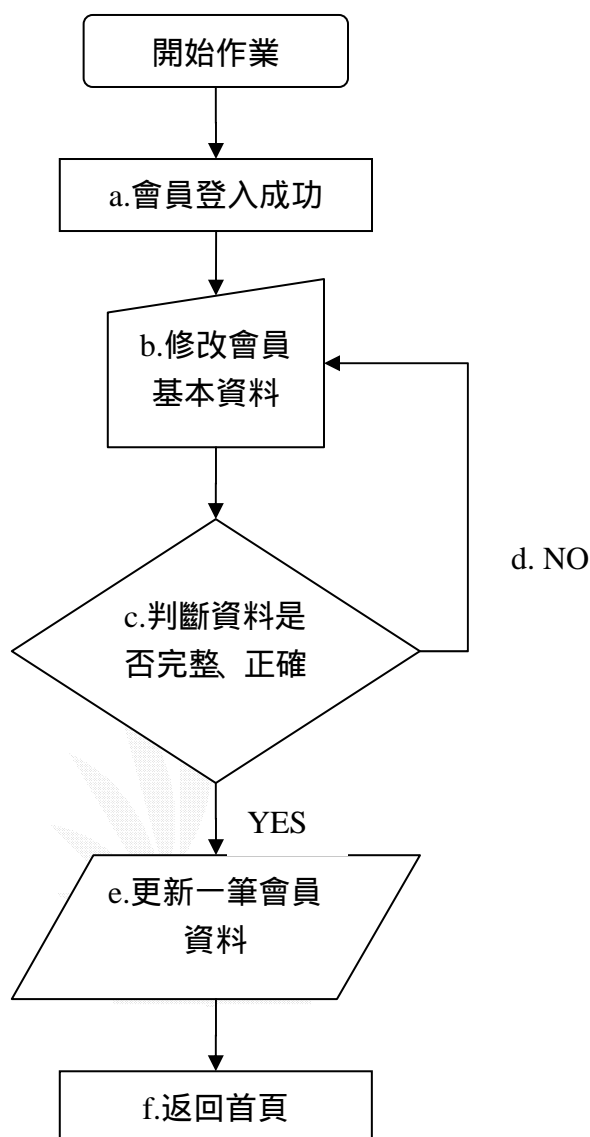


圖 3-12 會員管理註冊系統流程圖-會員修改

- a. 登入成功的會員可以進行修改帳號與密碼。
- b. 除會員帳號、姓名、生日外，可以修改此會員基本資料欄位。
- c. 系統會檢查資料及欄位的正確性，正確→e. 不正確→d.。
- d. 資料及欄位不正確，需重新填寫，返回 b.。
- e. 資料及欄位正確，更新資料庫，並返回 f. 首頁。

#### 4. 查詢發送記錄

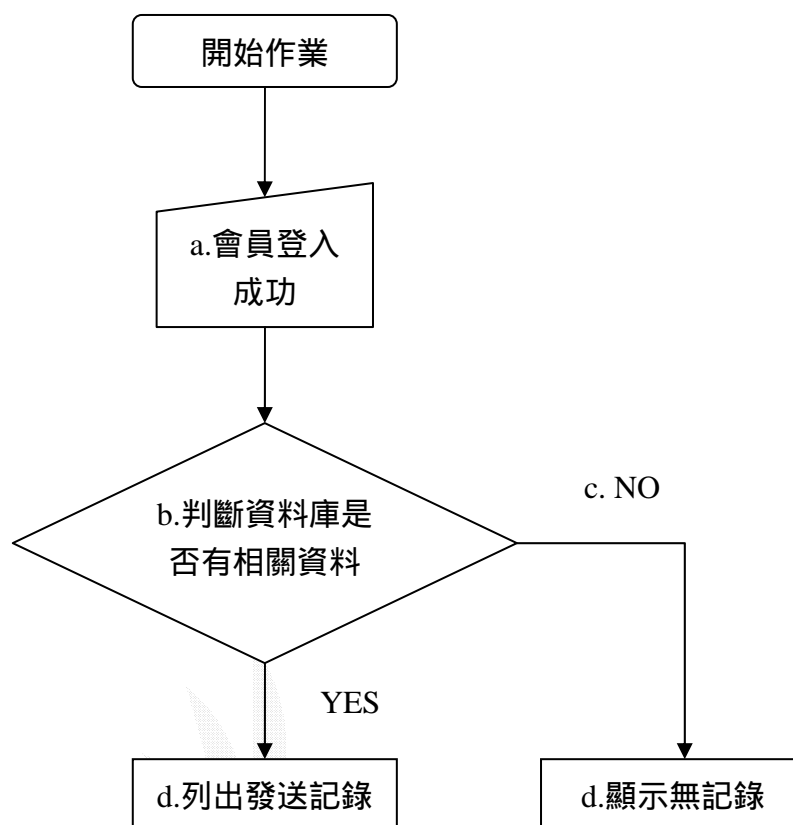


圖 3-13 會員管理註冊系統流程圖-查詢發送記錄

- a. 登入成功的會員可以查詢簡訊發送記錄。
- b. 系統會到資料庫搜尋以會員帳號關鍵字的發送記錄。
- c. 顯示無發送記錄。
- d. 有發送記錄則顯示發送記錄日期、簡訊內容。



### 3.3.2 文字簡訊服務系統

#### 系統功能

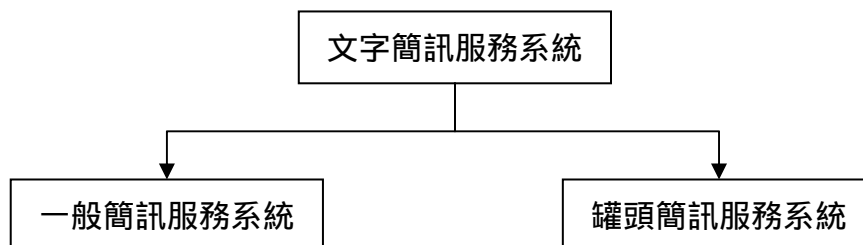


圖 3-14 簡訊系統圖

簡訊服務系統提供 2 種功能：

1. 一般簡訊服務系統：它提供一個介面給使用者輸入簡訊內容以及對方電話，讓使用者能編輯內容訊息，並且發送出去。
2. 罐頭簡訊服務系統：在罐頭簡訊裡面有許多已經分類好的現成簡訊可以直接點選，讓使用者不必再自行編寫，可以依自己的需求尋找分類並且只要點一下就可以發送。

## 系統流程

### 1. 一般簡訊服務系統

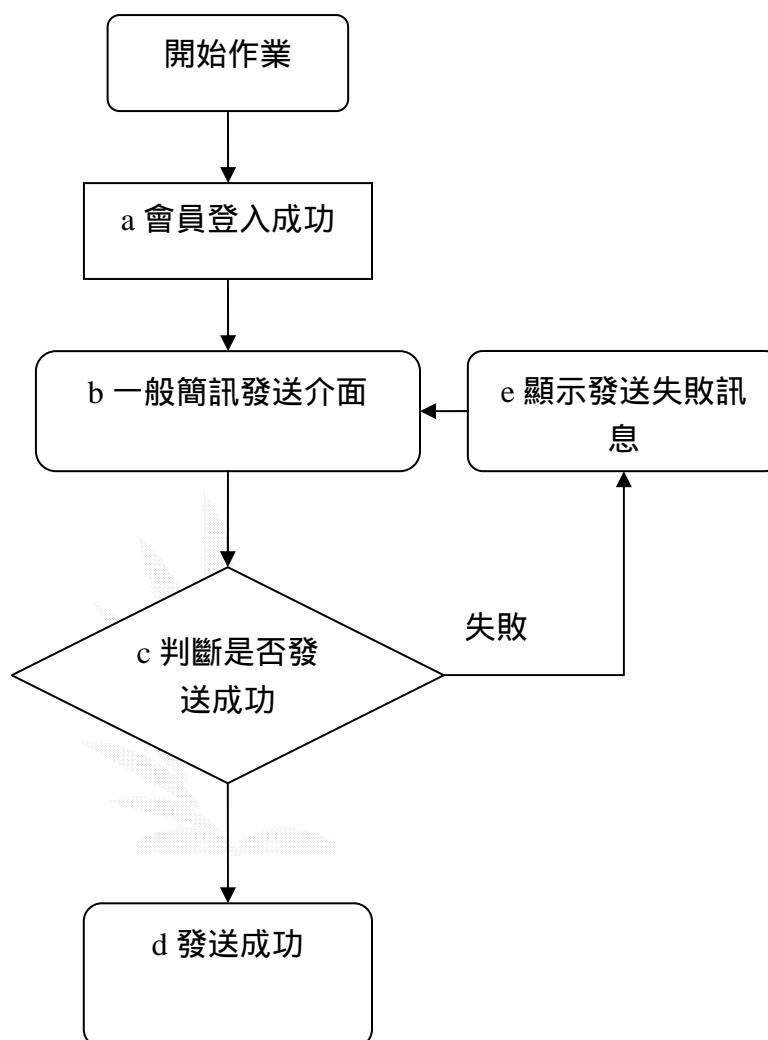


圖 3-15 簡訊發送流程圖

- a. 會員登入成功之後即可使用一般簡訊服務系統
- b. 到了簡訊發送介面之後即可鍵入對方電話以及簡訊內容
- c. 判斷簡訊發送後是否有成功
- d. 簡訊發送成功
- e. 簡訊發送失敗後會出現訊息告知使用者訊息發送失敗並且會到發送介面

## 2. 罐頭簡訊服務系統

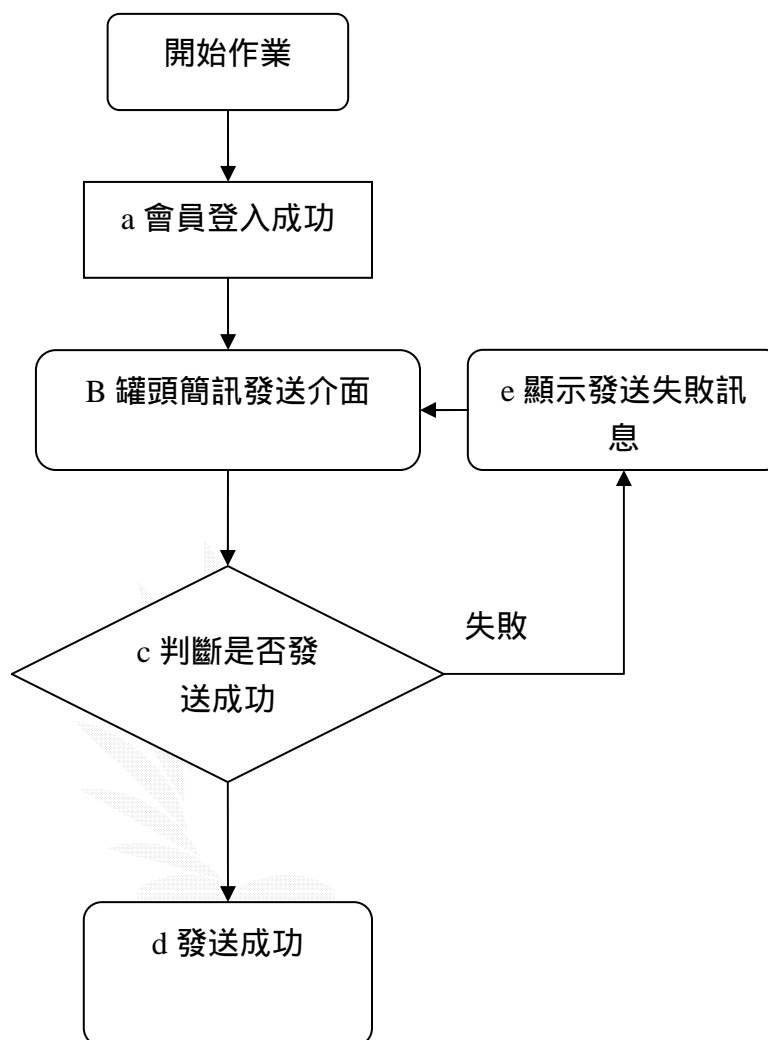


圖 3-16 罐頭簡訊服務系統流程圖

- a. 會員登入成功之後即可使用罐頭簡訊服務系統
- b. 到了罐頭簡訊發送之後可以選擇想要的罐頭簡訊，選擇之後即可到發送介面
- c. 判斷簡訊發送後是否有成功
- d. 簡訊發送成功
- e. 簡訊發送失敗後會出現訊息告知使用者訊息發送失敗並且會到發送介面

### 3.3.3 鈴聲服務系統

#### 系統功能

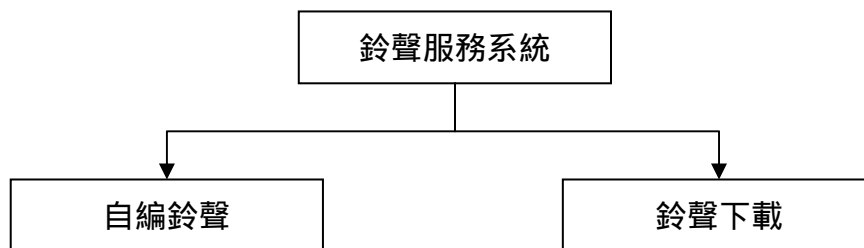


圖 3-17 鈴聲系統圖

鈴聲服務系統提供 2 種功能：

1. 自編鈴聲：它提供一個介面給使用者編輯鈴聲內容以及對方電話，讓使用者能編輯內容訊息，並且發送出去。
2. 鈴聲下載：在鈴聲下載裡有許多已經分類好的現成鈴聲可以直接點選，讓使用者不必再自行編寫，可以依自己的需求尋找分類並且只要點一下就可以發送。

## 系統流程

### 1. 自編鈴聲

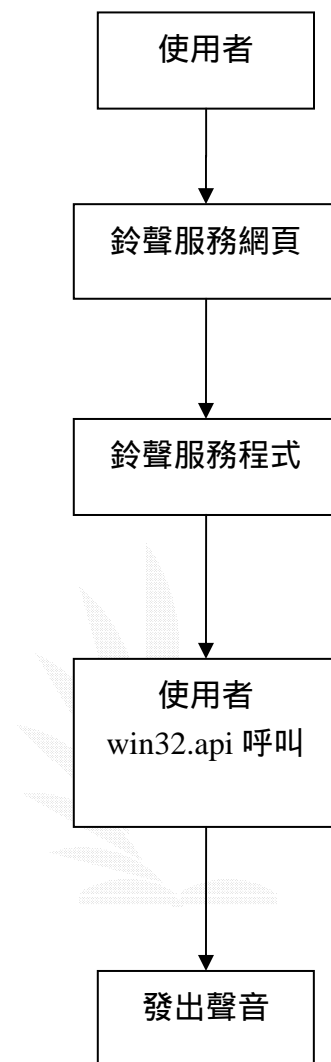


圖 3-18 鈴聲發聲流程圖

自編鈴聲的構想很簡單，就是讓使用者自己輸入音符，然後再讓那些音符發出聲音這樣就可以自己輸入音符來編曲了，然後再透過編碼把那些音符轉換成手機鈴聲格式的內碼就可以了，這部分的重點就是在讓音符發出聲音，還有轉換內碼這兩個部分，一開始先把程式寫成 win32 Api 然後再把他改寫成網路版的，使用者可以直接在網路上編曲試聽，然後再把編好的曲子透過簡訊發送系統送到使用者所想要送到的手機，可以隨自己的喜好任意編出自己要的曲子，也可以直接選擇已經預設編好的曲子送出去，然後再透過使用者計費系統紀錄你送了

多少歌曲出去來計算以後要收多少錢的依據。

NOKIA 手機的內碼:

```

<ringing-tones-text-transfer-language> :=
    <name> <sep> [<defaults>] <sep> <note-command>+
<name> := <char>+ ; maximum name length 11 characters
<sep> := ":"
<defaults> :=
    <def-note-duration> |
    <def-note-scale> |
    <def-beats> |
    <def-volume> |
    <def-style>
<def-note-duration> := "d=" <duration>
<def-note-scale> := "o=" <scale>
<def-beats> := "b=" <beats-per-minute>
<def-volume> := "v=" <volume>
<def-style> := "s=" <style>
<beats-per-minute> := 25,28,...,900 ; decimal value
<volume> := 0..15 ; 0 - no tone, 15 - maximal volume, decimal
value
<style> := 1 | ; normal
           2 | ; continuous
           3 ; staccato
; If not specified, defaults are
; 4 = duration
; 5 = scale
; 63 = beats-per-minute
; 7 = volume
; 1 = style
<note-command> :=
    [<duration>] <note> [<scale>] [<special-duration>]
<delimiter>
<duration> :=
    "1" | ; Full 1/1 note
    "2" | ; 1/2 note
    "4" | ; 1/4 note
    "8" | ; 1/8 note
    "16" | ; 1/16 note
    
```

```

    "32" | ; 1/32 note
<note> :=
    "P" | ; pause
    "C" |
    "C#" |
    "D" |
    "D#" |
    "E" |
    "F" |
    "F#" |
    "G" |
    "G#" |
    "A" |
    "A#" |
    "B" |
    "H"
<scale> :=
    "4" | ; Note A is 440Hz
    "5" | ; Note A is 880Hz
    "6" | ; Note A is 1.76 kHz
    "7" | ; Note A is 3.52 kHz
<special-duration> :=
    "." | ; Dotted note
    ";" | ; Double dotted note
    "&" | ; 2/3 length
<delimiter> := ","
; End of specification

```

依照手機內碼的編碼然後再讓程式發出那些音階所代表的頻率的聲音，然後在設定好每個音符間隔的時間，就可以讓這些編碼變成一首曲子了，這個程式遇到最大的問題就是聲音跟聲音之間會有殘音，導致聲音全部連在一起，撥放的時候聽不出正確的音，這個問題是最大的困擾，最後是直接把尾音調整成無聲，這樣聲音跟聲音之間，只要間隔不要太短就不會混在一起了。

Public Sub device()

```

Dim Index As Short
curDevice = Index - 1

```

```
rc = midiOutClose(hmidi)
rc = midiOutOpen(hmidi, curDevice, 0, 0, 0)
If (rc <> 0) Then
    MsgBox("找不到 midi 裝置, rc = " & rc)
End If
```

End Sub

這段程式碼是用來搜尋可以使用的 midi 裝置，然後開啟 midi 裝置這樣就可以利用 midi 裝置來發出聲音了，如果找不到 midi 裝置時就會出現找不到 midi 裝置的訊息，他會自動去尋找可以使用的的 midi 裝置，然後再去開啟裝置。

Module Module1

```
Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
```

```
Declare Function midiOutGetNumDevs Lib "winmm" () As Short
```

```
Declare Function midiOutGetDevCaps Lib "winmm.dll" ()
```

```
Declare Function midiOutClose Lib "winmm.dll" (ByVal hMidiOut As Integer) As Integer
```

```
Declare Function midiOutOpen Lib "winmm.dll" (ByRef lphMidiOut As Integer, ByVal uDeviceID As Integer, ByVal dwCallback As Integer, ByVal dwInstance As Integer, ByVal dwFlags As Integer) As Integer
```

```
Declare Function midiOutShortMsg Lib "winmm.dll" (ByVal hMidiOut As Integer, ByVal dwMsg As Integer) As Integer
```

End Module

這些都是程式裡面所要用到的一些函式庫，必須要先宣告才可以使用，我是把他另外寫到 module 這裡面，這樣寫有個好處就是以後要修改時可以知道自己用了哪些函式庫，如果以後要增加或是修改時，就直接到 module 裡面找，而且最重要的是非常清楚一目了然，你可以很快的知道自己用了哪些函式庫。

```
Public Sub musicbase(ByVal index As Integer)
```

```
    midimsg = &H90S + ((note + index) * &H100S) + (voule * &H10000)
```



```

midiOutShortMsg(hmidi, midimsg)
Sleep(100)
midimsg = &H80S + ((note + index) * &H100S)
midiOutShortMsg(hmidi, midimsg)

midimsg = &H0
midiOutShortMsg(hmidi, midimsg)
End Sub

```

然後這段程式碼是最主要的部分，就是控制裝置發出哪些聲音，這段是研究了最久的部分，要怎樣讓程式去發出你所想要的聲音，一開始的  $\text{midimsg} = \&H90S + ((\text{note} + \text{index}) * \&H100S)$   $\text{note}$  是設定要發出多少音階為基準的聲音基本預設值是 60 設的越高聲音越高，而  $\text{index}$  則代表要發出哪一種聲音例如 do re me……。

這是音符所對應的  $\text{index}$  的表格

音符	Do	Do#	Re	Re#	Me	Fe	Fe#	So	So#	la
index	1	2	3	4	5	6	7	8	9	10

La	La#	he
11	12	13

基本音是從 1 開始，如果你要高一點的聲音可以去把  $\text{note}$  調高，或是直接從 14 起跳或是 28 起跳，就可以設計出比較高的聲音了。

而  $\text{voule} * \&H10000$  則是控制發出聲音音量的大小，最低是 0 而最高是 255 越高則聲音的音量越大， $\text{midiOutShortMsg}(\text{hmidi}, \text{midimsg})$  就是把你設定好要發出的聲音送給  $\text{midi}$  的發聲裝置他就會發出你所設定好的聲音。

```

Sleep(100)
midimsg = &H80S + ((note + index) * &H100S)
midiOutShortMsg(hmidi, midimsg)

```

而  $\text{sleep}(100)$  的作用則是先讓程式暫停 0.1 秒，人的聽覺會有暫留的現象，這 0.1 秒就是讓第一個起頭的音停留在你的耳中，然後接下來的  $\text{midimsg} = \&H80S + ((\text{note} + \text{index}) * \&H100S)$  則是在最後發出一段較低頻率的聲音，這樣兩個合起來才是一個完整的音符，由高然後

稍微變低這樣聲音聽起來才會圓滑比較順耳，而和絃則是利用多個單音同時撥放一段時間，在搭配不同的區段用不同的和絃就是一首悅耳的曲子了，不過因為不是每支手機都有和絃功能而且每支手機的廠牌不一樣，裡面的內部設定還有編碼也不一樣，沒有標準的規格，所以主要是以 NOKIA 的手機為主，設計出單音的鈴聲撥放。

midmsg = &H0

而這個是把聲音直接變成無聲的作用，因為發出聲音會持續一段短暫的時間，所以必須要直接強制它變成無聲讓他停下來，這樣在連續撥放時才能夠讓聲音不會全都連在一起，這一段是研究最久的地方，因為資料很難找你要自己去嘗試多少的數字會發出什麼聲音，要測試許多次，然後在研究出要降低聲音讓他衰弱下來聽起來才會悅耳，這些都是花了許多時間去測試研究的，然後再撥放時遭遇到聲音連在一起的困難，測試了好幾個方法才想到要加一個結束聲音把聲音停住。

在把音符跟音符之間加上不同的拍子，就可以撥放出一段簡單的音樂了，然後在把對應的音符和拍子轉換成手機接收鈴聲訊息的格式傳送給手機，手機就可以收到你自己所編輯的鈴聲了。



而這個程式的介面是利用 vb.net 裡面的圖片按鈕，vb.net 的優點就是可以直接利用一堆元件設計出介面，很快的就可以把你所想要的介面做出來，利用它的圖片按鈕然後就可以把按鈕加入你所要的圖案，而這幾個圖片是利用小畫家做出來的圖片，在把圖片加進去按鈕裡面，就可以做出這個鍵盤了。

## 2. 鈴聲下載

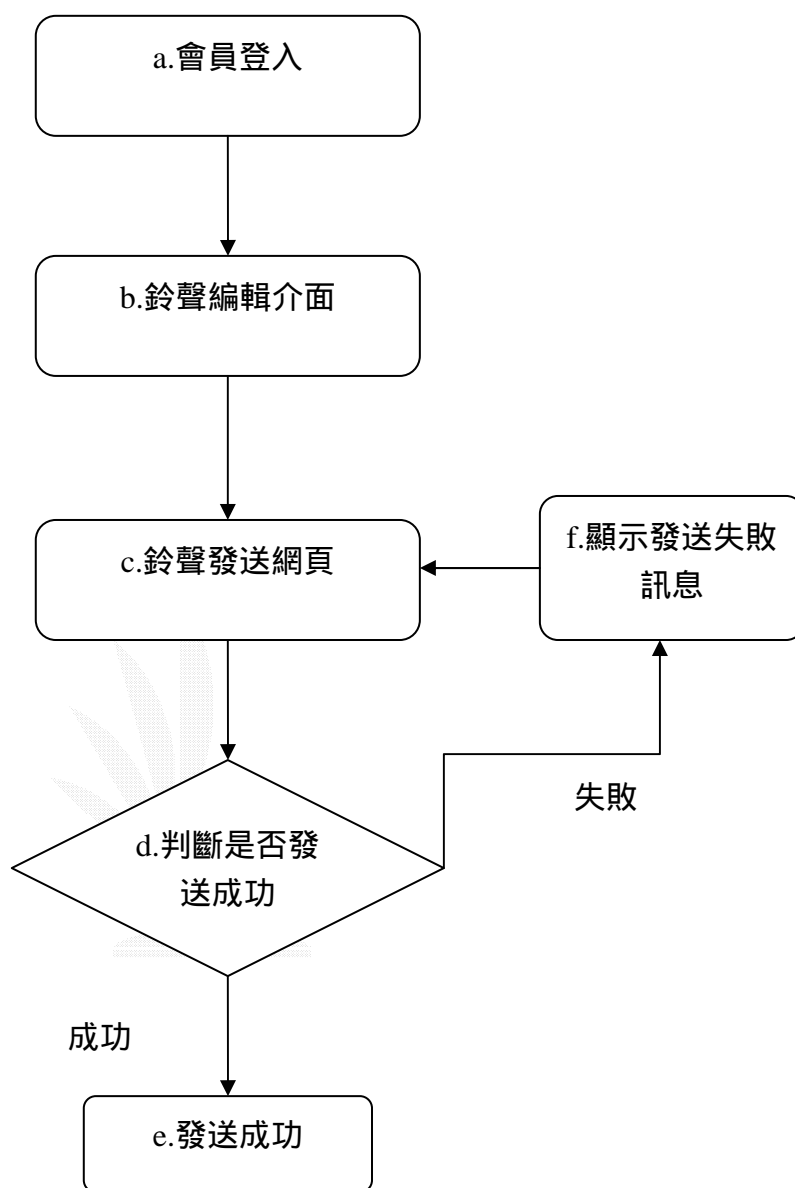


圖 3-19 鈴聲下載流程圖

- a. 會員登入成功之後即可使用鈴聲服務系統
- b. 編輯自己的鈴聲
- c. 發送網頁
- d. 判斷簡訊發送後是否有成功
- e. 簡訊發送成功
- f. 簡訊發送失敗後會出現訊息告知使用者訊息發送失敗並且會回到發送網頁

### 3.3.4 圖片服務系統

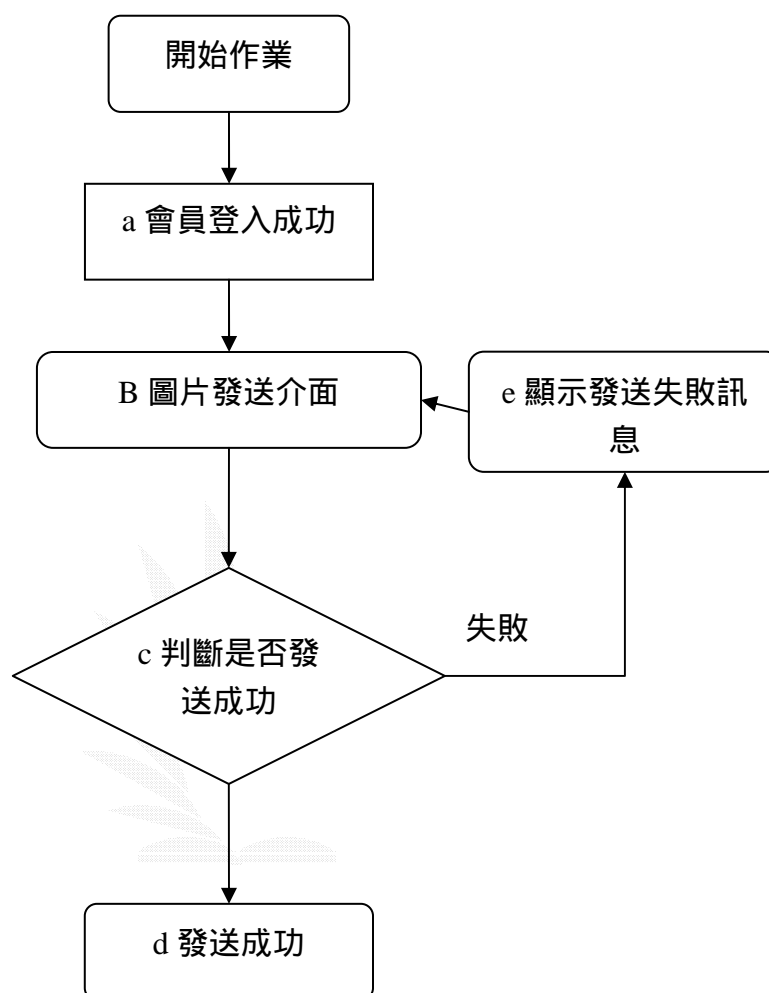


圖 3-20 圖片服務系統流程圖

- a. 會員登入成功之後即可使用圖片服務系統
- b. 到了圖片下載之後可以選擇想要的圖片訊息，選擇之後即可到發送介面
- c. 判斷圖片發送後是否有成功
- d. 圖片發送成功
- e. 圖片發送失敗後會出現訊息告知使用者訊息發送失敗並且會到發送介面。

### 3.3.5 計費系統：

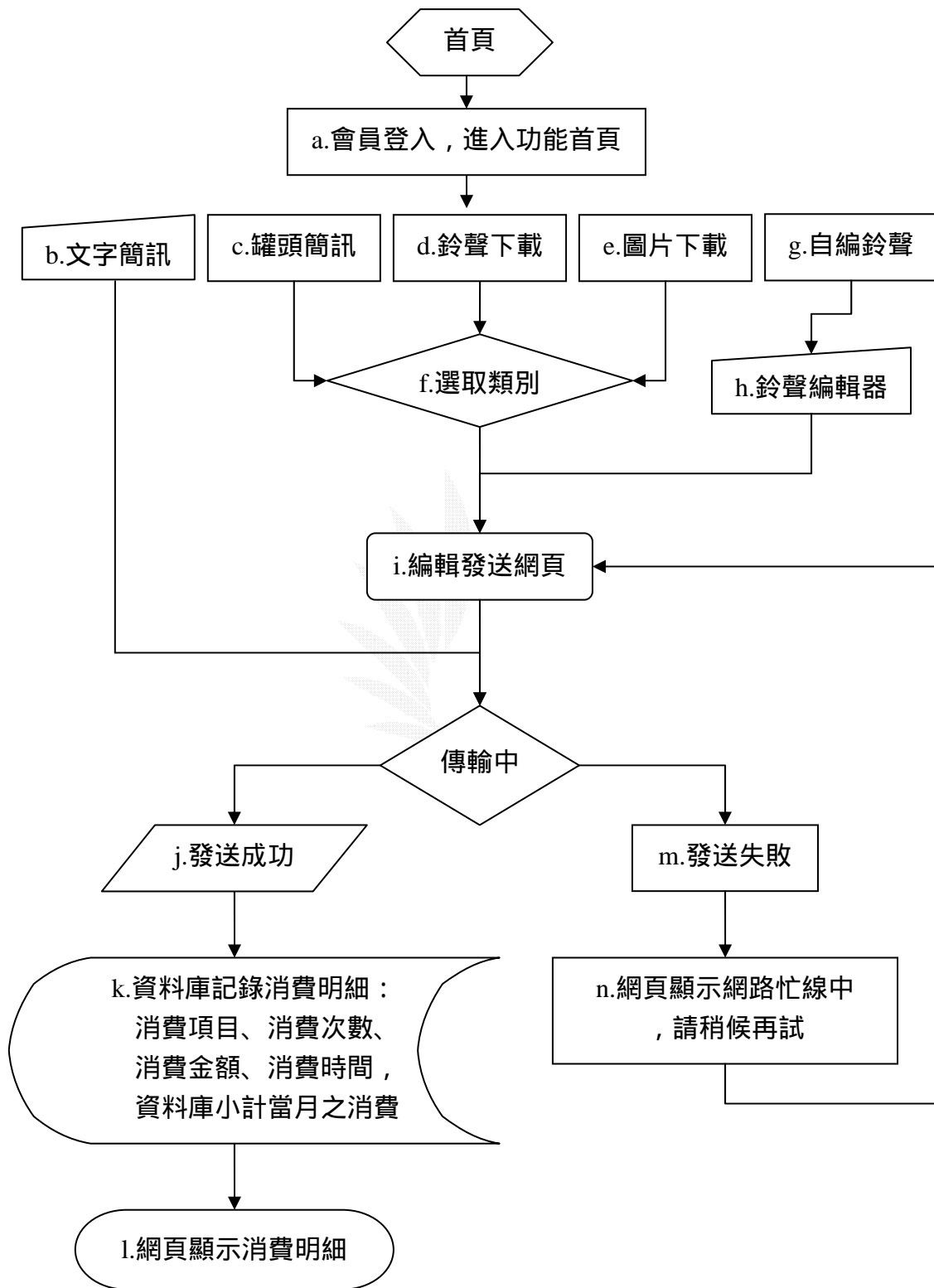


圖3-21 計費系統流程圖

**【圖 3-21】計費系統流程圖：**

- a. 會員由首頁登入之後便可連結到功能首頁，功能首頁上可以連結到 b. 文字簡訊、c. 罐頭簡訊、d. 鈴聲下載、e. 圖片下載、f. 自編鈴聲等五個不同功能的網頁。
- b. 進入文字簡訊網頁後可以在此自編簡訊，輸入要傳送的手機號碼之後按確定就會發送出去。
- c. 進入罐頭簡訊網頁之後，有提供好的罐頭簡訊可供挑選。
- d. 進入鈴聲下載網頁之後，有提供好的已編鈴聲可供下載。
- e. 進入圖片下載網頁之後，有提供好的 logo 圖片可供下載。
- f. 在 c. 罐頭簡訊、d. 鈴聲下載或 e. 圖片下載網頁中選取類別，之後挑選好自己想要的簡訊或鈴聲、圖片，在按確定之後就傳送到 i. 編輯發送網頁。
- g. 點選自編鈴聲網頁會連結到鈴聲編輯網頁。
- h. 利用鈴聲編輯器編寫出自己喜愛的取子，編完之後再傳送到 i. 編輯發送網頁。
- i. 編輯發送網頁可以重新編輯已選定的罐頭簡訊或是自編鈴聲，也可以確認挑選過要下載的鈴聲和圖片，最後再傳輸到發送元件。
- j. 發送成功，把會員消費的資料傳送到資料庫，讓資料庫作運算。
- k. 將由網頁傳送過來的會員消費資料依序輸入到資料庫裡。  
順序為 1. 分類消費項目。  
2. 消費次數+1。  
3. 計算消費金額=消費金額\*消費次數。  
4. 紀錄消費時間。  
最後結算當月累計之所有消費金額。
- l. 將所有消費資料呼叫出來顯示在網頁上作為消費明細。
- m. 發送失敗，收到失敗的傳回值之後導向網頁 n.。
- n. 顯示發送失敗的訊息：網路忙線中，請稍候再試，將網頁導回 i.。

### 3.4 實作成果

本節為展示系統實作成果，以下由系統畫面來做各個功能的介紹。



圖 3-22 系統主畫面

上圖為系統的主登錄畫面，所有系統使用者皆由此登入。

#### 3.4.1 文字簡訊服務系統部分

##### 1. 文字簡訊編輯



圖 3-23 文字簡訊編輯與發送畫面

## 2. 罐頭簡訊部分

● 簡訊類別

愛情 笑話 加油 安慰  
 道歉 祝福 分手 求婚

簡訊類別 愛情

Message	MessageID
因為你看不見我 所以不知道我在想你 因為你聽不見我 所以不知道我在念你 因為你在天涯 我在海角 所以你不知道我的淚輕輕滑落	1
今夜還吹著風 想起你好溫柔 有你的日子分外的輕鬆 也不是無影蹤 只是想得太濃 怎麼會無時無刻把你夢..(親密愛人/梅艷芳)點撥給您! 錦棠	2
收到你的簡訊我每一封會存起來,有時會看看你以前寫給我的簡訊,都讓我感覺很窩心,不知道這是不是代表我愛上你了,不知道你是不是也這樣做?!	3
每思念你一次就加深想見你的念頭,每見你一次就加深我愛你的感覺,多愛你一點就加深我想與你長相廝守的心情,願能執子之手與子偕老.	4
被不愛的人愛著是種負擔;愛著不愛的人是有所承擔;彼此相愛需要包容與分擔。 Chang	5
我是你的小風箏,飛的再遠也有一條線相繫,你是我的廣場好大一片土地,不論我掉在哪裡,你都接的住我。	6
好想念你.今生能遇見你真好.沉悶的世界因為有了你.一切都改變了.因為你是我的唯一.我們是生命共體.相信我寶貝我不會讓你心碎.永遠不會	7
在兩人的世界裡.就算沒有人會看好.但是幸福是需要我們去互相依靠.牽著手一起去尋找.相信我們的幸福就在不遠處等著	8

圖 3-24 罐頭簡訊分類選取畫面

說明：點選罐頭簡訊後，選取喜愛的訊息內容，選擇發送後，訊息內容將會傳送到罐頭簡訊發送網頁。





圖 3-25 罐頭簡訊發送畫面

### 3.4.2 鈴聲服務系統部分

#### 1. 自編鈴聲部分



圖 3-26 鈴聲編碼畫面

使用者透過介面輸入所要的音符和節奏，將使用者所想要的各個音符節奏輸入，然後再撥放出來就可以變成一首自己創作的曲子，然後按

撥放即可以試聽，直到滿意後再選擇傳送



圖 3-27 鈴聲編碼傳送畫面

將使用者輸入的音符節奏轉換成手機的編碼格式，然後再將這串編碼自編鈴聲發送網頁送給使用者



圖 3-28 自編鈴聲發送網頁畫面

## 2. 鈴聲下載部分



圖 3-29 鈴聲下載分類選取畫面

說明：點選鈴聲下載後，選取喜愛的鈴聲訊息，選擇發送後，訊息內容將會傳送到鈴聲發送網頁。



圖 3-30 鈴聲發送畫面

### 3.4.3 圖片服務系統



圖 3-31 圖片下載分類選取畫面

說明：點選圖片下載後，選取喜愛的圖片訊息，選擇發送後，訊息內容將會傳送到圖片發送網頁。



圖 3-32 圖片發送畫面

## 第四章 系統評估與討論

### 4.1 系統評估

我們一開始選定的語言是用 VB.NET，利用 .NET 元件有一個好處就是未來在子系統程式結合時較沒有相容性的問題，而且每一樣重覆的程式碼皆讓它元件化，可以節省開發時間成本。另外當初會使用 .NET 還有一個原因，就是微軟的 Windows 在市場上佔有一席之地，而且在未來所有的 Windows 系統皆內附 .NET Framework，也就是說使用者不需要再額外加裝 .NET Framework 就可以執行 .NET 程式，達到跨平台效果，比較不容易發生相容性的問題，這樣的好處就是在未來大部分的電腦都可以直接在網路上使用我們的系統，而不需要做額外的設定。

本系統還有許多未完善的地方，例如計費系統中，當客戶發送之後系統只會記下會員曾經發送過的次數，尚無法達到收費的要求。在圖鈴方面，主要支援 Nokia 與其他具有 EMS 功能的手機，沒有將市面上所有的手機包含在內。

還有一次發送多個簡訊的功能，由於目前是使用 Nokia 8210 來替代 GSM Modem，串列埠通信是獨佔資源的設備，因此當一個簡訊在發送時其他的只有等待，有鑑於傳輸速度上無法突破而放棄。

目前只有鈴聲能提供會員來做自編鈴聲，圖片只能讓使用者直接點選網頁上的圖片來傳送，無法提供會員來做自編圖片，因此也無法讓會員上傳圖片，未來要是能提供圖片上傳功能並且加入積分系統會使系統功能更強。

### 4.2 未來展望

手機在現在的社會扮演了越來越重的腳色，就連小孩也幾乎人手一機，所以在手機的市場是相當有利潤的，本系統就是針對手機使用者而發展，可能有人懶的用手機打簡訊、忘記帶手機卻有急需還是需要短時間大量發送簡訊、想要編出自己想要的音樂或是自己編輯一張手機的 logo 圖，這些都是我們系統的服務項目，因此我們的系統在未來

只要隨著潮流增加新的東西，是相當有可看性的，而且手機 MMS 現在正當紅，等發展成熟後手機將會是影音的世界，在那時候將會是更重要的一種工具、一種電話、一種娛樂，就如同日本一樣。

另外要是能紀錄每一個罐頭簡訊發送的次數紀錄下來且作排名，要是排名越高，該簡訊的作者能獲得越多積分，此積分可以用來抵傳送簡訊的費用。

### 4.3 心得討論

這次的專題是大家第一次嘗試以合作的方式完成的，以前寫程式都是獨自構想全部然後再以自己的方法把它寫出來，不用考慮太多其他因素，這次的專題是一個很好的經驗。

嘗試如何分工合作，一起設計構想整各專題的架構細節，每個人都有各自的意見風格，大家在這次專題中嘗試擷取各人的意見然後整合出來，這是一個人寫程式所無法體會到的經驗。

最重要的是程式要切割分工，大家要照一起訂立的規格、方式、語言寫出來，這樣各自完成後才能迅速的整合在一起，這才是此次專題學到最寶貴的經驗，因為未來大型程式專案絕對不是靠一兩個人就可以完成的，而是要靠一群人同心合作、提供意見、群策群力完成的。還有就是學習如何溝通，每個人的意見都不一樣，要如何去蕪存菁結合成一個大家都能接受的意見是一件很重要的事，不能硬要自己的想法被接受，不過每個人都可以提出任何想法而且被尊重；有新的想法我們都會提出來一起討論研究，不過偶而也會有意見爭執不下的時候就是了，在衝突中慢慢尋找平衡點，專題最大的意義應該就是讓我們學著如何一起去構思一份專案，除了要有自己的創意，更要試著去思考並且構思架構，從無到有。

這次專題遇到最大的困難是蒐集資料，而不是在程式的方面，因為相關的資料不好找，有一大部分時間都是花在找資料，寫程式的能力有，可是不光只是有寫程式的能力就夠了，還要程式的相關專業知識和瞭解，才有足夠的能力去把你所要的東西呈現出來。這次深深體會到資工系的不光是要有寫程式的能力，還要有能夠快速尋找資料以及迅速學習其他專業知識的能力，其他的專業能力也影響了你所寫的出來的程式的範圍，以前寫的程式都是學校老師給了題目，所以並不需

要其他相關的專業知識，這次的專題讓我們大家受益匪淺，不單只是用來畢業而已，從中學到的東西我們覺得非常的寶貴。



## 參考資料

- [1] GSM 03.38 (ETS 300 900): "Digital cellular telecommunications system (Phase 2+), Alphabets and language-specific information", GSM 03.08 version 5.4.0, ETSI, November, 1996.
- [2] GSM 03.40 (ETS 300 901): "Digital cellular telecommunications system (Phase 2+), Technical realization of the Short Message Service (SMS) Point-to-Point (PP)", GSM 03.40 version 5.4.0, ETSI, November, 1996.
- [3] GSM 07.05: "Digital cellular telecommunication system (Phase 2+), Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)", GSM 07.05 version 5.5.0, ETSI, January, 1998.
- [4] "SMART MESSAGING SPECIFICATION", version 3.0.0, Nokia Mobile Phones Ltd., December, 2000.
- [5] "SMART MESSAGING FAQ", version 2.0, Nokia Mobile Phones Ltd., September, 2002.
- [6] Steven Holzner 著, “精通 Visual Basic .Net 中文版黑皮書”, 彭靖灝譯/超維度工作室 審校, 博碩文化, May, 2003
- [7] 董大偉 編著, “ASP.NET 程式設計徹底研究”, 文魁資訊股份有限公司, May, 2003