

**DESIGNING INFORMATION SYSTEMS' APPLICATIONS
FOR PUBLIC TRANSPORT SYSTEM IN SINGAPORE**

LIM TAI CHING

NATIONAL UNIVERSITY OF SINGAPORE

2003

**DESIGNING INFORMATION SYSTEMS' APPLICATIONS
FOR PUBLIC TRANSPORT SYSTEM IN SINGAPORE**

LIM TAI CHING
(B.Eng.(Hons.), NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the following people whose help was instrumental to the successful completion of this project:

- 1) My supervisor A/P HK Garg for providing constant help, guidance and valuable advice throughout the duration of the project.
- 2) The Lab Officer of ECE-I²R, Mr Eric Siow Hong Lin, who provided the hardware and software support during the entire course of this project.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
TABLE OF CONTENTS	ii
SUMMARY	viii
NOMENCLATURE	ix
LIST OF FIGURES	xi
1 INTRODUCTION	1
1.1 Motivations	3
1.2 Objectives	4
1.3 Organization of the Thesis	6
2 BACKGROUND THEORY	7
2.1 The GSM System	7
2.2 History of GSM	8
2.3 Services Provided by GSM	10
2.4 Architecture of the GSM Network	11
2.4.1 Mobile Station	12
2.4.2 Base Station Subsystem	13
2.4.3 Network Subsystem	13
2.5 Radio Link Aspects	15
2.5.1 Multiple Access and Channel Structure	15

2.5.1.1	<i>Traffic Channels</i>	16
2.5.1.2	<i>Control Channels</i>	17
2.5.1.3	<i>Burst Channels</i>	18
2.5.2	Speech Coding	18
2.5.3	Channel Coding and Modulation	19
2.5.4	Multipath Equalization	21
2.5.5	Frequency Hopping	21
2.5.6	Discontinuous Transmission	21
2.5.7	Discontinuous Reception	22
2.5.8	Power Control	22
2.6	Network Aspects	23
2.6.1	Radio Resources Management	25
2.6.1.1	<i>Handover</i>	25
2.6.2	Mobility Management	27
2.6.2.1	<i>Location Updating</i>	27
2.6.2.2	<i>Authentication and Security</i>	29
2.6.3	Communication Management	30
2.6.3.1	<i>Call Routing</i>	30
2.7	The Short Message Service	32
2.7.1	Benefits of SMS	34
2.7.2	Network Elements and Architecture	36

2.7.2.1	<i>External Short Messaging Entities</i>	36
2.7.2.2	<i>SMSC</i>	37
2.7.2.3	<i>Signal Transfer Point</i>	37
2.7.2.4	<i>HLR</i>	38
2.7.2.5	<i>Visitor Location Register (VLR)</i>	38
2.7.2.6	<i>MSC</i>	38
2.7.2.7	<i>Air Interface</i>	38
2.7.2.8	<i>The Base Station System</i>	39
2.7.2.9	<i>The Mobile Device</i>	39
2.7.3	Signalling Elements	40
2.7.3.1	<i>Service Elements</i>	41
2.7.4	Mobile-Terminated Short Message Example	42
2.7.5	Mobile-Originated Short Message Example	44
2.7.6	SMS Applications	46
3	GENERAL PROTOTYPE SYSTEM AND HARDWARE COMPONENTS	49
3.1	Software Requirements	49
3.2	Software Overview	50
3.2.1	Software Modules	50
3.2.1.1	<i>Server Module</i>	50
3.2.1.2	<i>Static Module</i>	51
3.2.1.3	<i>Dynamic Module</i>	51
3.2.1.4	<i>Database Module</i>	52
3.3	General Prototype System	53

3.4	Hardware Equipment	54
3.5	Hardware Overview	54
3.5.1	Hardware Setup	55
3.5.2	Trouble Shooting	58
4	SMS Applications	60
4.1	Public Service Transport System	60
4.1.1	SMS Implementation and its Objectives	61
4.1.2	Functional Flow Block Diagram	63
4.1.3	Prototype System Overview	65
4.1.4	Prototype Software Developments and Testing	65
4.1.4.1	<i>Public Service Transport Database Module</i>	66
4.1.4.2	<i>Public Service Transport Static Module and Server Module</i>	67
4.1.4.3	<i>Public Service Transport Dynamic Module and Server Module</i>	70
4.2	Car Parking System	73
4.2.1	SMS Implementation and its Objectives	74
4.2.2	Functional Flow Block Diagram	75
4.2.3	Prototype System Overview	76
4.2.4	Prototype Software Developments and Testing	76
4.2.4.1	<i>Car Parking Database Module</i>	76
4.2.4.2	<i>Car Parking Software and Server Module</i>	78
4.3	SMRT And Changi Airport	80
4.3.1	SMS Implementation and its Objectives	81
4.3.2	Functional Flow Block Diagram	82

4.3.3	Prototype System Overview	83
4.3.4	Prototype Software Developments and Testing	83
4.3.4.1	<i>MRT and Airport Database Module</i>	83
4.3.4.2	<i>MRT and Airport Software Module</i>	84
4.4	Illegal Parking Ticket System	86
4.4.1	SMS Implementation and its Objectives	87
4.4.2	Functional Flow Block Diagram	88
4.4.3	Prototype System Overview	89
4.4.4	Prototype Software Developments and Testing	89
4.4.4.1	<i>Illegal Parking Fine Database Module</i>	89
4.4.4.2	<i>Illegal Parking Fine Software and Server Module</i>	90
4.5	ERP Billing System	93
4.5.1	SMS Implementation and its Objectives	94
4.5.2	Functional Flow Block Diagram	95
4.5.3	Prototype System Overview	96
4.5.4	Prototype Software Developments and Testing	96
4.5.4.1	<i>ERP Billing Database Module</i>	97
4.5.4.2	<i>ERP Billing Software and Server Module</i>	97
4.6	Vehicle Burglar Alarm System	100
4.6.1	SMS Implementation and its Objectives	100
4.6.2	Functional Flow Block Diagram	102
4.6.3	Prototype System Overview	103
4.6.4	Prototype Software Developments and Testing	104

4.6.4.1	<i>Vehicle Burglar Alarm Database Module</i>	-----104
4.6.4.2	<i>Vehicle Burglar Alarm Software and Server Module</i>	-----104
4.7	Singapore Street Directory System	-----106
4.7.1	SMS Implementation and its Objectives	-----106
4.7.2	Functional Flow Block Diagram	-----108
4.7.3	Prototype System Overview	-----109
4.7.4	Prototype Software Developments and Testing	-----109
4.7.4.1	<i>Street Directory Database Module</i>	-----109
4.7.4.2	<i>Street Directory Software and Server Module</i>	-----110
4.8	Comparison Between Applications	-----111
4.8.1	Similarities	-----111
4.8.2	Differences	-----112
5	Conclusions and Future Works	-----114
5.1	Conclusions	-----114
5.2	Recommendations for Future Works	-----115
REFERENCES		-----117
APPENDICES		-----119
Appendix A		-----119
Appendix B		-----121
Appendix C		-----223
Appendix D		-----224

SUMMARY

This thesis deals with designing SMS based information systems for the public transport system in Singapore. SMS is chosen for the project, as it is fast and efficient. Additionally, it is significantly cheaper than WAP and GPRS services and popular among the mobile phone users. Further, not all mobile phones currently in use may have WAP and GPRS features.

There were seven applications developed in the thesis:

- 1) Public service transport system,
- 2) Car parking system,
- 3) Vehicle burglar alarm,
- 4) SMRT and Changi Airport,
- 5) Illegal parking fining system,
- 6) ERP billing system, and
- 7) Singapore Street Directory.

Prototype was developed to test out the feasibility of these seven applications in real life. The prototype was designed using a single desktop Pentium PC, a GSM data cable and a GSM phone. Additionally with the help of software, the prototype was able to manage SMS requests from users' mobile phones. The software used were Nokia SDK developer, Visual Basic and Microsoft Access database.

The prototype was able to provide simple SMS information queries and more interactive features whereby users were able to do billing, booking, etc with the prototype. Constant testing and fine-tuning was done when the prototype's hardware and software were integrated together to ensure that the combination produced minimal erroneous results.

NOMENCLATURE

AGCH	Access Grant Channel
AMR	Adaptive Multi-Rate
AuC	Authentication Center
AVL	Automatic Vehicle Location
BCCH	Broadcast Control Channel
BSC	Base Station Controller
BSS	Base Station System
BTS	Base Transceiver Station
CEPT	Conference of European Posts and Telegraphs
DTX	Discontinuous transmission
EIR	Equipment Identity Register
ERP	Electronic Road Pricing
ETSI	European Telecommunication Standards Institute
FCCH	Frequency Correction Channel
FDMA	Frequency-Division Multiple Access
GIF	Graphic Interchange Format
GMSK	Gaussian-filtered Minimum Shift Keying
GSM	Global System for Mobile Communication
HLR	Home Location Register
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
ITU	International Telecommunication Union

IU	In-vehicle Unit
JPEG	Joint Photographic Experts Group
MAP	Mobile Application Part
MMS	Multimedia Messaging Service
MRT	Mass Rapid Transit
MS	Mobile Station
MSC	Mobile Switching Center
PCH	Paging Channel
PCM	Pulse Coded Modulation
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephony Network
RACH	Random Access Channel
SCH	Synchronization Channel
SDCCH	Stand-alone Dedicated Control Channels
SIA	Singapore Airlines
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SMS-C	Short Message Service Center
SS7	Signalling System Number 7
TCH	Traffic Channel
TDMA	Time-Division Multiple Access
UMTS	Universal Mobile Telecommunication System
VLR	Visitor Location Register
VLSI	Very Large Scale Integration

LIST OF FIGURES

Figure 1.1 – Information System for The Public Transport of Singapore -----	4
Figure 2.1 – General Architecture of A GSM Network -----	12
Figure 2.2 – Organization of bursts, TDMA frames, and multiframes for speech and data - -----	17
Figure 2.3 – Signalling Protocol Structure in GSM -----	24
Figure 2.4 – Call Routing for A Mobile Terminating Call -----	32
Figure 2.5 – Call Basic Network Architecture for An SMS Deployment (IS-41) -----	33
Figure 2.6 – MT – SM Scenario (GSM) -----	42
Figure 2.7– MT Short Message Scenario (IS-41) -----	43
Figure 2.8– MO—SM Scenario (GSM) -----	44
Figure 2.9– MO—SM Scenario (IS-41) -----	45
Figure 3.1 – General System Software Structure-----	50
Figure 3.2 – General Prototype System Model -----	53
Figure 3.3 – System Hardware Structure -----	55
Figure 3.4 – Installation of Nokia PC Connectivity SDK -----	56
Figure 3.5 – Data Cable Connection to Nokia Phone -----	56
Figure 3.6 – Data Cable Connection to PC serial port -----	57
Figure 3.7 – Nokia Connection Manager Detection of Nokia 8210-----	57
Figure 4.1 – Service Database GUI for Public Service Transport System -----	67
Figure 4.2 – Service Guide Search Through Road Name -----	68
Figure 4.3 – Road-to-Road Service Guide -----	68

Figure 4.4 – Stop-to-Road Service Guide with Bus Service Transfer -----	69
Figure 4.5 – Bus Arrival Times -----	71
Figure 4.6 – Bus Flagging -----	71
Figure 4.7 – SMS Alert When Reaching Specific Stop -----	71
Figure 4.8 – Prototype Road Map with Car Parks -----	77
Figure 4.9 – Car Parks’ Capacity Information -----	78
Figure 4.10 – Parking Request and Start of Parking Procedure -----	79
Figure 4.11 – End of Parking Procedure and Billing Information -----	79
Figure 4.12 – MRT Arrivals Information to Changi Airport Station -----	84
Figure 4.13 – MRT Arrivals Information based on Flight Departure -----	85
Figure 4.14 – Registering Mobile Phone Number to the Database -----	91
Figure 4.15 – Illegal Parking Fine Procedure -----	92
Figure 4.16 – Registering into ERP Billing System -----	98
Figure 4.17 – Acknowledgement of Payment from ERP Billing System -----	99
Figure 4.18 – Vehicle security Alert to Motorist -----	105
Figure 4.19 – Vehicle security Alert to Police -----	105
Figure 4.20 – Location-to-Location Query for Singapore Street Directory System -----	110
Figure 4.21 – Location-to-Location Query in Case of Traffic Jam -----	111

1 INTRODUCTION

Although technology is changing rapidly, and communications channels and computers are adapting to the increased interest in colour images, voice and video with many exciting applications being introduced, one wireless technology has taken everyone in the mobile industry by surprise – Short Messaging Service (SMS).

Few people predicted that this difficult of use service would take off. There was hardly any promotion for or mention of SMS by network operators until after SMS started to be successful. SMS advertising went from showing business people in suits entering text messages to bright pink and yellow advertisements aimed at the youth markets that adopted SMS.

Today, numerous SMS applications have been realized. One of them is SingTel SmartStock. It uses SMS service to allow users to inquire for the most current stocks information. With the service available on demand, users are able to get the stock information as and when they request it.

Singapore International Airlines (SIA) also uses SMS service to provide worldwide mobile flight alert services to its users. Users are able to specify the SMS alerts they would like to receive, such as alerts of flight departures, arrivals, delays or gate changes.

SBS Transit and SingTel [10] had developed the first SMS bus guide whereby the information system consists of SBS Transit timings service for this system is able to automatically determine the sender's location using SingTel GSM base stations around the country and inform the sender of the available bus services in the vicinity. By keying in the desired destination, a point-to-point bus travel solution will be provided

to the sender. Commuters can also request for the route of any SBS Transit bus service via this SMS service.

In this thesis, we were looking to use the low cost and universal availability of SMS to develop new innovative applications for the public transport system in Singapore. Prototypes were developed and tested to see if it was feasible and user-friendly. These prototypes were realized with the hardware of a GSM data cable, PC and a mobile phone. With a suitable program developed for the PC, any mobile phone user can send SMS to the prototype. The information system formulates a result and sends it back to the user's mobile phone.

There were seven applications developed and their prototypes designed and tested in the thesis:

- 1) Public service transport system
- 2) SMRT and Changi Airport
- 3) Car parking system
- 4) Illegal parking fining system
- 5) Vehicle burglar alarm
- 6) ERP billing system
- 7) Singapore Street Directory

In the public service transport system, a SMS based information system was designed to provide an interactive travel guide, which integrated SMRT train and bus service information. It also provided features like bus arrival times and bus stop reaching alert. Another SMS based information system was designed to combine airport flight information with SMRT train service. Passengers who intended to travel to the airport by SMRT train would be able to check when the SMRT train could take them to the airport in time for departure.

In the car parking system, a SMS based information system was designed to provide services whereby motorists could pay their parking through SMS; in addition, it provided information about the capacity of car parks.

In the case of illegal parking, a SMS based information system was designed to book any illegally parked vehicle and to inform the owner of the automobile through SMS. A prototype was designed to test out the possibility of the booking and the payment of fines through SMS.

SMS can be integrated into the vehicle security alert system such that any vehicle theft would sound off the alarm in a vehicle as well as a SMS alert message to vehicle's owner. A prototype was developed and tested out on the application.

In the ERP billing system, a SMS based information system was designed for Electronic Road Pricing (ERP) payment. The prototype was designed to test out whether it was a more convenient service than cash card payment, whereby motorists would not need to worry about topping up their cash card, while payment could be credited into their phone bills.

For the Singapore street directory, a SMS based information system was designed to provide location-to-location information through the use of SMS.

1.1 Motivations

The motivation of using SMS to design and develop applications for the public transport system in Singapore is to integrate a wireless environment into the transport system. Users will be able to use SMS to look out for travel information; motorists will be able to check out the traffic situation, car park information and pay parking fees via SMS; traffic fines by SMS will be easier and faster to process.

These applications are to provide simple, fast, efficient and economical ways of bringing information to people, as well as to make transactions such as payments of fines easier.

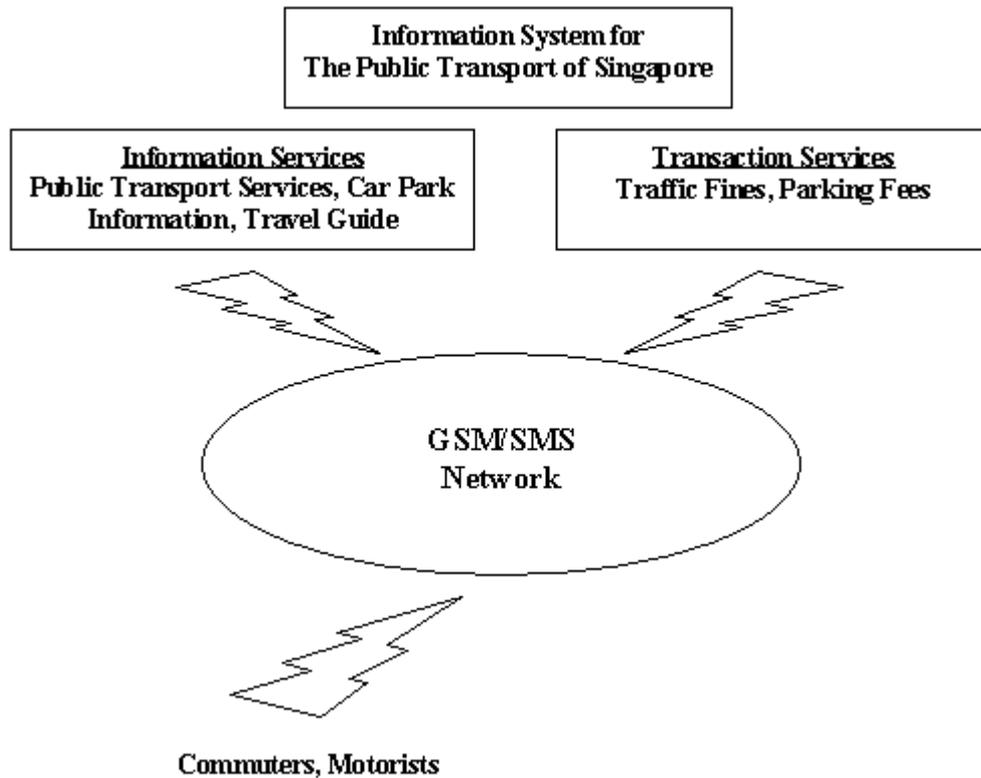


Fig. 1.1 – Information System for The Public Transport of Singapore

Prototypes produced are as close as possible to the actual applications, as they used GSM data cable and mobile phone connected to the PC. Therefore, actual SMS queries could be sent to this mobile phone, whereby a program in the PC processes the queries and provides necessary results, which in turn are sent out using the mobile phone. It is far more realistic than plain simulations that use software running on the PC.

1.2 Objectives

A prototype should be developed for each application to test out its feasibility. It should be able to establish communication with any user's mobile phone using SMS.

The thesis was aimed at meeting two main objectives for such an information system design: hardware and software.

The hardware objectives were as following:

- The components used should be reliable. The prototype was tested for its consistency in sending and receiving SMS messages, making sure that the components did not malfunction.
- The information that was being sent and received through the components should be precise and timely. The prototype was tested to see whether the components were able to convert SMS messages into text messages correctly in the software programmed on the PC.

The software objectives were as following:

- The software should be capable of analyzing the SMS message and break it down into relevant text messages.
- The software should have a good and accurate database management and communication with the user's mobile phone through SMS.
- The software should be able to perform real-time applications.
- The software should be capable of arranging the information into one or more text messages and send it back to the correct user's mobile phone by SMS.

Constant testing and fine-tuning was required when the prototype's hardware and software were integrated together to ensure that the combination produced minimal erroneous results.

Most importantly, the prototype should be user friendly after integration. Testing was done to see if a user was able to understand SMS messages received from the application when the user requested for information. The user should also understand

messages and replied accordingly if the application sent SMS options for the user to choose.

1.3 Organization of the Thesis

The thesis consists of ten chapters.

In Chapter one, we have discussed the benefits of the SMS based information system.

We proposed information systems for the public transport system in Singapore, which are analyzed and further prototypes are designed and developed. The motivation factors and objectives for the thesis are also mentioned.

The second chapter focuses on the literature reviews of the GSM system and Short Message Service. The function of the SMS is included.

In Chapter three, the general prototype system model of the seven SMS applications is illustrated. In details, the hardware setup of the prototype is also described in this chapter.

In Chapter four, the seven applications and their prototypes are described in details. The details include about information how they work, what software modules are responsible and how they are combined together to respond to a user's request. A brief comparison of the seven applications is also mentioned.

Chapter five concentrates on the contributions made during the development of the prototypes. It concludes with discussion on future work in this field.

2 BACKGROUND THEORY

This chapter is organized in the following way: after an overview of the GSM System and the services it can support, we focus on the characteristics of the SMS, and the possible set of applications which are based on these characteristics.

2.1 The GSM System

In this chapter, an overview of the GSM system will be presented to give a general flavour of GSM and the philosophy behind its design. It is a standard that ensures interoperability without stifling competition and innovation among suppliers, to the benefit of the public both in terms of cost and service quality. For example, by using Very Large Scale Integration (VLSI) microprocessor technology, many functions of the mobile station can be built on one chipset, resulting in lighter, more compact, and more energy-efficient terminals.

Telecommunications are evolving towards personal communication networks, whose objective can be stated as the availability of all communication services anytime, anywhere, to anyone, by a single identity number. Having a multitude of incompatible systems throughout the world moves us farther away from this ideal. The economies of scale created by a unified system are enough to justify its implementation, not to mention the convenience to people of carrying just one communication terminal anywhere they go, regardless of national boundaries.

The GSM system and its sibling systems operating at 1.8 GHz (called DCS1800) and 1.9 GHz (called GSM1900 or PCS1900, and operating in North America) are a first approach at a true personal communication system. The SIM card is a novel approach that implements personal mobility in addition to terminal mobility. Together with

international roaming, and support for a variety of services such as telephony, data transfer, fax, Short Message Service, and supplementary services, GSM comes close to fulfilling the requirements for a personal communication system: close enough that it is being used as a basis for the next generation of mobile communication technology in Europe, the Universal Mobile Telecommunication System (UMTS).

GSM is a very complex standard, but that is probably the price that must be paid to achieve the level of integrated service and quality offered while subject to the rather severe restrictions imposed by the radio environment.

2.2 History of GSM

During the early 1980s, analog cellular telephone systems were experiencing rapid growth in Europe, particularly in Scandinavia and the United Kingdom, but also in France and Germany. Each country developed its own system, which was incompatible with everyone else's in equipment and operation. This was an undesirable situation, because not only was the mobile equipment limited to operation within national boundaries, which in a unified Europe were increasingly unimportant, but there was also a very limited market for each type of equipment, so economies of scale and the subsequent savings could not be realized.

The Europeans realized this early on, and in 1982 the Conference of European Posts and Telegraphs (CEPT) formed a study group called the Groupe Spécial Mobile (GSM) to study and develop a pan-European public land mobile system. The proposed system had to meet certain criteria:

- Good subjective speech quality
- Low terminal and service cost
- Support for international roaming

- Ability to support handheld terminals
- Support for range of new services and facilities
- Spectral efficiency
- ISDN compatibility

In 1989, GSM responsibility was transferred to the European Telecommunication Standards Institute (ETSI), and phase I of the GSM specifications was published in 1990. Commercial service was started in mid-1991, and by 1993 there were 36 GSM networks in 22 countries [16]. Although standardized in Europe, GSM is not only a European standard. Over 200 GSM networks (including DCS1800 and PCS1900) are operational in 110 countries around the world. In the beginning of 1994, there were 1.3 million subscribers worldwide [18], which had grown to more than 55 million by October 1997. With North America making a delayed entry into the GSM field with a derivative of GSM called PCS1900, GSM systems exist on every continent, and the acronym GSM now aptly stands for Global System for Mobile communications.

The developers of GSM chose an unproven (at the time) digital system, as opposed to the then-standard analog cellular systems like AMPS in the United States and TACS in the United Kingdom. They had faith that advancements in compression algorithms and digital signal processors would allow the fulfillment of the original criteria and the continual improvement of the system in terms of quality and cost. The over 8000 pages of GSM recommendations try to allow flexibility and competitive innovation among suppliers, but provide enough standardization to guarantee proper inter-working between the components of the system. This is done by providing functional and interface descriptions for each of the functional entities defined in the system.

2.3 Services Provided by GSM

From the beginning, the planners of GSM wanted ISDN compatibility in terms of the services offered and the control signalling used. However, radio transmission limitations, in terms of bandwidth and cost, do not allow the standard ISDN B-channel bit rate of 64 kbps to be practically achieved.

Telecommunication services can be divided into bearer services, teleservices, and supplementary services. The most basic teleservice supported by GSM is telephony. As with all other communications, speech is digitally encoded and transmitted through the GSM network as a digital stream. There is also an emergency service, where the nearest emergency-service provider is notified by dialing three digits (similar to 911).

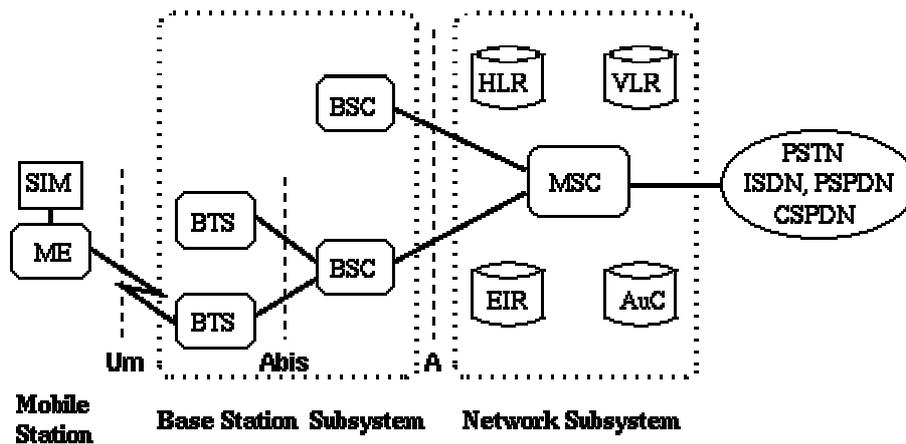
A variety of data services are offered. GSM users can send and receive data, at rates up to 9600 bps, to users on POTS (Plain Old Telephone Service), ISDN, Packet Switched Public Data Networks, and Circuit Switched Public Data Networks using a variety of access methods and protocols, such as X.25 or X.32. Since GSM is a digital network, a modem is not required between the user and GSM network, although an audio modem is required inside the GSM network to inter-work with POTS.

Other data services include Group 3 facsimile, as described in ITU-T recommendation T.30, which is supported by use of an appropriate fax adaptor. A unique feature of GSM, not found in older analog systems, is the Short Message Service (SMS). SMS is a bi-directional service for short alphanumeric (up to 160 bytes) messages. Messages are transported in a store-and-forward fashion. For point-to-point SMS, a message can be sent to another subscriber to the service, and an acknowledgement of receipt is provided to the sender. SMS can also be used in a cell-broadcast mode, for sending messages such as traffic updates or news updates. Messages can also be stored in the SIM card for later retrieval [14].

Supplementary services are provided on top of teleservices or bearer services. In the current (Phase I) specifications, they include several forms of call forward (such as call forwarding when the mobile subscriber is unreachable by the network), and call barring of outgoing or incoming calls, for example when roaming in another country. Many additional supplementary services will be provided in the Phase 2 specifications, such as caller identification, call waiting, multi-party conversations.

2.4 Architecture of the GSM Network

A GSM network is composed of several functional entities, whose functions and interfaces are specified. Figure 2.1 shows the layout of a generic GSM network. The GSM network can be divided into three broad parts. The Mobile Station is carried by the subscriber. The Base Station Subsystem controls the radio link with the Mobile Station. The Network Subsystem, the main part of which is the Mobile services Switching Center (MSC), performs the switching of calls between the mobile users, and between mobile and fixed network users. The MSC also handles the mobility management operations. Not shown is the Operations and Maintenance Center, which oversees the proper operation and setup of the network. The Mobile Station and the Base Station Subsystem communicate across the Um interface, also known as the air interface or radio link. The Base Station Subsystem communicates with the Mobile services Switching Center across the A interface.



SIM	Subscriber Identity Module	BSC	Base Station Controller	MSC	Mobile services Switching Center
ME	Mobile Equipment	HLR	Home Location Register	EIR	Equipment Identity Register
BTS	Base Transceiver Station	VLR	Visitor Location Register	AuC	Authentication Center

Fig. 2.1 – General Architecture of A GSM Network

2.4.1 Mobile Station

The mobile station (MS) consists of the mobile equipment (the terminal) and a smart card called the Subscriber Identity Module (SIM). The SIM provides personal mobility, so that the user can have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another GSM terminal, the user is able to receive calls at that terminal, make calls from that terminal, and receive other subscribed services.

The mobile equipment is uniquely identified by the International Mobile Equipment Identity (IMEI). The SIM card contains the International Mobile Subscriber Identity (IMSI) used to identify the subscriber to the system, a secret key for authentication, and other information. The IMEI and the IMSI are independent, thereby allowing personal mobility. The SIM card may be protected against unauthorized use by a password or personal identity number.

2.4.2 Base Station Subsystem

The Base Station Subsystem is composed of two parts, the Base Transceiver Station (BTS) and the Base Station Controller (BSC). These communicate across the standardized Abis interface, allowing (as in the rest of the system) operation between components made by different suppliers.

The Base Transceiver Station houses the radio transceivers that define a cell and handles the radio-link protocols with the Mobile Station. In a large urban area, there will potentially be a large number of BTSs deployed, thus the requirements for a BTS are ruggedness, reliability, portability, and minimum cost.

The Base Station Controller manages the radio resources for one or more BTSs. It handles radio-channel setup, frequency hopping, and handovers, as described below. The BSC is the connection between the mobile station and the Mobile service Switching Center (MSC).

2.4.3 Network Subsystem

The central component of the Network Subsystem is the Mobile services Switching Center (MSC). It acts like a normal switching node of the PSTN or ISDN, and additionally provides all the functionality needed to handle a mobile subscriber, such as registration, authentication, location updating, handovers, and call routing to a roaming subscriber. These services are provided in conjunction with several functional entities, which together form the Network Subsystem. The MSC provides the connection to the fixed networks (such as the PSTN or ISDN). Signalling between functional entities in the Network Subsystem uses Signalling System Number 7 (SS7), used for trunk signalling in ISDN and widely used in current public networks.

The Home Location Register (HLR) and Visitor Location Register (VLR), together with the MSC, provide the call-routing and roaming capabilities of GSM. The HLR contains all the administrative information of each subscriber registered in the corresponding GSM network, along with the current location of the mobile. The location of the mobile is typically in the form of the signalling address of the VLR associated with the mobile station. The actual routing procedure will be described later. There is logically one HLR per GSM network, although it may be implemented as a distributed database.

The Visitor Location Register (VLR) contains selected administrative information from the HLR, necessary for call control and provision of the subscribed services, for each mobile currently located in the geographical area controlled by the VLR. Although each functional entity can be implemented as an independent unit, all manufacturers of switching equipment to date implement the VLR together with the MSC, so that the geographical area controlled by the MSC corresponds to that controlled by the VLR, thus simplifying the signalling required. Note that the MSC contains no information about particular mobile stations --- this information is stored in the location registers.

The other two registers are used for authentication and security purposes. The Equipment Identity Register (EIR) is a database that contains a list of all valid mobile equipment on the network, where each mobile station is identified by its International Mobile Equipment Identity (IMEI). An IMEI is marked as invalid if it has been reported stolen or is not type approved. The Authentication Center (AuC) is a protected database that stores a copy of the secret key stored in each subscriber's SIM card, which is used for authentication and encryption over the radio channel.

2.5 Radio Link Aspects

The International Telecommunication Union (ITU), which manages the international allocation of radio spectrum (among many other functions), allocated the bands 890-915 MHz for the uplink (mobile station to base station) and 935-960 MHz for the downlink (base station to mobile station) for mobile networks in Europe. Since this range was already being used in the early 1980s by the analog systems of the day, the CEPT had the foresight to reserve the top 10 MHz of each band for the GSM network that was still being developed. Eventually, GSM will be allocated the entire 2x25 MHz bandwidth.

2.5.1 Multiple Access and Channel Structure

Since radio spectrum is a limited resource shared by all users, a method must be devised to divide up the bandwidth among as many users as possible. The method chosen by GSM is a combination of Time- and Frequency-Division Multiple Access (TDMA/FDMA). The FDMA part involves the division by frequency of the (maximum) 25 MHz bandwidth into 124 carrier frequencies spaced 200 kHz apart. One or more carrier frequencies are assigned to each base station. Each of these carrier frequencies is then divided in time, using a TDMA scheme. The fundamental unit of time in this TDMA scheme is called a *burst period* and it lasts $15/26$ ms (or approx. 0.577 ms). Eight burst periods are grouped into a *TDMA frame* ($120/26$ ms, or approx. 4.615 ms), which forms the basic unit for the definition of logical channels. One physical channel is one burst period per TDMA frame.

Channels are defined by the number and position of their corresponding burst periods. All these definitions are cyclic, and the entire pattern repeats approximately every 3

hours. Channels can be divided into *dedicated channels*, which are allocated to a mobile station, and *common channels*, which are used by mobile stations in idle mode.

2.5.1.1 Traffic Channels

A traffic channel (TCH) is used to carry speech and data traffic. Traffic channels are defined using a 26-frame multiframe, or group of 26 TDMA frames. The length of a 26-frame multiframe is 120 ms, which is how the length of a burst period is defined (120 ms divided by 26 frames divided by 8 burst periods per frame). Out of the 26 frames, 24 are used for traffic, 1 is used for the Slow Associated Control Channel (SACCH) and 1 is currently unused (see Figure 2.2). TCHs for the uplink and downlink are separated in time by 3 burst periods, so that the mobile station does not have to transmit and receive simultaneously, thus simplifying the electronics.

In addition to these *full-rate* TCHs, there are also *half-rate* TCHs defined, although they are not yet implemented. Half-rate TCHs will effectively double the capacity of a system once half-rate speech coders are specified (i.e., speech coding at around 7 kbps, instead of 13 kbps). Eighth-rate TCHs are also specified, and are used for signalling. In the recommendations, they are called Stand-alone Dedicated Control Channels (SDCCH).

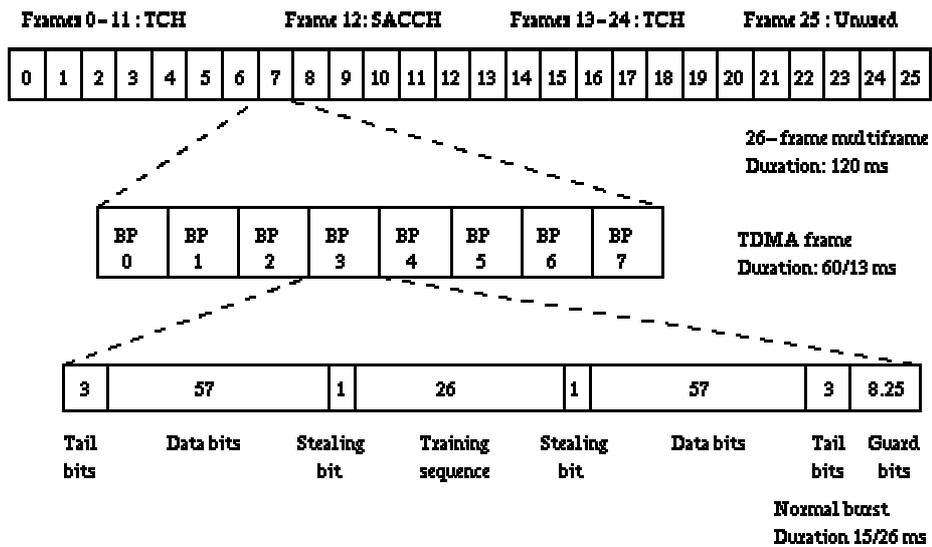


Fig. 2.2 – Organization of bursts, TDMA frames, and multiframes for speech and data

2.5.1.2 Control Channels

Common channels can be accessed both by idle mode and dedicated mode mobiles. The common channels are used by idle mode mobiles to exchange the signalling information required to change to dedicated mode. Mobiles already in dedicated mode monitor the surrounding base stations for handover and other information. The common channels are defined within a 51-frame multiframe, so that dedicated mobiles using the 26-frame multiframe TCH structure can still monitor control channels. The common channels include:

- Broadcast Control Channel (BCCH) - Continually broadcasts, on the downlink, information including base station identity, frequency allocations, and frequency-hopping sequences.
- Frequency Correction Channel (FCCH) and Synchronization Channel (SCH) - Used to synchronize the mobile to the time slot structure of a cell by defining the boundaries of burst periods, and the time slot numbering. Every cell in a

GSM network broadcasts exactly one FCCH and one SCH, which are by definition on time slot number 0 (within a TDMA frame).

- Random Access Channel (RACH) - Slotted Aloha channel used by the mobile to request access to the network.
- Paging Channel (PCH) - Used to alert the mobile station of an incoming call.
- Access Grant Channel (AGCH) - Used to allocate an SDCCH to a mobile for signalling (in order to obtain a dedicated channel), following a request on the RACH.

2.5.1.3 Burst Structure

There are four different types of bursts used for transmission in GSM [17]. The normal burst is used to carry data and most signalling. It has a total length of 156.25 bits, made up of two 57 bit information bits, a 26 bit training sequence used for equalization, 1 stealing bit for each information block (used for FACCH), 3 tail bits at each end, and an 8.25 bit guard sequence, as shown in Figure 2.2. The 156.25 bits are transmitted in 0.577 ms, giving a gross bit rate of 270.833 kbps.

The F burst, used on the FCCH, and the S burst, used on the SCH, have the same length as a normal burst, but a different internal structure, which differentiates them from normal bursts (thus allowing synchronization). The access burst is shorter than the normal burst, and is used only on the RACH.

2.5.2 Speech Coding

GSM is a digital system, so speech which is inherently analog, has to be digitized. The method employed by ISDN, and by current telephone systems for multiplexing voice lines over high speed trunks and optical fiber lines, is Pulse Coded Modulation (PCM).

The output stream from PCM is 64 kbps, too high a rate to be feasible over a radio link. The 64 kbps signal, although simple to implement, contains much redundancy. The GSM group studied several speech coding algorithms on the basis of subjective speech quality and complexity (which is related to cost, processing delay, and power consumption once implemented) before arriving at the choice of a Regular Pulse Excited -- Linear Predictive Coder (RPE--LPC) with a Long Term Predictor loop. Basically, information from previous samples, which does not change very quickly, is used to predict the current sample. The coefficients of the linear combination of the previous samples, plus an encoded form of the residual, the difference between the predicted and actual sample, represent the signal. Speech is divided into 20 millisecond samples, each of which is encoded as 260 bits, giving a total bit rate of 13 kbps. This is the so-called Full-Rate speech coding.

2.5.3 Channel Coding and Modulation

Because of natural and man-made electromagnetic interference, the encoded speech or data signal transmitted over the radio interface must be protected from errors. GSM uses convolutional encoding and block interleaving to achieve this protection. The exact algorithms used differ for speech and for different data rates. The method used for speech blocks will be described below.

Recall that the speech codec produces a 260 bit block for every 20 ms speech sample. From subjective testing, it was found that some bits of this block were more important for perceived speech quality than others. The bits are thus divided into three classes:

- **Class Ia** 50 bits - most sensitive to bit errors
- **Class Ib** 132 bits - moderately sensitive to bit errors
- **Class II** 78 bits - least sensitive to bit errors

Class Ia bits have a 3 bit Cyclic Redundancy Code added for error detection. If an error is detected, the frame is judged too damaged to be comprehensible and it is discarded. It is replaced by a slightly attenuated version of the previous correctly received frame. These 53 bits, together with the 132 Class Ib bits and a 4 bit tail sequence (a total of 189 bits), are input into a 1/2 rate convolutional encoder of constraint length 4. Each input bit is encoded as two output bits, based on a combination of the previous 4 input bits. The convolutional encoder thus outputs 378 bits, to which are added the 78 remaining Class II bits, which are unprotected. Thus every 20 ms speech sample is encoded as 456 bits, giving a bit rate of 22.8 kbps.

To further protect against the burst errors common to the radio interface, each sample is interleaved. The 456 bits output by the convolutional encoder are divided into 8 blocks of 57 bits, and these blocks are transmitted in eight consecutive time-slot bursts. Since each time-slot burst can carry two 57 bit blocks, each burst carries traffic from two different speech samples.

Recall that each time-slot burst is transmitted at a gross bit rate of 270.833 kbps. This digital signal is modulated onto the analog carrier frequency using Gaussian-filtered Minimum Shift Keying (GMSK). GMSK was selected over other modulation schemes as a compromise between spectral efficiency, complexity of the transmitter, and limited spurious emissions. The complexity of the transmitter is related to power consumption, which should be minimized for the mobile station. The spurious radio emissions, outside of the allotted bandwidth, must be strictly controlled so as to limit adjacent channel interference, and allow for the co-existence of GSM and the older analog systems (at least for the time being).

2.5.4 Multipath Equalization

At the 900 MHz range, radio waves bounce off everything - buildings, hills, cars, airplanes, etc. Thus many reflected signals, each with a different phase, can reach an antenna. Equalization is used to extract the desired signal from the unwanted reflections. It works by finding out how a known transmitted signal is modified by multipath fading, and constructing an inverse filter to extract the rest of the desired signal. This known signal is the 26-bit training sequence transmitted in the middle of every time-slot burst. The actual implementation of the equalizer is not specified in the GSM specifications.

2.5.5 Frequency Hopping

The mobile station already has to be frequency agile, meaning it can move between a transmit, receive, and monitor time slot within one TDMA frame, which normally are on different frequencies. GSM makes use of this inherent frequency agility to implement slow frequency hopping, where the mobile and BTS transmit each TDMA frame on a different carrier frequency. The frequency hopping algorithm is broadcast on the Broadcast Control Channel. Since multipath fading is dependent on carrier frequency, slow frequency hopping helps alleviate the problem. In addition, co-channel interference is in effect randomized.

2.5.6 Discontinuous Transmission

Minimizing co-channel interference is a goal in any cellular system, since it allows better service for a given cell size, or the use of smaller cells, thus increasing the overall capacity of the system. Discontinuous transmission (DTX) is a method that takes advantage of the fact that a person speaks less than 40 percent of the time in

normal conversation [20], by turning the transmitter off during silence periods. An added benefit of DTX is that power is conserved at the mobile unit.

The most important component of DTX is, of course, Voice Activity Detection. It must distinguish between voice and noise inputs, a task that is not as trivial as it appears, considering background noise. If a voice signal is misinterpreted as noise, the transmitter is turned off and a very annoying effect called clipping is heard at the receiving end. If, on the other hand, noise is misinterpreted as a voice signal too often, the efficiency of DTX is dramatically decreased. Another factor to consider is that when the transmitter is turned off, there is total silence heard at the receiving end, due to the digital nature of GSM. To assure the receiver that the connection is not dead, *comfort noise* is created at the receiving end by trying to match the characteristics of the transmitting end's background noise.

2.5.7 Discontinuous Reception

Another method used to conserve power at the mobile station is discontinuous reception. The paging channel, used by the base station to signal an incoming call, is structured into sub-channels. Each mobile station needs to listen only to its own sub-channel. In the time between successive paging sub-channels, the mobile can go into sleep mode, when almost no power is used.

2.5.8 Power Control

There are five classes of mobile stations defined, according to their peak transmitter power, rated at 20, 8, 5, 2, and 0.8 watts. To minimize co-channel interference and to conserve power, both the mobiles and the Base Transceiver Stations operate at the lowest power level that will maintain an acceptable signal quality. Power levels can be

stepped up or down in steps of 2 dB from the peak power for the class down to a minimum of 13 dBm (20 milliwatts).

The mobile station measures the signal strength or signal quality (based on the Bit Error Ratio), and passes the information to the Base Station Controller, which ultimately decides if and when the power level should be changed. Power control should be handled carefully, since there is the possibility of instability. This arises from having mobiles in co-channel cells alternately increase their power in response to increased co-channel interference caused by the other mobile increasing its power. This is unlikely to occur in practice but it is (or was as of 1991) under study.

2.6 Network Aspects

Ensuring the transmission of voice or data of a given quality over the radio link is only part of the function of a cellular mobile network. A GSM mobile can seamlessly roam nationally and internationally, which requires that registration, authentication, call routing and location updating functions exist and are standardized in GSM networks. In addition, the fact that the geographical area covered by the network is divided into cells necessitates the implementation of a handover mechanism. These functions are performed by the Network Subsystem, mainly using the Mobile Application Part (MAP) built on top of the Signalling System No. 7 protocol.

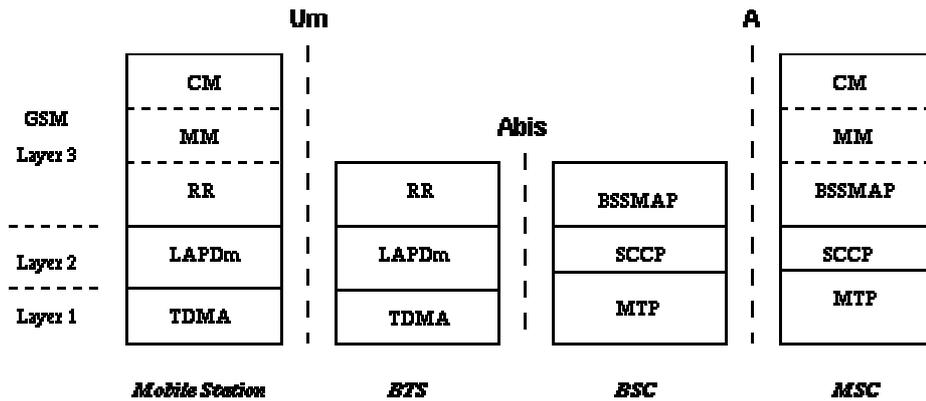


Fig. 2.3 – Signalling Protocol Structure in GSM

The signalling protocol in GSM is structured into three general layers [13], [19], depending on the interface, as shown in Figure 2.3. Layer 1 is the physical layer, which uses the channel structures discussed above over the air interface. Layer 2 is the data link layer. Across the Um interface, the data link layer is a modified version of the LAPD protocol used in ISDN, called LAPDm. Across the A interface, the Message Transfer Part layer 2 of Signalling System Number 7 is used. Layer 3 of the GSM signalling protocol is itself divided into 3 sublayers.

- Radio Resources Management - Controls the setup, maintenance, and termination of radio and fixed channels, including handovers.
- Mobility Management - Manages the location updating and registration procedures, as well as security and authentication.
- Connection Management - Handles general call control, similar to CCITT Recommendation Q.931, and manages Supplementary Services and the Short Message Service.

Signalling between the different entities in the fixed part of the network, such as between the HLR and VLR, is accomplished through the Mobile Application Part (MAP). MAP is built on top of the Transaction Capabilities Application Part (TCAP,

the top layer of Signalling System Number 7. The specification of the MAP is quite complex, and at over 500 pages, it is one of the longest documents in the GSM recommendations [17].

2.6.1 Radio Resources Management

The radio resources management (RR) layer oversees the establishment of a link, both radio and fixed, between the mobile station and the MSC. The main functional components involved are the mobile station, and the Base Station Subsystem, as well as the MSC. The RR layer is concerned with the management of an RR-session [17], which is the time that a mobile is in dedicated mode, as well as the configuration of radio channels including the allocation of dedicated channels.

An RR-session is always initiated by a mobile station through the access procedure, either for an outgoing call, or in response to a paging message. The details of the access and paging procedures, such as when a dedicated channel is actually assigned to the mobile, and the paging sub-channel structure, are handled in the RR layer. In addition, it handles the management of radio features such as power control, discontinuous transmission and reception, and timing advance.

2.6.1.1 Handover

In a cellular network, the radio and fixed links required are not permanently allocated for the duration of a call. Handover, or handoff as it is called in North America, is the switching of an on-going call to a different channel or cell. The execution and measurements required for handover form one of basic functions of the RR layer.

There are four different types of handover in the GSM system, which involve transferring a call between:

- Channels (time slots) in the same cell
- Cells (Base Transceiver Stations) under the control of the same Base Station Controller (BSC),
- Cells under the control of different BSCs, but belonging to the same Mobile services Switching Center (MSC), and
- Cells under the control of different MSCs.

The first two types of handover, called internal handovers, involve only one Base Station Controller (BSC). To save signalling bandwidth, they are managed by the BSC without involving the Mobile services Switching Center (MSC), except to notify it at the completion of the handover. The last two types of handover, called external handovers, are handled by the MSCs involved. An important aspect of GSM is that the original MSC, the *anchor MSC*, remains responsible for most call-related functions, with the exception of subsequent inter-BSC handovers under the control of the new MSC, called the *relay MSC*.

Handovers can be initiated by either the mobile or the MSC (as a means of traffic load balancing). During its idle time slots, the mobile scans the Broadcast Control Channel of up to 16 neighboring cells, and forms a list of the six best candidates for possible handover, based on the received signal strength. This information is passed to the BSC and MSC, at least once per second, and is used by the handover algorithm.

The algorithm for when a handover decision should be taken is not specified in the GSM recommendations. There are two basic algorithms used, both closely tied in with power control. This is because the BSC usually does not know whether the poor signal quality is due to multipath fading or to the mobile having moved to another cell. This is especially true in small urban cells.

The 'minimum acceptable performance' algorithm [15] gives precedence to power control over handover, so that when the signal degrades beyond a certain point, the power level of the mobile is increased. If further power increases do not improve the signal, then a handover is considered. This is the simpler and more common method, but it creates 'smeared' cell boundaries when a mobile transmitting at peak power goes some distance beyond its original cell boundaries into another cell.

The 'power budget' method [15] uses handover to try to maintain or improve a certain level of signal quality at the same or lower power level. It thus gives precedence to handover over power control. It avoids the 'smeared' cell boundary problem and reduces co-channel interference, but it is quite complicated.

2.6.2 Mobility Management

The Mobility Management layer (MM) is built on top of the RR layer, and handles the functions that arise from the mobility of the subscriber, as well as the authentication and security aspects. Location management is concerned with the procedures that enable the system to know the current location of a powered-on mobile station so that incoming call routing can be completed.

2.6.2.1 Location Updating

A powered-on mobile is informed of an incoming call by a paging message sent over the PAGCH channel of a cell. One extreme would be to page every cell in the network for each call, which is obviously a waste of radio bandwidth. The other extreme would be for the mobile to notify the system, via location updating messages, of its current location at the individual cell level. This would require paging messages to be sent to exactly one cell, but would be very wasteful due to the large number of location

updating messages. A compromise solution used in GSM is to group cells into *location areas*. Updating messages are required when moving between location areas, and mobile stations are paged in the cells of their current location area.

The location updating procedures, and subsequent call routing, use the MSC and two location registers: the Home Location Register (HLR) and the Visitor Location Register (VLR). When a mobile station is switched on in a new location area, or it moves to a new location area or different operator's PLMN, it must register with the network to indicate its current location. In the normal case, a location update message is sent to the new MSC/VLR, which records the location area information, and then sends the location information to the subscriber's HLR. The information sent to the HLR is normally the SS7 address of the new VLR, although it may be a routing number. The reason a routing number is not normally assigned, even though it would reduce signalling, is that there is only a limited number of routing numbers available in the new MSC/VLR and they are allocated on demand for incoming calls. If the subscriber is entitled to service, the HLR sends a subset of the subscriber information, needed for call control, to the new MSC/VLR, and sends a message to the old MSC/VLR to cancel the old registration.

For reliability reasons, GSM also has a periodic location updating procedure. If an HLR or MSC/VLR fails, to have each mobile register simultaneously to bring the database up to date would cause overloading. Therefore, the database is updated as location updating events occur. The enabling of periodic updating, and the time period between periodic updates, is controlled by the operator, and is a trade-off between signalling traffic and speed of recovery. If a mobile does not register after the updating time period, it is deregistered.

A procedure related to location updating is the IMSI attach and detach. A detach lets the network know that the mobile station is unreachable, and avoids having to needlessly allocate channels and send paging messages. An attach is similar to a location update, and informs the system that the mobile is reachable again. The activation of IMSI attach/detach is up to the operator on an individual cell basis.

2.6.2.2 Authentication and Security

Since the radio medium can be accessed by anyone, authentication of users to prove that they are who they claim to be, is a very important element of a mobile network. Authentication involves two functional entities, the SIM card in the mobile, and the Authentication Center (AuC). Each subscriber is given a secret key, one copy of which is stored in the SIM card and the other in the AuC. During authentication, the AuC generates a random number that it sends to the mobile. Both the mobile and the AuC then use the random number, in conjunction with the subscriber's secret key and a ciphering algorithm called A3, to generate a signed response (SRES) that is sent back to the AuC. If the number sent by the mobile is the same as the one calculated by the AuC, the subscriber is authenticated [17].

The same initial random number and subscriber key are also used to compute the ciphering key using an algorithm called A8. This ciphering key, together with the TDMA frame number, use the A5 algorithm to create a 114 bit sequence that is XORed with the 114 bits of a burst (the two 57 bit blocks). Enciphering is an option for the fairly paranoid, since the signal is already coded, interleaved, and transmitted in a TDMA manner, thus providing protection from all but the most persistent and dedicated eavesdroppers.

Another level of security is performed on the mobile equipment itself, as opposed to the mobile subscriber. As mentioned earlier, each GSM terminal is identified by a unique International Mobile Equipment Identity (IMEI) number. A list of IMEIs in the network is stored in the Equipment Identity Register (EIR). The status returned in response to an IMEI query to the EIR is one of the following:

- White-listed - The terminal is allowed to connect to the network.
- Grey-listed - The terminal is under observation from the network for possible problems.
- Black-listed - The terminal has either been reported stolen, or is not type approved (the correct type of terminal for a GSM network). The terminal is not allowed to connect to the network.

2.6.3 Communication Management

The Communication Management layer (CM) is responsible for Call Control (CC), supplementary service management, and short message service management. Each of these may be considered as a separate sublayer within the CM layer. Call control attempts to follow the ISDN procedures specified in Q.931, although routing to a roaming mobile subscriber is obviously unique to GSM. Other functions of the CC sublayer include call establishment, selection of the type of service (including alternating between services during a call), and call release.

2.6.3.1 Call Routing

Unlike routing in the fixed network, where a terminal is semi-permanently wired to a central office, a GSM user can roam nationally and even internationally. The directory number dialed to reach a mobile subscriber is called the Mobile Subscriber ISDN

(MSISDN), which is defined by the E.164 numbering plan. This number includes a country code and a National Destination Code which identifies the subscriber's operator. The first few digits of the remaining subscriber number may identify the subscriber's HLR within the home PLMN.

An incoming mobile terminating call is directed to the Gateway MSC (GMSC) function. The GMSC is basically a switch which is able to interrogate the subscriber's HLR to obtain routing information, and thus contains a table linking MSISDNs to their corresponding HLR. A simplification is to have a GMSC handle one specific PLMN. It should be noted that the GMSC function is distinct from the MSC function, but is usually implemented in an MSC.

The routing information that is returned to the GMSC is the Mobile Station Roaming Number (MSRN), which is also defined by the E.164 numbering plan. MSRNs are related to the geographical numbering plan, and not assigned to subscribers, nor are they visible to subscribers.

The most general routing procedure begins with the GMSC querying the called subscriber's HLR for an MSRN. The HLR typically stores only the SS7 address of the subscriber's current VLR, and does not have the MSRN (see the location updating section). The HLR must therefore query the subscriber's current VLR, which will temporarily allocate an MSRN from its pool for the call. This MSRN is returned to the HLR and back to the GMSC, which can then route the call to the new MSC. At the new MSC, the IMSI corresponding to the MSRN is looked up, and the mobile is paged in its current location area (see Figure 2.4).

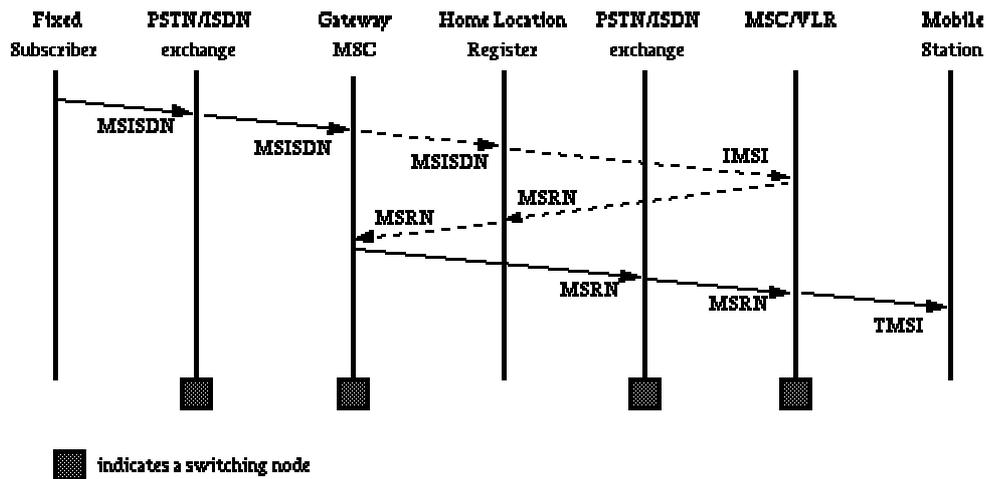


Fig. 2.4 – Call Routing for A Mobile Terminating Call

2.7 The Short Message Service

The GSM network offers to the users some base services in addition to voice communication. Among them, the most important is the Short Message Service (SMS) [11]. It allows the transfer of textual messages (with a maximum of 160 characters) or binary data (with a maximum of 140 bytes) between two mobile stations, one of which has to be a user of the GSM network.

Figure 2.5 represents the basic network architecture for an IS-41 SMSC deployment handling multiple input sources, including a voice-mail system (VMS), Web-based messaging, e-mail integration, and other external short message entities (ESMEs). Communication with the wireless network elements such as the home location register (HLR) and mobile switching center (MSC) is achieved through the signal transfer point (STP).

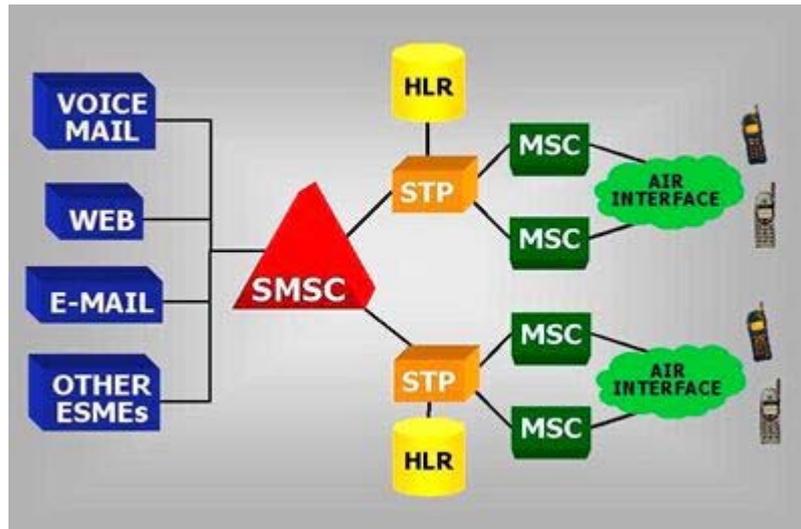


Fig. 2.5 – Basic Network Architecture for An SMS Deployment (IS-41)

SMS provides a mechanism for transmitting short messages to and from wireless devices. The service makes use of an SMSC, which acts as a store-and-forward system for short messages. The wireless network provides the mechanisms required to find the destination station(s) and transports short messages between the SMSCs and wireless stations. In contrast to other existing text-message transmission services such as alphanumeric paging, the service elements are designed to provide guaranteed delivery of text messages to the destination. Additionally, SMS supports several input mechanisms that allow interconnection with different message sources and destinations. A distinguishing characteristic of the service is that an active mobile handset is able to receive or submit a short message at any time, independent of whether a voice or data call is in progress (in some implementations, this may depend on the MSC or SMSC capabilities). SMS also guarantees delivery of the short message by the network. Temporary failures due to unavailable receiving stations are identified, and the short message is stored in the SMSC until the destination device becomes available.

SMS is characterized by out-of-band packet delivery and low-bandwidth message transfer, which results in a highly efficient means for transmitting short bursts of data. Initial applications of SMS focused on eliminating alphanumeric pagers by permitting two-way general-purpose messaging and notification services, primarily for voice mail. As technology and networks evolved, a variety of services have been introduced, including e-mail, fax, and paging integration, interactive banking, information services such as stock quotes, and integration with Internet-based applications. Wireless data applications include downloading of subscriber identity module (SIM) cards for activation, debit, profile-editing purposes, wireless points of sale (POSs), and other field-service applications such as automatic meter reading, remote sensing, and location-based services. Additionally, integration with the Internet spurred the development of Web-based messaging and other interactive applications such as instant messaging, gaming, and chatting.

2.7.1 Benefits of SMS

In today's competitive world, differentiation is a significant factor in the success of the service provider. Once the basic services, such as voice telephony, are deployed, SMS provides a powerful vehicle for service differentiation. If the market allows for it, SMS can also represent an additional source of revenue for the service provider. The benefits of SMS to subscribers center around convenience, flexibility, and seamless integration of messaging services and data access. From this perspective, the primary benefit is the ability to use the handset as an extension of the computer. SMS also eliminates the need for separate devices for messaging because services can be integrated into a single wireless device - the mobile terminal. These benefits normally

depend on the applications that the service provider offers. At a minimum, SMS benefits include the following:

- Delivery of notifications and alerts
- Guaranteed message delivery
- Reliable, low-cost communication mechanism for concise information
- Ability to screen messages and return calls in a selective way
- Increased subscriber productivity

More sophisticated functionality provides the following enhanced subscriber benefits:

- Delivery of messages to multiple subscribers at a time
- Ability to receive diverse information
- E-mail generation
- Creation of user groups
- Integration with other data and Internet-based applications

The benefits of SMS to the service provider are as follows:

- Ability to increment average revenue per user (due to increased number of calls on wireless and wireline networks by leveraging the notification capabilities of SMS)
- An alternative to alphanumeric paging services, which may replace or complement an existing paging offer
- Ability to enable wireless data access for corporate users
- New revenue streams resulting from addition of value-added services such as e-mail, voice mail, fax, and Web-based application integration, reminder service, stock and currency quotes, and airline schedules
- Provision of key administrative services such as advice of charge, over-the-air downloading, and over-the-air service provisioning

- Protection of important network resources (such as voice channels), due to SMS' sparing use of the control and traffic channels
- Notification mechanisms for newer services such as those utilizing wireless application protocol (WAP)

All of these benefits are attainable quickly, with modest incremental cost and short payback periods, which make SMS an attractive investment for service providers.

2.7.2 Network Elements and Architecture

The basic network structure of the SMS in an IS-41 network is depicted in figure 2.5.

2.7.2.1 External Short Messaging Entities

An ESME is a device that may receive or send short messages. The short message entity (SME) may be located in the fixed network, a mobile device, or another service center.

- **VMS**—The VMS is responsible for receiving, storing, and playing voice messages intended for a subscriber that was busy or not available to take a voice call. It is also responsible for sending voice-mail notifications for those subscribers to the SMSC.
- **Web**—The growth of the Internet has also affected the world of SMS. Therefore, it is almost mandatory to support interconnections to the World Wide Web for the submission of messages and notifications. The increasing number of Internet users has a positive impact on the SMS traffic increment experienced in the last few years.
- **E-Mail**—Probably the most demanded application of SMS is the ability to deliver e-mail notifications and to support two-way e-mail, using an SMS–

compliant terminal. The SMSC must support interconnection to e-mail servers acting as message input/output mechanisms.

- **Others**—There are several other mechanisms to submit short messages to the SMSC that include, but are not limited to, paging networks, specialized software for PC-based messaging and operator bureaus.

2.7.2.2 SMSC

SMSC is a combination of hardware and software responsible for the relaying and storing and forwarding of a short message between an SME and mobile device.

The SMSC must have high reliability, subscriber capacity, and message throughput. In addition, the system should be easily scalable to accommodate growing demand for SMS in the network.

Normally, an IN-based solution will allow for a lower entry cost compared to point solutions because it can support other applications on a single hardware platform and share resources, thereby spreading the deployment cost over several services and applications.

Another factor to be considered is the ease of operation and maintenance of the application, as well as the flexibility to activate new services and upgrade to new software releases.

2.7.2.3 Signal Transfer Point

The STP is a network element normally available on IN deployments that allows IS-41 interconnections over signalling system 7 (SS7) links with multiple network elements.

2.7.2.4 HLR

The HLR is a database used for permanent storage and management of subscriptions and service profiles. Upon interrogation by the SMSC, the HLR provides the routing information for the indicated subscriber. Also, if the destination station was not available when the message delivery was attempted, the HLR informs the SMSC that the station is now recognized by the mobile network to be accessible, and thus the message can be delivered.

2.7.2.5 Visitor Location Register (VLR)

The visitor location register is a database that contains temporary information about subscribers homed in one HLR who are roaming into another HLR. This information is needed by the MSC to service visiting subscribers.

2.7.2.6 MSC

The MSC performs the switching functions of the system and controls calls to and from other telephone and data systems. The MSC will deliver the short message to the specific mobile subscriber through the proper base station.

2.7.2.7 Air Interface

The air interface is defined in each one of the different wireless technologies (GSM, TDMA, and CDMA). These standards specify how the voice or data signals are transferred from the MSC to the handset and back, as well as the utilization of transmission frequencies, considering the available bandwidth and the system's capacity constraints.

2.7.2.8 The Base Station System

All functions related to the transmission of electromagnetic radio signals between the MSC and the mobile devices are performed in the base station (BS). The BS consists of base station controllers (BSCs) and the base transceiver stations (BTSs), also known as cell sites or simply “cells.” The BSC may control one or more BTSs and is in charge of the proper resource assignment when a subscriber moves from one sector of one BTS to another, regardless of whether the next sector lies within the same BTS or in a different one.

2.7.2.9 The Mobile Device

The mobile device is the wireless terminal capable of receiving and originating short messages. Commonly, these devices have been digital cellular phones, but more recently the application of SMS has been extended to other terminals such as POS, handheld computers, and personal digital assistants (PDAs). The wireless network-signalling infrastructure is based on SS7. SMS makes use of the Mobile Application Part (MAP), which defines the methods and mechanisms of communication in wireless networks and employs the services of the SS7 transactional capabilities application part (TCAP). An SMS service layer makes use of the MAP signalling capabilities and enables the transfer of short messages between the peer entities.

The capabilities of the terminal vary depending on the wireless technology supported by the terminal. Some functionality, although defined in the SMS specification for a given wireless technology, may not be fully supported in the terminal, which may represent a limitation in the services that the carrier can provide. This trend, however, is disappearing as service providers’ merger and acquisition activity demands uniform functionality across all the constituents of the parent companies. Also, some

manufacturers may include additional functionality, not considered in the specification, attempting to offer a more attractive product for service providers as well as end users. This will be the case more often as service providers continue to incorporate SMS into their revenue-generating and customer-loyalty strategies.

2.7.3 Signalling Elements

The MAP layer defines the operations necessary to support SMS. Both American and international standards bodies have defined a MAP layer using the services of the SS7 TCAP. The American standard is published by Telecommunication Industry Association and is referred to as IS-41. The international standard is defined by the European Telecommunications Standards Institute (ETSI) and is referred to as GSM MAP.

The following basic MAP operations are necessary to provide the end-to-end SMS:

- **Routing Information Request**—Before attempting delivery of a short message, the SMSC must receive routing information to determine the serving MSC for the mobile device at the time of the delivery attempt. This is accomplished by way of an interrogation of the destination handset's HLR, which is accomplished via the use of the *SMSrequest* and *SendRoutingInfoForShortMsg* mechanisms in IS-41 and GSM, respectively.
- **Point-to-Point Short Message Delivery**—The mechanism provides a means for the SMSC to transfer a short message to the MSC that is serving the addressed mobile device. After the address of said MSC has been obtained from the station's HLR, the short message delivery operation provides a confirmed delivery service. The operation works in conjunction with the base station subsystem while the message is being forwarded from the MSC to the

MS. Therefore, the outcome of the operation comprises either success (such as delivery to the mobile) or failure caused by one of several possible reasons. The point-to-point short message delivery is accomplished via the use of the short message delivery–point-to-point (SMD–PP) and *forwardShortMessage* mechanisms in IS-41 and GSM, respectively.

- **Short Message Waiting Indication**—The operation is activated when a short message delivery attempt by the SMSC fails due to a temporary failure, such as the station being unregistered, and provides a means for the SMSC to request the HLR to notify the SMSC when the indicated mobile device becomes available. This short message waiting indication is realized via the use of the *SMS_notification* indicator and *set_message_waiting_data* mechanisms in IS-41 and GSM, respectively.
- **Service Center Alert**—The operation provides a means for the HLR to inform the SMSC, which has requested a notification that a specific mobile device is now recognized by the mobile network to be available. This service center alert is accomplished via the use of the *SMS_notification* and *alert_service_center* mechanisms in IS-41 and GSM, respectively.

2.7.3.1 Service Elements

SMS is comprised of several service elements relevant to the reception and submission of short messages:

- **Message Expiration**—The SMSC will store and reattempt delivery of messages for unavailable recipients until either the delivery is successful or the expiration time—set on a per-message basis or on a platform-wide basis—arrives.

- **Priority**—This is the information element provided by an SME to indicate the urgent messages and differentiate them from the normal priority messages. Urgent messages usually take priority over normal messages, regardless of the time of arrival to the SMSC platform.
- **Message Escalation**—The SMSC stores the message for a period no longer than the expiration time (it is assumed that the escalation time is smaller than the expiration time associated with the message), and after said escalation time expires, the message will be sent to an alternate message system (such as a paging network or an e-mail server) for delivery to the user.

In addition, SMS provides a time stamp reporting the time of submission of the message to the SMSC and an indication to the handset of whether or not there are more messages to send (GSM) or the number of additional messages to send (IS-41).

2.7.4 Mobile-Terminated Short Message Example

Figure 2.6 depicts the successful MT—SM scenario for GSM.

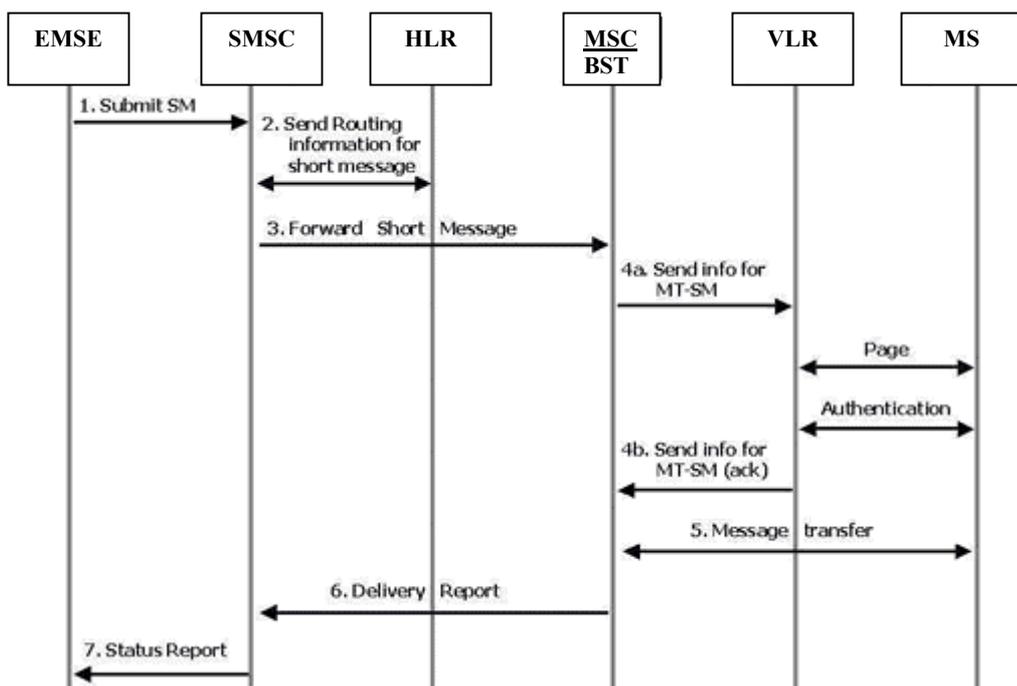


Fig. 2.6 – MT – SM Scenario (GSM)

1. The short message is submitted from the ESME to the SMSC.
2. After completing its internal processing, the SMSC interrogates the HLR and receives the routing information for the mobile subscriber.
3. The SMSC sends the short message to the MSC using the forward short message operation.
4. The MSC retrieves the subscriber information from the VLR. This operation may include an authentication procedure.
5. The MSC transfers the short message to the MS.
6. The MSC returns to the SMSC the outcome of the forwardShortMessage operation.
7. If requested by the ESME, the SMSC returns a status report indicating delivery of the short message.

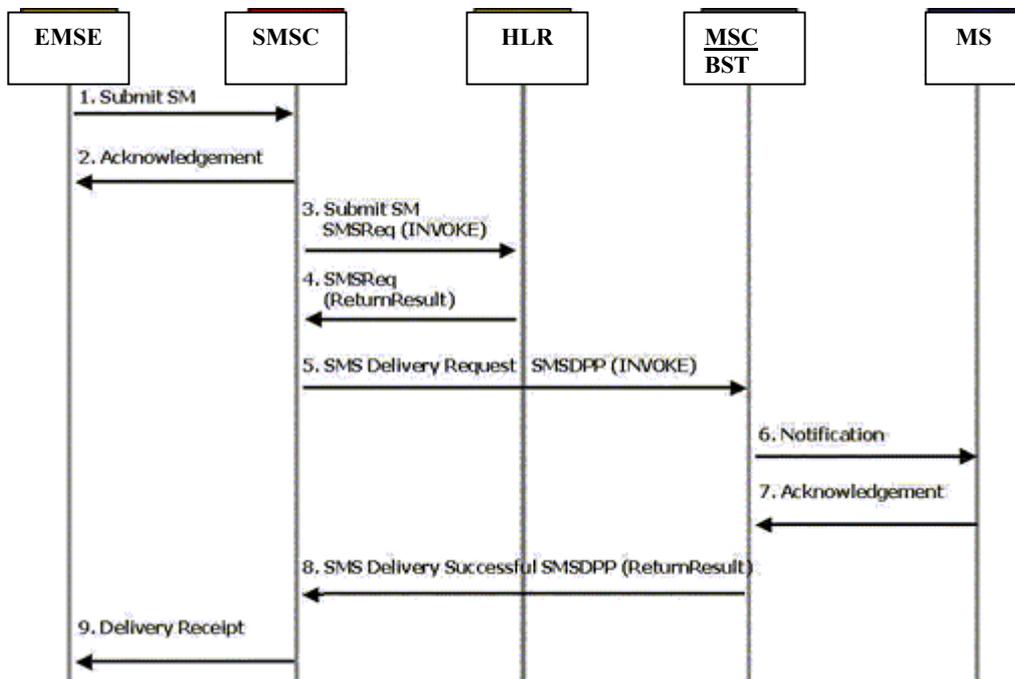


Fig. 2.7 – MT Short Message Scenario (IS-41)

1. The short message is submitted from the ESME to the SMSC.

2. The SMSC sends an acknowledgement to the ESME, indicating reception of the short message.
3. After completing its internal processing, the SMSC interrogates the HLR.
4. The HLR sends the routing information for the mobile subscriber to the SMSC.
5. The SMSC sends the short message to the MSC using the SMSDPP Invoke operation.
6. The MSC transfers the short message to the MS.
7. The MS returns an acknowledgement to the MSC.
8. The MSC returns to the SMSC the outcome of the SMSDPP operation.
9. If requested by the ESME, the SMSC returns a delivery receipt indicating successful delivery of the short message.

2.7.5 Mobile-Originated Short Message Example

Figure 2.8 depicts the successful MO–SM scenario, utilizing the GSM method. The IS-41 method for the MO-SM scenario is depicted in figure 2.9.

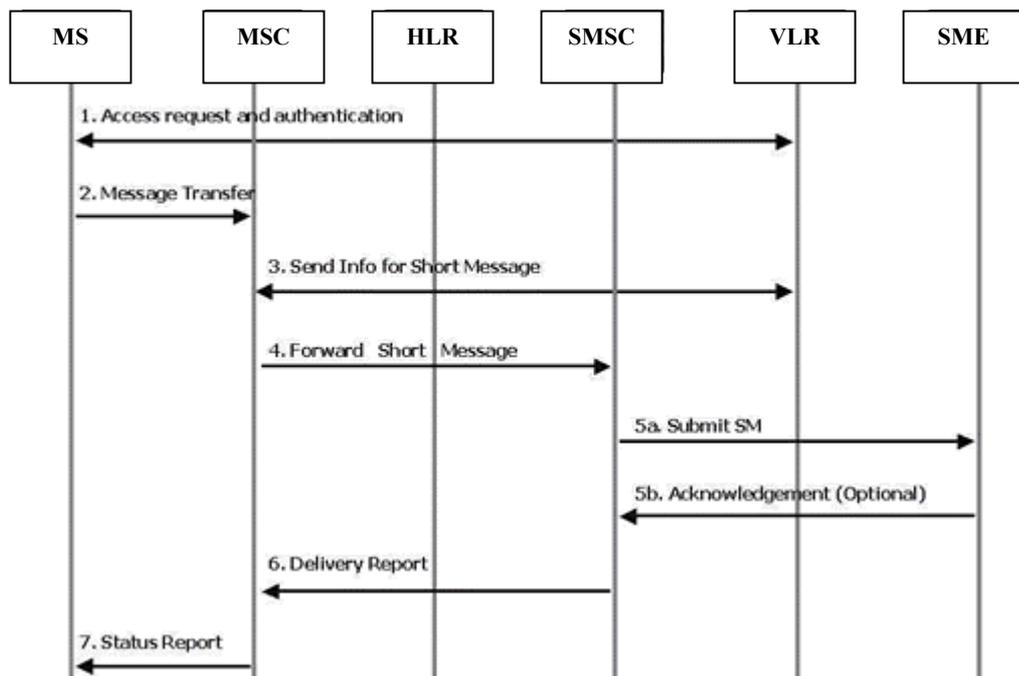


Fig. 2.8 – MO—SM Scenario (GSM)

1. The MS is powered on and registered with the network.
2. The MS transfers the SM to the MSC.
3. The MSC interrogates the VLR to verify that the message transfer does not violate the supplementary services invoked or the restrictions imposed.
4. The MSC sends the short message to the SMSC using the forwardShortMessage operation.
5. The SMSC delivers the short message to the SME (and optionally receives acknowledgment).
6. The SMSC acknowledges to the MSC the successful outcome of the forwardShortMessage operation.
7. The MSC returns to the MS the outcome of the MO-SM operation.

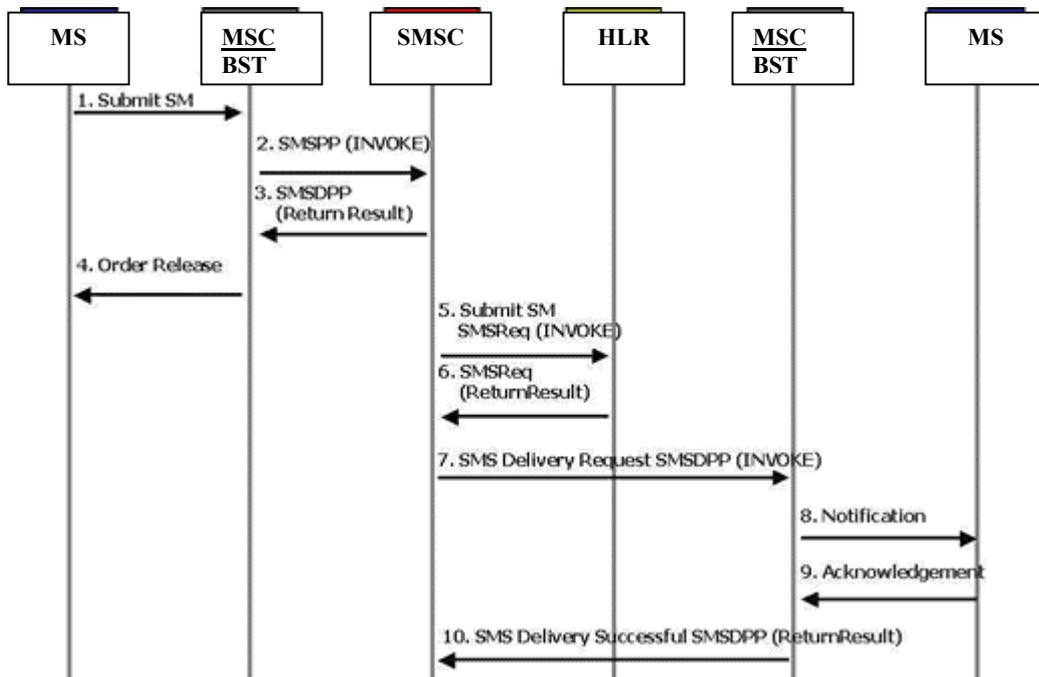


Fig. 2.9 – MO—SM Scenario (IS-41)

1. The MS transfers the SM to the MSC.
2. The MSC interrogates the home SMSC to verify that the message transfer does not violate the supplementary services invoked or the restrictions imposed. The

MSC sends the short message to the home SMSC using the SMSPP Invoke operation.

3. The SMSC delivers an acknowledgment to the MSC.
4. The MSC returns order release to the MS.
5. The SMSC queries the HLR for the location of the destination MS.
6. The HLR returns the destination (MSC) serving the destination MS.
7. The SMSC delivers SM to the MSC serving the destination MS.
8. The SMSC delivers the short message to the MS.
9. The MS acknowledges to the MSC the successful outcome of the SMSDPP operation.
10. The MSC returns to the SMSC the outcome of the MO–SM operation (delivery successful).

2.7.6 SMS Applications

SMS was initially designed to support limited-size messages, mostly notifications and numeric or alphanumeric pages. While these applications are and will continue to be widely used, there are more recent niches that SMS still can exploit.

Short bursts of data are at the heart of many applications that were restricted to the world of data networks with fixed terminals attached to a local-area network (LAN) or wide-area network (WAN). However, many of these applications are better served if the data communication capabilities could be added to the mobility of the station. Thus, a waiter who can charge a customer's credit card right at the table, at any time, instead of going to a fixed POS terminal located by the register will be able to help customers in a faster, more convenient way.

Also, the ability to track the location of a moving asset such as a truck or its load is very valuable for both providers and clients. This application, again, just needs to interchange small amounts of information, such as the longitude and latitude at a current time of the day, and perhaps other parameters like temperature or humidity.

This application does not necessarily require the monitored entity to be in movement. The requirements are basically short burst data and a location that has digital network coverage. For example, in a neighborhood, it would be faster, easier, and cheaper to drive a truck from the local power company, which interrogates intelligent meters to obtain their current readings and then forwards them via short message to a central data processing center to generate the billing. Similarly, delivery trucks could be alerted of the inventory of a customer running low, when the truck is close to the customer's facilities. The truck driver could place a quick phone call to the customer to offer a short-time replenishment at a low cost for the distributor.

Another family of applications that can use SMS as a data transport mechanism is banking. It is no secret that automated teller machine (ATM) and Internet transactions are less costly than transactions completed at a branch. Internet transactions are even cheaper than ATM transactions. Therefore, enabling wireless subscribers to check their balances, transfer funds between accounts, pay their bills and credit cards is valuable, not only for the subscriber but also for financial institutions.

Entertainment applications are also good drivers of SMS usage. Examples of these are simple short message exchanges between two parties ("texting") or between multiple participants ("chat"). Also, delivery of information that the subscriber can tailor to his or her lifestyle represents an attractive proposition for wireless users.

Wireless Web browsing allows the users to search for information without the physical restrictions of a PC. College students certainly appreciate not having to go to the

computer lab to check e-mail or find out what the required book is for the semester that is about to start.

E-mail continues to be by far the most used wireless data application. However, handsets are evolving quickly and are including more and more functionality that supports newer applications at the same time that user friendliness increases. Probably the next big success beyond wireless Web will be Internet shopping and other e-commerce applications such as electronic coupons, advertising, etc.

The potential for applications is enormous, and new needs appear to arise constantly, demanding a solution that may travel over SMS.

3 GENERAL PROTOTYPE SYSTEM AND HARDWARE COMPONENTS

This chapter explains the software used and the general prototype system model undertaken in the project. It includes software overview and the software programming modules of the prototype system. It also touches on the hardware used in the project. It includes the hardware overview and the hardware specifications of all the prototype systems.

3.1 Software Requirements

The programming software used in the prototype is Visual Basic, which is installed on the PC. It is user-friendly to be programmed with Microsoft Access database. Additionally, the references that are needed to send and receive SMS messages could be embedded with Visual Basic. The references are Nokia PC Connectivity SDK 3.0 [5], which provided communication between the Nokia 8210 GSM phone and the prototype program. It helps the prototype program to manage SMS messages with the GSM phone. Outgoing text messages from the prototype are sent to the GSM phone for transmission as SMS and incoming SMS messages from the GSM phone are communicated to the prototype as text message.

All the prototypes in the project require large amount of information to be stored in the PC. Microsoft Access database is chosen to store the information because it is user-friendly and Visual Basic can link the solutions developed to store, read and update the information from Microsoft Access database.

3.2 Software Overview

Figure 3.1 illustrates the general system software structure developed to run the SMS applications.

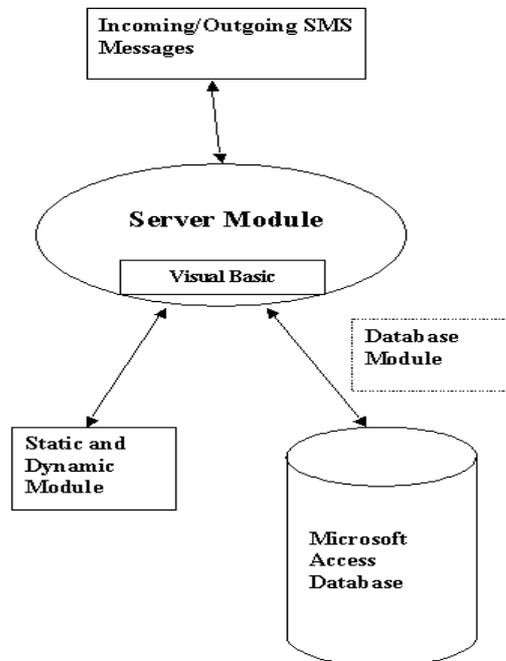


Fig. 3.1 – General System Software Structure

3.2.1 Software Modules

The programming work of all the prototypes' software is categorized into four modules: server, static, dynamic, and database. Each of them has its own task to complete when called for and they are interconnected.

3.2.1.1 Server Module

The server module is developed in the prototypes such that it is always active when it is running on the PC. It is responsible for all message assembly, and delivery for transmission via the GSM data cable and GSM phone. Internal server information is kept up to date with any real time input as the database on disk is scanned at regular intervals for any new information entered by the system's operator. Depending on how

an individual prototype works, the server module interacts with other modules in order to process any scheduled or unscheduled requests that were made by users of the service.

3.2.1.2 Static Module

The static module is responsible for handling SMS requests for information that is stored in the database. The static module is needed when the server module receives and identifies a SMS message as request for information. The server module then sends the message as text to the static module. The static module organizes the message and takes out the key words. From the key words, the static module interacts with the database module whereby it extracts out the relevant information from the database according to the key words and sent it back to the static module. The static module arranges the results in a SMS message form and passes it to the server module for transmission.

3.2.1.3 Dynamic Module

The dynamic module is responsible of handling SMS requests that require information manipulations in the database. The dynamic module is needed when the server module receives and identifies the SMS message as a request for information changes such as new data entry. It then sends the message as text to the dynamic module. The dynamic module extracts the key words and organizes them according to different database fields. The dynamic module then passes these records to the database module whereby it enters them into the database. The dynamic module can also handle scheduled SMS requests whereby users wanted information to be received at certain specific time instances and/or intervals. This is done by making the dynamic module active at all

times and interact with the database module to see if any time fields in the database had matched the clock of the PC. When a match occurs, the dynamic module extracts out the required information from the database module and sends it to the server module for transmission.

3.2.1.4 Database Module

The database module is essential in all prototypes designed, as it takes charge of the database management. The dynamic and static modules always interact with the database module for information queries, updates or changes. The difficulties of designing the algorithm in the database module depend on the number of database fields and how closely the fields are related.

When the static module passes the key words to the database module, it makes use of the key words' request and searches for records in the database fields that match the key words request. Depending on the nature of prototypes designed, the database module might need to search for records differently. In some of prototypes it is easier to search for information by the database module, as records in the database fields are different from one another. In certain other prototypes whose records are quite similar, the database module might need a more sophisticated algorithm to identify and extract the correct information from the database.

During database management, the database module is designed to make sure records received from the dynamic module are entered into the correct fields of the database. Prototypes with a lot of database fields might need a more powerful algorithm to run the database module. One additional feature designed in the database module is timer. This is to facilitate scheduled SMS request whereby users might want information to be sent to them at certain instances and/or periods of time. This is achieved by adding a

time field in the database. For example, user might send a SMS request to the prototype to alert them of certain information at a certain time. The dynamic module interacts with the database module to record down the time and the user's information in the database. When the time is reached, the dynamic module is supposed to discover it and inform the database module to extract and send back the required information from the database. After organizing the information into proper SMS message, the dynamic module then passes it to the server module for transmission.

3.3 General Prototype System

Figure 3.2 in the general prototype system model shows how mobile terminals are linked to the central computer system through the GSM network.

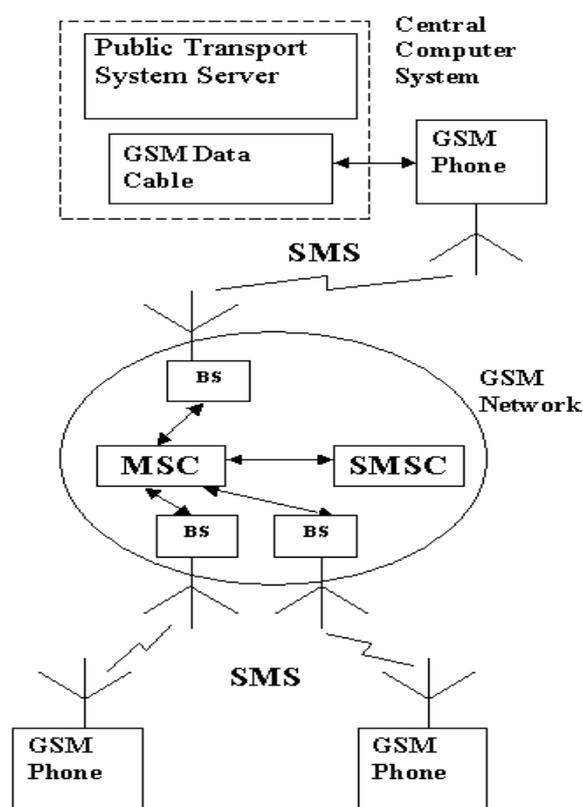


Fig. 3.2 – General Prototype System Model

The public transport system server acts as the source of all related information, which the central computer system utilizes. There is clearly no restriction on where this server is located and will most likely not be located anywhere geographically close to the service platform. It maintains databases which, depending on prototypes, includes information such as road names, bus services, car parks, etc. The system server is able to keep an up-to-date record of the current parameters (e.g. arrival times, current fines, current car-park vacancy, etc) in databases of interest required by users.

Firstly, a user has to send SMS request across the GSM network to the GSM phone number that is attached by mean of a GSM data cable to the central computer system. The GSM phone takes in information in the SMS message and the user's GSM phone number to the central computer system. The system server searches through the database for best-effort results and sends it back through the GSM phone to the user.

3.4 Hardware Equipment

- 1) PC with CPU processor speed Pentium II 400MHz – the prototype was able to search through the database and send and receive SMS messages in a few milliseconds.
- 2) GSM data cable compatible with Nokia brand GSM phone – it was needed to link SMS data to be sent or received to and fro the PC and the phone.
- 3) Nokia 8210 GSM phone – it was used to send and receive SMS messages across the GSM network with other mobile phones.

3.5 Hardware Overview

The prototype systems were designed around a single desktop Pentium PC, which had hardware equipped to support the variety of communication links required for the

system's operation. In particular the basic PC system was equipped with:

- A GSM data cable which provided access between the GSM phone (thus providing the GSM SMS hardware functionality) and the PC,
- A GSM phone connected to the PC, that took care of sending and receiving SMS messages over the GSM network.

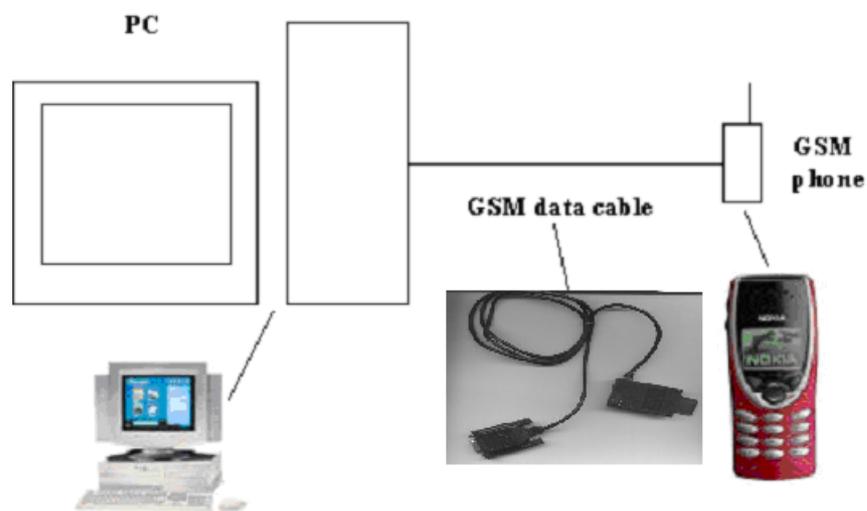


Fig. 3.3 – System Hardware Structure

The three hardware components were essential in developing all the prototypes. In each prototype, a software program was developed for running on the PC. Database was also created to link with the program for some prototypes. Additional codes were written for the software program to communicate with the GSM phone, and the GSM data cable was needed to link the communication between the program and the GSM phone.

3.5.1 Hardware Setup

Step 1 - Install Nokia PC Connectivity SDK 3.0.



Fig. 3.4 – Installation of Nokia PC Connectivity SDK

The Nokia PC Connectivity SDK 3.0 can be downloaded from the Nokia website <http://www.forum.nokia.com/main.html> under tools & SDKs. After installation, activate the Nokia connection manager to check the functionality of the SDK. Currently, no GSM phone is detected as seen in figure 3.4.

Step 2 - Connect the Nokia GSM phone to the GSM data cable.

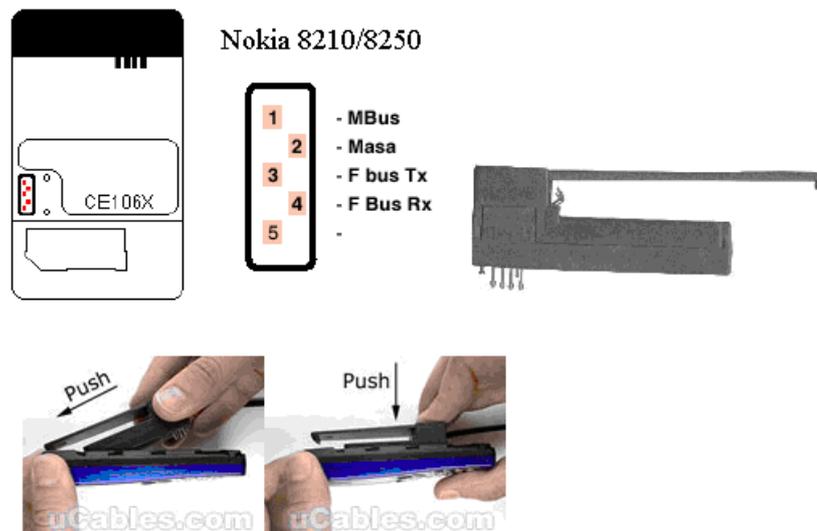


Fig. 3.5 – Data Cable Connection to Nokia Phone

The Nokia 8210 does not have a visible data port, so the data cable attaches in a special way to the back of the handset as a bridge between the phone and the battery. Firstly, the back cover and battery has to be removed from the phone and the battery is placed into the battery holder on the connector of the data cable. The connector is then inserted into the battery holder on the phone. The connector should fit securely and clip into place.

Step 3 - Connect the GSM data cable to PC serial port.



Fig. 3.6 – Data Cable Connection to PC serial port

The other end of the cable is connected to the 9 pin serial port of the PC.

Step 4 - Check to see if the SDK detect the Nokia GSM phone.



Fig. 3.7 – Nokia Connection Manager Detection of Nokia 8210

After connection is completed, turn the phone on and run the required phone software to check that all the software settings are correct. The connection manager should detect the phone as shown in figure 3.7.

3.5.2 Trouble Shooting

1. Repeat the Data Cable Set-Up instructions, but try turning your phone on after the software is running.
2. Serial port settings must not been in use by another device (eg. internal modem). You can check this through the Device Manager that can be found in Windows Control Panel.
3. There must be no IRQ/DMA conflicts on the serial port. You can check this through the Device Manager that can be found in Windows Control Panel.
4. The correct serial port must be selected in the software. Try different serial ports if you are unsure what serial port to use.
5. The cable must be configured correctly in the software settings. (eg. MBus/FBus)
6. All connections must be secure to get a reliable connection.
7. If software can work on both M and F bus, then cable settings must match software settings.
8. The software you are using must be compatible with your computers operating system (eg. Windows 98) and the software must support your cable/phone.
9. The software you are using must be installed correctly, and you should always reset your computer after installing new software.

10. You should only have one program that uses the serial port running at once.

Having two or more programs trying to access the serial port at once will cause port conflicts and errors.

11. Check the both the battery and sim card are securely inserted into the phone.

4 SMS APPLICATIONS

This chapter gives a detailed report of how SMS based information systems were implemented into the public transport system in Singapore. Each system's present development is introduced first and each prototype's application is presented with a more detailed functional flow block diagram. Following this, each prototype development is discussed, which is categorized into a few sections. And lastly, a comparison between applications is reviewed.

4.1 Public Service Transport System

There are two major bus services companies – Singapore Bus Services Transit (SBS Transit) [1] and Trans-Island Bus Services (TIBS) [3] and one mass rapid transit company – Singapore Mass Rapid Transit (SMRT) [2] in Singapore. The three companies set up TransitLink [4] to develop an integrated public transport system by bringing buses, MRT and LRT trains together as one comprehensive network. Currently, TransitLink provides fare, information, and network integration. For the information integration, which is the main issue in this project, TransitLink provides information on almost all aspects of traveling on buses, MRT, and LRT trains in the form of a booklet called TransitLink Guide. Additionally, TransitLink puts up comprehensive information panels at MRT stations and major bus stops for the ease of commuters transferring services. Commuters are required to transfer services when they are going from one service route to another. The service routes will either serve the same stop, or stops a short walk apart.

SBS Transit and SingTel [10] have recently developed their first SMS bus guide whereby the information primarily consists of SBS Transit service. SingTel is a

leading integrated communications service provider in the Asia Pacific. It has a well-established and extensive communications network and infrastructure in Singapore and Australia. Its advanced mobile networks cover 100% and 94% of the Singapore and Australian populations respectively. In the SMS bus guide, the cell ID system is able to automatically determine the commuter's location using SingTel GSM base stations around the country and informs the commuter of the available bus services in the vicinity. By keying in the desired destination, a point-to-point bus travel solution is provided to the commuter. Commuters can also request for the route of any SBS Transit bus service via this SMS service.

4.1.1 SMS Implementation and its Objectives

For the prototype system discussed in this project, it will further provide not only bus service information for commuters, but as well as MRT information and it will integrate the two services based on a near far basis. That is, the system is able to formulate a point-to-point travel solution that suggest taking a MRT train if the desired destination is far from the current location of the commuter, but it is near a MRT station.

Real-time information will also be available, such as bus and MRT arrival times. In this way, commuters will be able to plan their journeys such that they can:

- Time their arrivals at bus stops, MRT stations;
- Plan for links with other modes of transport;
- Make an informed choice on the best mode of transport;
- Be aware of bus delays, detours and alternative routes; and
- Determine alternative travel plans.

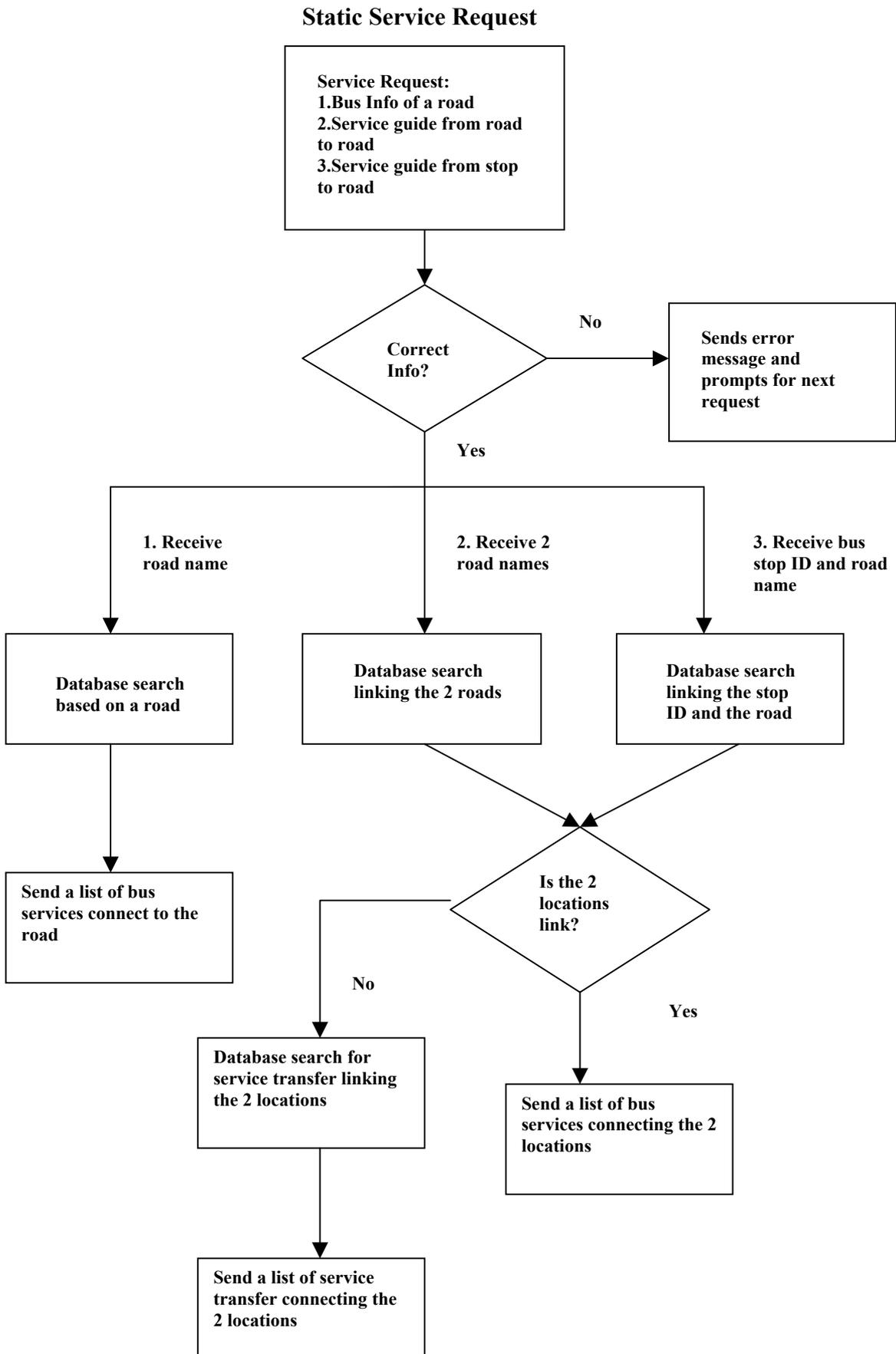
In order for real-time information delivery to be achievable, certain tracking system has to be deployed in every bus and MRT train. For MRT trains, they can be monitored electronically by the MRT system, thus real-time information is possible. As for buses, Automatic Vehicle Location (AVL) is used to track them and acquire their arrival times. Currently, SBS Transit is still testing the system which uses Global Positioning System to locate buses' positions.

The system will also develop a communication platform between bus drivers and commuters; so that commuters can flag a bus via SMS before reaching their bus stops and this information will be related to the bus driver. Additionally, bus drivers can also communicate with commuters about delays, congestions, accidents, etc via SMS.

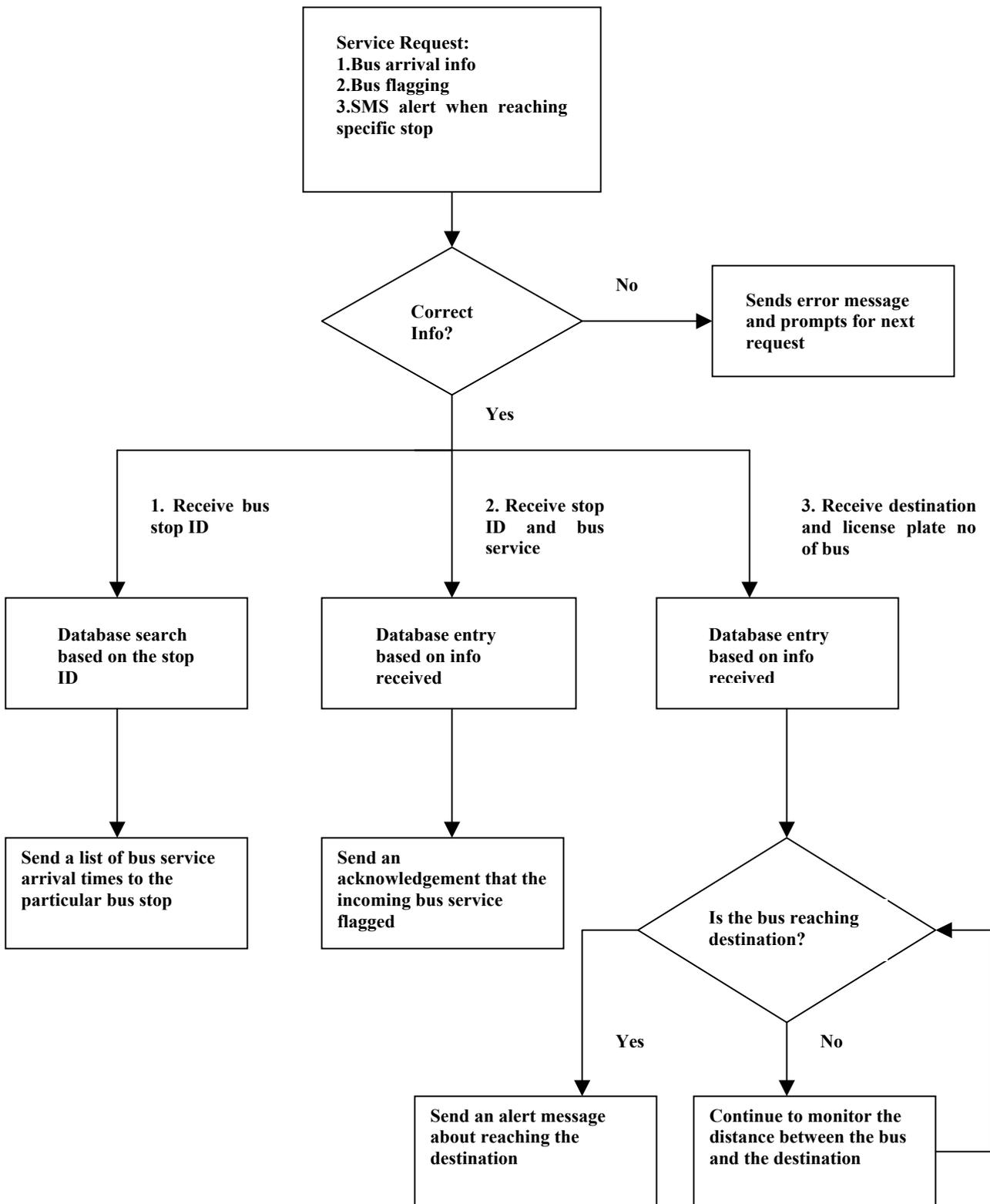
Commuters can also make use of the system to receive alerts when they are reaching their destinations. They will have to send a SMS message to the system about their destinations, and through real-time monitoring of bus movements, the system will be able to send back an SMS alert message to inform commuters that they are reaching their destinations.

In this project, a working prototype is designed to realise all possible factors of the system.

4.1.2 Functional Flow Block Diagram



Dynamic Service Request



4.1.3 Prototype System Overview

Based on the general prototype system model in figure 3.2, the system server acts as the source of all related bus and MRT information, which the central computer system utilizes. It maintains databases of bus and MRT routes. Necessary graphic user interfaces are designed for easy entry of new services and changes of service routes for the system operator. It is also responsible for answering the SMS requests by commuters, bus drivers and system operators.

The system server was designed to provide features such as travel guide, arrivals query, bus flagging and stop reaching alert. Firstly, a commuter has to send SMS request across the GSM network to the GSM phone number that is attached by mean of a GSM data cable to the central computer system. The GSM phone takes in information in the SMS message and the commuter's GSM phone number to the central computer system. The system server searches through the database for best-effort results and sends it back through the GSM phone to the commuter. For the system operator, he/she may send SMS to the system about urgent changes to database, whether route changes or schedule changes, when he/she is not near the system. The system recognizes the operator's GSM phone number and makes necessary changes to the database.

The hardware development for the prototype and its features are mentioned in Chapter 3. When it needs AVL to track the vehicles, the actual location values are replaced by simulation values.

4.1.4 Prototype Software Developments and Testing

Based on the general system software structure of figure 3.1, the public service transport system is broken down into several modules.

4.1.4.1 Public Service Transport Database Module

Database of bus and MRT services is stored in the PC. For the prototype system a few bus services were chosen to be included in the Microsoft Access database. In Singapore, there are currently two types of route services for bus; one is loop type, whereby the service has one route that starts from the bus interchange and returns back to the same interchange. The other is two-way type, whereby the service has two routes that start from one interchange and end at another interchange and vice versa. Some of the bus services that are linked are therefore appropriately recorded into the database. A database graphical user interface is created that takes the form of a series of windows, which contains the options necessary for administering the database. The database contains all necessary information for the bus and MRT services such as, service number, station name, road name, fare stage, nearness to MRT station, and so on.

Figure 4.1 illustrates one particular level in the GUI, which is used in the processes for adding new service information to the database and updating the existing service information. Buttons and drop down lists enable simple adding, removal and editing of service data, while text boxes allow fast searching of the service entries by bus stop ID, road name, service number, etc. Database entries are directly edited by finding the record of interest and typing in the new data. Service information can also be removed from the database in a keystroke.

ID	road_name	description	arrival	fare
1	Eng Kong Rd	Eng Kong Gdn	7:00	\$5.00
2	Eng Kong Rd	Eng Kong Gdn	7:10	\$5.00
3	Eng Kong Rd	Eng Kong Gdn	7:30	\$5.00
4	Eng Kong Rd	Eng Kong Gdn	7:50	\$5.00
5	Eng Kong Rd	Eng Kong Gdn	8:10	\$5.00
6	Eng Kong Rd	Eng Kong Gdn	8:20	\$5.00
7	Eng Kong Rd	Eng Kong Gdn	18:20	\$4.50
8	Eng Kong Rd	Eng Kong Gdn	18:55	\$4.50
9	Eng Kong Rd	Eng Kong Gdn	19:10	\$4.50
10	Eng Kong Rd	Eng Kong Gdn	19:35	\$4.50
11	Eng Kong Rd	Eng Kong Gdn	20:00	\$4.50
12	Toh Tuck Rd	Goodluck Gdn	7:10	\$5.00
13	Toh Tuck Rd	Goodluck Gdn	7:20	\$5.00
14	Toh Tuck Rd	Goodluck Gdn	7:40	\$5.00
15	Toh Tuck Rd	Goodluck Gdn	8:00	\$5.00
16	Toh Tuck Rd	Goodluck Gdn	8:20	\$5.00
17	Toh Tuck Rd	Goodluck Gdn	8:30	\$5.00
18	Dunearn Rd	Sith Ave Ctr	7:20	\$4.50

Fig. 4.1 – Service Database GUI for Public Service Transport System

4.1.4.2 Public Service Transport Static Module and Server Module

The static module, together with the server module is responsible for fixed data transmission via the GSM data cable and GSM phone. The process involves queries of fixed bus and MRT route and schedule information. The server module is ready for any SMS messages received from the GSM phone. The static module takes in the SMS message in text form and the commuter’s mobile phone number. Depending on what the message queries about, the static module looks into the service database and formulates the best answer for the server module to transmit it back to the commuter’s mobile phone. The static module responds to several types of query:

- Commuters send in a street/road name and the prototype responds a list of bus and MRT services that are located at the street/road. If the information of the street/road results in more than one location, such as the name Clementi which can mean Clementi road, Clementi ave 3, Clementi ave 6, etc, the prototype responds with the list of services that are located at all these locations. The prototype will reply with an error message if the commuter sends unclear queries like road, ave, A, etc.

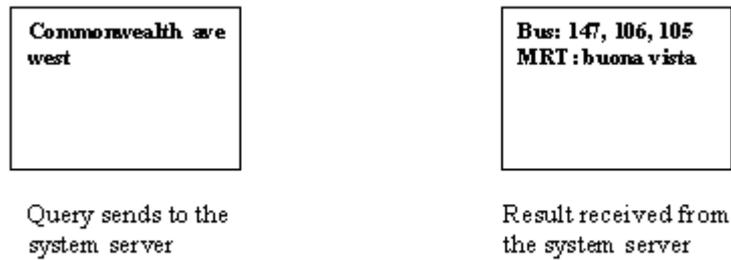


Fig. 4.2 – Service Guide Search Through Road Name

- Commuters send in two street/road names, current location and destination, and the prototype responds with services, bus or MRT train that connect the two streets/roads. If the information of both streets/roads results in more than one location, such as the name Clementi which can mean Clementi road, Clementi ave 3, Clementi ave 6, etc and the name Jurong which can mean Jurong road, Jurong east, etc, the prototype searches and responds the list of bus services that link both locations. The prototype will reply with an error message if the commuter sends unclear queries like road, ave, A, etc. The service guide is made user-friendly as a commas or just a space can separate the two locations when the commuter sends the message. The prototype is designed to identify the two locations.

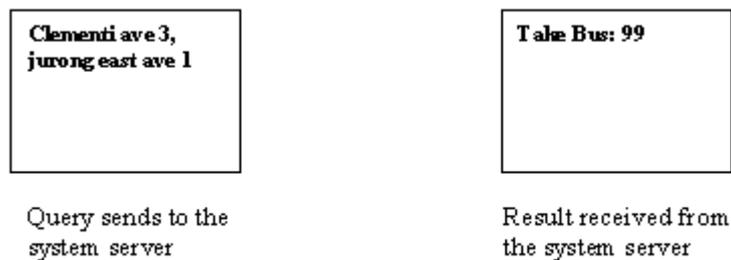


Fig. 4.3 – Road-to-Road Service Guide

- Commuters send in a bus stop ID and a road name, and the prototype responds with services that linked between the bus stop and the road. The service guide is made user-friendly as a commas or just a space can separate the two

locations when the commuter sends the message. The prototype is designed to identify the two locations.



Fig. 4.4 – Stop-to-Road Service Guide with Bus Service Transfer

When two locations are not connected by a service, the static module searches through the database and looks out for possible service transfer. The process is completed in a few stages. When the module discovers that two locations are not connected by a service, it records down all bus services for the two locations. Then, it looks through the database of road names and checks whether any road had bus services from the two locations. The module might find several road names and it would record down the earliest service transfer. All information is sent back to the commuter's mobile phone, informing the commuter which bus service to take, and which stop at the road to get off for service transfer to the destination.

If the two locations is quite far and traveling by bus might take a long time, the module is able to search for possible transfers by MRT train. In such cases the module employed the near far basis, by which the process could be broken into two levels; first, it analyzes how far apart are the two locations. For the prototype, every road is categorized into area IDs according to the road map of Singapore [12]. Roads belonging to the east side of the map are split and given area ID e1, e2, e3, etc and roads belonging to the west side of the map are also split and given area ID w1, w2, w3, etc and so on. This is to simulate SingTel's method of using base stations to locate commuters. For SingTel's method, the bus services located in a base station are

recorded in the database. For the prototype, it not just only includes bus services that are located in an area ID but also MRT services. If the two locations are located in the same area ID or in area IDs close to each other, the module considers the two locations as near to each other, and informs commuters to take bus service transfer. If the two locations are located in area IDs that were quite far apart, the module analyzes whether there are MRT stations located in the two area IDs. If there are, the module informs commuters to take service transfer by MRT trains. If there are no MRT stations located in the two area IDs, the process goes on to the next level.

In the next level, the module records down the bus services in the current location and searches through the database of every bus service to see whether they pass by any MRT station along the way. If a bus service happens to pass by MRT stations that are near to the two locations, the module explores whether it is worth making a transfer by MRT trains. This is done by analyzing whether the two stations are located in area IDs that are quite far apart. All these processes take a few milliseconds due to the fast processor of the PC, so time taken is minimal even if the service transfer by MRT trains is impossible after all searching processes.

Every bus stop in Singapore is given a unique ID number, and this helps when commuters can actually send in the bus stop ID and their destination to check whether they are taking the bus service on the correct side of the road.

4.1.4.3 Public Service Transport Dynamic Module and Server Module

Dynamic module, together with server module is responsible for real time transmission via the GSM data cable and GSM phone. This includes features like arrival information, bus flagging, bus stopping alert, and ad hoc messages. The dynamic module responds to several types of query:

- Commuters send in the bus stop ID and the bus service with additional words like arrival, to query for the arrival times of a particular bus service.



Fig. 4.5 – Bus Arrival Times

- Commuters send in the bus stop ID and the bus service with additional word like flag, to flag for the next incoming bus at the bus stop.



Fig. 4.6 – Bus Flagging

- Commuters send in their destination road or bus stop ID with additional word like bus alert, road name and license plate number to set an alert to the system to inform that they are reaching their destination.

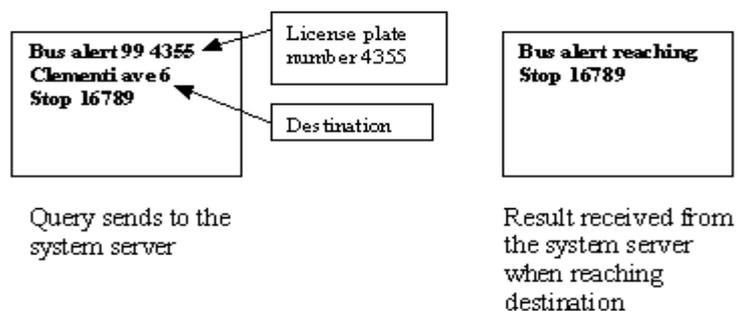


Fig. 4.7 – SMS Alert When Reaching Specific Stop

- Bus drivers or operators send in ad hoc messages to the system.

For the prototype system, a few bus services with fixed arrival schedule are chosen. To simulate real time arrival without the use of Automatic Vehicle Location (AVL), the dynamic module needs to add in a delay generator. The delay generator occasionally adds in a few minutes delay to the arrival schedule. This way, commuters receive SMS messages showing the fixed arrival times of the bus with the delay. (e.g 18:00 +3min)

When commuters flag a bus via SMS, the module compares the current time with the fixed schedule of the bus service together with the delay. The module looks out for the incoming bus arriving at the bus stop and sends a flagging alert to the bus driver. In the case of full capacity of the bus during peak hours, the bus driver can actually send in ad hoc message to the system informing it that his/her bus capacity is full. The module then redirects the flagging message to the next incoming bus. Bus drivers can also send in other ad hoc messages to the system, in the case of accidents, congestions, etc. In the case of emergency cancellation of roads, the system operator can make use of the system to send out ad hoc information to related bus services on the roads.

The module has feature whereby commuters set an alarm to alert them when they are reaching a particular road or bus stop. A database is created to store information of the commuters' mobile phone number, their destinations that can be either road names or bus stop IDs, and most importantly the bus service license plate number. When buses are reaching destinations, the module immediately sends SMS messages to commuters' mobile phone to alert them that they are reaching their destinations.

After development, testing is done to further fine tune the prototype system, and it proves to be a good working system.

4.2 Car Parking System

At present, motorists use parking coupons for most public car parks in Singapore. When required, parking coupon is to be displayed on the dashboard. Motorists can look for signboards which indicate parking rates and free parking times as they vary for different car parks.

From 1st September 2002, the new hourly parking charges for cars at public car parks in the Restricted Zone is \$1.00 per half hour, and \$0.50 per half hour at car parks outside the Restricted Zone. New parking charge for motorcycles is \$0.65 per day. The hourly parking charges for heavy vehicles has been revised to \$2.00 per half hour at car parks in the Restricted Zone, and \$1.00 per half hour at car parks outside the Restricted Zone. Details of the revised parking charges are provided in Appendix A.

Parking coupons are available from petrol stations, post offices, shopping centers and neighbourhood spots. Presently, there are car park attendants in car parks that check for illegal parking. They usually have to physically go in front of the windscreen and check whether the coupons are there or the coupons have used up the time period. If there is illegal parking, the car park attendant will book the car with a handheld device whereby it will issue out a fine ticket to be placed on the windscreen.

There are several ways to replace the coupon parking payment system, for instance cash cards can be used for car parking payment. Cash card payments at auto pay stations in car parks started as a pilot project in 1997 at two HDB multi-storey car parks. Since then, the project had been extended to close to 100 car parks island-wide. Today, NETS processes monthly average about 700,000 cash card transactions from car parks such as Great World City, Ngee Ann City, and Jurong Point. With more cash card-enabled car parks available, motorists can look forward to even greater convenience in settling their parking fees. There is now little need for notes and spare

change. Not only do such car parks bring about convenience for motorists, the automation of parking fees payment helps operators to save time, manpower, and money in the long-term.

4.2.1 SMS Implementation and its Objectives

In this project, SMS service will be used for car parking payment. In addition, it can provide information like the current capacity of a car park and its parking rate. All parking costs during the month are added into motorists' mobile phone bill.

There are several advantages of this service, some of which are as follows:

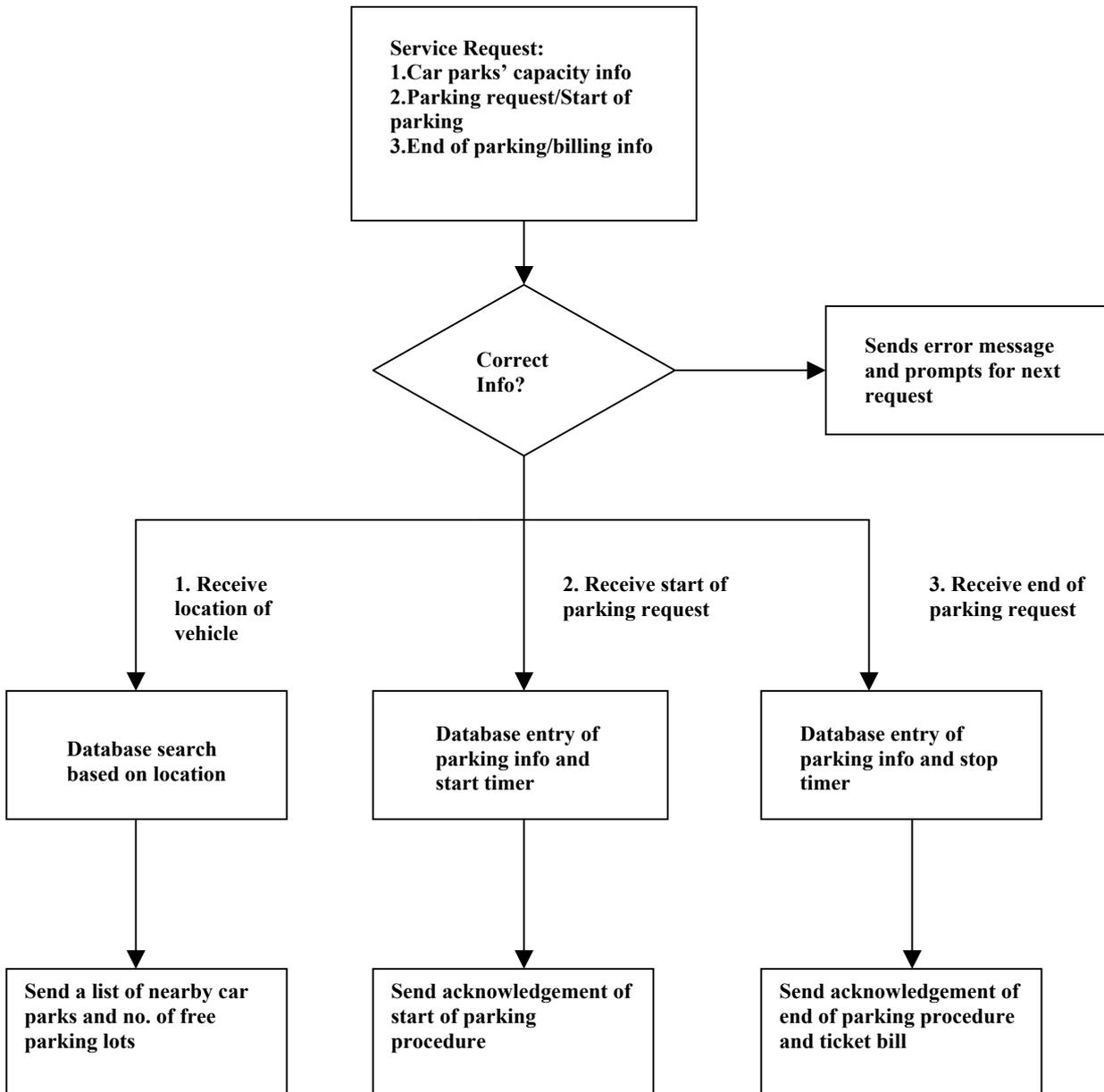
- Motorists need not waste time looking for a place to park in a car park, especially during peak periods.
- They can save the trouble of using parking coupon as payment. When the parking time is used up, the motorists need not have to drive off his car or add another coupon to extend the parking time.
- Motorists need not have to occasionally check whether their cash cards have enough stored value with the cash card system.

The SMS system developed in the car parking system will first provide an information service for motorists. This information includes the availability of car parks near a motorist's area when he/she requests for it. Availability of parking space for each car park will also be available in the information service if the car park is monitored.

Secondly, the car parking payment will be made via SMS in the system. The system will record the motorist vehicle information when he/she sends his/her parking request via SMS to the system. In this way, the system will be able to monitor each car park effectively and the information provided for the availability of parking space will be more accurate. A prototype is developed to test out the ease of use of the system.

4.2.2 Functional Flow Block Diagram

Parking Service Request



4.2.3 Prototype System Overview

Based on the general prototype system model in figure 3.2, the central computer system consists of system server, GSM data cable, and GSM phone connected to the server via the cable. The system server is needed as the source of interaction between car parks' information and motorists. The system server is used to maintain databases of car parks and provides this information to motorists when requested. When a motorist sends an SMS message through the GSM network to the central computer system, the server takes note of the kind of request from the motorist. The request can be queries about the availability of parking space in his/her area, or about parking his/her vehicle in a particular car park. For the query of parking place, the system server replies with a list of car parks and their capacities. And for the latter, the system server automatically adds the information into the database to start the parking procedure as well as updates itself on the information of the car park. It communicates with the motorist's mobile phone by means of SMS through the GSM data cable connected to a GSM phone.

The prototype system hardware is the same for all prototypes discussed in the project. It is described in Chapter 3.

4.2.4 Prototype Software Developments and Testing

The system software is composed of a number of different modular blocks as shown in figure 3.1 in chapter 3.

4.2.4.1 Car Parking Database Module

For the prototype system, few car parks' information is recorded in the Microsoft Access database. They are named car park A, B and C. The information in the database

includes capacities of each car park, parking procedure like start and end time, cost of parking fee, license plate number, mobile phone number, etc. During prototype testing, a motorist sends a SMS request on the capacity of car parks around his/her area, and the database module extracts from the car park databases a list of car parks near the motorist's area and their capacities. Figure 4.8 shows a road map example of a motorist looking for a car park. The motorist get information of car parks A, B and C, which are near him, and not car park D.

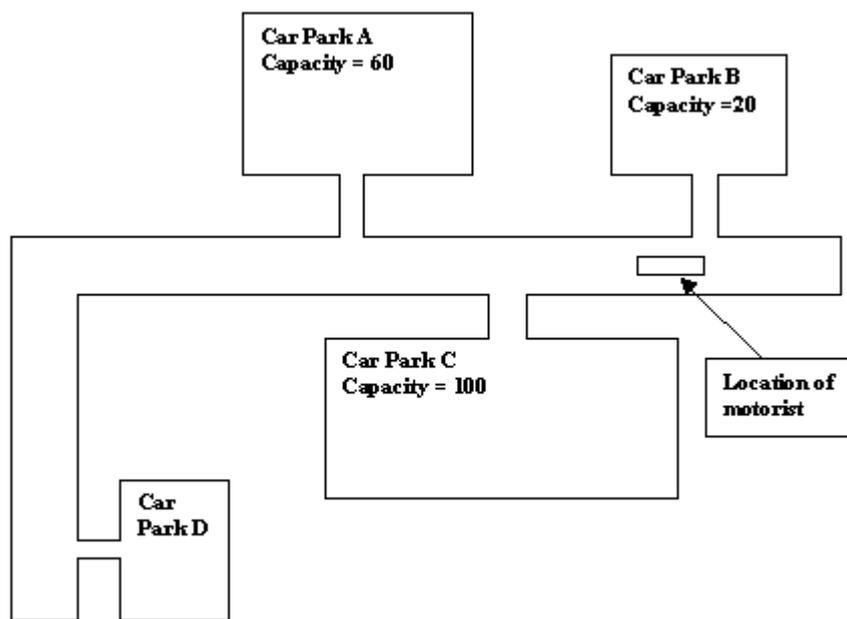


Fig. 4.8 – Prototype Road Map with Car Parks

After the motorist has chosen a car park and parks into a parking lot, he/she sends a SMS message to the system informing the start time of the parking period. The database module then records this information and replies to the motorist with a confirmation message. With the help of SMS, all these procedures happen in a few milliseconds, thus, time is saved as compared to time taken in placing of parking coupon, where motorists need to note down the time, date, etc on the coupon. When the motorist is exiting the car park, he/she just needs to reply back to the confirmation

message that he/she is exiting. The database module records down the ending time and the cost of parking and replies this information to the motorist.

4.2.4.2 Car Parking Software and Server Module

The software and server module is responsible to organize and recognize the text message received from the motorist via SMS. It determines the information that the motorist has requested from the received message. When a motorist requests for car parks' information, it interacts with the database module to get the required data. At the end of a parking procedure, the module takes note of the parking period and its cost. Then, it sends this information to the database module for updating the database, as well as, informs the motorist of the parking cost via SMS.

The module reacts to SMS requests from motorists and provides the following information:

- **Car parks' capacity information:** motorists are able to see the capacity of various car parks near his area when they request the system for this information through SMS. The result shows the available car parks and the number of free parking lots.

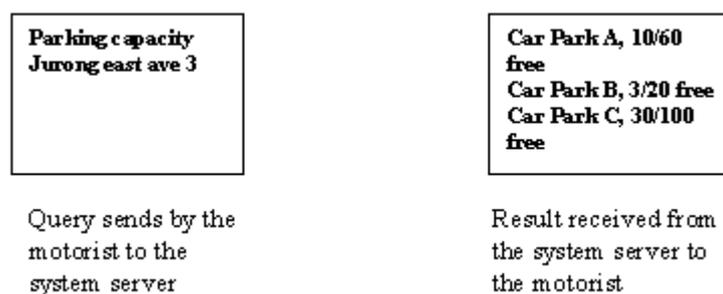


Fig. 4.9 – Car Parks' Capacity Information

- **Parking request and start of parking procedure:** motorists are able to park their cars using SMS. A motorist needs to state the name of the car park, where

he/she needs to park. The system then starts the parking procedure as well as replies to the motorist informing him/her of the start of the parking time.

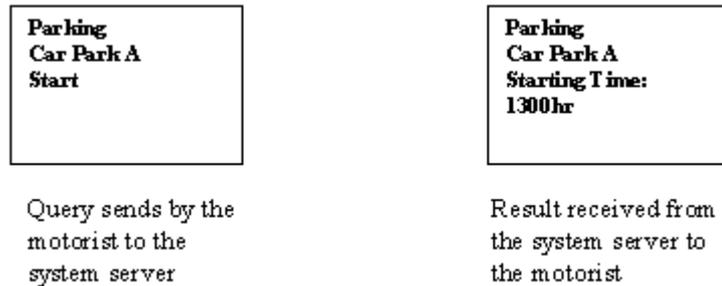


Fig. 4.10 – Parking Request and Start of Parking Procedure

- **End of parking procedure and billing information:** A motorist needs to end a parking procedure by sending a SMS message to the system. He/She get notified of the end time of their parking period and the cost of parking by the system. At the same time, the car park database frees up a vacancy for the car park and updates itself over the server.

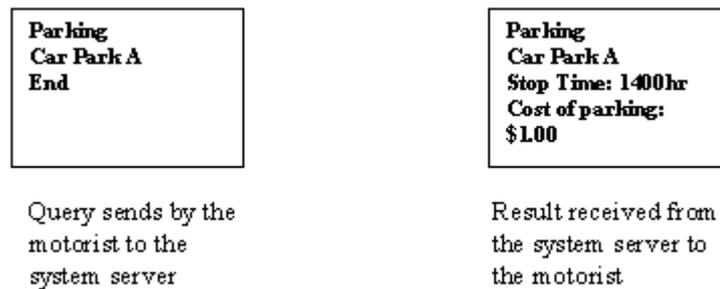


Fig. 4.11 – End of Parking Procedure and Billing Information

After the development, testing is done whereby several sample mobile numbers with vehicle information are simulated in the system and it works well.

4.3 SMRT And Changi Airport

Changi Airport is now connected with the rest of Singapore through the MRT network. Passengers take about 30 minutes to travel between the airport and the city. The through service between Boon Lay and Expo stations has been extended to serve Changi Airport station. This through service, which provides a direct connection between the Changi Airport MRT Extension and the East West MRT Line, operates at an average frequency of 12 minutes.

The first and last train timings of the Changi Airport Extension are as follows:

	First Train		Last Train
	Mon - Sat	Sun / Public Holidays	Daily
Changi Airport to City Hall	5.31 am	5.59 am	11.18 am
City Hall to Changi Airport	6.09 am	6.45 am	12.03 am

Passengers heading to Changi Airport station are advised to note the terminal stations (Pasir Ris or Changi Airport) displayed on the plasma screens at the station platform. As a reminder to passengers, in-train announcements will also be made from Kembangan station onwards for trains headed for Changi Airport station. For passenger convenience, the 12 trains deployed on the through service are retrofitted with luggage racks and have free space next to the doorways. Passengers are advised to note that the luggage brought into the train should not exceed 81 cm x 58 cm x 30 cm.

4.3.1 SMS Implementation and its Objectives

By linking the information systems of MRT and Changi Airport [8] with SMS service, commuters will be able to plan their journeys to the airport. Advantages of this service are as follows:

- Commuters will be able to know whether the next train arriving is heading towards Changi Airport station.
- Commuters will know when to take the train service that will bring them to the airport in time.

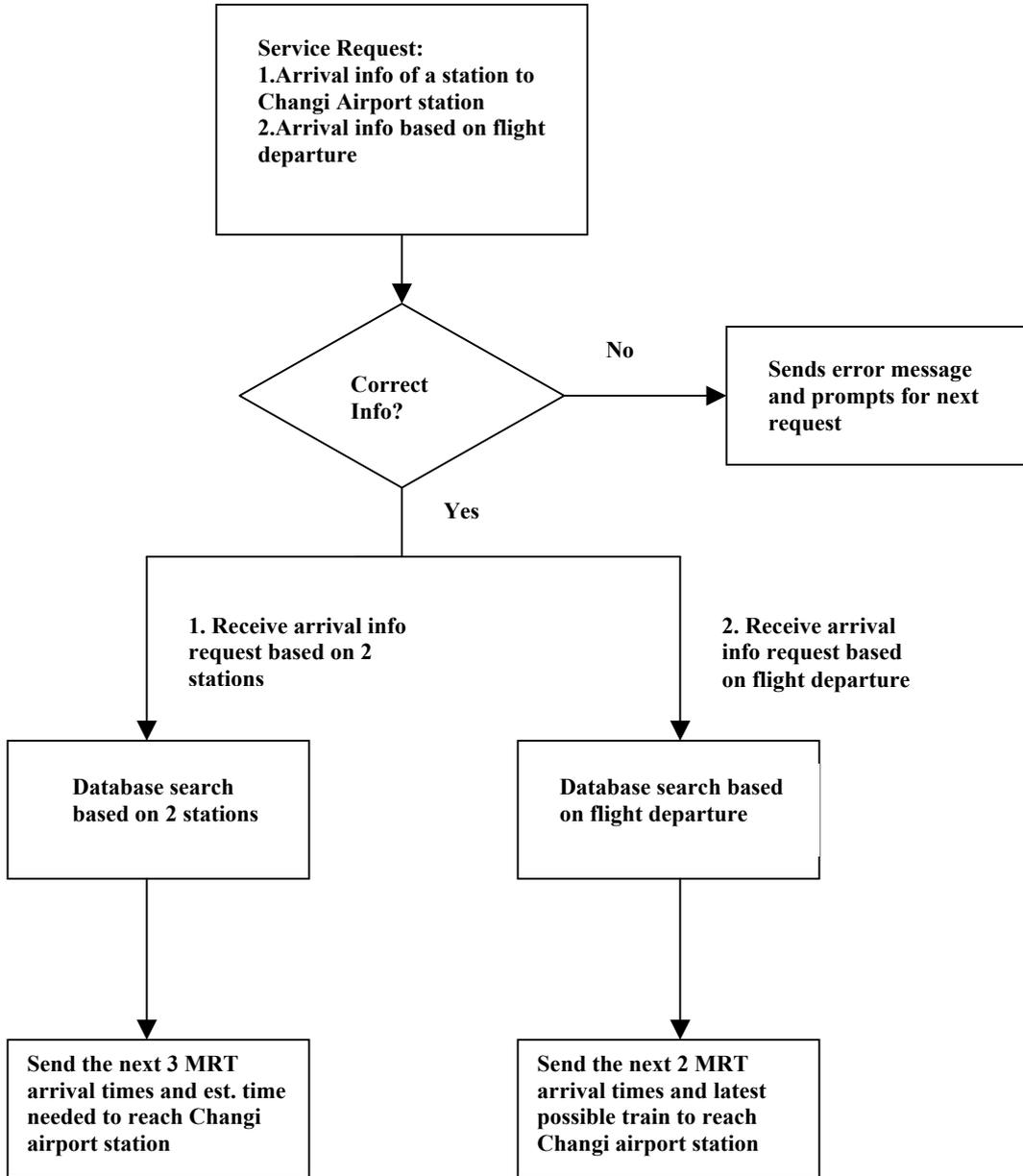
The SMS system will need two types of information: MRT train schedules and airport schedules. Commuters traveling on the North South MRT Line to Changi Airport station will need to transfer train service at Raffles or City Hall MRT interchange stations. Therefore it is important for the SMS system to have good software accuracy so that commuters do not miss their trains when they transfer at the interchange stations.

Additionally, the system will integrate airport schedules and MRT train schedules. The system software has to plan out accurately both schedules such that commuters going overseas do not miss their departures if they intend to travel by MRT train.

In this project, a prototype is developed to integrate both schedules to facilitate the passengers traveling to and from the Changi Airport.

4.3.2 Functional Flow Block Diagram

MRT Arrival Service Request



4.3.3 Prototype System Overview

From the general prototype system model in figure 3.2, the prototype system consists of three parts; central computer system, GSM network and GSM phone users. The central computer system is responsible for providing relevant information to commuters who intend to travel to the Airport by MRT train. For the prototype, fixed schedules are used for the testing just to prove the accessibility and convenience of the system. In real time, there may be cases of delays for MRT train arriving and changes of timing for flight arrivals and departures in the airport, and these situations require the system to be updated constantly. The GSM phone connected to the system server via the GSM data cable is responsible for sending and receiving SMS messages. The system only reacts when it receives commuters' SMS requests for information. According to the type of request from the commuter, the system searches through the database for the required information and sends it as SMS message to the commuter via the GSM phone.

The GSM network manages all SMS messages and making sure that they are sent to the correct recipients. The prototype system hardware is the same for all prototypes discussed in the project. It was described in Chapter 3.

4.3.4 Prototype Software Developments and Testing

The system software is composed of a number of different modular blocks as shown in figure 3.1 in chapter 3.

4.3.4.1 MRT and Airport Database Module

For the prototype system, fixed timing for MRT trains heading to Changi Airport station are stored in the Microsoft Access database. As the trains go from west to east

on the MRT network, stations along the east-west line of the network are recorded for further simplification. As for the airport information, simulated timings are used for the prototype. The module extracts information from the database only when the commuter requests for it. The information includes arrival times of MRT trains toward Changi Airport at each station, approximate time to reach Changi Airport station, etc. Therefore, there is no requirement for the prototype to have entry for changes of data.

4.3.4.2 MRT and Airport Software Module

The software module breaks down the text message received from the commuter via SMS and tries to identify the type of queries that the commuter requested. Based on the type of queries, the software module interacts with the database module to extract the relevant information from the database.

The module reacts to SMS requests from commuters and provides the following information:

- **MRT arrival information to Changi Airport station:** Commuter sends a SMS message on request for arrival information between a MRT station and the Changi Airport station. The system replies to the commuter on the next three MRT arrivals to the station as well as the estimated time needed to travel to Changi Airport station.

```
MRT
Changi Airport
Clementi
```

Query of MRT train
arrivals toward airport
at a station

```
MRT arrivals:
1) 2min
2) 13min
3) 25min
take abt 45 min to airport
```

Result sent to the user
about arrival times and
time needed to airport

Fig. 4.12 – MRT Arrivals Information to Changi Airport Station

- **MRT arrival information bases on flight departure:** Commuter sends a SMS message with the information on a flight departure and the MRT station he/she is located. The system replies to the commuter on the next two MRT arrivals to the station as well as the estimated time needed to travel to Changi Airport station. Most importantly, it also states the latest possible train the commuter must take, so that he/she will not be late.

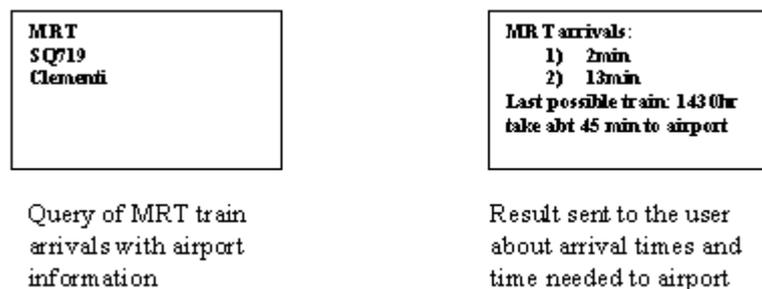


Fig. 4.13 – MRT Arrivals Information Based on Flight Departure

The prototype is tested out for its accessibility with the fixed schedules only. The prototype shows promising results and future work whereby unfixed and more dynamic schedules will be used to test out the full functionality of the prototype system.

4.4 Illegal Parking Ticket System

In Singapore, the number of tickets issues to local motorists average 5,200 a day. The police have engaged a private organization, that sends out traffic attendants around the country to fine motorists for parking at wrong places. Some of the most notable offenses that can be ticket include motorist speeding, smoking outside the designated smoking zone by 3 centimeters and insufficient amount stored in the cash-card while going through an ERP gantry.

Here are some ways that motorists can pay their fines:

- Online payment at the vPOST website: www.vpost.com.sg.
- Motorists can pay by NETS at all SAM (Self-service Automated Machine) before the expiry date of payment.
- Use the Automated Traffic Offence Management System (ATOMS) to settle their fines via AXS Station for minor traffic offences.
- Go to any post office with your notice of fine and pay with NETS, CashCard or Cash. If motorists are paying by Cheque, Money Orders or Postal Orders, it is to be made payable to Singapore Post Pte Ltd.
- By Cheques or Postal Orders making it payable to the “Land Transport Authority” and send it to 10 Sin Ming Drive, Singapore 575701.

In the illegal parking fine system, one of the major shortcomings is inefficiency. This is a serious problem, especially when there are a lot of vehicles parked illegally along a roadside. The traffic attendant usually brings along an electronic device whereby he/she keys in the vehicle’s license plate number into the device. The device is supposed to print out a ticket to be issued to the vehicle. During this period, motorists who spot the attendant have sufficient time to drive off their vehicles.

Another shortcoming creates difficulty for the motorists. They may find it troublesome to pay up their tickets at designated areas. Some may forget about paying up or topping up their cash cards to pay the fines.

4.4.1 SMS Implementation and its Objectives

By using SMS technology to pay the ticket, motorists need not even use the e-commerce system. A system that uses SMS technology can be designed to do all the tickets payment. Here are some of the advantages of SMS fine payment system:

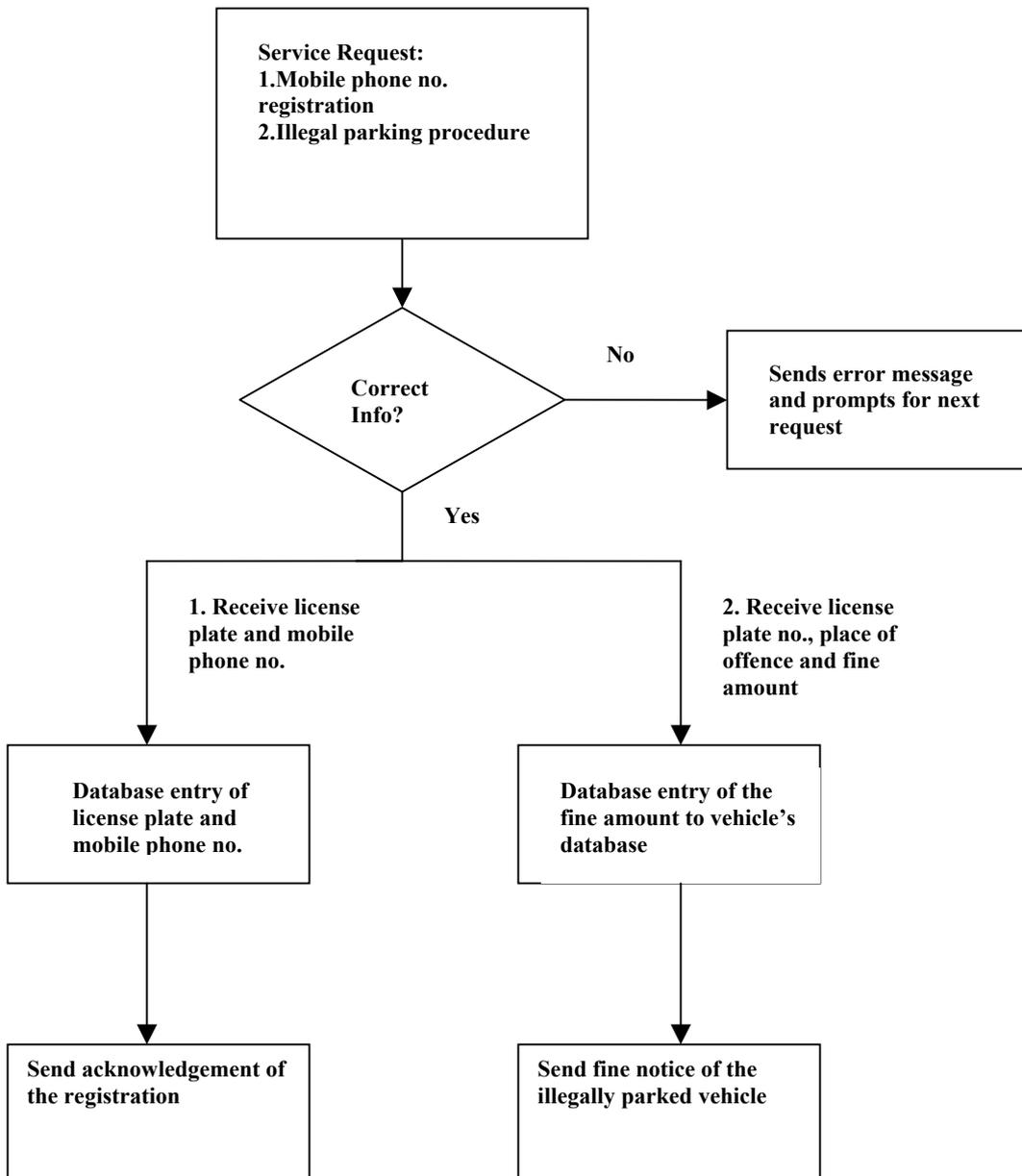
- Motorists need not go to post office or SAM to pay their fines.
- Saves from troubles of forgetting to pay or losing their tickets.
- They need not have to top up their cash cards to pay their tickets.
- Traffic attendant can immediately book motorists whom have parked illegally and they will be informed about the tickets via SMS.

The SMS system has database of all vehicles and their owners' mobile phone numbers and GSM network companies. Microsoft Access is used to contain the database. There are two databases in this system. One is used for recording information of the vehicles. The required information fields are vehicles' license plate numbers, owners' mobile phone numbers and the owners' home addresses. The other database is used for recording tickets notice for each owner. The required fields are date/time of the ticket issued, amount of ticket payment and the place of parking offence. The system will be on standby at all times to receive any ticket notices from traffic attendants. For any ticket notice received, the system takes note of the vehicle and its owner and sends a SMS notification to the owner's mobile phone number. The place, time and the amount of fine are reflected in the SMS message received by the owner and the amount of fine will be added into the owner's mobile phone bill.

This project will produce a working prototype to test out the system's functionality and speed of issuing tickets to motorists.

4.4.2 Functional Flow Block Diagram

Illegal Parking Service Request



4.4.3 Prototype System Overview

Using the general prototype system model in figure 3.2, the central computer system consists of the system server, GSM data cable and the GSM phone connected to the server via the cable. The system server is responsible for the interaction between illegal parking information, traffic attendant and motorists. It records every illegal parking bookings received from the traffic attendant via SMS into the database. The central computer system then sends a notification message to the motorist who is being issued the ticket. Whenever a traffic attendant spots a vehicle parking illegally, he/she books the vehicle by sending a SMS message of the vehicle license plate number to the system by using either a GSM phone or a specially designed booking device. The GSM network transmits this information to the GSM phone connected to the server. After the ticket is recorded, the motorist is notified of the ticket via SMS sent by the system. The ticket information that the motorist receives via SMS consisted of the name of the road, the amount of that ticket, and the total amount of tickets for the month.

The prototype system hardware is the same for all prototypes discussed in the project. It is described in Chapter 3.

4.4.4 Prototype Software Developments and Testing

Using the general system software structure in figure 3.1, the prototype is composed of a number of different modular blocks.

4.4.4.1 Illegal Parking Fine Database Module

In the prototype system, the database module is responsible for recording fine information into the database. There are two types of record set in the database. One is

for recording fine information and the other one is for registering of mobile phone number of vehicle. In order for the prototype system to work, a motorist needs to register his/her mobile phone number with his/her vehicle to the database. When the traffic attendant sends a fine request to the system with a vehicle's license plate number, the database module identifies the owner's mobile phone number and other pertinent information. The fine request includes the license plate number of the vehicle, the amount of fine and the name of the road that it parks illegally. The database module then combines all the information and adds it into the database. The information includes the license plate number, the motorist's mobile phone number, the amount of fine, etc. After adding the information into the database, the module then interacts with the software module to send a SMS notification message to the owner 's mobile phone.

4.4.4.2 Illegal Parking Fine Software and Server Module

The software and server module is responsible for identifying and analyzing the text message received from the traffic attendant and the motorist via SMS. It is supposed to know when the motorist registers his/her mobile phone number with his/her vehicle into the database. When a traffic attendant sends a fine request, it is supposed to break down the message into relevant details and forward them to the database module for recording. The relevant details include the vehicle license plate number, the name of road, the amount of ticket, the time of the ticket issued, etc.

These are the following steps that the module reacts to SMS messages from registering and fine request:

- **Registering mobile phone number to the database:** Motorists are required to register his/her mobile phone number and the vehicle registration number with

the system. Motorists can just send his/her vehicle license plate number to the system and the software takes in the license plate number as well as the motorist's mobile phone number. Any ticket will be automatically added into his/her mobile phone bill.

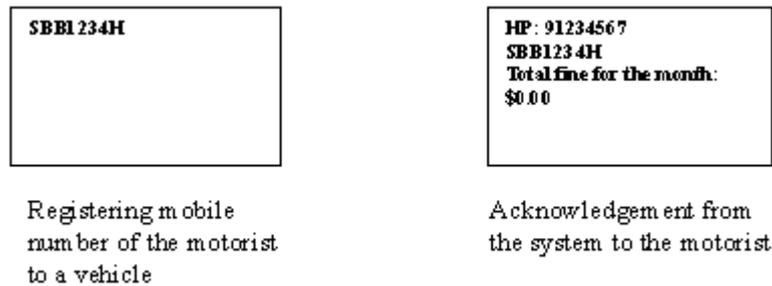


Fig. 4.14 – Registering Mobile Phone Number to The Database

- **Illegal Parking Ticket procedure:** Upon finding an illegal parked vehicle, the traffic attendant sends a SMS message that includes the vehicle license plate number, the name of the road, and the amount of ticket to the system. The module then breaks down this information and interacts with the database module to record this information. After the ticket procedure is completed, the module interacts again with the database module for contact information on the motorist. It then sends a notification message from the server to the motorist. The notification message includes the vehicle license plate number, the name of the road, time of the ticket issued, the amount of the ticket, and the total tickets for the month.

SBB1234H
Serangoon road
Illegal parking \$70.00

Fine statement sent to
the system by Traffic
Police

SBB1234H
Serangoon road
Time: 1434 hr
Illegal parking \$70.00
Total fine for the month:
\$200.00

Fine notice sent to the
motorist by the system

Fig. 4.15 – Illegal Parking Fine Procedure

Additionally, the prototype can also help to find stolen vehicle. If a vehicle has been reported stolen, the prototype notifies the traffic attendant when he/she has booked the vehicle, which might happen to be illegally parked. Testing of the prototype is done and it proves to be faster than the usual method of issuing fines.

4.5 ERP Billing System

Electronic Road Pricing (ERP) is an electronic system of road pricing [7]. It is designed to automate the current road pricing system of Singapore - no more paper coupons or enforcement officers at the gantries. The main difference is the pay-when-you-use principle. This is a fair system as the motorist is charged only if he passes through the ERP gantry.

The ERP system consists of four main components: the In-vehicle unit (IU), the cash card, the gantry, and the central computer system. The IU is an electronic device installed in the vehicle that communicates with the gantry point equipment and accepts a cash card for payment of ERP charges. There are different types of IU, differentiated by color, for different categories of vehicle to allow different road pricing. The cash card is a stored value card for ERP payment. Motorists in Singapore can easily buy the card in banks, Singapore Post Outlet, cash card auto machines, and petrol stations. There is a low balance indicator in the IU such that warning will be given if the cash card balance is low. The driver then can top-up the cash card at ATM machine, cash card auto machines, NETS kiosks of cash card service terminals spread throughout Singapore. The original cash card needed to be kept away from sunlight but the new one with a sun logo is heat resistant such that the motorists can leave the card in IU for a reasonable length of time. License plates of vehicles, which make illegal entries are photographed by the gantry cameras for subsequent enforcement action. The central computer system in the control room monitors the traffic network and equipment. It also recognizes the license plate on vehicles in the photographs and classifies the errors or violations automatically.

Although the system is successful until now, there are still some shortcomings:

- Motorists have to top up their cash cards when values are low. This is troublesome and a waste of time for motorists to go to ATM machines for topping up their cash cards.
- They have to remember to insert their cash card into the IU before entering the gantries.
- Thefts of cash card may happen when motorists leave their cash card in the vehicles.
- The cash cards or the IU may be faulty sometimes and the gantry point equipment may not detect the payment.

4.5.1 SMS Implementations and its Objectives

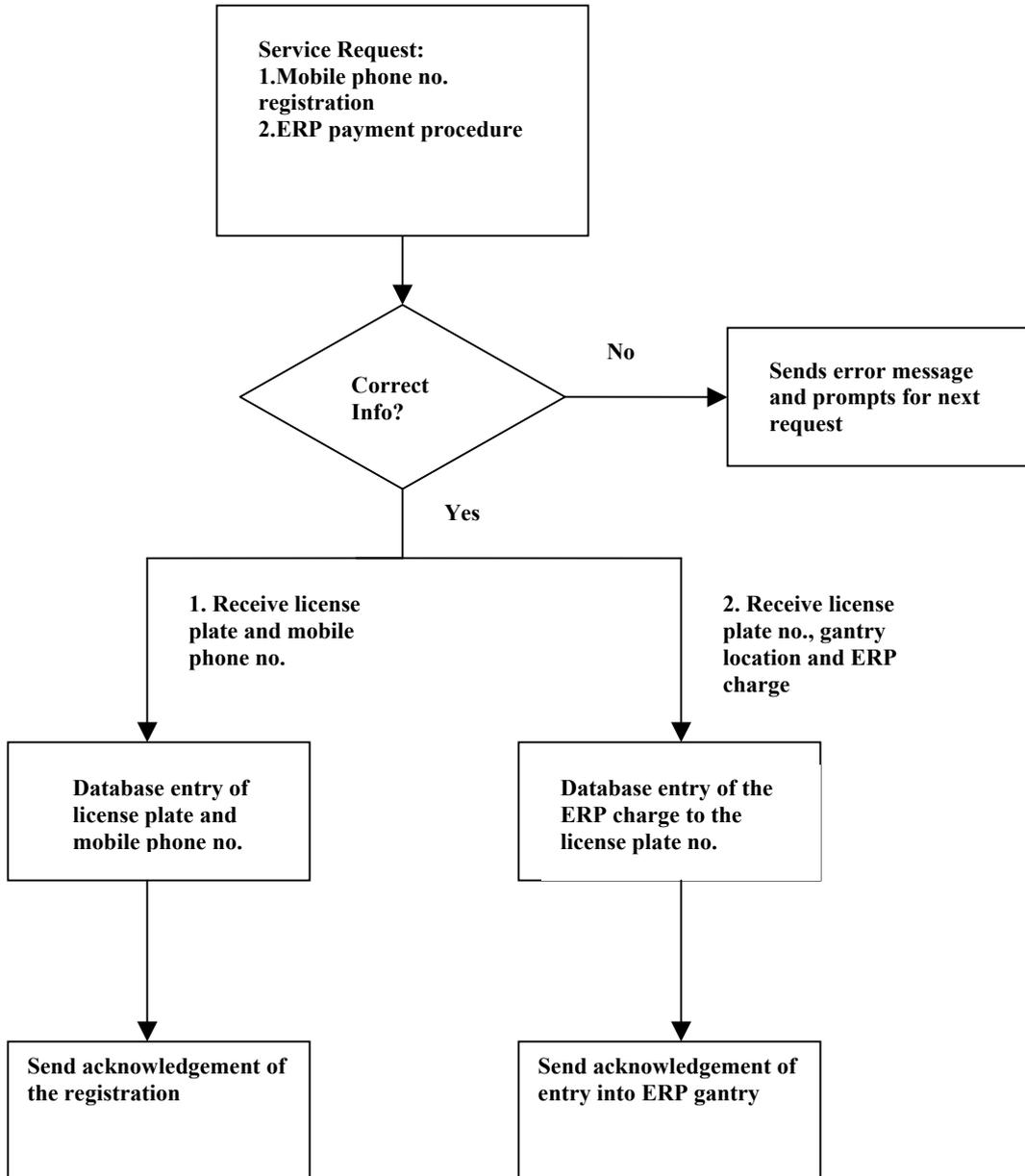
The cash card payment for the ERP system can be replaced by SMS technology. In this way, motorists do not need to worry about topping up their cash cards, as gantry payment will be added to their phone bills. The IU can still be used when entering the gantry, but no cash card needs to be inserted into it. Instead, once the scan at the gantry notices the vehicle and IU, the central computer system sends a SMS message to the mobile phone of the motorist, informing him that he has entered and been scanned by the gantry.

In case, when the IU becomes faulty, the scan at the gantry will notice the faulty IU and the camera will take picture of the vehicle. The central computer system then notes down the license plate of the vehicle and sends a SMS message to the mobile phone of the motorist. This time, the system will inform him about the entry of the gantry and additionally, informing about the faulty IU in his vehicle. This saves the trouble of the motorist going to the traffic police department and explaining to them that his/her IU unit is faulty.

A prototype is developed to simulate the ERP system with SMS and testing is conducted to see if the system works and is better than the old system.

4.5.2 Functional Flow Block Diagram

ERP Billing Service Request



4.5.3 Prototype System Overview

Based on the general prototype system model in figure 3.2, the system server, GSM data cable and the GSM phone connected to the server via the cable form together the central computer system. For the prototype, the system server take note of which mobile phone number belong to which vehicle, so that ERP payment will not charge wrong motorist. In real case, upon entry of the gantry, the gantry sensor identifies the IU unit of the vehicle and sends the license plate number to the system. The system then searches through the database to find the mobile phone number belonging to the vehicle. With the information, the system sends a SMS message to the motorist's mobile phone number through the GSM network informing him/her of the payment for the ERP entry. The system server is also responsible for the illegal entry of ERP gantry information. It records information of motorist who illegally enters the gantry via SMS into the database. As the prototype system is built within a PC with GSM data cable and phone, certain technical components like the ERP gantry, sensor and camera are included. In this case, some software programs are used to simulate motorist entering the gantry and the system records down the motorist's information. The objective of the prototype is to show the accessibility of implementing SMS payment for the ERP system instead of cash card payment.

The prototype system hardware is the same for all prototypes discussed in the project. It is mentioned in Chapter 3.

4.5.4 Prototype Software Developments and Testing

The system software is composed of a number of different modular blocks as shown in figure 3.1.

4.5.4.1 ERP Billing Database Module

There are two types of record set in the ERP Billing database. One is for registration of mobile phone numbers of vehicle and the other one is for recording license plate number of vehicle that enters the gantry illegally. In the prototype system, the database module is responsible for recording fine information into the database. For the prototype system, only a few mobile numbers and their assigned vehicle license plate numbers are stored in the database for simulation.

When a simulated vehicle passes through the gantry, the license plate number of the vehicle is sent to the database module. The database module then searches through the database for the mobile number of the vehicle. If the search is successful, it interacts with the software module to send out a notification message to the motorist's mobile phone number. If there is no such license plate number found in the database, the module considers that the vehicle has already entered the ERP gantry illegally and it is recorded in the illegal ERP entry database.

Additional database is also setup to record all the ERP bills belonging to individual motorist.

4.5.4.2 ERP Billing Software and Server Module

The software and server module is needed to identify and analyze the text message received from motorists during registration of their mobile phone numbers with their vehicles. When a vehicle passes through the ERP gantry, the module needs to send SMS information of the ERP payment notice to the correct motorist's mobile phone.

These are the steps that the module performs based on SMS messages for registrations and SMS payment notices for ERP entry:

- Registering into ERP Billing System:** Before the ERP system can work, the motorist have to register his/her mobile phone number with his/her license plate number. The motorist has to send a SMS message of his/her vehicle license plate number to the system. The software and server module then take in the message and the mobile phone number and pass them to the database module, which records them into the database. For the prototype system, some mobile phone numbers and vehicles' license plate numbers are stored in the database as simulation parameters.

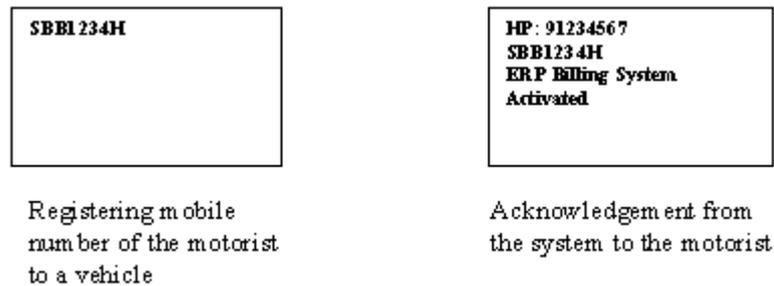


Fig. 4.16 – Registering into ERP Billing System

- Acknowledgement of payment from ERP Billing System:** When the registered vehicle passes through the ERP gantry, the software and server module receive the license plate number of the vehicle and send it to the database module. The database module then searches through the database for the mobile phone number of the vehicle. If the database module finds the number, it sends it to the software and server module. The software and server module then send an acknowledgment SMS message to the motorist's mobile phone, informing him/her of the entry into the ERP gantry. For the prototype system, simulation of faulty IU unit is done to test out if the module informs the motorist about it. In real case, if the gantry sensor cannot sense the IU unit, the gantry camera has to take picture of the vehicle's license plate number. The

system then can make use of this picture to find out if the motorist and his/her vehicle information are in the database. If it is, an acknowledgment of gantry entry with faulty IU is sent to the motorist; else the vehicle is recorded as illegal entry into the ERP gantry.

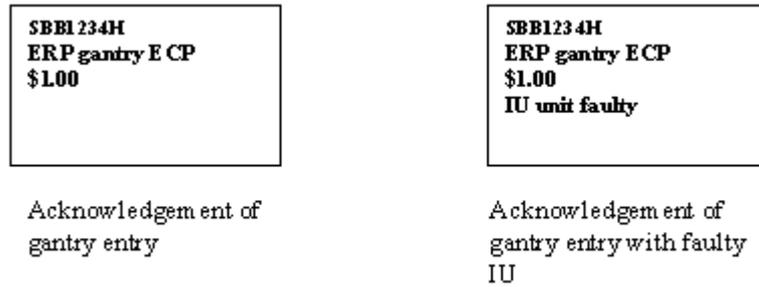


Fig. 4.17 – Acknowledgement of Payment From ERP Billing System

The testing of the prototype proves to be promising, although the hardware used is limited. Some further testing/simulation and field trials are also recommended before the system is launched.

4.6 Vehicle Burglar Alarm System

Although vehicle theft is not a big problem in Singapore, vehicles get stolen despite stringent policing and strict enforcement by the authorities [9]. About 1651 vehicle thefts were reported in 2000, a decrease of 23.2% from 1999 (2149 cases). This averages out to one vehicle stolen every 5.3 hours. Cars that are stolen are usually of high value and often find their way to neighbouring countries through one of the two causeways or via car ferry services, as a convenient means of disposing them.

Similarly, there is also a growing concern regarding theft of personal effects and accessories from cars. After all, some owners discard the standard audio systems fitted in their cars for those that cost more than a complete home theatre system. In addition, cash cards that are left inserted in the IU (In-vehicle Unit) are extremely attractive to thieves looking for prey. Many motorists are also in the habit of leaving their bags, laptops, cameras and other expensive items in their cars, in clear view of others.

Although some owners install vehicle security systems in their vehicle, vehicle thefts can still happen. Some thieves normally just smash the window screens of vehicles and steal any valuables inside. This takes only a few seconds. By the time the alarm sounds off and the owner rushes back, it may be too late. Some vehicle security systems are able to alert owner of thefts by transmitting radio wave to their mobile security kit. But normally, vehicle thefts happen during late nights or early mornings where vehicle owners are asleep and might not be waken up by the alarm.

4.6.1 SMS Implementations and its Objectives

The SMS system aims to design a vehicle security system that can send alert messages not only to the vehicle owners' mobile phone only, but also to the police on patrol via SMS when necessary. In this way, the police will be able to get to the location of the

stolen vehicle, when informed via SMS and they might be fast enough to catch the thief.

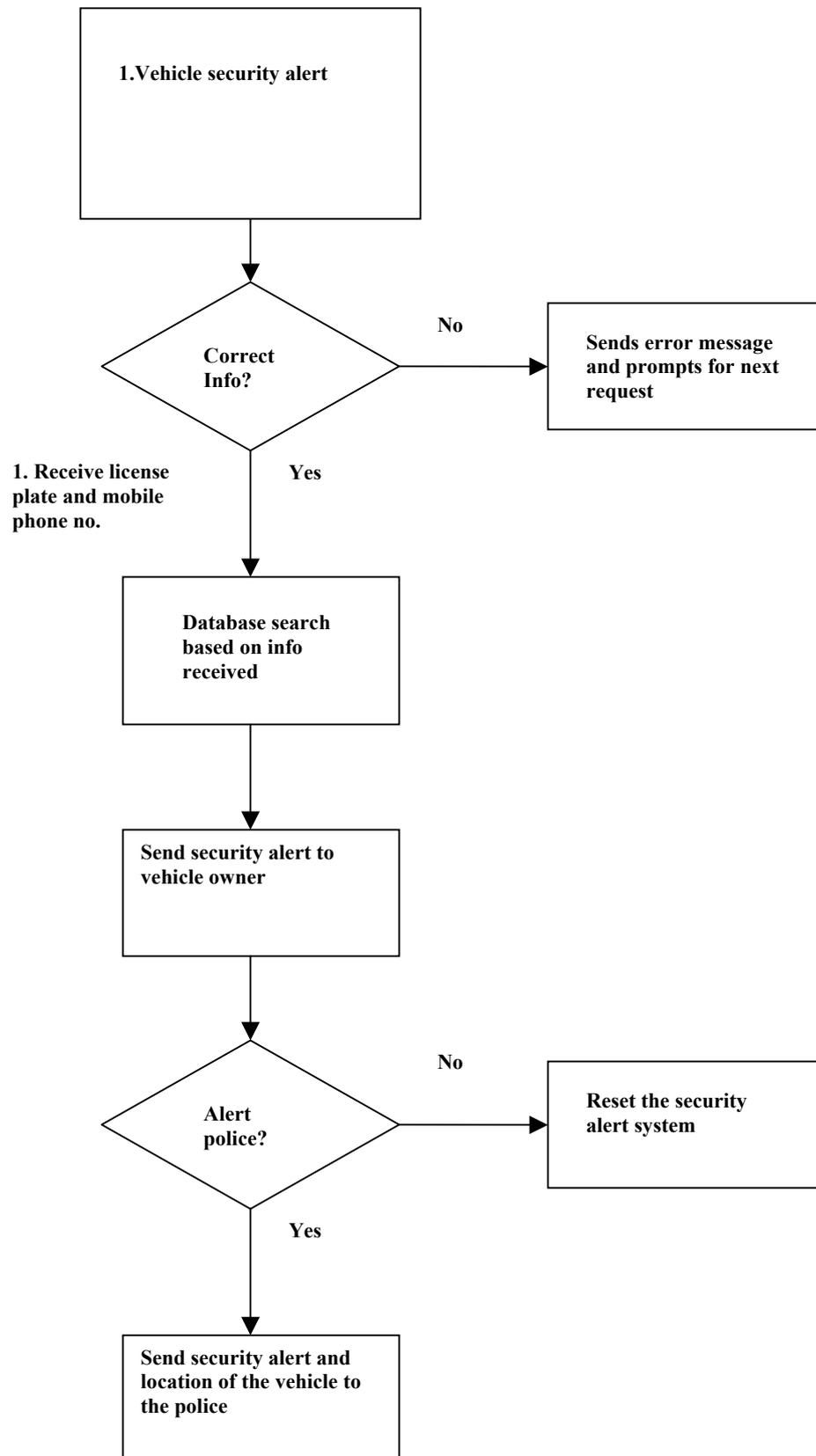
The security system can also be customized not only just to alert the owner's mobile phone, and also his family members who have mobile phones. Here are the following features of the vehicle security system:

- 1) Once the alarm sounds off, an alert message will appear on the owner's mobile phone.
- 2) The owner can also make use of the vehicle security system to send the message to multiple people who own mobile phones, for example, his family members.
- 3) If the security system can be customized in such that it can record down the car park information. When the alarm sounds off and if the owner discovers that the thief is still in the area of crime, he/she can send the alert to the police on patrol.

Prototype is developed to simulate SMS with the alarm system. It is tested to analyze the effectiveness of implementing SMS alert to the alarm system in vehicle.

4.6.2 Functional Flow Block Diagram

Vehicle Security Alert



4.6.3 Prototype System Overview

Referring from the general prototype system model in figure 3.2, the prototype shows a diagram of the environment in which the prototype system operates. If the prototype is effective, it can be embedded into the actual alarm system.

For the prototype system, it is important that it sends out alert message in time when the alarm sounds off. Some vehicles' information is therefore stored in the database that includes their owners' mobile phone number and locations they are at. An alarm program is designed within the PC to simulate the alarm system inside vehicles that triggers an alarm randomly to any vehicle in the database. Within the central computer system, the system server is responsible for sending out the SMS alert message to the correct owner. The message has to be sent first to the GSM phone connected to the system server through the GSM data cable. The GSM phone then sends the message through the GSM network where it looks for the correct owner to receive the message. The message receives by the owner includes the license plate number of the vehicle and additional information like the location of the vehicle.

From there, the owner decides if he/she wants to forward the message to the police. In real situation, some alarm systems go off accidentally due to faults and therefore it may not be advisable to alert the police immediately. It may be better if the owner checks out his/her vehicle to see if there is a need to alert the police by forwarding the message. The location of the vehicle is already included in the message; therefore the police will have sufficient time to reach the location.

The prototype system hardware is the same for all prototypes discussed in the project. It is described in Chapter 3.

4.6.4 Prototype Software Developments and Testing

The system software is composed of a number of different modular blocks as shown in figure 3.1.

4.6.4.1 Vehicle Burglar Alarm Database Module

The vehicles' information database contains vehicles' license plate numbers, the owners' mobile phone number, and their location. In this prototype, fixed locations are stored in the database. For the real case, the system should be designed to record real dynamic information of the vehicles' whereabouts. One way to achieve this is that the motorist sends the location of the vehicle to the system whenever he/she parks his/her vehicle. It is simpler if the vehicle is fitted with a GPS. Then the location of the vehicle can be automatically recorded in the system wherever it is. For the prototype, the alarm simulation randomly triggers an alarm in a vehicle. When the prototype security alert is activated from a vehicle, the database module searches for the mobile phone number and also the location where it parks. The database module then interacts with the software and server module to send out the alert message to the owner.

4.6.4.2 Vehicle Burglar Alarm Software and Server Module

The software and server module is responsible for sending out alert message to the owner. When the alarm system within the PC triggers an alarm on a vehicle, the license plate number of the vehicle is sent to the database module. The database module then collects all the required information about the vehicle and sends it to the software and server module. It then sorts out the information into text message and sends it to the owner. When necessary, the owner forwards the message to the police.

The message received by the police is more detailed. The differences between the two messages are as follows:

- **Vehicle security alert to the owner:** This message is simple, as the owner already knew where he/she had parked his/her vehicle.

```
Security alert!!!  
SCW1234H
```

Alert message sent to the motorist by the system

Fig. 4.18 – Vehicle Security Alert to Motorist

- **Vehicle security alert to the police:** Relevant information is needed for the police to track down the vehicle. Therefore the location of the vehicle is sent to the police.

```
Security alert !!!  
SCW1234H  
Jurong east ave 1  
Blk 13 car park
```

Alert message sent to Traffic Police by the system

Fig. 4.19 – Vehicle Security Alert to Police

4.7 Singapore Street Directory System

The Singapore Street Directory [6,12] basically provides Singapore maps information in a more detailed manner. It is helpful when people need to find out about places. It is available as a book in most of the bookshops and also on the Internet. Most buyers are motorists where they can refer to it when they need to travel to unfamiliar areas.

It is inconvenient for motorists to constantly flip through pages for the road information while driving. Considering the weight of the book, it is not possible for people to carry it along while going out. It is also time consuming while searching for road information in the book.

4.7.1 SMS Implementations and its Objectives

Although SMS technology cannot provide the visual information like the street directory, it is able to provide sufficient information in the form of text messages. People who use the street directory usually need to find out about how to get to places from where they are. This can be done very effectively using SMS.

Users need to send a SMS to the system about their recent location and their destination. The system then replies to them with the route information they are supposed to take. The path information includes important road names that the users can look out for while they are traveling. For example, a motorist while driving is able to refer to the mobile phone for the road names he/she is supposed to look out for. Additionally, the message includes directions like turn left or right.

The system helps to cut down search time for road information and reduces the inconvenience and danger that motorists experience in flipping through pages of the street directory while driving.

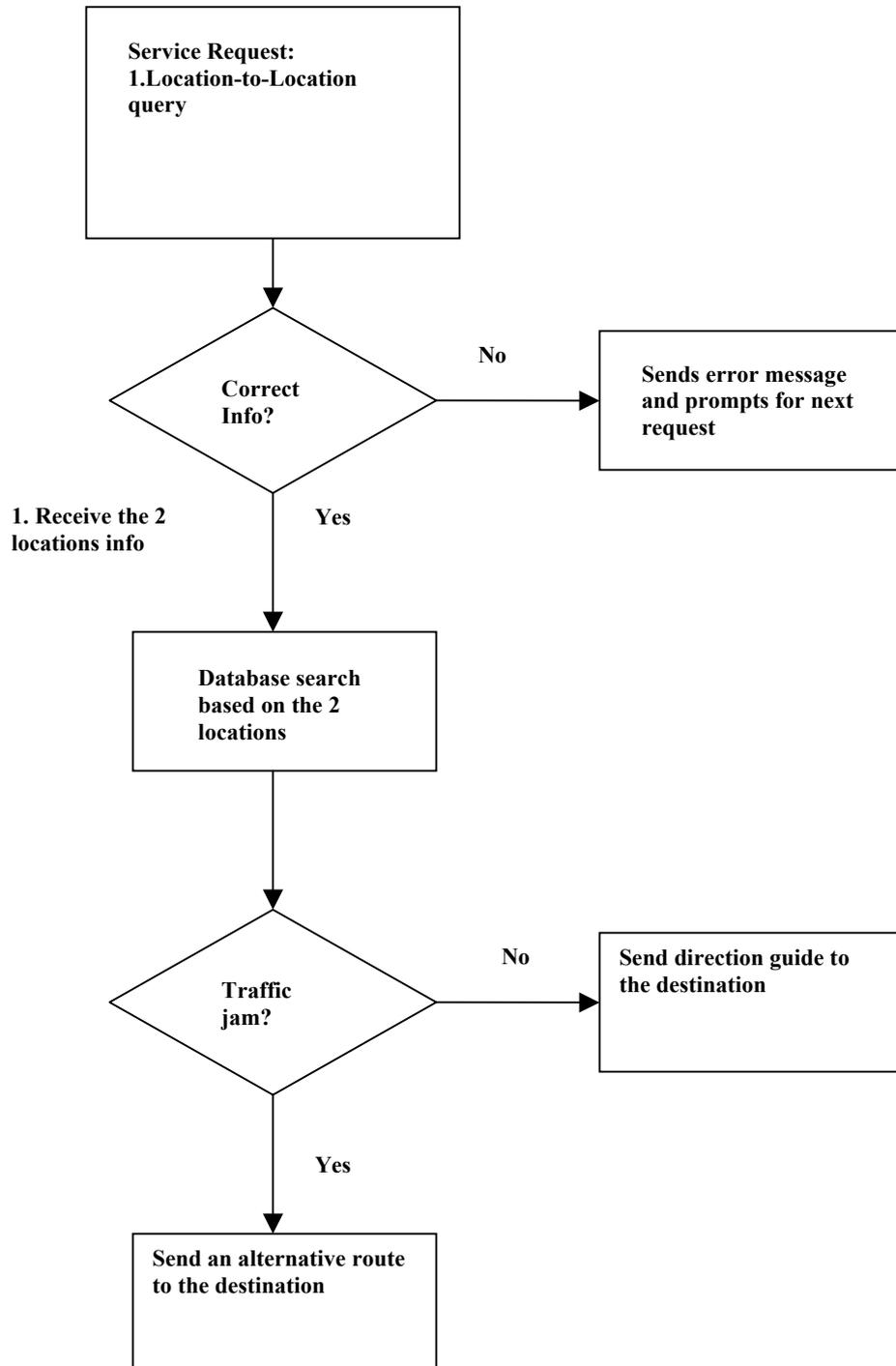
The way that the system functions is as follows:

- 1) It takes down the information in the received SMS, the current location, and the destination. It looks in its database for the possible roads depending on what the user sends. For example, the user may just send the destination information as Clementi, but there are many roads that start with Clementi like Clementi Rd, Clementi Ave 3, Clementi Ave 5, etc.
- 2) For the example of Clementi, the system is designed in a way that it takes note of all the roads that start with Clementi. In the road database, all the roads are separated into different categories: express ways, major roads, minor roads, streets, avenues, etc.
- 3) As the user has just sent the destination information as Clementi, it takes it as the most major road, that is, Clementi Rd. It is the road that links to small roads, for example, Clementi Ave 3, Clementi Ave 5, etc.
- 4) It formulates a shortest possible route to take by looking into the roads that link both locations.
- 5) When there is a traffic jam due to accidents, congestions, constructions, etc, along the route, it informs the user and suggests alternate routes if possible.

A prototype is developed to test out how efficient and user-friendly the SMS street directory proves to be. The main aim is to provide a clear and understandable travel guide in the SMS message.

4.7.2 Functional Flow Block Diagram

Street Directory Service Request



4.7.3 Prototype System Overview

Referring on the general prototype system model in figure 3.2, the central computer system in the prototype system consists of the system server, GSM data cable and the GSM phone connected to the server via the cable. Some of the road information is stored in the prototype database. They are chosen in such a way that they be linked. User is needed to send a SMS message of two road names to the system server via the GSM network. Based on the request, the prototype tries to search for roads that link both locations. The result of the road names is sent to the user's mobile phone via SMS. The prototype arranges the roads in an order that the user can use them to navigate himself/herself to his/her destination.

The prototype system hardware is the same for all prototypes discussed in the project. It is described in Chapter 3.

4.7.4 Prototype Software Developments and Testing

The system software is composed of a number of different modular blocks as shown in figure 3.1.

4.7.4.1 Street Directory Database Module

In the street directory database, the roads are classified into different categories, based on the number of links to other roads. For every road, roads that are linked to it are also recorded. For the prototype, there are three categories named as major, middle, and minor roads. Major roads have more than ten road links; middle roads have between four to ten road links, and minor roads have less than four road links. In the prototype system, the database module is responsible for taking down the two road names from the software module. The software module is responsible in breaking

down the SMS message from user into two relevant road names. It then searches through the database and records down all the available road links that connect the two roads. However, the prototype needs to organize the road links in such a way that it is navigation friendly. It has to ensure that when a user goes from one location to another, he/she makes use of the results received from the system to reach his/her destination. Simple directions are provided in the results.

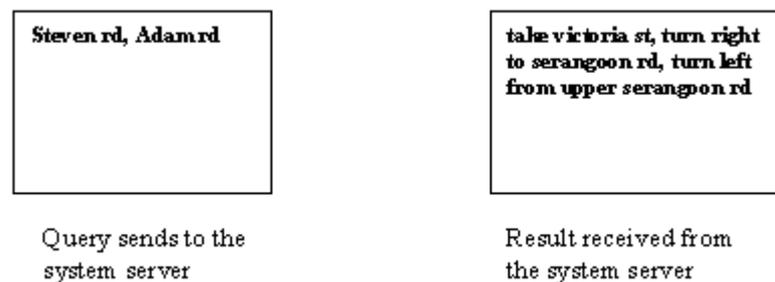


Fig. 4.20 – Location-to-Location Query for Singapore Street Directory System

4.7.4.2 Street Directory Software and Server Module

The software and server module are responsible for identifying and analyzing the text message received from users. Based on the example of Figure 9.3, the software module collects the SMS message from the user and breaks it down into relevant information. As for this case, the module identifies that there are two locations and it sends these locations to the database module where it searches out the list of road links and directions. The database module then arranges and sends the result to the software module. The software module sends the results back to the user who requests for the information. Therefore, it is important that the software module takes note of the user's mobile number and his/her SMS request; making sure that it doesn't send the result to the wrong person. In case when there is a traffic jam or a road is closed, the prototype updates itself. The server module is used to update any changes with the database module. When a road is having a traffic jam, the server module sends the information

to the database module where it is updated in the database. The database module then tries to search for an alternative route to the destination and the result is sent to the software module, which in turns sends it to the user's mobile phone via SMS.

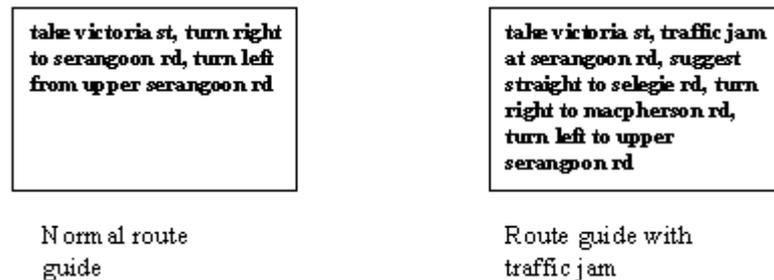


Fig. 4.21 – Location-to-Location Query in The Case of Traffic Jam

The prototype is developed and is tested out. It is found that some results need more text spaces as the roads and directions sent to users exceeded the 160 text spaces of the mobile phone. This problem is solved by splitting the message into several messages that are less than 160 text spaces. The system sends these consecutively to users.

4.8 Comparison Between Applications

4.8.1 Similarities

In this project, all applications make use of pull technologies. Pull technologies are retrieval mechanisms to search for information once information needs are specified. A limitation of pull technology is that users must know what their information needs are, and they must know how to specify their information needs in a manner that will achieve the desired results. Error messages are sent to users if they accidentally key in the wrong search words. Information access from all of them characterizes situations where users actively search for information (either in the form of information retrieval or information exploration).

A real time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation but also upon the time at which the result is produced. If the timing constraints of the system are not met, system failure is said to have occurred. Real time system can be seen in all applications, and it is an important feature in this project as real-time availability provides up-to-date information that is essential for users.

These seven applications make use of a similar system model which consists of the central computer system, the GSM network and the mobile terminals, and the mean of communication between them is the short message service (SMS). They also utilize a similar system software structure which consists of four modules: server, static, dynamic, and database.

4.8.2 Differences

Although these seven applications use the pull technology for information retrieval, their search mechanisms are different from one another. Every search query in each application is therefore unique and has its own purpose; road names in the bus guide application are essential for users who need to know directions traveling by bus, vehicle license plate numbers are needed by traffic attendants to issue illegal parking fines to motorists, etc.

Real time systems are more dynamic in the bus arrival alert application and the MRT arrival application as accuracy in time is important for users to alight at the right stop or board the right MRT train in order to reach the airport in time. Some other applications are less active in SMS results reporting, as they need time in processing the results. For instance, the car parking application has to be precise in displaying the cost of the parking fees for motorists to prevent confusion that may lead to complaints

from them. And in the illegal parking application, correct amount of fine has to be sent to the motorist's mobile phone in order to avoid wrong bill charges to them.

Some applications also need users to register their information into database before they can use them. For example, users need to register their vehicle license plate numbers in the vehicle security application in order for them to be alert by the application if there is any theft from their vehicles.

5 CONCLUSIONS AND FUTURE WORKS

This chapter concludes on all prototypes done in this project and how SMS is able to contribute to the public transport system in Singapore. Following this, future works are recommended.

5.1 Conclusions

The project starts with the understanding of how SMS works and contributes in real life. As it is relatively cheaper than WAP and GPRS services and popular among mobile phone users, it is chosen to provide applications for all prototypes designed in the project. At the same time, the project looks into the public transport system in Singapore and analyzes to see if the system can benefit from these wireless applications. Therefore, it is motivating to integrate this simple wireless technology with the public transport system. And for some of the prototypes developed, SMS is not only used for providing information; it is also able to provide more advanced services like billing and booking.

In providing information, SMS has shown promising results when prototypes are tested. They are fast and efficient, and most importantly, the information received by users is accurate and comprehensive despite the fact that text spaces per SMS message are limited to 160. For instance, the travel service guide developed in the public service transport prototype is successful in providing point-to-point travel solutions in one SMS message. Additionally, some of the prototypes are required to provide interactive and more advanced information. In the bus-reaching alert of the public service transport prototype, user is able to request for the prototype to alert them when he/she is arriving at a particular bus stop. The SMRT and Changi Airport prototype is also able to plan out a route for user to get to the airport in time for departure if he/she

sends a SMS message of the departure time of a flight and the MRT station where he/she is going to take the train.

More advanced features of the project are considered and SMS is applied for some prototypes to provide billing and booking services. In the car parking prototype, users are able to make use of SMS to pay their parking fees and when illegal parking happened, traffic attendant is able to book the vehicle efficiently with SMS and any fines accumulated is added to the user's mobile phone bill. In the vehicle burglar alarm prototype, the project further showed potential of SMS in the security issue. It is able to send out SMS alarm fast to the car owner's mobile phone when the alarm system in the vehicle sounds off.

In conclusion, the integration of wireless technology with the public transport system shows a great potential and benefits to mobile phone users. Furthermore SMS applications prove that there is little or no need for complicated wireless technology to fulfill normal services provided by the public transport system. Thus SMS may still be considered as a basis for future wireless designs.

5.2 Recommendations for Future Works

Further testing can be done using a GSM modem instead of the GSM data cable to evaluate how well the prototypes developed can handle multiple messages from users. Especially, database modules in the software design of the prototypes are tested to see if they are able to manage records accurately and in a timely manner when multiple database requests happen. Some of the prototypes have features whereby outdoor monitoring is necessary. For instance, the bus arrivals feature developed in the public service transport prototype will be more accurate in reporting arrival times of buses if testing is done with the use of AVL to monitor buses.

Some of the prototypes can be enhanced using Multimedia messaging service (MMS). MMS [5] is a logical extension of SMS and it encompasses a wide range of content types, making it easily adoptable for today's generation of mobile users. Just as the SMS, MMS provides automatic and immediate delivery of personal messages. Unlike the SMS however, MMS allows mobile phone users to enhance their messages by incorporating sound, images, and other rich content, transforming it into a personalized visual and audio message. Another advantage of MMS is that the message is a multimedia presentation in a single entry, not a text file with attachments, making it much simpler and user-friendly.

With MMS, graphical guide will be able to help user to visualize the information better. The graphical advantages of MMS are as follow:

- It can help to direct users to the nearest bus stops in the public service transport prototype.
- It can provide views of car park locations to motorists near their areas in the car parking prototype.
- It can provide graphical travel solutions to users in the street directory prototype.
- If web cam can be integrated with the burglar alarm system, it can take pictures of the surrounding or important spots and send them via MMS.

The MMS standard lists JPEG, GIF, text, AMR voice, and other formats as supported media types, while unsupported formats are handled in a controlled way. Like SMS, MMS is an open industry standard, and MMS messages can be delivered using existing networks and protocols. MMS is also bearer-independent, which means it is not limited to GSM or WCDMA networks.

REFERENCES

- [1] <http://www.sbstransit.com.sg/>
- [2] <http://www.smrtcorp.com/smrt/index.htm>
- [3] <http://www.tibs.com.sg/>
- [4] <http://www.transitlink.com.sg/index.html>
- [5] <http://www.forum.nokia.com/main/>
- [6] http://www.streetdirectory.com/new_sd_member_no.cgi?home=1
- [7] <http://www.lta.gov.sg/MenuFrame2.htm>
- [8] <http://www.changi.airport.com.sg/>
- [9] <http://www.aas.com.sg/features/archive/otr07011.htm>
- [10] http://www.sbstransit.com.sg/whatsnew_20may2002.asp
- [11] Giovanni Martini, Giorgio Rosenga, “Distributed Architecture for Applications based on the GSM Short Message Service” in IEEE, 0-8186-7092-4/95, 1995
- [12] Mighty Minds Publishing Pte Ltd, “Singapore Street Directory”, 2002/2003 Edition
- [13] Jan A. Audestad. Network aspects of the GSM system. In *EUROCON 88*, June 1988.
- [14] D. M. Balston. The pan-European system: GSM. In D. M. Balston and R.C.V. Macario, editors, *Cellular Radio Systems*. Artech House, Boston, 1993.
- [15] David M. Balston. The pan-European cellular technology. In R.C.V. Macario, editor, *Personal and Mobile Radio Systems*. Peter Peregrinus, London, 1991.
- [16] C. Déchaux and R. Scheller. What are GSM and DCS. *Electrical Communication*, 2nd Quarter 1993.

- [17] Michel Mouly and Marie-Bernadette Pautet. *The GSM System for Mobile Communications*. Published by the authors, 1992.
- [18] Torbjorn Nilsson. Toward a new era in mobile communications. <http://193.78.100.33/> (Ericsson WWW server).
- [19] Moe Rahnema. Overview of the GSM system and protocol architecture. *IEEE Communications Magazine*, April 1993.
- [20] C. B. Southcott et al. Voice control of the pan-European digital mobile radio system. In *IEEE GLOBECOM 1989*, November 1989.

APPENDICES

Appendix A: Revised Parking Rates

Coupon Parking Rates For Cars

Type of Charges	Current	Revised
Half-Hourly Parking Charges		
a. Day Parking on Weekdays		
»RZ	\$0.90	\$1.00
»ORZ	\$0.45	\$0.50
b. Night Parking on Weekdays	\$0.45	\$0.50
c. Day and Night Parking on Sundays & Public Holidays	\$0.45	\$0.50
Whole-Day Coupon Parking Charges		
a. Park and Ride Schemes	\$ 2.70	\$ 3.00
(7am to 9pm on Mon to Fri & 7am to 3pm on Sat)		
b. Whole Day parking in selected car parks	\$ 3.60	\$ 4.00
(Pasir Ris Close, Maritime Square D and Seah Im Road)		
c. Whole Day Parking Coupons		
»Islandwide	\$15.00	\$16.50
»ORZ only	\$ 9.00	\$10.00
Monthly Parking Coupons		
»Islandwide	\$420.00	\$460.00
»ORZ only	\$270.00	\$300.00
Night Parking Coupons	\$2.00	\$2.00 (Unchanged)

Coupon Parking Rates For Heavy Vehicles

Type of Charges	Current	Revised
Half-Hourly Parking Charges		
a. Day Parking on Weekdays		
»RZ	\$1.80	\$2.00
»ORZ	\$0.90	\$1.00
b. Night Parking on Weekdays	\$0.90	\$1.00
c. Day and Night Parking on Sundays & Public Holidays	\$0.90	\$1.00

Coupon Parking Rates For Motor Cycles

Type of Charges	Current	Revised
a. Whole day Parking Charges		
»Day and Night parking	\$0.60	\$0.65
»Overnight parking	\$0.60	\$0.65

Appendix B: Source Codes

VERSION 5.00

Begin VB.Form BusGuide

Caption = "BusGuide"
ClientHeight = 5385
ClientLeft = 5175
ClientTop = 2835
ClientWidth = 5640
LinkTopic = "Form1"
ScaleHeight = 5385
ScaleWidth = 5640

Begin VB.CommandButton Command1

Caption = "Command1"
Height = 375
Left = 3480
TabIndex = 13
Top = 3720
Width = 1455

End

Begin VB.CommandButton Close

Caption = "Close"
Height = 375
Left = 3600
TabIndex = 5
Top = 4560
Width = 975

End

Begin VB.TextBox SMSCAddress

Height = 285
Left = 360
TabIndex = 4
ToolTipText = "Short message centre address"
Top = 1440
Width = 2535

End

Begin VB.TextBox OtherEndAddress

Height = 285
Left = 360
TabIndex = 3
Top = 840
Width = 2535

End

Begin VB.TextBox SMSText

Height = 1455
Left = 360
MaxLength = 160
MultiLine = -1 'True'
TabIndex = 2
ToolTipText = "Write the message here (max 160 characters)"
Top = 2040
Width = 3495

End

Begin VB.TextBox Hits

Height = 375
Left = 3840
TabIndex = 1
Top = 840
Width = 735

End

Begin VB.TextBox Text1

Height = 375
Left = 360
TabIndex = 0
Top = 3720
Width = 2775

End

Begin VB.Timer Timer1

Interval = 1000
Left = 4320
Top = 1920

```

End
Begin VB.Label Label4
    Caption    = "/ 160"
    Height    = 255
    Left      = 3480
    TabIndex  = 12
    Top      = 1800
    Width    = 495
End
Begin VB.Label Length
    Caption    = "0"
    Height    = 255
    Left      = 3120
    TabIndex  = 11
    Top      = 1800
    Width    = 375
End
Begin VB.Label MessageStatus
    Height    = 255
    Left      = 1680
    TabIndex  = 10
    Top      = 240
    Width    = 2295
End
Begin VB.Label Label3
    Caption    = "Message status:"
    Height    = 255
    Left      = 360
    TabIndex  = 9
    Top      = 240
    Width    = 1215
End
Begin VB.Label Label2
    Caption    = "SMSC Address"
    Height    = 255
    Left      = 360
    TabIndex  = 8
    Top      = 1200
    Width    = 1575
End
Begin VB.Label FromTo
    Caption    = "From"
    Height    = 255
    Left      = 360
    TabIndex  = 7
    Top      = 600
    Width    = 1935
End
Begin VB.Label Label1
    Caption    = "Message"
    Height    = 255
    Left      = 360
    TabIndex  = 6
    Top      = 1800
    Width    = 735
End
End
Attribute VB_Name = "BusGuide"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private WithEvents puISink As SMS3ASuiteLib.SMS_SuiteAdapter
Attribute puISink.VB_VarHelpID = -1

Public puSMSSuiteAdapter As SMS3ASuiteLib.SMS_SuiteAdapter
Public puSMSSMemory As SMS3ASuiteLib.ISMSMemory
Public puSMSSettings As SMS3ASuiteLib.ISMSSettings

Public puSMSSMemory As SMS3ASuiteLib.SMS_MEMORY_LOCATION
Public puNewMessage As Boolean, puReceivedMessage As Boolean

Private prIShortMessage As SMS3ASuiteLib.ShortMessage
Public puMemType As SMS3ASuiteLib.SMS_MEMORY_LOCATION

```



```

Public puMemIndex As Long
Public puShowMemParameters As Boolean
Public puCancel As Boolean

Dim BusService As ADODB.Connection
Dim ConnRoad As ADODB.Connection
Dim ConnBus As ADODB.Connection
Dim ConnSmrt As ADODB.Connection
Dim ConnService As ADODB.Connection
Dim ConnService1 As ADODB.Connection
Dim ConnService2 As ADODB.Connection

Dim ListRecordset As ADODB.Recordset
Dim ServiceRecordset As ADODB.Recordset
Dim NumRecordset As ADODB.Recordset
Dim StopRecordset As ADODB.Recordset
Dim RoadRecordset As ADODB.Recordset
Dim MatchRecordset As ADODB.Recordset
Dim SerNoRecordset As ADODB.Recordset
Dim FromRecordset As ADODB.Recordset
Dim ToRecordset As ADODB.Recordset
Dim TransRecordset As ADODB.Recordset
Dim LoopListRecordset As ADODB.Recordset
Dim DualListRecordset As ADODB.Recordset
Dim SmrtRecordset As ADODB.Recordset

Dim Word, Char, SortWord, Topstr, ServiceChar, SerNoChar, LoopListChar, DualListChar
Dim Pos, Ticker, i, j, k, l, m

Dim Serno, ServiceNo, ServiceNodir1, ServiceNodir2
Dim GotService, CheckDual, DataNum, MiddleDir, QuartDir
Dim GotBusStop, BusStopNo
Dim GotRoad, RoadInfo, CheckRoad, RoadWord, PassFrom, PassTo, CheckStart, CheckSerNo
Dim MiddleRd, QuartRd, StartRd, EndRd, CheckRoadPass
Dim MatchCount, FromCount, ToCount
Dim CheckLink, CheckTransfer
Dim CheckSmrtWord, SmrtService, FromMrt, ToMrt, MrtWord, CheckMrtName, GotMrt
Dim FromMrtSpot, ToMrtSpot, FromMrtRoad, ToMrtRoad
Dim MrtText

Dim FromRoad(10), ToRoad(10), Service(10), FromService(10), ToService(10), FromRegion(10), ToRegion(10)
Dim TransRoad(10), FromTrans(10), ToTrans(10), TransServiceRoad(50)
Dim ToDir, OppToDir, OppStopNo, StopNo, OppStopDir, StopDir
Dim FromCharService, ToCharService, TransCharService, FromCharSer, ToCharSer
Dim CheckFromService, CheckToService, CheckFromDual, CheckToDual, CheckRegionMatch

Private Sub Close_Click()

    Unload BusGuide

End Sub

Private Sub Command1_Click()
    Call TextSort
End Sub

Private Sub Form_Load()
    Set puSMSSuiteAdapter = New SMS3ASuiteLib.SMS_SuiteAdapter
    Set puSMSSettings = puSMSSuiteAdapter
    Set puSMSMemory = puSMSSuiteAdapter
    Set puSink = puSMSSuiteAdapter
    Call puSMSSuiteAdapter.StartListeningEvents
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub
Private Sub puSink_ShortMsgReceived(ByVal SMSMemory As SMS3ASuiteLib.SMS_MEMORY_LOCATION, ByVal Index As Long, ByVal pSMS As SMS3ASuiteLib.IShortMessage)

    puSMSMemory = SMSMemory

    puNewMessage = False

    Call ShowSMS(pSMS)
End Sub

```

```

Public Sub ShowSMS(ByVal cSMS As SMS3ASuiteLib.ShortMessage)

    Set prlShortMessage = cSMS

    ' if this is received message
    If puNewMessage = False Then

        'display short message
        Text1.Text = cSMS.UserDataText

        'sender address
        OtherEndAddress.Text = cSMS.OtherEndAddress

    'if this is a message to be sent or stored into memory

    End If

    'SMSC address
    SMSCAddress.Text = cSMS.SCAddress

    'storage status
    Dim StorageStatus As SMS3ASuiteLib.STORAGE_STATUS

    StorageStatus = cSMS.StorageStatus

    If StorageStatus = NOT_SENT_FROM_PHONE Then
        MessageStatus.Caption = "Not sent."
    ElseIf StorageStatus = SENT_FROM_PHONE Then
        MessageStatus.Caption = "Sent."
    ElseIf StorageStatus = DELIVERED Then
        MessageStatus.Caption = "Delivered."
    ElseIf StorageStatus = READ_FROM_PHONE Then
        MessageStatus.Caption = "Read."
    ElseIf StorageStatus = NOT_READ_FROM_PHONE Then
        MessageStatus.Caption = "Not read."
    End If

    Set cSMS = Nothing

End Sub
Private Sub TextSort()
    CheckRegionMatch = 0
    CheckSmrtWord = 0
    CheckMrtName = ""
    MiddleRd = ""
    QuartRd = ""
    QuartDir = ""
    MiddleDir = ""
    GotService = ""
    GotBusStop = ""
    GotMrt = ""
    GotRoad = ""
    CheckDual = ""
    CheckStart = 0
    CheckSerNo = 0
    CheckRoadPass = 0
    RoadWord = ""
    FromCount = 0
    MatchCount = 0
    PassFrom = 0
    PassTo = 0
    Word = ""
    ServiceChar = ""
    Char = ""
    Topstr = ""
    FromMrt = ""
    ToMrt = ""

    SortWord = Text1.Text
    For i = 0 To 10
        FromRoad(i) = ""
        ToRoad(i) = ""
        FromRegion(i) = ""
        ToRegion(i) = ""
    Next

```

```

If Mid(SortWord, Len(SortWord), 1) = " " Then
  Do Until Mid(SortWord, Len(SortWord), 1) <> " "
    SortWord = Mid(SortWord, 1, Len(SortWord) - 1)
  Loop
End If
Ticker = 0
Pos = 1
SortWord = SortWord & " "

For i = 1 To Len(SortWord)
Char = Mid(SortWord, i, 1)
  For 1st word
    If Char = " " And Ticker = 0 Then
      Word = Mid(SortWord, Pos, i - 1)

      If IsNumeric(Word) = True Then

        Set BusService = New ADODB.Connection
        BusService.Open "busdir"

        Set ServiceRecordset = New ADODB.Recordset
        ServiceRecordset.Open "SELECT * FROM busdir", BusService

        Set ConnRoad = New ADODB.Connection
        ConnRoad.Open "road"

        Set StopRecordset = New ADODB.Recordset
        StopRecordset.Open "SELECT * FROM road", ConnRoad

        Do Until ServiceRecordset.EOF
          If Word = ServiceRecordset("service no") Then
            GotService = Word
            CheckDual = ServiceRecordset("dual direction")
            Exit Do
          End If
          ServiceRecordset.MoveNext
        Loop

        Do Until StopRecordset.EOF

          If Word = StopRecordset("bus stop no") Then
            GotBusStop = Word
            BusStopNo = Word
            FromRoad(0) = StopRecordset("road name")
            FromRegion(0) = StopRecordset("region")
            Exit Do
          End If

          StopRecordset.MoveNext
        Loop

        If GotService <> "" Then

          ServiceNo = Word
          Serno = Word
          Call RoadListText
          Exit Sub

        End If

      ElseIf IsNumeric(Word) = False Then

        RoadWord = Word
        If RoadWord <> "mrt" Then

          Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
          'look at each char in text string
          Topstr = Topstr & "ORDER BY [road name]"

          Set ConnRoad = New ADODB.Connection
          ConnRoad.Open "road"

          Set RoadRecordset = ConnRoad.Execute(Topstr)

```

```

Do Until RoadRecordset.EOF

    If FromRoad(0) = "" Then
        FromRoad(0) = RoadRecordset("road name")
        FromRegion(0) = RoadRecordset("region")
        GotRoad = RoadRecordset("road name")
        FromCount = 1
    End If
    CheckRoadPass = 1
    For j = 0 To 10

        If FromRoad(j) <> "" Then
            If FromRoad(j) = RoadRecordset("road name") Then
                CheckRoadPass = 0
                Exit For
            End If
        End If

    Next
    If CheckRoadPass = 1 Then
        For j = 0 To 10
            If FromRoad(j) = "" Then
                FromRoad(j) = RoadRecordset("road name")
                FromRegion(j) = RoadRecordset("region")
                FromCount = FromCount + 1
                Exit For
            End If
        Next
    End If

    RoadRecordset.MoveNext
    Loop
    ElseIf RoadWord = "mrt" Then

        CheckSmrtWord = 1

    End If

End If
Pos = i
Ticker = 1

'for middle words
Elseif Char = " " And Ticker = 1 And i <> Len(SortWord) Then

    Word = Mid(SortWord, Pos + 1, i - Pos - 1)

    If CheckSmrtWord = 0 Then
        'first word is bus stop number
        If GotBusStop <> "" And PassTo = 0 Then
            RoadWord = Word

            Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%'"
            'look at each char in text string
            Topstr = Topstr & "ORDER BY [road name]"

            Set ConnRoad = New ADODB.Connection
            ConnRoad.Open "road"

            Set RoadRecordset = ConnRoad.Execute(Topstr)
            Do Until RoadRecordset.EOF
                If ToRoad(0) = "" Then
                    ToRoad(0) = RoadRecordset("road name")
                    ToRegion(0) = RoadRecordset("region")
                    ToCount = 1
                End If
                CheckRoadPass = 1
                For j = 0 To 10

                    If ToRoad(j) <> "" Then
                        If ToRoad(j) = RoadRecordset("road name") Then
                            CheckRoadPass = 0
                            Exit For
                        End If
                    End If
                Next
            End If
        End If
    End If

```

```

        End If

    Next
    If CheckRoadPass = 1 Then
        For j = 0 To 10
            If ToRoad(j) = "" Then
                ToRoad(j) = RoadRecordset("road name")
                ToRegion(j) = RoadRecordset("region")
                ToCount = ToCount + 1
            Exit For
            End If
        Next
    End If

    RoadRecordset.MoveNext
    Loop
    PassTo = 1

ElseIf GotBusStop <> "" And PassTo = 1 Then

    CheckRoad = ""
    RoadWord = RoadWord & " " & Word
    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set ConnRoad = New ADODB.Connection
    ConnRoad.Open "road"

    Set MatchRecordset = ConnRoad.Execute(Topstr)
    Do Until MatchRecordset.EOF

        If CheckRoad = "" Or CheckRoad <> MatchRecordset("road name") Then
            If CheckRoad = "" Then
                CheckRoad = MatchRecordset("road name")
                MatchCount = 1
            End If
            If CheckRoad <> MatchRecordset("road name") Then

                For j = 0 To 10

                    If ToRoad(j) = CheckRoad Then
                        MatchCount = MatchCount + 1
                    End If

                Next
                CheckRoad = MatchRecordset("road name")

            End If

        End If

    MatchRecordset.MoveNext
    Loop

    If MatchCount <> ToCount And MatchCount <> 0 Then
        For j = 0 To 10
            ToRoad(j) = ""
        Next
        ToCount = 0
        Set RoadRecordset = ConnRoad.Execute(Topstr)
        Do Until RoadRecordset.EOF

            If ToRoad(0) = "" Then
                ToRoad(0) = RoadRecordset("road name")
                ToRegion(0) = RoadRecordset("region")
                ToCount = 1
            End If
            CheckRoadPass = 1
            For j = 0 To 10

                If ToRoad(j) <> "" Then
                    If ToRoad(j) = RoadRecordset("road name") Then
                        CheckRoadPass = 0
                    Exit For
                End If
            Next
        Next
    End If

```

```

        End If
    End If

    Next
    If CheckRoadPass = 1 Then
        For j = 0 To 10
            If ToRoad(j) = "" Then
                ToRoad(j) = RoadRecordset("road name")
                ToRegion(j) = RoadRecordset("region")
                ToCount = ToCount + 1
            Exit For
        End If
    Next
    End If

    RoadRecordset.MoveNext
    Loop
End If

'first word is road name
ElseIf GotRoad <> "" And PassFrom = 0 Then
    CheckRoad = ""
    RoadWord = RoadWord & " " & Word
    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set MatchRecordset = ConnRoad.Execute(Topstr)
    Do Until MatchRecordset.EOF

        If CheckRoad = "" Then
            CheckRoad = MatchRecordset("road name")
            MatchCount = 1
        End If
        If CheckRoad <> MatchRecordset("road name") Then

            For j = 0 To 10

                If FromRoad(j) = CheckRoad Then
                    MatchCount = MatchCount + 1
                End If

            Next
            CheckRoad = MatchRecordset("road name")

        End If

        MatchRecordset.MoveNext
    Loop

    If MatchCount <> FromCount And MatchCount <> 0 Then
        For j = 0 To 10
            FromRoad(j) = ""
        Next
        FromCount = 0
        Set RoadRecordset = ConnRoad.Execute(Topstr)
        Do Until RoadRecordset.EOF

            If FromRoad(0) = "" Then
                FromRoad(0) = RoadRecordset("road name")
                FromRegion(0) = RoadRecordset("region")
                GotRoad = RoadRecordset("road name")
                FromCount = 1
            End If
            CheckRoadPass = 1
            For j = 0 To 10

                If FromRoad(j) <> "" Then
                    If FromRoad(j) = RoadRecordset("road name") Then
                        CheckRoadPass = 0
                        Exit For
                    End If
                End If
            Next
            If CheckRoadPass = 1 Then

```

```

        For j = 0 To 10
            If FromRoad(j) = "" Then
                FromRoad(j) = RoadRecordset("road name")
                FromRegion(j) = RoadRecordset("region")
                FromCount = FromCount + 1
            Exit For
        End If
    Next
End If
RoadRecordset.MoveNext
Loop
End If

ElseIf GotRoad <> "" And PassFrom = 1 And PassTo = 0 Then

    RoadWord = RoadWord & " " & Word

    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set ConnRoad = New ADODB.Connection
    ConnRoad.Open "road"

    Set RoadRecordset = ConnRoad.Execute(Topstr)
    Do Until RoadRecordset.EOF
        If ToRoad(0) = "" Then
            ToRoad(0) = RoadRecordset("road name")
            ToRegion(0) = RoadRecordset("region")
            GotRoad = RoadRecordset("road name")
            ToCount = 1
        End If
        CheckRoadPass = 1
        For j = 0 To 10

            If ToRoad(j) <> "" Then
                If ToRoad(j) = RoadRecordset("road name") Then
                    CheckRoadPass = 0
                    Exit For
                End If
            End If
        Next

        If CheckRoadPass = 1 Then
            For j = 0 To 10
                If ToRoad(j) = "" Then
                    ToRoad(j) = RoadRecordset("road name")
                    ToRegion(j) = RoadRecordset("region")
                    ToCount = ToCount + 1
                Exit For
            End If
        End If
    Next
    End If

    RoadRecordset.MoveNext
    Loop
    PassTo = 1

ElseIf GotRoad <> "" And PassFrom = 1 And PassTo = 1 Then
    CheckRoad = ""
    RoadWord = RoadWord & " " & Word
    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set ConnRoad = New ADODB.Connection
    ConnRoad.Open "road"

    Set MatchRecordset = ConnRoad.Execute(Topstr)
    Do Until MatchRecordset.EOF

        If CheckRoad = "" Or CheckRoad <> MatchRecordset("road name") Then
            If CheckRoad = "" Then
                CheckRoad = MatchRecordset("road name")
                MatchCount = 1
            End If
        End If
    End If

```

```

End If
If CheckRoad <> MatchRecordset("road name") Then

    For j = 0 To 10

        If ToRoad(j) = CheckRoad Then
            MatchCount = MatchCount + 1
        End If

    Next
    CheckRoad = MatchRecordset("road name")

End If

End If

MatchRecordset.MoveNext
Loop

If MatchCount <> ToCount And MatchCount <> 0 Then
    For j = 0 To 10
        ToRoad(j) = ""
    Next
    ToCount = 0
    Set RoadRecordset = ConnRoad.Execute(Topstr)
    Do Until RoadRecordset.EOF

        If ToRoad(0) = "" Then
            ToRoad(0) = RoadRecordset("road name")
            ToRegion(0) = RoadRecordset("region")
            GotRoad = RoadRecordset("road name")
            ToCount = 1
        End If
        CheckRoadPass = 1
        For j = 0 To 10

            If ToRoad(j) <> "" Then
                If ToRoad(j) = RoadRecordset("road name") Then
                    CheckRoadPass = 0
                    Exit For
                End If
            End If

        Next
        If CheckRoadPass = 1 Then
            For j = 0 To 10
                If ToRoad(j) = "" Then
                    ToRoad(j) = RoadRecordset("road name")
                    ToRegion(j) = RoadRecordset("region")
                    ToCount = ToCount + 1
                End If
            Next
        End If

        RoadRecordset.MoveNext
    Loop
End If

End If

If MatchCount = 0 And PassFrom = 0 Then
    PassFrom = 1
    RoadWord = Word
End If

MatchCount = 0
Pos = i

ElseIf CheckSmrtWord = 1 Then
    If PassFrom = 0 And GotMrt = "" Then
        MrtWord = Word
        Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "'"
        'look at each char in text string
        Topstr = Topstr & "ORDER BY [mrt name]"
    End If
End If

```



```

Set ConnSmrt = New ADODB.Connection
ConnSmrt.Open "smrt"

Set SmrtRecordset = ConnSmrt.Execute(Topstr)

Do Until SmrtRecordset.EOF

    CheckMrtName = SmrtRecordset("mrt name")

    SmrtRecordset.MoveNext
    Loop
    If CheckMrtName <> "" Then
        GotMrt = CheckMrtName
        FromMrt = CheckMrtName
    End If
    CheckMrtName = ""

ElseIf PassFrom = 0 And GotMrt <> "" Then
    MrtWord = MrtWord & " " & Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"

    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
        Loop
        If CheckMrtName = "" Then
            PassFrom = 1
        ElseIf CheckMrtName <> "" And FromMrt <> CheckMrtName Then
            FromMrt = CheckMrtName
        End If
        CheckMrtName = ""
    End If

If PassFrom = 1 And PassTo = 0 And GotMrt <> "" Then
    MrtWord = Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)
    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
        Loop
        If CheckMrtName <> "" Then

            ToMrt = CheckMrtName
            PassTo = 1

        End If

        CheckMrtName = ""

ElseIf PassTo = 1 And PassFrom = 1 And GotMrt <> "" Then
    MrtWord = MrtWord & " " & Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

```

```

Do Until SmrtRecordset.EOF

    CheckMrtName = SmrtRecordset("mrt name")

    SmrtRecordset.MoveNext
    Loop

    If CheckMrtName <> "" And ToMrt <> CheckMrtName Then
        ToMrt = CheckMrtName
    End If
    CheckMrtName = ""
End If

    Pos = i

End If

'for last word
ElseIf Ticker = 1 And i = Len(SortWord) Then

    Word = Mid(SortWord, Pos + 1, i - Pos - 1)

    If CheckSmrtWord = 0 Then

        'first word is bus stop number
        If GotBusStop <> "" And PassTo = 0 Then

            RoadWord = Word
            Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & '%"
            'look at each char in text string
            Topstr = Topstr & "ORDER BY [road name]"

            Set ConnRoad = New ADODB.Connection
            ConnRoad.Open "road"

            Set RoadRecordset = ConnRoad.Execute(Topstr)
            Do Until RoadRecordset.EOF
                If ToRoad(0) = "" Then
                    ToRoad(0) = RoadRecordset("road name")

                    ToCount = 1
                End If
                CheckRoadPass = 1
                For j = 0 To 10

                    If ToRoad(j) <> "" Then
                        If ToRoad(j) = RoadRecordset("road name") Then
                            CheckRoadPass = 0
                            Exit For
                        End If
                    End If
                End If

                Next
                If CheckRoadPass = 1 Then
                    For j = 0 To 10
                        If ToRoad(j) = "" Then
                            ToRoad(j) = RoadRecordset("road name")
                            ToCount = ToCount + 1
                            Exit For
                        End If
                    Next
                End If

                RoadRecordset.MoveNext
            Loop
            PassTo = 1

        ElseIf GotBusStop <> "" And PassTo = 1 Then

            CheckRoad = ""

```

```

RoadWord = RoadWord & " " & Word
Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
'look at each char in text string
Topstr = Topstr & "ORDER BY [road name]"

Set ConnRoad = New ADODB.Connection
ConnRoad.Open "road"

Set MatchRecordset = ConnRoad.Execute(Topstr)
Do Until MatchRecordset.EOF

    If CheckRoad = "" Or CheckRoad <> MatchRecordset("road name") Then
        If CheckRoad = "" Then
            CheckRoad = MatchRecordset("road name")
            MatchCount = 1
        End If
        If CheckRoad <> MatchRecordset("road name") Then

            For j = 0 To 10

                If ToRoad(j) = CheckRoad Then
                    MatchCount = MatchCount + 1
                End If

            Next
            CheckRoad = MatchRecordset("road name")

        End If

    End If

MatchRecordset.MoveNext
Loop

If MatchCount <> ToCount And MatchCount <> 0 Then
    For j = 0 To 10
        ToRoad(j) = ""
    Next
    ToCount = 0
    Set RoadRecordset = ConnRoad.Execute(Topstr)
    Do Until RoadRecordset.EOF

        If ToRoad(0) = "" Then
            ToRoad(0) = RoadRecordset("road name")

            ToCount = 1
        End If
        CheckRoadPass = 1
        For j = 0 To 10

            If ToRoad(j) <> "" Then
                If ToRoad(j) = RoadRecordset("road name") Then
                    CheckRoadPass = 0
                    Exit For
                End If
            End If

        Next
        If CheckRoadPass = 1 Then
            For j = 0 To 10
                If ToRoad(j) = "" Then
                    ToRoad(j) = RoadRecordset("road name")
                    ToCount = ToCount + 1
                End If
            Next
        End If

    RoadRecordset.MoveNext
    Loop
End If

'first word is road name
ElseIf GotRoad <> "" And PassFrom = 0 Then
    CheckRoad = ""
    RoadWord = RoadWord & " " & Word

```

```

Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
'look at each char in text string
Topstr = Topstr & "ORDER BY [road name]"

Set MatchRecordset = ConnRoad.Execute(Topstr)
Do Until MatchRecordset.EOF
    If CheckRoad = "" Then
        CheckRoad = MatchRecordset("road name")
        MatchCount = 1
    End If
    If CheckRoad <> MatchRecordset("road name") Then

        For j = 0 To 10

            If FromRoad(j) = CheckRoad Then
                MatchCount = MatchCount + 1
            End If

        Next
        CheckRoad = MatchRecordset("road name")

    End If

MatchRecordset.MoveNext
Loop

If MatchCount <> FromCount And MatchCount <> 0 Then

    For j = 0 To 10
        FromRoad(j) = ""
    Next
    FromCount = 0
    Set RoadRecordset = ConnRoad.Execute(Topstr)
    Do Until RoadRecordset.EOF

        If FromRoad(0) = "" Then
            FromRoad(0) = RoadRecordset("road name")
            GotRoad = RoadRecordset("road name")
            FromCount = 1
        End If
        CheckRoadPass = 1
        For j = 0 To 10

            If FromRoad(j) <> "" Then
                If FromRoad(j) = RoadRecordset("road name") Then
                    CheckRoadPass = 0
                    Exit For
                End If
            End If

        Next
        If CheckRoadPass = 1 Then
            For j = 0 To 10
                If FromRoad(j) = "" Then
                    FromRoad(j) = RoadRecordset("road name")
                    FromCount = FromCount + 1
                    Exit For
                End If
            Next
        End If

        RoadRecordset.MoveNext
    Loop
End If

ElseIf GotRoad <> "" And PassFrom = 1 And PassTo = 0 Then

    RoadWord = RoadWord & " " & Word

    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set ConnRoad = New ADODB.Connection
    ConnRoad.Open "road"

```

```

Set RoadRecordset = ConnRoad.Execute(Topstr)
Do Until RoadRecordset.EOF

    If ToRoad(0) = "" Then
        ToRoad(0) = RoadRecordset("road name")
        GotRoad = RoadRecordset("road name")
        ToCount = 1
    End If
    CheckRoadPass = 1
    For j = 0 To 10

        If ToRoad(j) <> "" Then
            If ToRoad(j) = RoadRecordset("road name") Then
                CheckRoadPass = 0
                Exit For
            End If
        End If

    Next
    If CheckRoadPass = 1 Then
        For j = 0 To 10
            If ToRoad(j) = "" Then
                ToRoad(j) = RoadRecordset("road name")
                ToCount = ToCount + 1
                Exit For
            End If
        Next
    End If

    RoadRecordset.MoveNext
Loop

ElseIf GotRoad <> "" And PassFrom = 1 And PassTo = 1 Then
    CheckRoad = ""
    RoadWord = RoadWord & " " & Word
    Topstr = "SELECT * from road where [road name] Like '%" & RoadWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set ConnRoad = New ADODB.Connection
    ConnRoad.Open "road"

    Set MatchRecordset = ConnRoad.Execute(Topstr)
    Do Until MatchRecordset.EOF

        If CheckRoad = "" Or CheckRoad <> MatchRecordset("road name") Then
            If CheckRoad = "" Then
                CheckRoad = MatchRecordset("road name")
                MatchCount = 1
            End If
            If CheckRoad <> MatchRecordset("road name") Then

                For j = 0 To 10

                    If ToRoad(j) = CheckRoad Then
                        MatchCount = MatchCount + 1
                    End If

                Next
                CheckRoad = MatchRecordset("road name")

            End If

        End If

    MatchRecordset.MoveNext
Loop

If MatchCount <> ToCount And MatchCount <> 0 Then
    For j = 0 To 10
        ToRoad(j) = ""
    Next
    ToCount = 0
    Set RoadRecordset = ConnRoad.Execute(Topstr)
    Do Until RoadRecordset.EOF

```

```

    If ToRoad(0) = "" Then
        ToRoad(0) = RoadRecordset("road name")
        GotRoad = RoadRecordset("road name")
        ToCount = 1
    End If
    CheckRoadPass = 1
    For j = 0 To 10

        If ToRoad(j) <> "" Then
            If ToRoad(j) = RoadRecordset("road name") Then
                CheckRoadPass = 0
                Exit For
            End If
        End If

    Next
    If CheckRoadPass = 1 Then
        For j = 0 To 10
            If ToRoad(j) = "" Then
                ToRoad(j) = RoadRecordset("road name")
                ToCount = ToCount + 1
                Exit For
            End If
        Next
    End If

    RoadRecordset.MoveNext
    Loop
End If

ElseIf CheckSmrtWord = 1 Then

If PassFrom = 0 And GotMrt = "" Then
    MrtWord = Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"

    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
        Loop
        If CheckMrtName <> "" Then
            GotMrt = CheckMrtName
            FromMrt = CheckMrtName
        End If
        CheckMrtName = ""
    End If

ElseIf PassFrom = 0 And GotMrt <> "" Then
    MrtWord = MrtWord & " " & Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
        Loop
        If CheckMrtName = "" Then
            PassFrom = 1
        End If
    End If

```

```

ElseIf CheckMrtName <> "" And FromMrt <> CheckMrtName Then
    FromMrt = CheckMrtName
End If
CheckMrtName = ""
End If

If PassFrom = 1 And PassTo = 0 And GotMrt <> "" Then
    MrtWord = Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
    Loop
    If CheckMrtName <> "" Then

        ToMrt = CheckMrtName
        PassTo = 1

    End If
    CheckMrtName = ""

ElseIf PassTo = 1 And PassFrom = 1 And GotMrt <> "" Then
    MrtWord = MrtWord & " " & Word

    Topstr = "SELECT * from smrt where [mrt name] Like '%" & MrtWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [mrt name]"
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = ConnSmrt.Execute(Topstr)

    Do Until SmrtRecordset.EOF

        CheckMrtName = SmrtRecordset("mrt name")

        SmrtRecordset.MoveNext
    Loop

    If CheckMrtName <> "" And ToMrt <> CheckMrtName Then
        ToMrt = CheckMrtName
    End If
    CheckMrtName = ""
End If
End If
Next

If CheckSmrtWord = 0 Then
    'Deciding which SubFunctions to go to
    'asking for roadinfo onli
    If PassFrom = 0 And GotBusStop = "" And GotRoad <> "" Then
        Call RoadInfoList
        Exit Sub
    'Asking for road to road guide
    ElseIf PassFrom = 1 And GotRoad <> "" Then
        Call RoadToRoad
        Exit Sub
    'Asking for stop to road guide
    ElseIf GotBusStop <> "" And PassTo = 1 Then
        Call StopToRoad
        Exit Sub
    'No results
    ElseIf GotBusStop = "" And GotRoad = "" Then
        SMSText.Text = "No Results"
    End If
ElseIf CheckSmrtWord = 1 Then
    If FromMrt <> "" And ToMrt <> "" Then

```

```

    Call MrtToMrt
    Exit Sub
ElseIf FromMrt <> "" And ToMrt = "" Then
    Call MrtListInfo
    Exit Sub
ElseIf FromMrt = "" And ToMrt = "" Then
    SMSText.Text = "No Results"
End If
End If

End Sub
Private Sub RoadListText()

    DataNum = 0

    If CheckDual = 1 Then

        ServiceNo = ServiceNo & "dir1"

        Set ConnService = New ADODB.Connection
        ConnService.Open ServiceNo

        Set NumRecordset = New ADODB.Recordset
        NumRecordset.Open "SELECT * FROM " & ServiceNo, ConnService

        Do Until NumRecordset.EOF

            DataNum = DataNum + 1

            NumRecordset.MoveNext
            Loop

            Set ListRecordset = New ADODB.Recordset
            ListRecordset.Open "SELECT * FROM " & ServiceNo, ConnService

            MiddleDir = Int(DataNum / 2) + 1

            Do Until ListRecordset.EOF

                If ListRecordset("direction") = 1 Then
                    StartRd = ListRecordset("road name")
                End If
                If MiddleDir = ListRecordset("direction") Then
                    MiddleRd = ListRecordset("road name")
                End If
                If DataNum = ListRecordset("direction") Then
                    EndRd = ListRecordset("road name")
                End If

                ListRecordset.MoveNext
            Loop

            SMSText.Text = "Bus " & Serno & ": service to and fro " & StartRd _
                & " and " & EndRd & " via " & MiddleRd

        Else

            Set ConnService = New ADODB.Connection
            ConnService.Open ServiceNo

            Set NumRecordset = New ADODB.Recordset
            NumRecordset.Open "SELECT * FROM " & ServiceNo, ConnService

            Do Until NumRecordset.EOF

                DataNum = DataNum + 1
                NumRecordset.MoveNext
            Loop

            Set ListRecordset = New ADODB.Recordset
            ListRecordset.Open "SELECT * FROM " & ServiceNo, ConnService

            MiddleDir = Int(DataNum / 2)
            QuartDir = Int(DataNum / 4) + 1

```



```

Do Until ListRecordset.EOF

    If ListRecordset("direction") = 1 Then
        StartRd = ListRecordset("road name")
    End If
    If QuartDir = ListRecordset("direction") Then
        QuartRd = ListRecordset("road name")
    End If
    If MiddleDir = ListRecordset("direction") Then
        MiddleRd = ListRecordset("road name")
    End If

    ListRecordset.MoveNext
Loop

SMSText.Text = "Bus " & Serno & ": loop service from " & StartRd _
& " to " & MiddleRd & " via " & QuartRd

End If

End Sub

Private Sub StopToRoad()
    For i = 0 To 10
        For j = 0 To 10
            If FromRegion(i) = ToRegion(j) And FromRegion(i) <> "" And ToRegion(j) <> "" Then
                CheckRegionMatch = 1
                Exit For
            End If
        Next
    Next
    TransCharService = ""
    FromCharService = ""
    ToCharService = ""
    CheckLink = 0
    For i = 0 To 10
        FromService(i) = ""
        ToService(i) = ""
        Service(i) = ""
        TransServiceRoad(i) = ""
    Next
    For i = 0 To 50
        TransServiceRoad(i) = ""
    Next
    CheckStart = 0
    Set FromRecordset = New ADODB.Recordset
    FromRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad

    'looking at the from services
    Do Until FromRecordset.EOF

        If GotBusStop <> "" And GotBusStop = FromRecordset("bus stop no") Then
            If CheckStart = 0 Then
                For j = 1 To 7
                    SerNoChar = "service" & j
                    If FromRecordset(SerNoChar) <> "" Then
                        FromService(j) = FromRecordset(SerNoChar)
                    End If
                Next
                CheckStart = 1
            ElseIf CheckStart = 1 Then

                For j = 1 To 7
                    SerNoChar = "service" & j
                    CheckSerNo = 0
                    If FromRecordset(SerNoChar) <> "" Then
                        For k = 1 To 7
                            If FromService(k) = FromRecordset(SerNoChar) And FromService(k) <> "" Then
                                CheckSerNo = 1
                            End If
                        Next
                    If CheckSerNo = 0 Then
                        For k = 1 To 7
                            If FromService(k) = "" Then
                                FromService(k) = FromRecordset(SerNoChar)
                            Exit For
                        Next
                    End If
                Next
            End If
        End If
    End Sub

```

```

        End If
      Next
    End If
  End If
Next

End If

End If

FromRecordset.MoveNext
Loop

CheckStart = 0
Set ToRecordset = New ADODB.Recordset
ToRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad

'looking at the to services
Do Until ToRecordset.EOF
  For i = 0 To 10
    If ToRoad(i) <> "" And ToRoad(i) = ToRecordset("road name") Then

      If CheckStart = 0 Then
        For j = 1 To 7
          SerNoChar = "service" & j
          If ToRecordset(SerNoChar) <> "" Then
            ToService(j) = ToRecordset(SerNoChar)
          End If
        Next
        CheckStart = 1
      ElseIf CheckStart = 1 Then

        For j = 1 To 7
          SerNoChar = "service" & j
          CheckSerNo = 0
          If ToRecordset(SerNoChar) <> "" Then
            For k = 1 To 7
              If ToService(k) = ToRecordset(SerNoChar) And ToService(k) <> "" Then
                CheckSerNo = 1
              End If
            Next
            If CheckSerNo = 0 Then
              For k = 1 To 7
                If ToService(k) = "" Then
                  ToService(k) = ToRecordset(SerNoChar)
                Exit For
              End If
            Next
          End If
        Next
      End If
    Next
  End If
  ToRecordset.MoveNext
Loop

'Checking whether services are linked
Dim StopToRoadCheckService
StopToRoadCheckService = 0
For i = 1 To 7
  For j = 1 To 7
    If FromService(i) = ToService(j) And FromService(i) <> "" And ToService(j) <> "" Then
      CheckLink = 1
      For k = 1 To 7
        If Service(k) = FromService(i) And Service(k) <> "" Then
          StopToRoadCheckService = 1
        End If
      Next
      If Service(1) = "" Then
        Service(1) = FromService(i)
      End If
      If StopToRoadCheckService = 1 Then
        StopToRoadCheckService = 0
        For k = 1 To 7
          If Service(k) = "" Then

```

```

        Service(k) = FromService(i)
    Exit For
End If
Next
End If
End If
Next
Next

```

```

If CheckLink = 1 Then
'roads connected by services
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
For i = 1 To 7

```

```

    If Service(i) <> "" Then
        Do Until ServiceRecordset.EOF
            If Service(i) = ServiceRecordset("service no") Then
                CheckDual = ServiceRecordset("dual direction")
                Exit Do
            End If
            ServiceRecordset.MoveNext
        Loop
    End If

```

```

    If CheckDual = 0 Then
        Call LoopList
        Exit Sub
    ElseIf CheckDual = 1 Then
        Call DualList
        Exit Sub
    End If
Next

```

```

ElseIf CheckLink = 0 Then

```

```

CheckFromService = 0
CheckToService = 0
FromCharService = ""
ToCharService = ""
TransCharService = ""

```

```

'roads not connected by services
Set RoadRecordset = New ADODB.Recordset
RoadRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad
Do Until RoadRecordset.EOF
    For i = 1 To 7
        SerNoChar = "service" & i
        For j = 1 To 7
            If FromService(j) = RoadRecordset(SerNoChar) And FromService(j) <> "" Then
                CheckFromService = 1
                FromCharSer = FromService(j)
            End If
        Next
        For j = 1 To 7
            If ToService(j) = RoadRecordset(SerNoChar) And ToService(j) <> "" Then
                CheckToService = 1
                ToCharSer = ToService(j)
            End If
        Next
    Next
Next
If CheckFromService = 1 And CheckToService = 1 Then
    FromCharService = FromCharSer
    ToCharService = ToCharSer
    If FromCharService <> "" And ToCharService <> "" Then
        For j = 0 To 50
            If TransServiceRoad(j) = "" Then
                TransServiceRoad(j) = RoadRecordset("road name")
                Exit For
            End If
        Next
    End If
Next
End If

```

```

    End If
    If CheckToService = 0 Then
        CheckFromService = 0
        FromCharSer = ""
    End If
    If CheckFromService = 0 Then
        CheckToService = 0
        ToCharSer = ""
    End If
    RoadRecordset.MoveNext
Loop

'check the two services for loop or dual
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
Do Until ServiceRecordset.EOF
    If FromCharService = ServiceRecordset("service no") Then
        CheckFromDual = ServiceRecordset("dual direction")
    End If
    If ToCharService = ServiceRecordset("service no") Then
        CheckToDual = ServiceRecordset("dual direction")
    End If
    ServiceRecordset.MoveNext
Loop
If CheckFromDual = 0 And CheckToDual = 0 Then
    Call LoopToLoop
    Exit Sub
ElseIf CheckFromDual = 0 And CheckToDual = 1 Then
    Call LoopToDual
    Exit Sub
ElseIf CheckFromDual = 1 And CheckToDual = 0 Then
    Call DualToLoop
    Exit Sub
ElseIf CheckFromDual = 1 And CheckToDual = 1 Then
    Call DualToDual
    Exit Sub
End If

End If

End Sub
Private Sub RoadToRoad()

    For i = 0 To 10
        For j = 0 To 10
            If FromRegion(i) = ToRegion(j) And FromRegion(i) <> "" And ToRegion(j) <> "" Then
                CheckRegionMatch = 1
                Exit For
            End If
        Next
    Next

    CheckLink = 0
    For i = 0 To 10
        FromService(i) = ""
        ToService(i) = ""
        Service(i) = ""
    Next
    CheckStart = 0
    Set FromRecordset = New ADODB.Recordset
    FromRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad
    'looking at the from services
    Do Until FromRecordset.EOF
        For i = 0 To 10
            If FromRoad(i) <> "" And FromRoad(i) = FromRecordset("road name") Then
                If CheckStart = 0 Then
                    For j = 1 To 7
                        SerNoChar = "service" & j
                        If FromRecordset(SerNoChar) <> "" Then
                            FromService(j) = FromRecordset(SerNoChar)
                        End If
                    Next
                    CheckStart = 1
                ElseIf CheckStart = 1 Then
                    For j = 1 To 7

```

```

SerNoChar = "service" & j
CheckSerNo = 0
If FromRecordset(SerNoChar) <> "" Then
    For k = 1 To 7
        If FromService(k) = FromRecordset(SerNoChar) And FromService(k) <> "" Then
            CheckSerNo = 1
        End If
    Next
    If CheckSerNo = 0 Then
        For k = 1 To 7
            If FromService(k) = "" Then
                FromService(k) = FromRecordset(SerNoChar)
            Exit For
        End If
    Next
End If
Next
End If
Next
End If
FromRecordset.MoveNext
Loop

CheckStart = 0
Set ToRecordset = New ADODB.Recordset
ToRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad

'looking at the to services
Do Until ToRecordset.EOF
    For i = 0 To 10
        If ToRoad(i) <> "" And ToRoad(i) = ToRecordset("road name") Then

            If CheckStart = 0 Then
                For j = 1 To 7
                    SerNoChar = "service" & j
                    If ToRecordset(SerNoChar) <> "" Then
                        ToService(j) = ToRecordset(SerNoChar)
                    End If
                Next
                CheckStart = 1
            ElseIf CheckStart = 1 Then

                For j = 1 To 7
                    SerNoChar = "service" & j
                    CheckSerNo = 0
                    If ToRecordset(SerNoChar) <> "" Then
                        For k = 1 To 7
                            If ToService(k) = ToRecordset(SerNoChar) And ToService(k) <> "" Then
                                CheckSerNo = 1
                            End If
                        Next
                        If CheckSerNo = 0 Then
                            For k = 1 To 7
                                If ToService(k) = "" Then
                                    ToService(k) = ToRecordset(SerNoChar)
                                Exit For
                            End If
                        End If
                    Next
                End If
            Next
        End If
    Next
    ToRecordset.MoveNext
Loop
'Checking whether services are linked
For i = 1 To 7
    For j = 1 To 7
        If FromService(i) = ToService(j) And FromService(i) <> "" And ToService(j) <> "" Then
            CheckLink = 1
        For k = 1 To 7
            If Service(k) = "" Then
                Service(k) = FromService(i)
            Exit For
        End If
    Next
End If
Next

```

```

        End If
    Next
End If
Next
Next

'roads connected by services
If CheckLink = 1 Then
    ServiceChar = ""
    For i = 1 To 7
        If Service(i) <> "" And ServiceChar = "" Then
            ServiceChar = Service(i)
        ElseIf Service(i) <> "" And ServiceChar <> "" Then
            ServiceChar = ServiceChar & ", " & Service(i)
        End If
    Next
    SMSText.Text = "available bus service: " & ServiceChar

ElseIf CheckLink = 0 Then
'roads not connected by services
    Call TransferService
    Exit Sub

End If

If CheckLink = 1 And CheckRegionMatch = 0 Then
'roads connected by services but far
    Call FarRoadtoRoad
    Exit Sub
End If

End Sub
Private Sub RoadInfoList()

    For i = 0 To 10
        Service(i) = ""
    Next

    Set SerNoRecordset = New ADODB.Recordset
    SerNoRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad

    Do Until SerNoRecordset.EOF

        For i = 0 To 10
            If FromRoad(i) <> "" And FromRoad(i) = SerNoRecordset("road name") Then
                If CheckStart = 0 Then
                    For j = 1 To 7
                        SerNoChar = "service" & j
                        If SerNoRecordset(SerNoChar) <> "" Then
                            Service(j) = SerNoRecordset(SerNoChar)
                        End If
                    Next
                    CheckStart = 1
                ElseIf CheckStart = 1 Then

                    For j = 1 To 7
                        SerNoChar = "service" & j
                        CheckSerNo = 0
                        If SerNoRecordset(SerNoChar) <> "" Then
                            For k = 1 To 7
                                If Service(k) = SerNoRecordset(SerNoChar) And Service(k) <> "" Then
                                    CheckSerNo = 1
                                End If
                            Next
                            If CheckSerNo = 0 Then
                                For k = 1 To 7
                                    If Service(k) = "" Then
                                        Service(k) = SerNoRecordset(SerNoChar)
                                    Exit For
                                End If
                            Next
                        End If
                    End If
                End If
            Next
        End If
    Next
Next

```

```

        End If
    End If
Next

SerNoRecordset.MoveNext
Loop

ServiceChar = ""
For i = 1 To 7
    If Service(i) <> "" And ServiceChar = "" Then
        ServiceChar = Service(i)
    ElseIf Service(i) <> "" And ServiceChar <> "" Then
        ServiceChar = ServiceChar & ", " & Service(i)
    End If
Next
SMSText.Text = "available bus service: " & ServiceChar

End Sub
Private Sub LoopToLoop()
    Dim CheckRepeatLoop, CheckRepeatTransLoop
    ToDir = ""
    OppToDir = ""
    OppStopNo = ""
    StopNo = ""
    StopDir = ""
    OppStopDir = ""
    LoopListChar = ""
    CheckRepeatTransLoop = 0

    ServiceNo = FromCharService
    Set ConnBus = New ADODB.Connection
    ConnBus.Open ServiceNo
    Set LoopListRecordset = New ADODB.Recordset
    LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
    Do Until LoopListRecordset.EOF
        For j = 0 To 50
            If TransServiceRoad(j) = LoopListRecordset("road name") Then
                TransCharService = TransServiceRoad(j)
                Exit Do
            End If
        Next
        LoopListRecordset.MoveNext
        Loop
        ToDir = ""
        OppToDir = ""
        OppStopNo = ""
        StopNo = ""
        StopDir = ""
        OppStopDir = ""
        Set LoopListRecordset = New ADODB.Recordset
        LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
        Do Until LoopListRecordset.EOF
            If TransCharService = LoopListRecordset("road name") Then
                If LoopListRecordset("bus stop no") = LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("opp direction")
                    OppToDir = LoopListRecordset("opp direction")
                ElseIf LoopListRecordset("bus stop no") <> LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("direction")
                    OppToDir = LoopListRecordset("opp direction")
                End If
            End If

            End If

            If BusStopNo = LoopListRecordset("bus stop no") Then
                StopNo = BusStopNo
                OppStopNo = LoopListRecordset("opp bus stop no")
                StopDir = LoopListRecordset("direction")
                OppStopDir = LoopListRecordset("opp direction")
            End If
        LoopListRecordset.MoveNext
        Loop
        If OppToDir = ToDir Then
            If StopDir >= OppStopDir Then
                If LoopListChar = "" Then
                    LoopListChar = "take bus:" & FromCharService
                    CheckRepeatLoop = FromCharService
                End If
            End If
        End If
    End Sub

```

```

ElseIf LoopListChar <> "" Then
    If CheckRepeatLoop <> FromCharService Then
        LoopListChar = LoopListChar & ", " & FromCharService
    End If
End If
ElseIf OppStopDir > StopDir Then
    If LoopListChar = "" Then
        LoopListChar = "go opposite bus stop, take bus:" & FromCharService
        CheckRepeatLoop = FromCharService
    ElseIf LoopListChar <> "" Then
        If CheckRepeatLoop <> FromCharService Then
            LoopListChar = LoopListChar & ", " & FromCharService
        End If
    End If
End If
ElseIf ToDir <> OppToDir Then

    If ToDir > StopDir Then
        If LoopListChar = "" Then
            LoopListChar = "take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If

    ElseIf ToDir < StopDir Then
        If LoopListChar = "" Then
            LoopListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    End If
End If

ServiceNo = ToCharService
Set ConnBus = New ADODB.Connection
ConnBus.Open ServiceNo
Set LoopListRecordset = New ADODB.Recordset
LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
ToDir = ""
OppToDir = ""
OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
For j = 0 To 10
    If ToRoad(j) <> "" Then
        Do Until LoopListRecordset.EOF
            If ToRoad(j) = LoopListRecordset("road name") Then
                If LoopListRecordset("bus stop no") = LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("opp direction")
                    OppToDir = LoopListRecordset("opp direction")
                ElseIf LoopListRecordset("bus stop no") <> LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("direction")
                    OppToDir = LoopListRecordset("opp direction")
                End If
            End If
        End If
        If TransCharService = LoopListRecordset("road name") And StopNo = "" Then
            StopNo = LoopListRecordset("bus stop no")
            OppStopNo = LoopListRecordset("opp bus stop no")
            StopDir = LoopListRecordset("direction")
            OppStopDir = LoopListRecordset("opp direction")
        End If
        LoopListRecordset.MoveNext
    Loop
    If OppToDir = ToDir Then
        If StopDir <= ToDir Then
            If LoopListChar <> "" And CheckRepeatTransLoop = 0 Then

                LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransLoop = 1
            End If
        End If
    End If
Next j

```



```

        CheckRepeatLoop = ToCharService
    End If
    If CheckRepeatLoop <> ToCharService Then
        LoopListChar = LoopListChar & ", " & ToCharService
    End If

    ElseIf OppStopDir < ToDir Then
        If LoopListChar <> "" And CheckRepeatTransLoop = 0 Then

            LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
            CheckRepeatTransLoop = 1
            CheckRepeatLoop = ToCharService
        End If
        If CheckRepeatLoop <> ToCharService Then
            LoopListChar = LoopListChar & ", " & ToCharService
        End If
    End If
    ElseIf ToDir <> OppToDir Then

        If ToDir > StopDir Then
            If LoopListChar <> "" And CheckRepeatTransLoop = 0 Then

                LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransLoop = 1
                CheckRepeatLoop = ToCharService
            End If
            If CheckRepeatLoop <> ToCharService Then
                LoopListChar = LoopListChar & ", " & ToCharService
            End If
        End If

        ElseIf ToDir < StopDir Then
            If LoopListChar <> "" And CheckRepeatTransLoop = 0 Then

                LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
                CheckRepeatTransLoop = 1
                CheckRepeatLoop = ToCharService
            End If
            If CheckRepeatLoop <> ToCharService Then
                LoopListChar = LoopListChar & ", " & ToCharService
            End If
        End If
    End If
Next
If LoopListChar = "" Then
    SMSText.Text = "no results"
ElseIf LoopListChar <> "" Then
    SMSText.Text = LoopListChar
End If
If CheckRegionMatch = 0 Then
    Call FarStopToRoad
End If
End Sub
Private Sub LoopToDual()
    Dim CheckRepeatLoop
    Dim CheckRepeatDual, CheckDualDir1, CheckDualDir2, CheckRepeatTransDual
    ToDir = ""
    OppToDir = ""
    OppStopNo = ""
    StopNo = ""
    StopDir = ""
    OppStopDir = ""
    LoopListChar = ""
    CheckRepeatTransDual = 0
    DualListChar = ""
    CheckDualDir1 = 0
    CheckDualDir2 = 0

    ServiceNo = FromCharService
    Set ConnBus = New ADODB.Connection
    ConnBus.Open ServiceNo
    Set LoopListRecordset = New ADODB.Recordset
    LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
    ToDir = ""

```

```

OppToDir = ""
OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
Set LoopListRecordset = New ADODB.Recordset
LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until LoopListRecordset.EOF
    For j = 0 To 50
        If TransServiceRoad(j) = LoopListRecordset("road name") Then
            TransCharService = TransServiceRoad(j)
            Exit Do
        End If
    Next
LoopListRecordset.MoveNext
Loop
Do Until LoopListRecordset.EOF
    If TransCharService = LoopListRecordset("road name") Then
        If LoopListRecordset("bus stop no") = LoopListRecordset("opp bus stop no") Then
            ToDir = LoopListRecordset("opp direction")
            OppToDir = LoopListRecordset("opp direction")
        ElseIf LoopListRecordset("bus stop no") <> LoopListRecordset("opp bus stop no") Then
            ToDir = LoopListRecordset("direction")
            OppToDir = LoopListRecordset("opp direction")
        End If

    End If

    If BusStopNo = LoopListRecordset("bus stop no") Then
        StopNo = BusStopNo
        OppStopNo = LoopListRecordset("opp bus stop no")
        StopDir = LoopListRecordset("direction")
        OppStopDir = LoopListRecordset("opp direction")
    End If
LoopListRecordset.MoveNext
Loop
If OppToDir = ToDir Then
    If StopDir >= OppStopDir Then
        If LoopListChar = "" Then
            LoopListChar = "take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    ElseIf OppStopDir > StopDir Then
        If LoopListChar = "" Then
            LoopListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    End If
ElseIf ToDir <> OppToDir Then

    If ToDir > StopDir Then
        If LoopListChar = "" Then
            LoopListChar = "take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    End If

    ElseIf ToDir < StopDir Then
        If LoopListChar = "" Then
            LoopListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatLoop = FromCharService
        ElseIf LoopListChar <> "" Then
            If CheckRepeatLoop <> FromCharService Then
                LoopListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    End If

```

```

End If
End If
End If

ToDir = ""
OppToDir = ""
OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
For j = 0 To 10
  If ToRoad(j) <> "" Then
    ServiceNo = ToCharService & "dir1"
    Set ConnBus = New ADODB.Connection
    ConnBus.Open ServiceNo
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
      If ToRoad(j) = DualListRecordset("road name") Then

        If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
          ToDir = DualListRecordset("opp direction")
          OppToDir = DualListRecordset("opp direction")
        ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
          ToDir = DualListRecordset("direction")
          OppToDir = DualListRecordset("opp direction")
        End If

      End If

      If TransCharService = DualListRecordset("road name") Then
        CheckDualDir1 = 1
        StopNo = DualListRecordset("bus stop no")
        OppStopNo = DualListRecordset("opp bus stop no")
        StopDir = DualListRecordset("direction")
        OppStopDir = DualListRecordset("opp direction")
      End If
    DualListRecordset.MoveNext
  Loop

  ServiceNo = ToCharService & "dir2"
  Set ConnBus = New ADODB.Connection
  ConnBus.Open ServiceNo
  Set DualListRecordset = New ADODB.Recordset
  DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
  Do Until DualListRecordset.EOF
    If ToRoad(j) = DualListRecordset("road name") Then

      If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
        ToDir = DualListRecordset("opp direction")
        OppToDir = DualListRecordset("opp direction")
      ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
        ToDir = DualListRecordset("direction")
        OppToDir = DualListRecordset("opp direction")
      End If

    End If

    If TransCharService = DualListRecordset("road name") Then
      CheckDualDir2 = 1
      StopNo = DualListRecordset("bus stop no")
      OppStopNo = DualListRecordset("opp bus stop no")
      StopDir = DualListRecordset("direction")
      OppStopDir = DualListRecordset("opp direction")
    End If
  DualListRecordset.MoveNext
Loop

If CheckDualDir1 = 1 Then
  If OppToDir = ToDir Then
    If StopDir >= OppStopDir Then

      If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

        LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
        CheckRepeatTransDual = 1
        CheckRepeatDual = ToCharService
      End If
    End If
  End If
End If

```

```

End If
If CheckRepeatDual <> ToCharService Then
  LoopListChar = LoopListChar & ", " & ToCharService
End If

ElseIf OppStopDir > StopDir Then
  If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

    LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
    CheckRepeatTransDual = 1
    CheckRepeatDual = ToCharService
  End If
  If CheckRepeatDual <> ToCharService Then
    LoopListChar = LoopListChar & ", " & ToCharService
  End If

End If
ElseIf ToDir <> OppToDir Then

  If ToDir > StopDir Then
    If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

      LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
      CheckRepeatTransDual = 1
      CheckRepeatDual = ToCharService
    End If
    If CheckRepeatDual <> ToCharService Then
      LoopListChar = LoopListChar & ", " & ToCharService
    End If

  ElseIf ToDir < StopDir Then
    If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

      LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
      CheckRepeatTransDual = 1
      CheckRepeatDual = ToCharService
    End If
    If CheckRepeatDual <> ToCharService Then
      LoopListChar = LoopListChar & ", " & ToCharService
    End If
  End If
  End If
  CheckDualDir1 = 0

ElseIf CheckDualDir2 = 1 Then

  If OppToDir = ToDir Then
    If StopDir >= OppStopDir Then

      If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

        LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
        CheckRepeatTransDual = 1
        CheckRepeatDual = ToCharService
      End If
      If CheckRepeatDual <> ToCharService Then
        LoopListChar = LoopListChar & ", " & ToCharService
      End If

    ElseIf OppStopDir > StopDir Then
      If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

        LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
        CheckRepeatTransDual = 1
        CheckRepeatDual = ToCharService
      End If
      If CheckRepeatDual <> ToCharService Then
        LoopListChar = LoopListChar & ", " & ToCharService
      End If
    End If
  End If

```

```

    End If
    ElseIf ToDir <> OppToDir Then

        If ToDir > StopDir Then
            If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

                LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                LoopListChar = LoopListChar & ", " & ToCharService
            End If

        ElseIf ToDir < StopDir Then
            If LoopListChar <> "" And CheckRepeatTransDual = 0 Then

                LoopListChar = LoopListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                LoopListChar = LoopListChar & ", " & ToCharService
            End If
        End If
    End If

    CheckDualDir2 = 0

    End If
    End If
Next

If LoopListChar = "" Then
    SMSText.Text = "no results"
ElseIf LoopListChar <> "" Then
    SMSText.Text = LoopListChar
End If
If CheckRegionMatch = 0 Then
    Call FarStopToRoad
End If
End Sub
Private Sub DualToLoop()
    Dim CheckRepeatLoop, CheckRepeatTransLoop
    Dim CheckRepeatDual, CheckDualDir1, CheckDualDir2
    ToDir = ""
    OppToDir = ""
    OppStopNo = ""
    StopNo = ""
    StopDir = ""
    OppStopDir = ""
    DualListChar = ""
    CheckDualDir1 = 0
    CheckDualDir2 = 0
    LoopListChar = ""
    CheckRepeatTransLoop = 0

    ServiceNo = FromCharService & "dir1"
    Set ConnBus = New ADODB.Connection
    ConnBus.Open ServiceNo
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        If BusStopNo = DualListRecordset("bus stop no") Then
            CheckDualDir1 = 1
            StopNo = BusStopNo
            OppStopNo = DualListRecordset("opp bus stop no")
            StopDir = DualListRecordset("direction")
            OppStopDir = DualListRecordset("opp direction")
        End If
        DualListRecordset.MoveNext
    Loop
    If CheckDualDir1 = 1 Then
        Set DualListRecordset = New ADODB.Recordset

```

```

DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
  For j = 0 To 50
    If TransServiceRoad(j) = DualListRecordset("road name") Then
      TransCharService = TransServiceRoad(j)
      Exit Do
    End If
  Next
DualListRecordset.MoveNext
Loop

Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
  If TransCharService = DualListRecordset("road name") Then

    If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
      ToDir = DualListRecordset("opp direction")
      OppToDir = DualListRecordset("opp direction")
    ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
      ToDir = DualListRecordset("direction")
      OppToDir = DualListRecordset("opp direction")
    End If

  End If
DualListRecordset.MoveNext
Loop
End If

ServiceNo = FromCharService & "dir2"
Set ConnBus = New ADODB.Connection
ConnBus.Open ServiceNo
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF

  If BusStopNo = DualListRecordset("bus stop no") Then
    CheckDualDir2 = 1
    StopNo = BusStopNo
    OppStopNo = DualListRecordset("opp bus stop no")
    StopDir = DualListRecordset("direction")
    OppStopDir = DualListRecordset("opp direction")
  End If
DualListRecordset.MoveNext
Loop
If CheckDualDir2 = 1 Then
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
  For j = 0 To 50
    If TransServiceRoad(j) = DualListRecordset("road name") Then
      TransCharService = TransServiceRoad(j)
      Exit Do
    End If
  Next
DualListRecordset.MoveNext
Loop

Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
  If TransCharService = DualListRecordset("road name") Then

    If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
      ToDir = DualListRecordset("opp direction")
      OppToDir = DualListRecordset("opp direction")
    ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
      ToDir = DualListRecordset("direction")
      OppToDir = DualListRecordset("opp direction")
    End If

  End If
DualListRecordset.MoveNext
Loop
End If
If CheckDualDir1 = 1 Then

```

```

If OppToDir = ToDir Then
  If StopDir >= OppStopDir Then
    If DualListChar = "" Then
      DualListChar = "take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = DualListChar & ", " & FromCharService
      End If
    End If
  ElseIf OppStopDir > StopDir Then
    If DualListChar = "" Then
      DualListChar = "go opposite bus stop, take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = DualListChar & ", " & FromCharService
      End If
    End If
  End If
ElseIf ToDir <> OppToDir Then

  If ToDir > StopDir Then
    If DualListChar = "" Then
      DualListChar = "take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = DualListChar & ", " & FromCharService
      End If
    End If

  ElseIf ToDir < StopDir Then
    If DualListChar = "" Then
      DualListChar = "go opposite bus stop, take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = LoopListChar & ", " & FromCharService
      End If
    End If
  End If
End If
CheckDualDir1 = 0

ElseIf CheckDualDir2 = 1 Then
  If OppToDir = ToDir Then
    If StopDir >= OppStopDir Then
      If DualListChar = "" Then
        DualListChar = "take bus:" & FromCharService
        CheckRepeatDual = FromCharService
      ElseIf DualListChar <> "" Then
        If CheckRepeatDual <> FromCharService Then
          DualListChar = DualListChar & ", " & FromCharService
        End If
      End If
    End If
  ElseIf OppStopDir > StopDir Then
    If DualListChar = "" Then
      DualListChar = "go opposite bus stop, take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = DualListChar & ", " & FromCharService
      End If
    End If
  End If
ElseIf ToDir <> OppToDir Then

  If ToDir > StopDir Then
    If DualListChar = "" Then
      DualListChar = "take bus:" & FromCharService
      CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
      If CheckRepeatDual <> FromCharService Then
        DualListChar = DualListChar & ", " & FromCharService
      End If
    End If
  End If
End If

```

```

        End If
    End If

    ElseIf ToDir < StopDir Then
        If DualListChar = "" Then
            DualListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = DualListChar & ", " & FromCharService
            End If
        End If
    End If
End If

CheckDualDir2 = 0

End If

ServiceNo = ToCharService
Set ConnBus = New ADODB.Connection
ConnBus.Open ServiceNo
Set LoopListRecordset = New ADODB.Recordset
LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
ToDir = ""
OppToDir = ""
OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
For j = 0 To 10
    If ToRoad(j) <> "" Then
        Do Until LoopListRecordset.EOF
            If ToRoad(j) = LoopListRecordset("road name") Then
                If LoopListRecordset("bus stop no") = LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("opp direction")
                    OppToDir = LoopListRecordset("opp direction")
                ElseIf LoopListRecordset("bus stop no") <> LoopListRecordset("opp bus stop no") Then
                    ToDir = LoopListRecordset("direction")
                    OppToDir = LoopListRecordset("opp direction")
                End If
            End If
            If TransCharService = LoopListRecordset("road name") And StopNo = "" Then
                StopNo = LoopListRecordset("bus stop no")
                OppStopNo = LoopListRecordset("opp bus stop no")
                StopDir = LoopListRecordset("direction")
                OppStopDir = LoopListRecordset("opp direction")
            End If
            LoopListRecordset.MoveNext
        Loop
        If OppToDir = ToDir Then
            If StopDir >= OppStopDir Then
                If DualListChar <> "" And CheckRepeatTransLoop = 0 Then

                    DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                    CheckRepeatTransLoop = 1
                    CheckRepeatLoop = ToCharService
                End If
                If CheckRepeatLoop <> ToCharService Then
                    DualListChar = DualListChar & ", " & ToCharService
                End If

                ElseIf OppStopDir > StopDir Then
                    If DualListChar <> "" And CheckRepeatTransLoop = 0 Then

                        DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
                        CheckRepeatTransLoop = 1
                        CheckRepeatLoop = ToCharService
                    End If
                    If CheckRepeatLoop <> ToCharService Then
                        DualListChar = DualListChar & ", " & ToCharService
                    End If
                End If
            ElseIf ToDir <> OppToDir Then

```



```

If ToDir > StopDir Then
    If DualListChar <> "" And CheckRepeatTransLoop = 0 Then

        DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
        CheckRepeatTransLoop = 1
        CheckRepeatLoop = ToCharService
    End If
    If CheckRepeatLoop <> ToCharService Then
        DualListChar = DualListChar & ", " & ToCharService
    End If

ElseIf ToDir < StopDir Then
    If DualListChar <> "" And CheckRepeatTransLoop = 0 Then

        DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
        CheckRepeatTransLoop = 1
        CheckRepeatLoop = ToCharService
    End If
    If CheckRepeatLoop <> ToCharService Then
        DualListChar = DualListChar & ", " & ToCharService
    End If
    End If
End If
End If
Next

If DualListChar = "" Then
    SMSText.Text = "no results"
ElseIf DualListChar <> "" Then
    SMSText.Text = DualListChar
End If
If CheckRegionMatch = 0 Then
    Call FarStopToRoad
End If
End Sub
Private Sub DualToDual()
    Dim CheckRepeatDual, CheckDualDir1, CheckDualDir2, CheckRepeatTransDual
    ToDir = ""
    OppToDir = ""
    OppStopNo = ""
    StopNo = ""
    StopDir = ""
    OppStopDir = ""
    DualListChar = ""
    CheckDualDir1 = 0
    CheckDualDir2 = 0
    LoopListChar = ""

    ServiceNo = FromCharService & "dir1"
    Set ConnBus = New ADODB.Connection
    ConnBus.Open ServiceNo
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        If BusStopNo = DualListRecordset("bus stop no") Then
            CheckDualDir1 = 1
            StopNo = BusStopNo
            OppStopNo = DualListRecordset("opp bus stop no")
            StopDir = DualListRecordset("direction")
            OppStopDir = DualListRecordset("opp direction")
        End If
        DualListRecordset.MoveNext
    Loop
    If CheckDualDir1 = 1 Then
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
        Do Until DualListRecordset.EOF
            For j = 0 To 50
                If TransServiceRoad(j) = DualListRecordset("road name") Then
                    TransCharService = TransServiceRoad(j)
                    Exit Do
                End If
            Next
        DualListRecordset.MoveNext
    Loop

```

```

Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
    If TransCharService = DualListRecordset("road name") Then

        If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("opp direction")
            OppToDir = DualListRecordset("opp direction")
        ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("direction")
            OppToDir = DualListRecordset("opp direction")
        End If

    End If
DualListRecordset.MoveNext
Loop
End If

ServiceNo = FromCharService & "dir2"
Set ConnBus = New ADODB.Connection
ConnBus.Open ServiceNo
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF

    If BusStopNo = DualListRecordset("bus stop no") Then
        CheckDualDir2 = 1
        StopNo = BusStopNo
        OppStopNo = DualListRecordset("opp bus stop no")
        StopDir = DualListRecordset("direction")
        OppStopDir = DualListRecordset("opp direction")
    End If
DualListRecordset.MoveNext
Loop
If CheckDualDir2 = 1 Then
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
    For j = 0 To 50
        If TransServiceRoad(j) = DualListRecordset("road name") Then
            TransCharService = TransServiceRoad(j)
            Exit Do
        End If
    Next
DualListRecordset.MoveNext
Loop
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
    If TransCharService = DualListRecordset("road name") Then

        If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("opp direction")
            OppToDir = DualListRecordset("opp direction")
        ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("direction")
            OppToDir = DualListRecordset("opp direction")
        End If

    End If
DualListRecordset.MoveNext
Loop
End If
If CheckDualDir1 = 1 Then
    If OppToDir = ToDir Then
        If StopDir >= OppStopDir Then
            If DualListChar = "" Then
                DualListChar = "take bus:" & FromCharService
                CheckRepeatDual = FromCharService
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> FromCharService Then
                    DualListChar = DualListChar & ", " & FromCharService
                End If
            End If
        End If
    ElseIf OppStopDir > StopDir Then
        If DualListChar = "" Then

```

```

        DualListChar = "go opposite bus stop, take bus:" & FromCharService
        CheckRepeatDual = FromCharService
    ElseIf DualListChar <> "" Then
        If CheckRepeatDual <> FromCharService Then
            DualListChar = DualListChar & ", " & FromCharService
        End If
    End If
End If
ElseIf ToDir <> OppToDir Then

    If ToDir > StopDir Then
        If DualListChar = "" Then
            DualListChar = "take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = DualListChar & ", " & FromCharService
            End If
        End If
    End If

    ElseIf ToDir < StopDir Then
        If DualListChar = "" Then
            DualListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = LoopListChar & ", " & FromCharService
            End If
        End If
    End If
End If
CheckDualDir1 = 0

Elseif CheckDualDir2 = 1 Then
    If OppToDir = ToDir Then
        If StopDir >= OppStopDir Then
            If DualListChar = "" Then
                DualListChar = "take bus:" & FromCharService
                CheckRepeatDual = FromCharService
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> FromCharService Then
                    DualListChar = DualListChar & ", " & FromCharService
                End If
            End If
        End If
    ElseIf OppStopDir > StopDir Then
        If DualListChar = "" Then
            DualListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = DualListChar & ", " & FromCharService
            End If
        End If
    End If
Elseif ToDir <> OppToDir Then

    If ToDir > StopDir Then
        If DualListChar = "" Then
            DualListChar = "take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = DualListChar & ", " & FromCharService
            End If
        End If
    End If

    ElseIf ToDir < StopDir Then
        If DualListChar = "" Then
            DualListChar = "go opposite bus stop, take bus:" & FromCharService
            CheckRepeatDual = FromCharService
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> FromCharService Then
                DualListChar = DualListChar & ", " & FromCharService
            End If
        End If
    End If
End If

```

```

    End If
End If

CheckDualDir2 = 0

End If
ToDir = ""
OppToDir = ""
OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
For j = 0 To 10
    If ToRoad(j) <> "" Then
        ServiceNo = ToCharService & "dir1"
        Set ConnBus = New ADODB.Connection
        ConnBus.Open ServiceNo
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
        Do Until DualListRecordset.EOF

            If TransCharService = DualListRecordset("road name") Then
                CheckDualDir1 = 1
                StopNo = DualListRecordset("bus stop no")
                OppStopNo = DualListRecordset("opp bus stop no")
                StopDir = DualListRecordset("direction")
                OppStopDir = DualListRecordset("opp direction")
            End If
            DualListRecordset.MoveNext
        Loop

        If CheckDualDir1 = 1 Then

            Set DualListRecordset = New ADODB.Recordset
            DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
            Do Until DualListRecordset.EOF
                If ToRoad(j) = DualListRecordset("road name") Then

                    If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
                        ToDir = DualListRecordset("opp direction")
                        OppToDir = DualListRecordset("opp direction")
                    ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
                        ToDir = DualListRecordset("direction")
                        OppToDir = DualListRecordset("opp direction")
                    End If

                End If
                DualListRecordset.MoveNext
            Loop
        End If

        ServiceNo = ToCharService & "dir2"
        Set ConnBus = New ADODB.Connection
        ConnBus.Open ServiceNo
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
        Do Until DualListRecordset.EOF

            If TransCharService = DualListRecordset("road name") Then
                CheckDualDir2 = 1
                StopNo = DualListRecordset("bus stop no")
                OppStopNo = DualListRecordset("opp bus stop no")
                StopDir = DualListRecordset("direction")
                OppStopDir = DualListRecordset("opp direction")
            End If
            DualListRecordset.MoveNext
        Loop

        If CheckDualDir2 = 1 Then

            Set DualListRecordset = New ADODB.Recordset
            DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
            Do Until DualListRecordset.EOF
                If ToRoad(j) = DualListRecordset("road name") Then

```

```

If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
    ToDir = DualListRecordset("opp direction")
    OppToDir = DualListRecordset("opp direction")
ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
    ToDir = DualListRecordset("direction")
    OppToDir = DualListRecordset("opp direction")
End If

End If
DualListRecordset.MoveNext
Loop
End If
If CheckDualDir1 = 1 Then
    If OppToDir = ToDir Then
        If StopDir >= OppStopDir Then

            If DualListChar <> "" And CheckRepeatTransDual = 0 Then

                DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If

        ElseIf OppStopDir > StopDir Then
            If DualListChar <> "" And CheckRepeatTransDual = 0 Then

                DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If

        End If
    ElseIf ToDir <> OppToDir Then

        If ToDir > StopDir Then
            If DualListChar <> "" And CheckRepeatTransDual = 0 Then

                DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If

        ElseIf ToDir < StopDir Then
            If DualListChar <> "" And CheckRepeatTransDual = 0 Then

                DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If
        End If
    End If
    CheckDualDir1 = 0

ElseIf CheckDualDir2 = 1 Then

    If OppToDir = ToDir Then
        If StopDir >= OppStopDir Then

            If DualListChar <> "" And CheckRepeatTransDual = 0 Then

```

```

        DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
        CheckRepeatTransDual = 1
        CheckRepeatDual = ToCharService
    End If
    If CheckRepeatDual <> ToCharService Then
        DualListChar = DualListChar & ", " & ToCharService
    End If

ElseIf OppStopDir > StopDir Then
    If DualListChar <> "" And CheckRepeatTransDual = 0 Then

ToCharService
        DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
        CheckRepeatTransDual = 1
        CheckRepeatDual = ToCharService
    End If
    If CheckRepeatDual <> ToCharService Then
        DualListChar = DualListChar & ", " & ToCharService
    End If

    End If
ElseIf ToDir <> OppToDir Then

    If ToDir > StopDir Then
        If DualListChar <> "" And CheckRepeatTransDual = 0 Then

                DualListChar = DualListChar & ", transfer at " & TransCharService & ", take bus:" & ToCharService
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If

ElseIf ToDir < StopDir Then
        If DualListChar <> "" And CheckRepeatTransDual = 0 Then

ToCharService
                DualListChar = DualListChar & ", transfer at " & TransCharService & ", go opposite bus stop, take bus:" &
                CheckRepeatTransDual = 1
                CheckRepeatDual = ToCharService
            End If
            If CheckRepeatDual <> ToCharService Then
                DualListChar = DualListChar & ", " & ToCharService
            End If
        End If
    End If

    CheckDualDir2 = 0

    End If
End If
Next

If DualListChar = "" Then
    SMSText.Text = "no results"
ElseIf DualListChar <> "" Then
    SMSText.Text = DualListChar
End If
If CheckRegionMatch = 0 Then
    Call FarStopToRoad
End If
End Sub
Private Sub TransferService()
    CheckTransfer = 0
    For i = 0 To 10
        TransRoad(i) = ""
        FromTrans(i) = ""
        ToTrans(i) = ""
    Next
    Set TransRecordset = New ADO.DB.Recordset
    TransRecordset.Open "SELECT * FROM road ORDER BY [road name]", ConnRoad

    Do Until TransRecordset.EOF

```

```

For i = 1 To 7
  If FromService(i) <> "" Then
    For j = 1 To 7
      If FromService(i) = TransRecordset("service" & j) Then
        For k = 1 To 7
          If ToService(k) <> "" Then
            For m = 1 To 7
              If ToService(k) = TransRecordset("service" & m) Then
                If TransRoad(0) = "" Then
                  TransRoad(0) = TransRecordset("road name")
                  FromTrans(0) = FromService(i)
                  ToTrans(0) = ToService(k)
                End If
                For l = 0 To 10
                  If TransRoad(l) = TransRecordset("road name") Then
                    CheckTransfer = 1
                  End If
                Next
                If CheckTransfer = 0 Then
                  For l = 0 To 10
                    If TransRoad(l) = "" Then
                      TransRoad(l) = TransRecordset("road name")
                      FromTrans(l) = FromService(i)
                      ToTrans(l) = ToService(k)
                    End If
                  Next
                End If
              End If
            Next
          End If
        Next
      End If
    Next
  End If
Next
TransRecordset.MoveNext
Loop

'do smstext for road to road
ServiceChar = ""
For i = 0 To 10
  If TransRoad(i) <> "" Then
    ServiceChar = FromTrans(i) & " transfer at: " & TransRoad(i) & " and take bus:" & ToTrans(i)
  End If
Next
SMSText.Text = "Take Bus:" & ServiceChar

If CheckLink = 0 And CheckRegionMatch = 0 Then
  'roads connected by services but far
  Call FarTransferService
  Exit Sub
End If

End Sub
Private Sub LoopList()
  Dim CheckRepeatLoop
  ToDir = ""
  OppToDir = ""
  OppStopNo = ""
  StopNo = ""
  StopDir = ""
  OppStopDir = ""
  LoopListChar = ""
  For i = 1 To 7
    If Service(i) <> "" Then
      ServiceNo = Service(i)
      Set ConnBus = New ADODB.Connection
      ConnBus.Open ServiceNo
      Set LoopListRecordset = New ADODB.Recordset
      LoopListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
      For j = 0 To 10
        If ToRoad(j) <> "" Then
          ToDir = ""
          OppToDir = ""

```

```

OppStopNo = ""
StopNo = ""
StopDir = ""
OppStopDir = ""
Do Until LoopListRecordset.EOF
  If ToRoad(j) = LoopListRecordset("road name") Then

    If LoopListRecordset("bus stop no") = LoopListRecordset("opp bus stop no") Then
      ToDir = LoopListRecordset("opp direction")
      OppToDir = LoopListRecordset("opp direction")
    ElseIf LoopListRecordset("bus stop no") <> LoopListRecordset("opp bus stop no") Then
      ToDir = LoopListRecordset("direction")
      OppToDir = LoopListRecordset("opp direction")
    End If

  End If

  If BusStopNo = LoopListRecordset("bus stop no") Then
    StopNo = BusStopNo
    OppStopNo = LoopListRecordset("opp bus stop no")
    StopDir = LoopListRecordset("direction")
    OppStopDir = LoopListRecordset("opp direction")
  End If
LoopListRecordset.MoveNext
Loop

If OppToDir = ToDir Then
  If StopDir >= OppStopDir Then
    If LoopListChar = "" Then
      LoopListChar = "take bus:" & Service(i)
      CheckRepeatLoop = Service(i)
    ElseIf LoopListChar <> "" Then
      If CheckRepeatLoop <> Service(i) Then
        LoopListChar = LoopListChar & ", " & Service(i)
      End If
    End If
  ElseIf OppStopDir > StopDir Then
    If LoopListChar = "" Then
      LoopListChar = "go opposite bus stop, take bus:" & Service(i)
      CheckRepeatLoop = Service(i)
    ElseIf LoopListChar <> "" Then
      If CheckRepeatLoop <> Service(i) Then
        LoopListChar = LoopListChar & ", " & Service(i)
      End If
    End If
  End If
ElseIf ToDir <> OppToDir Then

  If ToDir > StopDir Then
    If LoopListChar = "" Then
      LoopListChar = "take bus:" & Service(i)
      CheckRepeatLoop = Service(i)
    ElseIf LoopListChar <> "" Then
      If CheckRepeatLoop <> Service(i) Then
        LoopListChar = LoopListChar & ", " & Service(i)
      End If
    End If

  ElseIf ToDir < StopDir Then
    If LoopListChar = "" Then
      LoopListChar = "go opposite bus stop, take bus:" & Service(i)
      CheckRepeatLoop = Service(i)
    ElseIf LoopListChar <> "" Then
      If CheckRepeatLoop <> Service(i) Then
        LoopListChar = LoopListChar & ", " & Service(i)
      End If
    End If
  End If
End If
End If
Next
End If
Next

```



```

If LoopListChar = "" Then
    SMSText.Text = "no results"
Elseif LoopListChar <> "" Then
    SMSText.Text = LoopListChar
End If

If CheckRegionMatch = 0 Then
    Call FarLoopList
    Exit Sub
End If
End Sub
Private Sub DualList()

    Dim CheckRepeatDual, CheckDualDir1, CheckDualDir2
    ToDir = ""
    OppToDir = ""
    OppStopNo = ""
    StopNo = ""
    StopDir = ""
    OppStopDir = ""
    DualListChar = ""
    CheckDualDir1 = 0
    CheckDualDir2 = 0

    For i = 1 To 7
        If Service(i) <> "" Then
            ToDir = ""
            OppToDir = ""
            OppStopNo = ""
            StopNo = ""
            StopDir = ""
            OppStopDir = ""
            For j = 0 To 10
                If ToRoad(j) <> "" Then
                    ServiceNo = Service(i) & "dir1"
                    Set ConnBus = New ADODB.Connection
                    ConnBus.Open ServiceNo
                    Set DualListRecordset = New ADODB.Recordset
                    DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
                    Do Until DualListRecordset.EOF
                        If ToRoad(j) = DualListRecordset("road name") Then

                            If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
                                ToDir = DualListRecordset("opp direction")
                                OppToDir = DualListRecordset("opp direction")
                            Elseif DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
                                ToDir = DualListRecordset("direction")
                                OppToDir = DualListRecordset("opp direction")
                            End If

                        End If
                    If BusStopNo = DualListRecordset("bus stop no") Then
                        CheckDualDir1 = 1
                        StopNo = BusStopNo
                        OppStopNo = DualListRecordset("opp bus stop no")
                        StopDir = DualListRecordset("direction")
                        OppStopDir = DualListRecordset("opp direction")
                    End If
                    DualListRecordset.MoveNext
                Loop
                If CheckDualDir1 = 1 Then
                    Set DualListRecordset = New ADODB.Recordset
                    DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
                    Do Until DualListRecordset.EOF
                        If ToRoad(j) = DualListRecordset("road name") Then

                            If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
                                ToDir = DualListRecordset("opp direction")
                                OppToDir = DualListRecordset("opp direction")
                            Elseif DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
                                ToDir = DualListRecordset("direction")
                                OppToDir = DualListRecordset("opp direction")
                            End If

                        End If
                    End If
                    DualListRecordset.MoveNext
                End If
            Next j
        End If
    Next i

```

```
Loop
End If
```

```
ServiceNo = Service(i) & "dir2"
Set ConnBus = New ADODB.Connection
ConnBus.Open ServiceNo
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
```

```
If BusStopNo = DualListRecordset("bus stop no") Then
    CheckDualDir2 = 1
    StopNo = BusStopNo
    OppStopNo = DualListRecordset("opp bus stop no")
    StopDir = DualListRecordset("direction")
    OppStopDir = DualListRecordset("opp direction")
End If
```

```
DualListRecordset.MoveNext
Loop
```

```
If CheckDualDir2 = 1 Then
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ServiceNo & " ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
```

```
    If ToRoad(j) = DualListRecordset("road name") Then
```

```
        If DualListRecordset("bus stop no") = DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("opp direction")
            OppToDir = DualListRecordset("opp direction")
        ElseIf DualListRecordset("bus stop no") <> DualListRecordset("opp bus stop no") Then
            ToDir = DualListRecordset("direction")
            OppToDir = DualListRecordset("opp direction")
        End If
```

```
    End If
```

```
DualListRecordset.MoveNext
```

```
Loop
End If
```

```
If CheckDualDir1 = 1 Then
```

```
    If OppToDir = ToDir Then
```

```
        If StopDir >= OppStopDir Then
```

```
            If DualListChar = "" Then
```

```
                DualListChar = "take bus:" & Service(i)
```

```
                CheckRepeatDual = Service(i)
```

```
            ElseIf DualListChar <> "" Then
```

```
                If CheckRepeatDual <> Service(i) Then
```

```
                    DualListChar = DualListChar & ", " & Service(i)
```

```
                End If
```

```
            End If
```

```
        ElseIf OppStopDir > StopDir Then
```

```
            If DualListChar = "" Then
```

```
                DualListChar = "go opposite bus stop, take bus:" & Service(i)
```

```
                CheckRepeatDual = Service(i)
```

```
            ElseIf DualListChar <> "" Then
```

```
                If CheckRepeatDual <> Service(i) Then
```

```
                    DualListChar = DualListChar & ", " & Service(i)
```

```
                End If
```

```
            End If
```

```
        End If
```

```
    ElseIf ToDir <> OppToDir Then
```

```
        If ToDir > StopDir Then
```

```
            If DualListChar = "" Then
```

```
                DualListChar = "take bus:" & Service(i)
```

```
                CheckRepeatDual = Service(i)
```

```
            ElseIf DualListChar <> "" Then
```

```
                If CheckRepeatDual <> Service(i) Then
```

```
                    DualListChar = DualListChar & ", " & Service(i)
```

```
                End If
```

```
            End If
```

```
        ElseIf ToDir < StopDir Then
```

```
            If DualListChar = "" Then
```

```

        DualListChar = "go opposite bus stop, take bus:" & Service(i)
        CheckRepeatDual = Service(i)
        ElseIf DualListChar <> "" Then
            If CheckRepeatDual <> Service(i) Then
                DualListChar = LoopListChar & ", " & Service(i)
            End If
        End If
    End If
End If
End If
CheckDualDir1 = 0

ElseIf CheckDualDir2 = 1 Then
    If OppToDir = ToDir Then
        If StopDir >= OppStopDir Then
            If DualListChar = "" Then
                DualListChar = "take bus:" & Service(i)
                CheckRepeatDual = Service(i)
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> Service(i) Then
                    DualListChar = DualListChar & ", " & Service(i)
                End If
            End If
        ElseIf OppStopDir > StopDir Then
            If DualListChar = "" Then
                DualListChar = "go opposite bus stop, take bus:" & Service(i)
                CheckRepeatDual = Service(i)
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> Service(i) Then
                    DualListChar = DualListChar & ", " & Service(i)
                End If
            End If
        End If
    ElseIf ToDir <> OppToDir Then

        If ToDir > StopDir Then
            If DualListChar = "" Then
                DualListChar = "take bus:" & Service(i)
                CheckRepeatDual = Service(i)
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> Service(i) Then
                    DualListChar = DualListChar & ", " & Service(i)
                End If
            End If

        ElseIf ToDir < StopDir Then
            If DualListChar = "" Then
                DualListChar = "go opposite bus stop, take bus:" & Service(i)
                CheckRepeatDual = Service(i)
            ElseIf DualListChar <> "" Then
                If CheckRepeatDual <> Service(i) Then
                    DualListChar = DualListChar & ", " & Service(i)
                End If
            End If
        End If
    End If
    CheckDualDir2 = 0

End If
End If
Next
End If
Next

If DualListChar = "" Then
    SMSText.Text = "no results"
ElseIf DualListChar <> "" Then
    SMSText.Text = DualListChar
End If
If CheckRegionMatch = 0 Then
    Call FarDualList
    Exit Sub
End If
End Sub
Private Sub MrtToMrt()

```

```

Dim FromMrtDirection, ToMrtDirection, Fromnsew, Tonsew
Dim FromInterchangeDirection, ToInterchangeDirection
FromMrtDirection = ""
ToMrtDirection = ""
Fromnsew = ""
Tonsew = ""
FromInterchangeDirection = ""
ToInterchangeDirection = ""
Set ConnSmrt = New ADODB.Connection
ConnSmrt.Open "smrt"
Set SmrtRecordset = New ADODB.Recordset
SmrtRecordset.Open "SELECT * FROM smrt", ConnSmrt

Do Until SmrtRecordset.EOF

    If FromMrt = SmrtRecordset("mrt name") Then
        FromMrtDirection = SmrtRecordset("direction")
        Fromnsew = SmrtRecordset("nsew")
        If SmrtRecordset("connection") <> "" Then
            FromInterchangeDirection = SmrtRecordset("connection")
        End If
    End If
    If ToMrt = SmrtRecordset("mrt name") Then
        ToMrtDirection = SmrtRecordset("direction")
        Tonsew = SmrtRecordset("nsew")
        If SmrtRecordset("connection") <> "" Then
            ToInterchangeDirection = SmrtRecordset("connection")
        End If
    End If

SmrtRecordset.MoveNext
Loop

If FromInterchangeDirection <> "" And ToInterchangeDirection = "" Then
    If ToMrtDirection = "ew1" Or ToMrtDirection = "ew2" Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    ElseIf ToMrtDirection = "ns1" Then
        If Tonsew = 1 Then
            SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
        End If
    ElseIf ToMrtDirection = "ns2" Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    End If
ElseIf FromInterchangeDirection <> "" And ToInterchangeDirection <> "" Then
    SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
ElseIf FromInterchangeDirection = "" And ToInterchangeDirection <> "" Then
    If FromMrtDirection = "ew1" Or FromMrtDirection = "ew2" Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    ElseIf FromMrtDirection = "ns1" Then
        If Fromnsew = 1 Then
            SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
        End If
    ElseIf FromMrtDirection = "ns2" Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    End If
ElseIf FromInterchangeDirection = "" And ToInterchangeDirection = "" Then
    If (FromMrtDirection = "ew1" Or FromMrtDirection = "ew2") And (ToMrtDirection = "ew1" Or ToMrtDirection = "ew2")
Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    ElseIf (FromMrtDirection = "ns1" Or FromMrtDirection = "ns2") And (ToMrtDirection = "ns1" Or ToMrtDirection = "ns2")
Then
        SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
    ElseIf (FromMrtDirection = "ew1" And ToMrtDirection = "ns2") Or (FromMrtDirection = "ns2" And ToMrtDirection =
"ew1") Then
        SMSText.Text = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
    ElseIf (FromMrtDirection = "ew2" And ToMrtDirection = "ns2") Or (FromMrtDirection = "ns2" And ToMrtDirection =
"ew2") Then
        SMSText.Text = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
    ElseIf FromMrtDirection = "ew1" And ToMrtDirection = "ns1" Then
        If Tonsew = 1 Then
            SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
        Else
            SMSText.Text = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
        End If
    ElseIf FromMrtDirection = "ns1" And ToMrtDirection = "ew1" Then

```

```

    If Fromnsew = 1 Then
        SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
    Else
        SMSText.Text = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
    End If
ElseIf FromMrtDirection = "ew2" And ToMrtDirection = "ns1" Then
    If (Fromnsew - Tonsew) > 10 Then
        If FromMrt = "jurong east" Then
            SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
        Else
            SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
        End If
    ElseIf (Fromnsew - Tonsew) <= 10 Then
        SMSText.Text = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
    End If
ElseIf FromMrtDirection = "ns1" And ToMrtDirection = "ew2" Then
    If (Tonsew - Fromnsew) > 10 Then
        If ToMrt = "jurong east" Then
            SMSText.Text = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
        Else
            SMSText.Text = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
        End If
    ElseIf (Tonsew - Fromnsew) <= 10 Then
        SMSText.Text = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
    End If
End If
End Sub

Private Sub MrtListInfo()
    Dim SmrtBusList(10)
    Dim CheckSmrtBusList
    CheckSmrtBusList = ""
    For i = 1 To 7
        SmrtBusList(i) = ""
    Next
    Set ConnSmrt = New ADODB.Connection
    ConnSmrt.Open "smrt"
    Set SmrtRecordset = New ADODB.Recordset
    SmrtRecordset.Open "SELECT * FROM smrt", ConnSmrt

    Do Until SmrtRecordset.EOF

        If FromMrt = SmrtRecordset("mrt name") Then
            For i = 1 To 7
                If SmrtRecordset("service" & i) <> "" Then
                    For j = 1 To 7
                        If SmrtBusList(j) = "" Then
                            SmrtBusList(j) = SmrtRecordset("service" & i)
                            Exit For
                        End If
                    Next
                End If
            Next
        End If
    Next
    SmrtRecordset.MoveNext
Loop

    If SmrtBusList(1) = "" Then
        SMSText.Text = "No Results"
    ElseIf SmrtBusList(1) <> "" Then
        SMSText.Text = "available bus services at " & FromMrt & ": " & SmrtBusList(1)
        For i = 2 To 7
            If SmrtBusList(i) <> "" Then
                SMSText.Text = SMSText.Text & ", " & SmrtBusList(i)
            End If
        Next
    End If
End Sub

Private Sub FarRoadtoRoad()

    Dim FromDir1, ToDir1, FromDir2, ToDir2
    FromMrtSpot = 0
    ToMrtSpot = 0
    FromMrtRoad = ""
    ToMrtRoad = ""

```

```

Set BusService = New ADODB.Connection
BusService.Open "busdir"
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
Do Until ServiceRecordset.EOF
    If Service(1) = ServiceRecordset("service no") Then
        CheckDual = ServiceRecordset("dual direction")
        Exit Do
    End If
ServiceRecordset.MoveNext
Loop

If CheckDual = 0 Then
    Set ConnBus = New ADODB.Connection
    ConnBus.Open Service(1)
    Set LoopListRecordset = New ADODB.Recordset
    LoopListRecordset.Open "SELECT * FROM " & Service(1) & " ORDER BY [direction]", ConnBus
    Do Until LoopListRecordset.EOF
        For i = 0 To 10
            If LoopListRecordset("road name") = FromRoad(i) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If LoopListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = LoopListRecordset("mrt name")
                FromMrtRoad = LoopListRecordset("road name")
            End If
        End If
        If ToMrtSpot = 0 And FromMrt <> "" Then
            If LoopListRecordset("mrt id") <> "no" And FromMrt <> LoopListRecordset("mrt name") Then
                ToMrt = LoopListRecordset("mrt name")
                ToMrtRoad = LoopListRecordset("road name")
            End If
        End If
        For i = 0 To 10
            If LoopListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
                ToMrtSpot = 1
                Exit For
            End If
        Next
        LoopListRecordset.MoveNext
    Loop
ElseIf CheckDual = 1 Then
    Set ConnBus = New ADODB.Connection
    ConnBus.Open Service(1) & "dir1"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir1 ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        If FromRoad(0) = DualListRecordset("road name") Then
            FromDir1 = DualListRecordset("direction")
        End If
        If ToRoad(0) = DualListRecordset("road name") Then
            ToDir1 = DualListRecordset("direction")
        End If
        DualListRecordset.MoveNext
    Loop
    Set ConnBus = New ADODB.Connection
    ConnBus.Open Service(1) & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir2 ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        If FromRoad(0) = DualListRecordset("road name") Then
            FromDir2 = DualListRecordset("direction")
        End If
        If ToRoad(0) = DualListRecordset("road name") Then
            ToDir2 = DualListRecordset("direction")
        End If
        DualListRecordset.MoveNext
    Loop
    If ToDir1 > FromDir1 Then
        Set ConnBus = New ADODB.Connection
        ConnBus.Open Service(1) & "dir1"
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir1 ORDER BY [direction]", ConnBus
    End If

```

```

Do Until DualListRecordset.EOF
  For i = 0 To 10
    If DualListRecordset("road name") = FromRoad(i) Then
      FromMrtSpot = 1
      Exit For
    End If
  Next
  If FromMrtSpot = 1 Then
    If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
      FromMrt = DualListRecordset("mrt name")
      FromMrtRoad = DualListRecordset("road name")
    End If
  End If
  If ToMrtSpot = 0 And FromMrt <> "" Then
    If DualListRecordset("mrt id") <> "no" And FromMrt <> DualListRecordset("mrt name") Then
      ToMrt = DualListRecordset("mrt name")
      ToMrtRoad = DualListRecordset("road name")
    End If
  End If
  For i = 0 To 10
    If DualListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
      ToMrtSpot = 1
      Exit For
    End If
  Next
  DualListRecordset.MoveNext
Loop
End If
If ToDir2 > FromDir2 Then
  Set ConnBus = New ADODB.Connection
  ConnBus.Open Service(1) & "dir2"
  Set DualListRecordset = New ADODB.Recordset
  DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir2 ORDER BY [direction]", ConnBus
  Do Until DualListRecordset.EOF
    For i = 0 To 10
      If DualListRecordset("road name") = FromRoad(i) Then
        FromMrtSpot = 1
        Exit For
      End If
    Next
    If FromMrtSpot = 1 Then
      If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
        FromMrt = DualListRecordset("mrt name")
        FromMrtRoad = DualListRecordset("road name")
      End If
    End If
    If ToMrtSpot = 0 And FromMrt <> "" Then
      If DualListRecordset("mrt id") <> "no" And FromMrt <> DualListRecordset("mrt name") Then
        ToMrt = DualListRecordset("mrt name")
        ToMrtRoad = DualListRecordset("road name")
      End If
    End If
    For i = 0 To 10
      If DualListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
        ToMrtSpot = 1
        Exit For
      End If
    Next
    DualListRecordset.MoveNext
  Loop
End If
End If
If FromMrt = "" Or ToMrt = "" Then
  Exit Sub
Else
  Call MrtTransfer
  ServiceChar = ""
  For i = 1 To 7
    If Service(i) <> "" And ServiceChar = "" Then
      ServiceChar = Service(i)
    ElseIf Service(i) <> "" And ServiceChar <> "" Then
      ServiceChar = ServiceChar & ", " & Service(i)
    End If
  Next
  SMSText.Text = SMSText.Text & ", Or " & "at " & FromMrtRoad & ", " & MrtText & ", at " & ToMrtRoad & ", take " & ServiceChar
End If

```

```

End Sub
Private Sub FarTransferService()

    Dim FromMrtTemp, ToMrtTemp
    Dim FromDir1, ToDir1, FromDir2, ToDir2
    FromMrtSpot = 0
    ToMrtSpot = 0
    FromMrtRoad = ""
    ToMrtRoad = ""
    'check fromservice route for mrt
    For i = 0 To 10
        Set BusService = New ADODB.Connection
        BusService.Open "busdir"
        Set ServiceRecordset = New ADODB.Recordset
        ServiceRecordset.Open "SELECT * FROM busdir", BusService
        Do Until ServiceRecordset.EOF
            If FromTrans(i) = ServiceRecordset("service no") Then
                CheckDual = ServiceRecordset("dual direction")
                Exit Do
            End If
        ServiceRecordset.MoveNext
    Loop
    If CheckDual = 0 Then
        Set ConnBus = New ADODB.Connection
        If FromTrans(i) <> "" Then
            ConnBus.Open FromTrans(i)
            Set LoopListRecordset = New ADODB.Recordset
            LoopListRecordset.Open "SELECT * FROM " & FromTrans(i) & " ORDER BY [direction]", ConnBus
        End If
        Do Until LoopListRecordset.EOF
            For j = 0 To 10
                If LoopListRecordset("road name") = FromRoad(j) Then
                    FromMrtSpot = 1
                    Exit For
                End If
            Next
            If FromMrtSpot = 1 Then
                If LoopListRecordset("mrt id") <> "no" And FromMrt = "" Then
                    FromMrt = LoopListRecordset("mrt name")
                    FromMrtRoad = LoopListRecordset("road name")
                End If
            End If
        LoopListRecordset.MoveNext
    Loop
    ElseIf CheckDual = 1 Then
        Set ConnBus = New ADODB.Connection
        If FromTrans(i) <> "" Then
            ConnBus.Open FromTrans(i) & "dir1"
            Set DualListRecordset = New ADODB.Recordset
            DualListRecordset.Open "SELECT * FROM " & FromTrans(i) & "dir1 ORDER BY [direction]", ConnBus
        End If
        Do Until DualListRecordset.EOF
            If FromRoad(0) = DualListRecordset("road name") Then
                FromDir1 = DualListRecordset("direction")
            End If
            If TransRoad(0) = DualListRecordset("road name") Then
                ToDir1 = DualListRecordset("direction")
            End If
        DualListRecordset.MoveNext
    Loop
        Set ConnBus = New ADODB.Connection
        If FromTrans(i) <> "" Then
            ConnBus.Open FromTrans(i) & "dir2"
            Set DualListRecordset = New ADODB.Recordset
            DualListRecordset.Open "SELECT * FROM " & FromTrans(i) & "dir2 ORDER BY [direction]", ConnBus
        End If
        Do Until DualListRecordset.EOF
            If FromRoad(0) = DualListRecordset("road name") Then
                FromDir2 = DualListRecordset("direction")
            End If
            If TransRoad(0) = DualListRecordset("road name") Then
                ToDir2 = DualListRecordset("direction")
            End If
        DualListRecordset.MoveNext
    Loop

```



```

If ToDir1 > FromDir1 Then
Set ConnBus = New ADODB.Connection
If FromTrans(i) <> "" Then
ConnBus.Open FromTrans(i) & "dir1"
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & FromTrans(i) & "dir1 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
For j = 0 To 10
If DualListRecordset("road name") = FromRoad(j) Then
FromMrtSpot = 1
Exit For
End If
Next
If FromMrtSpot = 1 Then
If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
FromMrt = DualListRecordset("mrt name")
FromMrtRoad = DualListRecordset("road name")
End If
End If
DualListRecordset.MoveNext
Loop
End If
If ToDir2 > FromDir2 Then
Set ConnBus = New ADODB.Connection
If FromTrans(i) <> "" Then
ConnBus.Open FromTrans(i) & "dir2"
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & FromTrans(i) & "dir2 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
For j = 0 To 10
If DualListRecordset("road name") = FromRoad(j) Then
FromMrtSpot = 1
Exit For
End If
Next
If FromMrtSpot = 1 Then
If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
FromMrt = DualListRecordset("mrt name")
FromMrtRoad = DualListRecordset("road name")
End If
End If
DualListRecordset.MoveNext
Loop
End If
End If
Next

'check toservice route for mrt

For i = 0 To 10
Set BusService = New ADODB.Connection
BusService.Open "busdir"
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
Do Until ServiceRecordset.EOF
If ToTrans(i) = ServiceRecordset("service no") Then
CheckDual = ServiceRecordset("dual direction")
Exit Do
End If
ServiceRecordset.MoveNext
Loop
If CheckDual = 0 Then
Set ConnBus = New ADODB.Connection
If ToTrans(i) <> "" Then
ConnBus.Open ToTrans(i)
Set LoopListRecordset = New ADODB.Recordset
LoopListRecordset.Open "SELECT * FROM " & ToTrans(i) & " ORDER BY [direction]", ConnBus
End If
Do Until LoopListRecordset.EOF
For j = 0 To 10
If LoopListRecordset("road name") = ToRoad(j) Then
ToMrtSpot = 1
Exit For
End If

```

```

Next
If ToMrtSpot = 0 Then
  If LoopListRecordset("mrt id") <> "no" Then
    ToMrt = LoopListRecordset("mrt name")
    ToMrtRoad = LoopListRecordset("road name")
  End If
End If
If ToMrtSpot = 1 Then
  Exit Do
End If
LoopListRecordset.MoveNext
Loop
Elseif CheckDual = 1 Then
  Set ConnBus = New ADODB.Connection
  If ToTrans(i) <> "" Then
    ConnBus.Open ToTrans(i) & "dir1"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ToTrans(i) & "dir1 ORDER BY [direction]", ConnBus
  End If
  Do Until DualListRecordset.EOF
    If TransRoad(0) = DualListRecordset("road name") Then
      FromDir1 = DualListRecordset("direction")
    End If
    If ToRoad(0) = DualListRecordset("road name") Then
      ToDir1 = DualListRecordset("direction")
    End If
  DualListRecordset.MoveNext
  Loop
  Set ConnBus = New ADODB.Connection
  If ToTrans(i) <> "" Then
    ConnBus.Open ToTrans(i) & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ToTrans(i) & "dir2 ORDER BY [direction]", ConnBus
  End If
  Do Until DualListRecordset.EOF
    If TransRoad(0) = DualListRecordset("road name") Then
      FromDir2 = DualListRecordset("direction")
    End If
    If ToRoad(0) = DualListRecordset("road name") Then
      ToDir2 = DualListRecordset("direction")
    End If
  DualListRecordset.MoveNext
  Loop
  If ToDir1 > FromDir1 Then
    Set ConnBus = New ADODB.Connection
    If ToTrans(i) <> "" Then
      ConnBus.Open ToTrans(i) & "dir1"
      Set DualListRecordset = New ADODB.Recordset
      DualListRecordset.Open "SELECT * FROM " & ToTrans(i) & "dir1 ORDER BY [direction]", ConnBus
    End If
    Do Until DualListRecordset.EOF
      For j = 0 To 10
        If DualListRecordset("road name") = ToRoad(j) Then
          ToMrtSpot = 1
          Exit For
        End If
      Next
      If ToMrtSpot = 0 Then
        If DualListRecordset("mrt id") <> "no" Then
          ToMrt = DualListRecordset("mrt name")
          ToMrtRoad = DualListRecordset("road name")
        End If
      End If
      If ToMrtSpot = 1 Then
        Exit Do
      End If
    DualListRecordset.MoveNext
  Loop
End If
If ToDir2 > FromDir2 Then
  Set ConnBus = New ADODB.Connection
  If ToTrans(i) <> "" Then
    ConnBus.Open ToTrans(i) & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ToTrans(i) & "dir2 ORDER BY [direction]", ConnBus
  End If

```

```

Do Until DualListRecordset.EOF
  For j = 0 To 10
    If DualListRecordset("road name") = ToRoad(j) Then
      ToMrtSpot = 1
      Exit For
    End If
  Next
  If ToMrtSpot = 0 Then
    If DualListRecordset("mrt id") <> "no" Then
      ToMrt = DualListRecordset("mrt name")
      ToMrtRoad = DualListRecordset("road name")
    End If
  End If
  If ToMrtSpot = 1 Then
    Exit Do
  End If
  DualListRecordset.MoveNext
Loop
End If
End If
Next

'combine results

If FromMrt = "" Or ToMrt = "" Then
  Exit Sub
Else
  Call MrtTransfer
  ServiceChar = ""
  For i = 1 To 7
    If ToTrans(i) <> "" And ServiceChar = "" Then
      ServiceChar = ToTrans(i)
    ElseIf ToTrans(i) <> "" And ServiceChar <> "" Then
      ServiceChar = ServiceChar & ", " & ToTrans(i)
    End If
  Next
  SMSText.Text = SMSText.Text & ", Or " & "at " & FromMrtRoad & ", " & MrtText & ", at " & ToMrtRoad & ", take " &
ServiceChar
End If

End Sub
Private Sub MrtTransfer()

  Dim FromMrtDirection, ToMrtDirection, Fromnsew, Tonsew
  Dim FromInterchangeDirection, ToInterchangeDirection
  FromMrtDirection = ""
  ToMrtDirection = ""
  Fromnsew = ""
  Tonsew = ""
  FromInterchangeDirection = ""
  ToInterchangeDirection = ""
  Set ConnSmrt = New ADODB.Connection
  ConnSmrt.Open "smrt"
  Set SmrtRecordset = New ADODB.Recordset
  SmrtRecordset.Open "SELECT * FROM smrt", ConnSmrt

  Do Until SmrtRecordset.EOF

    If FromMrt = SmrtRecordset("mrt name") Then
      FromMrtDirection = SmrtRecordset("direction")
      Fromnsew = SmrtRecordset("nsew")
      If SmrtRecordset("connection") <> "" Then
        FromInterchangeDirection = SmrtRecordset("connection")
      End If
    End If
    If ToMrt = SmrtRecordset("mrt name") Then
      ToMrtDirection = SmrtRecordset("direction")
      Tonsew = SmrtRecordset("nsew")
      If SmrtRecordset("connection") <> "" Then
        ToInterchangeDirection = SmrtRecordset("connection")
      End If
    End If

    SmrtRecordset.MoveNext
  Loop

```

```

If FromInterchangeDirection <> "" And ToInterchangeDirection = "" Then
  If ToMrtDirection = "ew1" Or ToMrtDirection = "ew2" Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  ElseIf ToMrtDirection = "ns1" Then
    If Tonsew = 1 Then
      MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
    End If
  ElseIf ToMrtDirection = "ns2" Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  End If
ElseIf FromInterchangeDirection <> "" And ToInterchangeDirection <> "" Then
  MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
ElseIf FromInterchangeDirection = "" And ToInterchangeDirection <> "" Then
  If FromMrtDirection = "ew1" Or FromMrtDirection = "ew2" Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  ElseIf FromMrtDirection = "ns1" Then
    If Fromnsew = 1 Then
      MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
    End If
  ElseIf FromMrtDirection = "ns2" Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  End If
ElseIf FromInterchangeDirection = "" And ToInterchangeDirection = "" Then
  If (FromMrtDirection = "ew1" Or FromMrtDirection = "ew2") And (ToMrtDirection = "ew1" Or ToMrtDirection = "ew2")
  Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  ElseIf (FromMrtDirection = "ns1" Or FromMrtDirection = "ns2") And (ToMrtDirection = "ns1" Or ToMrtDirection = "ns2")
  Then
    MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
  ElseIf (FromMrtDirection = "ew1" And ToMrtDirection = "ns2") Or (FromMrtDirection = "ns2" And ToMrtDirection =
  "ew1") Then
    MrtText = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
  ElseIf (FromMrtDirection = "ew2" And ToMrtDirection = "ns2") Or (FromMrtDirection = "ns2" And ToMrtDirection =
  "ew2") Then
    MrtText = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
  ElseIf FromMrtDirection = "ew1" And ToMrtDirection = "ns1" Then
    If Tonsew = 1 Then
      MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
    Else
      MrtText = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
    End If
  ElseIf FromMrtDirection = "ns1" And ToMrtDirection = "ew1" Then
    If Fromnsew = 1 Then
      MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
    Else
      MrtText = "take " & FromMrt & " mrt, transferring at city hall mrt, to " & ToMrt & " mrt"
    End If
  ElseIf FromMrtDirection = "ew2" And ToMrtDirection = "ns1" Then
    If (Fromnsew - Tonsew) > 10 Then
      If FromMrt = "jurong east" Then
        MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
      Else
        MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
      End If
    ElseIf (Fromnsew - Tonsew) <= 10 Then
      MrtText = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
    End If
  ElseIf FromMrtDirection = "ns1" And ToMrtDirection = "ew2" Then
    If (Tonsew - Fromnsew) > 10 Then
      If ToMrt = "jurong east" Then
        MrtText = "take " & FromMrt & " mrt to " & ToMrt & " mrt"
      Else
        MrtText = "take " & FromMrt & " mrt, transferring at jurong east mrt, to " & ToMrt & " mrt"
      End If
    ElseIf (Tonsew - Fromnsew) <= 10 Then
      MrtText = "take " & FromMrt & " mrt, transferring at raffles place mrt, to " & ToMrt & " mrt"
    End If
  End If
End If

End Sub
Private Sub FarLoopList()

  FromMrtSpot = 0
  ToMrtSpot = 0

```

```

FromMrtRoad = ""
ToMrtRoad = ""

Set BusService = New ADODB.Connection
BusService.Open "busdir"
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
Do Until ServiceRecordset.EOF
    If Service(1) = ServiceRecordset("service no") Then
        CheckDual = ServiceRecordset("dual direction")
        Exit Do
    End If
ServiceRecordset.MoveNext
Loop

Set ConnBus = New ADODB.Connection
ConnBus.Open Service(1)
Set LoopListRecordset = New ADODB.Recordset
LoopListRecordset.Open "SELECT * FROM " & Service(1) & " ORDER BY [direction]", ConnBus
Do Until LoopListRecordset.EOF
    For i = 0 To 10
        If LoopListRecordset("road name") = FromRoad(i) Then
            FromMrtSpot = 1
            Exit For
        End If
    Next
    If FromMrtSpot = 1 Then
        If LoopListRecordset("mrt id") <> "no" And FromMrt = "" Then
            FromMrt = LoopListRecordset("mrt name")
            FromMrtRoad = LoopListRecordset("road name")
        End If
    End If
    If ToMrtSpot = 0 And FromMrt <> "" Then
        If LoopListRecordset("mrt id") <> "no" And FromMrt <> LoopListRecordset("mrt name") Then
            ToMrt = LoopListRecordset("mrt name")
            ToMrtRoad = LoopListRecordset("road name")
        End If
    End If
    For i = 0 To 10
        If LoopListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
            ToMrtSpot = 1
        End If
    Next
LoopListRecordset.MoveNext
Loop
If FromMrt = "" Or ToMrt = "" Then
    Exit Sub
Else
    Call MrtTransfer
    ServiceChar = ""
    For i = 1 To 7
        If Service(i) <> "" And ServiceChar = "" Then
            ServiceChar = Service(i)
        ElseIf Service(i) <> "" And ServiceChar <> "" Then
            ServiceChar = ServiceChar & ", " & Service(i)
        End If
    Next
    SMSText.Text = SMSText.Text & ", Or " & "at " & FromMrtRoad & ", " & MrtText & ", at " & ToMrtRoad & ", take " &
ServiceChar
End If
End Sub
Private Sub FarDualList()
    Dim FromDir1, ToDir1, FromDir2, ToDir2
    FromMrtSpot = 0
    ToMrtSpot = 0
    FromMrtRoad = ""
    ToMrtRoad = ""

    Set BusService = New ADODB.Connection
    BusService.Open "busdir"
    Set ServiceRecordset = New ADODB.Recordset
    ServiceRecordset.Open "SELECT * FROM busdir", BusService
    Do Until ServiceRecordset.EOF
        If Service(1) = ServiceRecordset("service no") Then
            CheckDual = ServiceRecordset("dual direction")
            Exit Do
        End If
    End If

```

```

End If
ServiceRecordset.MoveNext
Loop
Set ConnBus = New ADODB.Connection
ConnBus.Open Service(1) & "dir1"
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir1 ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
    If FromRoad(0) = DualListRecordset("road name") Then
        FromDir1 = DualListRecordset("direction")
    End If
    If ToRoad(0) = DualListRecordset("road name") Then
        ToDir1 = DualListRecordset("direction")
    End If
DualListRecordset.MoveNext
Loop
Set ConnBus = New ADODB.Connection
ConnBus.Open Service(1) & "dir2"
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir2 ORDER BY [direction]", ConnBus
Do Until DualListRecordset.EOF
    If FromRoad(0) = DualListRecordset("road name") Then
        FromDir2 = DualListRecordset("direction")
    End If
    If ToRoad(0) = DualListRecordset("road name") Then
        ToDir2 = DualListRecordset("direction")
    End If
DualListRecordset.MoveNext
Loop
If ToDir1 > FromDir1 Then
    Set ConnBus = New ADODB.Connection
    ConnBus.Open Service(1) & "dir1"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir1 ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        For i = 0 To 10
            If DualListRecordset("road name") = FromRoad(i) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = DualListRecordset("mrt name")
                FromMrtRoad = DualListRecordset("road name")
            End If
        End If
        If ToMrtSpot = 0 And FromMrt <> "" Then
            If DualListRecordset("mrt id") <> "no" And FromMrt <> DualListRecordset("mrt name") Then
                ToMrt = DualListRecordset("mrt name")
                ToMrtRoad = DualListRecordset("road name")
            End If
        End If
        For i = 0 To 10
            If DualListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
                ToMrtSpot = 1
            End If
        Next
    DualListRecordset.MoveNext
Loop
End If
If ToDir2 > FromDir2 Then
    Set ConnBus = New ADODB.Connection
    ConnBus.Open Service(1) & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & Service(1) & "dir2 ORDER BY [direction]", ConnBus
    Do Until DualListRecordset.EOF
        For i = 0 To 10
            If DualListRecordset("road name") = FromRoad(i) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = DualListRecordset("mrt name")
            End If
        End If
    End If

```

```

        FromMrtRoad = DualListRecordset("road name")
    End If
End If
If ToMrtSpot = 0 And FromMrt <> "" Then
    If DualListRecordset("mrt id") <> "no" And FromMrt <> DualListRecordset("mrt name") Then
        ToMrt = DualListRecordset("mrt name")
        ToMrtRoad = DualListRecordset("road name")
    End If
End If
For i = 0 To 10
    If DualListRecordset("road name") = ToRoad(i) And ToMrt = "" Then
        ToMrtSpot = 1
    End If
Next
DualListRecordset.MoveNext
Loop
End If

If FromMrt = "" Or ToMrt = "" Then
    Exit Sub
Else
    Call MrtTransfer
    ServiceChar = ""
    For i = 1 To 7
        If Service(i) <> "" And ServiceChar = "" Then
            ServiceChar = Service(i)
        ElseIf Service(i) <> "" And ServiceChar <> "" Then
            ServiceChar = ServiceChar & ", " & Service(i)
        End If
    Next
    SMSText.Text = SMSText.Text & ", Or " & "at " & FromMrtRoad & ", " & MrtText & ", at " & ToMrtRoad & ", take " &
    ServiceChar
End If

End Sub

Private Sub FarStopToRoad()
    Dim FromMrtTemp, ToMrtTemp
    Dim FromDir1, ToDir1, FromDir2, ToDir2
    FromMrtSpot = 0
    ToMrtSpot = 0
    FromMrtRoad = ""
    ToMrtRoad = ""
    'check fromservice route for mrt

    Set BusService = New ADODB.Connection
    BusService.Open "busdir"
    Set ServiceRecordset = New ADODB.Recordset
    ServiceRecordset.Open "SELECT * FROM busdir", BusService
    Do Until ServiceRecordset.EOF
        If FromCharService = ServiceRecordset("service no") Then
            CheckDual = ServiceRecordset("dual direction")
            Exit Do
        End If
    ServiceRecordset.MoveNext
Loop
If CheckDual = 0 Then
    Set ConnBus = New ADODB.Connection
    If FromCharService <> "" Then
        ConnBus.Open FromCharService
        Set LoopListRecordset = New ADODB.Recordset
        LoopListRecordset.Open "SELECT * FROM " & FromCharService & " ORDER BY [direction]", ConnBus
    End If
    Do Until LoopListRecordset.EOF
        For j = 0 To 10
            If LoopListRecordset("road name") = FromRoad(j) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If LoopListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = LoopListRecordset("mrt name")
                FromMrtRoad = LoopListRecordset("road name")
            End If
        End If
    End If

```

```

LoopListRecordset.MoveNext
Loop
Elseif CheckDual = 1 Then
Set ConnBus = New ADODB.Connection
If FromCharService <> "" Then
    ConnBus.Open FromCharService & "dir1"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & FromCharService & "dir1 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
    If FromRoad(0) = DualListRecordset("road name") Then
        FromDir1 = DualListRecordset("direction")
    End If
    If TransCharService = DualListRecordset("road name") Then
        ToDir1 = DualListRecordset("direction")
    End If
DualListRecordset.MoveNext
Loop
Set ConnBus = New ADODB.Connection
If FromCharService <> "" Then
    ConnBus.Open FromCharService & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & FromCharService & "dir2 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
    If FromRoad(0) = DualListRecordset("road name") Then
        FromDir2 = DualListRecordset("direction")
    End If
    If TransCharService = DualListRecordset("road name") Then
        ToDir2 = DualListRecordset("direction")
    End If
DualListRecordset.MoveNext
Loop
If ToDir1 > FromDir1 Then
    Set ConnBus = New ADODB.Connection
    If FromCharService <> "" Then
        ConnBus.Open FromCharService & "dir1"
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & FromCharService & "dir1 ORDER BY [direction]", ConnBus
    End If
    Do Until DualListRecordset.EOF
        For j = 0 To 10
            If DualListRecordset("road name") = FromRoad(j) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = DualListRecordset("mrt name")
                FromMrtRoad = DualListRecordset("road name")
            End If
        End If
        DualListRecordset.MoveNext
    Loop
End If
If ToDir2 > FromDir2 Then
    Set ConnBus = New ADODB.Connection
    If FromCharService <> "" Then
        ConnBus.Open FromCharService & "dir2"
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & FromCharService & "dir2 ORDER BY [direction]", ConnBus
    End If
    Do Until DualListRecordset.EOF
        For j = 0 To 10
            If DualListRecordset("road name") = FromRoad(j) Then
                FromMrtSpot = 1
                Exit For
            End If
        Next
        If FromMrtSpot = 1 Then
            If DualListRecordset("mrt id") <> "no" And FromMrt = "" Then
                FromMrt = DualListRecordset("mrt name")
                FromMrtRoad = DualListRecordset("road name")
            End If
        End If
    End If
End If

```



```

        DualListRecordset.MoveNext
    Loop
End If
End If

'check toservice route for mrt
Set BusService = New ADODB.Connection
BusService.Open "busdir"
Set ServiceRecordset = New ADODB.Recordset
ServiceRecordset.Open "SELECT * FROM busdir", BusService
Do Until ServiceRecordset.EOF
    If ToCharService = ServiceRecordset("service no") Then
        CheckDual = ServiceRecordset("dual direction")
        Exit Do
    End If
ServiceRecordset.MoveNext
Loop
If CheckDual = 0 Then
    Set ConnBus = New ADODB.Connection
    If ToCharService <> "" Then
        ConnBus.Open ToCharService
        Set LoopListRecordset = New ADODB.Recordset
        LoopListRecordset.Open "SELECT * FROM " & ToCharService & " ORDER BY [direction]", ConnBus
    End If
    Do Until LoopListRecordset.EOF
        For j = 0 To 10
            If LoopListRecordset("road name") = ToRoad(j) Then
                ToMrtSpot = 1
                Exit For
            End If
        Next
        If ToMrtSpot = 0 Then
            If LoopListRecordset("mrt id") <> "no" Then
                ToMrt = LoopListRecordset("mrt name")
                ToMrtRoad = LoopListRecordset("road name")
            End If
        End If
        If ToMrtSpot = 1 Then
            Exit Do
        End If
    LoopListRecordset.MoveNext
Loop
ElseIf CheckDual = 1 Then
    Set ConnBus = New ADODB.Connection
    If ToCharService <> "" Then
        ConnBus.Open ToCharService & "dir1"
        Set DualListRecordset = New ADODB.Recordset
        DualListRecordset.Open "SELECT * FROM " & ToCharService & "dir1 ORDER BY [direction]", ConnBus
    End If
    Do Until DualListRecordset.EOF
        If TransCharService = DualListRecordset("road name") Then
            FromDir1 = DualListRecordset("direction")
        End If
        If ToRoad(0) = DualListRecordset("road name") Then
            ToDir1 = DualListRecordset("direction")
        End If
    DualListRecordset.MoveNext
Loop
Set ConnBus = New ADODB.Connection
If ToCharService <> "" Then
    ConnBus.Open ToCharService & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ToCharService & "dir2 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
    If TransCharService = DualListRecordset("road name") Then
        FromDir2 = DualListRecordset("direction")
    End If
    If ToRoad(0) = DualListRecordset("road name") Then
        ToDir2 = DualListRecordset("direction")
    End If
DualListRecordset.MoveNext
Loop
If ToDir1 > FromDir1 Then
    Set ConnBus = New ADODB.Connection
    If ToCharService <> "" Then

```

```

ConnBus.Open ToCharService & "dir1"
Set DualListRecordset = New ADODB.Recordset
DualListRecordset.Open "SELECT * FROM " & ToCharService & "dir1 ORDER BY [direction]", ConnBus
End If
Do Until DualListRecordset.EOF
  For j = 0 To 10
    If DualListRecordset("road name") = ToRoad(j) Then
      ToMrtSpot = 1
      Exit For
    End If
  Next
  If ToMrtSpot = 0 Then
    If DualListRecordset("mrt id") <> "no" Then
      ToMrt = DualListRecordset("mrt name")
      ToMrtRoad = DualListRecordset("road name")
    End If
  End If
  If ToMrtSpot = 1 Then
    Exit Do
  End If
  DualListRecordset.MoveNext
Loop
End If
If ToDir2 > FromDir2 Then
  Set ConnBus = New ADODB.Connection
  If ToCharService <> "" Then
    ConnBus.Open ToCharService & "dir2"
    Set DualListRecordset = New ADODB.Recordset
    DualListRecordset.Open "SELECT * FROM " & ToCharService & "dir2 ORDER BY [direction]", ConnBus
  End If
  Do Until DualListRecordset.EOF
    For j = 0 To 10
      If DualListRecordset("road name") = ToRoad(j) Then
        ToMrtSpot = 1
        Exit For
      End If
    Next
    If ToMrtSpot = 0 Then
      If DualListRecordset("mrt id") <> "no" Then
        ToMrt = DualListRecordset("mrt name")
        ToMrtRoad = DualListRecordset("road name")
      End If
    End If
    If ToMrtSpot = 1 Then
      Exit Do
    End If
    DualListRecordset.MoveNext
  Loop
End If
End If

'combine results

If FromMrt = "" Or ToMrt = "" Or FromMrt = ToMrt Then
  Exit Sub
Else
  Call MrtTransfer
  SMSText.Text = SMSText.Text & ", Or " & "at " & FromMrtRoad & ", " & MrtText & ", at " & ToMrtRoad & ", take " &
ToCharService
End If
End Sub
VERSION 5.00
Begin VB.Form parking
  Caption = "car parking system"
  ClientHeight = 5850
  ClientLeft = 60
  ClientTop = 345
  ClientWidth = 5895
  LinkTopic = "Form1"
  ScaleHeight = 5850
  ScaleWidth = 5895
  StartUpPosition = 3 'Windows Default
  Begin VB.Data carparkinfo
    Caption = "carparkInfo"
    Connect = "Access 2000;"
    DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
  End

```

```

DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparks"
Top = 3360
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkC
Caption = "carparkC"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkC"
Top = 2880
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkB
Caption = "carparkB"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkB"
Top = 2400
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkA
Caption = "carparkA"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkA"
Top = 1920
Visible = 0 'False
Width = 1695
End
Begin VB.TextBox Text4
Height = 375
Left = 1080
TabIndex = 20
Top = 5160
Width = 2775
End
Begin VB.Timer Timer1
Interval = 1000
Left = 5160
Top = 720
End
End

```

```

Begin VB.TextBox Text1
  DataSource = "Data1"
  Height = 375
  Left = 1080
  TabIndex = 8
  Top = 3600
  Width = 2775
End
Begin VB.TextBox Hits
  Height = 375
  Left = 3840
  TabIndex = 7
  Top = 720
  Width = 735
End
Begin VB.TextBox SMSText
  Height = 1455
  Left = 360
  MaxLength = 160
  MultiLine = -1 'True
  TabIndex = 6
  ToolTipText = "Write the message here (max 160 characters)"
  Top = 1920
  Width = 3495
End
Begin VB.TextBox OtherEndAddress
  DataSource = "Data1"
  Height = 285
  Left = 360
  TabIndex = 5
  Top = 720
  Width = 2535
End
Begin VB.TextBox SMSCAddress
  Height = 285
  Left = 360
  TabIndex = 4
  ToolTipText = "Short message centre address"
  Top = 1320
  Width = 2535
End
Begin VB.CommandButton Close
  Caption = "Close"
  Height = 375
  Left = 4560
  TabIndex = 3
  Top = 5160
  Width = 1095
End
Begin VB.CommandButton Command1
  Caption = "Send"
  Height = 375
  Left = 4560
  TabIndex = 2
  Top = 4560
  Width = 1095
End
Begin VB.TextBox Text2
  DataSource = "Data1"
  Height = 375
  Left = 1080
  TabIndex = 1
  Top = 4080
  Width = 2775
End
Begin VB.TextBox Text3
  DataSource = "Data1"
  Height = 375
  Left = 1080
  TabIndex = 0
  Top = 4560
  Width = 2775
End
Begin VB.Label Label8
  Caption = "Mobile No:."
  Height = 255

```

```

Left      = 120
TabIndex = 19
Top       = 4680
Width     = 855
End
Begin VB.Label Label7
Caption   = "Car Park:"
Height   = 255
Left     = 120
TabIndex = 18
Top      = 4200
Width    = 735
End
Begin VB.Label Label6
Caption   = "Plate No:"
Height   = 255
Left     = 120
TabIndex = 17
Top      = 3720
Width    = 855
End
Begin VB.Label Label5
Caption   = "Info: Carpark"
Height   = 255
Left     = 120
TabIndex = 16
Top      = 5280
Width    = 975
End
Begin VB.Label Label11
Caption   = "Message"
Height   = 255
Left     = 360
TabIndex = 15
Top      = 1680
Width    = 735
End
Begin VB.Label FromTo
Caption   = "From"
Height   = 255
Left     = 360
TabIndex = 14
Top      = 480
Width    = 1935
End
Begin VB.Label Label2
Caption   = "SMSC Address"
Height   = 255
Left     = 360
TabIndex = 13
Top      = 1080
Width    = 1575
End
Begin VB.Label Label3
Caption   = "Message status:"
Height   = 255
Left     = 360
TabIndex = 12
Top      = 120
Width    = 1215
End
Begin VB.Label MessageStatus
Height   = 255
Left     = 1680
TabIndex = 11
Top      = 120
Width    = 2295
End
Begin VB.Label Length
Caption   = "0"
Height   = 255
Left     = 3120
TabIndex = 10
Top      = 1680
Width    = 375
End
End

```

```

Begin VB.Label Label4
    Caption      = "/ 160"
    Height       = 255
    Left         = 3480
    TabIndex     = 9
    Top          = 1680
    Width        = 495
End
End
Attribute VB_Name = "parking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim PlateNo, MobileNo, CarparkId
Dim HourCal, HourPay
Dim MinCal, MinPay
Dim TotalParkFee, StartStop
Dim CapCount, CarAmount, AmountCap
Dim CouponAmount

Private Sub Close_Click()
    Unload parking
End Sub

Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text = "" Then
        Call ParkCoupon
    ElseIf Text1.Text = "" And Text2.Text = "" And Text3.Text = "" And Text4.Text <> "" Then
        Call ParkInfo
    Else
        SMSText.Text = "No Results"
    End If
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

Private Sub ParkInfo()
    CarparkId = Null
    CarAmount = 0
    AmountCap = 0
    CarparkId = Text4.Text
    If CarparkId = "a" Or CarparkId = "A" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "a" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "b" Or CarparkId = "B" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "b" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "c" Or CarparkId = "C" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "c" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    End If

    SMSText.Text = "Current capacity for car park:" & CarparkId & " is " & CarAmount & " out of " & AmountCap & "."

```

```

Text4.Text = ""
End Sub

Private Sub ParkCoupon()
PlateNo = Null
CarparkId = Null
MobileNo = Null
StartStop = 0
CapCount = 0
CouponAmount = 0
PlateNo = Text1.Text
CarparkId = Text2.Text
MobileNo = Text3.Text

If CarparkId = "a" Or CarparkId = "A" Then
carpkA.Refresh
Do Until carpkA.Recordset.EOF
If carpkA.Recordset("plate_no") = PlateNo Then
If carpkA.Recordset("time_end") = 0 Then
StartStop = 1
StartTime = carpkA.Recordset("time_start")
Call cmdDifference
With carpkA.Recordset
.Edit
!time_end = Format(Now, "Short Time")
!fee_amount = TotalParkFee
.Update
End With
End If
End If
carpkA.Recordset.MoveNext
Loop
If StartStop = 0 Then
With carpkA.Recordset
.AddNew
!mobile_no = MobileNo
!plate_no = PlateNo
!date_park = Date$
!time_start = Format(Now, "Short Time")
.Update
End With
StartTime = Format(Now, "Short Time")
End If
carpkA.Refresh
Do Until carpkA.Recordset.EOF
If carpkA.Recordset("time_end") = 0 Then
CapCount = CapCount + 1
End If
carpkA.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
If carpkinfo.Recordset("carparks") = "a" Then
With carpkinfo.Recordset
.Edit
!car_amount = CapCount
.Update
End With
End If
carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
carpkB.Refresh
Do Until carpkB.Recordset.EOF
If carpkB.Recordset("plate_no") = PlateNo Then
If carpkB.Recordset("time_end") = 0 Then
StartStop = 1
StartTime = carpkB.Recordset("time_start")
Call cmdDifference
With carpkB.Recordset
.Edit
!time_end = Format(Now, "Short Time")
!fee_amount = TotalParkFee
.Update
End With

```

```

        End If
    End If
    carpkB.Recordset.MoveNext
Loop
If StartStop = 0 Then
    With carpkB.Recordset
        .AddNew
        !mobile_no = MobileNo
        !plate_no = PlateNo
        !date_park = Date$
        !time_start = Format(Now, "Short Time")
        .Update
    End With
    StartTime = Format(Now, "Short Time")
End If
carpkB.Refresh
Do Until carpkB.Recordset.EOF
    If carpkB.Recordset("time_end") = 0 Then
        CapCount = CapCount + 1
    End If
    carpkB.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
    If carpkinfo.Recordset("carparks") = "b" Then
        With carpkinfo.Recordset
            .Edit
            !car_amount = CapCount
            .Update
        End With
    End If
    carpkinfo.Recordset.MoveNext
Loop

ElseIf CarparkId = "c" Or CarparkId = "C" Then
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("plate_no") = PlateNo Then
            If carpkC.Recordset("time_end") = 0 Then
                StartStop = 1
                StartTime = carpkC.Recordset("time_start")
                Call cmdDifference
                With carpkC.Recordset
                    .Edit
                    !time_end = Format(Now, "Short Time")
                    !fee_amount = TotalParkFee
                    .Update
                End With
            End If
        End If
        carpkC.Recordset.MoveNext
    Loop
    If StartStop = 0 Then
        With carpkC.Recordset
            .AddNew
            !mobile_no = MobileNo
            !plate_no = PlateNo
            !date_park = Date$
            !time_start = Format(Now, "Short Time")
            .Update
        End With
        StartTime = Format(Now, "Short Time")
    End If
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("time_end") = 0 Then
            CapCount = CapCount + 1
        End If
        carpkC.Recordset.MoveNext
    Loop
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "c" Then
            With carpkinfo.Recordset
                .Edit
                !car_amount = CapCount
            End With
        End If
    End Do
End If

```



```

        .Update
    End With
End If
carpkinfo.Recordset.MoveNext
Loop
End If

If CouponAmount <> 0 Then
    SMSText.Text = CouponAmount & " will be billed into your account."
ElseIf CouponAmount = 0 Then
    SMSText.Text = "Thanks for parking in car park:" & Text2.Text & "."
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub
Private Sub cmdDifference()

HourCal = GetHours(StartTime, Format(Now, "Short Time"))
MinCal = GetMins(StartTime, Format(Now, "Short Time"))

    Call CalDiff

End Sub
Private Sub CalDiff()

    If HourCal <> "" Then
        HourPay = HourCal * 100
    End If
    If MinCal <> "" Then
        MinPay = MinCal * 2
    End If

    TotalParkFee = HourPay + MinPay
    If TotalParkFee >= 100 Then
        TotalParkFee = TotalParkFee / 100
        CouponAmount = "$" & TotalParkFee
    Else
        CouponAmount = TotalParkFee & " cents"
    End If

End Sub
VERSION 5.00
Begin VB.Form parking
    Caption       = "car parking system"
    ClientHeight  = 5850
    ClientLeft   = 60
    ClientTop    = 345
    ClientWidth  = 5895
    LinkTopic    = "Form1"
    ScaleHeight  = 5850
    ScaleWidth   = 5895
    StartUpPosition = 3 'Windows Default
    Begin VB.Data carpkinfo
        Caption       = "carparkInfo"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left         = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparks"
        Top          = 3360
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carpkC
        Caption       = "carparkC"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
    End

```

```

DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkC"
Top = 2880
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkB
Caption = "carparkB"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkB"
Top = 2400
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkA
Caption = "carparkA"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkA"
Top = 1920
Visible = 0 'False
Width = 1695
End
Begin VB.TextBox Text4
Height = 375
Left = 1080
TabIndex = 20
Top = 5160
Width = 2775
End
Begin VB.Timer Timer1
Interval = 1000
Left = 5160
Top = 720
End
Begin VB.TextBox Text1
DataSource = "Data1"
Height = 375
Left = 1080
TabIndex = 8
Top = 3600
Width = 2775
End
Begin VB.TextBox Hits
Height = 375
Left = 3840
TabIndex = 7
Top = 720
Width = 735
End
Begin VB.TextBox SMSText
Height = 1455

```

```

Left      = 360
MaxLength = 160
MultiLine = -1 'True
TabIndex  = 6
ToolTipText = "Write the message here (max 160 characters)"
Top       = 1920
Width     = 3495
End
Begin VB.TextBox OtherEndAddress
    DataSource = "Data1"
    Height     = 285
    Left      = 360
    TabIndex  = 5
    Top       = 720
    Width     = 2535
End
Begin VB.TextBox SMSCAddress
    Height     = 285
    Left      = 360
    TabIndex  = 4
    ToolTipText = "Short message centre address"
    Top       = 1320
    Width     = 2535
End
Begin VB.CommandButton Close
    Caption    = "Close"
    Height     = 375
    Left      = 4560
    TabIndex  = 3
    Top       = 5160
    Width     = 1095
End
Begin VB.CommandButton Command1
    Caption    = "Send"
    Height     = 375
    Left      = 4560
    TabIndex  = 2
    Top       = 4560
    Width     = 1095
End
Begin VB.TextBox Text2
    DataSource = "Data1"
    Height     = 375
    Left      = 1080
    TabIndex  = 1
    Top       = 4080
    Width     = 2775
End
Begin VB.TextBox Text3
    DataSource = "Data1"
    Height     = 375
    Left      = 1080
    TabIndex  = 0
    Top       = 4560
    Width     = 2775
End
Begin VB.Label Label8
    Caption    = "Mobile No:"
    Height     = 255
    Left      = 120
    TabIndex  = 19
    Top       = 4680
    Width     = 855
End
Begin VB.Label Label7
    Caption    = "Car Park:"
    Height     = 255
    Left      = 120
    TabIndex  = 18
    Top       = 4200
    Width     = 735
End
Begin VB.Label Label6
    Caption    = "Plate No:"
    Height     = 255
    Left      = 120

```

```

    TabIndex    = 17
    Top        = 3720
    Width      = 855
End
Begin VB.Label Label5
    Caption     = "Info: Carpark"
    Height     = 255
    Left       = 120
    TabIndex   = 16
    Top        = 5280
    Width      = 975
End
Begin VB.Label Label1
    Caption     = "Message"
    Height     = 255
    Left       = 360
    TabIndex   = 15
    Top        = 1680
    Width      = 735
End
Begin VB.Label FromTo
    Caption     = "From"
    Height     = 255
    Left       = 360
    TabIndex   = 14
    Top        = 480
    Width      = 1935
End
Begin VB.Label Label2
    Caption     = "SMSC Address"
    Height     = 255
    Left       = 360
    TabIndex   = 13
    Top        = 1080
    Width      = 1575
End
Begin VB.Label Label3
    Caption     = "Message status:"
    Height     = 255
    Left       = 360
    TabIndex   = 12
    Top        = 120
    Width      = 1215
End
Begin VB.Label MessageStatus
    Height     = 255
    Left       = 1680
    TabIndex   = 11
    Top        = 120
    Width      = 2295
End
Begin VB.Label Length
    Caption     = "0"
    Height     = 255
    Left       = 3120
    TabIndex   = 10
    Top        = 1680
    Width      = 375
End
Begin VB.Label Label4
    Caption     = "/ 160"
    Height     = 255
    Left       = 3480
    TabIndex   = 9
    Top        = 1680
    Width      = 495
End
End
Attribute VB_Name = "parking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim PlateNo, MobileNo, CarparkId
Dim HourCal, HourPay

```

```

Dim MinCal, MinPay
Dim TotalParkFee, StartStop
Dim CapCount, CarAmount, AmountCap
Dim CouponAmount

Private Sub Close_Click()
    Unload parking
End Sub

Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text = "" Then
        Call ParkCoupon
    ElseIf Text1.Text = "" And Text2.Text = "" And Text3.Text = "" And Text4.Text <> "" Then
        Call ParkInfo
    Else
        SMSText.Text = "No Results"
    End If
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

Private Sub ParkInfo()
    CarparkId = Null
    CarAmount = 0
    AmountCap = 0
    CarparkId = Text4.Text
    If CarparkId = "a" Or CarparkId = "A" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "a" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "b" Or CarparkId = "B" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "b" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "c" Or CarparkId = "C" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "c" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    End If

    SMSText.Text = "Current capacity for car park:" & CarparkId & " is " & CarAmount & " out of " & AmountCap & "."

    Text4.Text = ""
End Sub

Private Sub ParkCoupon()
    PlateNo = Null
    CarparkId = Null
    MobileNo = Null
    StartStop = 0
    CapCount = 0
    CouponAmount = 0
    PlateNo = Text1.Text
    CarparkId = Text2.Text
    MobileNo = Text3.Text

    If CarparkId = "a" Or CarparkId = "A" Then
        carpkA.Refresh
    End If
End Sub

```

```

Do Until carpkA.Recordset.EOF
  If carpkA.Recordset("plate_no") = PlateNo Then
    If carpkA.Recordset("time_end") = 0 Then
      StartStop = 1
      StartTime = carpkA.Recordset("time_start")
      Call cmdDifference
      With carpkA.Recordset
        .Edit
        !time_end = Format(Now, "Short Time")
        !fee_amount = TotalParkFee
      .Update
      End With
    End If
  End If
  carpkA.Recordset.MoveNext
Loop
If StartStop = 0 Then
  With carpkA.Recordset
    .AddNew
    !mobile_no = MobileNo
    !plate_no = PlateNo
    !date_park = Date$
    !time_start = Format(Now, "Short Time")
  .Update
  End With
  StartTime = Format(Now, "Short Time")
End If
carpkA.Refresh
Do Until carpkA.Recordset.EOF
  If carpkA.Recordset("time_end") = 0 Then
    CapCount = CapCount + 1
  End If
  carpkA.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
  If carpkinfo.Recordset("carparks") = "a" Then
    With carpkinfo.Recordset
      .Edit
      !car_amount = CapCount
    .Update
    End With
  End If
  carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
  carpkB.Refresh
  Do Until carpkB.Recordset.EOF
    If carpkB.Recordset("plate_no") = PlateNo Then
      If carpkB.Recordset("time_end") = 0 Then
        StartStop = 1
        StartTime = carpkB.Recordset("time_start")
        Call cmdDifference
        With carpkB.Recordset
          .Edit
          !time_end = Format(Now, "Short Time")
          !fee_amount = TotalParkFee
        .Update
        End With
      End If
    End If
    carpkB.Recordset.MoveNext
  Loop
  If StartStop = 0 Then
    With carpkB.Recordset
      .AddNew
      !mobile_no = MobileNo
      !plate_no = PlateNo
      !date_park = Date$
      !time_start = Format(Now, "Short Time")
    .Update
    End With
    StartTime = Format(Now, "Short Time")
  End If
  carpkB.Refresh
  Do Until carpkB.Recordset.EOF

```

```

    If carpkB.Recordset("time_end") = 0 Then
        CapCount = CapCount + 1
    End If
    carpkB.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
    If carpkinfo.Recordset("carparks") = "b" Then
        With carpkinfo.Recordset
            .Edit
            !car_amount = CapCount
            .Update
        End With
    End If
    carpkinfo.Recordset.MoveNext
Loop

ElseIf CarparkId = "c" Or CarparkId = "C" Then
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("plate_no") = PlateNo Then
            If carpkC.Recordset("time_end") = 0 Then
                StartStop = 1
                StartTime = carpkC.Recordset("time_start")
                Call cmdDifference
                With carpkC.Recordset
                    .Edit
                    !time_end = Format(Now, "Short Time")
                    !fee_amount = TotalParkFee
                    .Update
                End With
            End If
        End If
        carpkC.Recordset.MoveNext
    Loop
    If StartStop = 0 Then
        With carpkC.Recordset
            .AddNew
            !mobile_no = MobileNo
            !plate_no = PlateNo
            !date_park = Date$
            !time_start = Format(Now, "Short Time")
            .Update
        End With
        StartTime = Format(Now, "Short Time")
    End If
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("time_end") = 0 Then
            CapCount = CapCount + 1
        End If
        carpkC.Recordset.MoveNext
    Loop
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "c" Then
            With carpkinfo.Recordset
                .Edit
                !car_amount = CapCount
                .Update
            End With
        End If
        carpkinfo.Recordset.MoveNext
    Loop
End If

If CouponAmount <> 0 Then
    SMSText.Text = CouponAmount & " will be billed into your account."
ElseIf CouponAmount = 0 Then
    SMSText.Text = "Thanks for parking in car park:" & Text2.Text & "."
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub

```

```

Private Sub cmdDifference()

HourCal = GetHours(StartTime, Format(Now, "Short Time"))
MinCal = GetMins(StartTime, Format(Now, "Short Time"))

    Call CalDiff

End Sub
Private Sub CalDiff()

    If HourCal <> "" Then
        HourPay = HourCal * 100
    End If
    If MinCal <> "" Then
        MinPay = MinCal * 2
    End If

    TotalParkFee = HourPay + MinPay
    If TotalParkFee >= 100 Then
        TotalParkFee = TotalParkFee / 100
        CouponAmount = "$" & TotalParkFee
    Else
        CouponAmount = TotalParkFee & " cents"
    End If

End Sub
VERSION 5.00
Begin VB.Form parking
    Caption       = "car parking system"
    ClientHeight  = 5850
    ClientLeft    = 60
    ClientTop     = 345
    ClientWidth   = 5895
    LinkTopic     = "Form1"
    ScaleHeight   = 5850
    ScaleWidth    = 5895
    StartUpPosition = 3 'Windows Default
    Begin VB.Data carparkinfo
        Caption       = "carparkInfo"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left          = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparks"
        Top           = 3360
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carparkC
        Caption       = "carparkC"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left          = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparkC"
        Top           = 2880
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carparkB
        Caption       = "carparkB"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"

```



```

DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkB"
Top = 2400
Visible = 0 'False
Width = 1695
End
Begin VB.Data carpkA
Caption = "carparkA"
Connect = "Access 2000;"
DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkA"
Top = 1920
Visible = 0 'False
Width = 1695
End
Begin VB.TextBox Text4
Height = 375
Left = 1080
TabIndex = 20
Top = 5160
Width = 2775
End
Begin VB.Timer Timer1
Interval = 1000
Left = 5160
Top = 720
End
Begin VB.TextBox Text1
DataSource = "Data1"
Height = 375
Left = 1080
TabIndex = 8
Top = 3600
Width = 2775
End
Begin VB.TextBox Hits
Height = 375
Left = 3840
TabIndex = 7
Top = 720
Width = 735
End
Begin VB.TextBox SMSText
Height = 1455
Left = 360
MaxLength = 160
MultiLine = -1 'True
TabIndex = 6
ToolTipText = "Write the message here (max 160 characters)"
Top = 1920
Width = 3495
End
Begin VB.TextBox OtherEndAddress
DataSource = "Data1"
Height = 285
Left = 360
TabIndex = 5
Top = 720
Width = 2535
End
Begin VB.TextBox SMSCAddress

```

```

Height      = 285
Left        = 360
TabIndex    = 4
ToolTipText = "Short message centre address"
Top         = 1320
Width       = 2535
End
Begin VB.CommandButton Close
Caption     = "Close"
Height     = 375
Left       = 4560
TabIndex   = 3
Top        = 5160
Width      = 1095
End
Begin VB.CommandButton Command1
Caption     = "Send"
Height     = 375
Left       = 4560
TabIndex   = 2
Top        = 4560
Width      = 1095
End
Begin VB.TextBox Text2
DataSource = "Data1"
Height     = 375
Left       = 1080
TabIndex   = 1
Top        = 4080
Width      = 2775
End
Begin VB.TextBox Text3
DataSource = "Data1"
Height     = 375
Left       = 1080
TabIndex   = 0
Top        = 4560
Width      = 2775
End
Begin VB.Label Label8
Caption    = "Mobile No:"
Height    = 255
Left      = 120
TabIndex  = 19
Top       = 4680
Width     = 855
End
Begin VB.Label Label7
Caption    = "Car Park:"
Height    = 255
Left      = 120
TabIndex  = 18
Top       = 4200
Width     = 735
End
Begin VB.Label Label6
Caption    = "Plate No:"
Height    = 255
Left      = 120
TabIndex  = 17
Top       = 3720
Width     = 855
End
Begin VB.Label Label5
Caption    = "Info: Carpark"
Height    = 255
Left      = 120
TabIndex  = 16
Top       = 5280
Width     = 975
End
Begin VB.Label Label1
Caption    = "Message"
Height    = 255
Left      = 360
TabIndex  = 15

```

```

    Top      = 1680
    Width    = 735
End
Begin VB.Label FromTo
    Caption  = "From"
    Height   = 255
    Left     = 360
    TabIndex = 14
    Top      = 480
    Width    = 1935
End
Begin VB.Label Label2
    Caption  = "SMSC Address"
    Height   = 255
    Left     = 360
    TabIndex = 13
    Top      = 1080
    Width    = 1575
End
Begin VB.Label Label3
    Caption  = "Message status:"
    Height   = 255
    Left     = 360
    TabIndex = 12
    Top      = 120
    Width    = 1215
End
Begin VB.Label MessageStatus
    Height   = 255
    Left     = 1680
    TabIndex = 11
    Top      = 120
    Width    = 2295
End
Begin VB.Label Length
    Caption  = "0"
    Height   = 255
    Left     = 3120
    TabIndex = 10
    Top      = 1680
    Width    = 375
End
Begin VB.Label Label4
    Caption  = "/ 160"
    Height   = 255
    Left     = 3480
    TabIndex = 9
    Top      = 1680
    Width    = 495
End
End
Attribute VB_Name = "parking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim PlateNo, MobileNo, CarparkId
Dim HourCal, HourPay
Dim MinCal, MinPay
Dim TotalParkFee, StartStop
Dim CapCount, CarAmount, AmountCap
Dim CouponAmount

Private Sub Close_Click()
    Unload parking
End Sub

Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text = "" Then
        Call ParkCoupon
    ElseIf Text1.Text = "" And Text2.Text = "" And Text3.Text = "" And Text4.Text <> "" Then
        Call ParkInfo
    Else
        SMSText.Text = "No Results"
    End If
End Sub

```

```

    End If
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

Private Sub ParkInfo()
    CarparkId = Null
    CarAmount = 0
    AmountCap = 0
    CarparkId = Text4.Text
    If CarparkId = "a" Or CarparkId = "A" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "a" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "b" Or CarparkId = "B" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "b" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "c" Or CarparkId = "C" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "c" Then
                CarAmount = carpkinfo.Recordset("car_amount")
                AmountCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    End If

    SMSText.Text = "Current capacity for car park:" & CarparkId & " is " & CarAmount & " out of " & AmountCap & "."

    Text4.Text = ""
End Sub

Private Sub ParkCoupon()
    PlateNo = Null
    CarparkId = Null
    MobileNo = Null
    StartStop = 0
    CapCount = 0
    CouponAmount = 0
    PlateNo = Text1.Text
    CarparkId = Text2.Text
    MobileNo = Text3.Text

    If CarparkId = "a" Or CarparkId = "A" Then
        carpkA.Refresh
        Do Until carpkA.Recordset.EOF
            If carpkA.Recordset("plate_no") = PlateNo Then
                If carpkA.Recordset("time_end") = 0 Then
                    StartStop = 1
                    StartTime = carpkA.Recordset("time_start")
                    Call cmdDifference
                    With carpkA.Recordset
                        .Edit
                        !time_end = Format(Now, "Short Time")
                        !fee_amount = TotalParkFee
                    .Update
                End With
            End If
        End If
        carpkA.Recordset.MoveNext
    Loop
    If StartStop = 0 Then

```

```

With carpkA.Recordset
    .AddNew
    !mobile_no = MobileNo
    !plate_no = PlateNo
    !date_park = Date$
    !time_start = Format(Now, "Short Time")
    .Update
End With
StartTime = Format(Now, "Short Time")
End If
carpkA.Refresh
Do Until carpkA.Recordset.EOF
    If carpkA.Recordset("time_end") = 0 Then
        CapCount = CapCount + 1
    End If
    carpkA.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
    If carpkinfo.Recordset("carparks") = "a" Then
        With carpkinfo.Recordset
            .Edit
            !car_amount = CapCount
            .Update
        End With
    End If
    carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
    carpkB.Refresh
    Do Until carpkB.Recordset.EOF
        If carpkB.Recordset("plate_no") = PlateNo Then
            If carpkB.Recordset("time_end") = 0 Then
                StartStop = 1
                StartTime = carpkB.Recordset("time_start")
                Call cmdDifference
                With carpkB.Recordset
                    .Edit
                    !time_end = Format(Now, "Short Time")
                    !fee_amount = TotalParkFee
                    .Update
                End With
            End If
        End If
        carpkB.Recordset.MoveNext
    Loop
    If StartStop = 0 Then
        With carpkB.Recordset
            .AddNew
            !mobile_no = MobileNo
            !plate_no = PlateNo
            !date_park = Date$
            !time_start = Format(Now, "Short Time")
            .Update
        End With
        StartTime = Format(Now, "Short Time")
    End If
    carpkB.Refresh
    Do Until carpkB.Recordset.EOF
        If carpkB.Recordset("time_end") = 0 Then
            CapCount = CapCount + 1
        End If
        carpkB.Recordset.MoveNext
    Loop
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "b" Then
            With carpkinfo.Recordset
                .Edit
                !car_amount = CapCount
                .Update
            End With
        End If
        carpkinfo.Recordset.MoveNext
    Loop

```

```

ElseIf CarparkId = "c" Or CarparkId = "C" Then
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("plate_no") = PlateNo Then
            If carpkC.Recordset("time_end") = 0 Then
                StartStop = 1
                StartTime = carpkC.Recordset("time_start")
                Call cmdDifference
                With carpkC.Recordset
                    .Edit
                    !time_end = Format(Now, "Short Time")
                    !fee_amount = TotalParkFee
                    .Update
                End With
            End If
        End If
        carpkC.Recordset.MoveNext
    Loop
    If StartStop = 0 Then
        With carpkC.Recordset
            .AddNew
            !mobile_no = MobileNo
            !plate_no = PlateNo
            !date_park = Date$
            !time_start = Format(Now, "Short Time")
            .Update
        End With
        StartTime = Format(Now, "Short Time")
    End If
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("time_end") = 0 Then
            CapCount = CapCount + 1
        End If
        carpkC.Recordset.MoveNext
    Loop
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "c" Then
            With carpkinfo.Recordset
                .Edit
                !car_amount = CapCount
                .Update
            End With
        End If
        carpkinfo.Recordset.MoveNext
    Loop
End If

If CouponAmount <> 0 Then
    SMSText.Text = CouponAmount & " will be billed into your account."
ElseIf CouponAmount = 0 Then
    SMSText.Text = "Thanks for parking in car park:" & Text2.Text & "."
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub
Private Sub cmdDifference()

HourCal = GetHours(StartTime, Format(Now, "Short Time"))
MinCal = GetMins(StartTime, Format(Now, "Short Time"))

    Call CalDiff

End Sub
Private Sub CalDiff()

    If HourCal <> "" Then
        HourPay = HourCal * 100
    End If
    If MinCal <> "" Then
        MinPay = MinCal * 2
    End If

```

```

TotalParkFee = HourPay + MinPay
If TotalParkFee >= 100 Then
    TotalParkFee = TotalParkFee / 100
    CouponAmount = "$" & TotalParkFee
Else
    CouponAmount = TotalParkFee & " cents"
End If

End Sub
VERSION 5.00
Begin VB.Form parking
    Caption       = "car parking system"
    ClientHeight = 5850
    ClientLeft   = 60
    ClientTop    = 345
    ClientWidth  = 5895
    LinkTopic    = "Form1"
    ScaleHeight  = 5850
    ScaleWidth   = 5895
    StartUpPosition = 3 'Windows Default
    Begin VB.Data carparkinfo
        Caption       = "carparkInfo"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left          = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparks"
        Top           = 3360
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carparkC
        Caption       = "carparkC"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left          = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparkC"
        Top           = 2880
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carparkB
        Caption       = "carparkB"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
        DefaultCursorType= 0 'DefaultCursor
        DefaultType   = 2 'UseODBC
        Exclusive     = 0 'False
        Height        = 375
        Left          = 4080
        Options       = 0
        ReadOnly      = 0 'False
        RecordsetType  = 1 'Dynaset
        RecordSource  = "carparkB"
        Top           = 2400
        Visible       = 0 'False
        Width         = 1695
    End
    Begin VB.Data carparkA
        Caption       = "carparkA"
        Connect       = "Access 2000;"
        DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
    End

```

```

DefaultCursorType= 0 'DefaultCursor
DefaultType = 2 'UseODBC
Exclusive = 0 'False
Height = 375
Left = 4080
Options = 0
ReadOnly = 0 'False
RecordsetType = 1 'Dynaset
RecordSource = "carparkA"
Top = 1920
Visible = 0 'False
Width = 1695
End
Begin VB.TextBox Text4
Height = 375
Left = 1080
TabIndex = 20
Top = 5160
Width = 2775
End
Begin VB.Timer Timer1
Interval = 1000
Left = 5160
Top = 720
End
Begin VB.TextBox Text1
DataSource = "Data1"
Height = 375
Left = 1080
TabIndex = 8
Top = 3600
Width = 2775
End
Begin VB.TextBox Hits
Height = 375
Left = 3840
TabIndex = 7
Top = 720
Width = 735
End
Begin VB.TextBox SMSText
Height = 1455
Left = 360
MaxLength = 160
MultiLine = -1 'True
TabIndex = 6
ToolTipText = "Write the message here (max 160 characters)"
Top = 1920
Width = 3495
End
Begin VB.TextBox OtherEndAddress
DataSource = "Data1"
Height = 285
Left = 360
TabIndex = 5
Top = 720
Width = 2535
End
Begin VB.TextBox SMSCAddress
Height = 285
Left = 360
TabIndex = 4
ToolTipText = "Short message centre address"
Top = 1320
Width = 2535
End
Begin VB.CommandButton Close
Caption = "Close"
Height = 375
Left = 4560
TabIndex = 3
Top = 5160
Width = 1095
End
Begin VB.CommandButton Command1
Caption = "Send"

```



```

Height      = 375
Left        = 4560
TabIndex    = 2
Top         = 4560
Width       = 1095
End
Begin VB.TextBox Text2
  DataSource = "Data1"
  Height     = 375
  Left       = 1080
  TabIndex   = 1
  Top        = 4080
  Width      = 2775
End
Begin VB.TextBox Text3
  DataSource = "Data1"
  Height     = 375
  Left       = 1080
  TabIndex   = 0
  Top        = 4560
  Width      = 2775
End
Begin VB.Label Label8
  Caption    = "Mobile No:"
  Height     = 255
  Left       = 120
  TabIndex   = 19
  Top        = 4680
  Width      = 855
End
Begin VB.Label Label7
  Caption    = "Car Park:"
  Height     = 255
  Left       = 120
  TabIndex   = 18
  Top        = 4200
  Width      = 735
End
Begin VB.Label Label6
  Caption    = "Plate No:"
  Height     = 255
  Left       = 120
  TabIndex   = 17
  Top        = 3720
  Width      = 855
End
Begin VB.Label Label5
  Caption    = "Info: Carpark"
  Height     = 255
  Left       = 120
  TabIndex   = 16
  Top        = 5280
  Width      = 975
End
Begin VB.Label Label11
  Caption    = "Message"
  Height     = 255
  Left       = 360
  TabIndex   = 15
  Top        = 1680
  Width      = 735
End
Begin VB.Label FromTo
  Caption    = "From"
  Height     = 255
  Left       = 360
  TabIndex   = 14
  Top        = 480
  Width      = 1935
End
Begin VB.Label Label2
  Caption    = "SMSC Address"
  Height     = 255
  Left       = 360
  TabIndex   = 13
  Top        = 1080

```

```

    Width      = 1575
End
Begin VB.Label Label3
    Caption    = "Message status:"
    Height     = 255
    Left       = 360
    TabIndex   = 12
    Top        = 120
    Width      = 1215
End
Begin VB.Label MessageStatus
    Height     = 255
    Left       = 1680
    TabIndex   = 11
    Top        = 120
    Width      = 2295
End
Begin VB.Label Length
    Caption    = "0"
    Height     = 255
    Left       = 3120
    TabIndex   = 10
    Top        = 1680
    Width      = 375
End
Begin VB.Label Label4
    Caption    = "/ 160"
    Height     = 255
    Left       = 3480
    TabIndex   = 9
    Top        = 1680
    Width      = 495
End
End
Attribute VB_Name = "parking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim PlateNo, MobileNo, CarparkId
Dim HourCal, HourPay
Dim MinCal, MinPay
Dim TotalParkFee, StartStop
Dim CapCount, CarAmount, AmountCap
Dim CouponAmount

Private Sub Close_Click()
    Unload parking
End Sub

Private Sub Command1_Click()
    If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text = "" Then
        Call ParkCoupon
    ElseIf Text1.Text = "" And Text2.Text = "" And Text3.Text = "" And Text4.Text <> "" Then
        Call ParkInfo
    Else
        SMSText.Text = "No Results"
    End If
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

Private Sub ParkInfo()
    CarparkId = Null
    CarAmount = 0
    AmountCap = 0
    CarparkId = Text4.Text
    If CarparkId = "a" Or CarparkId = "A" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "a" Then
                CarAmount = carpkinfo.Recordset("car_amount")
            End If
        Loop
    End If
End Sub

```

```

        AmountCap = carpkinfo.Recordset("capacity")
    End If
    carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "b" Then
            CarAmount = carpkinfo.Recordset("car_amount")
            AmountCap = carpkinfo.Recordset("capacity")
        End If
        carpkinfo.Recordset.MoveNext
    Loop
ElseIf CarparkId = "c" Or CarparkId = "C" Then
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "c" Then
            CarAmount = carpkinfo.Recordset("car_amount")
            AmountCap = carpkinfo.Recordset("capacity")
        End If
        carpkinfo.Recordset.MoveNext
    Loop
End If

SMSText.Text = "Current capacity for car park:" & CarparkId & " is " & CarAmount & " out of " & AmountCap & "."

Text4.Text = ""
End Sub

Private Sub ParkCoupon()
    PlateNo = Null
    CarparkId = Null
    MobileNo = Null
    StartStop = 0
    CapCount = 0
    CouponAmount = 0
    PlateNo = Text1.Text
    CarparkId = Text2.Text
    MobileNo = Text3.Text

    If CarparkId = "a" Or CarparkId = "A" Then
        carpkA.Refresh
        Do Until carpkA.Recordset.EOF
            If carpkA.Recordset("plate_no") = PlateNo Then
                If carpkA.Recordset("time_end") = 0 Then
                    StartStop = 1
                    StartTime = carpkA.Recordset("time_start")
                    Call cmdDifference
                    With carpkA.Recordset
                        .Edit
                        !time_end = Format(Now, "Short Time")
                        !fee_amount = TotalParkFee
                        .Update
                    End With
                End If
            End If
            carpkA.Recordset.MoveNext
        Loop
        If StartStop = 0 Then
            With carpkA.Recordset
                .AddNew
                !mobile_no = MobileNo
                !plate_no = PlateNo
                !date_park = Date$
                !time_start = Format(Now, "Short Time")
                .Update
            End With
            StartTime = Format(Now, "Short Time")
        End If
        carpkA.Refresh
        Do Until carpkA.Recordset.EOF
            If carpkA.Recordset("time_end") = 0 Then
                CapCount = CapCount + 1
            End If
            carpkA.Recordset.MoveNext
        Loop
    End If

```

```

carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
  If carpkinfo.Recordset("carparks") = "a" Then
    With carpkinfo.Recordset
      .Edit
      !car_amount = CapCount
      .Update
    End With
  End If
  carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
  carpkB.Refresh
  Do Until carpkB.Recordset.EOF
    If carpkB.Recordset("plate_no") = PlateNo Then
      If carpkB.Recordset("time_end") = 0 Then
        StartStop = 1
        StartTime = carpkB.Recordset("time_start")
        Call cmdDifference
        With carpkB.Recordset
          .Edit
          !time_end = Format(Now, "Short Time")
          !fee_amount = TotalParkFee
          .Update
        End With
      End If
    End If
    carpkB.Recordset.MoveNext
  Loop
  If StartStop = 0 Then
    With carpkB.Recordset
      .AddNew
      !mobile_no = MobileNo
      !plate_no = PlateNo
      !date_park = Date$
      !time_start = Format(Now, "Short Time")
      .Update
    End With
    StartTime = Format(Now, "Short Time")
  End If
  carpkB.Refresh
  Do Until carpkB.Recordset.EOF
    If carpkB.Recordset("time_end") = 0 Then
      CapCount = CapCount + 1
    End If
  carpkB.Recordset.MoveNext
Loop
  carpkinfo.Refresh
  Do Until carpkinfo.Recordset.EOF
    If carpkinfo.Recordset("carparks") = "b" Then
      With carpkinfo.Recordset
        .Edit
        !car_amount = CapCount
        .Update
      End With
    End If
    carpkinfo.Recordset.MoveNext
  Loop

ElseIf CarparkId = "c" Or CarparkId = "C" Then
  carpkC.Refresh
  Do Until carpkC.Recordset.EOF
    If carpkC.Recordset("plate_no") = PlateNo Then
      If carpkC.Recordset("time_end") = 0 Then
        StartStop = 1
        StartTime = carpkC.Recordset("time_start")
        Call cmdDifference
        With carpkC.Recordset
          .Edit
          !time_end = Format(Now, "Short Time")
          !fee_amount = TotalParkFee
          .Update
        End With
      End If
    End If
    carpkC.Recordset.MoveNext
  Loop

```

```

Loop
If StartStop = 0 Then
  With carpkC.Recordset
    .AddNew
    !mobile_no = MobileNo
    !plate_no = PlateNo
    !date_park = Date$
    !time_start = Format(Now, "Short Time")
    .Update
  End With
  StartTime = Format(Now, "Short Time")
End If
carpkC.Refresh
Do Until carpkC.Recordset.EOF
  If carpkC.Recordset("time_end") = 0 Then
    CapCount = CapCount + 1
  End If
  carpkC.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
  If carpkinfo.Recordset("carparks") = "c" Then
    With carpkinfo.Recordset
      .Edit
      !car_amount = CapCount
      .Update
    End With
  End If
  carpkinfo.Recordset.MoveNext
Loop
End If

If CouponAmount <> 0 Then
  SMSText.Text = CouponAmount & " will be billed into your account."
ElseIf CouponAmount = 0 Then
  SMSText.Text = "Thanks for parking in car park:" & Text2.Text & "."
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub
Private Sub cmdDifference()

HourCal = GetHours(StartTime, Format(Now, "Short Time"))
MinCal = GetMins(StartTime, Format(Now, "Short Time"))

  Call CalDiff

End Sub
Private Sub CalDiff()

  If HourCal <> "" Then
    HourPay = HourCal * 100
  End If
  If MinCal <> "" Then
    MinPay = MinCal * 2
  End If

  TotalParkFee = HourPay + MinPay
  If TotalParkFee >= 100 Then
    TotalParkFee = TotalParkFee / 100
    CouponAmount = "$" & TotalParkFee
  Else
    CouponAmount = TotalParkFee & " cents"
  End If

End Sub
VERSION 5.00
Begin VB.Form MapGuide
  Caption       = "Map Guide"
  ClientHeight  = 5190
  ClientLeft   = 60
  ClientTop    = 345
  ClientWidth  = 4965

```

```

LinkTopic = "Form1"
ScaleHeight = 5190
ScaleWidth = 4965
StartPosition = 3 'Windows Default
Begin VB.Timer Timer1
    Interval = 1000
    Left = 4200
    Top = 1920
End
Begin VB.TextBox Text1
    Height = 375
    Left = 240
    TabIndex = 6
    Top = 3720
    Width = 2775
End
Begin VB.TextBox Hits
    Height = 375
    Left = 3720
    TabIndex = 5
    Top = 840
    Width = 735
End
Begin VB.TextBox SMSText
    Height = 1455
    Left = 240
    MaxLength = 160
    MultiLine = -1 'True
    TabIndex = 4
    ToolTipText = "Write the message here (max 160 characters)"
    Top = 2040
    Width = 3495
End
Begin VB.TextBox OtherEndAddress
    Height = 285
    Left = 240
    TabIndex = 3
    Top = 840
    Width = 2535
End
Begin VB.TextBox SMSCAddress
    Height = 285
    Left = 240
    TabIndex = 2
    ToolTipText = "Short message centre address"
    Top = 1440
    Width = 2535
End
Begin VB.CommandButton Close
    Caption = "Close"
    Height = 375
    Left = 3480
    TabIndex = 1
    Top = 4560
    Width = 975
End
Begin VB.CommandButton Command1
    Caption = "Command1"
    Height = 375
    Left = 3360
    TabIndex = 0
    Top = 3720
    Width = 1455
End
Begin VB.Label Label1
    Caption = "Message"
    Height = 255
    Left = 240
    TabIndex = 13
    Top = 1800
    Width = 735
End
Begin VB.Label FromTo
    Caption = "From"
    Height = 255
    Left = 240

```

```

    TabIndex    = 12
    Top        = 600
    Width     = 1935
End
Begin VB.Label Label2
    Caption    = "SMSC Address"
    Height    = 255
    Left      = 240
    TabIndex  = 11
    Top       = 1200
    Width     = 1575
End
Begin VB.Label Label3
    Caption    = "Message status:"
    Height    = 255
    Left      = 240
    TabIndex  = 10
    Top       = 240
    Width     = 1215
End
Begin VB.Label MessageStatus
    Height    = 255
    Left     = 1560
    TabIndex = 9
    Top      = 240
    Width    = 2295
End
Begin VB.Label Length
    Caption    = "0"
    Height    = 255
    Left     = 3000
    TabIndex = 8
    Top      = 1800
    Width    = 375
End
Begin VB.Label Label4
    Caption    = "/ 160"
    Height    = 255
    Left     = 3360
    TabIndex = 7
    Top      = 1800
    Width    = 495
End
End
Attribute VB_Name = "MapGuide"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim FromRoad, ToRoad
Dim FromRegion, ToRegion, FromRank, ToRank
Dim i, j, k, l, m
Dim SortWord, Ticker, Pos, Char, Word, Topstr
Dim FromRoadWord, ToRoadWord
Dim CheckFromRoad, CheckToRoad, CheckWordPass, CheckRoadLink
Dim FromMap(10), ToMap(10), FromLink(10), ToLink(10), RdTake(10)
Dim Link(10)
Dim MapCount, LinkCount

Dim dbRoad As Connection
Dim WithEvents RoadPrimaryRS As Recordset
Attribute RoadPrimaryRS.VB_VarHelpID = -1

Private Sub Close_Click()
    Unload MapGuide
End Sub

Private Sub Command1_Click()
    Call TextSort
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

```

```

Private Sub TextSort()

    CheckFromRoad = 0
    CheckToRoad = 0
    CheckWordPass = 0

    FromRoadWord = ""
    ToRoadWord = ""
    FromRoad = ""
    ToRoad = ""
    SortWord = Text1.Text
    For i = 0 To 10

        RdTake(i) = ""
    Next
    Ticker = 0
    Pos = 1
    SortWord = SortWord & " "

    For i = 1 To Len(SortWord)
    Char = Mid(SortWord, i, 1)
    'For 1st word
    If Char = " " And Ticker = 0 Then
        Word = Mid(SortWord, Pos, i - 1)
        Word = ReplaceCharacter(Word, "", "")

        'Set RoadPrimaryRS = New Recordset
        'RoadPrimaryRS.Open "select [mobile no] from commuterlist Order by ID", dbRoad, adOpenStatic, adLockOptimistic

        FromRoadWord = Word
        Topstr = "SELECT * from roadmap where [road name] Like '%" & FromRoadWord & "%"
        'look at each char in text string
        Topstr = Topstr & "ORDER BY [road name]"

        Set dbRoad = New Connection
        dbRoad.CursorLocation = adUseClient
        dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
        Set RoadPrimaryRS = dbRoad.Execute(Topstr)

        Do Until RoadPrimaryRS.EOF

            If FromRoad = "" Then
                FromRoad = RoadPrimaryRS("road name")

            End If

            RoadPrimaryRS.MoveNext
        Loop

        Pos = i
        Ticker = 1
        'for middle words
        ElseIf Char = " " And Ticker = 1 And i <> Len(SortWord) Then
            Word = Mid(SortWord, Pos + 1, i - Pos - 1)
            Word = ReplaceCharacter(Word, "", "")
            If FromRoad = "" Then
                FromRoadWord = Word
                Topstr = "SELECT * from roadmap where [road name] Like '%" & FromRoadWord & "%"
                'look at each char in text string
                Topstr = Topstr & "ORDER BY [road name]"

                Set dbRoad = New Connection
                dbRoad.CursorLocation = adUseClient
                dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
                Set RoadPrimaryRS = dbRoad.Execute(Topstr)

                Do Until RoadPrimaryRS.EOF

                    If FromRoad = "" Then
                        FromRoad = RoadPrimaryRS("road name")

                    End If

                    RoadPrimaryRS.MoveNext
                Loop
            End If
        End If
    Next
End Sub

```



```

ElseIf ToRoad = "" And CheckWordPass = 1 Then
    ToRoadWord = Word
    Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = dbRoad.Execute(Topstr)

    Do Until RoadPrimaryRS.EOF

        If ToRoad = "" Then
            ToRoad = RoadPrimaryRS("road name")

        End If

        RoadPrimaryRS.MoveNext
    Loop

ElseIf FromRoadWord <> "" And CheckWordPass = 0 Then
    FromRoadWord = FromRoadWord & " " & Word

    Topstr = "SELECT * from roadmap where [road name] Like '%" & FromRoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = dbRoad.Execute(Topstr)
    Do Until RoadPrimaryRS.EOF
        If RoadPrimaryRS("road name") <> "" And CheckFromRoad = 0 Then
            CheckFromRoad = 1

            FromRoad = ""

        End If

        FromRoad = RoadPrimaryRS("road name")

    RoadPrimaryRS.MoveNext
    Loop
    If CheckFromRoad = 0 Then
        CheckWordPass = 1
        FromRoadWord = ""
        ToRoadWord = Word
        Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%"
        'look at each char in text string
        Topstr = Topstr & "ORDER BY [road name]"

        Set dbRoad = New Connection
        dbRoad.CursorLocation = adUseClient
        dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
        Set RoadPrimaryRS = dbRoad.Execute(Topstr)

        Do Until RoadPrimaryRS.EOF

            If ToRoad = "" Then
                ToRoad = RoadPrimaryRS("road name")

            End If

            RoadPrimaryRS.MoveNext
        Loop
    End If
ElseIf ToRoadWord <> "" And CheckWordPass = 1 Then
    ToRoadWord = ToRoadWord & " " & Word

    Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%"

```

```

'look at each char in text string
Topstr = Topstr & "ORDER BY [road name]"

Set dbRoad = New Connection
dbRoad.CursorLocation = adUseClient
dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
Set RoadPrimaryRS = dbRoad.Execute(Topstr)
Do Until RoadPrimaryRS.EOF
    If RoadPrimaryRS("road name") <> "" And CheckToRoad = 0 Then
        CheckToRoad = 1

        ToRoad = ""

    End If

    If ToRoad = "" Then
        ToRoad = RoadPrimaryRS("road name")

    End If

    RoadPrimaryRS.MoveNext
Loop
End If

CheckToRoad = 0
CheckFromRoad = 0
Pos = i
'for last word
ElseIf Ticker = 1 And i = Len(SortWord) Then
    Word = Mid(SortWord, Pos + 1, i - Pos - 1)
    Word = ReplaceCharacter(Word, "", "")
    If FromRoad = "" Then
        FromRoadWord = Word
        Topstr = "SELECT * from roadmap where [road name] Like '%" & FromRoadWord & "%'"
        'look at each char in text string
        Topstr = Topstr & "ORDER BY [road name]"

        Set dbRoad = New Connection
        dbRoad.CursorLocation = adUseClient
        dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
        Set RoadPrimaryRS = dbRoad.Execute(Topstr)

        Do Until RoadPrimaryRS.EOF

            If FromRoad = "" Then
                FromRoad = RoadPrimaryRS("road name")

            End If

            RoadPrimaryRS.MoveNext
Loop

ElseIf ToRoad = "" And CheckWordPass = 1 Then
    ToRoadWord = Word
    Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%'"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = dbRoad.Execute(Topstr)

    Do Until RoadPrimaryRS.EOF

        If ToRoad = "" Then
            ToRoad = RoadPrimaryRS("road name")

        End If

        RoadPrimaryRS.MoveNext
Loop

ElseIf FromRoadWord <> "" And CheckWordPass = 0 Then
    FromRoadWord = FromRoadWord & " " & Word

```

```

Topstr = "SELECT * from roadmap where [road name] Like '%" & FromRoadWord & "%"
'look at each char in text string
Topstr = Topstr & "ORDER BY [road name]"

Set dbRoad = New Connection
dbRoad.CursorLocation = adUseClient
dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
Set RoadPrimaryRS = dbRoad.Execute(Topstr)
Do Until RoadPrimaryRS.EOF
    If RoadPrimaryRS("road name") <> "" And CheckFromRoad = 0 Then
        CheckFromRoad = 1

        FromRoad = ""

    End If

    If FromRoad = "" Then
        FromRoad = RoadPrimaryRS("road name")

    End If

    RoadPrimaryRS.MoveNext
Loop
If CheckFromRoad = 0 Then
    CheckWordPass = 1
    FromRoadWord = ""
    ToRoadWord = Word
    Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = dbRoad.Execute(Topstr)

    Do Until RoadPrimaryRS.EOF

        If ToRoad = "" Then
            ToRoad = RoadPrimaryRS("road name")

        End If

        RoadPrimaryRS.MoveNext
    Loop
End If
ElseIf ToRoadWord <> "" And CheckWordPass = 1 Then
    ToRoadWord = ToRoadWord & " " & Word

    Topstr = "SELECT * from roadmap where [road name] Like '%" & ToRoadWord & "%"
    'look at each char in text string
    Topstr = Topstr & "ORDER BY [road name]"

    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = dbRoad.Execute(Topstr)
    Do Until RoadPrimaryRS.EOF
        If RoadPrimaryRS("road name") <> "" And CheckToRoad = 0 Then
            CheckToRoad = 1

            ToRoad = ""

        End If

        If ToRoad = "" Then
            ToRoad = RoadPrimaryRS("road name")

        End If

        RoadPrimaryRS.MoveNext
    Loop
End If
CheckToRoad = 0
CheckFromRoad = 0

```

```

    End If
Next
If FromRoad <> "" And ToRoad <> "" Then
    Call DirectionSearch
Else
    SMSText.Text = "No Results"
End If
End Sub
Private Sub DirectionSearch()
    MapCount = 3
    LinkCount = 10
    For i = 0 To 10
        FromMap(i) = ""
        ToMap(i) = ""
        FromLink(i) = ""
        ToLink(i) = ""
        Link(i) = ""
    Next
    Set dbRoad = New Connection
    dbRoad.CursorLocation = adUseClient
    dbRoad.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data Source=D:\documents\Masproj\MapGuide\roadmap.mdb;"
    Set RoadPrimaryRS = New Recordset
    RoadPrimaryRS.Open "select * from roadmap Order by ID", dbRoad, adOpenStatic, adLockOptimistic
    Do Until RoadPrimaryRS.EOF
        For i = 0 To 10
            If FromRoad = RoadPrimaryRS("road name") And FromRoad <> "" Then
                FromRegion = RoadPrimaryRS("region")
                FromRank = RoadPrimaryRS("rank")

                For j = 1 To MapCount
                    If RoadPrimaryRS("map" & j) <> "" Then
                        For k = 1 To MapCount
                            If FromMap(k) = "" Then
                                FromMap(k) = RoadPrimaryRS("map" & j)
                                Exit For
                            End If
                        Next
                    End If
                Next
                For j = 1 To LinkCount
                    If RoadPrimaryRS("link" & j) <> "" Then
                        For l = 1 To 10
                            If FromLink(l) = "" Then
                                FromLink(l) = RoadPrimaryRS("link" & j)
                                Exit For
                            End If
                        Next
                    End If
                Next
            End If
        Next
    End If
Next

For i = 0 To 10
    If ToRoad = RoadPrimaryRS("road name") And ToRoad <> "" Then
        ToRegion = RoadPrimaryRS("region")
        ToRank = RoadPrimaryRS("rank")

        For j = 1 To MapCount
            If RoadPrimaryRS("map" & j) <> "" Then
                For k = 1 To MapCount
                    If ToMap(k) = "" Then
                        ToMap(k) = RoadPrimaryRS("map" & j)
                        Exit For
                    End If
                Next
            End If
        Next
        For j = 1 To LinkCount
            If RoadPrimaryRS("link" & j) <> "" Then
                For l = 1 To 10
                    If ToLink(l) = "" Then
                        ToLink(l) = RoadPrimaryRS("link" & j)
                        Exit For
                    End If
                Next
            End If
        Next
    End If
End If

```

```

        Next
    End If
Next
RoadPrimaryRS.MoveNext
Loop

'Check whether both roads on the same maps
For i = 1 To MapCount
    For j = 1 To MapCount
        If FromMap(i) = ToMap(j) And FromMap(i) <> "" And ToMap(j) <> "" Then
            Call MapMatch
            Exit Sub
        End If
    Next
Next
If FromRegion = ToRegion And FromRegion <> "" And ToRegion <> "" Then
    Call RegionMatch
    Exit Sub
End If

End Sub
Private Sub MapMatch()
    CheckRoadLink = 0
    For k = 0 To 10
        For i = 0 To 10
            For j = 1 To 10
                If FromRoad = ToLink(j) And ToLink(j) <> "" And CheckRoadLink = 0 Then
                    CheckRoadLink = 1
                    If RdTake(k) = "" Then
                        RdTake(k) = ToLink(j)
                    Exit For
                End If
            End If
        Next
    Next
Next
Next

If CheckRoadLink = 1 Then
    SMSText.Text = RdTake(0) & ", " & ToRoad
ElseIf CheckRoadLink = 0 Then
    For i = 1 To 10
        For j = 1 To 10
            If FromLink(i) = ToLink(j) Then
                Link(0) = FromLink(i)
            End If
        Next
    Next
    SMSText.Text = FromRoad & ", " & Link(0) & ", " & ToRoad
End If

End Sub
Private Sub RegionMatch()
    CheckRoadLink = 0
    For k = 0 To 10
        For i = 0 To 10
            For j = 1 To 10
                If FromRoad = ToLink(j) And ToLink(j) <> "" And CheckRoadLink = 0 Then
                    CheckRoadLink = 1
                    If RdTake(k) = "" Then
                        RdTake(k) = ToLink(j)
                    Exit For
                End If
            End If
        Next
    Next
Next
Next
If CheckRoadLink = 1 Then
    SMSText.Text = RdTake(0) & ", " & ToRoad
ElseIf CheckRoadLink = 0 Then
    For i = 1 To 10
        For j = 1 To 10
            If FromLink(i) = ToLink(j) Then
                Link(0) = FromLink(i)
            End If
        Next
    Next

```

```

Next

    SMSText.Text = FromRoad & ", " & Link(0) & ", " & ToRoad
End If
End Sub
Function ReplaceCharacter(TextIn, Optional ByVal SearchStr As String = "", _
    Optional ByVal Replacement As String = "", _
    Optional ByVal CompMode As Long = 0)

'replace character "" with "" for easy searching algo
Dim WorkText As String, Pointer As Long

If IsNull(TextIn) Then
    ReplaceCharacter = Null
Else
    WorkText = TextIn
    Pointer = InStr(1, WorkText, SearchStr, vbTextCompare)

    Do While Pointer > 0
        WorkText = Left(WorkText, Pointer - 1) & Replacement & _
            Mid(WorkText, Pointer + Len(SearchStr))
        Pointer = InStr(Pointer + Len(Replacement), WorkText, SearchStr, CompMode)
    Loop
    ReplaceCharacter = WorkText
End If

DoEvents
End Function
VERSION 5.00
Begin VB.Form Security
    Caption       = "Vehicle Security system"
    ClientHeight  = 5850
    ClientLeft    = 60
    ClientTop     = 345
    ClientWidth   = 5895
    LinkTopic     = "Form1"
    ScaleHeight   = 5850
    ScaleWidth    = 5895
    StartUpPosition = 3 'Windows Default
Begin VB.Data carparkinfo
    Caption       = "carparkInfo"
    Connect       = "Access 2000;"
    DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
    DefaultCursorType= 0 'DefaultCursor
    DefaultType   = 2 'UseODBC
    Exclusive     = 0 'False
    Height        = 375
    Left          = 4080
    Options       = 0
    ReadOnly      = 0 'False
    RecordsetType  = 1 'Dynaset
    RecordSource  = "carparks"
    Top           = 3360
    Visible       = 0 'False
    Width         = 1695
End
Begin VB.Data carparkC
    Caption       = "carparkC"
    Connect       = "Access 2000;"
    DatabaseName  = "D:\documents\Masproj\CarParking\car_parks.mdb"
    DefaultCursorType= 0 'DefaultCursor
    DefaultType   = 2 'UseODBC
    Exclusive     = 0 'False
    Height        = 375
    Left          = 4080
    Options       = 0
    ReadOnly      = 0 'False
    RecordsetType  = 1 'Dynaset
    RecordSource  = "carparkC"
    Top           = 2880
    Visible       = 0 'False
    Width         = 1695
End

```

```

Begin VB.Data carpkB
  Caption      = "carparkB"
  Connect      = "Access 2000;"
  DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
  DefaultCursorType= 0 'DefaultCursor
  DefaultType  = 2 'UseODBC
  Exclusive    = 0 'False
  Height       = 375
  Left         = 4080
  Options      = 0
  ReadOnly     = 0 'False
  RecordsetType = 1 'Dynaset
  RecordSource = "carparkB"
  Top          = 2400
  Visible      = 0 'False
  Width        = 1695
End
Begin VB.Data carpkA
  Caption      = "carparkA"
  Connect      = "Access 2000;"
  DatabaseName = "D:\documents\Masproj\CarParking\car_parks.mdb"
  DefaultCursorType= 0 'DefaultCursor
  DefaultType  = 2 'UseODBC
  Exclusive    = 0 'False
  Height       = 375
  Left         = 4080
  Options      = 0
  ReadOnly     = 0 'False
  RecordsetType = 1 'Dynaset
  RecordSource = "carparkA"
  Top          = 1920
  Visible      = 0 'False
  Width        = 1695
End
Begin VB.TextBox Text4
  Height       = 375
  Left         = 1080
  TabIndex     = 20
  Top          = 5160
  Width        = 2775
End
Begin VB.Timer Timer1
  Interval     = 1000
  Left         = 5160
  Top          = 720
End
Begin VB.TextBox Text1
  DataSource   = "Data1"
  Height       = 375
  Left         = 1080
  TabIndex     = 8
  Top          = 3600
  Width        = 2775
End
Begin VB.TextBox Hits
  Height       = 375
  Left         = 3840
  TabIndex     = 7
  Top          = 720
  Width        = 735
End
Begin VB.TextBox SMSText
  Height       = 1455
  Left         = 360
  MaxLength    = 160
  MultiLine   = -1 'True
  TabIndex     = 6
  ToolTipText = "Write the message here (max 160 characters)"
  Top          = 1920
  Width        = 3495
End
Begin VB.TextBox OtherEndAddress
  DataSource   = "Data1"
  Height       = 285
  Left         = 360
  TabIndex     = 5

```

```

    Top      = 720
    Width    = 2535
End
Begin VB.TextBox SMSCAddress
    Height   = 285
    Left     = 360
    TabIndex = 4
    ToolTipText = "Short message centre address"
    Top      = 1320
    Width    = 2535
End
Begin VB.CommandButton Close
    Caption   = "Close"
    Height    = 375
    Left      = 4560
    TabIndex  = 3
    Top       = 5160
    Width     = 1095
End
Begin VB.CommandButton Command1
    Caption   = "Send"
    Height    = 375
    Left      = 4560
    TabIndex  = 2
    Top       = 4560
    Width     = 1095
End
Begin VB.TextBox Text2
    DataSource = "Data1"
    Height     = 375
    Left       = 1080
    TabIndex   = 1
    Top        = 4080
    Width      = 2775
End
Begin VB.TextBox Text3
    DataSource = "Data1"
    Height     = 375
    Left       = 1080
    TabIndex   = 0
    Top        = 4560
    Width      = 2775
End
Begin VB.Label Label8
    Caption   = "Mobile No:"
    Height    = 255
    Left      = 120
    TabIndex  = 19
    Top       = 4680
    Width     = 855
End
Begin VB.Label Label7
    Caption   = "Car Park:"
    Height    = 255
    Left      = 120
    TabIndex  = 18
    Top       = 4200
    Width     = 735
End
Begin VB.Label Label6
    Caption   = "Plate No:"
    Height    = 255
    Left      = 120
    TabIndex  = 17
    Top       = 3720
    Width     = 855
End
Begin VB.Label Label5
    Caption   = "Info: Carpark"
    Height    = 255
    Left      = 120
    TabIndex  = 16
    Top       = 5280
    Width     = 975
End
Begin VB.Label Label1

```



```

Caption    = "Message"
Height    = 255
Left      = 360
TabIndex  = 15
Top       = 1680
Width     = 735
End
Begin VB.Label FromTo
Caption    = "From"
Height    = 255
Left      = 360
TabIndex  = 14
Top       = 480
Width     = 1935
End
Begin VB.Label Label2
Caption    = "SMSC Address"
Height    = 255
Left      = 360
TabIndex  = 13
Top       = 1080
Width     = 1575
End
Begin VB.Label Label3
Caption    = "Message status:"
Height    = 255
Left      = 360
TabIndex  = 12
Top       = 120
Width     = 1215
End
Begin VB.Label MessageStatus
Height    = 255
Left      = 1680
TabIndex  = 11
Top       = 120
Width     = 2295
End
Begin VB.Label Length
Caption    = "0"
Height    = 255
Left      = 3120
TabIndex  = 10
Top       = 1680
Width     = 375
End
Begin VB.Label Label4
Caption    = "/ 160"
Height    = 255
Left      = 3480
TabIndex  = 9
Top       = 1680
Width     = 495
End
End
Attribute VB_Name = "parking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim PlateNo, MobileNo, CarparkId
Dim HourCal, HourPay
Dim MinCal, MinPay
Dim TotalParkFee, StartStop
Dim CapCount, CarAlert, AlertCap
Dim CouponAlert

Private Sub Close_Click()
Unload parking
End Sub

Private Sub Command1_Click()
If Text1.Text <> "" And Text2.Text <> "" And Text3.Text <> "" And Text4.Text = "" Then
Call ParkCoupon

```

```

ElseIf Text1.Text = "" And Text2.Text = "" And Text3.Text = "" And Text4.Text <> "" Then
    Call ParkInfo
Else
    SMSText.Text = "No Results"
End If
End Sub

Private Sub SMSText_Change()
    Length.Caption = Str(Len(SMSText.Text))
End Sub

Private Sub ParkInfo()
    CarparkId = Null
    CarAlert = 0
    AlertCap = 0
    CarparkId = Text4.Text
    If CarparkId = "a" Or CarparkId = "A" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "a" Then
                CarAlert = carpkinfo.Recordset("car_alert")

                End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "b" Or CarparkId = "B" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "b" Then
                CarAlert = carpkinfo.Recordset("car_Alert")
                AlertCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    ElseIf CarparkId = "c" Or CarparkId = "C" Then
        carpkinfo.Refresh
        Do Until carpkinfo.Recordset.EOF
            If carpkinfo.Recordset("carparks") = "c" Then
                CarAlert = carpkinfo.Recordset("car_Alert")
                AlertCap = carpkinfo.Recordset("capacity")
            End If
            carpkinfo.Recordset.MoveNext
        Loop
    End If

    SMSText.Text = "Current capacity for car park:" & CarparkId & " is " & CarAlert & " out of " & AlertCap & "."

    Text4.Text = ""
End Sub

Private Sub ParkCoupon()
    PlateNo = Null
    CarparkId = Null
    MobileNo = Null
    StartStop = 0
    CapCount = 0
    CouponAlert = 0
    PlateNo = Text1.Text
    CarparkId = Text2.Text
    MobileNo = Text3.Text

    If CarparkId = "a" Or CarparkId = "A" Then
        carpkA.Refresh
        Do Until carpkA.Recordset.EOF
            If carpkA.Recordset("plate_no") = PlateNo Then
                If carpkA.Recordset("time_end") = 0 Then
                    StartStop = 1
                    StartTime = carpkA.Recordset("time_start")
                    Call cmdDifference
                    With carpkA.Recordset
                        .Edit
                        !time_end = Format(Now, "Short Time")
                        !fee_Alert = TotalParkFee
                        .Update
                    End With
                End If
            End If
        End If
    End If

```

```

End If
carpkA.Recordset.MoveNext
Loop
If StartStop = 0 Then
  With carpkA.Recordset
    .AddNew
    !mobile_no = MobileNo
    !plate_no = PlateNo
    !date_park = Date$
    !time_start = Format(Now, "Short Time")
    .Update
  End With
  StartTime = Format(Now, "Short Time")
End If
carpkA.Refresh
Do Until carpkA.Recordset.EOF
  If carpkA.Recordset("time_end") = 0 Then
    CapCount = CapCount + 1
  End If
carpkA.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
  If carpkinfo.Recordset("carparks") = "a" Then
    With carpkinfo.Recordset
      .Edit
      !car_Alert = CapCount
      .Update
    End With
  End If
carpkinfo.Recordset.MoveNext
Loop
ElseIf CarparkId = "b" Or CarparkId = "B" Then
carpkB.Refresh
Do Until carpkB.Recordset.EOF
  If carpkB.Recordset("plate_no") = PlateNo Then
    If carpkB.Recordset("time_end") = 0 Then
      StartStop = 1
      StartTime = carpkB.Recordset("time_start")
      Call cmdDifference
      With carpkB.Recordset
        .Edit
        !time_end = Format(Now, "Short Time")
        !fee_Alert = TotalParkFee
        .Update
      End With
    End If
  End If
carpkB.Recordset.MoveNext
Loop
If StartStop = 0 Then
  With carpkB.Recordset
    .AddNew
    !mobile_no = MobileNo
    !plate_no = PlateNo
    !date_park = Date$
    !time_start = Format(Now, "Short Time")
    .Update
  End With
  StartTime = Format(Now, "Short Time")
End If
carpkB.Refresh
Do Until carpkB.Recordset.EOF
  If carpkB.Recordset("time_end") = 0 Then
    CapCount = CapCount + 1
  End If
carpkB.Recordset.MoveNext
Loop
carpkinfo.Refresh
Do Until carpkinfo.Recordset.EOF
  If carpkinfo.Recordset("carparks") = "b" Then
    With carpkinfo.Recordset
      .Edit
      !car_Alert = CapCount
      .Update
    End With
  End If

```

```

    End If
    carpkinfo.Recordset.MoveNext
Loop

ElseIf CarparkId = "c" Or CarparkId = "C" Then
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("plate_no") = PlateNo Then
            If carpkC.Recordset("time_end") = 0 Then
                StartStop = 1
                StartTime = carpkC.Recordset("time_start")
                Call cmdDifference
                With carpkC.Recordset
                    .Edit
                    !time_end = Format(Now, "Short Time")

                    .Update
                End With
            End If
        End If
        carpkC.Recordset.MoveNext
    Loop
    If StartStop = 0 Then
        With carpkC.Recordset
            .AddNew
            !mobile_no = MobileNo
            !plate_no = PlateNo
            !date_park = Date$
            !time_start = Format(Now, "Short Time")
            .Update
        End With
        StartTime = Format(Now, "Short Time")
    End If
    carpkC.Refresh
    Do Until carpkC.Recordset.EOF
        If carpkC.Recordset("time_end") = 0 Then
            CapCount = CapCount + 1
        End If
    Loop
    carpkinfo.Refresh
    Do Until carpkinfo.Recordset.EOF
        If carpkinfo.Recordset("carparks") = "c" Then
            With carpkinfo.Recordset
                .Edit
                !car_Alert = CapCount
                .Update
            End With
        End If
        carpkinfo.Recordset.MoveNext
    Loop
End If

If CouponAlert <> 0 Then
    SMSText.Text = "You can alert the police by forward this message"
ElseIf CouponAlert = 0 Then
    SMSText.Text = "Your Vehicle had encounter a breach of security:" & Text2.Text & "."
End If

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
End Sub

```

Appendix C: Details of Publications

In the process of doing the M.Eng project, the following paper are being published

1. T.C Lim and H.K Garg, “Designing SMS Applications for Public Transport Service System in Singapore”, *8th IEEE International Conference on Communication Systems 2002 (ICCS 2002)*, Singapore, Nov 2002

Appendix D: Singapore Map with MRT routes

