OXFORD BROOKES UNIVERSITY

Oxford Brookes University

BSc (Hons) Computer and Information Systems

U08096 Final Year Project

SMS Service Tracker

Supervisor	: Lo Chi Wing, Peter
Student Name	: Choy Man Kong, Jeffrey
Student ID	: 0301-0107-0324
Module Number	: U08096
Submission Date	: 31 Jul 2008

OXFORD BROOKES UNIVERSITY

Project Summary

SURNAME **CHOY MAN KONG** FORENAME **JEFFREY**

PROJECT TITLE **SMS SERVICE TRACKER**

1) Use not more then 150 words to descript your project briefly and what are the main objects of your project.

Location aware smart phones are becoming more and more popular, and location based services is a new way to allow mobile phone users to interact between information and the communication devices. By using features available in location aware smart phones, users can carry out location based interactions with their mobile phone applications to achieve its goals. My project is to study various locations based services and its relevant technologies to implement a service tracker. The implementation includes a mobile phone interaction, application a location parsing and replying service, and a web application for handling the location details. The implementation the service allows easy finding of useful information of a service allows easy finding of useful information of a service, such as an ATM machine, on where you are located.

2) What is the implication of your project? Eg .your idea can it be used as a solution for some other problem ? Or any relevance to the current issues in computing and the application domain

From the project results, SMS Service Tracker is proved to be very reliable and produces result quickly. If the tools are widely deployed, it will greatly enhance the daily life of every user, not only for the said services, but can also be a tracker of other services, from bus stations to restaurants to anything imaginable. The tools can also be developed a bit differently to become a social utility to allow people within the same area to chat and discuss about the services or shops within the same area.

OXFORD BROOKES UNIVERSITY

3) What methodology you applied in your project and how you applied. Eg. Any Data structure(stack, tree..), algorithm(Searching, Sorting ,graph theory.) or database design, etc.

As discussed in the report, the SMS retrieval and delivery service is written using the Object Oriented Methodology, from the AT commands controller, the Win32 Service to the database connection. By implementing OO, the complexity of each layer is hidden from the above layer and the codes can be easily followed.

4) What techniques you used in solving the problem and how the problem was solved. Eg. ASP, JSP, DAO, CGI , Active-X, Database Server, SQL, Find Shortest Path or you own method.

There are several techniques used to solving the problem, including the PHP scripting, MySQL database sever, Apache Webserver. As mentioned in the report, the SMS retrieval and delivery service uses PHP to act as a service in Windows, and by using the service, the retrieval and sending of SMS runs continuously without interruption, and everything goes straight to the database. The web interface is written in PHP too, and it uses Apache Webserver to allow others to access the interface from within the local area network.

5) If you do this project again which area you should / can improve? eg. Technique, written presentation, module, research area, scope, complexity, capacity, etc.

To do the project again, I will carefully evaluate the required tools further, also research on ways to deal with the problems discussed in the Possible Improvements and Future Work section of the report, such as mobile phone platform issue, mobile operator issue, multiple response issue, security issue, etc. I will also spend more time to write the report in a more professional manner.

6) What you have learned from doing this project?

I have learnt a number of mobile and server platform development technologies, and also web and database technologies that are useful to my project. I have learnt how to manage a project, from scheduling the times, researching, preparing the required materials, implementing the project, work around with problems, testing the results and writing the report.

OXFORD BROOKES UNIVERSITY

Abstract

Nowadays, every person has a mobile phone and it has become an essential tool for human lives. While people call modern mobile phones as smart phones, most of them are still a basic communication device with Internet connectivity only. To fully utilize mobile phone as a communication platform, new technologies and services that allow you to interact with information are desirable, and that can greatly enhance our lifestyle.

Location aware smart phones are becoming more and more popular, and location based services is a new way to allow mobile phone users to interact between information and the communication devices. By using features available in location aware smart phones, users can carry out location based interactions with their mobile phone applications to achieve its goals. Imagine that you want to find an ATM machine, what you can do at the moment is to either call up a friend or to use your smart phone to search for the information on the Internet manually. Though, by implementing the services in this project, you can instantly get the information by simply specifying ATM, and your smart phone will return you with the location of the ATM that is closest to you.

In this project, various location based services and its relevant technologies will be studied. An implementation of the mobile phone interaction application, a location parsing and replying service, and a web application for handling the location details will then be built. Finally the results and further enhancements that can be made will be discussed.

OXFORD BROOKES UNIVERSITY

Acknowledgements

I would like to take this opportunity to express my sincere thanks to Peter Lo, my project supervisor, for his valuable advice on the project, his useful ideas and information on the technical aspects of the project. His encouragements to my self proposed project have made me feel confident in this Final Year Project.

I would like to acknowledge my grateful to Mr. Vincent Ching, for his helpful guidance, his experience on writing the report skills and spent his valuable time with me to discuss the proposal and structure of report.

In addition, special thanks must give to my colleague, Mr. Edwin Chan, for his advises on PHP and English skills on writing this report.

Jeffrey Choy
14 July 2008

OXFORD BROOKES UNIVERSITY

Table of Contents

Project Submission Form……..1

Project Summary.....1

Abstract.....3

Acknowledgements.....4

Table of Contents.....5

Chapter 1 Introduction.....10

1.1 Background.....10

1.2 Motivation.....10

1.3 Objective.....11

1.4 Assumptions.....11

1.5 Organization of the Report.....12

Chapter 2 Literature Review of.....14

Technologies.....14

2.1 Location Based Technologies.....14

2.1.1 GPS technology.....14

2.1.1.1 Description.....14

2.1.1.2 Examples.....15

2.1.1.3 Advantages and Disadvantages.....15

2.1.2 Wi-Fi technology.....16

2.1.2.1 Description.....16

2.1.2.2 Examples.....16

2.1.3.3 Advantages and Disadvantages.....16

2.1.3 SMS technology.....17

2.1.3.1 Description.....17

2.1.3.2 Examples.....18

2.1.3.3 Advantages and Disadvantages.....18

2.2 Smart Phone Technologies.....19

2.2.1 Windows Mobile.....19

2.2.1.1 Description 19
2.2.1.2 Advantages and Disadvantages 19
2.2.2 Symbian OS 20
2.2.2.1 Description 20
2.2.2.2 Advantages and Disadvantages 20
2.2.3 BlackBerry 21
2.2.3.1 Description 21
2.2.3.2 Advantages and Disadvantages 21
Chapter 3 Analysis of Problem Requirements 22
3.1 Requirements of the Program 22
3.1.1 Features of the program 22
3.1.1.1 The Mobile Phone Platform 22
3.1.1.2 The Server Platform 23
3.1.2 Interface requirements 24
3.1.2.1 The Mobile Phone Interface 24
3.1.2.1 The Web Interface 24
3.1.3 Database requirements 24
3.2 Choice of Technologies 25
3.2.1 Choice of Location Technology 25
3.2.2 Choice of Mobile Platform and Development Technology 25
3.2.3 Choice of Web and Database Technologies 26
3.2.4 Choice of Server Platform Development Technology 26
3.3 Project Requirements 27
3.3.1 Mobile Phone Platform 27
3.3.2 Server Platform 28
Chapter 4 Design and Implementation 29
4.1 Information System Functions Design 29
4.1.1 Mobile Phone Interface 30
4.1.2 SMS Parsing Function 32
4.1.3 Service Information Sending Function 33
4.1.4 Service Information Web Interfaces 35

4.2 Database Design....36
4.2.1 SMS Handling....36
4.2.2 Service Information....37
4.3 Implementation....38
4.3.1 The Mobile Phone Interface....38
4.3.1.1 Setting up the Mobile Phone....38
4.3.1.2 Setting up the Development Environment....40
4.3.1.3 Creating the Program and Transferring It to the Phone....40
4.3.2 The Server Interface....42
4.3.2.1 Setting up the Servers....42
4.3.2.2 Setting up the Development Environment....42
4.3.2.3 Developing the Programs....43
4.3.3 Location Information Retrieval....48
4.3.3.1 Switching on the Debug mode....48
4.3.3.3 Retrieving the Cell Site IDs....48
Chapter 5 Results and Discussions....49
5.1 Testing Methodology....49
5.2 Testing Results....49
5.2.1 The Server Backend....49
5.2.1.1 Running Server Backend as Service....50
5.2.1.2 Running Server Backend from Command Line....52
5.2.2 The Phone Interface....54
5.2.2.1 Start Python....54
5.2.2.2 Executing the Python Script....55
5.2.2.3 Test Results....57
5.2.3 The Web Interface....58
5.2.3.1 The Index....58
5.2.3.2 View Details Screen....59
5.2.3.3 Add Records....60
5.2.3.4 Edit Records....61
5.2.3.5 Delete Records....62

5.3 Discussions 63
5.3.1 Discussion on Achievements 63
5.3.2 Discussion on Performance and Problems Encountered 64
5.3.2.1 System Performance 64
5.3.2.2 Problems Encountered 64
5.3.3 Possible Improvements and Future Work 65
Chapter 6 Conclusion 67
References and Citation 68
Appendix 75
A. Program Source Codes Listing 75
A.1 Client Side Program – Python Client 75
A.1.1 sms_cellid.py – The Client Side Interface 75
A.2 Server End Program – SMSServer (PHP) 77
A.2.1 server.php – The Main Driver 77
A.2.2 classes/ATCommands.php – The AT Commands Controller 77
A.2.3 classes/DB.php – The Database Abstract Class 80
A.2.4 classes/DB_MySQL.php – The MySQL Database Driver 81
A.2.5 classes/SMSParser.php – The SMS Parser class 81
A.2.6 classes/SMSServer.php – The SMS Server Class 83
A.2.7 classes/Win32Service.php – The Win32 Interface 86
A.3 Server End Program – Web Interface for Editing Details (PHP) 87
A.3.1 index.php – the Main Interface 87
A.3.2 db_functions.php – The Database Functions 88
A.3.4 footer.php – The Script to Output the Footer 93
A.3.5 del.php – The Script to Delete Details from Database 93
A.3.6 add_edit.php – The Script to Add / Edit Details 94
A.3.7 add_edit_process.php – Scripts to Process Adds and Edits 95
A.3.8 config.php – The Configuration Script 96
A.3.9 db_connect.php – The Database Connection Script 96
A.3.10 db_disconnect.php – The Database Disconnection Script 96
B. Database Structure Listing 97

B.1 The Database 97
B.1.1 fyp – The Main Database 97
B.2 The SMS mailboxes 97
B.2.1 sms_inbox – The SMS Inbox Table 97
B.2.1 sms_outbox – The SMS Inbox Table 97
B.3 The Information Services 98
B.3.1 atmsrv – The ATM Service Table 98
B.3.2 bankcslist – The Bank Cellsite List Table 98
B.3.3 branch – The Branch Table 98
B.3.4 cellsite – The Cellsite Table 98
B.3.5 company – The Company Table 99
B.3.6 district – The District Table 99
B.3.7 machine – The Machine Table 99
B.3.8 petrolcslist – The Petrol Cellsite List Table 99
B.3.9 ptlsrv – The Petrol Service Table 99
B.3.10 station – The Petrol Station Table 100
B.3.11 street – The Street Table 100
B.3.12 type – The Service Type Table 100
C. Project Proposal 101
D. Project Progress 107
E. MyDropBox Report 113
F. Presentation 114

OXFORD BROOKES UNIVERSITY

Chapter 1 Introduction

1.1 Background

In recent rapid development of the century, mobile phones have become an indispensable communications tool [1][2], and many features have now been provided on top of the mobile phone platform [3]. While mobile phone has provided easy access to various Internet services, such as mobile banking, some of the features have never been thought of. In order to fully utilize mobile phone as a communication platform, some new services can be introduced to enhance our lifestyle.

1.2 Motivation

Mobile phones have become a very common device and they are now similar to clothes, which people wear them on no matter where they are. With a mobile phone constantly reflecting the location of a person, by knowing surrounding information about the location where the person is at, can dramatically increase the functionality of a mobile phone and to greatly enhance the daily life of a person.

It is very common that you may want to find a specific type of shop that you have never been to, and has no knowledge of the location of when you are on the street. Say if you want to find an ATM machine of a particular bank, what you would normally do is to blindly ask pedestrians or call up a friend, and hope that the person can answer you the correct information. This action may be very time consuming and there is a high chance that you will not get the correct information. However, by using a mobile phone as a

communication platform, a service that can utilize the location information of the mobile phone can now be implemented, and easily find out the information of a service, such as an ATM machine, on where you are located.

1.3 Objective

There are a number of location based technologies available. However, most of the services have different and unique hardware requirements in order to provide information of the surrounding services available to a mobile phone user. To overcome the technical challenges, different types of location based technologies, the mobile phone platforms and determine the cheapest and least system overhead approach to the problem will be reviewed.

After having chose the technology, the objective of the remaining of the project is to develop an information system that provides useful details and information on where he or she is located, and together with a customizable interface, users not only can access information, but they can now interact with information using a communication device.

1.4 Assumptions

In this SMS Service Tracker's project, all interfaces, information and location details are assumed to be in English only and the project may not function in other languages, such as Traditional Chinese.

OXFORD BROOKES UNIVERSITY

1.5 Organization of the Report

This report is separated into 6 chapters and the contents of each chapter are explained below.

Chapter 1 Introduction

This chapter explains the background, motivation, objectives and assumptions of the problem.

Chapter 2 Literature Review of Technologies

This chapter introduces various kinds of technologies. It will review the current location technologies including GPS, Wi-Fi and SMS. It will also review the current smart phone technologies such as the Windows Mobile, Symbian OS and BlackBerry OS platform.

Chapter 3 Analysis of Problem Requirements

This chapter analyzes the problem requirements and provides the reasons behind the choices of technologies in the project.

Chapter 4 Design and Implementation

This chapter details the flow of the system, discussion on the programming logic of the project and provides the details of the database structure.

OXFORD BROOKES UNIVERSITY

Chapter 5 Results and Discussions

This chapter explains the results of the project, and discusses on the success or failure of the results and provides details on various improvements that can be made.

Chapter 6 Conclusion

This chapter provides a summary of the problem and concludes the SMS Service Tracker project.

OXFORD BROOKES UNIVERSITY

Chapter 2 Literature Review of Technologies

2.1 Location Based Technologies

Location based technologies are highly demanded for resource tracking or proximity based notification [4], such as locating a parcel or sending targeted advertising. There are a couple of technologies that can be used to transfer location information, such as GPS, Wi-Fi and SMS. In this chapter, the above technologies, their advantages and disadvantages, and examples of each technology will be reviewed and discussed.

2.1.1 GPS technology

GPS stands for Global Positioning System and it is a location technology that requires access of the GPS satellites in orbit.

2.1.1.1 Description

GPS tracking is a technology that uses the freely available Global Positioning System to track for location information. Microwave signals from Medium Earth Orbit satellites are received to calculating location, speed, direction, and time [5].

The GPS navigation system is made up of a network of 24 satellites that orbit around the earth to perform location tracking functions [6]. There are 3 types of GPS trackers: i) data logger, ii) data pusher, and iii) data puller.

A data logger logs the relative GPS data in a regular interval or when it is required. It is widely used in outdoor sport to track down the position, speed, direction and time of

competitors. Digital photographing also uses it as a geotagging in the EXIF information for each photo.

A data pusher may not log its GPS data in memory; however, it will send the GPS data to a server at a regular interval. It is usually used for security reason, such as surveillance and animal control.

A data puller queries for GPS data when needed. It is used on some mobile map applications.

2.1.1.2 Examples

In China, Beijing will establish an electronic guidance system for public lavatories before the Olympic Games. People can use guides set up on 210 roads or a GPS locator to find out the closest one among 5600 public lavatories [7].

In Hong Kong, Kowloon Motor Bus has trialed the Global Positioning System to allow passengers to view the location of the bus they want to abroad on the bus route display screen at the bus station, and the control room can also use the system to track the statuses of the buses. Unfortunately, due to the environment of Hong Kong, where many high rise buildings blocked the signals, the project was terminated [8].

2.1.1.3 Advantages and Disadvantages

The advantage of the GPS is that the system is freely available anywhere around the world, a standalone GPS receiver is cheap, and the accuracy of consumer level GPS receivers is within 100 meters of distance [9][44]. The disadvantage of GPS, however, is that it is not possible to access GPS satellites underground or in dense populated area, such as in Hong Kong, where many high rise buildings are actually blocking the coverage of the GPS and unable to retrieve the GPS signals. On the other hand, GPS receiver is not common in mobile phones and only very few smart phones have built in GPS receiver.

2.1.2 Wi-Fi technology

Wi-Fi is a frequently used wireless technology that is supported in many devices, from computer, video games machines to many mobile phones.

2.1.2.1 Description

Similar to GPS tracking, Wi-Fi Positioning System (WPS) uses the MAC addresses and signal strength of nearby Wi-Fi access point to determine the location of the device with 20-30 meter accuracy and near 100% availability indoors and in dense urban areas [10].

2.1.2.2 Examples

Many hospitals have adopted Wi-Fi Patient Tracking System. The system is designed to avoid medical errors and improve hospital efficiency. Patients are tagged with Wi-Fi transmitters for tracking their movement around the hospital. Beside the tracking function, the Wi-Fi transmitters can also integrate with monitoring devices and send the collected measurements to the dedicated server at a regular interval or when threshold is reached.

2.1.3.3 Advantages and Disadvantages

The advantage is that Wi-Fi is a cheap technology, the cost of building a Wi-Fi hotspot is not expensive and can be done in a small amount of budget. Many countries have widely used it for local area network access or used as a tracking system. Due to the signal strength of Wi-Fi, the accuracy to the actual distance is usually within 100m.

The disadvantage, on the other hand, is that the signal range of Wi-Fi is limited. In Hong Kong, only certain places have Wi-Fi hotspots, which also mean that there are many blind spots when using the Wi-Fi technology. While more smarts phones have built in Wi-Fi feature than GPS receivers, Wi-Fi functionality is still not too common on smart phones.

2.1.3 SMS technology

SMS stands for Short Message Service and is used in mobile phone to transmit and receive text messages between mobile phone users using the Short Message Service Center (SMSC) of a mobile operator [11]. Each SMS message can contain up to 140 ASCII characters [12], or a maximum of 160 characters in English words.

2.1.3.1 Description

Beside the person-to-person messaging or interactive information services used on the SMS applications, location-based services are also a value-added function with the use of SMS [13][43].

Many people refers mobile phones as cellular, the reason is because the concept of mobile phone systems is to dividing different small regions into cells – a coverage area of the wireless receiver, which is normally called a cell site by mobile operators. In order to provide continuous service to the mobile connection, such that when a person moves from one cell to another, a hidden cell change process will need to take place [14]. To achieve that, each cell must have unique cell site identification called the cell site ID, for the hidden cell change process automatically and accurately transfers a phone or data connection and hand it over to the neighboring cell [15][44].

With the characteristics of the mobile phone platform as discussed above, smart phones can base on the cell site id to relate a mobile phone to a particular location, such as an address or a building. When specific location information, various value added features can be provided, such as mobile operators can base on the cell site id of where a person is located to send an advertising SMS to the mobile phone of their customer, or programmers can program the smart phone to retrieve the cell site id and perform some other location related functions based on information using the SMS technology.

2.1.3.2 Examples

A Singaporean telecom operator called StarHub launched a location-based mobile advertising service in April 2008. With the location-based service, marketers can advertise in key locations, such as shopping malls, to broadcast advertisements to their targeted consumers [16].

2.1.3.3 Advantages and Disadvantages

The advantage of the SMS technology is that basically all mobiles phones, whether or not they are smart phones, all have SMS built it and hence no extra cost or hardware upgrades are required for using SMS based technology.

The disadvantage of SMS is that by default, SMS does not contain any location based information and the functionality has to be manually integrated, and different type of smart phones requires a different way to handle the location information, in some cases, the smart phone does not even have any way to retrieve the location information.

In addition, since the number of cell sites greatly affects the accuracy of the results, if there are only very few cell sites around, which are very common in rural area of many countries, and the information retrieved may be as far as 100km from the real location in some extreme cases [14]. However, Hong Kong is a dense populated area and mobile phone operators have so many cell sites around districts and there may not be a problem using this technology as usually a cell site is less than 100m away from the actual location in Hong Kong.

Moreover, since SMS is usually in plain text, without encoding of messages, there is basically no security at all and spoofing of messages may be possible.

OXFORD BROOKES UNIVERSITY

2.2 Smart Phone Technologies

There are currently a number of smart phone technologies available, the most commonly used technologies are the Symbian OS, Windows Mobile and BlackBerry. In this section, each technology will be reviewed in detail.

2.2.1 Windows Mobile

Windows Mobile is an operating system developed by Microsoft to simulate an interface similar to the desktop version of Microsoft Windows.

2.2.1.1 Description

The first release of the Windows Mobile is called Pocket PC 2000; it is based on the Windows CE technology and was introduced in April 2000 [17]. The main feature is to create a smart phone operating system that integrates both wireless and data into a single device, such that user can browse the web and retrieve e-mails from the smart phone, or to view office documents on the Windows Mobile device [18].

Similar to Microsoft Windows, the compact OS is based on the Win32 programming API to allow people to extend the functionality of the smart phones [19]. The native language of the mobile phone is C++ and developers are recommended to use Visual C++ to developed software with it [20], a compact version of the .NET framework is also available, such that users can develop software using Visual Basic .NET, C#, and other Microsoft proprietary development languages [21].

2.2.1.2 Advantages and Disadvantages

The advantage of Windows Mobile is that it is very similar to desktop operating systems and the interface is familiar to most users, the software development is also similar to that of Windows Operating System. The disadvantages, on the other hand, are that in compare with another popular operating system, the Symbian OS, Windows Mobile is a less common platform and also Win32 API and Visual C++ is proprietary and not a free development system for programmers to develop software.

2.2.2 Symbian OS

Symbian OS is an operating system developed specifically for mobile devices and it is the leading smart phone operating system with 65% market share [22].

2.2.2.1 Description

Symbian was founded in 1998 as a software licensing company that is owned by several leading mobile phone leaders including Nokia, Ericsson, Sony Ericsson, Panasonic, Siemens and Samsung [23]. The company develops a powerful, open standard multi-tasking operating system for mobile devices called the Symbian operating system, or Symbian OS in short.

The Symbian OS has a core set of API shared between different platforms of Symbian OS phones, making different Symbian OS devices interoperable, and the main platforms are S60 and UIQ. While the native language of the Symbian OS is C++, the Symbian OS has an open application environment and hence applications can also be programmed in other languages such as Python [24], Perl, Visual Basic, Java ME, OPL, Simkin and other mobile device based programming languages.

2.2.2.2 Advantages and Disadvantages

The advantage of the Symbian OS is that it is a widely used operating system and there are many free development tools available for extending the functionality of the smart phones. Symbian S60 has built in location feature and can be used by developers to develop software based on the API [25]. The disadvantage of the Symbian OS is that each application has to be signed [26] and signing of software is complicated and sometimes not free.

2.2.3 BlackBerry

BlackBerry is a device developed by Research In Motion and it is mainly used as integrating business e-mails with server backbends such as the Microsoft Exchange server.

2.2.3.1 Description

The smart phone version for BlackBerry was first introduced in 2002. The BlackBerry has a built in push e-mail functionality that allows instant notification of the arrival of electronic messages [27]. BlackBerry has also got the wireless capability and the operating system that drives the device is called the BlackBerry OS. Like other smart phone operating systems, BlackBerry OS provides multi-tasking capability and allow use to perform mobile phone related functions together with Internet functionalities.

The BlackBerry OS has a set of proprietary programming API called the BlackBerry API to allow third-party developers to write software to extend the functionality of BlackBerry devices. However, since applications usually require access to certain restricted functions of the BlackBerry device, many applications developed by a developer must be digitally signed, and the signing process is restricted to developers that have an account at Research In Motion [28].

2.2.3.2 Advantages and Disadvantages

The advantage of the BlackBerry OS is that it is a very easy to use platform for sending and receiving e-mails. The disadvantage is that the BlackBerry OS device is less advanced in compare with other smart phones, the user base is a lot less in compare to the Symbian OS platform, and also signing of applications is not an easy task for developers to create programs.

OXFORD BROOKES UNIVERSITY

Chapter 3 Analysis of Problem Requirements

3.1 Requirements of the Program

The program is to act as a service tracker, which means that a mobile phone user can ask for a service and the server immediately returns the information on the closest possible service location on where the mobile phone user is currently located. To achieve the goal, the following features and requirements are needed.

3.1.1 Features of the program

The project has at least two components, or in other words, two separate platforms, the client side and the server side. The client side is the mobile side of the application, which allows users to retrieve information from anywhere on the street; and the server side is stationary, waiting for signals passing through from the mobile users and then processes the data and send information back to the user. To simplify the naming convention, the client side will be called as The Mobile Phone Platform, and the server side will be called as The Server Platform.

3.1.1.1 The Mobile Phone Platform

The mobile phone platform has 3 required features and they are as follows:

1. Client Side Location Sender – This feature allows automatic sending of mobile phone user's current location and the requested service that user is interested in

2. Client Side Information Retriever – This feature should be a built in function in the mobile phone. The idea is to allow closest location of the service requested by user displayed on the screen of the user's mobile phone

3. Program Interface – The interface for interacting information between the user and the Server platform, the details of this interface will be discussed in the next subsection

3.1.1.2 The Server Platform

The server platform has 4 required features and they are as follows:

1. Server Side Location Reader – This feature allows the server to read the location information of the mobile phone user, together with the service information details the mobile user is interested, and then transfer the information to a database. This feature has to be connected to the mobile phone.

2. Processor – This feature processes all information in the database that is inserted by the Reader. The Processor has to validate the information, searches for closest location of a service, and then storing meaningful data back to the database outbox.

3. Server Side Information Sender – The sender reads all information stored on the database outbox and connects to the mobile phone for sending information back to the mobile user.

4. Web Interface – The interface allow administrators to add, edit or delete information on all available services that users may query. The details of this interface will be discussed in the next subsection

3.1.2 Interface requirements

The interface requirements list out the particular requirements for the user interface of both the mobile phone and the web interface for service information administrators.

3.1.2.1 The Mobile Phone Interface

The mobile phone interfaces needs to be simple and easy to use, so that the mobile phone user can get the information in a timely manner without the need of thinking how to use the program. The interface itself should contain a list of available service for the mobile phone user to select, and then click the OK button to immediately retrieve information from the server.

3.1.2.1 The Web Interface

The web interface allows administrators to manipulate with the service information, such as the service types, the addresses and the location of each of them. The interface should be flexible and must be able to let users to add, modify and remove information of each service and related details, such as adding new petrol station and new ATM machine of a bank, and relate them to a cell site ID.

3.1.3 Database requirements

The database stores all the data and it is very important for the project. The database must be fast, scalable and has foreign key support and hence it must be a reliable and fully relational database. To reduce the cost of the project, it is also desirable that the database server is free and cross platform portable for easy transfer between different types of systems is also required. In addition, easy administration and ease of use must also be considered.

OXFORD BROOKES UNIVERSITY

3.2 Choice of Technologies

After analyzing the problem requirements, the technologies to be used have been decided. This includes the choice of mobile platform development technology, the web and database technologies, and the choice of server platform technology.

3.2.1 Choice of Location Technology

As discussed in chapter 2, the numbers of GPS and Wi-Fi smart phones are very limited and the costs of acquiring such phones are very expensive in compare with other smart phones.

In addition, while SMS do not have built in location information, SMS function is freely available in all mobile phone hardware. It is possible to retrieve location information from a smart phone without any support required from telecom operators, and by using a self-written program in a smart phone; such information can be transmitted to an external servers to perform more advanced functions. Since security can be specifically handled on the program side, it is not a great concern of the project. Hence SMS, together with a self written program was chosen as the location technology.

3.2.2 Choice of Mobile Platform and Development Technology

As discussed in previous chapter, the most widely used platforms are Symbian OS and Windows Mobile. Since Symbian OS is the more widely used platform in compare with Windows Mobile, and also Symbian OS platform has many free software development tools, Symbian OS platform was chosen as the Mobile Platform.

For the development technology, there are two choices that can be considered, one is java and the other is Python [29]. Since Java has no location based service support in Symbian OS, Python was chosen as the development technology instead.

3.2.3 Choice of Web and Database Technologies

To maximize the possibility of the project and to limit the amount of money spent on the programs, open source technologies are to be used.

According to a netcraft survey [30], Apache web server [31] is the most widely used web server software that has over 50% market share, therefore Apache web server is chosen as the web technology of this project.

For the open source database, there are two well known databases to choose from; the MySQL server [32] and the Postgresql server [33], and both of them have all the features that are needed for the project. According different performance comparisons between the two servers [34][35], the performance of MySQL is faster than Postgresql and hence MySQL was chosen as the relational database for the project.

3.2.4 Choice of Server Platform Development Technology

Besides a web server, server side scripting technology is also needed to execute commands and produces the dynamic outputs to the browser.

There are a few scripting languages available, including Python, Perl and PHP Hypertext Preprocessor, or PHP in short [36]. Since there is a very high install base and usage of a combination of packages called the WAMP, an acronym of Windows, Apache, MySQL and PHP [37], and that the technology is suitable for the project, the PHP language is chosen as the server platform development technology.

3.3 Project Requirements

After deciding on the technologies, the requirements of hardware and software are now required, so that it is possible to achieve the project goals. In this section, all the requirements of the platforms will be listed out.

3.3.1 Mobile Phone Platform

The mobile phone platform that is used by any users, for the scope of this project, the hardware has been specifically chosen and is not portable to another device that is using another Symbian platform.

The hardware and software requirements for the mobile phone platform should be as follows:

Hardware:

Nokia or any other S60 v3 mobile phones

A Nokia file transfer cable (for file transfer to/from mobile and app install)

A personal computer (for file transfer to/from mobile)

Software:

Python for S60

Python Script Shell for S60

Microsoft Windows (for executing Nokia PC Suite for file transfer)

Nokia PC Suite (for file transfer to/from mobile and app install)

3.3.2 Server Platform

The server platform that handles incoming SMS, data manipulation and sending a returning SMS should be as follows:

Hardware 1 (Mobile Phone):

Nokia 6100 or any others that supports GSM 07.05 AT commands [38]

Hardware 2 (The Server):

CPU:	Intel Core 2 Duo or above CPU
RAM:	1GB
Hard disk:	120GB

Software:

OS:	Microsoft Windows Vista
Servers:	Apache 2.2
	MySQL 5.0
	PHP 5.2

Chapter 4 Design and Implementation

This project is to design a feedback system that relies on message passing using SMS as the communication channel. Since SMS is a format that can transmit very limited amount of information, careful design is required to provide both timely and reliable results.

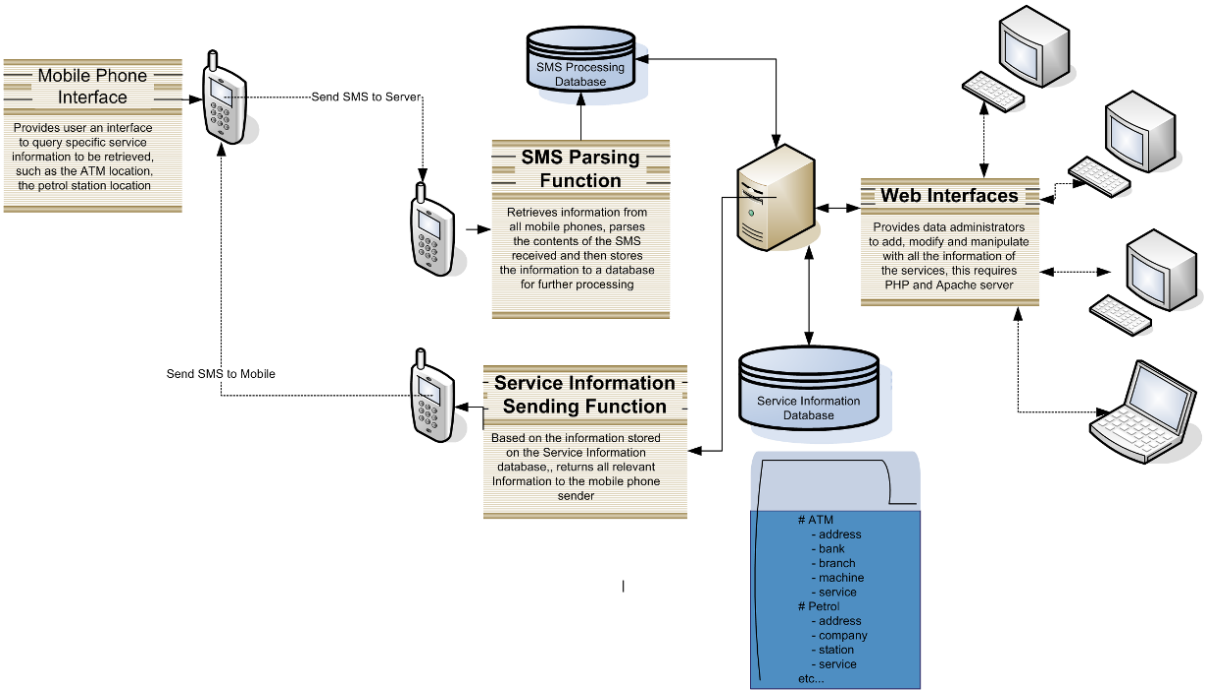
There are two different designs that are required, including the functions design and the database design. The functions design deals with the functionality required by passing information between mobile phone users, their smart phones and the server. Whereas, the database design is the data store to support the message sending, processing, and returning of results of the designed functions. In this chapter the above, together with the implementation details of the project will be discussed.

4.1 Information System Functions Design

To achieve the functionality of a feedback system, some important elements must exist. First, a user must be able to send information from his mobile phone device, and then the mobile phone has to transmit information to the server for retrieving information. Second, he server must be able to retrieve the request information in a timely manner, process the request, and produces a respond using the SMS message channel for the mobile phone user to retrieve. Finally, a function to allow modification of the service information is also needed.

From the above requirements, the design was separated into four information system functions. This includes the Mobile Phone Interface, the SMS Parsing Function, the Service Information Web Interfaces, and the Service Information Sending Function.

The diagram below shows the relationships between the different functions and it visualizes the information system functions design:



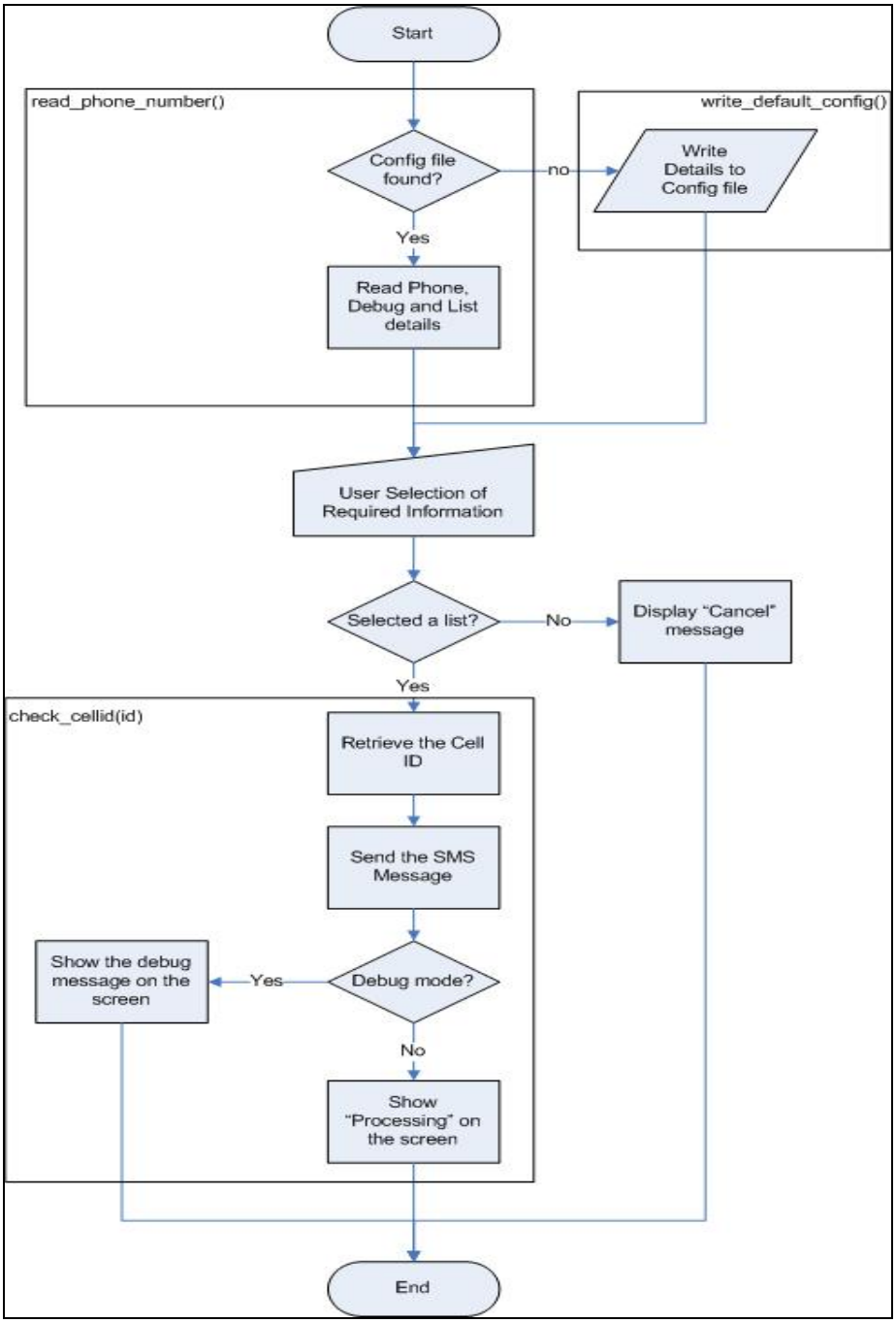
In the following subsections, the design of each function will be discussed in detail.

4.1.1 Mobile Phone Interface

The mobile phone interface is a user end program to allow user to query specific service information to be retrieved, such as the ATM location, the petrol station location, etc.

The program consists of a main program and 3 subroutines. The subroutines deal with reading the configuration files, writing configuration files, and retrieving your location, retrieving your required information and then sending them back to the server.

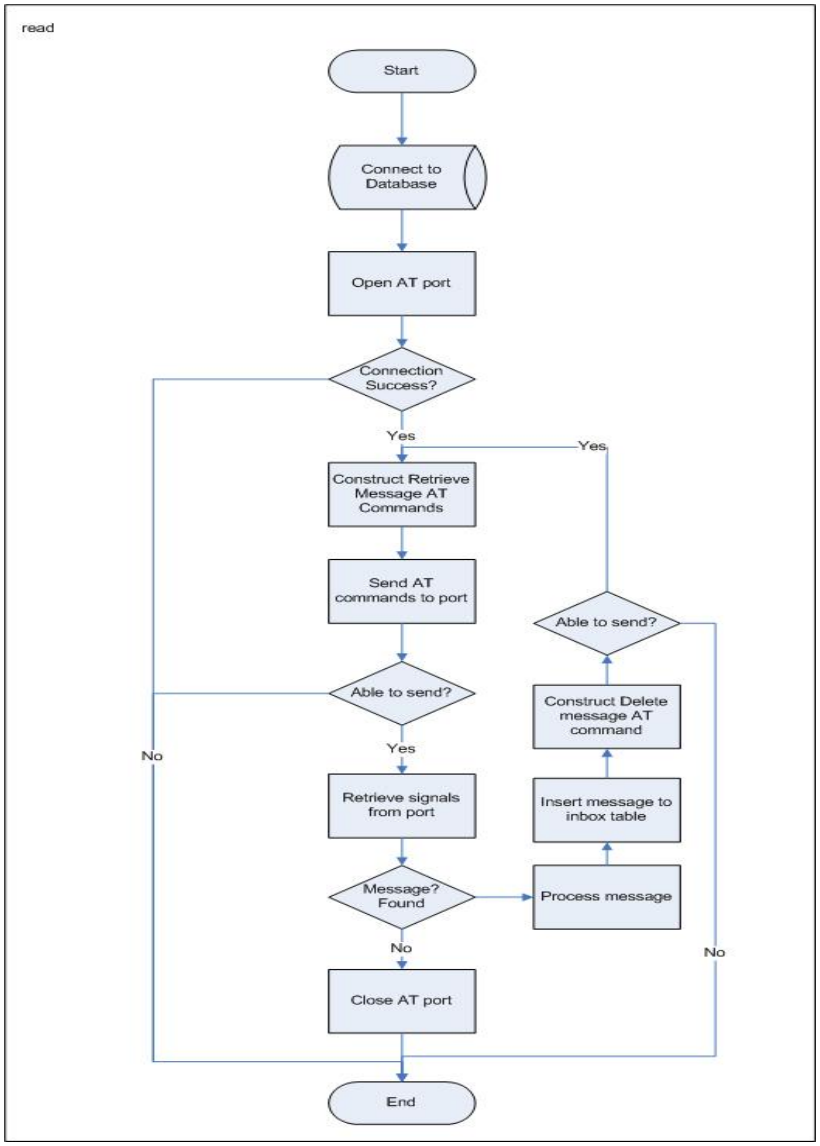
The program flow design of the interface is as follows:



4.1.2 SMS Parsing Function

The SMS Parsing function is a server end program that retrieves information from all mobile phones, parses the contents of the SMS received and then stores the information to a database for further processing.

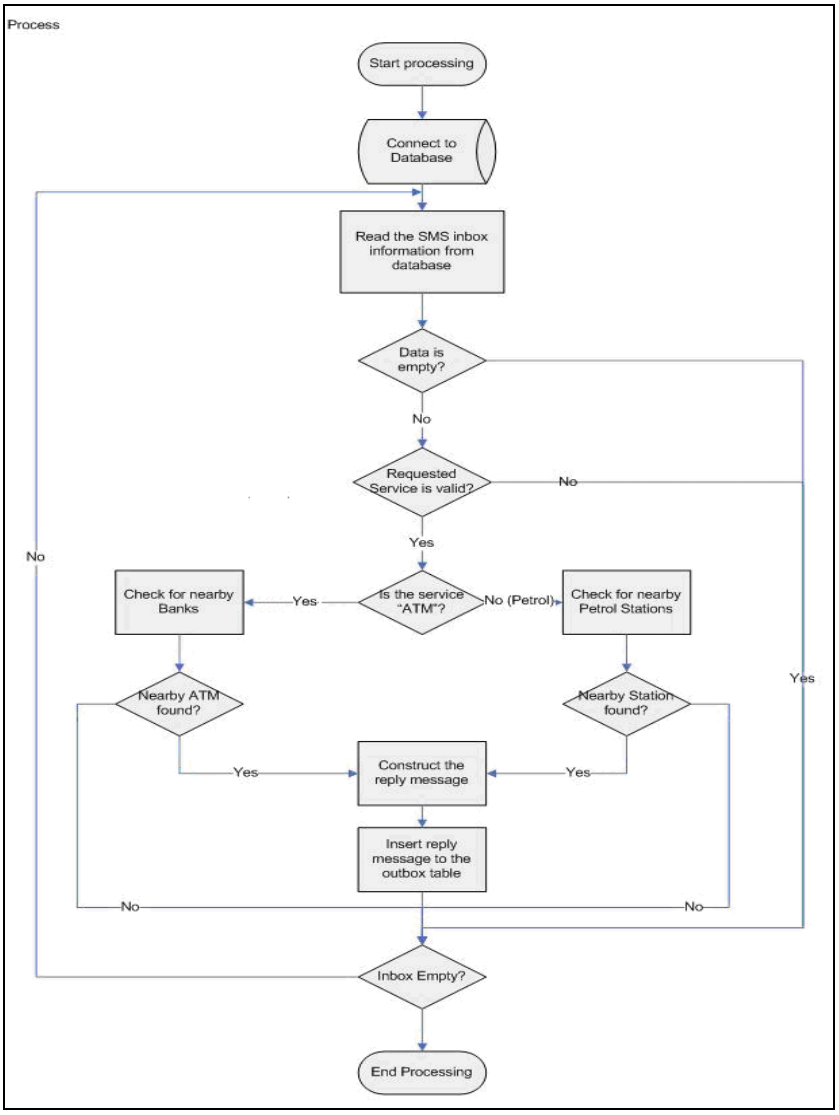
The program flow design of the SMS Parsing function is as follow:



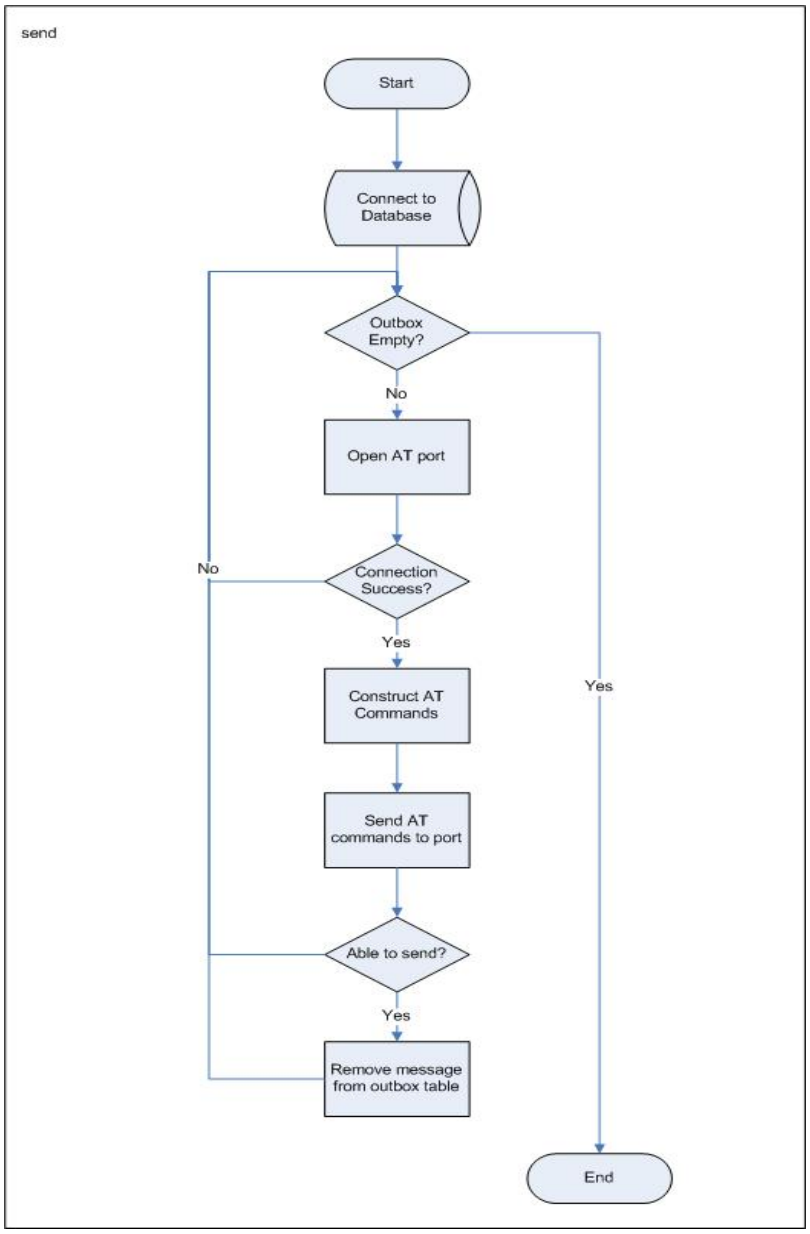
4.1.3 Service Information Sending Function

The service information sending function is also a server end program that retrieves information from the SMS Parsing Function database, and then based on the information stored on the service information database added by user from the service information web interfaces, parses it and returns all relevant information to the mobile phone sender.

The program flow design of the information processing function is as follow:



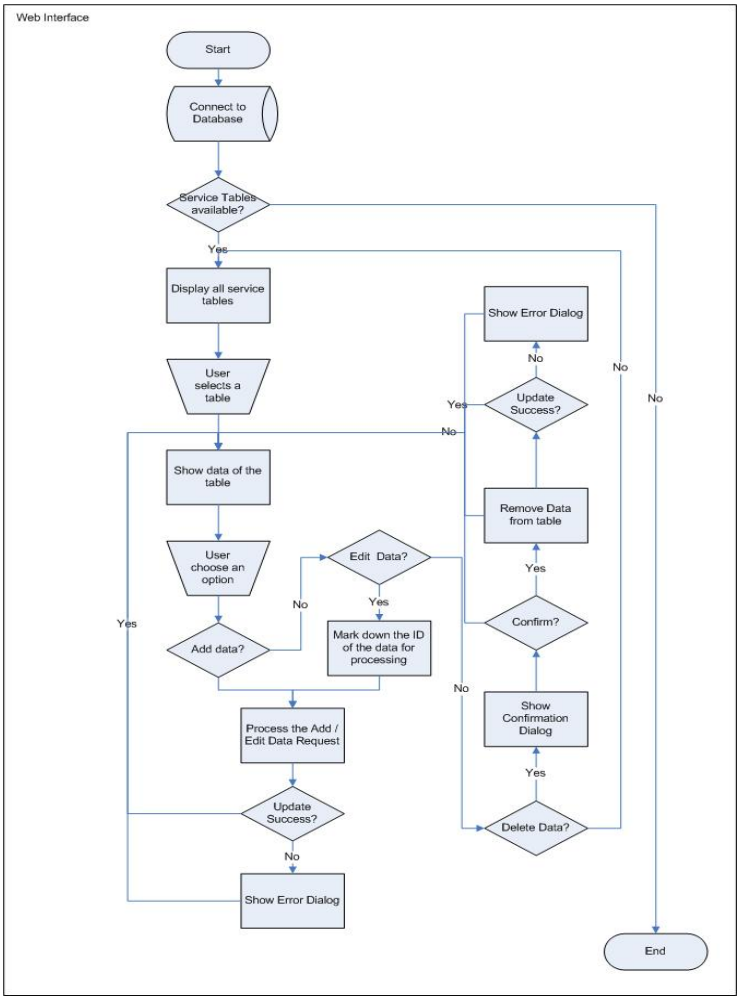
To return returns all relevant information to the mobile phone sender, a SMS sending function is also required and the program flow design of the function is as follow:



4.1.4 Service Information Web Interfaces

The web interfaces provides data administrators to add, modify and manipulate with all the information of the services, this is a server end program, and allow any users connected to the same local area network to access and update information.

The program should be generic and allows addition, edition and deletion of any tables without any changes to the source code. The program flow design of the web interfaces is as follow:

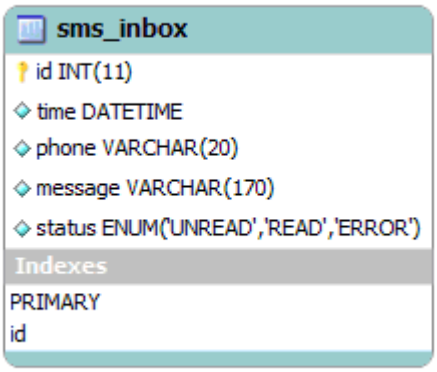


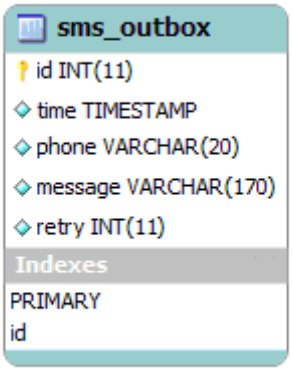
4.2 Database Design

The database design consists of two parts, the SMS handling part and the Service Information part.

4.2.1 SMS Handling

The schema of the SMS database schema is as follows:

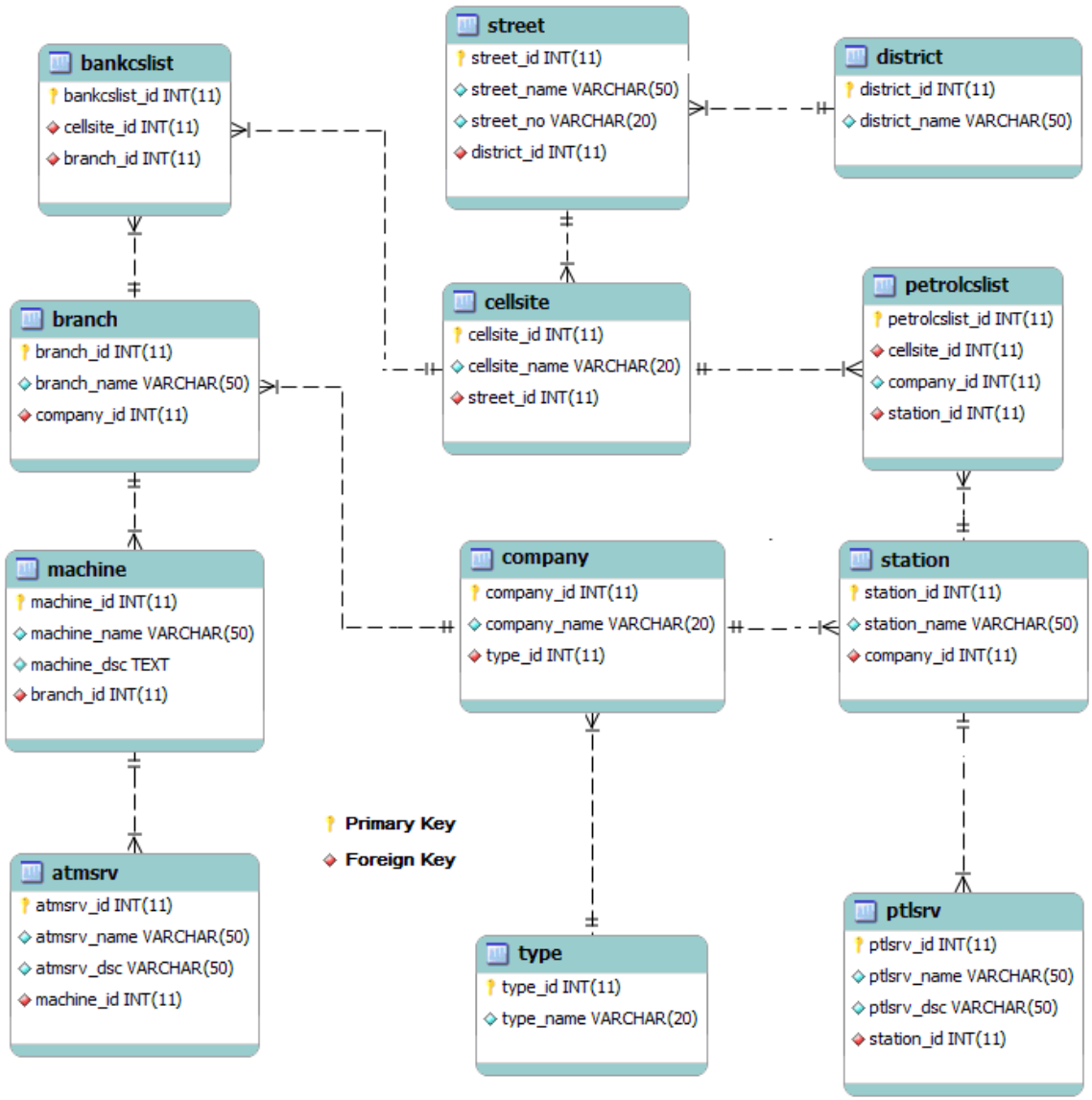




OXFORD BROOKES UNIVERSITY

4.2.2 Service Information

The service information database schema is as follows:



4.3 Implementation

In this section, the implementation of the project deliverables, from the development environment to a brief description on the setting up of system functions and interfaces will be described.

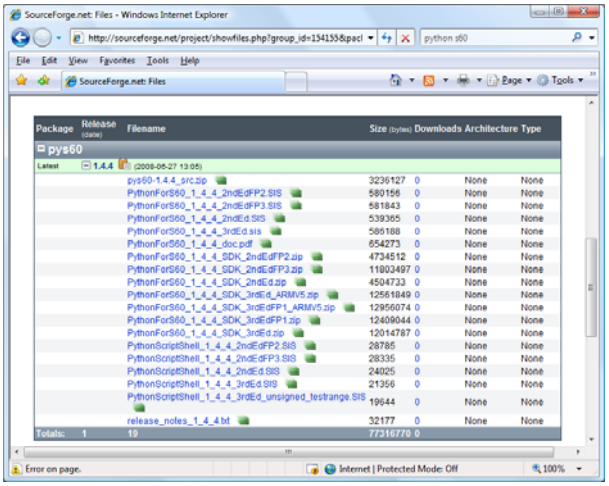
4.3.1 The Mobile Phone Interface

It consists of three different stages to complete the mobile phone interface; they are installing of python software to mobile phone, setting up the development environment, and finally developing and transferring the program to the mobile phone for execution. The details of how the interface is implemented are discussed in the following subsections.

4.3.1.1 Setting up the Mobile Phone

The mobile phone interface is to be developed in Python S60 and requires installing Python S60 to each mobile phone that needs to execute the software.

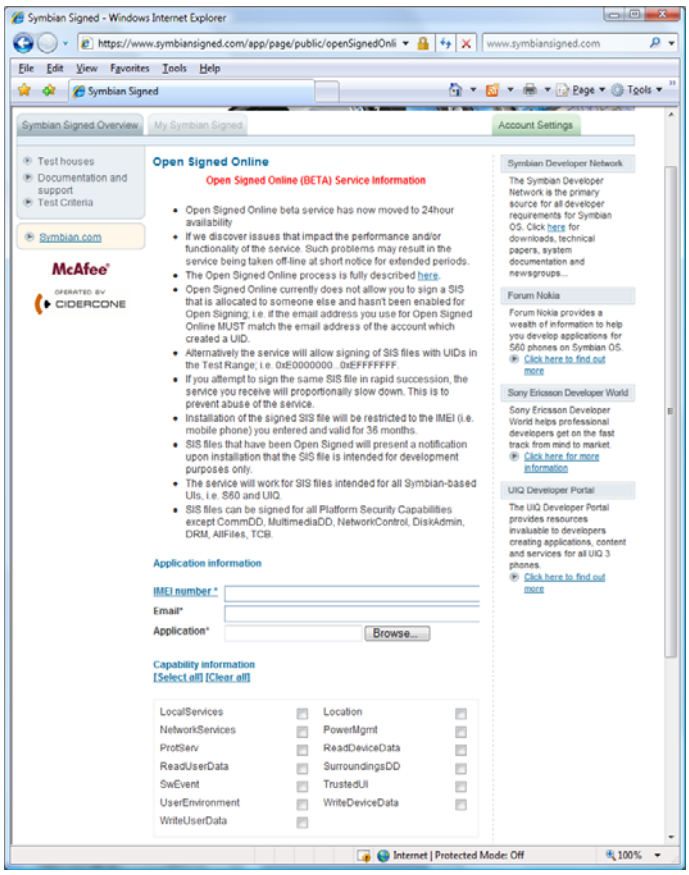
To install the python interpreter, both Python for S60 and Python Script Shell must be downloaded from the http://sourceforge.net/projects/pys60 website.



Since S60 3rd Edition phone are used, and as of writing of this report, the latest version of the pys60 interpreter is 1.4.4, so the following two files were downloaded:

1. PythonForS60_1_4_4_3rdEd.SIS
2. PythonScript_Shell_1_4_4_4rdEd_unsigned_testrange.SIS

The reason for downloading the unsigned Python Script Shell package is to allow me to create Python programs without needing to pay Symbian for this project. However, since all applications in S60 v3 must be signed before able to install, and accessing the http://www.symbiansigned.com page is required to key in the mobile IMEI number of the test phone and sign the Python Script Shell application before able to install it.



Once the application is signed, the test mobile phone can be connected to the computer, then double clicking Python for S60 and Python Script Shell (signed) can install the applications to the test mobile phone.

4.3.1.2 Setting up the Development Environment

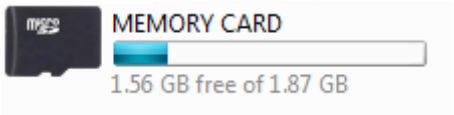
The mobile phone interface is to be developed in Python S60 and requires a text editor to create the python script. The python script itself is text based and the built in Notepad application in the Windows operating system is enough for editing the script, therefore no additional development environment is needed.

4.3.1.3 Creating the Program and Transferring It to the Phone

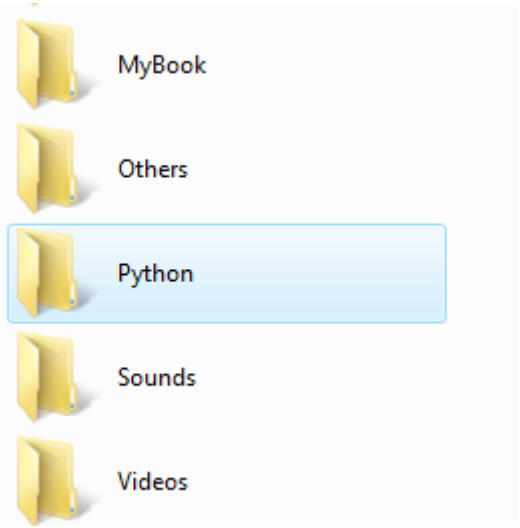
By using notepad, the program source code of the Interface was written, and the pseudo code of the implemented Python program is as follows:

```
main():
      read configurations  from config file
      if configuration does not exists then
            read the defaults
            write default configurations to the config file
      end if
      show the selection list the mobile phone screen
      if user cancelled the selection then
            show the cancel dialog
      else
            check the current location (cellsite id)
            send an sms message to the server
            show the status
      end if
```

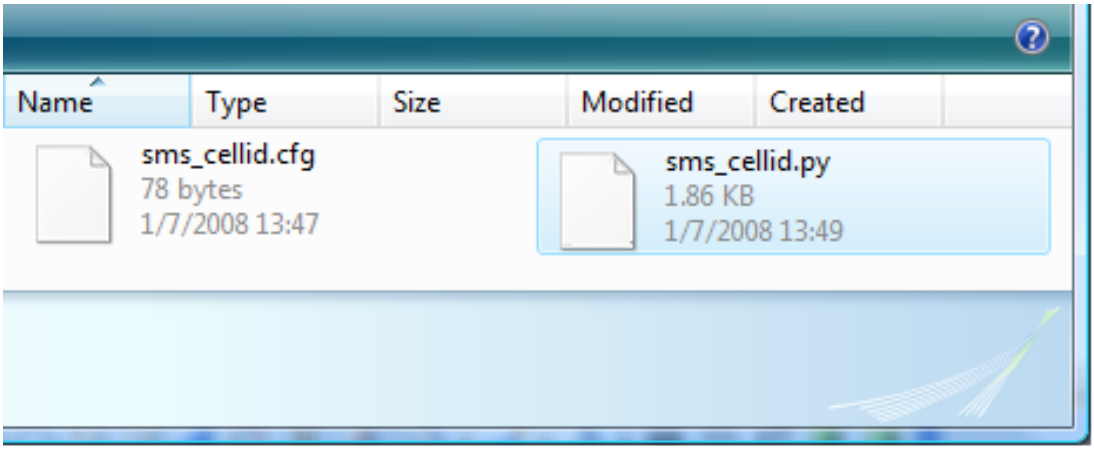
Finally, I have to transfer the source code to the mobile phone. I have to go to My Computer, and then click on my Mobile Phone icon, then double click MEMORY CARD icon similar to the one below:



Then I have to go to the Python folder:



Finally drag and drop the sms_cellid.py program that I wrote to the folder:



Once the above are done, the implementation of the Mobile Phone Interface is complete.

4.3.2 The Server Interface

It consists of three different stages to complete the Server interface; they are installing of all required software to the server, setting up the development environment, and finally developing the program for execution. The details of how the interface is implemented are discussed in the following subsections.

4.3.2.1 Setting up the Servers

The first step for getting the Server running is to install the server software, the Apache Web server, MySQL database server and PHP.

The servers are downloaded from the Apache website (http://httpd.apache.org), the PHP website (http://www.php.net), and the MySQL website (http://www.mysql.com).

The installation location was set to C:\FYP\Servers and hence all server software are located under C:\FYP folder.

4.3.2.2 Setting up the Development Environment

The second step is to setup the database and development environment.

I have chosen to install EMS SQL Manager for MySQL Freeware [39] was chosen for easy creation of database, and as a tool to configuring the MySQL tables.

Once the tables are created, the development environment is setup, since PHP is text based, notepad was chosen as the editor, an application that is built in to Windows, to edit the PHP scripts and hence no extra software is needed.

4.3.2.3 Developing the Programs

There are two server side programs to be developed, the SMS server backend, and also the interface that allows administrator to access from within the local area network. Both of them will be discussed in detail below.

4.3.2.3.1 The SMS Server

The SMS Server consists of three layers, the driver layer, the service layer and the application layer. The driver layer is a phone interface that connects to the mobile phone hardware to query for information, the service layer enables automatic startup of SMS server, and the application layer deals with retrieval, processing and sending of messages.

4.3.2.3.1.1 The Driver Layer

The driver layer was developed for sending and receiving AT commands from the mobile phone that is connected to the server. The pseudo code of the written source codes are as follows:

```
AT Commands driver:
      set AT port mode
      connect to AT port
      while message to be sent is not empty
            send a message to the AT port
            wait
            loop
                  retrieve signals from AT port
            while   signal   does   not   contain   "enter",
                    is   not   ">"   or   not   yet   timeout
      loop
      disconnect from AT port
      return the status or signals from generated by AT
      port
```

4.3.2.3.1.2 The Service Layer

The service layer was developed as a Windows System Service that allows the SMS sending and retrieval to be automatically startup when Windows start. The service layer is specifically written to optionally allow a white list; this is to deal with the possible security hazards that prevent someone from performing SMS spoofing.

The pseudo code of the written source codes are as follows:

Win32 Service:

```
switch Service Action {
	case install service:
		create win32 service with name jeffreysmsd
		if success then
			show successfully install message
		else
			show error message
		end if
		break;
	case remove service:
		remove win32 service with name jeffreysmsd
		if success then
			show successfully remove message
		else
			show error message
		break;
	case start service OR start as command line:
		open database
		retrieve messages from phone
		while messages found and valid
			process messages
			send responses to mobile phone user
		loop
		close database
		break;
	case others:
		shows up error message
}
```

4.3.2.3.1.3 The Application Layer

The application layer was developed for receiving, processing and sending SMS messages. The pseudo code of the written source codes are as follows:

Retrieve Messages:

```
open at port
if connection is successful then
        retrieve REC UNREAD sms messages
        while messages exists then
                parse the retrieved signal into ID, Phone, Date and
Subject
                insert the information to the database
                delete the message from phone
         loop
        close at port
        return success
else
        return failure
end if
```

Process Message:

```
read unprocessed messages from inbox table
while message found
        if the phone number is on the whitelist or the system has no
        whitelist at all then
                parse the information for correctness
                if correct then
                        check for nearby service
                        if nearby service found then
                                construct the reply message
                                mark message status as READ
                                insert the reply message to outbox
                        end if
                else
                        mark message status as ERROR to avoid processing
the
                        message again
                end if
        end if
loop
```

Send Messages:

```
while outbox contains unsent messages
        open AT port
        if connection is successful then
                send sms message to AT port
                if message sending succeeded then
                        delete the message from outbox
                else
                        update the message retry count
                end if
         loop
        close at port
loop
```

4.3.2.3.2 The Interface

The web interface was developed for administrator to add, edit, delete or viewing the information of available services. The pseudo code of the written source codes are as follows:

Main Interface:

```
show header
open database
if tables exists
        list out all tables
end if
close database
show footer
```

View/Edit/Add Interfaces:

```
show header
open database
if table entry exists
        show the details for viewing/editing/adding an entry
end if
close database
show footer
```

Add/Edit Processing Interfaces:

```
show header
open database
if table entry exists or addable
        check for errors
        if no errors found then
                insert / update table entrie
        end if
        show the status
        return to the previous page
end if
close database
show footer
```

Delete Interface:

```
show header
open database
if table entry exists then
        check for errors
        if no errors found then
                if user confirmed the warning prompt then
                        remove table entry
                        show status
                        return to previous page
                else
                        show warning prompt
                end if
        end if
end if
close database
show footer
```

4.3.3 Location Information Retrieval

Since there is no information on where the cell sites are, manually locating the cell site IDs and also going through different directions to determine the boundary of the cell sites is needed. In this section, the debug mode will be used to retrieve the information required.

4.3.3.1 Switching on the Debug mode

I have added a debug mode in the mobile phone Python program, and the debug mode will print the cell site information on the screen instead of sending information to the server.

To activate the debug mode, I have to use Nokia PC Suite [40] to retrieve the sms_cellid.cfg file from e:\Python folder, which is the same folder where I have uploaded the python script. After retrieving the file, I have to modify the debug parameter to 1 and from then on, cell site information will be displayed on smart phone screen instead.

4.3.3.3 Retrieving the Cell Site IDs

After the debug mode is activated, I traveled around and go to the different locations of ATM and Petrol station services I am interested and then retrieve the cell site IDs. The difficulties of the cell site ID retrieval are that I have no knowledge of the actual location of the cell site and the boundary of the cell sites, so that I have to go around every corner of the services and then experiment if multiple cell sites are available in the area. The process has to repeat for 5 or more times to obtain an accurate and reliable result. After marking down the location on paper, the location details are then entered in the interface as discussed in next chapter.

OXFORD BROOKES UNIVERSITY

Chapter 5 Results and Discussions

5.1 Testing Methodology

After the completion of the source code, the resulting deliverables has to be tested. The tests are based on the repeatability and check if the service requests are successfully performed.

During the implementation phase, most of the location details are already obtained and the tests are actually based on the repeatability of finding the information of a service at a particular location. The tests were performed for at least 5 times, if none of the tests deliver unpredictable result, the test is considered passed. The way to perform the tests and the test results are listed in the next sub-section.

5.2 Testing Results

There are three parts that needs to be tested, the server backend, the phone interface and the web interface. I have both marked down the test results that are listed below, and also performed a waiting time statistics and the performance results will be discussed in the next sub section.

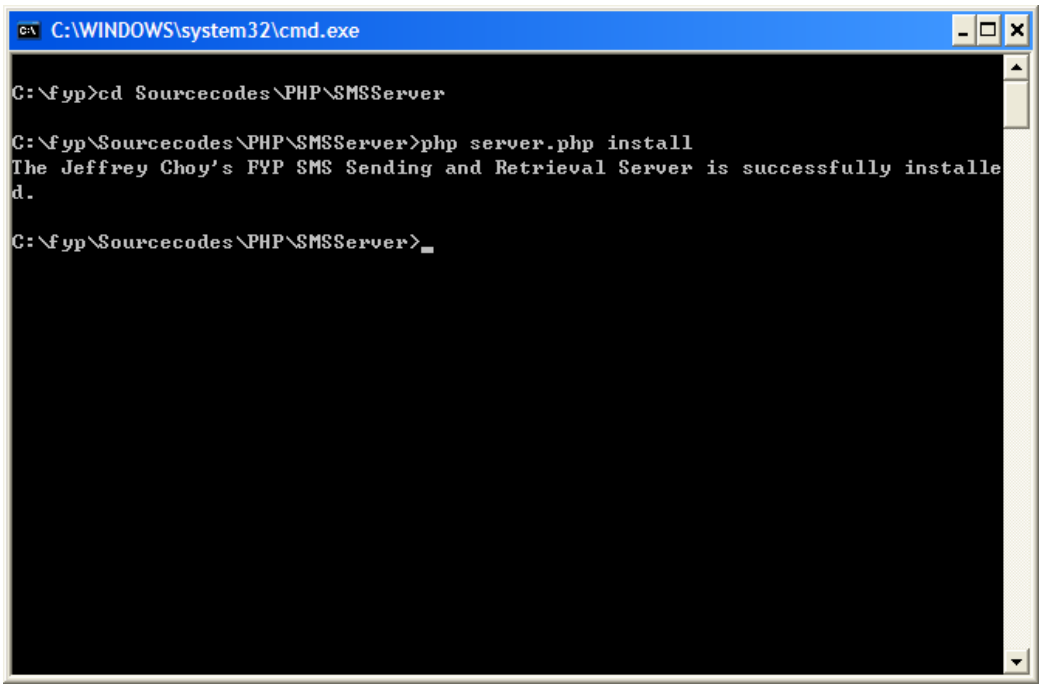
5.2.1 The Server Backend

The first deliverable is the SMS Server backend, which controls the retrieving, processing and returning of information retrieved from the SMS of a mobile phone user.

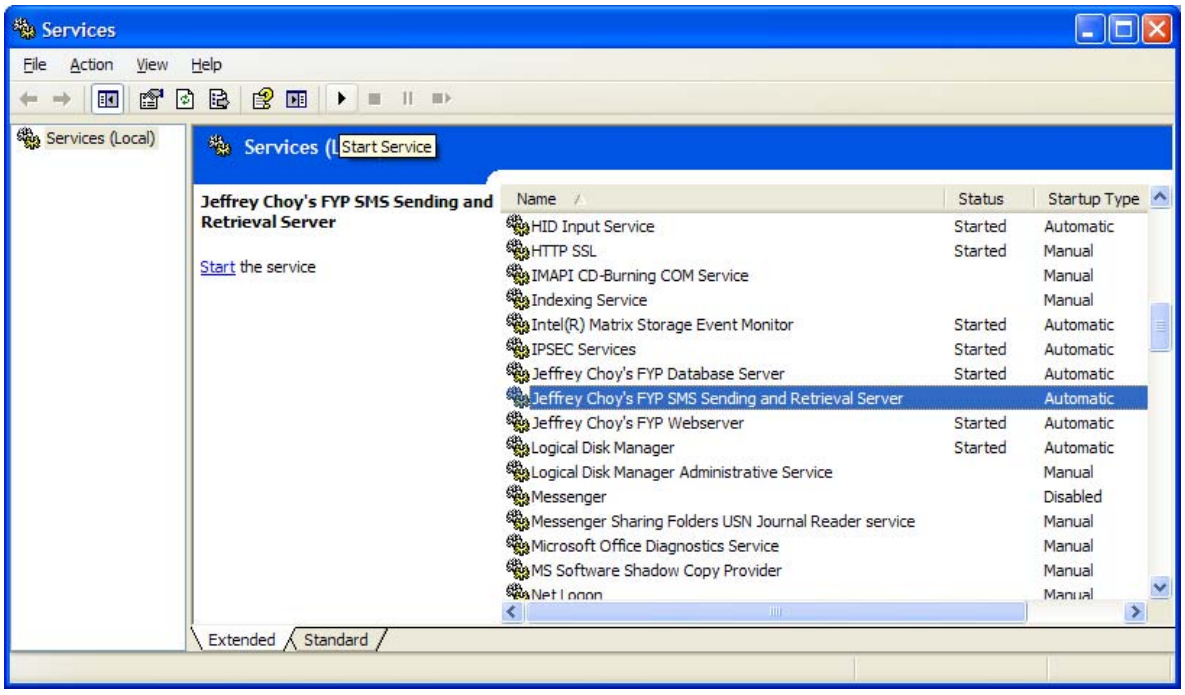
There are two ways to activate the SMS Server Backend, either by installing the backend as service or running it from the command line. The results of running both ways are successful and repeatable without any issue at all.

5.2.1.1 Running Server Backend as Service

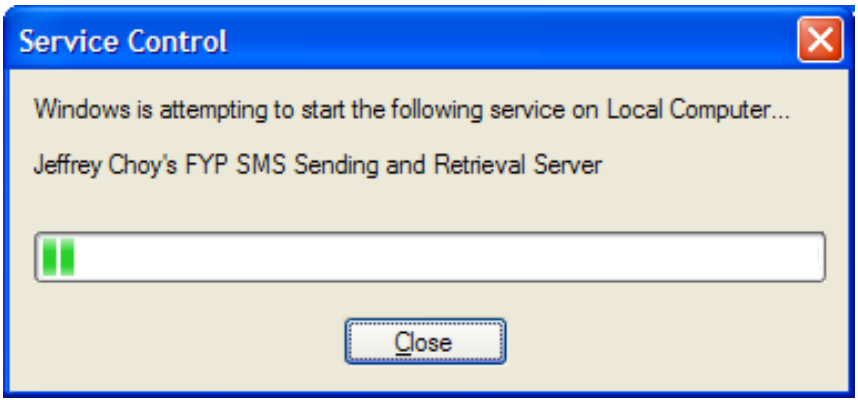
The way to test running the backend as service, I have started up the command line and type in “php server.php install” as indicated below:



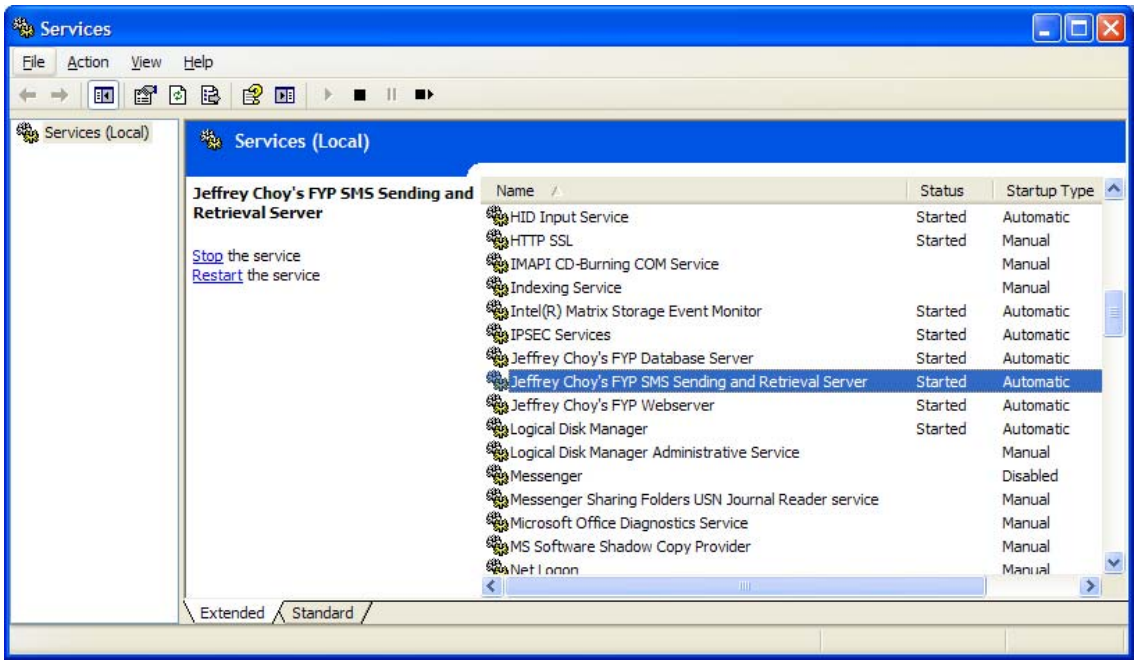
After the service is installed successfully, I have to go to the Control Panel and double click Services. I then find the “Jeffrey Choy’s FYP Sending and Retrieval Server” from the list of services and then select it:



Then I have to click on the Start button to start up the service:



The Status is then indicated as “Started”:

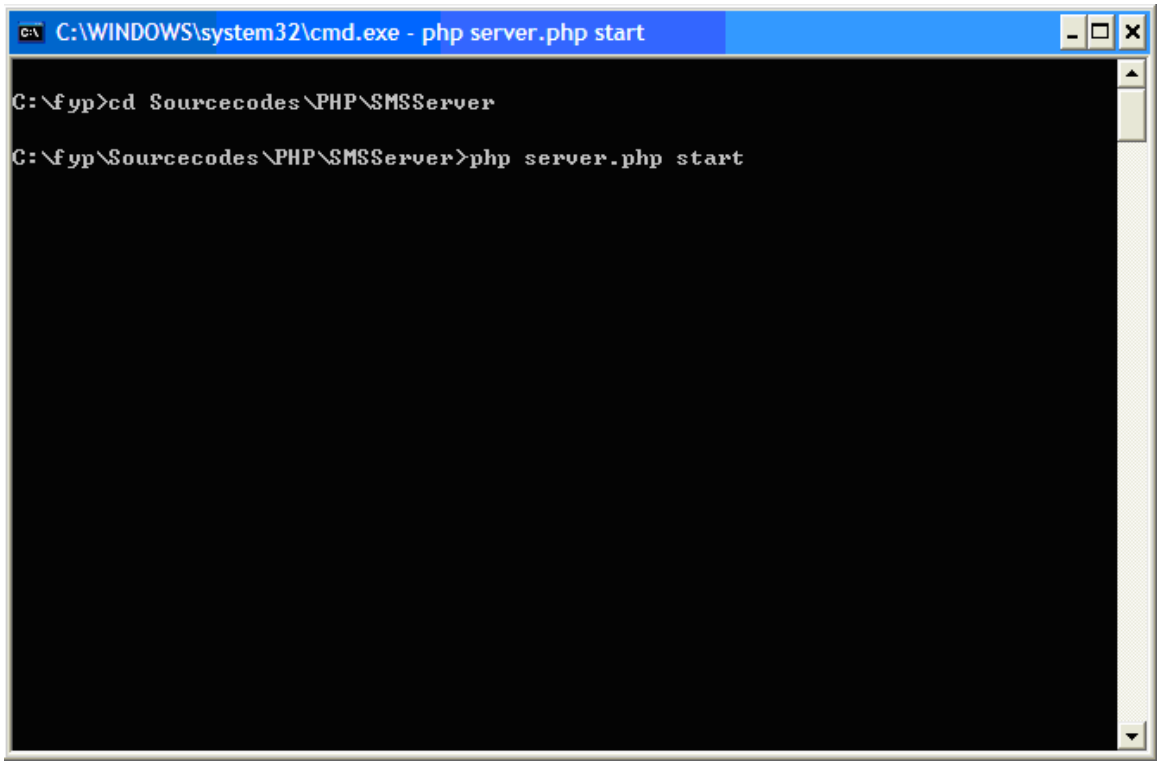


After numerious times of starting and stopping the server, or running “php server.php remove” to remove and then restart the whole process in this section, no issues were found.

5.2.1.2 Running Server Backend from Command Line

In order to avoid issues with the Windows service, such as the Vista User Account Control (UAC) protection [41], the server backend was designed to be able to run as a standalone program as well. I have tried to start the server in the standalone mode quite a few times, and no issues were found even running it from the Microsoft Windows Vista operating system.

A screenshot of running the SMS Server backend in the standalone mode is indicated as follows:

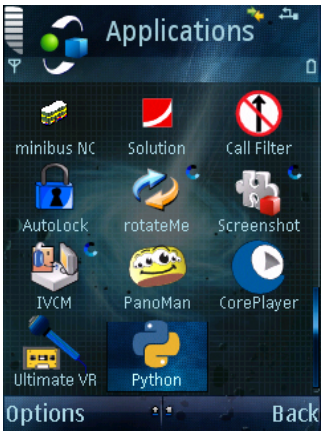


5.2.2 The Phone Interface

The phone interface is the main control of the system and most of the interaction is performed on users' mobile phone. With numerous testing of the interface, no issues were found.

5.2.2.1 Start Python

The first step is to start the Python Interpreter that can execute the source code written by me. The way to start up the interpreter is first go to Applications, then open Python as indicated below:



Once the Python interpreter is started, a copyright message appears, together with the information of the version of the Python interpreter:

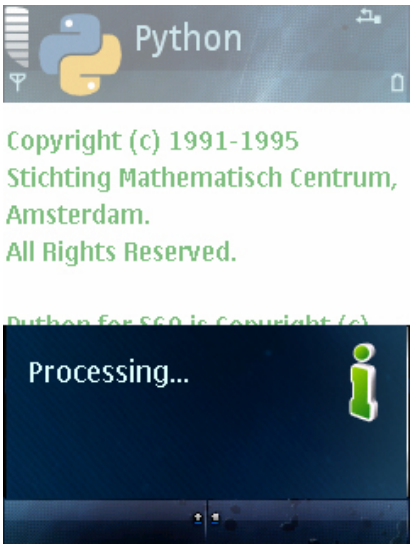
5.2.2.2 Executing the Python Script

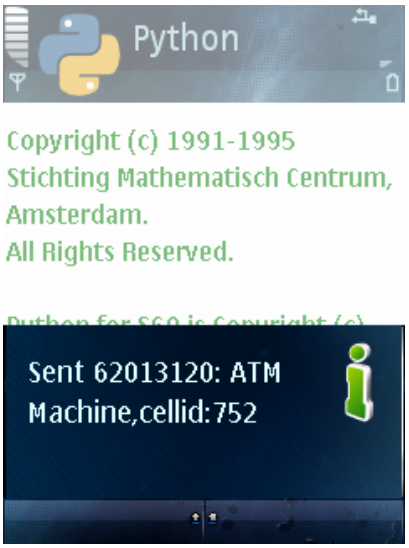
The second step is to execute the Python Script that is written specifically for handling the Service Tracking. To do so, I select options, and then choose the "Run script" command from within the Python interpreter:

Then I have to select "e:sms_cellid.py", the Service Tracking Interface that was written by me:

After that, I have to choose an option, at the moment, only 2 selections are available, that is either “ATM Machine” or “Petrol Station”:

Once the option is chosen, the Python program will automatically retrieve user’s current location, and then send the information to the Server. If debug mode is on, then the information that was sent to the server will be displayed, otherwise, “Processing…” will be displayed:





After a few seconds, the information regarding the closes service information will be transferred back from the server, and I am able to click the Show button to view the details of the information, such as the branch name, floor and street of the service I need:

5.2.2.3 Test Results

With the tests I have performed so far, none of the information was lost and I can receive an SMS message every time I execute the program with the phone on the white list, and for those without a service nearby, I received a no service available nearly notification. When I use a mobile that is not on the white list, as expected, I receive no response from the system.

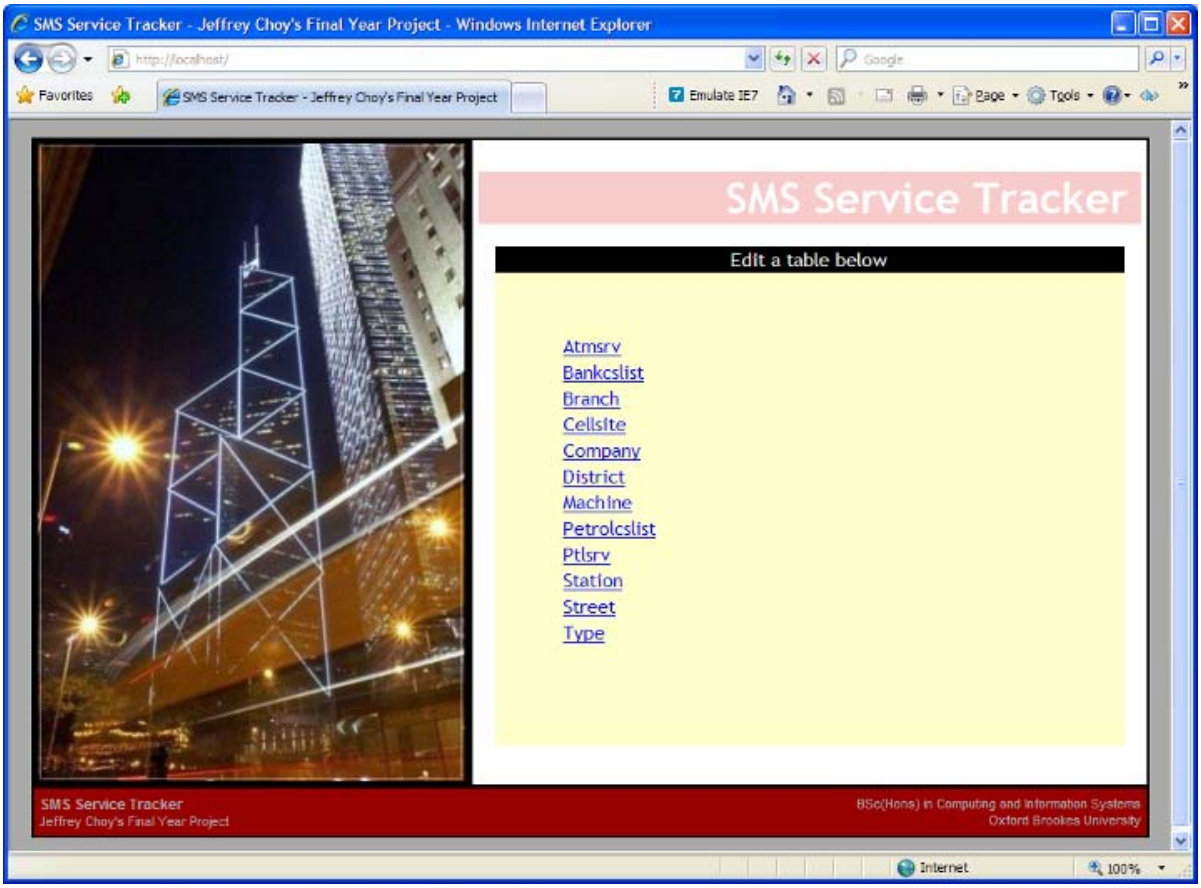
I have performed the above tests for various times with different locations, different mobile phones and at different times. Since all tests were completed without unexpected results, the tests were successfully.

5.2.3 The Web Interface

The web interface allows administrators to edit information to be sent back to the mobile phone users, and inaccurate information will cause trouble with the system. The web interface was designed generically and can deal with additional services if it is needed in the future. There are a couple of interfaces that needs to be tested and they are discussed in the subsections below.

5.2.3.1 The Index

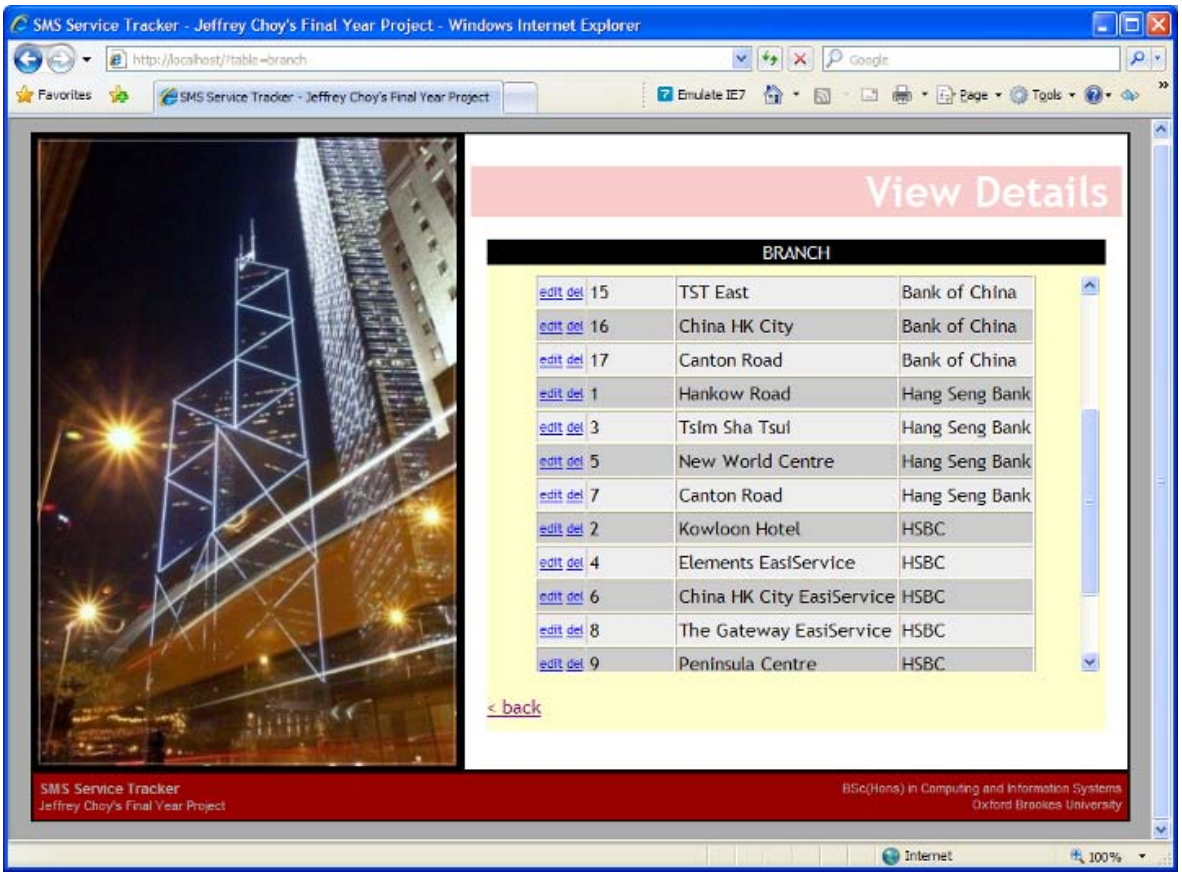
The index page automatically loads all the tables information to the screen and allow admin users to manipulate the tables.



Since the details of the automatically generated page are always as expected, the results were successful.

5.2.3.2 View Details Screen

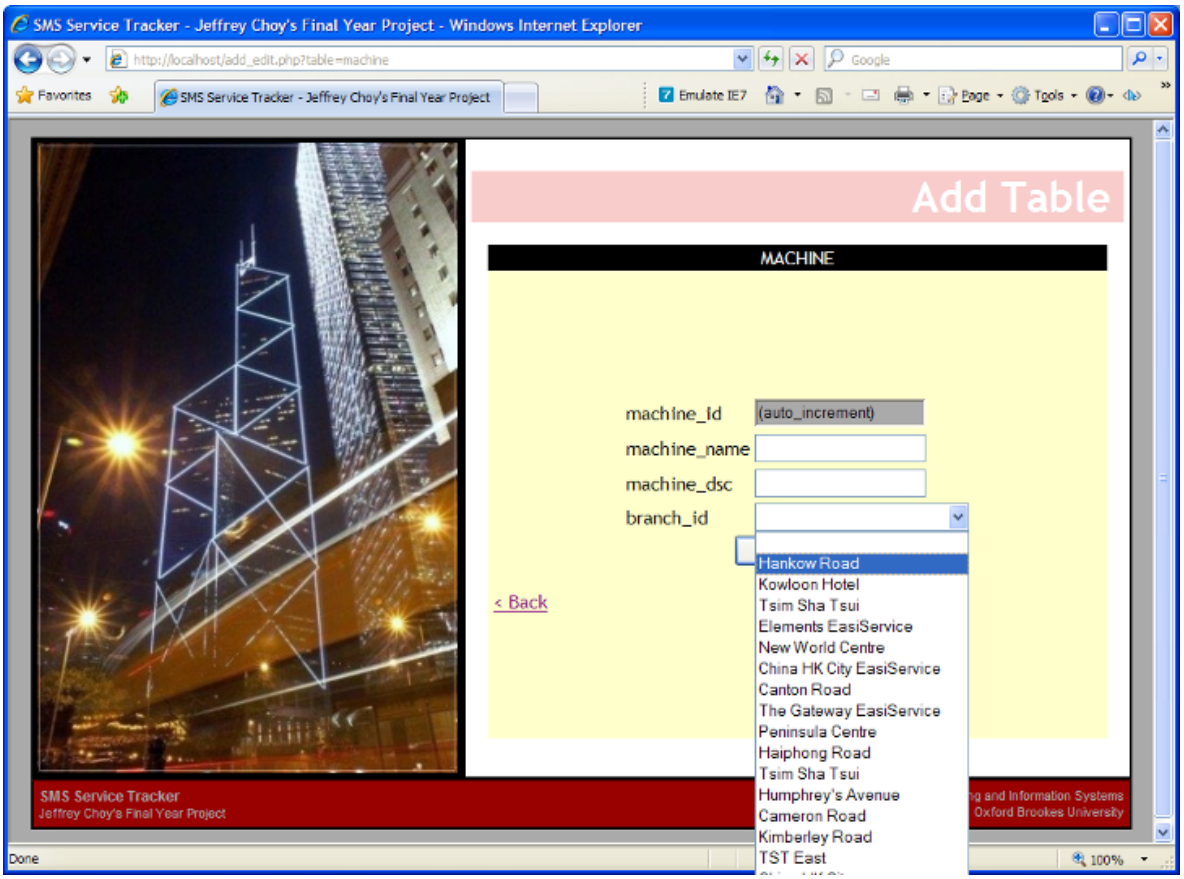
The view details screen provides the details of the selected table and show up relevant information if there are foreign keys in the table.



By manually verifying the details with the database, no issues were found in the tests performed.

5.2.3.3 Add Records

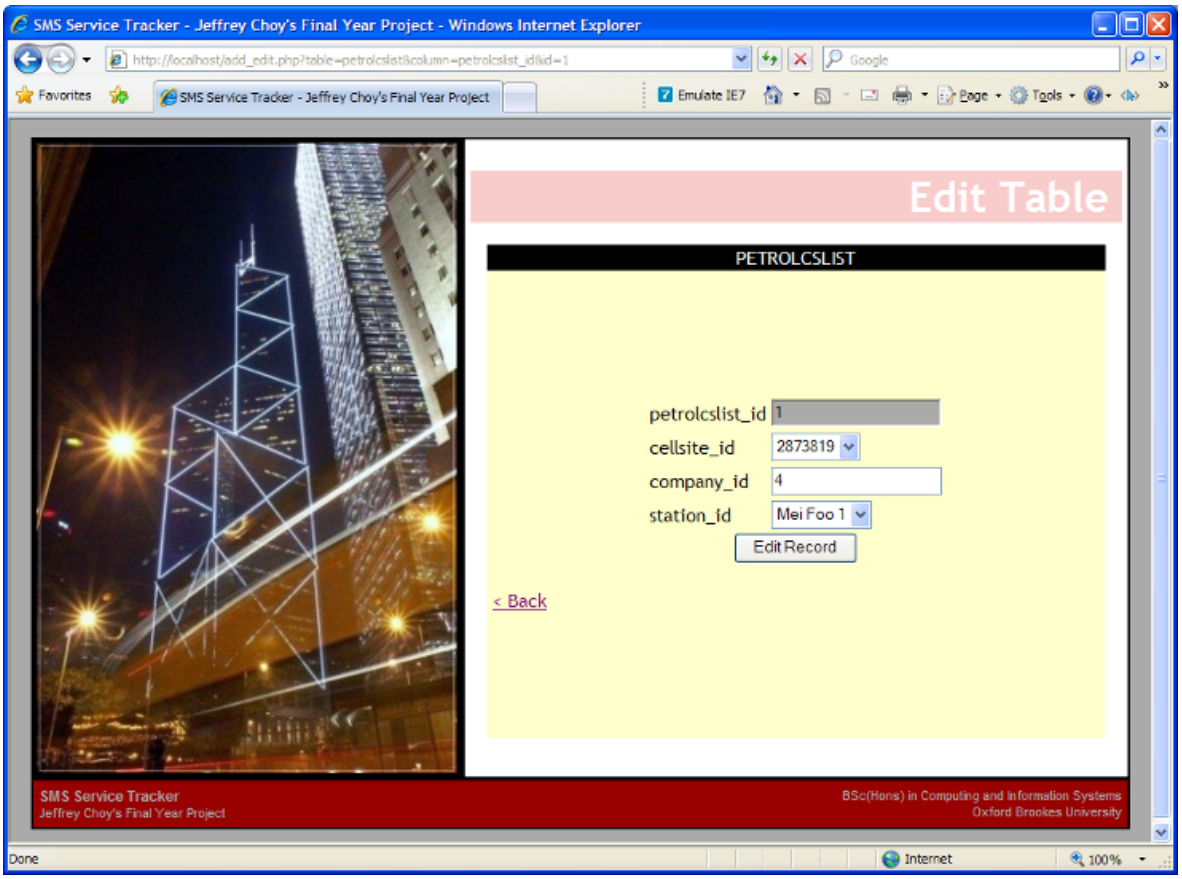
Add records screen provides administrators a way to add a new data to the selected table.



All information are entered through the add records screen and no problems were detected so far.

5.2.3.4 Edit Records

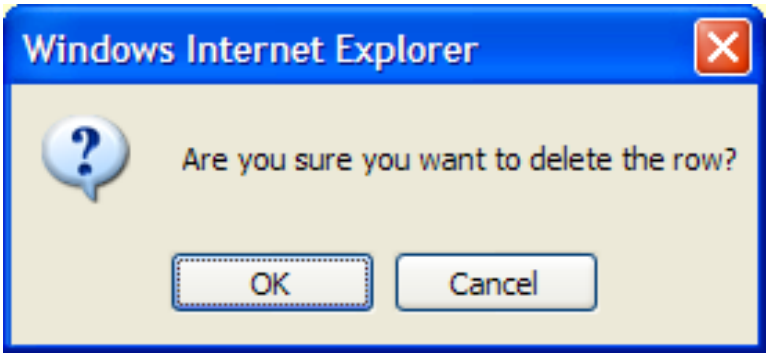
Edit records screen provides administrators a way to edit existing data of the selected table.



Of all the records edited, none of them produces error or strange results, so the tests were successful.

5.2.3.5 Delete Records

Delete records screen provides administrators a way to delete existing data from the selected table.



By verifying the details with the database each time, none of the records were found inconsistent, so the results were successful.

OXFORD BROOKES UNIVERSITY

5.3 Discussions

The project was completed successfully as all the results were as predicted and no problems were found. In this section, I will present the achievements, problems encountered and possible future improvement works that can further improve the project.

5.3.1 Discussion on Achievements

Below is the summary of the achievements of this project:

- **User interact able service location tracker**

  The system allows mobile phone users to track for the location of a nearby service, by using the service, useful information will be sent back to the user, so user no longer needs to manually or call up a friend to search for information

- **Easy to use mobile interface**

  The mobile phone interface was developed in Python and by using a simply interface, mobile phone users can perform information request within a few clicks.

- **Good quality of results**

  The system provides a web interface for administrators to modify the details of services, such as the address, service details, district etc., so the information is up to date and the quality of the results are always of high level

- **Timely processing of requests**

  The system has a SMS Server backend to read, process, and send information around the clock. Unlike some other programs that uses a scheduler to perform the processing in a set schedule, the process starts nearly immediately after a user request is received, and hence the result is produced nearly instantly

OXFORD BROOKES UNIVERSITY

5.3.2 Discussion on Performance and Problems Encountered

Apart from the achievements, the system performance is also a main issue to the project. In this section, I am going to discuss both the system performance and the problems encountered.

5.3.2.1 System Performance

The waiting time between mobile phone sending information to the server, then from the server sending back information to the mobile phone is on average 20 seconds or less. After many tests performed on different days, including weekends and weekdays, the results were never over half a minute, it is possible to say that the probability of congestion of the SMS network is very low. By comparing with starting up a Nokia N95 GPS service, which requires nearly 2 minutes to start up to track for satellites, the speed of the program is a lot faster and therefore in an acceptable range.

5.3.2.2 Problems Encountered

There were a few problems encountered during the development of the system and the details are listed below.

5.3.2.2.1 Problem with Gammu

In the beginning of the development, I have initially decided to use a tool called Gammu [42] to perform SMS sending and receiving. However, the Gammu tool is not compatible with all of the mobile phones I have tried, from Motorola mobile phones, Sony Ericsson mobile phones, to Nokia mobile. Because of that, I have finally decided to write an AT command library from stretch to perform the SMS sending and receiving task. It is not an easy task to write a system level driver using the PHP programming language, especially when sending SMS is not free, and it is not very easy to debug. In the end, the program was developed successfully without any major or minor issues.

5.3.2.2.1 Problem with Python Script Shell

In the beginning of writing the Python Mobile Phone application, I have encountered a serious problem with installing the Python Script Shell for S60 device. The reason is because Symbian has changed the application signing process and no longer allow applications to be signed if the application does not have an ID in the test range. Since Python Script Shell developer certificate version was not in the test range, I was unable to install the program on the mobile phone. Luckily, a new version came out 2 months after and I was able to continue with the project without problem.

5.3.2.2.2 Problem with Vista

The system was first developed under Windows XP and the SMS Server is created as a service for easy startup of the SMS sending and receiving service. However, Windows Vista has a completely different architecture, together with the User Access Control, makes it impossible to install the SMS Server as a service on the laptop I have got. I have spent quite a bit of time to fix the problem, and the final decision is to incorporate a new command line switch to allow execution of the SMS Server from the command line instead of running as a service.

5.3.3 Possible Improvements and Future Work

While the results are acceptable, some improvements can be made to the system to further enhance the functionality of the system, and they are listed below.

1. Platform Independent – at the moment, the system can only run from S60 v3 mobile devices, if it is possible to work on other devices such as the Windows Mobile platform, then it will greatly increase the possible user base of the system

2. Operator Independent – at the moment, there is no way to determine the service provider, and it is assumed that the mobile phone user and the server are all using the same service provider, with same set of cell site ID. If the program can determine the operator and then determine the cell site information based on the operator, users will no longer be forced to stick with a single mobile phone operator

3. Multiple Response – at the moment, there is no way to send more than one closest service information, if the feature is added, then mobile phone users will have a choice to choose from more than 1 location to go to.

4. Separating the SMS Servers – at the moment, SMS sending and receiving are using the same mobile phone, if separating the processes into two mobile phones, the response time will be better and the service can be used by more people at the same time

5. Security – at the moment, the web interface does not have any security built in, this is dangerous and should be handled to avoid guess modifying the database without permission; also, the SMS messages should be encrypted to avoid spoofing attacks

6. Advertisements – by adding a text based advertisements to the system, the service can be sustainable and profitable, so that the service can grow even the user base has greatly increased

OXFORD BROOKES UNIVERSITY

Chapter 6 Conclusion

In this project, I have created a system to allow smart phone users to track for a service of the person's interest. By implementing the project, I have simplified the way of how people can find out the information of a service, so that a person can instantly get the required information by using his or her smart phone with a few clicks.

I have started off studying various location based technologies and smart phone platforms. Furthermore, I have studied a number of mobile and server platform development technologies, and also web and database technologies that are useful to my project.

I then implemented the mobile phone interaction application to allow easy sending of cell site ID and wanted service details. After that, I have created the location parsing and replying service, and then a web application for handling the location details.

Due to time limitation, some of the problems were not considered and they can be part of the future work, these include platform and operator dependencies, multiple location response, different SMS sending and receiving queues for the server, security issues and advertisements.

Nevertheless, by repeatedly tracking for services with the implemented system, I have found that the SMS Service Tracker is very reliable and produces result quickly, and if the tools are widely deployed, it will greatly enhance the daily life of every user and calling up of friends or phone company to ask for a service location will become a history.

OXFORD BROOKES UNIVERSITY

References and Citation

1. Dow Jones Newswires, “UN Agency: World Now Has 4 Billion Phone Subscribers”, Cellular-News, Sep 2007
   URL: http://www.cellular-news.com/story/25833.php
   Accessed: 4 Jul 2008

2. AFP, “Up to 90 Percent of Globe to Have Mobile Coverage”, BNET, Oct 2006
   URL: http://findarticles.com/p/articles/mi_kmafp/is_200610/ai_n16932713
   Accessed: 4 Jul 2008

3. Timo Kopomaa, “SPEAKING MOBILE: THE CITY IN YOUR POCKET”
   URL: http://www.tkk.fi/Yksikot/YTK/julkaisu/mobile.html
   Accessed: 1 May 2008

4. Long M., “Longitude and Latitude: location technologies and privacy concerns”, 29th International Conference of Data Protection and Privacy Commissioners, Terra Incognita, Workbook Series #4, Sep 2007
   URL:
   http://www.privacyconference2007.gc.ca/workbooks/terra_incognita_workbook4_e.html
   Accessed: 3 Jul 2008

5. Nakamura K., “The GPS Resource Library”
   URL: http://gpsy.com/gpsinfo/
   Accessed: 2 Jul 2008

6. Garmin, “What is GPS?”
URL: http://www8.garmin.com/aboutGPS/
Accessed: 2 Jul 2008

7. People Web, “Beijing Toilets Accurate Positioning, Find a Toilet by Calling Up with a Mobile Phone”
URL: http://bj.people.com.cn/GB/7431296.html
Accessed: 1 July2008

8. Pang CH, “BizTech Management – KMB IT Management, Innovative Technology Operation Model”, e-Zone Issue 56, PC Section, pp168-171, Jul 2008

9. Wireless LAN Association, “The Learning Zone for Wireless Networking”,
URL: http://www.wlana.org/learn/educate1.htm
Accessed: 3 Jul 2008

10. WMI, “Global Positioning System”
URL: http://www.wmi.org/www/boating/gps.htm
Accessed: 4 Jul 2008

11. Brown J., Shipman B., Vetter R, “SMS: The Short Message Service”, IEEE Computer Magazine, Dec 2007
URL:
http://www.computer.org/portal/site/computer/menuitem.5d61c1d591162e4b0ef1bd108bcd45f3/index.jsp?&pName=computer_level1_article&TheCat=1055&path=computer/homepage/Dec07&file=howthings.xml&xsl=article.xsl
Accessed: 4 Jul 2008

12. Michael R., “Table of ASCII Characters”,
URL: http://web.cs.mun.ca/~michael/c/ascii-table.html
Accessed: 4 Jul 2008

13. Krishnamurthy N., “Using SMS to Deliver Location-based Services,” Personal Wireless Communications, 2002 IEEE International Conference, pp. 177-181, Dec 2002

14. GSM Association, “Permanent Reference Document SE23 - Location Based Services”, GSM World, pp. 9-12, Jan 2003
URL: http://www.gsmworld.com/documents/lbs/se23.pdf
Accessed: 3 Jul 2008

15. Coulton P, Edwards R, Clemson H., “S60 Programming: A Tutorial Guide”, Wiley, pp. 2-18, Apr 2007
URL: http://media.wiley.com/product_data/excerpt/57/04700276/0470027657.pdf
Accessed: 3 Jul 2008

16. StarHub, “StarHub launches Singapore’s first nation-wide location-based mobile advertising service”
URL:
http://www.starhub.com/portal/site/StarHub/menuitem.876159666306d8a8aa494b608324a5a0/?vgnextoid=99301840a0e3c010VgnVCM10000038425a0aRCRD&vgnextfmt=ItemCMId%3Adb8800e525e79110VgnVCM100000464114acRCRD
Accessed: 1 Jul 2008

17. Microsoft, “Pocket PC: Redefining the Rules for Mobile Computing”, Apr 2000
URL: http://www.microsoft.com/presspass/features/2000/04-19pocketpc.mspx
Accessed: 1 Jul 2008

18. Hilton, K., “Overview of the PocketPC”, Staffordshire University, Mar 2007
URL: http://gawain.soc.staffs.ac.uk/modules/levelm/ce00213-m/ms/mas_ppc_intro.ppt
Accessed: 2 Jul 2008

19. Microsoft Developer Network, “Overview of the Windows API”
URL: http://msdn.microsoft.com/en-us/library/aa383723(VS.85).aspx
Accessed: 2 Jul 2008

20. Microsoft, “Developing Native Applications for Microsoft Windows Mobile 6”
URL: http://www.microsoft.com/learning/en/us/syllabi/6067afinal.mspx
Accessed: 1 Jul 2008

21. Goon, D., “Develop for Windows Mobile 5.0 with Visual Studio 2005”
URL:
http://www.microsoft.com/emea/msdn/betaexperience/nlarchive/issue_3/develop_for_windows_mobile_5_with_visual_studio_2005.aspx
Accessed: 2 Jul 2008

22. Canalys, “Smart mobile device shipments hit 118 million in 2007, up 53% on 2006”
URL: http://www.canalys.com/pr/2008/r2008021.htm
Accessed: 1 Jul 2008

23. Symbian, “Ownership”
URL: http://www.symbian.com/about/overview/ownership/ownership.html
Accessed: 2 Jul 2008

24. Nokia Resource Center, “Python for S60”
URL: http://opensource.nokia.com/projects/pythonfors60/
Accessed: 1 May 2008

25. Nokia, “S60 Location Technologies”
URL:
http://www.forum.nokia.com/main/resources/technologies/location_based_services/S60_location_technologies.html
Accessed: 1 May 2008

26. Symbian Signed, “Announcement: Open Signed Online vs Developer Certificate Q&A”
URL: http://developer.symbian.com/forum/ann.jspa?annID=45
Accessed: 1 Jul 2008

27. BlackBerry, “An Introduction to BlackBerry”
URL: http://na.blackberry.com/eng/developers/started/introduction.jsp
Accessed: 2 Jul 2008

28. BlackBerry, “Controlled API and Code Signing”
URL: http://www.blackberry.com/developers/downloads/jde/api.shtml
Accessed: 2 Jul 2008

29. Nokia, “The Python programming language for S60 mobile phones”
URL: http://sourceforge.net/projects/pys60
Accessed: 1 Jun 2008

30. Netcraft, “August 2007 Web Server Survey “
URL:
http://news.netcraft.com/archives/2007/08/06/august_2007_web_server_survey.html
Accessed: 1 Jul 2008

31. The Apache Software Foundation, “About the Apache HTTP Server Project”
URL: http://httpd.apache.org/ABOUT_APACHE.html
Accessed: 1 Jul 2008

32. MySQL AB, “Why MySQL?”
URL: http://www.mysql.com/why-mysql/
Accessed: 1 Jul 2008

33. PostgreSQL, “About”
URL: http://www.postgresql.org/about/
Accessed: 1 Jul 2008

34. PHP, “PHP: Introduction”
URL: http://hk.php.net/manual/en/introduction.php
Accessed: 1 Jul 2008

35. 2bits, “Benchmarking PostgreSQL vs. MySQL performance using Drupal 5.x”
URL: http://2bits.com/articles/benchmarking-postgresql-vs-mysql-performance-using-drupal-5x.html
Accessed: 1 Jul 2008

36. Windows Skills, “PostgreSQL vs MySQL benchmark”
URL: http://wskills.blogspot.com/2007/01/postgresql-vs-mysql-benchmark.html
Accessed: 1 Jul 2008

37. Maddox, J., “The Art of Web Design”
URL: http://www.courses.psu.edu/art/art203_jxm22/php.html
Accessed: 1 Jul 2008

38. ETSI, “Digital cellular telecommunications system (Phase 2+); Use of Data Terminal Equipment - Data Cuipment (DTE – DCE) Interface for Short Message Service (SMS) and Cell Broadcast Service (CBS) (GSM 07.05 Version 5.5.0)”, GSM Association, pp. 35-54, Jan 1998
URL: http://www.ctiforum.com/standard/standard/etsi/0705.pdf
Accessed: 1 May 2008

39. Electronic Microsystems, “EMS SQL Manager for MySQL Overview”
URL: http://www.sqlmanager.net/products/mysql/manager
Accessed: 1 Jul 2008

40. Nokia Europe, “Nokia PC Suite”
http://www.nokia.com/pcsuite
Accessed: 3 Jul 2008

41. Johansson J., “The Long-Term Impact of User Account Control”, Microsoft Technet, Sep 2007
URL: http://technet.microsoft.com/en-us/magazine/cc137811(TechNet.10).aspx
Accessed: 3 Jul 2008

42. Cihar, M., “Gammu All Mobiles Management Utilities”
URL: http://cihar.com/gammu
Accessed: 1 Jun 2008

43. Göth, C., Frohberg D., Schwabe, G. The Focus Problem in Mobile Learning. Proc. of the 4th IEEE Int. Workshop on Wireless, Mobile and Ubiquitous Technologies in Education, 2006, Athens, Greece.

44. TREVISANI, E. AND VITALETTI, A. 2004. Cell-ID location technique, limits and benefits: An experimental study. Università di Roma "La Sapienza." Dec. 2004.

OXFORD BROOKES UNIVERSITY

Appendix

A. Program Source Codes Listing

Below are the source codes of all the programs written for the project, including the client side programs written in Python and also the server side programs written in PHP.

A.1 Client Side Program – Python Client

A.1.1 sms_cellid.py – The Client Side Interface

```
import location
import appuifw
import messaging
import os
import sys

# define defaults
default_phone_number = "62013120"
default_list = [u'ATM Machine', u'Petrol Station']
default_cellid = 0

# define variables
config_dir = "e:/python/"
config_file = "sms_cellid.cfg"
list = default_list
cellid = default_cellid
phone = default_phone_number
debug = 0

# check settings
def write_default_config():
   if not os.path.isdir(config_dir):
     os.makedirs(config_dir)
   config=os.path.join(config_dir,config_file)
   settings={}
   settings['phone'] = default_phone_number
   settings['debug'] = debug
   settings['list'] = list


   f=open(config,'wt')
   f.write(repr(settings))
```

```
   f.close()

def read_phone_number():
   global phone, debug, list
   if not os.path.isdir(config_dir):
     os.makedirs(config_dir)
   config=os.path.join(config_dir,config_file)
   try:
     f=open(config,'rt')
     try:
       content = f.read()
       config = eval(content)
       f.close()
       phone = config.get('phone',default_phone_number)
       debug = config.get('debug',0)
       list  = config.get('list', default_list)
     except:
       phone = default_phone_number
       debug = 0
       write_default_config()
   except:
     phone = default_phone_number
     debug = 0
     write_default_config()

# check cell id
def check_cellid(id):
    global phone, debug

    (mcc, mnc, lac, cellid) = location.gsm_location()
    txt = list[id]+",cellid:"+str(cellid)
    if debug == 1:
      print(txt)
    else:
      messaging.sms_send(phone,txt)

    if debug == 1:
      appuifw.note(u"Sent "+str(phone)+": "+txt, "info")
    else:
      appuifw.note(u"Processing...", "info")



# main script
read_phone_number()
result =  appuifw.selection_list(choices=list , search_field=1)
```

```
if result==None:

  appuifw.note(u"Cancelled")

else:

  check_cellid(result)
```

A.2 Server End Program – SMSServer (PHP)

A.2.1 server.php – The Main Driver

```
<?php
  /*
    Jeffrey Choy's FYP

    The SMS Server:
      - the actual server and settings to be executed by Windows during start time
  */

  function __autoload($class) {
    // automatically load required class files when needed
    require_once str_replace("\\","/",dirname(__FILE__))."/classes/$class.php";
  }

  $class=new SMSServer;  // create an instance of smsServer
  $class->setDBConnectionString("localhost","root","","fyp");
  $class->setSvcParam('-c                                    c:\fyp\servers\php5.2\php.ini
c:\fyp\sourcecodes\php\smsserver\server.php');
  $class->setAtPort("COM7:");
  $class->setAcceptedPhoneNums(array("+85298441144","+85294700474"));
  $class->setLogType(LOG_FILE);
  $class->setLogFile('c:\fyp\logs\smsserver_at.log');
  $class->setArgv($argv);
  $class->startSvc();   // start the service after the settings are set
?>
```

A.2.2 classes/ATCommands.php – The AT Commands Controller

```
<?php
  /*
    Jeffrey Choy's FYP

    The AT Commands class:
      - the implementation of open, close and executing at commands
  */
  abstract class ATCommands extends DB_MySQL {

    private $port;
    private $fp;
    const LOG_NONE=0;
    const LOG_FILE = 16;
    const LOG_SCREEN=255;

    private $log_type=LOG_SCREEN;
    private $log_filename;
    private $shown_error=0;

    abstract protected function read();
    abstract protected function send();
```

```
    private function log($message) {
      date_default_timezone_set("Asia/Hong_Kong");
      $message=str_replace("\r","",$message)."\r\n";
      if ($this->log_type==LOG_SCREEN) {
        echo "(".date("H:i:s").") ".$message;
      } elseif ($this->log_type==LOG_FILE && $this->log_filename!="") {
        $error=0;
        $handle=@fopen($this->log_filename,'a+');
        if (!$handle && $this->shown_error==0) {
          $error=1;
        } else {
          $status=fwrite($handle,"[".date("d/m/Y:H:i:sO")."] (ATCommands) ".$message);
          if (!$status)
            $error=1;
          else
            $this->shown_error=0;
          fclose($handle);
        }
        if ($error==1 && $this->shown_error==0) {
          echo "(".date("H:i:s").") Failed to write log file `$this->log_filename`.\r\n";
          $this->shown_error=1;
        }
      }
    }

    public function setLogFile($filename) {
      $this->log_filename=$filename;
    }

    public function setLogType($type) {
      if ($type==LOG_DB || $type==LOG_FILE || $type==LOG_SCREEN || $type==LOG_NONE)
        $this->log_type=$type;
    }

    public function setAtPort($port) {
      if  (strlen($port)>4  &&  substr($port,0,3)=="COM"  &&  substr($port,strlen($port)-
1,1)==":")
        $this->port=$port;
      else {
        $this->log("Unable to set port to $this->port");


      die();
      }
    }

    public function getAtPort() {
      return $this->port;
    }

    // Common method
    protected function openAtPort() {
      exec("mode $this->port BAUD=115200 PARITY=N data=8 stop=1 XON=ON RTS=ON DTR=ON");
      $this->fp=@fopen($this->port,"w+b");
      if ($this->fp===FALSE) {
          $this->log("Failed to open AT port $this->port.");
          return false; // error opening port
      } else {
          $this->log("Successfully opened AT port $this->port.");
          return true;
      }
    }

     function sendToAtPort($cmds) {
       // declare variables for specific phone models, do not change!
       $loops_before_timeout=35;
       $at_receive_default_wait_time=1;
       $at_receive_smscenter_wait_time=1;

       // set defaults
```

```
       $error=0; $count=0; $status=0;
       $line=array();

       // Method for fopen
       for ($i=0;$i<count($cmds);$i++) {
        $message="";
        $buffer="";

        fwrite($this->fp,$cmds[$i]);
        $this->log("Successfully sent to AT port $this->port `".$cmds[$i]."`"); // debug
handling

        $loop=0;
        while ($loop<=$loops_before_timeout) {
          $loop++;
          if ($status!=2) {
            sleep($at_receive_default_wait_time);
          } else {
            $status=3;
            $this->log("Waiting  for  AT  port  $this->port  $at_receive_default_wait_time
seconds"); // debug handling

            sleep($at_receive_smscenter_wait_time);
          }

          while (false !==($char=fgetc($this->fp))) {
            if ($char=="\n") { // ignore enter
              //break;
            } elseif ($char!="\r") {
              $buffer.=$char;
              if ($buffer=="> ") {
                $this->log("Received  message  from  AT  port  $this->port  at  loop  $loop:
`".$buffer."`");
                $status=2;
                $buffer="";
                break 2;
              }


            } else {
              if ($buffer!="")
                $this->log("Received  message  from  AT  port  $this->port  at  loop  $loop:
`".$buffer."`");

              $loop=1; // received message, reset loop counter
              if  (trim($buffer)=="ERROR"  ||  substr(trim($buffer),0,11)=="+CMS  ERROR:")
{ // error occurred
                 $status=0;
                 $loop=0;
                 break 2;
              } else if (trim($buffer)=="OK") {
                 $status=1;
                 $loop=0;
                 break 2;
              } else {
                 if             ($buffer!=str_replace("\r","",$cmds[$i])             &&
$buffer!=str_replace("\r\n","",$cmds[$i]) && trim($buffer)!="")
                   $message.=$buffer."\r\n";
                 $buffer="";
              }
            }
          }
        }

        if ($loop>=($loops_before_timeout+1)) { // temporarily set to loop 4, was 6
           $this->log("Timeout waiting AT port $this->port: giving up after trying $loop
times for message=`".$buffer."`"); // debug handling
           @fwrite($this->fp,"\rAT\r"); // clean up
           sleep(1);
           while (false !==($char=fgetc($this->fp))) {} // clear buffer
```

```
           return false; // buffer overflow
         } else if ($status>=2) {
           // expects;
         } else if ($status==1) {
           if (trim($message)!="") {
             $line[$count]=$message;
             $count++;
           }
         } else {
           $error=1;
         }
       }

       if ($error==1)
         return false;
       else
         return $line;
     }

     protected function closeAtPort() {
       @fclose($this->fp);
       $this->log("Closed AT port: $this->port.");
     }
   }
?>
```

A.2.3 classes/DB.php – The Database Abstract Class

```
<?php
  /*
    Jeffrey Choy's FYP

    The DB class:
      - the abstract class to specify handling of open, close and executing database
queries
      - allows external setting and viewing of the connection strings
  */

  abstract class DB {
    protected $db_host,$db_user,$db_pass,$db_name;
    private
$default_host="localhost",$default_user="root",$default_pwd="",$default_db="smsdb";

    abstract public function openDB();
    abstract public function closeDB();
    abstract public function execSQL($sql);
    abstract public function queryDB($sql);

    public function setDBConnectionString($host,$user="",$pass="",$name="") {
      if ($host!="")
        $this->db_host=$host;
      if ($user!="")
        $this->db_user=$user;
      if ($pass!="")
        $this->db_pass=$pass;
      if ($name!="")
        $this->db_name=$name;
    }
    public function getDBConnectionString() {
      return                array(host=>$this->db_host,user=>$this->db_user,pwd=>$this-
>db_pass,db=>$this->db_name);
    }
    public function setDBConnectionStringDefault() {
      $this->setDBConnectionString($this->default_host,$this->default_user,$this-
>default_pwd,$this->default_db);
    }
  }
?>
```

A.2.4 classes/DB_MySQL.php – The MySQL Database Driver

```
<?php
  /*
    Jeffrey Choy's FYP

    The DB subclass:
      - the implementation of open, close and executing database instructions
  */
  abstract class DB_MySQL extends DB {
    private $db;

    public function openDB() {
      try {
        $this->db=new  PDO('mysql:host='.$this->db_host.';dbname='.$this->db_name,$this-
>db_user,$this->db_pass);
      } catch (PDOException $e) {
        die('Unable to connect to db');
      }
    }

    public function closeDB() {
      $this->db=null;
    }

    public function execSQL($sql) {
      $count=$this->db->exec($sql);
      return $count;
    }

    public function queryDB($sql) {
      $i=0;
      $query=$this->db->query($sql);
      $result=array();
      foreach($query as $row) {
        $result[$i]=$row;
        $i++;
      }
      $query=null;
      if ($i>0)
        return $result;
      else
        return false;
    }

  }
?>
```

A.2.5 classes/SMSParser.php – The SMS Parser class

```
<?php
  /*
    Jeffrey Choy's FYP

    The smsParser

  */

  class SMSParser extends DB_MySQL {
    public function parse($msg) {

      $result=split(',cellid:',$msg);

      if ($result==$msg || !is_array($result)) return false;
      $service=$result[0];
      $cellid=$result[1];
//      $cellid=2688947; // temporary
```

```
      $this->openDB();
      $query="SELECT `type_id` FROM `type` WHERE `type_name`='$service'";
      $result=$this->queryDB($query);
      $type_id=0;
      $return=false;
      if ($result) {
        $type_id=$result[0]['type_id'];

        if ($type_id==1) { // ATM
          $notfound="Sorry, unable to find any banks nearby!";
          $query="SELECT  `cellsite_id`,  `cellsite_name`,  `street_no`,  `street_name`,
(SELECT  `district_name`  FROM  `district`  WHERE  `district_id`=`street`.`district_id`)  AS
`district`   FROM   `cellsite`   LEFT   JOIN   `street`   ON   `cellsite`.`street_id`   =
`street`.`street_id` WHERE `cellsite`.`cellsite_id`=$cellid";
          $cellsite=$this->queryDB($query);
          // can check if there are more cellsite id with the same street name also
          if ($cellsite) {
             $query="SELECT `branch_id` FROM `bankcslist` WHERE `cellsite_id`=$cellid;";
             $bank=$this->queryDB($query);

             if ($bank) {
               $branch_id=$bank[0]['branch_id'];
               $query="SELECT  `company_name`,  `branch_name`  FROM  `branch`  LEFT  JOIN
`company`  ON  `company`.`company_id`=`branch`.`company_id`  WHERE  `branch_id`=$branch_id;";
               $company=$this->queryDB($query);

               if ($company) {
                 $query="select    `atmsrv_name`  from  atmsrv  left  join  machine  on
atmsrv.machine_id  =  machine.machine_id  where  machine.machine_id  in  (select  machine_id
from machine where branch_id=$branch_id)";
                 $atm=$this->queryDB($query);
                 if ($atm) {
                   $return=$atm[0]['atmsrv_name'].":     ".$company[0]['company_name']."
".$company[0]['branch_name']."           ->           ".trim($cellsite[0]['street_no']."
".$cellsite[0]['street_name']);
                 } else {
                   $return=$notfound;
                 }
               } else {
                  $return=$notfound;
               }
             } else {
               $return=$notfound;
             }


        } else {
            $return="Sorry, unable to find any banks nearby!";
          }

        } elseif ($type_id==2) { // Petrol
          $notfound="Sorry, unable to find any petrol stations nearby!";
          $query="SELECT  `cellsite_id`,  `cellsite_name`,  `street_no`,  `street_name`,
(SELECT  `district_name`  FROM  `district`  WHERE  `district_id`=`street`.`district_id`)  AS
`district`   FROM   `cellsite`   LEFT   JOIN   `street`   ON   `cellsite`.`street_id`   =
`street`.`street_id` WHERE `cellsite`.`cellsite_id`=$cellid";
          $cellsite=$this->queryDB($query);
          // can check if there are more cellsite id with the same street name also
          if ($cellsite) {
             $query="SELECT      `station_id`      FROM      `petrolcslist`      WHERE
`cellsite_id`=$cellid;";
             $station=$this->queryDB($query);
             if ($station) {
                $station_id=$station[0]['station_id'];
                $query="SELECT station_name,company_name FROM station LEFT JOIN company
ON company.company_id=station.company_id where station_id=$station_id";
                $company=$this->queryDB($query);
                if ($company) {
                   $query="select   distinct   `ptlsrv_name`   from   `ptlsrv`   where
station_id=$station_id";
```

```
                    $service_name=$this->queryDB($query);

                    if ($service_name) {
                      $return=$service_name[0]['ptlsrv_name'].":
".$company[0]['company_name']."                          ".$company[0]['station_name'].",
".trim($cellsite[0]['street_no']." ".$cellsite[0]['street_name']);
                    } else {
                      $return=$notfound;
                    }
                } else {
                  $return=$notfound;
                }

             } else {
               $return=$notfound;
             }
          } else {
            $return=$notfound;
          }
        }
      }
      $this->closeDB();


      return $return;

    }
  }

?>
```

A.2.6 classes/SMSServer.php – The SMS Server Class

```
<?php
  /*
    Jeffrey Choy's FYP

    The smsServer class:
      - it implements the Win32Service and can be installed, removed and started from
        within Windows Service, it extends the AT Commands class, so that information
        retrieved can be sent to the mobile phone
  */
  class SMSServer extends ATCommands implements Win32Service {
      private $timeout=10; // execute every 10 secs
      private $argv;
      private                $svc_param='-c                c:\fyp\servers\php5.2\php.ini
c:\fyp\sourcecodes\php\smsserver\server.php';
      private $accepted_phone_nums=array();
      protected $service_name="Jeffrey Choy's FYP SMS Sending and Retrieval Server";
      protected $service_short_name="jeffreysmsd";

      public  function  installSvc()  {  //  Handles  the  Win32  Service:installation  of
service
       $x = win32_create_service(array(
              'service' => $this->service_short_name,
              'display' => $this->service_name,
              'params' => $this->svc_param.' start'));
       if(is_bool($x) && $x==true)
          echo "The $this->service_name is successfully installed.";
        else
          echo "Unable to install $this->service_name!";
        echo "\r\n";
      }

      public function removeSvc() {  // Handles the Win32 Service: removal of service
        $x = win32_delete_service($this->service_short_name);
        if (is_long($x) && $x==0)
          echo  "The $this->service_name has been removed successfully.";
```

```
        else
          echo "Unable to remove $this->service_name!";
        echo "\r\n";
      }

      function __construct() {
        error_reporting(0);
        set_time_limit(0);
        ob_implicit_flush();
      }

      // set the acceptable phone numbers, empty means accept all
      public function setAcceptedPhoneNums($phone_array) {
        if (is_array($phone_array))
          $this->accepted_phone_nums=$phone_array;
      }

      public function setArgv($argv) { // set environmental variables
        $this->argv=$argv;
      }

      public function setSvcParam($param) { // set service parameters
        $this->svc_param=$param;
      }

      public function startSvc() { // function to start the windows service
        // handles windows service installation, uninstallation and starting
        if ($this->argv[1]=="install") {


         $this->installSvc();
          exit;
        } else if ($this->argv[1]=="uninstall") {
          $this->removeSvc();
          exit;
        } else if ($this->argv[1] != 'start') {
          die("invalid command!");
        }

        $x = win32_start_service_ctrl_dispatcher($this->service_name);
        $this->openDB();
        while (WIN32_SERVICE_CONTROL_STOP != win32_get_last_control_message()) {
          $continue=$this->read();
          if ($continue) {
            $this->process();
            $this->send();
          } else { // unable to handle as AT connection failure
          }
          sleep($this->timeout);
        }
        $this->closeDB();
      }

      // read from phone
      protected function read() {
         $status=$this->openAtPort();
         if (!$status) return false;
         $cmd=array();
         $cmd[0]="AT+CMGF=1\r";
         $cmd[1]="AT+CMGL=\"REC UNREAD\"\r";     // remember to change this back to REC
UNREAD afterwards
         $result=$this->sendToAtPort($cmd);

         // list the contents
         if ($result!=false) {
           $r=split('\+CMGL: ',$result[0]);
           if (count($r)>1) {
             for ($i=1;$i<count($r);$i++) {
               // parse the information into ID, Phone Number, Date Time and Subject
               $line2=split("\r\n",$r[$i]);
```

```
              $line1=split(",",$line2[0]);
              $id=$line1[0];
              $phone=str_replace('"',"",$line1[2]);
              $date=split('/',str_replace('"',"",$line1[4]));
              $date='20'.$date[0].'-'.$date[1].'-'.$date[2];
              $time=split('\+',$line1[5]);
              $time=$time[0];
              $datetime=$date." ".$time;
              $subject=str_replace("'","''",$line2[1]);
              // put the contents to inbox
              $query="INSERT   INTO   `sms_inbox`   (`time`,`phone`,`message`)   VALUES
('".$datetime."','".$phone."','".$subject."'";
              $count=$this->execSQL("INSERT INTO `sms_inbox` (`time`,`phone`,`message`)
VALUES ('".$datetime."','".$phone."','".$subject."')");
              if ($count==1) {
                // deletes the message from phone
                $cmd=array();
                $cmd[0]="AT+CMGD=".$id."\r";
                $this->sendToAtPort($cmd);   // remember to activate this afterwards
              }
            }
          }
        }
        $this->CloseAtPort();
        return true;


      }

      private function process() {  // read inbox where processed=0
        $result=$this->queryDB("SELECT * FROM `sms_inbox` WHERE `status`='UNREAD' ORDER
BY `id`");
        if ($result) {
          for ($i=0;$i<count($result);$i++) {
            $id=$result[$i]['id'];
            // update non processed to 1 or delete it from inbox
            $status='ERROR';
            if                  ($this->accepted_phone_nums==array()                  ||
in_array($result[$i]['phone'],$this->accepted_phone_nums)) {
              $phone=$result[$i]['phone'];
              $message=$result[$i]['message'];

              // process items
              $dbstr=$this->getDBConnectionString();
              $parser=new smsParser;
              if (is_array($dbstr))
                $parser-
>setDBConnectionString($dbstr['host'],$dbstr['user'],$dbstr['pwd'],$dbstr['db']);
              else
                $parser->setDBConnectionStringDefault();
              $send_msg=$parser->parse($message);
              $send_msg=str_replace("'","''",$send_msg);
              $parser=null;

              // put to outbox
              $count=0;
              if ($send_msg!=false && $send_msg!="")
                $count=$this->execSQL("INSERT   INTO   `sms_outbox`   (`phone`,`message`)
VALUES ('".$phone."','".$send_msg."')");

              // change the inbox status
              if ($count==1)
                $status='READ';
            }
            $this->execSQL("UPDATE   `sms_inbox`   SET   `status`='".$status."'   WHERE
`id`=$id"); // update the inbox so that it will not ne processed next time
          }
        }
      }
```

```
      protected function send() {
        // read outbox from database
        $result=$this->queryDB("SELECT * FROM `sms_outbox` ORDER BY `retry`,`id`");
        if ($result) {
           $this->openAtPort();

          // loop one by one
          for ($i=0;$i<count($result);$i++) {
            $id=$result[$i]['id'];

            //   send message
            $cmd[0]="AT+CMGF=1\r";
            $cmd[1]="AT+CMGS=\"".$result[$i]['phone']."\"\r";
            $cmd[2]=$result[$i]['message'].chr(26);
            $status=$this->sendToAtPort($cmd);

            //   update status ie) increment failure count or move to sent
            if (!$status)
              $this->execSQL("UPDATE `sms_outbox` SET `retry`=`retry`+1 WHERE `id`=$id");
            else
              $this->execSQL("DELETE FROM `sms_outbox` WHERE `id`=$id");
          }


          $this->CloseAtPort();
        }
      }
  }
?>
```

A.2.7 classes/Win32Service.php – The Win32 Interface

```
<?php
  /*
    Jeffrey Choy's FYP

    The Win32Service
      - the interface for Win32Service
  */
  interface Win32Service {
     public function installSvc();  // to install the service
     public function removeSvc();   // to remove the service
     public function startSvc();    // to start the service
  }

?>
```

OXFORD BROOKES UNIVERSITY

A.3 Server End Program – Web Interface for Editing Details (PHP)

A.3.1 index.php – the Main Interface

```
<?php

  function show_index($dbh) {
    echo "<blockquote><p align=left>";
     $stmt=$dbh->query('show tables in fyp');
     $result = $stmt->setFetchMode(PDO::FETCH_NUM);
     while ($row = $stmt->fetch()) {
       if (substr($row[0],0,4)!="sms_")
         echo                          "<a                          href='./?table=".$row[0]."'
target='_top'>".ucfirst($row[0])."</a><br>";
     }
     $stmt=null;
  }


  // use the connection here
   include "db_connect.php";

   if (!isset($_REQUEST['table'])) {
     $header="SMS Service Tracker";
     $header2="Edit a table below";
   } elseif (substr($_REQUEST['table'],0,4)!="sms_") {
     $header="View Details";
     $header2=strtoupper($_REQUEST['table']);
   } else {
     $header="<script>alert('Invalid table!');</script>";
     $header2="<script>history.back();</script>";
   }

   if (!isset($_REQUEST["showbody"]))
     include "header.php";

   include "db_functions.php";
   if    (isset($_REQUEST['table'])    &&    substr($_REQUEST['table'],0,4)!="sms_"    &&
isset($_REQUEST['showbody'])) {
     echo "    <style>
      body { font-family:trebuchet ms; font-size:12pt}
      h1 { font-family:arial; font-size:30pt; }
    </style><body  bgcolor=#ffffcc><table  height=90%  width=100%  valign=center><tr><td
align=center>";
     view_table($dbh,$_REQUEST['table']);
     echo "</td><tr><table>";
   } elseif (isset($_REQUEST['showbody'])) {
     echo "    <style>
      body { font-family:trebuchet ms; font-size:12pt}
      h1 { font-family:arial; font-size:30pt; }
    </style><body  bgcolor=#ffffcc><table  height=90%  width=100%  valign=center><tr><td
align=center>";
     show_index($dbh);
     echo "</td><tr><table>";
   } else {
     echo "<iframe src='./?";
     if   (isset($_REQUEST['table'])   &&   substr($_REQUEST['table'],0,4)!="sms_")   echo
"table=".$_REQUEST['table']."&";
     echo "showbody=1' frameborder='0' width='520' height='340'></iframe>";
     if   (isset($_REQUEST['table'])   &&   substr($_REQUEST['table'],0,4)!="sms_")   echo
"<br><p align=left><a href='./'>&lt; back</a></p>";
```

```
   }
  // and now we're done; close it
  include "db_disconnect.php";
  if (!isset($_REQUEST["showbody"]))
    include "footer.php";
?>
```

A.3.2 db_functions.php – The Database Functions

```
<?php

  function show_error($msg) {
    echo "<script language='javascript'>
            alert('$msg'); history.back();
          </script>";
  }

  function foreign_key_check($dbh,$table,$db='fyp') {
    $query="select      `TABLE_NAME`,      `COLUMN_NAME`,      `REFERENCED_TABLE_NAME`,
`REFERENCED_COLUMN_NAME`      from      `information_schema`.`key_column_usage`      WHERE
`REFERENCED_TABLE_SCHEMA`='$db' AND `TABLE_SCHEMA`='$db' AND `TABLE_NAME`='$table'";
    $stmt=$dbh->query($query);
    $result = $stmt->setFetchMode(PDO::FETCH_NUM);
    $i=0;
    $array=array();
    while ($row = $stmt->fetch()) {

$array[$i]=array(table=>$row[0],column=>$row[1],ref_table=>$row[2],ref_column=>$row[3]);
      $i++;
    }
    $stmt=null;
    return $array;
  }

  function view_table($dbh,$table,$limit="100") {
     $foreign_tables=foreign_key_check($dbh,$table);
     $special_keys="";
     $tables="";
     $columns=array();
     $joins="";
     for ($i=0;$i<count($foreign_tables);$i++) {
       if ($joins=="")
          $joins=" WHERE ";
       else
          $joins.=" AND ";


$special_keys.=",`".$foreign_tables[$i]['ref_table']."`.`".$foreign_tables[$i]['ref_tabl
e']."_name`";
```

```
       $tables[$i]=$foreign_tables[$i]['ref_table'];
       $columns[$i]=$foreign_tables[$i]['column'];

$joins.="`".$foreign_tables[$i]['table']."`.`".$foreign_tables[$i]['column']."`=`".$fore
ign_tables[$i]['ref_table']."`.`".$foreign_tables[$i]['ref_column']."`";
     }
     if ($tables!="") {
       $tbls=array_unique($tables);
       $foreign_table=",`".implode('`,`',$tbls)."`";
     }
     $resource="table=$table";

     $query="select  `$table`.*$special_keys  from  `$table`$foreign_table$joins  limit
$limit";
     $stmt=$dbh->query($query);
     $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
     $i=0;
     echo "<p align=center><a href='add_edit.php?table=$table' target='_top'>Add a new
record</a></p>";
     echo "<table cellspacing=0 border=1 cellpadding=2>";
     while ($row = $stmt->fetch()) {

        if ($i==0) {
          // print the header
          $ignorelist=array();
          echo "<tr>";
          echo "<td bgcolor=black> </td>";
          foreach($row as $key=>$column) {
            $skey=$key;
            if (in_array($key,$columns)) {
              $changed_key=str_replace("_id","_name",$key);
              $skey=str_replace("_id","",$key);
              array_push($ignorelist,$changed_key);
            }
            if (!in_array($key,$ignorelist)) {
               echo "<td bgcolor=black><font color=white>$skey</font></td>";
            }
          }
          echo "</tr>";
        }

        // print the rows


      if ($i%2==0)
            $bgcolor="#eeeeee";
        else
```

```
            $bgcolor="#cccccc";

        $col=0;
        echo "<tr bgcolor=$bgcolor>";
        foreach($row as $key=>$column) {
          if ($col==0) {
            $resource="table=$table&column=$key&id=$column";
            echo      "<td      style='font-size:8pt'      nowrap><a      target='_top'
href='add_edit.php?$resource'>edit</a>
                  <a target='_top' href='javascript:if (confirm(\"Are you sure you want
to delete the row?\")) {location.href=\"del.php?$resource\";}'>del</a></td>";
          }
          $col++;
          if (!in_array($key,$columns) && !in_array($key,$ignorelist)) {
             $print=$column;
          } elseif (!in_array($key,$ignorelist)) {
             $changed_key=str_replace("_id","_name",$key);
             $print=$row[$changed_key];
          } else {
             unset($print);
             unset($header);
          }


          if (isset($print))
            echo "<td>".$print."</td>";

        }
          echo "</tr><tr>";
        $i++;
     }
     echo "</table>";
     if ($i>0)
       echo "<p align=center><a href='add_edit.php?table=$table' target='_top'>Add a new
record</a></p>";
  }

  function show_row($dbh,$table,$editid=false) {
     $foreign_tables=foreign_key_check($dbh,$table);
     $results=array();


     if ($editid!=false) {
       $query="select * from `$table` where `".$table."_id`=".$editid;
       $stmt=$dbh->query($query);
       $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
       $row=$stmt->fetch();
```

```
       foreach($row as $key=>$column) {
         $results[$key]=$column;
       }
     }
     $query="SELECT
`column_name`,`data_type`,`character_maximum_length`,`numeric_precision`,`extra`    FROM
`information_schema`.`columns` where `table_schema`='fyp' and `table_name`='$table'";
     $stmt=$dbh->query($query);
     $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
     echo "<form action='add_edit_process.php' method='post'>";
     echo "<input type=hidden name=table value='".$table."'>";
     $edit_orig_value="<input           type=hidden           name='".$table."_orig_id'
value='".$results[$table.'_id']."'>";
     echo "<table>";
     while ($row = $stmt->fetch()) {
       $input="";
       echo "<tr>";
       echo "<td>".$row["column_name"]."</td>";
       echo "<td>";
       $value="";
       for ($i=0;$i<count($foreign_tables);$i++) {
         if ($foreign_tables[$i]['table']==$table &&
             $foreign_tables[$i]['column']==$row["column_name"]) {
             $query2="select
`".$foreign_tables[$i]['ref_table']."_id`,`".$foreign_tables[$i]['ref_table']."_name`
from             `".$foreign_tables[$i]['ref_table']."`             order             by
`".$foreign_tables[$i]['ref_table']."_id`";
             $stmt2=$dbh->query($query2);

             $result2 = $stmt2->setFetchMode(PDO::FETCH_ASSOC);
             $input="<select name='".$row["column_name"]."'>";
             $input.="<option value=''></option>";
             while ($row2 = $stmt2->fetch()) {
               $input.= "<option value='";
               $input.=$row2[$foreign_tables[$i]['ref_table']."_id"]."'";
               if
($results[$row["column_name"]]==$row2[$foreign_tables[$i]['ref_table']."_id"])  $input.=
" SELECTED";
               $input.= ">";

             $input.=$row2[$foreign_tables[$i]['ref_table']."_name"];
               $input.="</option>";
             }
             $input.="</select>";
             $stmt2=null;
```

```
            break;
          }
        }

        if ($input=="") {
          echo "<input type=text name='".$row["column_name"]."' ";
          if (isset($results[$row["column_name"]])) $value=$results[$row["column_name"]];
          if ($row["extra"]=="auto_increment") {
            echo "readonly style='background:#aaaaaa'";
            if ($value=="")
              $value="(auto_increment)";
          }
          echo " value='$value' ";
          echo ">";
        } else {
          echo $input;
        }
        echo "</td>";
        echo "</tr>";
      }
      echo "</table>";
      if ($editid!=false) {
        echo "\r\n<input type=submit value='Edit Record'>\r\n";
        echo $edit_orig_value;
        echo "<input type=hidden name='type' value='edit'>";
      } else {
        echo "<input type=submit value='Add Record'>";
        echo "<input type=hidden name='type' value='add'>";
      }
      $stmt=null;
      echo "</form>";
      echo "<p align=left> <a href='./'>&lt; Back</a>";
  }
  function row_del($dbh,$table,$column,$id) {
    $passed=false;
    try {


      $stmt = $dbh->prepare("DELETE FROM `$table` WHERE `$column`=".intval($id));
      $passed = $stmt->execute();
      $stmt=null;
    } catch (PDOException $e) { }
    return $passed;
  }
?>
```

A.3.3 header.php – The Script to Output the Header

```
<html>
  <head>
    <title>SMS Service Tracker - Jeffrey Choy's Final Year Project</title>
    <style>
      body { font-family:trebuchet ms; font-size:10pt}
      h1 { font-family:arial; font-size:30pt; }
    </style>
  </head>
  <body bgcolor=#aaaaaa>
    <table width=940 border=0 cellspacing=0 cellpadding=5 bgcolor=black align=center>
      <tr>
        <td width=359><img src=ckc.jpg></td>
        <td colspan=2 bgcolor=white style='border:2px solid black' valign=top><p> 
          <table  border=0  cellspacing=0  cellpadding=0  valign=top  width=100%><tr><td
nowrap align=right bgcolor=#f9cccc>
            <font          style='font-family:trebuchet          ms;font-size:27pt;font-
weight:bolder;color:#ffffff'><?php echo $header;?> </font>
          </td></tr></table><p>
          <table valign=top border=0 cellspacing=0 cellpadding=0 width=95% align=center>
          <tr><td bgcolor=black nowrap align=center>
          <font style='font-family:trebuchet ms;font-size:12pt;color:#ffffff'><?php echo
$header2;?></font>
          </td></tr>
          <tr><td bgcolor='#ffffcc' height=400 nowrap align=left>
             <table    width=100%    border=0    cellspacing=0    cellpadding=0><tr><td
align=center>
```

A.3.4 footer.php – The Script to Output the Footer

```
       </td></tr></table>
     </td></tr></table>
   </td>
  </tr>
  <tr>
    <td    colspan=2    style='border-left:2px    solid    black;border-top:2px    solid
#440000;border-bottom:2px     solid     black'     bgcolor=#990000><font     style='font-
family:arial;font-size:8pt;color:#aaaaaa'><font          style='font-weight:bolder;font-
size:9pt;color:#aaaaaa'>SMS   Service   Tracker</font><br>Jeffrey   Choy's   Final   Year
Project</font></td>
    <td   style='border-right:2px   solid   black;border-top:2px   solid   #440000;border-
bottom:2px    solid    black'    bgcolor=#990000><font    style='font-family:arial;font-
size:8pt;color:#aaaaaa' valign=top align=right><p align=right>BSc(Hons) in Computing and
Information Systems<br>Oxford Brookes University</font></td>
  </tr>
  </table>
  </body>
</html>
```

A.3.5 del.php – The Script to Delete Details from Database

```
<?php

  if (isset($_REQUEST["table"]) &&
        isset($_REQUEST["column"]) &&
        isset($_REQUEST["id"]) &&
        $_REQUEST["table"]!="" &&
        $_REQUEST["column"]!="" &&
        intval($_REQUEST["id"])>0
       ) {

      if(@$_SERVER['HTTP_REFERER'] != '')
         $location=$_SERVER['HTTP_REFERER']."?table=".$_REQUEST["table"];
      else
```

```
         $location='index.php?table='.$_REQUEST["table"];

      // use the connection here
      include "db_connect.php";

      include "db_functions.php";

$result=row_del($dbh,$_REQUEST["table"],$_REQUEST["column"],intval($_REQUEST["id"]));

      if (!$result)
        show_error('Foreign  key  constrain  error!\\r\\nUnable  to  delete  specified
'.$_REQUEST["table"].'!');
      else
        echo "<script language='javascript'>
                  location.href='$location';
               </script>";

      // and now we're done; close it
      include "db_disconnect.php";

  } else {

      show_error('Invalid request!');

  }

?>
```

A.3.6 add_edit.php – The Script to Add / Edit Details

```
<?php

  if (isset($_REQUEST["table"]) &&
        $_REQUEST["table"]!=""
       ) {

      if(@$_SERVER['HTTP_REFERER'] != '')
         $location=$_SERVER['HTTP_REFERER']."?table=".$_REQUEST["table"];
      else
         $location='index.php?table='.$_REQUEST["table"];

      // use the connection here
      include "db_connect.php";

      $header=(isset($_REQUEST["id"]))?"Edit":"Add";
      $header.=" Table";

      $header2=strtoupper($_REQUEST["table"]);

      if (!isset($_REQUEST["showbody"]))
        include "header.php";

      include "db_functions.php";

      $result=show_row($dbh,$_REQUEST["table"],$_REQUEST["id"]);

      // and now we're done; close it
      include "db_disconnect.php";

      if (!isset($_REQUEST["showbody"]))
        include "footer.php";

 } else {

      show_error('Invalid request!');

  }
?>
```

A.3.7 add_edit_process.php – Scripts to Process Adds and Edits

```
<?php
  $string_type=array('varchar','char','text','longtext','date','datetime','timestamp');
  if (isset($_REQUEST["table"]) &&
        $_REQUEST["table"]!=""
       ) {

      if(@$_SERVER['HTTP_REFERER'] != '')
         $location=$_SERVER['HTTP_REFERER']."?table=".$_REQUEST["table"];
      else
         $location='index.php?table='.$_REQUEST["table"];

      // use the connection here
      include "db_connect.php";

      include "db_functions.php";

      if           ($_REQUEST[$_REQUEST["table"]."_id"]=="(auto_increment)"           ||
$_REQUEST["type"]=="add") {
       $type='add';
      } else {
       $type='edit';
      }
      $query="SELECT
`column_name`,`data_type`,`character_maximum_length`,`numeric_precision`,`extra`     FROM
`information_schema`.`columns`          where          `table_schema`='fyp'          and
`table_name`='".$_REQUEST["table"]."'";
      $stmt=$dbh->query($query);
      $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
      $q1=""; $q2="";
      while ($row = $stmt->fetch()) {
          if ($type=='add' && $row["column_name"]!=$_REQUEST["table"]."_id") {
             if ($q1!="") { $q1.=","; $q2.=","; }
             $q1.="`".$row["column_name"]."`";
             if (in_array($row['data_type'],$string_type)) $q2.="'";
             $q2.=str_replace("\\'","''",$_REQUEST[$row["column_name"]]);
             if (in_array($row['data_type'],$string_type)) $q2.="'";
          } elseif ($type=='edit'){ //&& $row["column_name"]!=$_REQUEST["table"]."_id")
{
             if ($q1!="") $q1.=",";
             $q1.="`".$row["column_name"]."`=";
             if (in_array($row['data_type'],$string_type)) $q1.="'";
             $q1.=str_replace("\\'","''",$_REQUEST[$row["column_name"]]);
             if (in_array($row['data_type'],$string_type)) $q1.="'";
          }
      }

      if ($type=='add' && $_REQUEST[$_REQUEST["table"]."_id"]!="(auto_increment)") {
         if  (intval($_REQUEST[$_REQUEST["table"]."_id"])<=0)  $q1=",,,/ERROR/,,,";    //
stupid way to cause error
         if ($q1!="") { $q1.=","; $q2.=",";}
         $q1.="`".$_REQUEST["table"]."_id`";
         $q2.=intval($_REQUEST[$_REQUEST["table"]."_id"]);
      }
      $stmt=null;

      if ($type=='add')
        $query="INSERT INTO `".$_REQUEST["table"]."` (".$q1.") VALUES (".$q2.")";
      elseif ($type='edit')
        $query="UPDATE        `".$_REQUEST["table"]."`        SET        $q1        WHERE
`".$_REQUEST["table"]."_id`=".intval($_REQUEST[$_REQUEST["table"]."_orig_id"]);

      $passed=false;
      try {


        $stmt = $dbh->prepare($query);
         $passed = $stmt->execute();
```

```
        $stmt=null;
      } catch (PDOException $e) { }
      if (!$passed)
       echo "<script language='javascript'>
                              alert('Update Failed!');
                              location.href='/?table=".$_REQUEST['table']."';
                         </script>";
      else
        echo "<script language='javascript'>
                    alert('Update Success!');
                    location.href='/?table=".$_REQUEST['table']."';
               </script>";
      // and now we're done; close it
      include "db_disconnect.php";
  } else {
      show_error('Invalid request!');
  }

?>
```

A.3.8 config.php – The Configuration Script

```
<?php
  // database configuration
  $db_host="localhost";
  $db_name="fyp";
  $db_user="root";
  $db_pwd="";

  header( 'Expires: Tue, 31 Jul 2007 23:59:59 GMT' );
  header( 'Last-Modified: ' . gmdate( 'D, d M Y H:i:s' ) . ' GMT' );
  header( 'Cache-Control: no-store, no-cache, must-revalidate' );
  header( 'Cache-Control: post-check=0, pre-check=0', false );
  header( 'Pragma: no-cache' );
?>
```

A.3.9 db_connect.php – The Database Connection Script

```
<?php
  require_once 'config.php';
  if (!isset($dbh) || !is_object($dbh))
    $dbh = new PDO('mysql:host='.$db_host.';dbname='.$db_name, $db_user, $db_pwd);
?>
```

A.3.10 db_disconnect.php – The Database Disconnection Script

```
<?php
  if (isset($dbh) && is_object($dbh))
    $dbh=null;
?>
```

B. Database Structure Listing

Below are the database structure listings of the project. The listings include the main database structure, the SMS mailboxes tables and the information services tables.

B.1 The Database

B.1.1 fyp – The Main Database

```
DROP DATABASE IF EXISTS `fyp`;
CREATE DATABASE `fyp`;
USE `fyp`;
```

B.2 The SMS mailboxes

B.2.1 sms_inbox – The SMS Inbox Table

```
CREATE TABLE `sms_inbox` (
  `id` int(11) NOT NULL auto_increment,
  `time` datetime NOT NULL,
  `phone` varchar(20) NOT NULL,
  `message` varchar(170) NOT NULL,
  `status` enum('UNREAD','READ','ERROR') NOT NULL default 'UNREAD',
  PRIMARY KEY  (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

B.2.1 sms_outbox – The SMS Inbox Table

```
CREATE TABLE `sms_outbox` (
  `id` int(11) NOT NULL auto_increment,
  `time` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `phone` varchar(20) NOT NULL,
  `message` varchar(170) NOT NULL default '',
  `retry` int(11) default '0',
  PRIMARY KEY  (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

OXFORD BROOKES UNIVERSITY

B.3 The Information Services

B.3.1 atmsrv – The ATM Service Table

```
CREATE TABLE `atmsrv` (
  `atmsrv_id` int(11) NOT NULL auto_increment,
  `atmsrv_name` varchar(50) NOT NULL,
  `atmsrv_dsc` varchar(50) default NULL,
  `machine_id` int(11) NOT NULL,
  PRIMARY KEY  (`atmsrv_id`),
  KEY `machine_id` (`machine_id`),
  CONSTRAINT `atmsrv_fk` FOREIGN KEY (`machine_id`) REFERENCES `machine` (`machine_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

B.3.2 bankcslist – The Bank Cellsite List Table

```
CREATE TABLE `bankcslist` (
  `bankcslist_id` int(11) NOT NULL auto_increment,
  `cellsite_id` int(11) NOT NULL,
  `branch_id` int(11) NOT NULL,
  PRIMARY KEY  (`bankcslist_id`),
  KEY `cellsite_id` (`cellsite_id`),
  KEY `branch_id` (`branch_id`),
  CONSTRAINT `bankcslist_fk` FOREIGN KEY (`cellsite_id`) REFERENCES `cellsite`
(`cellsite_id`),
  CONSTRAINT `bankcslist_fk1` FOREIGN KEY (`branch_id`) REFERENCES `branch` (`branch_id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=latin1;
```

B.3.3 branch – The Branch Table

```
CREATE TABLE `branch` (
  `branch_id` int(11) NOT NULL auto_increment,
  `branch_name` varchar(50) NOT NULL,
  `company_id` int(11) NOT NULL,
  PRIMARY KEY  (`branch_id`),
  KEY `company_id` (`company_id`),
  CONSTRAINT `branch_fk` FOREIGN KEY (`company_id`) REFERENCES `company` (`company_id`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=latin1;
```

B.3.4 cellsite – The Cellsite Table

```
CREATE TABLE `cellsite` (
  `cellsite_id` int(11) NOT NULL,
  `cellsite_name` varchar(20) NOT NULL,
  `street_id` int(11) NOT NULL,
  PRIMARY KEY  (`cellsite_id`),
  UNIQUE KEY `cellsite_name` (`cellsite_name`),
  UNIQUE KEY `street_id` (`street_id`),


  CONSTRAINT `cellsite_fk` FOREIGN KEY (`street_id`) REFERENCES `street` (`street_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

B.3.5 company – The Company Table

```
CREATE TABLE `company` (
  `company_id` int(11) NOT NULL auto_increment,
  `company_name` varchar(20) NOT NULL,
  `type_id` int(11) NOT NULL,
  PRIMARY KEY  (`company_id`),
  UNIQUE KEY `company_name` (`company_name`),
  KEY `type_id` (`type_id`),
  CONSTRAINT `company_fk` FOREIGN KEY (`type_id`) REFERENCES `type` (`type_id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

B.3.6 district – The District Table

```
CREATE TABLE `district` (
  `district_id` int(11) NOT NULL auto_increment,
  `district_name` varchar(50) NOT NULL,
  PRIMARY KEY  (`district_id`),
  UNIQUE KEY `district_name` (`district_name`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

B.3.7 machine – The Machine Table

```
CREATE TABLE `machine` (
  `machine_id` int(11) NOT NULL auto_increment,
  `machine_name` varchar(50) NOT NULL,
  `machine_dsc` text,
  `branch_id` int(11) NOT NULL,
  PRIMARY KEY  (`machine_id`),
  KEY `branch_id` (`branch_id`),
  CONSTRAINT `machine_fk` FOREIGN KEY (`branch_id`) REFERENCES `branch` (`branch_id`)
) ENGINE=InnoDB AUTO_INCREMENT=29 DEFAULT CHARSET=latin1;
```

B.3.8 petrolcslist – The Petrol Cellsite List Table

```
CREATE TABLE `petrolcslist` (
  `petrolcslist_id` int(11) NOT NULL auto_increment,
  `cellsite_id` int(11) NOT NULL,
  `company_id` int(11) NOT NULL,
  `station_id` int(11) NOT NULL,
  PRIMARY KEY  (`petrolcslist_id`),
  KEY `cellsite_id` (`cellsite_id`),
  KEY `company_id` (`company_id`),
  KEY `station_id` (`station_id`),
  CONSTRAINT  `petrolcslist_fk`  FOREIGN  KEY  (`cellsite_id`)  REFERENCES  `cellsite`
(`cellsite_id`),
  CONSTRAINT  `petrolcslist_fk1`  FOREIGN  KEY  (`station_id`)  REFERENCES  `station`
(`station_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

B.3.9 ptlsrv – The Petrol Service Table

```
CREATE TABLE `ptlsrv` (
  `ptlsrv_id` int(11) NOT NULL auto_increment,
  `ptlsrv_name` varchar(50) NOT NULL,
  `ptlsrv_dsc` varchar(50) default NULL,
  `station_id` int(11) NOT NULL,
  PRIMARY KEY  (`ptlsrv_id`),
  KEY `station_id` (`station_id`),
  CONSTRAINT `ptlsrv_fk` FOREIGN KEY (`station_id`) REFERENCES `station` (`station_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

B.3.10 station – The Petrol Station Table

```
CREATE TABLE `station` (
  `station_id` int(11) NOT NULL auto_increment,
  `station_name` varchar(50) NOT NULL,
  `company_id` int(11) NOT NULL,
  PRIMARY KEY  (`station_id`),
  KEY `company_id` (`company_id`),
  CONSTRAINT `station_fk` FOREIGN KEY (`company_id`) REFERENCES `company` (`company_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

B.3.11 street – The Street Table

```
CREATE TABLE `street` (
  `street_id` int(11) NOT NULL auto_increment,
  `street_name` varchar(50) NOT NULL,
  `street_no` varchar(20) default NULL,
  `district_id` int(11) NOT NULL,
  PRIMARY KEY  (`street_id`),
  UNIQUE KEY `district_id` (`district_id`,`street_no`,`street_name`),
  KEY `district_id_2` (`district_id`),
  CONSTRAINT    `street_fk`    FOREIGN    KEY    (`district_id`)    REFERENCES    `district`
(`district_id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=latin1;
```

B.3.12 type – The Service Type Table

```
CREATE TABLE `type` (
  `type_id` int(11) NOT NULL auto_increment,
  `type_name` varchar(20) NOT NULL,
  PRIMARY KEY  (`type_id`),
  UNIQUE KEY `type_name` (`type_name`),
  UNIQUE KEY `type_name_2` (`type_name`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

C. Project Proposal

Oxford Brookes University

U08096 Proposal of Final Year Project

SMS Service Tracker

Student Name:	Choy Man Kong, Jeffrey
Student ID:	0301-0107-0324
Supervisor:	Lo Chi Wing, Peter
Module Number:	U08096
Programme:	BSc (Hons) Computer and Information System
Date:	01 May 2008

OXFORD BROOKES UNIVERSITY

Table of Contents

Table of Contents 102
Introduction 103
Objective 103
Initial Ideas 103
  Major functions 103
    Mobile Phone Interface 103
    SMS Parsing Function 104
    Service Information Web Interfaces 104
Project Requirements 105
  Mobile Phone Platform 105
  Server Platform 105
Reference 106

OXFORD BROOKES UNIVERSITY

Introduction

In recent rapid development of the century, mobile phones have become an indispensable communications tool and many services have provided in various field on top of the mobile phone platform. While mobile phone has provided easy access to various services, such as mobile banking, some of the features have never been thought of. In order to fully utilize mobile phone as a communication platform, some more new technologies and new services can be introduced to enhance our lifestyle.

Objective

It is very common that when you are on the street, you will want to find a specific type of shop that you have never been to and has no knowledge of the location of that shop. Say if you want to find an ATM machine of a particular bank, what you would normally do is to blindly ask pedestrians and hope that the person can answer you the location. This action may be very time consuming and there is a high chance that you will not get the correct information. However, by using a mobile phone as a communication platform, we can now implement a service that can utilize the location information of the mobile phone to find out information on where you are located.

By having the above idea in mind, the main objective of this project is to develop information system that provides the useful details for users on information where he/she is located.

Initial Ideas

There are a couple of technologies that can be used, such as SMS, GPS and Wi-fi, however, with the limited numbers of GPS and Wi-fi phones, the initial idea is to compare the efficiency of the various technologies and to develop a system based on SMS, with 4 major functions, and by linking the major functions together, we can come up with an SMS service tracker service to provide details to mobile phone users.

Major functions

The four major functions are the Mobile Phone Interface, the SMS Parsing Function, the Service Information Web Interfaces, and the Service Information Sending Function, and the details of the functions are as follows.

Mobile Phone Interface

The mobile phone interface is a user end program to allow user to query specific service information to be retrieved, such as the ATM location, the petrol station location, etc.

SMS Parsing Function

The SMS Parsing function is a server end program that retrieves information from all mobile phones, parses the contents of the SMS received and then stores the information to a database for further processing.

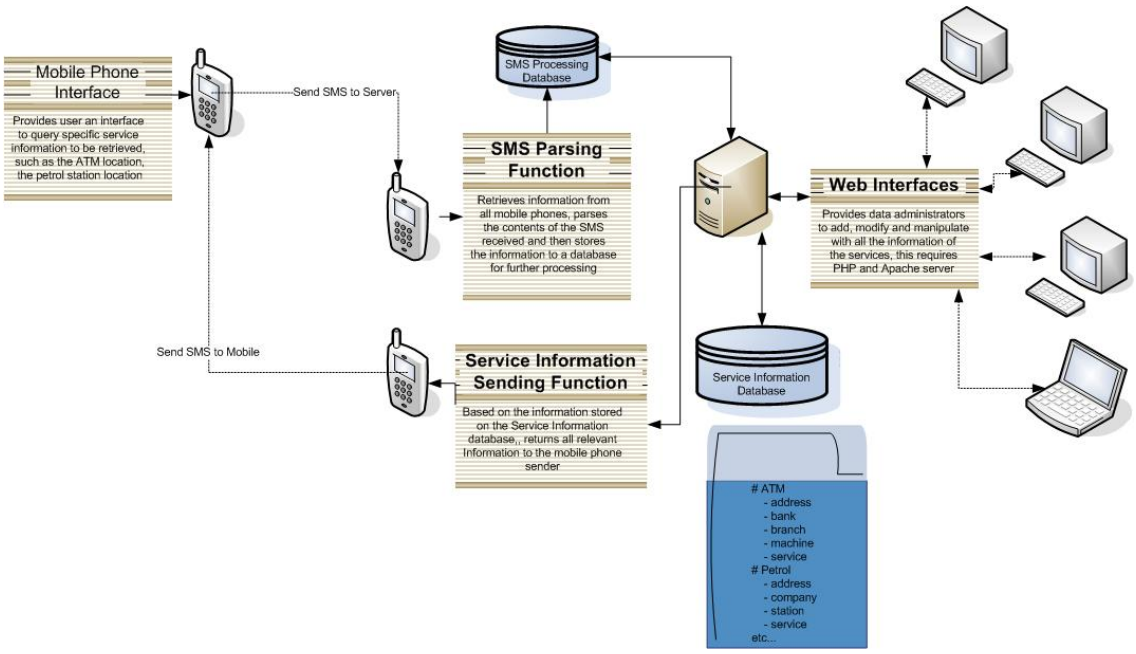
Service Information Sending Function

The service information sending function is also a server end program that retrieves information from the SMS Parsing Function database, and then based on the information stored on the service information database added by user from the service information web interfaces, returns all relevant information to the mobile phone sender.

Service Information Web Interfaces

The web interfaces provides data administrators to add, modify and manipulate with all the information of the services, this is a server end program, and allow any users connected to the same local area network to access and update information.

A diagram to show the relationship between the functions is as follows:



Project Requirements

To achieve the project goals, the following system platforms are required.

Mobile Phone Platform

The mobile phone platform should be as follows:

- Hardware:
  - Nokia or any other S60 V3 mobile phones
- Software:
  - Python for S60
  - Python Script Shell for S60

Server Platform

The server platform should be as follows:

- Hardware:
  - CPU: Intel Core 2 Duo or above CPU
  - RAM: 1GB
  - Hard disk: 120GB
- Software:
  - OS: Microsoft Windows Vista
  - Servers: Apache 2.2
    MySQL 5.0
    PHP 5.2
    Gammu 1.19

OXFORD BROOKES UNIVERSITY

Reference

1. Timo Kopomaa, “SPEAKING MOBILE: THE CITY IN YOUR POCKET”
   URL:http://www.tkk.fi/Yksikot/YTK/julkaisu/mobile.html
   Accessed: 1 May 2008

2. Nokia Resource Center, “Python for S60”
   URL:http://opensource.nokia.com/projects/pythonfors60/
   Accessed: 1 May 2008

3. Nokia, “S60 Location Technologies”
   URL:
http://www.forum.nokia.com/main/resources/technologies/location_based_services/S60_location_technologies.html
   Accessed: 1 May 2008

OXFORD BROOKES UNIVERSITY

D. Project Progress

Progress Report No : **1** Date : **25 FEB 2008**

(to be completed by student)

To study the project planning and project management control.

To study on location based-technologies such as GPS, Wi-Fi, SMS.

To study on smart phone technologies such as Windows Mobile, Symbian Os, BlackBerry.

To study on mobile platform development technologies such as Python, Java, Visual Basic, Visual C++.

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

_____

_____

Suprevisor Signature:_____ Duration **2.5** (hours)

Start Time: **19:00** End Time: **21:30**

OXFORD BROOKES UNIVERSITY

Progress Report No : **2** Date : **31 MAR 2008**

(to be completed by student)

To study the literature searching, discussing, finding information and writing the report style.

To study on server scripting technologies such as PHP, JSP

To study on web servers and database technologies such as Apache, Apache Tomcat, MySQL, PostgreSQL.

To decide on the choice of technologies such as plan the schedule

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

_____

_____

Suprevisor Signature:_____ Duration **2.5** (hours)

Start Time: **19:00** End Time: **21:30**

OXFORD BROOKES UNIVERSITY

Progress Report No : **3** Date : **28 APR 2008**

(to be completed by student)

Study how to prepare the write the report, report structure and presentation.

To design and implement the mobile phone interface

To research on alternatives to Gammu

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

_____

_____

Suprevisor Signature:_____ Duration **2.5** (hours)

Start Time: **19:00** End Time: **21:30**

OXFORD BROOKES UNIVERSITY

Progress Report No : **4** Date : **09 MAY 2008**

(to be completed by student)

To design the SMS handling database and SMS parsing function, then implement both of them.

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

_____

_____

Suprevisor Signature:_____ Duration **0.5** (hours)

Start Time: **21:30** End Time: **22:00**

OXFORD BROOKES UNIVERSITY

Progress Report No : **5** Date : **05 JUN 2008**

(to be completed by student)

To design the web interface and server information database, then implement both of them.

To populate the service information database using debug mode.

To test and check the SMS service tracker performance.

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

_____

_____

Suprevisor Signature:_____ Duration **0.5** (hours)

Start Time: **21:30** End Time: **22:00**

OXFORD BROOKES UNIVERSITY

Progress Report No : **6** Date : **09 JUL 2008**

(to be completed by student)

To perform evaluation of the project

To prepare the final report

To prepare the presentation slides

Progress evaluation (to be completed by the supervisor)

General Progress: (a) very slow (b) behind (c) satisfactory (d) impressive

Research: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Tool Studying: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Implementation: (a) very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments:

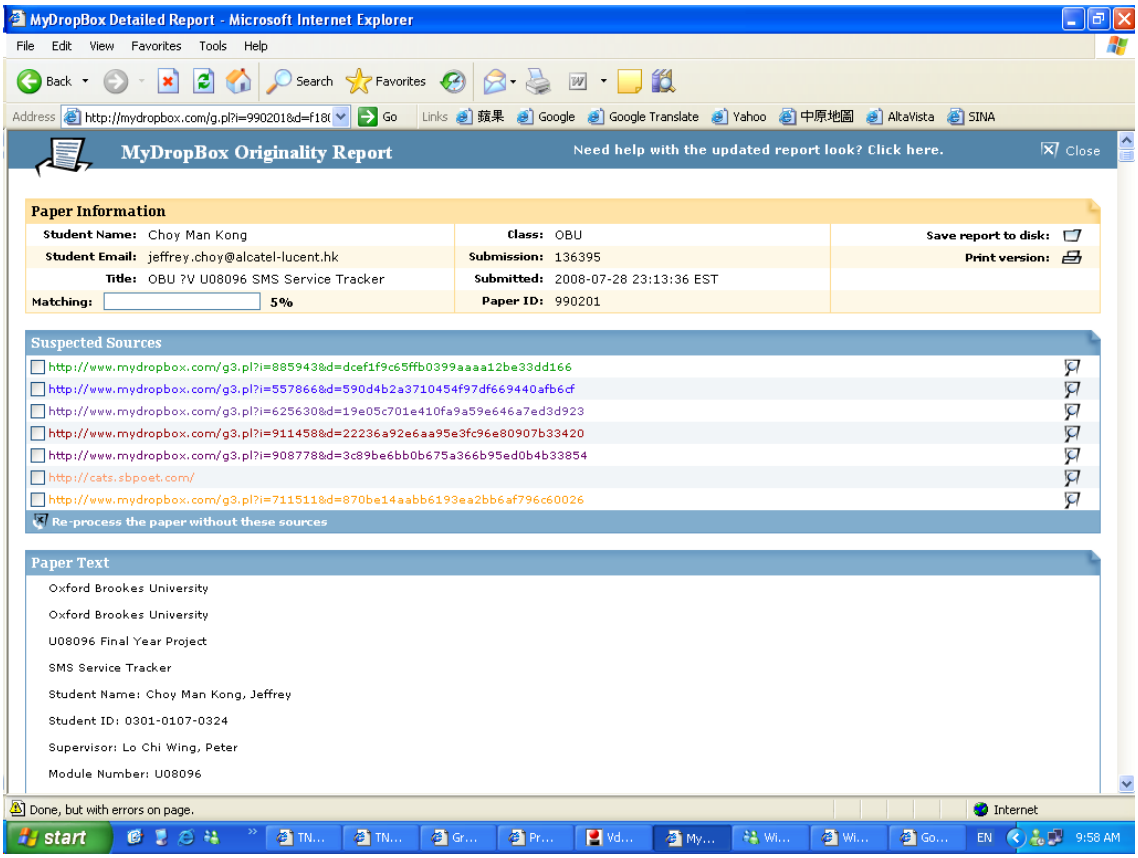
_____

_____

Suprevisor Signature:_____ Duration **0.5** (hours)

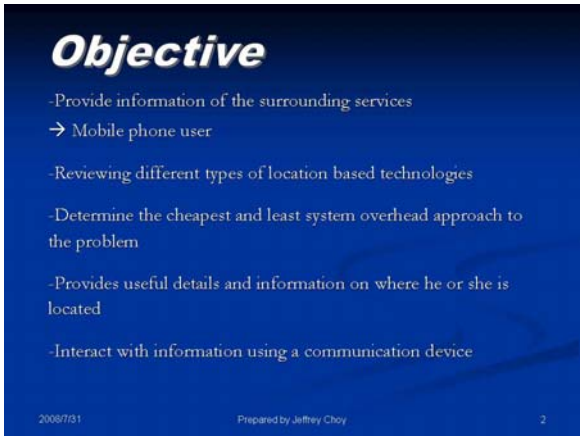
Start Time: **20:00** End Time: **20:30**

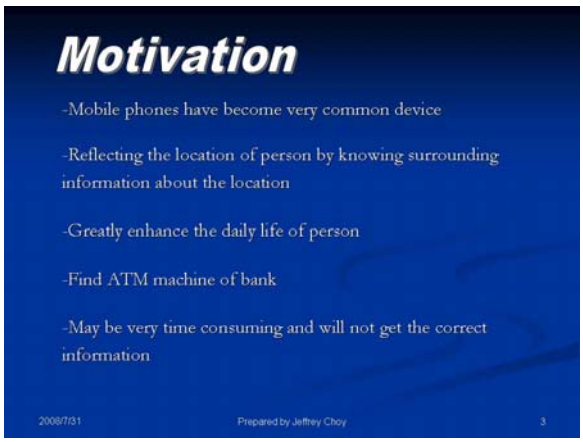
E. MyDropBox Report

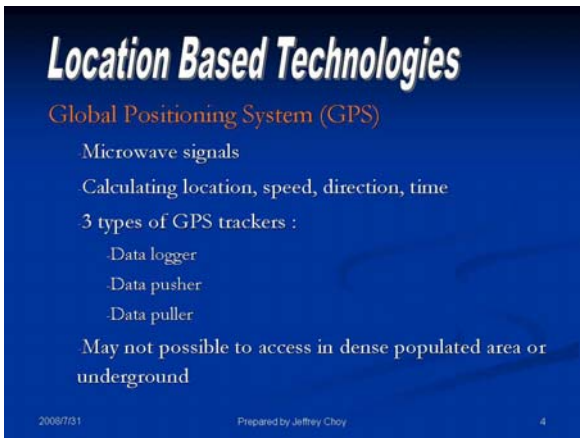


OXFORD BROOKES UNIVERSITY

F. Presentation







Smart Phone Technologies

Windows Mobile

- Developed by Microsoft
- Similar to desktop version of Microsoft Windows
- Retrieve e-mails from smart phone to view office documents
- Not a free development system for programmers to develop software

Smart Phone Technologies (Cont)

Black Berry

- Mainly used as integrated business e-mails with server
- Allow use to perform mobile phone related functions together with Internet
- Easy to use platform for send and receive e-mails.
- Not an easy task for developers to create programs

Smart Phone Technologies (Cont)

Symbian OS

- OS developed for mobile devices
- With 65% market share
- Symbian OS main platforms:
  - S60
  - UIQ
- Many development tools available for extending the functionality of smart phone
- Each application has to be signed

Decision of Technologies

Choice of Location Technology

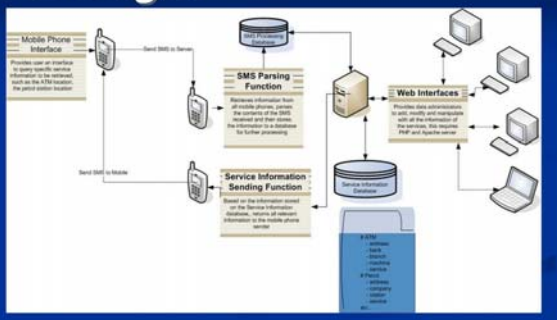
- GPS and Wi-Fi smart phone are very limited
- SMS function is freely in all mobile phone hardware
- Retrieve location information from mobile
- Using self-written program
- Security is not great concern of the project

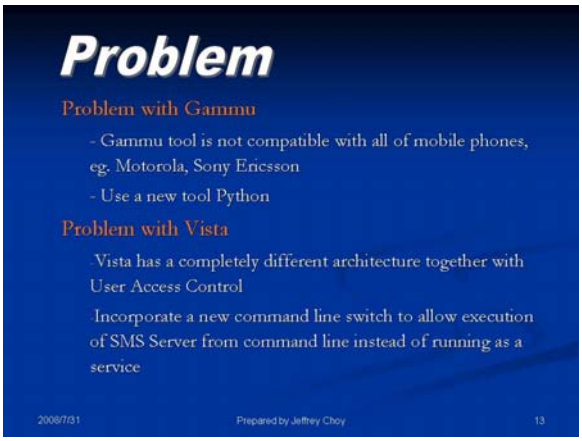
Decision of Technologies (Cont)

Choice of Mobile Platform Technology

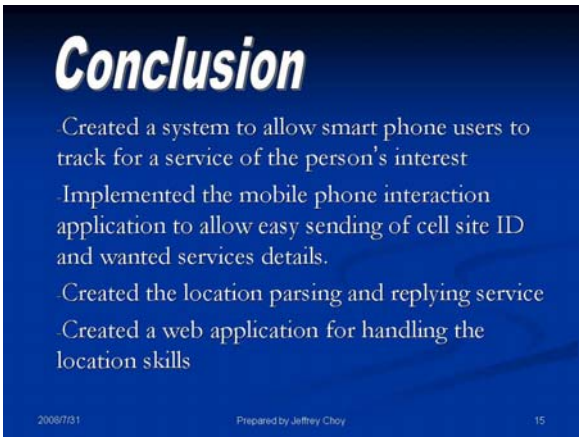
- Symbian OS is the more widely used platforms in compare
- Development technology:
  - JAVA
  - Python
- Java has not location based service support

Design

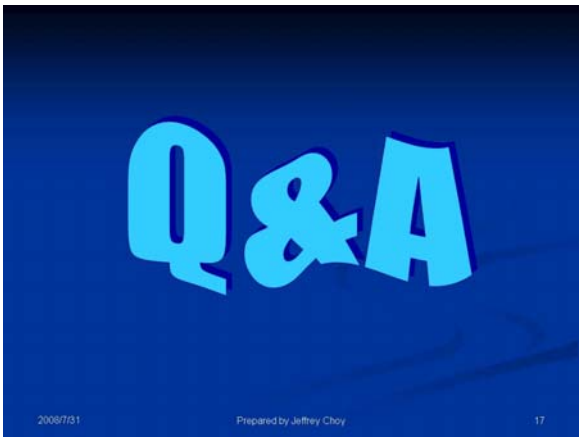



Problem
Problem with Gammu
- Gammu tool is not compatible with all of mobile phones, eg. Motorola, Sony Ericsson
- Use a new tool Python
Problem with Vista
-Vista has a completely different architecture together with User Access Control
-Incorporate a new command line switch to allow execution of SMS Server from command line instead of running as a service
2008/7/31
Prepared by Jeffrey Choy
13

Future Work
-Platform Independent
-Operator Independent
-Multiple Response
-Separating the SMS Servers
-Security
-Advertisements
2008/7/31
Prepared by Jeffrey Choy
14


Conclusion
-Created a system to allow smart phone users to track for a service of the person's interest
-Implemented the mobile phone interaction application to allow easy sending of cell site ID and wanted services details.
-Created the location parsing and replying service
-Created a web application for handling the location skills
2008/7/31
Prepared by Jeffrey Choy
15

Demostration
2008/7/31
Prepared by Jeffrey Choy
16


Q&A
2008/7/31
Prepared by Jeffrey Choy
17

Thank You
2008/7/31
Prepared by Jeffrey Choy
18