NILAI UNIVERSITY COLLEGE

OXFORD BROOKES UNIVERSITY
School of Technology

Department of Computing and Electronics

Bachelor of Science (Hons) Computing

U08096 BSc Computing Project

SMS COLLEGE RESULTS SYSTEM

Name: MAJID RAMADHAN MVULLE.
Student ID: 708-1339-03.

Supervisor: Ms. HARLINA HARUN.

Submission date: Wednesday, January 5, 2011

This project is submitted in fulfillment of the requirements for BSc (Hons) Computing, Oxford Brookes University, UK.

ABSTRACT

SMS Results System can be described as the system that enables results of any sort to be delivered to the appropriate individual through a means of SMS onto that individual's mobile phone. SCRS will enable students of a college to check for their previous semester results, check upcoming semester subjects, and do preregistration for upcoming semester modules.

This research was done to develop an SMS College Results System (SCRS), which will make use of mostly open source technology. The technologies include MySQL, GsmComm Library, etc. MySQL .NET Connector technology is used to access the MySQL database from within C# source codes.

The system that is to be developed will enhance the student's communication with their respective colleges since such a system, other than the delivery of results through SMS, could also incorporate other communications that a college may wish to pass to its students.

ACKNOWLEDGEMENT

I would like to give thanks to God for giving me life. My parents for striving with me through my education process since my first day of school until now and all the support they have given towards my education as a whole.

I sincerely thank my supervisor Ms. Harlina Harun who has been at the core of the successful completion of this project. She has guided me every step of the way by giving regular feedback on my performance on other subjects as well as this one.

I would also like to express my sincere gratitude to all the lecturers in the Department of Computing at Nilai University College who have taught me and given me the knowledge to undertake this project and complete it successfully. Most notably Ms. Rajesvary Rajoo for teaching me Object Oriented Programming Paradigm which is the programming design this project utilizes, and Mr. Philip Lee for covering Database Systems, the storage of data that this project heavily requires.

To all my friends and every one else who has been supportive throughout the striving period as I undertook this challenging task, I would like to give them a big “Thank You”.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations

SCRS	SMS College Results System
GSM	Global System for Mobile Communications (originally from: Groupe Spécial Mobile)
GPRS	General Packet Radio Service
GPA	Grade Point Average
CGPA	Cumulative Grade Point Average
HOD/HoD	Head of Department
API	Application Programming Interface
GUI	Graphical User Interface
SMSC	Short Message Service Centre

1 INTRODUCTION

1.1 Overview of the System

Text messaging has become part of our everyday lives. It is considered to be a cheap and simple way of communication in contrast to a phone call. Unlike a phone call whereby if a user does not wish to pick up the phone call or if the mobile phone - is out of network range, has no battery charge, or it does not have voicemail services - there is no way of getting a message through to the other party. However with an SMS, if the phone is facing one or all the above mentioned situations, it still thrives. Though not at 100% effectiveness, it can act as a substitute in such situations.

The use of SMS has become popular among individuals especially the youth and those of schooling age. The Mobile Data Association (MDA) (2007) notes that, in the UK approximately 4,825 billion text messages were sent during the fourth quarter of that year, averaging to 1.2 billion text messages every week. Though texting is not considered secure, young people look at it as a cool (‘hip’) way of communicating.

College Results SMS System is a relatively new and expanding system. The main idea of this system is to provide a system that will enable lecturers (HOD’s) to store student results. The system will then calculate the GPA and CGPA of the student and store this data and determine what modules the student is allowed to take the following semester. Students who have subscribed for the service can send a text message with a keyword and their college ID number and passport number to the system and it replies them their previous semester results. To get a list of modules to be offered the following semester and modules the student is allowed to take, the student sends a different keyword and this time also the student’s college ID number and passport number. The system then replies the list of modules offered the following semester for that student’s particular course and the modules the student is allowed to take according to the prerequisites that the student has completed.

1.2 Objective of the System

This project undertakes a partly comprehensive study of the SMS-based college results delivery system and students college system. This research will focus on developing desktop-based applications that can assist in storing HOD's for different departments in a college, storing students who have subscribed for the SCRS service, storing results of all students in each department, receiving of SMS with keywords, student's college ID number and passport number to check for results and modules allowed to be taken the following semester and sending the requested results and modules allowed for that particular student through SMS.

The objectives of the project can be summarized as below:

- To develop a system that will ease the process of students acquiring results by enabling them to get their results through SMS. Students will no longer have to queue up at the beginning of each semester to get their previous semester results and modules they are allowed to take.
- To develop a system that can store; students who have subscribed for SCRS service, students results, different schools in a college, departments in different schools in a college, modules in a department and HOD's of different departments in a college.
- To develop a system that will calculate GPA and CGPA of each student after their results have been stored in the system.
- To develop a system that will determine the modules a student is allowed to take the following semester after the student's results have been stored in the system.
- To develop a system that can enable students to perform preregistration of upcoming semester subjects through SMS.
- To develop a system that can receive a text message, perform a batch process if the keywords are correct and reply the output through a text message.
- To develop an SMS gateway using a GSM/GPRS modem that can handle receiving and sending of the text messages.

1.3 Project Scope

In conjunction with the objectives of the project, the scope of the project is defined to provide a basic guideline that enables the research to be conducted within a certain range and depth.

The statements below summarize the scope of the project according to the objectives mentioned earlier:

- ✦ The study and survey of the effects of text messaging in a society.
- ✦ The study and survey of the current manual retrieval of results and modules allowed to be taken the following semester.
- ✦ The study and survey of existing SMS-based college results systems.
- ✦ The study and survey of existing SMS systems.
- ✦ The study and survey of existing third party SMS gateways.
- ✦ The study and analysis of existing SMS API's, SMS development libraries and SMS systems development tools.
- ✦ The study and survey of all the components related to SMS systems.
- ✦ The analysis of the flow involved in storing of student results, how results are delivered to the students and how a student identifies the modules he or she is allowed to take.
- ✦ The study and exploring of how GPA and CGPA in a college are calculated.
- ✦ The study of open source technology and utilizing the technology in system development.

1.4 Problem Statement

The use of SMS to get school results has had a positive effect on college graduates and their relatives for being simple and convenient. It was used by approximately 90% of high school and college entrants to acquire their results in the last summer (Seris, 2010).

SMS-based delivery of college results is an effective and hassle free solution for acquisition of college results for the students. Currently, students in colleges have to queue up at the start of every semester to get their past semester results and know the modules they are eligible to undertake that following semester. This causes a delay in the registration process and perhaps delays in the starting of semester classes.

There is an existence of some SMS-based delivery of college results systems and a number of schools and colleges have implemented such systems. The main problem in such kinds of systems is that most of them are third-party. In that, an independent company may implement such a system whereby it links to several colleges within a specified area, possibly a country. For a student to get results, the student sends an SMS keyword with their ID number to a premium number which incurs extra charges on top of the operator charges for sending a regular SMS. These systems were designed with the aim of making profits through the money they make by adding extra charges per SMS. There is need to use an in-house system in a college that does not intend to make profits out of the service, rather to deliver results to its students and notify them of eligible subjects they can take in a simple, convenient manner. Hence, a proposed system that performs such tasks.

Another issue is that such systems have a limitation when it comes to security and privacy since there is no authentication procedure. Anyone who knows an ID number of a friend or relative can check their results. Most of the systems were designed to deliver college results regardless of who has access to whose results.

Also, since such systems are ran by independent companies, a college may not be able to pass communication to its students concerning different aspects of a college e.g upcoming events, cancellation of classes, library notifications, etc. Hence, the need for an in-house system that may incorporate such features in future enhancements.

1.5 System Limitation

Currently, this system is an independent system and it faces the following limitations when implemented;

- It is independent of any existing university database. Hence, some information like administration, admission, enrollment, finance, resource management will not be integrated into this system.
- As a standard, GSM mobile phone operators only allow mobile devices like a GSM modem or ordinary mobile phone to send 6 to 10 text messages per minute. This means that the system can only reply 10 requests per minute.
- If a GSM network is offline, then the system is not operational since it requires a GSM network to send/receive text messages.
- Unlike paper results, which are stamped, it is difficult to certify SMS results. However, mobile operators provide premium custom numbers that bear the name of the school or college. This way the user can be sure that the message originated from the right source.
- As a standard, the length of an SMS message is 160 characters. This is a limitation in SMS technology. Therefore, the text messages are abbreviated depending on the availability of space.
- The SMS technology does not guarantee set transmission times or guaranteed delivery of the message; therefore some messages may be delayed, blocked, or lost in transmission.
- GSM mobile phone operators might not have coverage in some areas, and some locations may have mobile phone jammers e.g in or around cinema halls, thereby preventing users from getting network signals.
- Not all students own a mobile phone. This is a negligible limitation considering the vast amount of mobile phone users.

1.6 Expected Outcome

This system is designed for actual implementation of an SMS-based results delivery system in colleges and schools.

The system should support access by a pool of HOD's of different departments in a college as well as students.

The useful information from this system will be the student results, which a parent can use to assess his or her child's performance.

1.7 Significance of the System

This is a system with real benefits to be gained from the development of the whole system. The significance of the project can be realized in:

- **Use of open source technologies.**
  The development of this system uses a number of open source technologies like MySQL, Apache. These technologies improve the scalability of the system.
- **Opens up new possibilities in delivery of college results.**
  The design of this system could result in developing newer ways of how colleges deliver semester results to their students.
- **Eases the means of communication between a college and its students.**
  The design of this system makes it easy for a college to pass communication to its students.
- **Parents can follow up on their children's performance.**
  This system will enable parents to follow up on how their children are performing in school in a fast and convenient way without waiting for the students results to be sent to them through mail.
- **Secure, cheap, convenient and fast way for students to get results.**
  This system conveys a secure, cheaper, convenient and fast option of students acquiring their results compared to checking online which requires an internet connection or having to queue at the start of the semester to find out their results.
- **Techniques applied could be used in other systems.**
  Techniques used in the development of this system can be used in development of other related systems. The techniques can be applied beyond the boundaries of an SMS-based delivery of college results.

1.8 Report Organization

This report consists of 8 chapters and its organization is outlined as below:

- Chapter 1: Introduction

  This chapter gives an overview of the system, system motivation, system objectives, system scope and the project schedule.

- Chapter 2: Literature Review

  This chapter contains the literature review of the project. In this chapter, reviews are conducted on all necessary areas that are relevant to the project. It provides study on existing SMS systems. The study includes all the development paths leading to the core technologies used in the project. There is also a review of the development tools that were considered for use in this system.

- Chapter 3: Methodology

  This chapter outlines the methodology used in arriving at this system, the different ways and techniques used to develop the whole system.

- Chapter 4: System Analysis

  This chapter lays out the details of system requirements, including functional and non-functional requirements, hardware and software requirements.

- Chapter 5: System Design

  This chapter describes in detail the system design, database design and user interface design.

- Chapter 6: System Implementation and Testing

  This chapter focuses on the development environment and strategy. It also lays out the testing process, and the various testing techniques.

- Chapter 7: System Evaluation and Conclusion

  This chapter discusses the procedures used to obtain evaluation results regarding the proposed SMS College Results System in terms of effectiveness and efficiency.

- Chapter 8: Conclusion and Future Work

  This chapter discusses problems encountered, system strength, system constraints, future enhancements, and knowledge and experience gained during the development of the system.

Throughout this documentation, some terms may be referred to as below or otherwise;

- The SMS College Results System may be referred to as ‘SCRS’ or ‘SMS-based delivery of college results’.
- SMS (text message), may be referred to as ‘SMS’ or ‘text’ or ‘text message’.
- GSM network operator may be referred to as ‘network operator’ or ‘GSM operator’ or ‘mobile operator’.

2 LITERATURE REVIEW

This chapter covers the study of existing systems. The purpose of this study is to obtain all the features that are necessary to be included in the SCRS system. The study seeks to discover how other related systems perform their tasks, their strengths and weaknesses.

This chapter also takes a look at the latest open source technologies in the Information Technology field. It also covers tools and methods that will be useful for the development of this project.

2.1 Analysis Studies

2.1.1 Overview of current manual way of how students get results

The current results delivery system consists of four main stages. Each of these stages presents a challenge in organization and implementation.

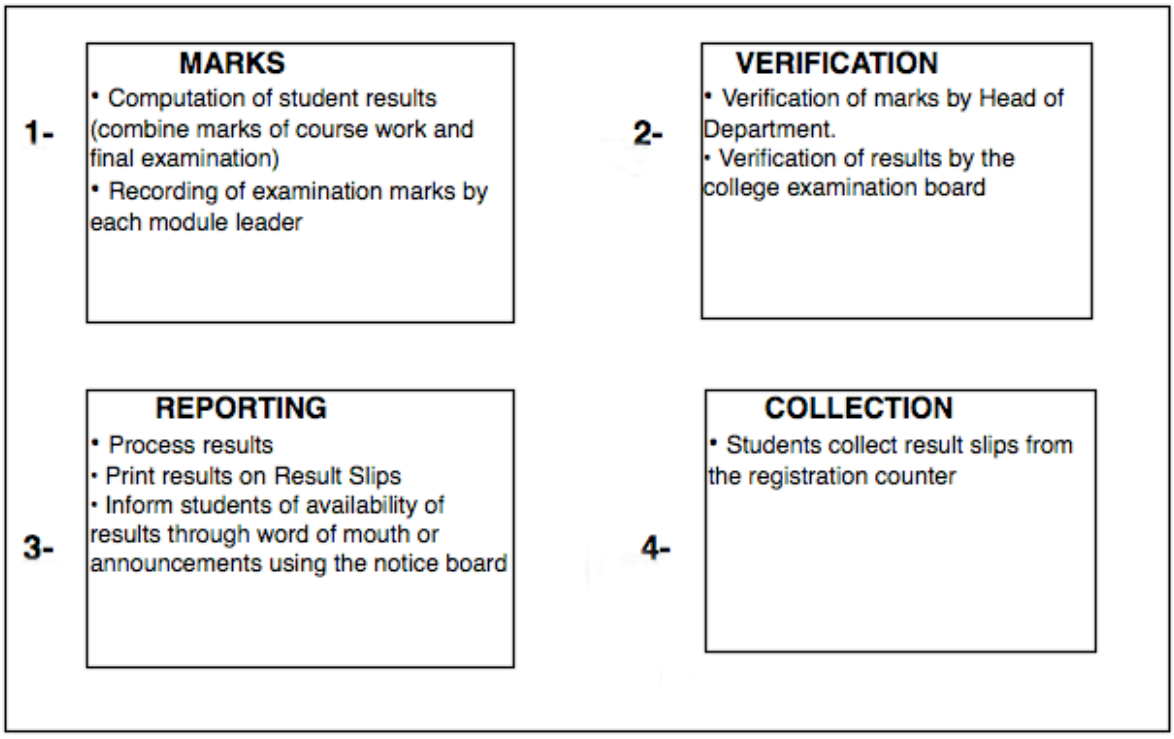


Fig. 1: Four stages in results delivery to students

The first stage involves module leaders (lecturers in charge of a particular module). After course work and final examinations have been marked, the module leaders compute the course work marks and final examination marks (if a module has a final examination) according to the weightage. Then, the results are recorded.

The second stage involves Heads of Departments who verify the marks entered by the module leaders before having a board meeting with the examination board for further verification of students' results.

The third stage involves the examination board who process the results including calculation of grades, print them onto result slips for each student then inform HoD's to inform their students of the availability of results.

The fourth stage involves the students who physically collect result slips during registration for a new semester or at a later date within a semester.

2.1.2 Current Process Flow (Manual)

The current manual system of handling results is that marks are entered in paper forms that are distributed between module leaders, HoD's and the examination board. Some lecturers might transfer these marks into a large master mark sheet. All the marks of a particular department are then bound together.

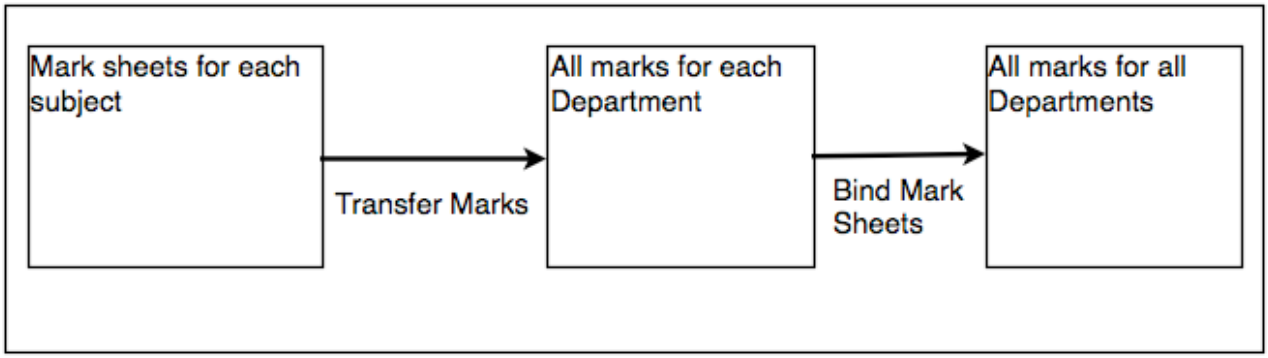


Fig. 2: Classic Results System

Some module leaders might enter marks in individual electronic spreadsheets to aid in the process. Others might use simple database programs like Microsoft Excel, Access to enter marks in their personal computers. As such, they use different file formats and different file structures using different types of programs. This makes it difficult to combine all the data in the various file formats to a form a unified results database.

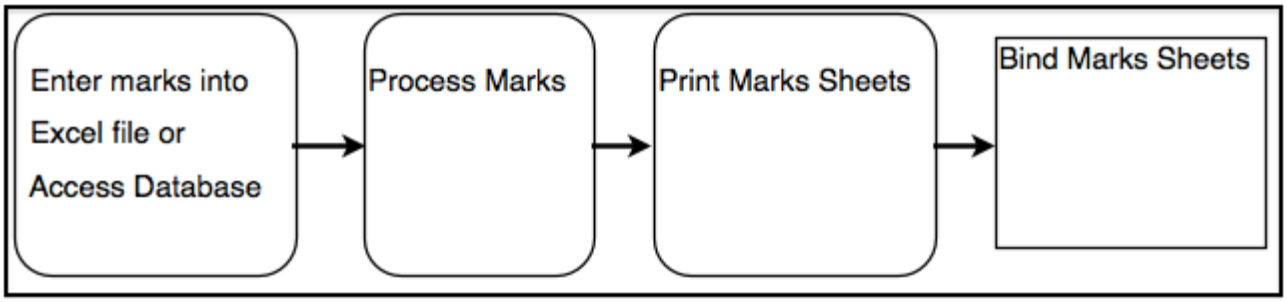


Fig. 3: Semi-Computerized Results System

2.1.3 Case Study: Sailing Race Results System

James Bell describes the Sailing Results System as an MS-DOS application that makes it easy to calculate the results for a dinghy sailing race.

The application runs on Windows platforms; Windows Vista, XP, 2000, NT, ME, 98, 95, 3.1, OS/2 but it requires Microsoft's Virtual PC software to run on Windows 7.

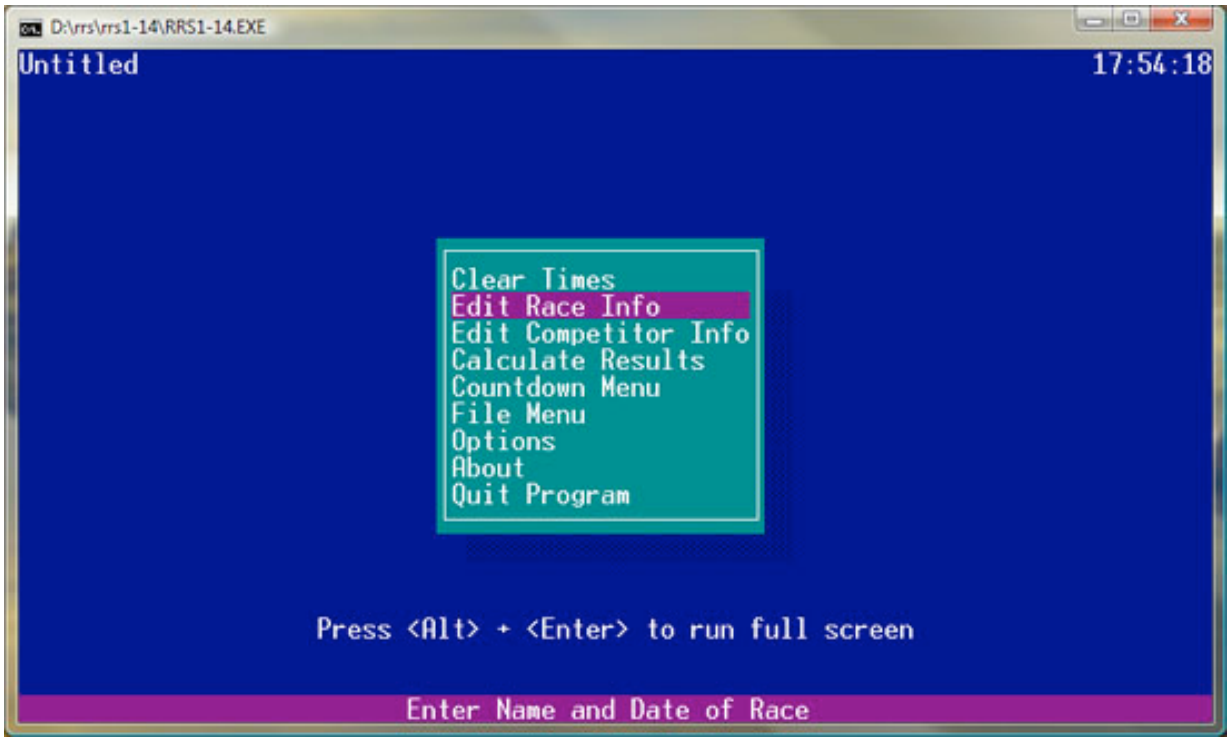


Fig. 4: Sailing Results System Screen Shot

This software is free to use and comparably easy to operate with very minimal hardware requirements (considerably any PC regardless of its age).

It has countdown features with either 5 or 1 minute options with the start time automatically recorded. 'TimeSheet' helps with noting finishing times and pressing the space bar records time on screen. The application also shows the time difference between adjacent results or finishers.

Results from the program can be exported as HTML for the web, or CSV for use with other applications.

2.1.4 Relationship to existing work and SCRS Approach

The results system reviewed in the subsection above somehow interrelates to the SCRS, in that they both deal with results of a particular entity. Though the Sailing Results System discussed does not provide an SMS feature, it is not considered useful due to the nature of the system, but it is still somehow related to the SCRS and its features are further discussed below.

The fact that the Sailing Race Results System has been designed and implemented to run on any version of Microsoft Windows from OS/2 all the way to Windows 7 may look advantageous. However, with the rapid development of today's technology, this means that this system is not able to take full advantage of today's PC features like; multithreading, using the full power of multi-core CPU's, faster memory running on DDR3 motherboards, etc.

Another issue is that the Sailing Race Results System runs as an MS-DOS application. Today's users want to see graphics, otherwise the GUI running on almost all operating systems wouldn't be such a big success. Users invest in LCD and LED monitors, graphics cards, memory, so as to enjoy rich graphics and this system clearly does not offer that. Generally, users would not be satisfied with such crude graphics unless it was made as a special request.

However, implementing such a system (i.e Sailing Race Results System) is cheaper and easier for developers, it would not be fit for today's users. Therefore, the proposed SMS College Results System will take advantage of multi-core CPU's, multithreading promising high performance for the software. It will also feature a GUI consisting of GUI-windows with point-and-clicks and interactive input and output.

2.2 Review on latest Technologies

2.2.1 Client-Server Architecture

In the early history of distributed computing, huge mainframe computers did most of the processing of data and functionality, handling multiple users. Dumb terminals were connected to access the resources on the mainframe. As desktop computers became cheaper and increased greatly in processing power, some of the processing was done on these terminals, later referred to as clients. The mainframe became the “Server”.
This led to the client-server model of computing, that involves a central server handling the database functionality. The ‘fat-client’ handles the user interface, and some of the business logic, requesting data access from the server.

However, there were problems with this architecture, especially when maintenance of the clients involved installing new versions in each client. With the advancement of the world wide web, ‘thin clients’ came into place, where the only software required is an internet browser. The processing is now distributed into a multi-tier architecture at the server level.

2.2.2 3-Tier Architecture

Three-tier architecture is a special type of client-server architecture consisting of three well defined and separate processes, and they could be running on different platforms.
The three tiers consist of:
Client tier: Client tier runs on the user’s computer and provides user interface.
Application-Server tier: Application server tier is the functional module that actually processes data. This middle tier is not present in a 2-tier architecture. This tier protects the data from direct access by the clients.
Data-Server tier: Data server tier is a Relational Database Management System (RDBMS) that stores the data required by the middle tier.

2.2.3 .NET Framework

The .NET platform consists of the following parts:

- The .NET framework.
- Visual Studio.NET, which creates .NET solutions using VB.NET, Managed C++, and C#.

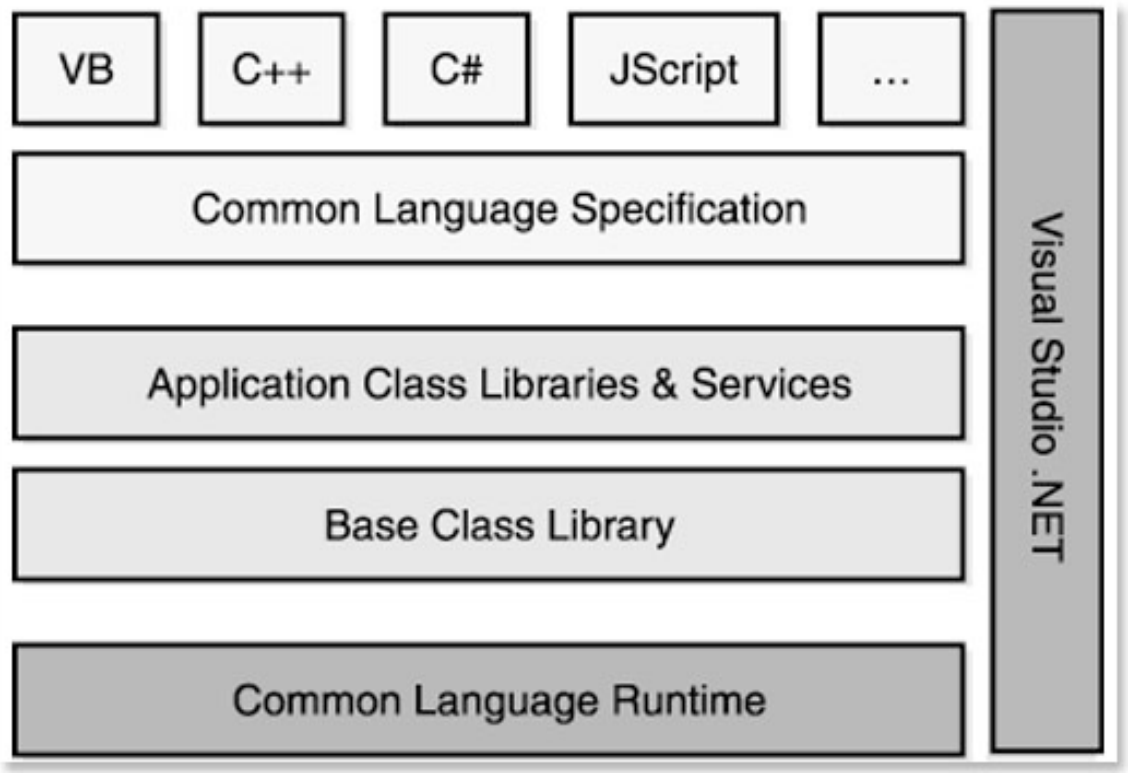


Fig. 5: The Microsoft .NET Platform

The .NET Framework is a Microsoft technology that consists of the following features:

- The Common Language Runtime (CLR) – hosts all .NET code and provides core services. It is a managed environment that executes all .NET programs and functions as the execution engine. This is similar to the Java Virtual Machine (VM). Code that runs within a CLR is referred to as ‘managed’ code.
- The Common Type System (CTS) and the Common Language Specification (CLS) – CTS fully defines all possible data types supported by the runtime. CLS is a subset of CTS that must be supported by all .NET languages.
- The Base Class Libraries – an object-oriented collection of reusable classes. These include ASP.NET, ADO.NET, web applications and web services classes.
- Microsoft Intermediate Language (MSIL) – all code written in .NET languages compiles to a platform-neutral code called MSIL. It is a binary intermediate form for a .NET program. This is similar to Java byte code.

- Assemblies – the fundamental unit of deployment, version control, reuse, activation and security.

2.2.4 GSM

GSM is a digital mobile telephone system that is widely used in the world. It operates at either the 900 MHz or 1,800 MHz frequency band.

2.2.4.1 GSM Operators (Prepaid/Postpaid)

Suruhanjaya Komunikasi Dan Multimedia Malaysia (Malaysian Communications and Multimedia Commission), the communications governing body of Malaysia has several numbers of mobile operators registered with it. However the 3 major GSM operators currently serving the Malaysian public are; Maxis, DiGi and Celcom. Though all the 3 operators offer postpaid services, they also offer prepaid services for the general public. It would be more practical for an organization planning to implement an SMS system to opt for a postpaid subscription, but for the sake of this project, a prepaid subscription would be sufficient. As of Wednesday, October 27, 2010 the local GSM operators prepaid rates are as below:

- **DiGi** - Voice Calls: 36 sen/min, SMS: 15 sen/SMS.
- **Celcom** (offers prepaid service as XPax) - Voice Calls: 38 sen/min, SMS: 10 sen/SMS.
- **Maxis** (offers prepaid service as Hotlink) - Voice Calls: 36 sen/min, SMS: 10 sen/SMS.

After conducting research and obtaining the above results, it became clear that Maxis (Hotlink) is the choice of GSM operator whose SIM-card will be used to provide the SMS features of the SCRS system.

2.2.4.2 GSM/GPRS Modem

A GSM modem is a special type of modem that uses a SIM card, and operates over a subscription (prepaid or postpaid) to a mobile operator, just like a mobile phone. A GSM operator detects a GSM modem like an ordinary mobile phone.

A GSM modem could also be a standard GSM mobile phone with the cables and software driver to connect to a USB port on a computer and one that also supports the "extended AT command set" for sending and receiving of SMS. However, not all mobile phones support this type of modem interface.

While using a GSM mobile phone as a modem may be a good way to set up a system with SMS capabilities, the best solution is to use a dedicated GSM modem device. That is why for the SCRS system, the GSM modem below, ***Wavecom WMOD2***, is to be used. According to the user manual, it has these specifications:

- Dual Band GSM 900/1800 MHz.
- Voice, SMS, fax, and data capabilities.
- GPRS and WAP compliant.
- SIM with sliding SIM card holder.
- Power supply.
- Open Software Platform (support for embedded user applications).
- Interfaces - RS-232 and audio through mini sub-D 15-pin connector.
- Supporting: - Remote control by AT commands (GSM 07.07 and 07.05). Baud rate from 300 to 115,200 bits/s.

Fig. 6: Wavecom WMOD2 GSM Modem

2.2.5 SMS

The Short Message Service (SMS) allows text-based messages to be sent to and from mobile telephones on a GSM network. A text message has a maximum length of 160 characters and SMS messages are divided into two categories: Mobile Terminate (where the SMS originates from the network provider) and Mobile Originate (where the consumer can send messages to other consumers).

2.2.5.1 SMS Gateway

An SMS gateway is a way a GSM modem or GSM mobile phone connects to different SMSCs (Short Message Service Centre). It translates one SMSC protocol into another, so different GSM operators can interconnect their SMSCs to enable the exchange of SMS messages.

2.2.5.2 HyperTerminal (AT Commands)

According to Microsoft Corporation, HyperTerminal is a program that can be used to connect to other computers, Telnet sites, bulletin board systems (BBSs), online services, and host computers, using either a modem, a null modem cable or Ethernet connection.

It can also be used to send SMS messages using a GSM modem connected to the computer. This is capable by using AT commands. AT commands (ATtention) has the following commands.

- AT+CMGD Delete message
- AT+CMGW Write message to memory
- AT+CMGS Send message
- AT+CMGR Read message
- AT+CMGL List messages

These are just a few of the commands and we won't go into details since the SCRS system will use an API(Application Programming Interface) to implement the communication with the modem. This will prevent the use of actual AT commands in the system's source code, instead an open source API called **GsmComm** will be used. This provides a set of .dll files that can be added to MS Visual Studio and used in any .NET application. This will speed up the development process since a developer does not need to study AT commands.

2.2.6 Application Server

An application server is a server program in a computer in a distributed network that provides the business logic for an application program. The application server is frequently viewed as part of a three-tier application, consisting of a GUI server, an application (business logic) server, and a database and transaction server.

2.2.6.1 Internet Information Services (IIS)

This is a Microsoft's web server that runs only on Windows platforms. It is an enterprise-level Web Server that is included from Windows 2000 onwards. It supports the Windows GUI and provides integration with other Windows mechanisms, such as Active Directory, MS SQL Server and Windows Security services.

2.2.6.2 Apache

The Apache Software Foundation maintains Apache Web Server. It is currently the most popular Web server because it is stable, reliable, efficient and portable. It is an open source product and runs on UNIX, Linux and Windows platforms. The source code for Apache is freely available. Apache can be managed through a PHP file or other scripting language or through a built-in GUI if a package has been installed; like WAMP for Windows, XAMP for Linux and UNIX and MAMP for Mac OS. This is the choice of web server for this project since it is open source and runs on multiple platforms.

2.2.7 Operating Systems

The following Operating Systems were reviewed.

2.2.7.1 Microsoft Windows

This is a set of commercial operating systems offered by Microsoft Corporation, and the latest products include Windows Vista and Windows 7. Windows is a menu-driven operating system that uses a GUI as its main method of communication with the user. Its user interface is user friendly and it is the most popular operating system.

2.2.7.2 Ubuntu Server

This is a free, open source operating system that is gaining widespread acceptance in the industry. It is originally based on the Debian GNU/Linux distribution. This operating system is more scalable than the Windows operating system. Ubuntu is currently funded by Canonical Ltd.

2.2.7.3 Apple Macintosh

This is a set of commercial operating systems offered by Apple Inc, and the latest products include Mac OS 10.6 (Snow Leopard) and the soon to be released Mac OS 10.7 (Lion). Compared to Windows OS, Mac OS has a much better user experience in terms of its GUI and much better security since it UNIX-based. It does not get infected by viruses and rarely crashes. This is the choice of operating system to run the SMS College Results System's web server and the database due to its reliability and security factors.

2.2.8 Database Server

A database engine is the core process that a database management system uses to store and maintain data. It serves two functions. The first is to store information, and the second is to process requests for stored information. The following Database Servers were reviewed.

2.2.8.1 MySQL

This database server will be used in the SCRS system.
MySQL is a multi-user, multithreaded RDBMS server that uses SQL to interact with and manipulate data. This is an open source relational database product. It can be downloaded for free. It is a high performance and scalable RDBMS. It has an in-memory query results cache that contributes to its performance. It works best when managing data and not executing transactions. Its features are:

- Multithreaded capability that enables the database to perform multiple tasks concurrently.
- Support for various programming languages: C, C++, C#, Java, Python, Perl, and PHP.
- It runs on just about any operating system. Specifically, it runs on Windows, Linux and UNIX.
- The ability to handle large databases.
- The ability to access tables from different databases by using a single query.

2.2.8.2 Microsoft SQL Server

SQL Server is a client-server database management system. Unlike file-based databases, client-server database engines manage read-write operations to the database. Because of this, client-server databases such as SQL Server can handle many more concurrent users and more amount of data.

SQL Server is a server-based database system. Requests for information are processed on the server and only the resulting data is sent back to the client application. This reduces network traffic; it also allows users to access data and experience high application performance without a need for a high-end client workstation.
SQL Server runs only on the Windows platform, which restricts its portability.

2.2.8.3 Oracle Database

This is a very reliable database product from Oracle Corporation, which is the current industry leader in database technology. There are more programmers who can program Oracle DBMS than any other. Oracle database comes in three editions: Enterprise, Standard and Personal. Using this database requires a lot of configurations, which are beyond the scope of this project.

2.2.9 Database Connectivity

Databases could be accessed using the following technologies.

2.2.9.1 Microsoft ADO.NET (Activex Data Objects)

This is the primary relational data access model for Microsoft .NET-based applications. It is multi-tier (code could be easily moved from tier to tier), disconnected, XML-based, scalable and fast. There is no Recordset. Instead, data is manipulated using DataSets. There are four classes that are used to read and write data from data sources:

1. Connection - Connect to data source.
2. Command - Execute stored procedures.
3. DataAdapter - Connects DataSet to database.
4. DataReader - Forward/only, read/only cursor.

2.2.9.2 MySQL Connector/NET

Connector/NET enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET tools. Developers can build applications using their choice of .NET languages. Connector/NET is a fully managed ADO.NET driver written in C#.

Connector/NET has the following features:

- Features provided by MySQL Server including MySQL Server version 5.5.
- Large-packet support for sending and receiving rows and BLOBs.
- Protocol compression which enables compressing the data stream between the client and server.
- Support for connecting using TCP/IP sockets, named pipes, or shared memory on Windows.
- Support for connecting using TCP/IP sockets or Unix sockets on Unix.
- Support for the Open Source Mono framework developed by Novell.
- Fully managed, does not utilize the MySQL client library.

This is the choice of Database connector for the SCRS system to go hand-in-hand with the MySQL database.

2.2.9.3 Open Database Connectivity (ODBC)

A group of companies designed ODBC, including Microsoft, Lotus, Oracle and IBM. It was one of the first attempts to create a generic way of talking to different database engines. However, it was built to access relational data from relational databases. It is accessed via a straight-C API.

2.2.10 Programming Languages

The following languages were reviewed.

2.2.10.1 C#

This is the language of choice for the .NET platform, as it is closely tied to the .NET CLR. It is an object-oriented language that closely resembles Java and C++. C# was designed in response to the strengths and weaknesses of other languages, in particular Java and C++. There are a few new syntax elements in C# that are not available in C++ or Java, namely properties, indexers and attributes. Properties give support to *get()* and *set()* methods. Indexers are a way to treat an object as an array. Attributes support declarative programming.

Features of C# are:

- Component orientation. This makes component building easy, with language constructs such as properties, events and attributes.
- Automatic and manual memory management. The programmer does not have to dispose objects, eliminating memory leaks and circular references.

Due to its important features, this is the programming language of choice for the SCRS system. It will use the MySQL Connector/NET to make connections to the MySQL database and retrieve or manipulate data.

2.2.10.2 Java

Java is an object-oriented language that is well suited to designing software for use over the Internet. It was developed at Sun Microsystems in 1991. It is a cross-platform language that can be used in many platforms, including Windows, Macintosh, Linux and most versions of UNIX, including Solaris. The language is quite complex and includes many disparate open source technologies. It is often run as an applet in internet browsers, which can also act as a SOAP (Simple Object Access Protocol) Client.

2.2.10.3 C++

In the .NET platform, C++ is used to program .NET, which performs better than XML Web Services. This is the language used in developing high performance ATL Server Web Services. The ATL Server uses C++ code that is compiled at development time before being deployed on the server. The ATL Server architecture runs on top of Internet Information Server (IIS).

3 METHODOLOGY

3.1 System Development Flow

After researching and retrieving feedback from lecturers' and students' requirements, a complete development flow was developed. Then, research was also done on findings about the suitable programming language, technology and application servers that are essential in developing the SMS College Results System.

The further procedure is configuring Apache, MySQL, as well as the .NET framework in the deployment machine.

During the development cycle, the system will be discussed to the project supervisor from time to time. The reason is to gain feedback and further enhancement from the supervisor.

Validations are done in many fields to ensure a user keys in data correctly and the system is able to process the data in a correct manner. The process of verifying the accuracy and correctness of data retrieval, data entry, front end displaying, etc have been closely analyzed. In addition, user friendly ability and other aspects have been verified as well.

Finally, upon the system's consideration as being stable and fulfills all mandatory functionalities, the system is transferred for a pilot run mainly for testing purposes. This is an important step to boost up the system's stability, robustness, user-friendliness and discover new hidden bugs.

3.2 Software Methodology

Methodology covers the methods used to discover past, existing and forthcoming new technologies in researching and developing of a new project. It is a collection of tools and techniques used to aid system developers in the development of a new system, including the proper procedures used to study existing systems with the aim of eliciting improvements to the current technology.

3.2.1 Techniques Used to Define Requirements

The techniques used to determine requirements for the SCRS System include the following:

- Analysis of existing systems
- Review of new technologies
- Library Research
- Internet Research

3.2.1.1 Analysis of existing systems

Existing systems were studied online to determine the features that they have and those that are missing.

3.2.1.2 Review of new technologies

New technologies in the software engineering world were analyzed before choosing the best and cheapest to be used in the completion of this project.

3.2.1.3 Library Research

The library at the Nilai University College stocks many books of great value to the understanding of some of the technology used to develop this project. Books were borrowed for study purposes, resulting in a change in some of the requirements of this system.

3.2.1.4 Internet Research

The internet has been helpful, whereby discussions with professionals in software engineering forums assisted in the identification of other requirements for this system. Also, search engines provided some results involving requirements for successfully implementing an SMS system.

3.2.2 Software Process Model

A software process consists of the activities and associated information that are required to develop a software system. An organization has its own specific software process but all these individual approaches usually follow some more abstract generic process model. (Sommerville, March 1996.)

A software process is, therefore, a method of developing or producing software. These include:

- The Waterfall Model.
- The Incremental Model.
- Evolutionary development.
- Formal systems development.
- Reuse-based development.
- Spiral Process Model.
- Object-oriented Software Process (OOSP) Model.

3.2.3 Choice of Incremental Development Model as Methodology

After reviewing some of the popular methodologies, which are currently being implemented widely, it was decided to choose Incremental Development Model as the methodology for SCRS. The incremental model is iteratively applying the waterfall model.

In this approach, development is organized into a series of short, fixed-length mini-projects called iterations; the outcome of each is a tested, integrated, and executable system. Each iteration includes its own requirements analysis, design, implementation, and testing activities.

With the incremental development, it will partition a system into subsystems by functionality. An early release starts with a small, functional subsystem, later releases will add addition functionality. Each incremental is a partial product, not a prototype. It will contain more functionality than the previous increment, but build on the previous increment. Increments progress parallel with the product delivery of one increment feeding into the design phase of the next. This model provides the user with early feedback, allowing alterations of the project to occur with limited impact. Each increment is functionally a more complete version of the product.

The incremental development model for SCRS system project comprises the following phases:

- **Requirements workflow**

The first phase of this development involves the Requirements Specification, which is the usual starting phase of all the software process models. These requirements are described and defined in detail, serving as the system specification.

The full details of the requirements specification are described in Chapter 4 (System Analysis). These involve the functional requirements and the non-functional requirements. The methodology used in gathering these requirements involves several ways of requirements discovery.

The software requirements specifications shall be documented using Use Case diagrams. The documentation shall use the implementation specific terms for the ease of understanding on the development side. One user requirement may be fulfilled by one or more software requirements specifications.

- **System Analysis and Design Workflow**

The next stage involves identifying user requirements. User requirements are prioritized and the highest priority requirements are included in the early increments. In this phase, the use cases are identified, the deployment and component diagrams are arrived at and classes and the collaboration among the classes are defined. The system design document shall be prepared. The class diagrams, sequence diagrams, state transition diagrams and collaboration diagram are included.

Components required for setting up the SCRS system are identified. The components that need to be developed are identified as follows:

- Design and implementation of a database management system
- Design of user client interfaces
- Design of user manager module
- Design of SMS output format

- **Implementation Workflow**

In this phase, the actual coding shall be done according to the programming standards. The code shall be unit tested. The programming standards to be used shall be identified. Coding shall be done according to the programming standards and class specifications. This is one of the most critical phases as failure in coding will result in collapse of the whole project. New skills are required, which have to be learned or developed. While system analysis and design workflow identifies the components that need to be developed, this phase develops and implements all the design requirements identified in the earlier phases.

The development involves the implementation of SCRS. This phase involves the installation of .NET Framework (4.0), the MySQL server, Apache server and the programming environment, which is the Visual Studio 10 (MSDEV), etc. To implement this phase, new skills have to be developed. Specifically, this means learning the C# language, the .NET Framework and how it implements SQL functionality.

- **Testing Workflow**

In the system testing phase, the product shall be tested module-wise and the interdependencies among the modules shall be validated. The functionality of the product shall be tested as a whole. The product needs to be tested for conformance with the system requirements as well as assessed as to its suitability for the task of processing student results and determining of subjects offered the following semester, and performing of preregistration via SMS, that is, the overall goal of the project.

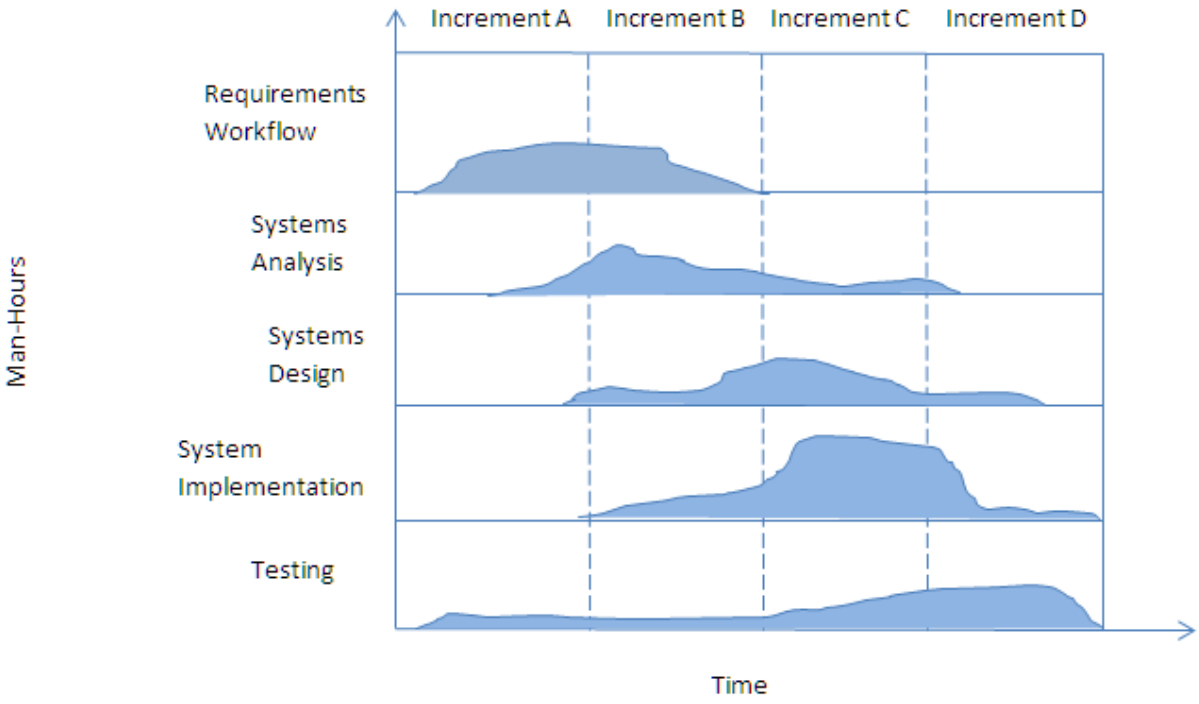


Fig.7 – Incremental Development Model

The iteration and incrementation are used in conjunction with one another. There is no single requirement phase or design phase. Instead, there are multiple instances of each phase. The number of incrementals will vary and all the workflows are performed over the entire life cycle.

The incremental development model is beneficial in such a way that a developer takes advantage of what was being learned during the development of earlier, incremental and deliverable versions of the system and the project complexity is reduced.

4 SYSTEM ANALYSIS

4.1 Requirement Analysis

Requirement Analysis is the detailed documentation of the system and constraints. These requirements should be precise and may serve as a contract between the system end user and the software developer. The use case analysis found herein is a product of much discussion regarding system requirements with Ms. Harlina Harun (Project Supervisor) as acting client. These are loosely divided into functional requirements and non-functional requirements.

4.2 Functional Requirements

The following use cases and actors were found to exist in the problem situation. Each use case is described in detail in their respective module section. The use case diagrams that follow up model the desired behavior of the system.

Functional requirements of the SCRS can be categorized into four main sections:

- Registrar section
- HOD section
- Lecturer section
- Student section

4.2.1 Registrar Section

This module enables the registrar to manage different personnel; students, lecturers, hod's, deans and other attributes that make up a college; schools, departments and modules.

The SCRS system is meant to get data like students, lecturers, schools, departments, etc from an existing college system. This means that this system would have to interface with other systems in a college e.g.; "Student Registration System" - for students' details, "Staff Registration System" - for lecturers', hod's and deans' details and other preexisting systems that manage the schools, departments, modules, etc.

Since this system is not interfaced with the above mentioned systems and for demonstration purposes, the SCRS system had to handle all those functionalities by itself.

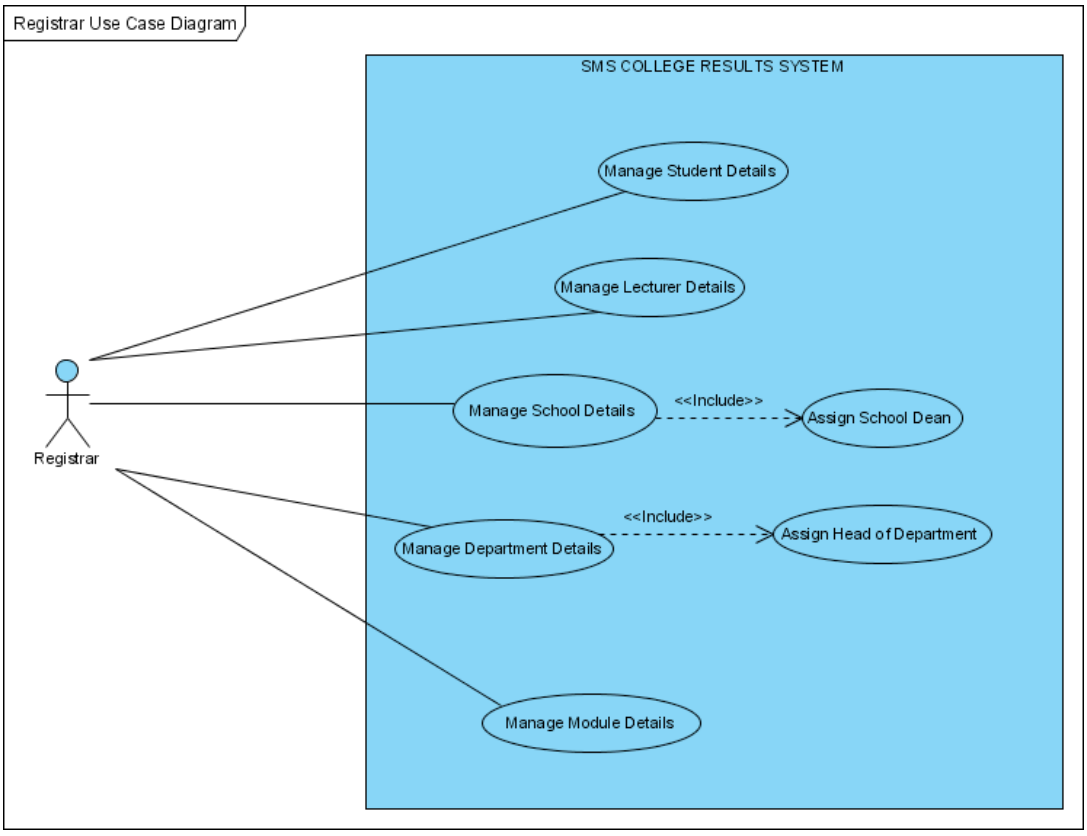


Fig. 8: Registrar Use Case Diagram

Actor	Description
Registrar	The system user who manages personnel and other day to day running of a college.

Table 1: Registrar Section Actor Description

Use Case	Description
Manage Student Details	This includes adding of new Students details to the system, viewing Students details in the system, deleting Students and editing Students details. Student details would be got from a “Students’ Registration System” if SCRS system was interfaced with that kind of a system.
Manage Lecturer Details	This includes adding of new Lecturers details to the system, viewing Lecturers details in the system, deleting Lecturers and editing Lecturers details. Lecturer details would be got from a “Staff Registration System” if SCRS system was interfaced with that kind of a system.
Manage School Details	This includes adding of new Schools details to the system, viewing Schools details in the system, deleting Schools and editing Schools details. A School must have a Dean. A Dean is got from a list of Lecturers who are neither deans already nor HOD’s.
Manage Department Details	This includes adding of new Departments details to the system, viewing Departments details in the system, deleting departments and editing departments details. A Department must belong to a School. A Department must have a Head. A Head of Department is got from a list of Lecturers who are neither HOD’s already nor Deans.
Manage Module Details	This includes adding of new Modules details to the system, viewing Modules details in the system, deleting Modules and editing Modules details. A Module must belong to a Department.

Table 2: Registrar Section Use Case Description

4.2.2 HOD Section

This section enables the Head of Department to;

- Register a student to a module, deregister a student from a module.
- Add module to be offered the following semester.
- Remove module from being offered the following semester
- View all modules to be offered the following semester.
- Remove all modules from being offered the following semester.

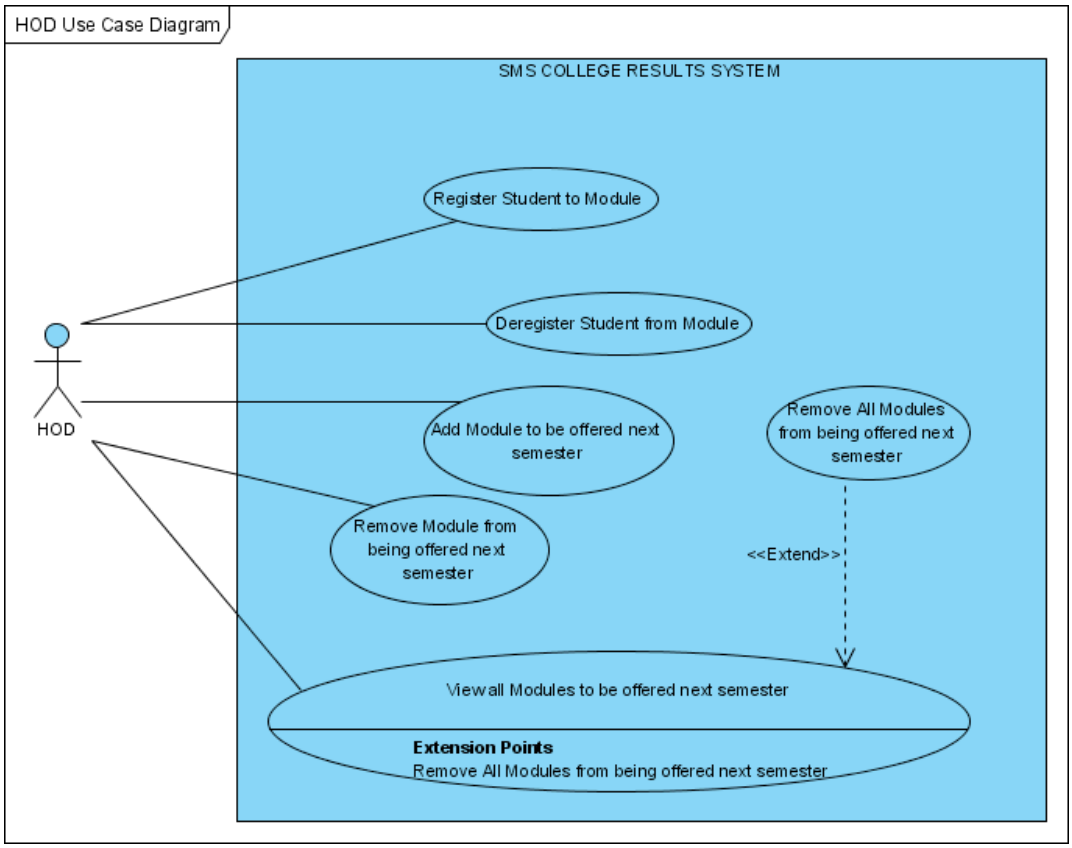


Fig. 9: HOD Use Case Diagram

Actor	Description
Head of Department	The Head of Department is responsible for the day-to-day running of a department. Different colleges have different departments according to the courses that they offer.

Table 3: HOD Section Actor Description

Use Case	Description
Register Student to Module	Without registering a Student to a Module, the Student's Results cannot be entered. Registration of a Student to a Module is done by this function.
Deregister Student from Module	A Student who has dropped a Module is Deregistered using this function.
Add Module to be offered next semester	This use case enables the HOD to set modules that will be offered the coming semester. This will enable students to preregister using SMS.
Remove Module from being offered next semester	This removes a previously added module from the Modules to be offered next semester list.
View all Modules to be offered next semester	Used to view all Modules that will be offered the upcoming semester. From this use case, an HOD may choose to delete all modules to be offered the coming semester. This may come in handy when a new semester has started and new modules are to be added.

Table 4: HOD Section Use Case Description

4.2.3 Lecturer Section

This section enables a lecturer to;

- Add a registered student's results.
- View previously added results.
- Edit previously added results.
- Delete a student's results.
- View all students' results.
- Delete all students' results.

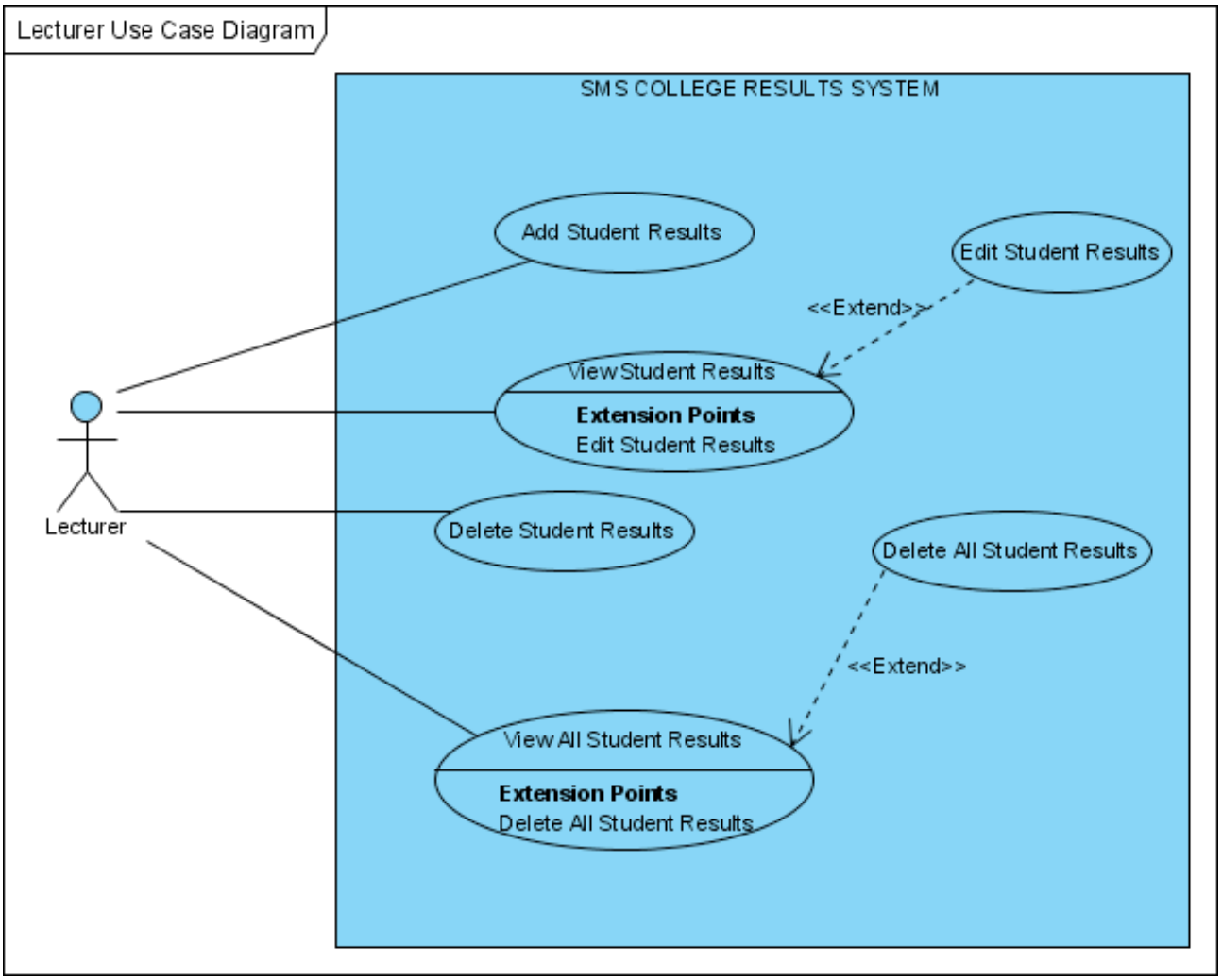


Fig. 10: Lecturer Use Case Diagram

Actor	Description
Lecturer	The lecturer is responsible for the tutoring of an assigned group of students. A particular lecturer is also responsible for the results of the students in his/her class.

Table 5: Lecturer Section Actor Description

Use Case	Description
Add Student Results	A student's results which will be available via SMS are added using this use case.
View Student Results	Student's results can be viewed by specifying the School, Department, Module and all students who have results and belong to that particular department will be available and their results can be viewed by selecting a particular student's ID.
Delete Student Results	Student's results can be deleted by a lecturer. This is useful so that new results are added. Otherwise, old and new results will both be available via SMS. However, this does not mean all results are deleted. There is another copy of a results table in the database that stores permanent results to be used to compare the student's prerequisites when performing preregistration via SMS.
View All Student Results	This enables the viewing of all the results in a table form. All the results may also be deleted with the click of a button.

Table 6: Lecturer Section Use Case Description

4.2.4 Student Section

This section enables a student to perform the following actions all through SMS;

- Check Results.
- Check SCRS Service description.
- Check modules a Student may take the following semester.
- Preregister.

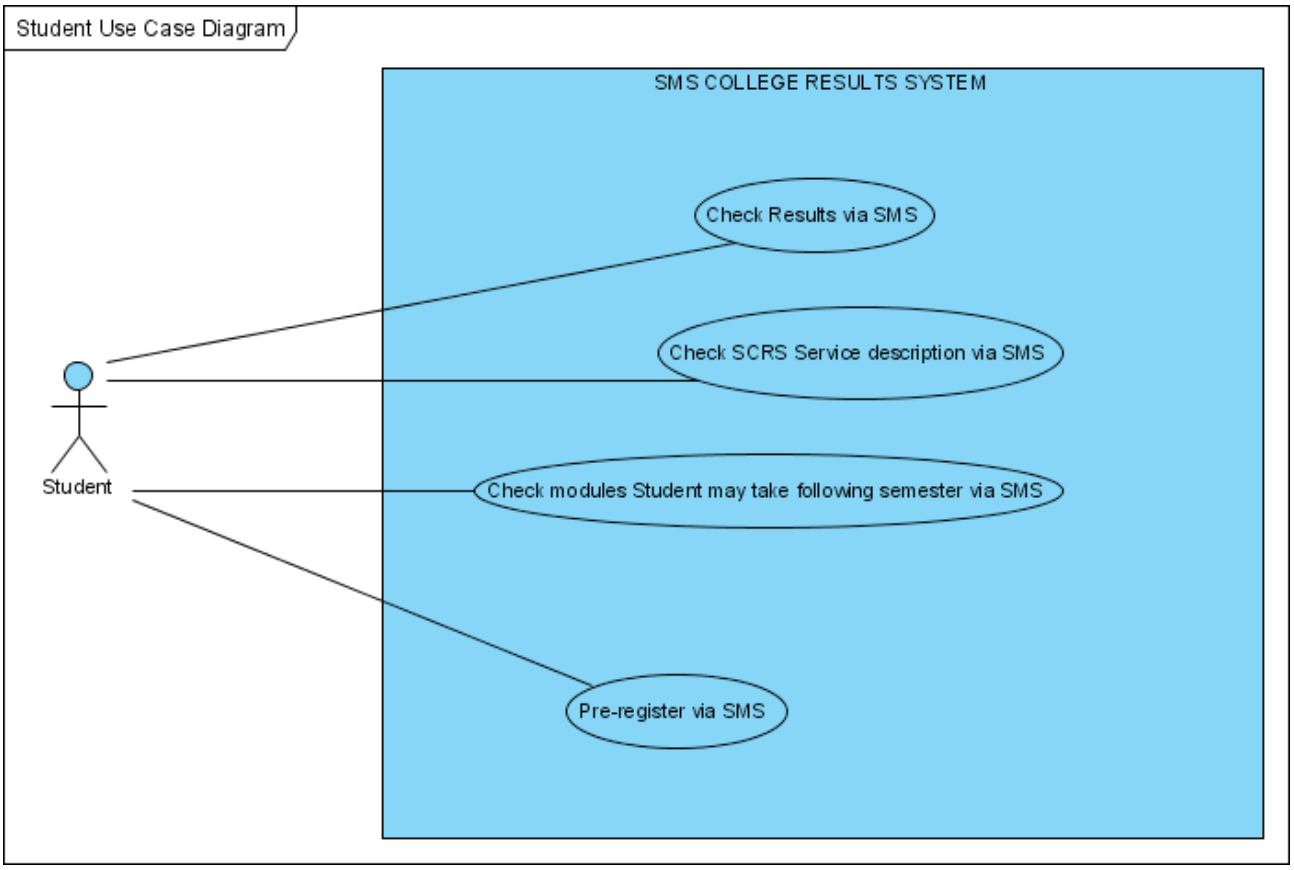


Fig. 11: Student Use Case Diagram

Actor	Description
Student	A student is anyone who is registered in the college as a tutee and attends classes taught by lecturers. A student must have results at the end of each semester unless otherwise.

Table 7: Student Section Actor Description

Use Case	Description
Check Results via SMS	By sending a keyword: RESULTS followed by the student's ID and student's IC/Passport No., a student is able to receive the previous semester's results replied to his/her SMS. Format is as below: RESULTS 708133903 AB071043
Check SCRS Service description via SMS	By sending a keyword: SCRS , a student or any other individual equipped with an SMS-enabled mobile phone is able to get the description of the SCRS, what the system performs, services it offers, etc. Format is as below: SCRS
Check Modules Student may take following semester via SMS	By sending a keyword: SUBJECTS followed by the student's ID and student's IC/Passport No., a student is able to receive the upcoming semester's subjects that he/she may study replied to his/her SMS. Format is as below: SUBJECTS 708133903 AB071043
Pre-register via SMS	By sending a keyword: PREREG followed by the student's ID and student's IC/Passport No., a student is able to perform preregistration for the upcoming semester's subjects that he/she may study replied to his/her SMS. This may be useful in a college where a number of students who may take a subject in the upcoming semester is to be known prior to allocating resources for that particular subjects. Resources like; lecturers, time, rooms, etc. Format is as below: PREREG 708133903 AB071043

Table 8: Student Section Use Case Description

4.3 Non-functional Requirements

Non-functional requirements do not directly govern the specific functions to be delivered by the system. They relate to certain performance and reliability issues, as well as the constraints under which the system must operate.

4.3.1 Product Requirements

- **Platform** - The system shall run on the Microsoft Windows platform with .NET Framework 4 or higher. This should be necessary for consistency of user interface in the initial implementation of the system.
- **Reliability of database** - The system should use a reliable database management system. All database updates must be dependable. As such, the database should be relational and use the current form of database access methods.
- **Robustness** - The system shall be built with robust error recovery routines to handle system failures. The system should be able to guide the user out of a situation where an error occurs. This is important to avoid human help support which is costly.

4.3.2 User Interface Requirements

The system must provide a user interface that is intuitive and does not cause confusion or ambiguity to the user. It is also preferable that the users can expect what the system should be able to provide, and provide those expectations. The users have used similar systems before. Therefore, the users should be able to easily find the same features without difficulty.

In addition, the user interface should meet the following requirements:

- Consistency of User Interface between various modules. Each module should be designed with a similar look-and-feel, with similar type of buttons and icons.
- User Familiarity based upon previous software systems. The user should know the capabilities of the system as a result of using previous systems, and expect at least those features which were previously present. The system can then provide newer features.

4.3.3 Organizational Requirements

- **Reliability of access** - The system should be reliably accessed over SMS. It can be assumed that the hardware enabling the SMS capability is dependable.
- **Conformance** - The database to be used shall closely conform to or can be easily integrated into the current database system used by the college.
- **Scalability** - The system shall use a database that is scalable.

4.3.4 External Requirements

- **Interoperability** - Assuming there are many systems already in use in the college. The system should provide features to enable data to be transferable between these systems.
- **Satisfaction** - The system should make the students, lecturers of the college satisfied, which leads to increased productivity.
- **Modularity** - The system should be able to operate on its own. It is independent of all other software systems except the underlying operating system.
- **Affordability** - The system should use technology that is fully open source. There is no need to use high-end technology, as long as it performs the tasks well.

4.4 System Tools Requirement

4.4.1 Hardware Requirements

Hardware requirements for SCRS are categorized into three:

- **Windows Machine** - This is the machine that will run the backend SMS manager. Minimum requirements are; 512MB RAM, 1GB Hard disk space, 1.5GHz CPU.

- **Mobile Phone or GSM Modem** - This is to enable the SMS capability of the SCRS system. Though for demonstration purposes a Wavecom WMOD2 GSM modem is used, any standard mobile phone with SMS capability connected to the machine running the backend SMS manager of the SCRS will do just fine.

- **Any other machine capable of running a database** - This is to run a MySQL database to make it available to different computers that may be running the Data Entry client of the SCRS system. Minimum specifications may be considered to be similar to those of the clients running the Backend Data Entry application of the SCRS system.

4.4.2 Software Requirements

To be able to run the SCRS system successfully, the following softwares are needed to be present in the computer;

- .NET Framework version 4 or higher.
- MySQL.data library
- .dll files of the GSMComm Library should be in the same folder as the SCRS (.exe). These are the libraries that were used to develop the SMS server.

4.4.3 Development Tools

The following tools are to be used in the development of the SCRS system;

- MySQL Community Server 5.5.8
- Navicat Premium version 9.0.4 (for Mac)
- Microsoft Visual Studio 2010 Ultimate version 10.0.30319.1
- Visual Paradigm for UML Enterprise Edition version 7.0
- GSMComm Library for .NET environment

5 SYSTEM DESIGN

System design is a phase of transforming the system requirements into solutions or system characteristic. Design of the system takes place at two levels: system design and detailed design. System design is concerned with the overall architecture of the system. Detailed design is concerned with designing individual components to fit this architecture.

In an object-oriented system, the detailed design is mainly concerned with the design of objects (class design). These two levels of design are described briefly in the following section.

5.1 System Architecture

SCRS will be using a three-tier system architecture design. The conceptual architecture of the three-tier application applies when an application is split across three tiers. The system is split into three logical components: user interface, computational logic and data storage. In reality, the three-tier SMS applications generally consist of a mobile phone UI for the user interface, a server connected to a middle tier application and a persistent storage that is frequently a relational database.

The main purpose of having three-tier architecture is to assign main functionality to each tier to ensure no function is overlapped. Different people could handle each tier using different languages. Therefore, whenever there is an error or a system fault occurs, the problems can be detected and fixed easily without interfering with other tiers.

5.2 System Design

System design reflects the overall architecture of the system. Design of the SCRS system is split into two separate entities: SMS College Results System Data Entry (Backend Data Entry) and SMS College Results System SMS Management (Backend SMS Manager). These are described more below:

5.2.1 SMS College Results System Data Entry (Backend Data Entry)

As mentioned earlier in the System's Analysis phase, the SCRS system has no linkage to other systems to get the required data like results, modules, departments, lecturers, student details, etc. Therefore, without data, the SMS functionality would face a disability. A backend data entry system had to be designed to cater for saving of results, student details, etc to be accessed via SMS.

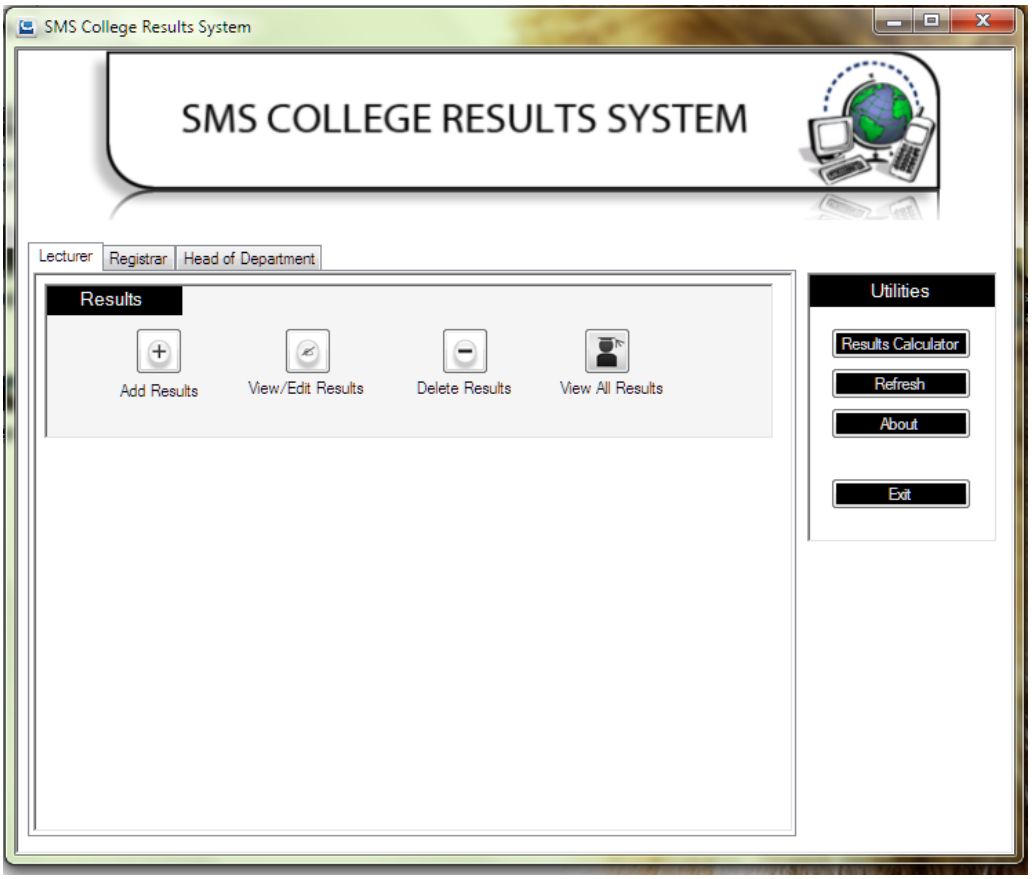


Fig. 12: SCRS Backend Main Page

The figure above is the main page of the backend data entry application. The application itself has 3 main parts: Lecturer,Registrar and Head of Department and a sidebar for various utilities as explained below.

- **Registrar**

This section has 3 parts: School, Lecturers and Students.

- School - Here a registrar enters data for different schools in a college, different departments and modules (subjects). The registrar can also view, edit and delete the schools, departments and modules (subjects). In this part, a registrar can also assign a dean to a school or an hod to a department.
- Lecturers - Here a registrar enters details for lecturers. These lecturers can be used to be set as deans or hod's for schools and departments. The registrar can also view, edit and delete lecturer details. Lecturers who are either deans or hod's cannot be deleted. A new dean or hod is to be set before the existing one can be deleted.
- Students - Here a registrar enters details for students. These details are critical especially "Student ID" and "IC/Passport No." as these will be required when a student wants to check for results or do preregistration via SMS. The registrar can also view, edit and delete student details.

- **Head of Department**

This section has 2 parts: Modules to be offered Next and Students.

- Modules to be offered Next - Here an HOD sets the modules that are to be offered the upcoming semester. This is also important because for a student to perform preregistration, there has to be a list of modules to be offered the next semester. So that, when a student sends an SMS for preregistration, the system will compare the modules he/she has completed and the modules to be offered the following semester and check to see if the student satisfies the prerequisites for the upcoming semester modules. If all is well, the student will be allowed to preregister.

  Hod can also "remove modules" to be offered the upcoming semester if a mistake was done or "remove all modules offered" when a semester has begun, so that students do not do preregistration in the middle of the semester.

- Students - Here the hod registers a student to a module. In order for a lecturer to enter a student's results, he/she must have been registered here by the hod. If a student is not registered here by the hod, his/her student ID will not appear in the section where a lecturer enters results, therefore, this particular student's results will not be available via SMS.

- **Lecturer** - This is a section where a lecturer enters the students results. For a student's results to be entered, the student must have been registered by the hod.

  When new results are entered, the results are entered into 2 separate tables in the database; "**Results**" Table and "**CompletedModulesResults**" Table. The reason for this is that the "**Results**" Table is used to extract results to be delivered via SMS. However, at the end of another semester, this results table will also contain previous semester's results. So, when a student checks for results via SMS, he will get both new and old results and the text message might be too long which is considered a nuisance. Therefore, the "**Results**" Table has to be cleared before results for another semester are to be entered.

  This brings up another issue. How about preregistration? How will students do preregistration via SMS if the "**Results**" Table is cleared every semester?
  This is why the "**CompletedModulesResults**" Table exists with a sole purpose of having a permanent storage of each student's results to be used to check for prerequisites when doing preregistration via SMS.

  The lecturer can also view, edit and delete. These actions are imitated for both "**Results**" Table and "**CompletedModulesResults**" Table except for **delete** for the "**CompletedModulesResults**" Table, results in this table cannot be deleted.

5.2.2 SMS College Results System SMS Management (Backend SMS Manager)

This application is part of the SMS College Results System and it is the one that handles the SMS functionality of the system. Without it, users would not be able to access results, check upcoming semester modules nor perform preregistration via SMS.

This application can also be used by an administrator to send manual SMS: incase a custom message SMS is to be sent to an individual, it is like having a computer act as a UI for a mobile phone, whereby one can send, read and delete the SMS.

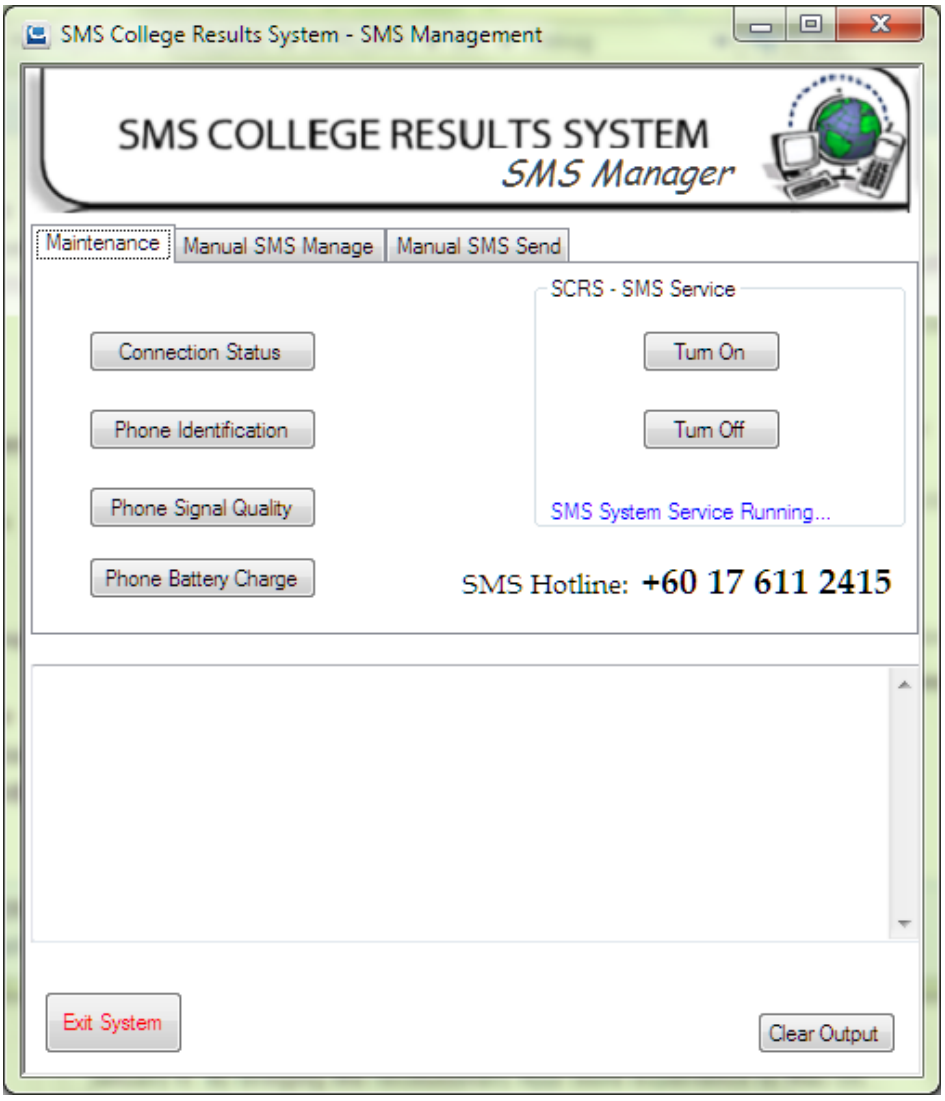


Fig. 13: SCRS Backend SMS Manager

The figure above is the main page for the SCRS Backend SMS Manager application. It has 3 parts; Maintenance, Manual SMS Manage and Manual SMS Send, as explained below.

- **Maintenance** - This is a section where a user can check for the current status of the SMS application as a whole.
  - Connection Status - This is used to check the connection status of the GSM/GPRS modem. At times, the modem may have lost connection to the computer using the **RS 232 serial port**, this confirms the connection status.
  - Phone Identification - This displays the details of the connected GSM modem or mobile phone: Manufacturer, Model, Revision and Serial Number.
  - Phone Signal Quality - This displays the GSM signal quality of the connected GSM device. It displays the Signal Strength in *Bits* and Bit Error Rate.
  - Phone Battery Charge - This displays the Battery Charge Level and Battery Charging Status incase the GSM device is battery powered. Useful if a battery powered device does not have a screen of its own.
  - SCRS - SMS Service - This can either be turned ON or OFF; when it is OFF, the SMS Service feature of the application is not running. This means that if a student checks for results, no reply will be sent. Turning the Auto-Reply feature off can be used for system maintenance purposes.

    When it is turned ON, all SMS's received while the system was off will be processed in the order they were received. This means, the system's Auto-Reply feature will be back on.

- **Manual SMS Manage** - In this section, a user can read all SMS, delete an SMS at a specific index or delete all SMS or enable/disable SMS notifications.
  - Read All SMS - This feature enables a user to read all SMS's contained in the storage. However, if the SMS Service is turned ON, the system will automatically delete each SMS that it has successfully processed. Reading of All SMS's could be used for system maintenance purposes incase of system downtime.
  - Delete SMS (at index) - This can be used to delete an SMS at a specified index.
  - Delete All SMS - This enables the user to delete all SMS contained in the storage. Though the system clears the SMS's when it detects the SIM storage is full, incase the system fails to do so and it has caused a traffic jam of SMS's, a user may manually delete all SMS in that storage. Doing so may, however, cause data loss since the received SMS will all be deleted without being processed first.
  - Enable/Disable SMS notifications - If enabled, a notification of a new SMS will be displayed on the computer screen.

- **Manual SMS Send** - This section enables the user to send SMS that the user has typed to a desired phone number.
  In a situation where a college may wish to contact a student through SMS. This feature can be used to send an SMS to that particular student. It could be; information about upcoming events, information about payment requirements, etc.

  A user can also send batch SMS's which will be processed in *batch mode* in the background.

5.3 Security Design

On the whole, students are still wary of letting their semester results be available via SMS. With the use of SMS, the main concern is with student authentication. Even though there is no way to ensure the identity of a student or perhaps the identity of a parent, a security measure had to be implemented.

The following is a procedure to implement security and ensure a bit of privacy:
SMS Format will be: *<keyword> <student_ID> <IC/Passport_No>*
e.g: RESULTS 708133903 AB071043

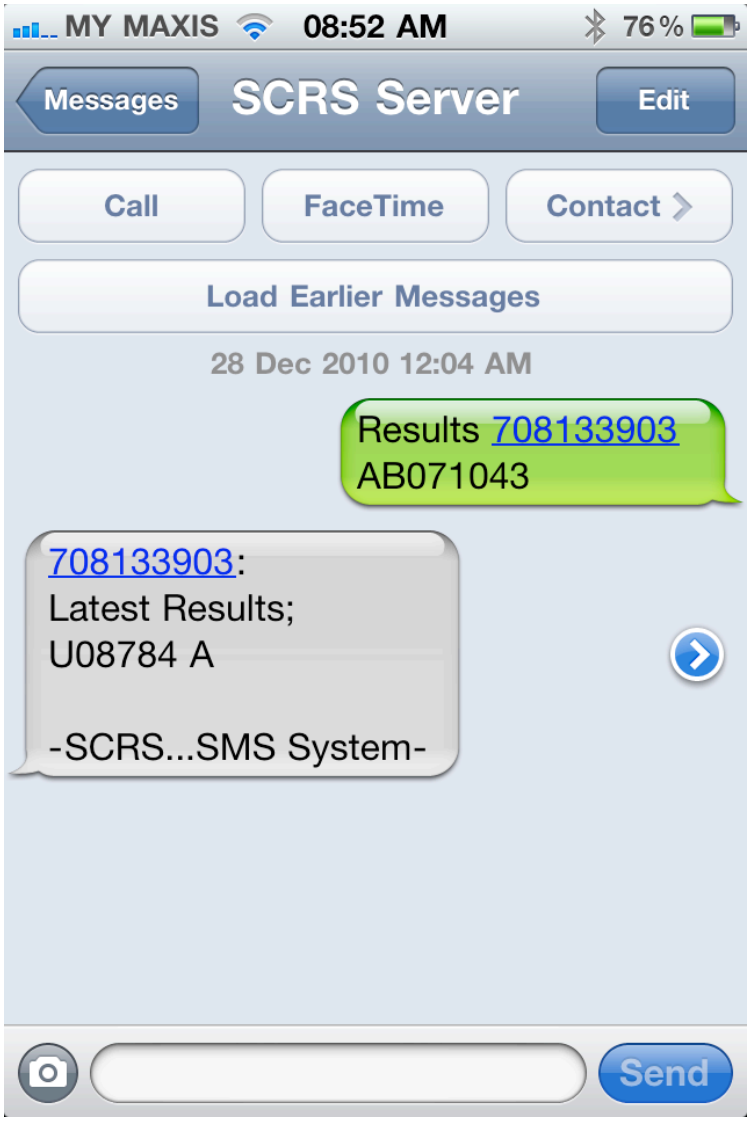


Fig. 14: SMS Security Design Screen Shot

This will ensure that for anyone to have access to a student's results, he/she has to know that student's ID number and that particular student's IC/Passport No. Though this is not a fool proof solution, it may act as a security measure at the moment.

5.4 Detailed Design

The following diagrams show the detailed design of SCRS Backend (Data Entry) application and SCRS Backend (SMS Manager) application.

5.4.1 SCRS Backend (Data Entry)

This class diagram describes the different classes and the interactions amongst each other in this part of the system.

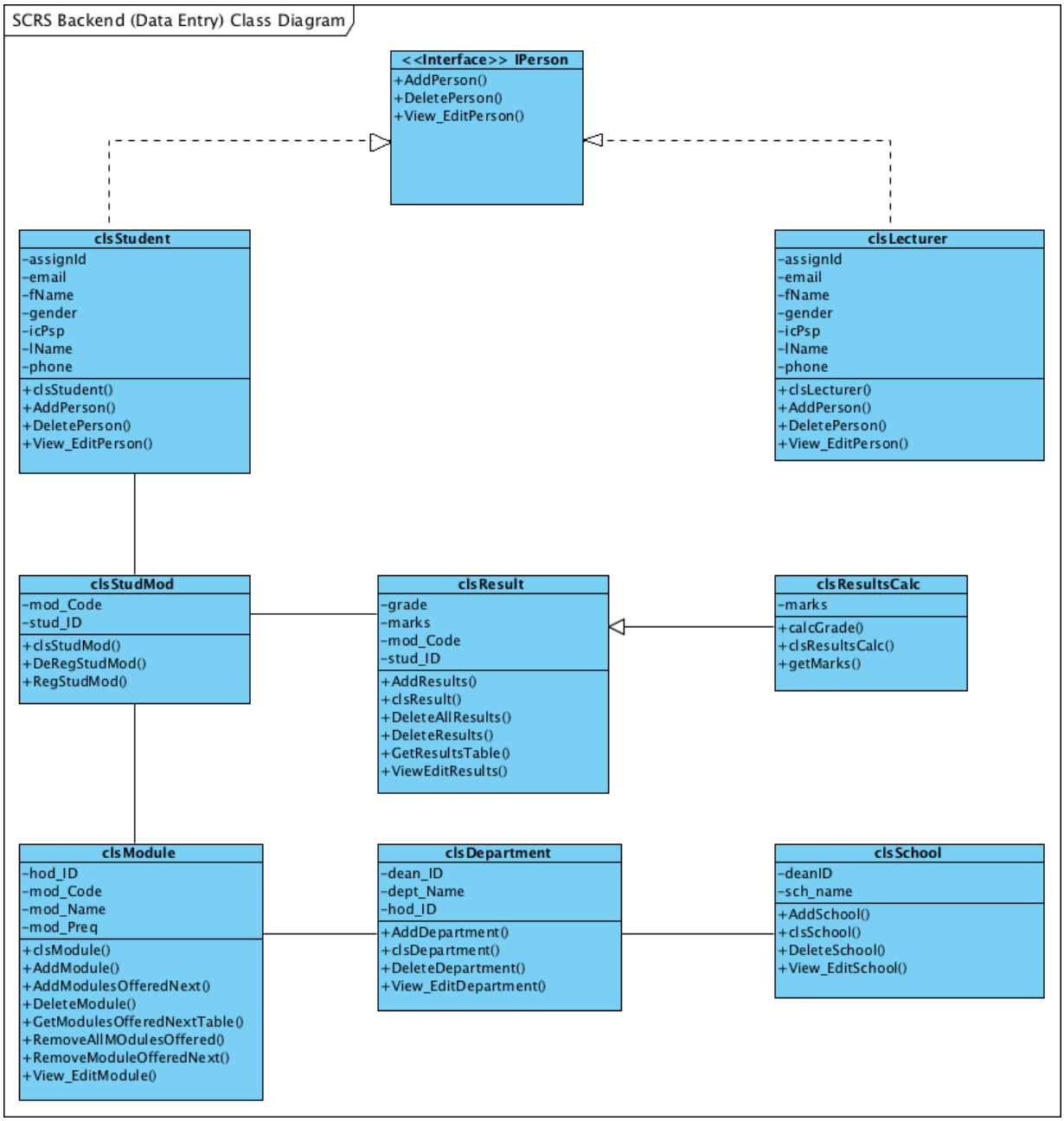


Fig. 15: SCRS Backend (Data Entry) Class Diagram

5.4.2 SCRS Backend (SMS Manager)

This class diagram describes the classes that handle the SMS functionality of the system.

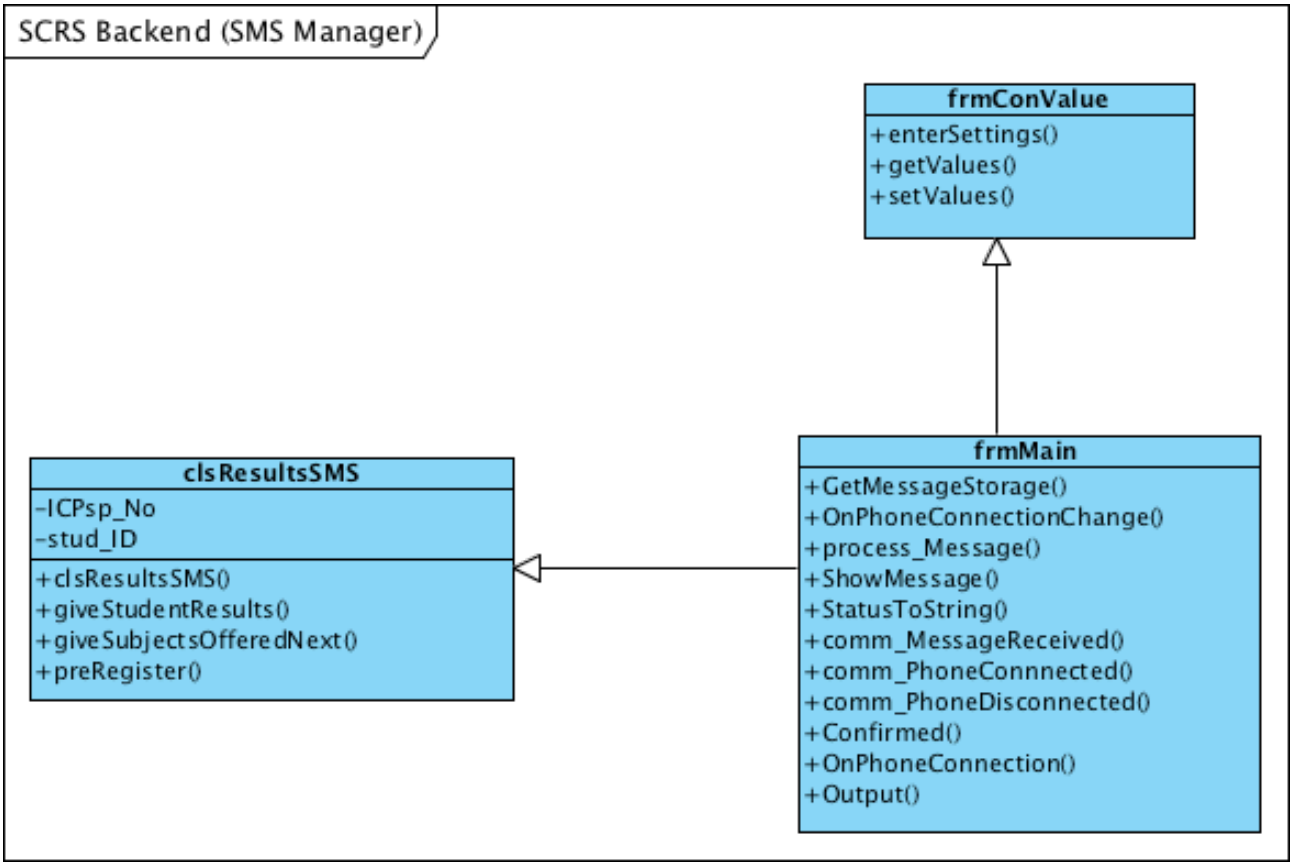


Fig. 16: SCRS Backend (SMS Manager) Class Diagram

5.5 Database Access System Design

Data Connectivity between the applications and the MySQL database was done using the .NET Connector library.

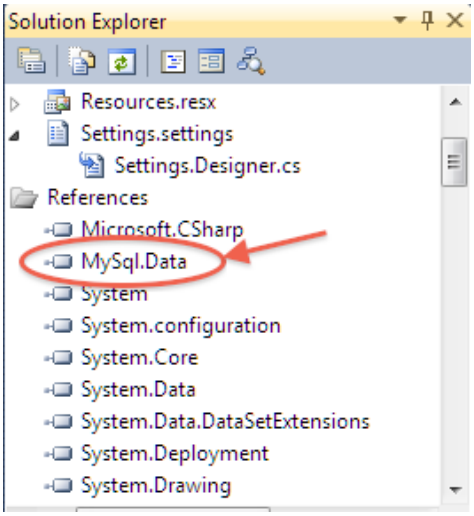


Fig. 17: MySQL.Data Reference

Database permissions and other configurations were saved in a Configuration Settings file (**App.config**) which was included in all classes that required database connectivity.

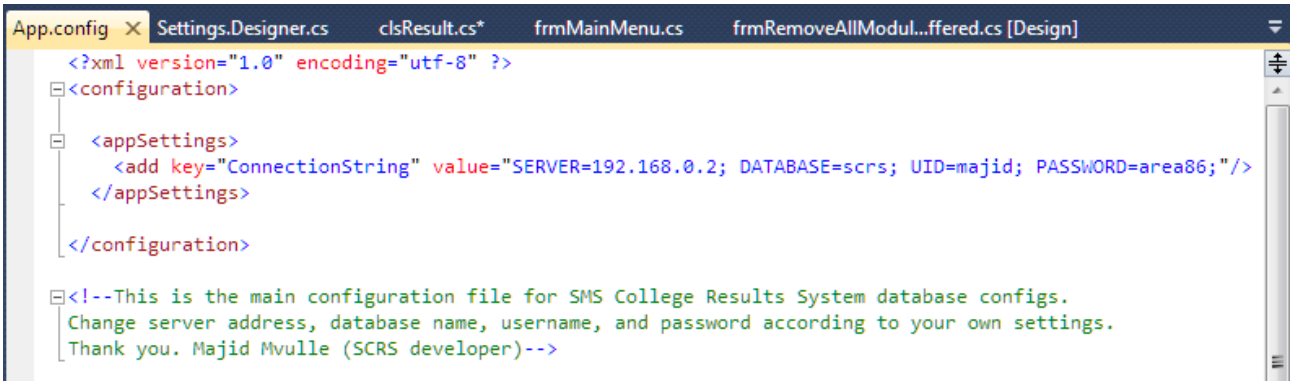


Fig. 18: Configuration file

Inclusion of .NET Connector Library, System Configuration files was done as demonstrated below:

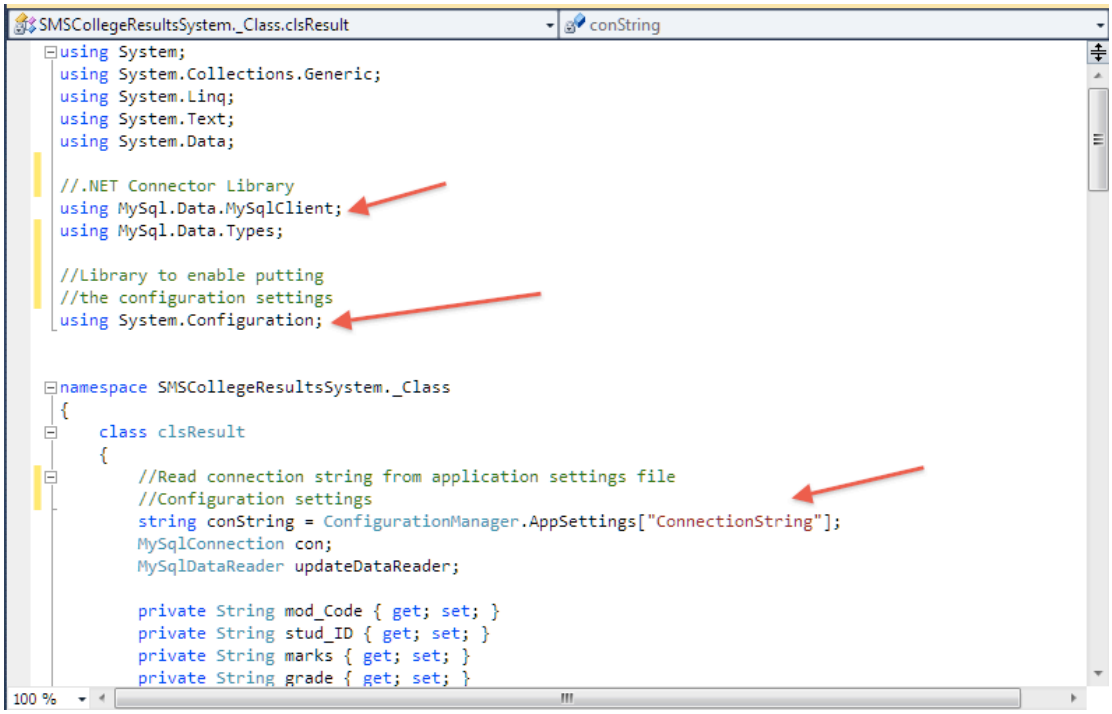


Fig. 19: .NET Connector and Configuration File inclusion

5.5.1 Entity Relationship Diagram

The diagram below represents the SCRS system Entity Relationship Diagram. It shows the entities, attributes and relationships between the Tables that make up the SCRS database.

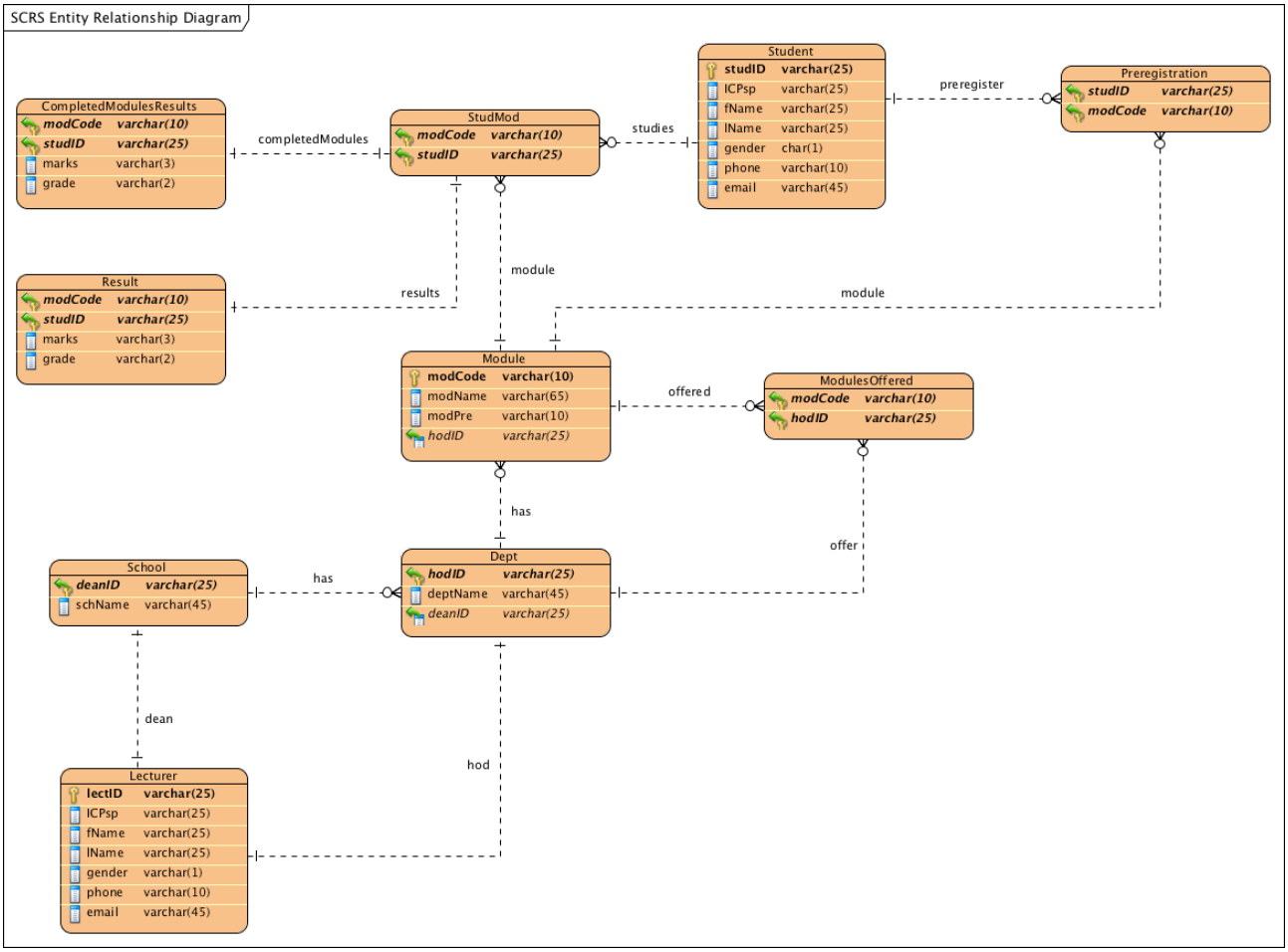


Fig. 20: SCRS Entity Relationship Diagram

5.5.2 Data Dictionary

The following are the tables derived from the EAR Diagram above and are contained in the SCRS database.

Dept

Field	Type	Allow Null	Default Value
hodID	varchar(25)	No	
deptName	varchar(45)	No	
deanID	varchar(25)	No	

Table 9: Dept Table

Lecturer

Field	Type	Allow Null	Default Value
lectID	varchar(25)	No	
ICPsp	varchar(25)	No	
fName	varchar(25)	No	
lName	varchar(25)	No	
gender	char(1)	No	
phone	varchar(10)	No	
email	varchar(45)	No	

Table 10: Lecturer Table

Module

Field	Type	Allow Null	Default Value
modCode	varchar(10)	No	
modName	varchar(65)	No	
modPre	varchar(10)	No	
hodID	varchar(25)	No	

Table 11: Module Table

ModulesOffered

Field	Type	Allow Null	Default Value
modCode	varchar(10)	No	
hodID	varchar(25)	No	

Table 12: ModulesOffered Table

Preregistration

Field	Type	Allow Null	Default Value
studID	varchar(25)	No	
modCode	varchar(10)	No	

Table 13: Preregistration Table

Result

Field	Type	Allow Null	Default Value
modCode	varchar(10)	No	
studID	varchar(25)	No	
marks	varchar(3)	No	
grade	varchar(2)	No	

Table 14: Result Table

School

Field	Type	Allow Null	Default Value
deanID	varchar(25)	No	
schName	varchar(45)	No	

Table 15: School Table

Student

Field	Type	Allow Null	Default Value
studID	varchar(25)	No	
lCPsp	varchar(25)	No	
fName	varchar(25)	No	
lName	varchar(25)	No	
gender	char(1)	No	
phone	varchar(10)	No	
email	varchar(45)	No	

Table 16: Student Table

StudMod

Field	Type	Allow Null	Default Value
modCode	varchar(10)	No	
studID	varchar(25)	No	

Table 17: StudMod Table

CompletedModulesResults

Field	Type	Allow Null	Default Value
modCode	varchar(10)	No	
studID	varchar(25)	No	
marks	varchar(3)	No	
grade	varchar(2)	No	

Table 18: CompletedModulesResults Table

6 SYSTEM IMPLEMENTATION AND TESTING

6.1 Introduction

In the process of system implementation, the system requirements are converted into program codes. This is the software development phase that is concerned with the development environment.

6.2 Development Environment

During the early stages of development, various tools were tested to ensure the suitability of the tools to the task. This is rather an important phase that precedes the actual development itself. There is a need to discover what tools are available, and to test out those tools that operate in the hardware and software development platform.

Admittedly, this phase took up a great deal of time in the search for the right development tools. Tools and development environments need to fit into the available resources. On the other hand, it may be necessary to acquire new equipment to properly execute new tools. The development and implementation were conducted in two locations:

- Development Environment - The development activities were conducted at home and in this case home serves as the main development environment.
- Field Trial Environment - The developed system was then moved over to a colleague's home for testing and implementation. Obviously, there were problems that arose in moving between the two environments.

6.2.1 Hardware Environment

The following hardware specifications have been used to develop the SCRS system. There are two sets of development machines at home - 1 Windows PC and 1 Macintosh Laptop. Macintosh Laptop was used to run the MySQL database.
Windows PC specifications:

- AMD Athlon X2 250 Processor running at 3.00 GHz
- 2 GB DDR3 RAM
- 500 GB Hard Disk
- Standard hardware for IBM PC compatible machine

Macintosh Laptop specifications:

- Intel Core 2 Duo Processor running at 2.26 GHz
- 2 GB DDR3 RAM
- 320 GB Hard Disk
- Standard Hardware for an Apple compatible computer

These two sets of machines also act as the server machines. The main application of the SCRS system runs on the Windows PC and the MySQL database runs on the Macintosh Laptop. This will enable many PC clients to be able to run the SCRS application since they will all be connected to a single database on the Macintosh Laptop running on the same network.

To provide a GSM connection, a GSM modem was used. **Wavecom WMOD2** GSM Modem was used. It has the following features:

- Dual-band GSM modem (EGSM900/1800 MHz or EGSM900/1900 MHz) designed for data, fax, SMS and voice applications.
- Short Messages Services features:
    - Text and PDU
    - Point to point (MT/MO)
    - Cell Broadcast
- ME+SIM phone book management
- Open Software Platform (support for embedded user applications)
- Interfaces: RS-232 and audio through mini sub-D 15-pin connector
- Supporting:
    - Remote control by AT commands (GSM 07.07 and 07.05)
    - Baud rate from 300 to 115,200 bits/s
    - Auto-bauding (300 to 38,400 bits/s)
    - Power supply through micro-FIT 4-pin connector
    - SMA antenna connector
    - Sliding SIM holder (3V/5V SIM interface)

(Source: Product documentation and user manual)

6.2.2 Software Environment

A wide variety of software tools were used in the development of the SCRS. A few of these tools could be obtained free from the Internet open source domain and a few others are commercially available.

Software Category	Software Name	Features
Operating System	Microsoft Windows 7 Ultimate (64 bit), Mac OS X 10.6.5	Platform for SCRS system
Database Server	MySQL Community Server 5.5.8	Database that hosts the SCRS database
Database Client	Navicat Premium version 9.0.4 (for Mac)	Tools used to connect to MySQL database server and browse the database schema
Software Development Tools	Microsoft Visual Studio 2010 Ultimate version 10.0.30319.1 (for Windows)	Tool used for coding
UML	Visual Paradigm for UML Enterprise Edition version 7.0	For drawing UML diagrams
Word Processor	Pages ‘09	For project write up and other word processing tasks

Table 19: Software Environment

6.3 System Development

The development strategy is a topdown approach, where the most important components were first developed. Specifically, the SCRS - Backend Data Entry system was developed at an early stage of the development. Then the SCRS - Backend SMS Manager system was developed later.

6.3.1 Classes and Components

This section will discuss the different classes and components used in the development and implementation of the SCRS. Since this research involves the development of a GSM server and subsequently the development of SCRS system, we will discuss the core components in details.

6.3.1.1 GSMComm Library

This is a GSM Communication library containing a set of components to aid developers in performing SMS-related tasks with compatible GSM mobile phones or GSM modems. It is open source and was developed by *Stefan Mayr (www.scampers.org/steve/sms/)*. This is the core library used to develop the SMS part of the SCRS system. It contains different functions required to: send, receive, read, delete, etc SMS messages. These will be discussed further below:

6.3.1.1.1 GSM Communication

The *GSMCommMain* is the main class here. This class enables communication with a GSM phone that supports AT commands. A few functions contained in this implementation class include:

- *BatteryChargeInfo* - Contains the ME battery charging status and charge level
- *DecodedShortMessage* - Represents a short message from the phone in its decoded state
- *GsmCommMain* - Interacts with a mobile phone to execute various functions
- *ShortMessage* - Represents a short message in undecoded PDU format.
- *SignalQualityInfo* - Contains the signal quality as calculated by the ME.

6.3.1.1.2 PDU Converter

This is responsible for creating and decoding SMS messages. No communication is done here. The classes *SmsSubmitPdu, SmsDeliverPdu and SmsStatusReportPdu* represent the most relevant message types:

- *SmsPdu* - This provides the base for an SMS PDU. This class is abstract and so cannot be instantiated.
- *SmsSubmitPdu* - Represents an SMS-SUBMIT PDU, an outgoing short message. Used to send SMS messages.
- *SmsDeliverPdu* - Represents an SMS-DELIVER PDU, a received short message. Used to receive SMS messages.
- *SmsStatusReportPdu* - Represents an SMS-STATUS-REPORT PDU, a status report. Used to check the status report of sent SMS messages.

6.3.1.2 SMS College Results System Data Entry (Backend Data Entry)

This is the backend application of the system that is used to enter data into the database. Data like; student results, student details, lecturer information, department details, etc. Classes in this application are divided into two; forms and classes. A class could be used by various forms as stated below:

Forms	Class	Functionality
- frmAddDept - frmViewDept - frmDeleteDept	clsDepartment	- Add Department - View Department - Edit Department - Delete Department
- frmAddLecturer - frmViewLecturer - frmDeleteLecturer	clsLecturer	- Add Person - View Person - Edit Person - Delete Person
- frmAddModule - frmViewModule - frmDeleteModule - frmAddModulesOfferedNext - frmRemoveModule - frmRemoveAllModulesOffered	clsModule	- Add Module - View Module - Edit Module - Delete Module - Add Modules offered next semester - Remove Module offered next semester - Get Modules offered next semester (return datatable) - Remove all Modules to be offered next semester
- frmAddResults - frmViewResults - frmDeleteResults - frmViewAllResults	clsResult	- Add Results - View Results - Edit Results - Delete Results - Get Results (return datatable) - Delete all Results
NIL	clsResultsCalc	- Calculate Grade

Forms	Class	Functionality
- frmAddSchool - frmViewSchool - frmDeleteSchool	clsSchool	- Add School - View School - Edit School - Delete School
- frmAddStudent - frmViewStudent - frmDeleteStudent	clsStudent	- Add Person - View Person - Edit Person - Delete Person
- frmRegStudMod - frmDeRegStudMod	clsStudMod	- Register Student to Module - DeRegister Student from Module
NIL	IPerson (interface class)	- Add Person - View Person - Edit Person - Delete Person
frmAbout	NIL	- Display information about application version, developer, etc
frmWelcome	NIL	- SCRS splash page
frmMainMenu	NIL	- Main Menu of the application

Table 20: Backend Data Entry Forms, Classes and their functionalities

6.3.1.2.1 How *forms* interact with *classes*

The code snippet below shows how a form can access a class to be able to use its functions in implementing a certain function. The process is done by creating an object of the class that is required in the form. The example below takes place when a user clicks “Save Changes” button on a *frmViewResults* form.

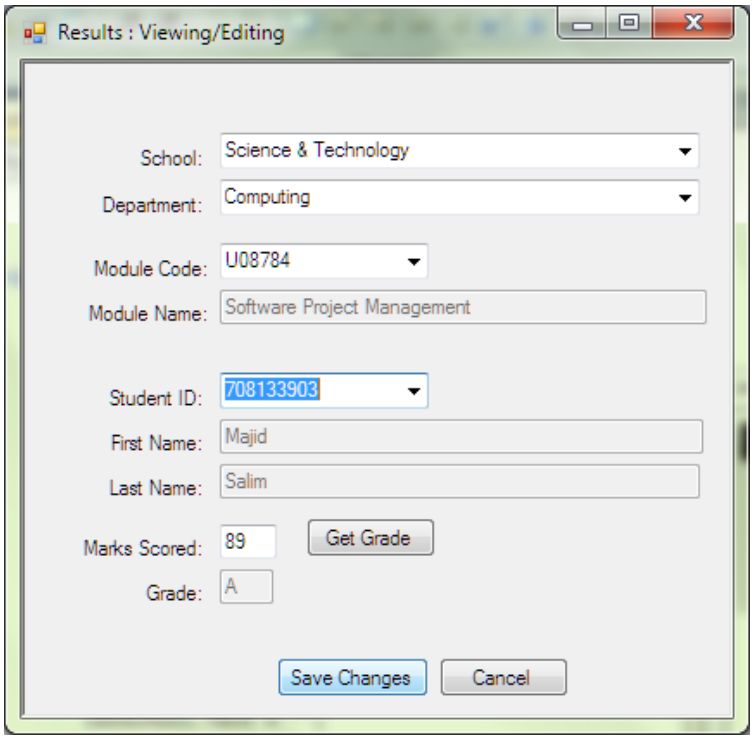


Fig. 21: View/Edit Results Screen Shot

```
private void btnSaveChanges_Click(object sender, EventArgs e)
{

    try
    {
        result = new _Class.clsResult(cboMod.Text.ToString(), cboStudID.Text.ToString(),
            tBoxMarks.Text.ToString(), grade);

        result.View_EditResults(cboMod.Text.ToString(), cboStudID.Text.ToString(),
            tBoxMarks.Text.ToString(), grade);

        MessageBox.Show
            ("Student (" + cboStudID.Text + ")'s results for (" + cboMod.Text + ") have been updated!",
            "Infomation",
            MessageBoxButtons.OK, MessageBoxIcon.Information);

        cboSchool.Text = "";

        cboMod.Text = "";
        cboMod.Enabled = false;



        tBoxFName.Text = "";
        tBoxLName.Text = "";
        tBoxModName.Text = "";
```

The code above is executed when the “Save Changes” button is clicked.

6.3.1.3 SMS College Results System SMS Management (Backend SMS Manager)

This is the backend application of the system that is used for SMS functionality. This includes; processing of results requests through SMS, sending of SMS, receiving of SMS, reading of SMS messages, managing the phone connection, etc. This application contains only two forms and one class as below:

Forms	Class	Functionality
frmMain	clsResultsSMS	- Give Student results - Connection Status - Phone Identification - Phone Signal Quality - Phone Battery Charge - Read SMS - Read all SMS - Send SMS - Delete SMS - Delete SMS at index - Delete all SMS - Enable/Disable SMS notifications - Process requests received via SMS - Enable/Disable SMS Service
frmConValue	NIL	- Set phone connection values (COM-Port, Baud Rate, Time Out)
frmWelcome	NIL	- SCRS splash page

Table 21: Backend SMS Manager Forms, Classes and their functionalities

6.3.2 Coding

While writing the programming codes for the SCRS system, the developer followed strict coding standards to make it easier to trace requirements from analysis through design and implementation. Other than that, coding and naming standards make a piece of coding more readable and eases the process of debugging. Naming standards were used for variables, objects, classes and controls.

6.3.4 Comments

Besides this, comments are inserted into the coding to help other developers understand and maintain the code. Comments begin with “//” or “/** ...*/” sign. An example would be as below:

```
private void btnExit_Click(object sender, EventArgs e)
{   /*
     * Display a message box asking a user if they really want to
     * Exit the system.
     */
    if (MessageBox.Show
        ("Are you sure you want to Exit SCRS - SMS Management?"+
        "\nStudents will not be able to use the SMS Service!",
        "SMS College Results System",
          MessageBoxButtons.OKCancel, MessageBoxIcon.Question,
          MessageBoxDefaultButton.Button2) //Set Focus on 'Cancel' button
          == DialogResult.OK)
    {
        // Clean up comm object
        if (comm != null)
        {

            // Close connection to phone
            if (comm != null && comm.IsOpen())
                comm.Close();

            comm = null;
        }

        dlg.Dispose(); //Memory management, just incase

        Application.Exit();//Quit all running processes of application
    }
```

6.3.5 Error Handling

The coding in the SCRS is supported by error-handling codes or segments. These segments are inserted into SCRS to intercept and handle all unexpected errors that may happen. Other than that, all error handling segments can be self-programmed and written, providing much flexibility to handle the errors according to the developer's choice. With all these error handling segments, SCRS is much more reliable and stable, because every time an error happens, the error handling code segment will be executed to avoid system crashes or unnecessary restarts. An example of error handling is shown in the figure below:

```
private void btnPhoneID_Click(object sender, EventArgs e)
{
    Cursor.Current = Cursors.WaitCursor;

    try
    {
        IdentificationInfo info = comm.IdentifyDevice();
        Output("Manufacturer: " + info.Manufacturer);
        Output("Model: " + info.Model);
        Output("Revision: " + info.Revision);
        Output("Serial number: " + info.SerialNumber);
        Output("");
    }
    catch (Exception ex)
    {
        Output("Exception: " + ex);
    }

    Cursor.Current = Cursors.Default;
}
```

6.3.6 Object Oriented Programming Methodology

SCRS was developed using C# programming language under the .NET Framework. C# supports Object Oriented Programming paradigm, hence, SCRS coding style is Object Oriented as well. Compared to structural programming, OOP enables the developer to reuse the codes written, saving precious time and promotes efficient usage of the objects and classes. In SCRS, all reusable functions are written in standard modules and can be invoked anywhere in the program. All the same classes are reused repeatedly in different modules.

6.3.7 Complexity of programming SCRS

To make SCRS efficient and accurate, some complex programming techniques were applied while developing the code for SCRS. For instance, Arrays and Collections were used to control and manage data streams to and from the database.

6.4 Testing Approach

The goals of software testing are to establish the functionality of a system as well as to detect the presence of errors or inaccuracies in the system. In a software production company, the testing activity is meant to minimize costs of quality and defects in software production. However, the main objective of testing is to find defects, verify capability and predict reliability. There are various approaches to testing; each type of testing validates different aspects of the software. The selection of the type of testing depends on the nature of the software system.

6.4.1 Type of Testing

The types of testing that are to be executed depends on the stage in the development and the testing to be done. The testing process of SCRS follows a pattern that can be shown in the figure below. The first stage in the testing process involves unit testing. There could be multiple units to test. After this, there is the module testing stage, which can involve multiple modules to be tested. If any of these modules are defective, it is necessary to return to unit testing. However, this return path is optional, and is performed only if the module testing failed. The dotted arrow indicates this optional path. This is followed by interface testing, to test the communication between the various modules. When all these are done, the integration phase begins, combining all the individual modules. Finally the whole system is tested.

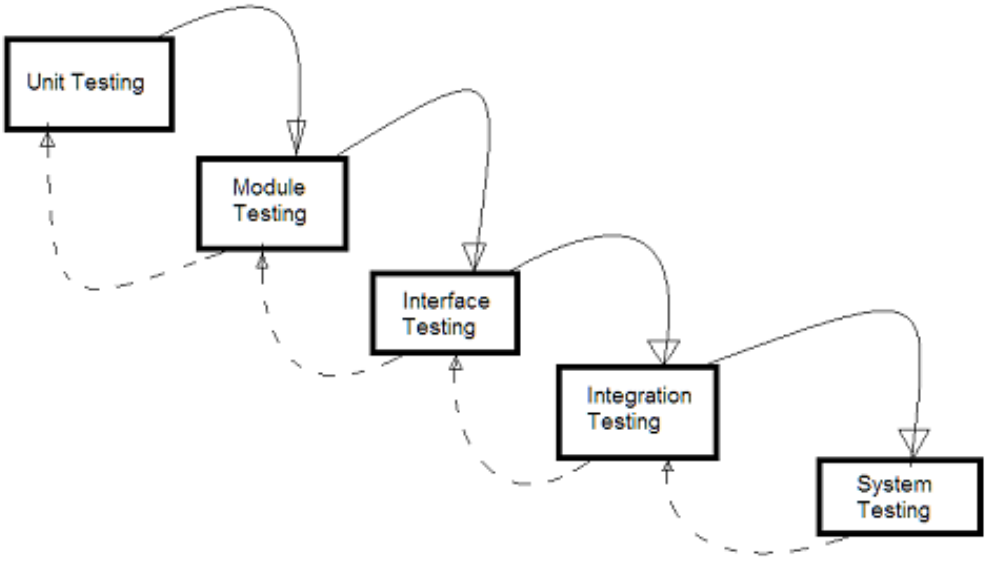


Fig. 22: Testing Process

6.4.1 Unit Testing

This stage aims at testing the smallest components of the code. The initial units in this project are the individual modules, each of which may consist of several units. The tests are done by using ***NUnit***. There are several approaches to unit testing, each of which has their own strengths and weaknesses.

6.4.1.1 Black Box Testing

This is a functional testing procedure to establish whether the unit meets its specifications. For example, in the screen that allows the entry of two inputs. The output is then studied. The result is studied whether it produces a successful output which indicates a successful test. In Black Box testing, we are only concerned about what the inputs are and what outputs are generated as a result of the test. Black box testing is only concerned with functionality. If the output is incorrect, or there is no data output, the test has failed.

6.4.1.2 White Box Testing

If the results of a black box testing indicates a failure, there are other approaches to rectify the problem. Here the structural testing approach is needed, where the code is analyzed. Knowledge of the structure of the component is vital in white box testing.

6.4.1.3 Code Flow

It is often necessary to examine the flow, or path of a code segment. This could lead to a deeper and deeper examination of the flow of the logic of the code. Path testing is also a part of examining code flow, which traces the flow of the program from one point to another.

6.5 Module Testing

When each of the units have been tested, it is necessary to ensure that the modules work together. Such is the case of the *results module* as one unit. Then there is the *student module* to initiate a call to the *results module* and consequently obtain the results from the module. The *results module* consists of several units that are combined (such as grade calculator, etc). Each of these units must have been tested earlier, and combined to form the *results module*. Each unit does not work alone, as each unit depends on another unit.The testing must ensure that these sub-units work correctly in a module.

6.5 Interface Testing

Interface testing in this project is very important. Data is passed between an SMS Phone Application (found in each mobile phone nowadays) and a Database Server. This data must conform to the same standards and must reach its destination in the form that is expected. In interface testing, we need to know whether these messages are in the format that is expected by both ends.

6.5 Integration Testing

There are two types of integration testing which are; incremental testing and non-incremental testing. Non-incremental testing will combine all the modules and test the program. With incremental testing, it systematically combines and tests each module. SCRS integration testing makes use of the thread testing approach (one of the testing approaches in incremental testing) to carry out the integration test. The integration test starts by integrating groups of modules that implement a function together. It then continues integrating and testing by adding one function at a time. This approach provides better visibility and tracking of integration testing activities. This testing will expose any inconsistencies in each module, as well as inconsistencies in the integration itself. The figure below depicts the thread testing approach:

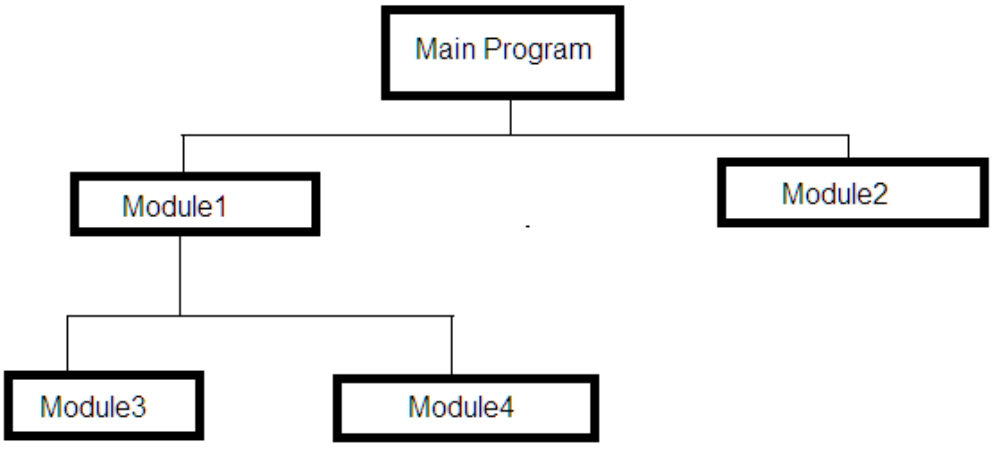


Fig. 23: Integration Thread Testing

6.6 System Testing

The final step in the test is system testing, where a series of tests are conducted to exercise the whole system. The whole system includes both the interaction between the modules, as well as the interaction with the hardware. It aims at finding discrepancies between the system and its requirements. It concerns validating system's functional requirements. In this project, there is also the network factor, and the interaction between different computers in the network.

If the system works on one machine, can it work on other machines with different sets of hardware and software? Here it was necessary to test by hosting SCRS on one machine, use of different mobile phones and performing a test on how clients from other machines and different phones access the SCRS database.

There is also the need to test for system performance, stress test, security access rights, usability, data integrity, error handling and recovery.

7 EVALUATION AND TEST RESULTS

7.1 Test No. 1: Test Checking of Results via SMS

Step no	Test description	Test data	Expected result
1.1	Check results of an existing student ID and IC/Passport No.	keyword: RESULTS student ID: 708133903 IC/Passport No.: AB071043	Student's results are replied in SMS.
1.2	Check results of a non-existing student ID and IC/ Passport No.	keyword: RESULTS student ID: 708Xydh IC/Passport No.: AB071043	Error message replied in SMS indicating that a particular student is not found.

7.2 Test No. 2: Test Preregistration via SMS

Step no	Test description	Test data	Expected result
2.1	Attempt Preregistration of an existing student ID and IC/Passport No.	keyword: PREREG student ID: 708133903 IC/Passport No.: AB071043	Student's ID is added to the preregistration table and a confirmation SMS is replied.
2.2	Attempt Preregistration of a non-existing student ID and IC/Passport No.	keyword: PREREG student ID: N;'Xydh IC/Passport No.: AB071043	Error message replied in SMS indicating that a particular student is not found.

7.3 Test No. 3: Test Upcoming Semester Modules via SMS

Step no	Test description	Test data	Expected result
3.1	Check upcoming semester modules of an existing student ID and IC/Passport No.	keyword: SUBJECTS student ID: 708133903 IC/Passport No.: AB071043	Student's results are replied in SMS.
3.2	Check upcoming semester modules of a non-existing student ID and IC/Passport No.	keyword: SUBJECTS student ID: 708>,Xydh IC/Passport No.: AB071043	Error message replied in SMS indicating that a particular student is not found.

7.4 Test No. 4: Test Delete a Lecturer who is either a Dean or HOD

Step no	Test description	Test data	Expected result
4.1	Attempt to select a Lecturer who is assigned as either Dean or HOD.	Open "Delete Lecturer" form.	This lecturer's ID will not be listed here, therefore cannot be deleted.
4.2	Attempt to select a Lecturer who is not assigned as either Dean or HOD.	Open "Delete Lecturer" form.	This lecturer's ID will be listed here, therefore can be deleted.

7.5 Test No. 5: Test Add a duplicate entry of results in backend system

Step no	Test description	Test data	Expected result
5.1	Add a Duplicate entry for a student's results	student ID: 602133401 module: EC3000	When a submit button is clicked, a Message Box is displayed indicating a duplicate entry error.
5.2	Add a non-Duplicate entry for a student's results	student ID: 602133401 module: EC3218	When a submit button is clicked, a Message Box is displayed indicating a successfully entry of results.

7.6 Test No. 6: Test Add invalid characters in Windows forms

Step no	Test description	Test data	Expected result
6.1	Add an invalid character in a Windows form input	First Name: _=.,mckd$^	When a submit button is clicked, a Message Box is displayed indicating invalid First Name error.
6.2	Add valid character in a Windows form input	First Name: Simon	When a submit button is clicked, a Message Box is displayed indicating a successfully entry of a person's name.

8 CONCLUSION AND FUTURE WORK

8.1 Summary of Project

This project report describes a research focusing on a system that provides SMS functionality. The work encompasses detailed investigation of the subject matter, development of an SMS server and a backend college results application, and analysis of the results from the system.

The report began by investigating the manual Results system and how students acquire their previous semester results at the start of a new semester, with emphasis on the Results delivery process and the SMS features. This was to provide the foundation on which the SMS College Results System was to be proposed.

In Chapter 3, the methodology used to develop the SMS College Results System was presented. The benefits and justification of using the Incremental Development Model to develop the SCRS was also discussed.

In Chapter 4 and Chapter 5, the system analysis and design of the SCRS. The flexibility of Object Oriented Design and the advantages of the C# programming language largely motivated the development of SCRS. The SCRS is an elegant, well-structured system that exploits modern programming paradigms and GUI application design. The analysis and design phase was modeled by Unified Modeling Language (UML) by using use cases, class diagrams.

The implementation and development of the SCRS system was detailed in Chapter 6. The SCRS provides a mechanism for students to acquire results via SMS, check for upcoming semester subjects via SMS and perform preregistration via SMS. The entire development process provided insight on the detailed workings of the various open-source technologies and component of the C# programming language.

Chapter 7 described the test results after the system was thoroughly evaluated and tested. These showed by how much the system satisfies the use cases that were identified earlier.

8.2 Problems encountered and their solutions

Many problems were encountered in the development of the SCRS system. Some are technical problems of hardware and software. There are also human relationship problems in implementing a project of this magnitude.

8.2.1 Problem with user access rights

Windows OS has some user access rights strategies, which is distinct from other components, such as the database servers. There were some difficulties in finding compatibility between these access privilege requirements.

8.2.2 Insufficient documentation for open-source libraries

Although there is vast amount of open source SMS libraries available on the internet, most of them have limited documentation. Lack of documentation on the features, usage and configuration for these library components have narrowed down the scope of using these freely available features. On top of that, much effort was spent on R&D for the proper usage of these library components.

8.3 Evaluation by End Users

A few friends were given the system to test and give their feedback on the user interface and performance of the system. Feedback from them had mixed reactions but overall it was positive and that was encouraging enough to be certain the system is up to par with the expectations set earlier.

8.4 System Constraints and Future Enhancement

8.4.1 System Constraints

As discussed earlier, these are some of the constraints that the system faces or might face depending on the circumstances:

- The system cannot reply an SMS that is more that 160 characters long. It can receive an SMS, however long it is, but if the reply is more than 160 characters, the system is unable to reply. This is due to the inability of the **GsmComm Library** to send concatenated SMS's.
- The system can only send a maximum 6-10 text messages per minute. This is a limitation on the GSM network. As a standard, GSM phones are only permitted to send 6 text messages per minute. This means that the system can only serve 6-10 SMS requests per minute.

- It is independent of any existing university database. Hence, some information like administration, admission, enrollment, finance, resource management will not be integrated into this system.
- If a GSM network is offline, then the system is not operational since it requires a GSM network to send and/or receive text messages.
- Unlike paper results, which are stamped, it is difficult to *certify* SMS results. However, mobile operators provide premium custom numbers that bear the name of the school or college. This way the user can be sure that the message originated from the right source.
- The SMS technology does not guarantee set transmission times or guaranteed delivery of the message; therefore some messages may be delayed, blocked, or lost in transmission.

8.4.2 Future Enhancements

8.4.2.1 Integration with other existing systems

The system could be broadened to include student registration system, student enrollment system, financial system, timetabling system, etc.

8.4.2.2 A Web page as a back up

A web page that also provides could also be created to cater for those who may be out of range of GSM network or when the Telecom network tower close to the system is down, thereby crippling the system of SMS features.

8.4.2.3 Students Subscription

This capability could also be added to the system in the future whereby the system automatically sends a student's results to his/her phone whenever that student's results become available. This would reduce costs for students who keep sending SMS requests to the system and at times they get the keywords wrong.

8.5 Knowledge and Experience Gained

During this project, many new technologies were learnt and this was a rewarding experience. A few of the technologies learnt are as below:

8.5.1 .NET Framework

The Microsoft .NET Framework is a software framework for Microsoft Windows ® operating systems. It includes a large library and it supports several programming languages which allows language interoperability.

C# is one of the major programming languages used to build the SCRS system and I had to study it from scratch to be able to complete this project successfully.

8.7.2 GsmComm Library

This is a set of GSM libraries freely available on the internet. They enable .NET programming languages to have SMS capability without the need of using AT commands directly in the source code.

8.7.3 Other SMS Technologies

Other SMS technologies were also learnt, like; PDU, SMSC, web-based SMS servers, etc.

9 APPENDIX A - LIST OF REFERENCES

Alexander Bain, M. 2009, *How to Connect to a MySQL Database with C#*, online, retrieved 29 June 2010, from http://www.suite101.com/content/how-to-connect-to-a-mysql-database-with-c-a121029

Bell, James 2010, *Sailing Race Results System,* online, retrieved 02 September 2010, from http://www.o6u.edu.eg/dpages.aspx?FactId=12&id=272

Bkankenhorn, Dana 2010, *SMS for Life results have Tanzania dreaming silicon dreams*, online, retrieved 26 October 2010, from http://www.zdnet.com/blog/healthcare/sms-for-life-results-have-tanzania-dreaming-silicon-dreams/3556

Connection to the MySQL Server, online , retrieved 30 June 2010, from http://www.functionx.com/mysqlnet/csharp/Lesson02.htm

Danuta Pavileanene, BC, Vlilnius 2010, *Higher school entry results will be sent by SMS messages in Lithuania*, online, retrieved 20 July 2010, from http://www.baltic-course.com/eng/education/?doc=29276

Digi Plan, online, retrieved 27 October 2010, from http://www.digi.com.my/prepaid/use/pre_use_sms.do?page=cha

Exams and Results System, online, retrieved 11 August 2010, from http://www.o6u.edu.eg/dpages.aspx?FactId=12&id=272

Hotlink Plan, online, retrieved 27 October 2010, from http://www.hotlink.com.my/web/portal/hotlink/hotlinkpage?WCM_GLOBAL_CONTEXT=/web/wcm/connect/Hotlink_Content_lib_en/Home_Site_en/Plans_SA_en&iStruct=0:1:2

Ian Sommerville, 1996, Software Engineering, 3rd Edition, Addison Wesley

Microsoft Technet 2005, *HyperTerminal Overview,* online, retrieved 06 July 2010, from http://technet.microsoft.com/en-us/library/cc736511(WS.10).aspx

Mayr, Steve, *SMS Libraries (for GSM Phones),* online, retrieved 13 November 2010, from http://www.scampers.org/steve/sms/libraries.htm

Paul J. Deitel, Harvey M. Deitel, 2008, C# 2008 for Programmers (3rd Edition), Prentice Hall

Results SMS, online, retrieved 10 September 2010, from http://resultssms.org/

Short Message/SMS Tutorial, online, retrieved 03 July 2010, from http://www.developershome.com/sms/

SMSLib, online, retrieved 05 July 2010, from http://smslib.org/

SMS Gateway Kit, online, retrieved 09 August 2010, from http://www.mobitek.com.my/SMSGDK/SMS%20Gateway.html

SMS Tutorial, online, retrieved 13 September 2010, from http://www.funsms.net/sms_tutorial.htm

Suruhanjaya Komunikasi Dan Multimedia Malaysia (Malaysian Communications and Multimedia Commission), online, retrieved 27 October 2010, from http://www.skmm.gov.my/index.php?c=public&v=main

Thomas Connoly, Carolyn Begg, 2005, Database Systems: A Practical Approach to Design, Implementation, and Management, Pearson Education, Essex

Vivek Gite 2006, *How Do I Enable Remote Access To MySQL Database Server?*, online, retrieved 19 August 2010, from http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database-server.html

Wavecom WMOD2, online, retrieved 28 December 2010, from http://ajilimojili.iespana.es/ciencia/pdf/wavecom_wmod2.pdf

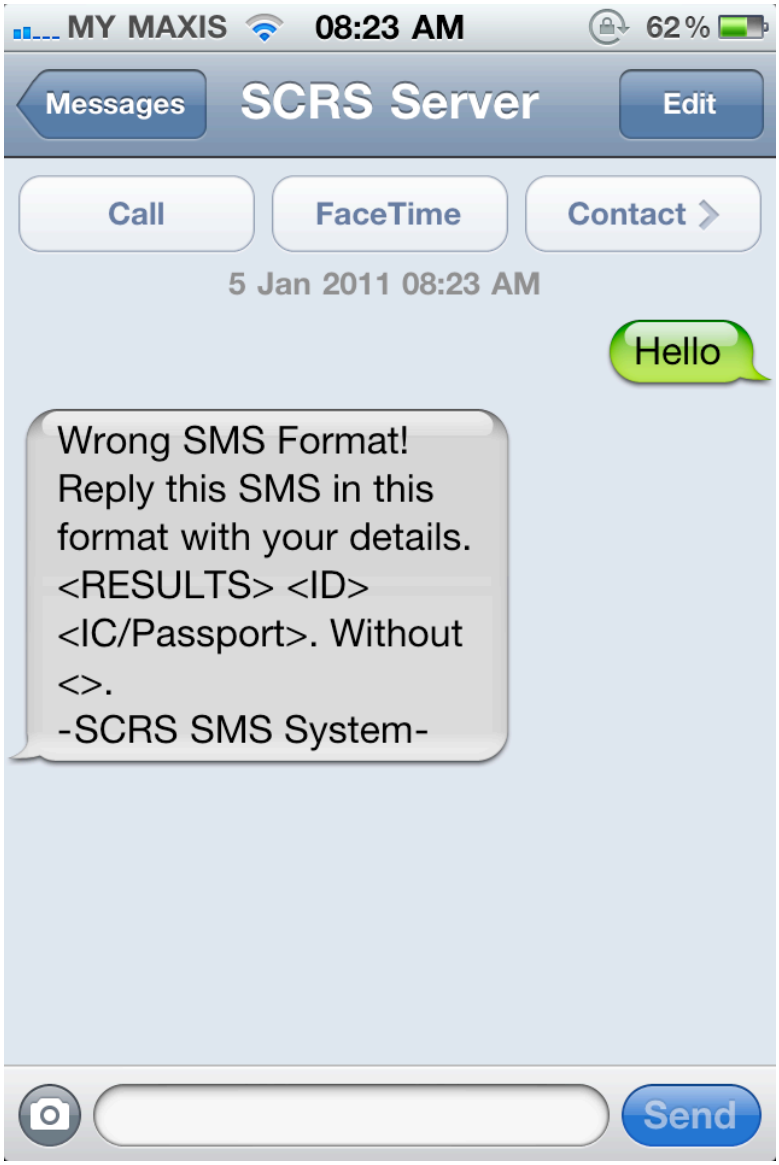
XPax Plan, online, retrieved 27 October 2010, from http://www.xpax.com.my/xpax_plan.html

10 APPENDIX B - SCRS SCREEN SHOTS

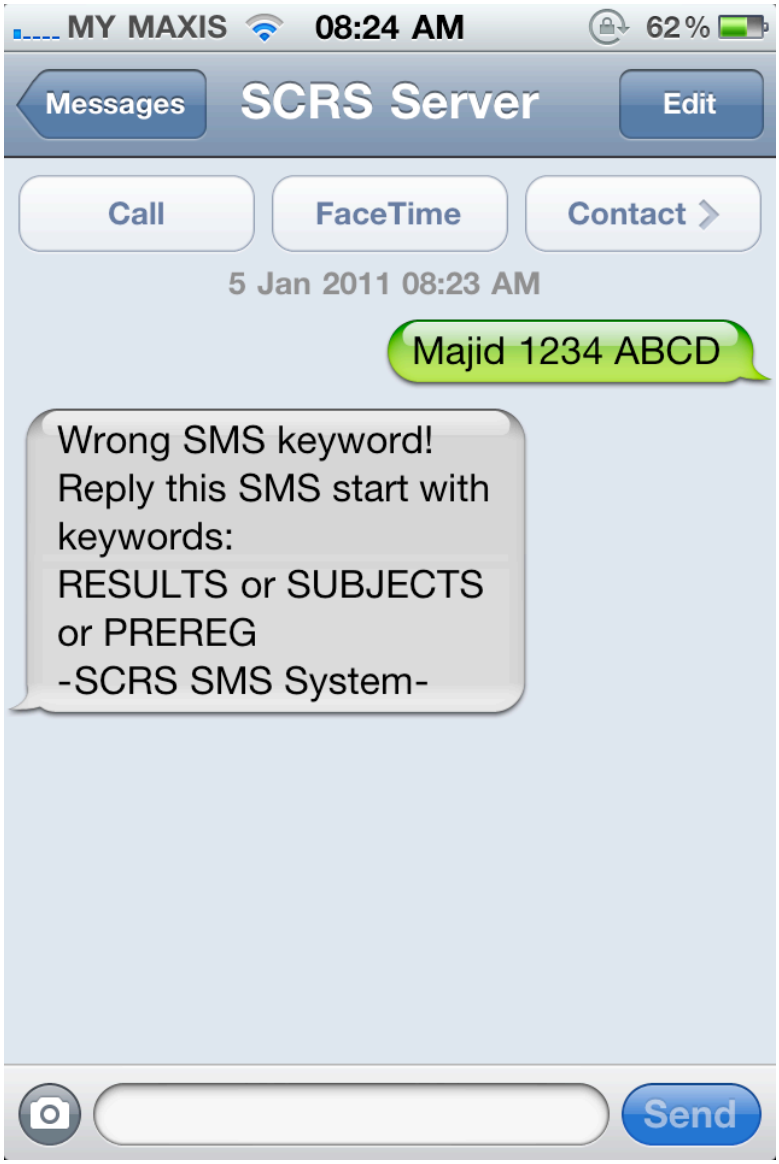
10.1 SMS Screen Shots

These are screen shots from a mobile for different requests sent to the SCRS system via SMS and the replies it gives back.

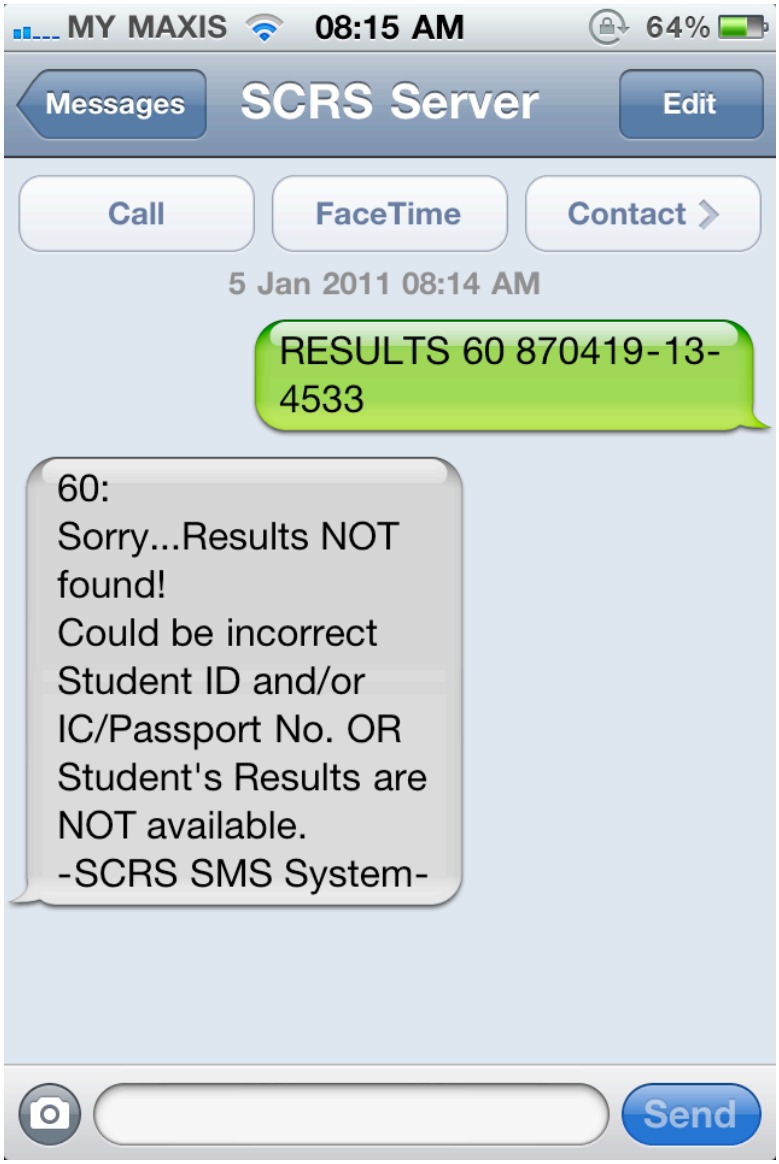
10.1.1 Wrong SMS format



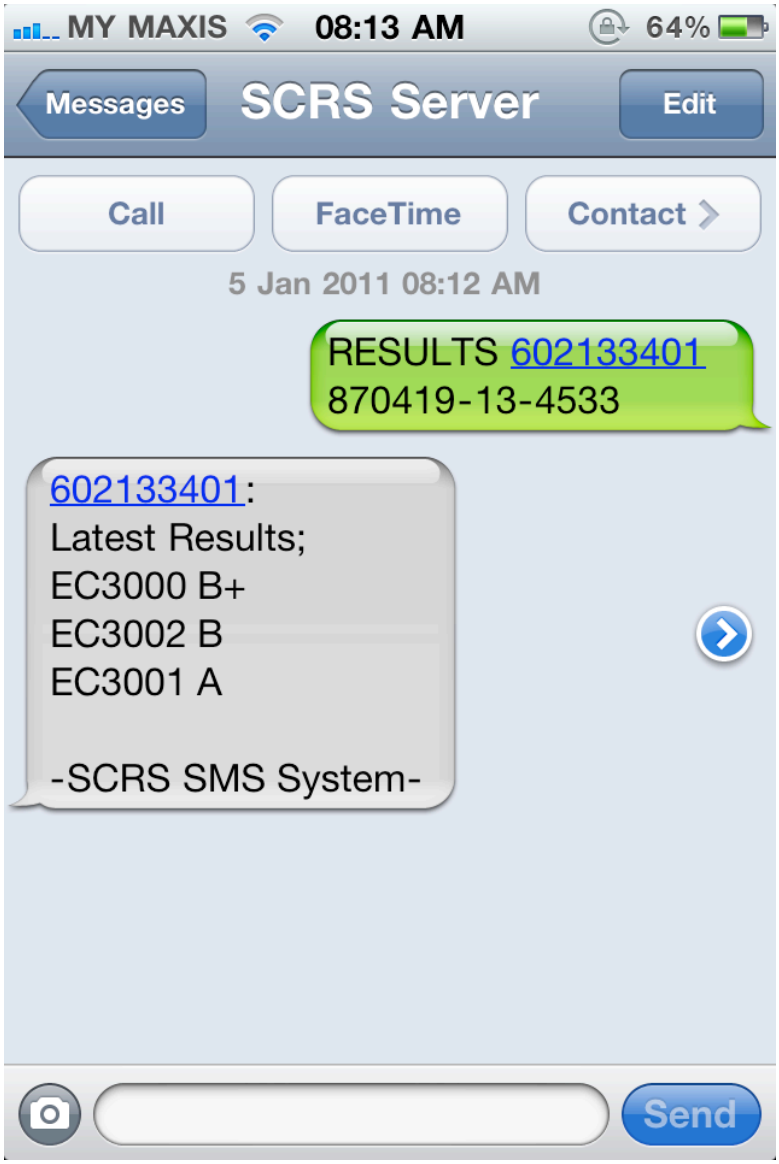
10.1.2 Wrong SMS keyword



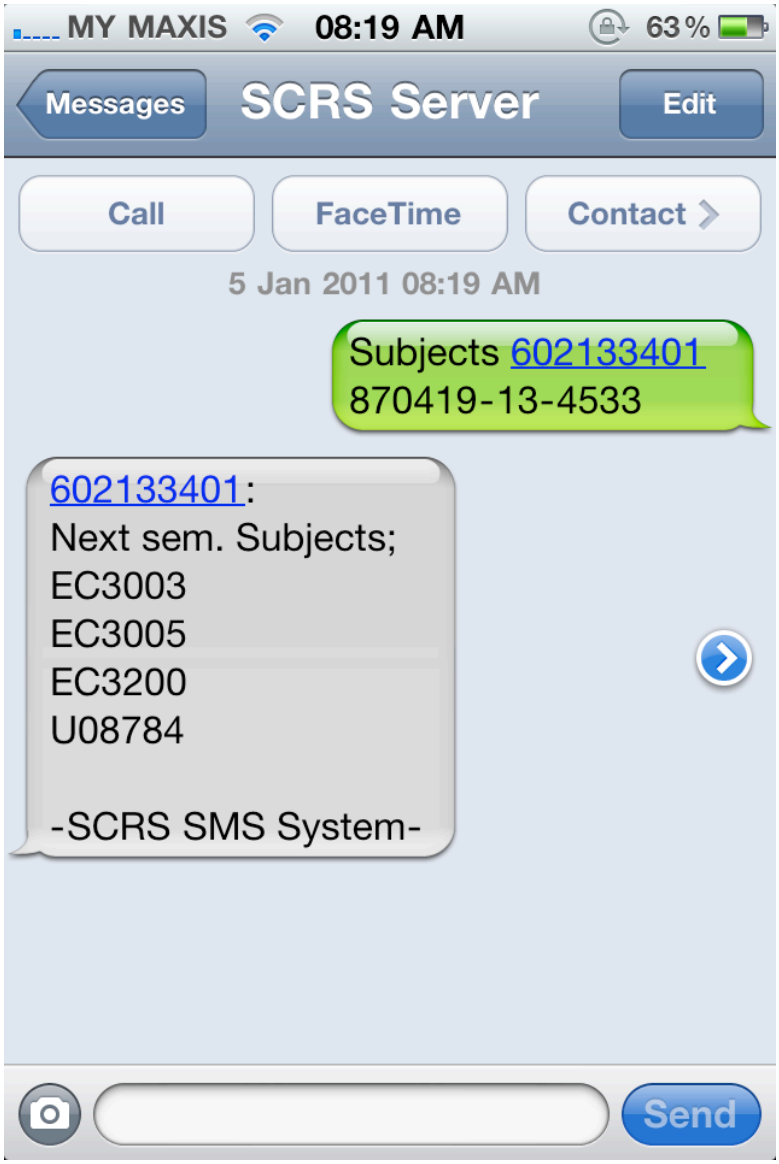
10.1.3 Wrong Student ID or IC/Passport No.



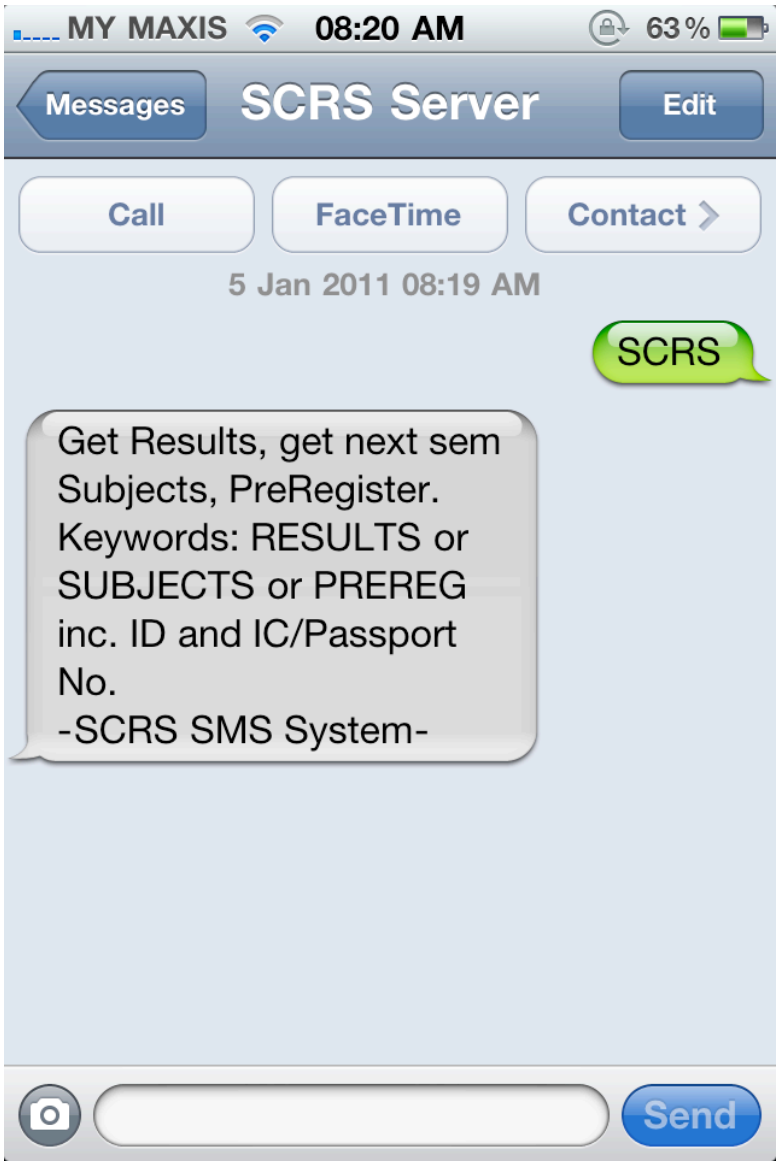
10.1.4 Get Results



10.1.5 Get Next Semester Subjects



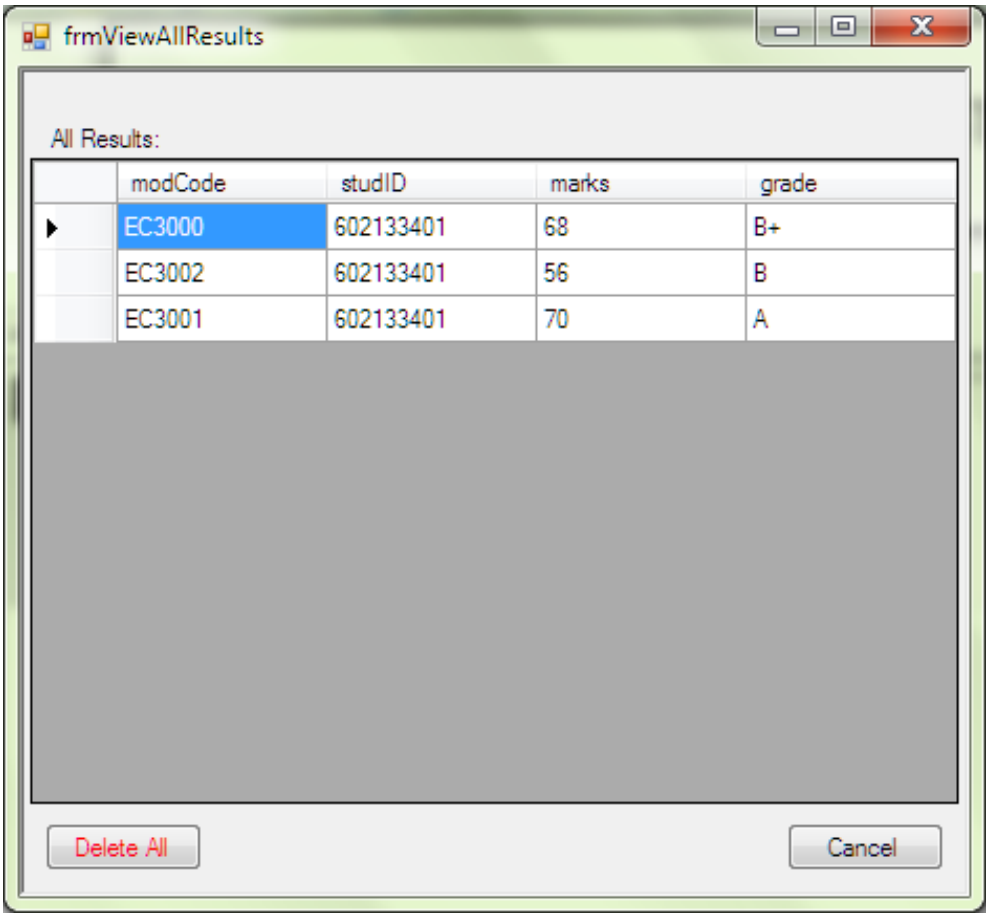
10.1.6 Get SCRS SMS Service Information



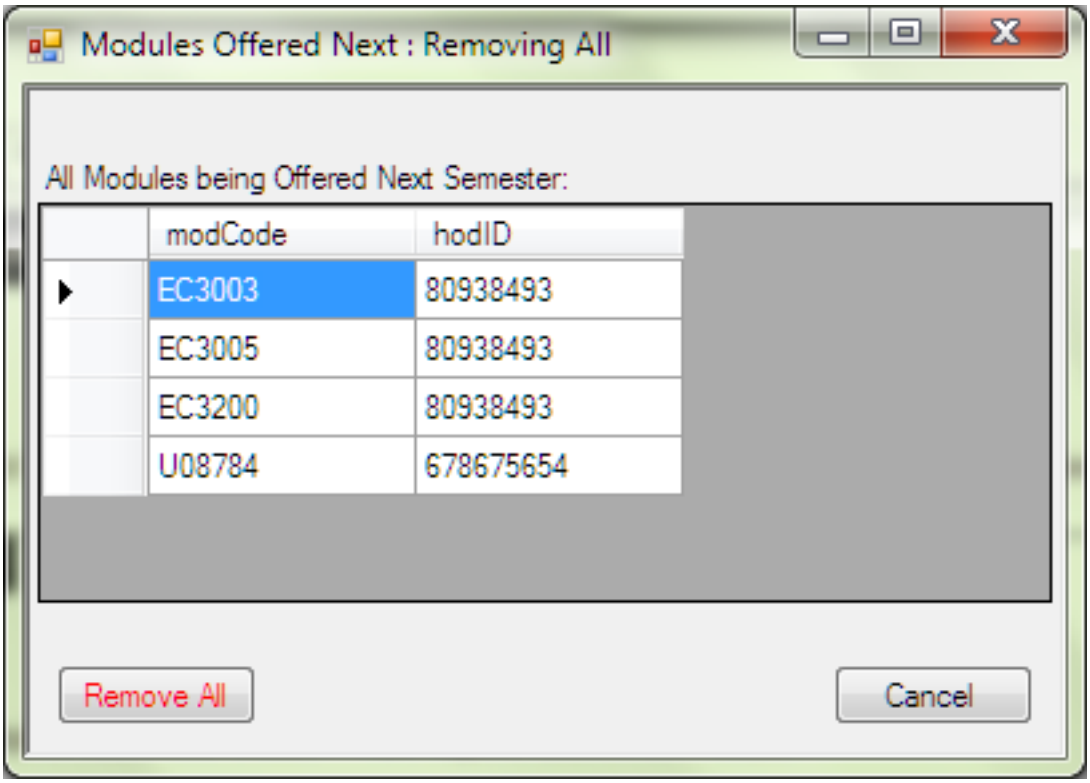
10.2 Backend Screen Shots

These are some of the screen shots for the backend part of the SCRS System.

10.2.1 View All Results

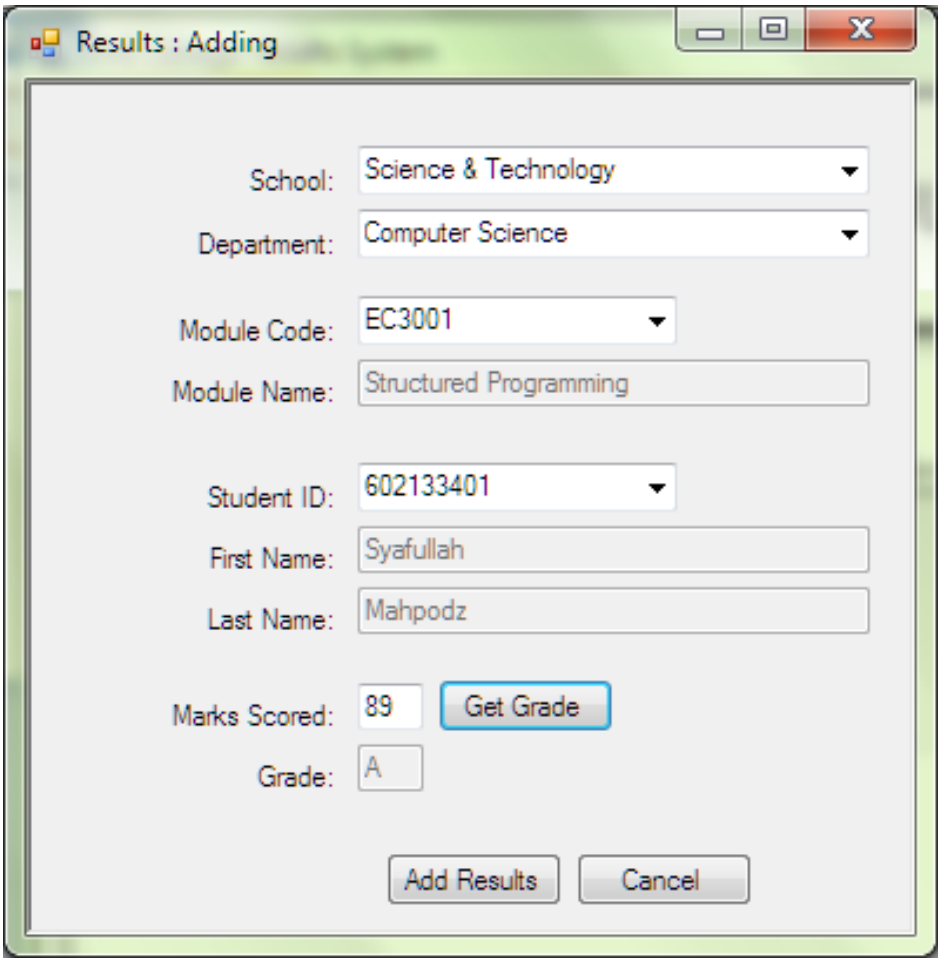


10.2.2 View All Modules Offered Next Semester

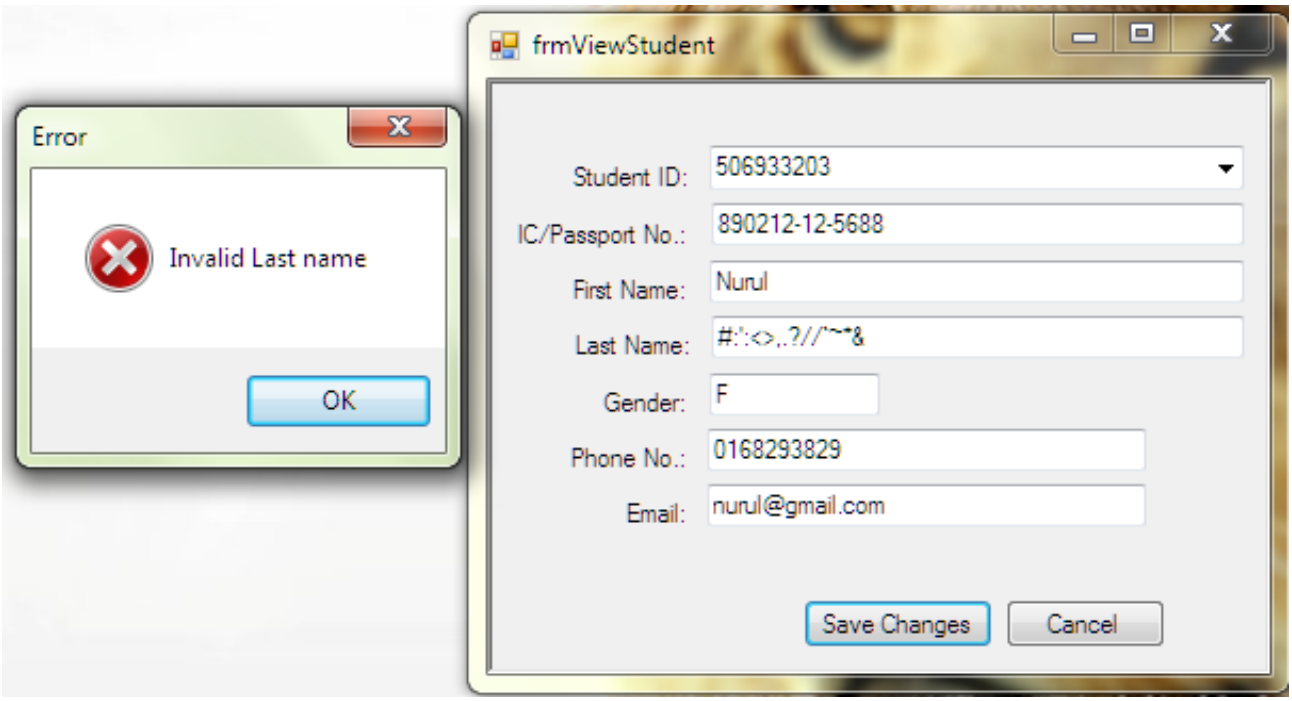


10.2.3 Add Results

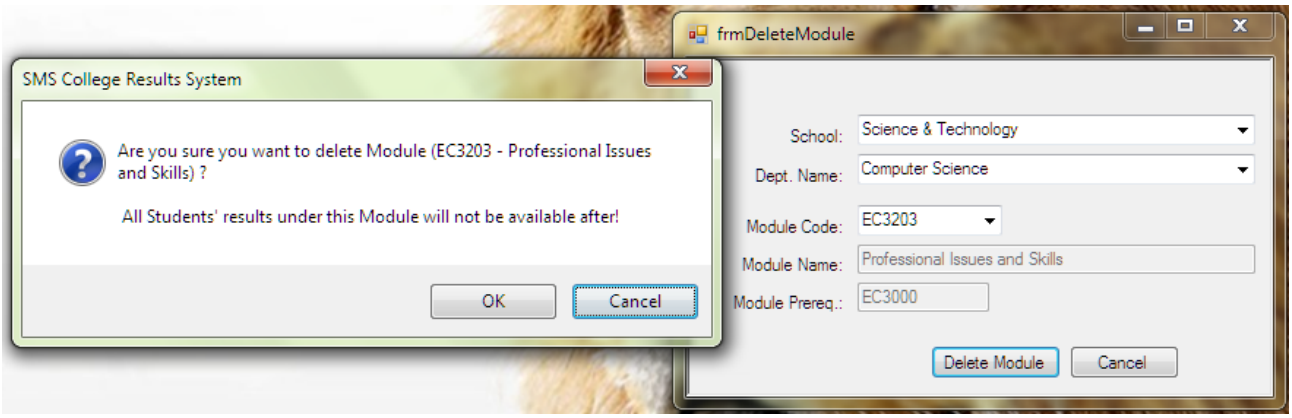
After entering marks scored and clicking “Get Grade” button, the system automatically calculates the Grade using the current OBU scheme.



10.2.4 Input Validation (Regular Expressions)



10.2.5 Delete Confirmation



10.2.6 Exit Confirmation