# Development Of An SMS Based Alert Systemusing Object Oriented Design Concept

Adewale Adeyinka. A., Abdulkareem Ademola, Adelakun Adebiyi. A.

**Abstract:** An automated lecture alert management system has been developed using java programming concept knownfor its portability. This backend system was interfaced with the GSM network through USB port of a PC and GSM modem. The desktop SMS application was developed using C# programming language. It generates updates and reminder from a time schedule stored in a database thereby making the system a time triggered application.

**Keywords:** SMS, GSM Module, Java, AT Command, MySQL

## 1.0 Introduction

In every organised system which is dependent on accurate time management, there is a need to have a system which can on its own serve the purpose of automatically sending alert text messages.Timemanagement plays important role in the success of any venture like a university environment where limited spaces (lecture halls), time and people have to be managed for effective lecture delivery. Itis important to remind or inform lecturers and students of lecture schedule so as to make them arrive early for their lectures. People can easily be reached through their handheld devices such as mobile phone irrespective of their location. Hence, there is the need to develop a system that will combine available resources and reach people via short message service (SMS) using GSM module [1]. Java is a portable object oriented programming (OOP) language which can be used to develop large-scale enterprise applications, to provide applications for devices like cell phones, pagers and personal digital assistants among many other purposes [6, 8]. In addition to using this OOP programming language for developing the application, C# programming language was also used to develop the desktop SMS application which generates updates and reminders from a time schedule based on contact details of members of staff and helps automate and document the process of sending information across an organised system at a reduced cost. The system architecture is shown in figure1 and it employed the use of SMS over circuit switched GSM network that allows the application to send and receive messages over the modem interface. The mobile operator charges for the message as if it was performed directly on a mobile phone by supporting an extended AT command set for sending and receiving SMS messages according to defined specifications [2, 9].

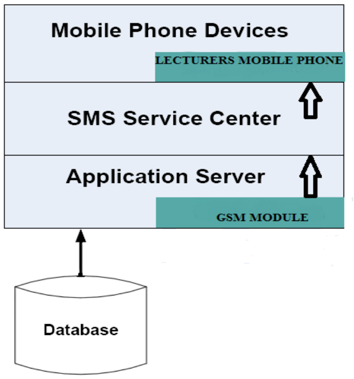


**Figure 1:** System Architecture

A GSM module can be a dedicated modem device or a standard GSM mobile phone with a serial, USB or Bluetooth connection and software driver to connect to a serial port or USB port of your computer or server as a cost effective solution for sending SMS messages [4, 9].The system is to function with a database which contained the contact information of each lecturer such as names and telephone numbers, along with other information such as the courses they handle, the start time of each lecture, location of the lecture and number of eligible students. This would be created by making use of MySQL and then installed to run on the server which would house the program. At the initiation of the program, the system is going to check the database to see which lecture on the time table falls within a 10 minutes start time and in the event of a lecture starting within the time; it forwards the constructed message to the GSM module for sending. The system block diagram is shown below in figure 2, followed by the procedure diagrams for the databsse and time table creation in figure 3 and figure 4 respectively. AGSM module is a type of modem with

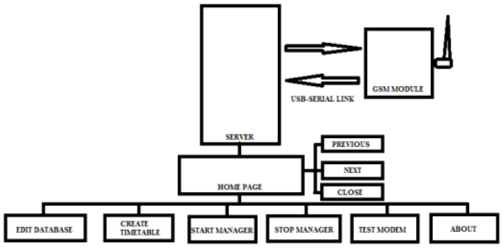


**Figure 2:** System Block Diagram

- *Adewale Adeyinka. A., Abdulkareem Ademola, Adelakun Adebiyi. A.*
- *Department of Electrical and Information Engineering, Covenant University, Ota, Ogun State, Nigeria*

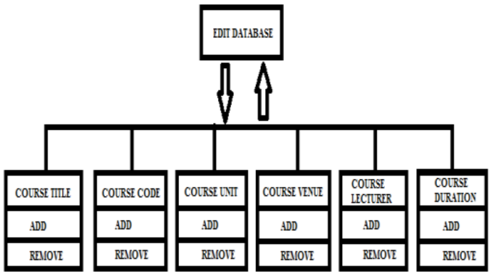


**Figure 3** Editing the Database

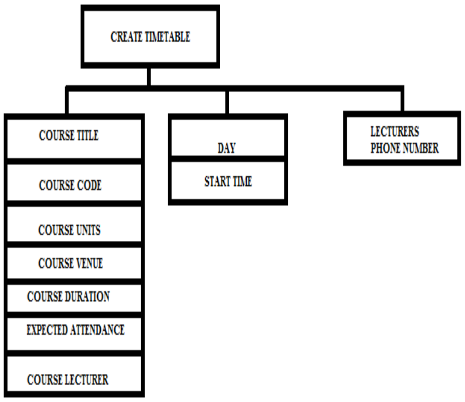


**Figure 4:** Creating Timetable Entries

provision to accept a subscriber identity module (SIM) card and operates over a mobile operator network just as a mobile phone would function. It can be a dedicated modem device with a serial, USB or Bluetooth connection, or simply a mobile phone that provides GSM modem capabilities. GSM modules are very similar to modems except that GSM modems are external equipment, whereas GSM modules can function as embedded pieces of hardware. They are designed to support an extended set of AT commands which are defined by the GSM standards. With these extended AT commands, one can read, write, delete SMS messages, send messages, monitor signal strength, charge level of the battery and search phone book entries.The following section describes similar systems that have been developed while this is followed by the application system design, implementation and a conclusion of the paper.

## 2. RELATED WORKS

The short message service (SMS) is a feature of second generation (2G) mobile technology and it must not be longer than 140/160 alphanumeric characters without images or graphics. It is a relatively simple messaging system supported by GSM, TDMA and CDMA based mobile phone networks. SMS based services are now more attractive to service providers and users as a result of the recent mobile phone use penetration and the large scale adoption of the existing services by users [4]. Several SMS based applications have been developed and are used for various purposes in fields such as health, education, business, security monitoring, supply chain management [7]. Systems such as “MobiRep” have been developed and it bridges the communication gap of reporting and responding to crisis, crime or an emergency. It simply accepts reports in the form of SMS from any mobile phone and can also broadcast bulk SMS to a group. Therecipient which in this case would be the police or any other emergency response unit can make use of the broadcasting feature incorporated into the system to take necessary actions such as dispatching resources to handle the distress call or notify people in the neighbourhood of imminent danger via SMS alerts [7]. This system is made up of four layers which areuser interface, business logic, object relational mapping and database layers. A layer can only interact with the layer next to it and no layer is allowed to interact with a layer it doesn’t have direct access to. This allows the codes to be readable, modifiable and maintainable. The business logic layer implements the business requirements and constraints without the user interface additional sub systems (SMS management, contact management and utility). Since the database was relational, an object relational mappinglayer was necessary to connect certain features of the database and the business layer. In [8] an SMS examination result notification system was designed to enable students have remote access to their examination results via mobile phones. Students query the database by sending SMS commands to the system to retrieve their results.NowSMS is a server software product which requires a connection to a GSM modem or to an external SMS service provider before it can be used to send any message. It is a fast and reliable way of deploying SMS and MMS content delivery solution. The software is suitable for any Windows platform and also functions as SMS and MMS Server Gateway, SMSC Multiplexer, 2-way SMS and MMS application enabler, Multimedia Message Service Centre (MMSC) WAP Push Proxy Gateway (PPG). Some of the client protocols used by NowSMS include: SMPP, UCP/EMI, CIMD2, HTTP, GSM Modem and the application connectivity for SMS submission or server side protocols are SMPP, HTTP andSMTP [1].

## 3. SYSTEM DSIGN

The system comprises primarily of a PC running the graphic user interface application and a modem interfaced with the PC over the Universal Serial Bus Port. A Sony-Ericcson K700i phone was used to send and receive SMS via its hardware serial port. The modem was configured for operation at 9600bps, 8 bit data size, no parity, and 1 stop bit (9600,8,N,1).The system flowchart is shown in figure 3. Once the program is initiated, it begins checking the database for a lecture which falls within the 10 minutes or less start time window, if one exists, the manager would begin collecting information from the database to generate the SMS which will be sent to the lecturer through the GSM module. Successful delivery of the SMS would be registered by the manager after which it begins checking for the next lecture. The manager was developed to accommodate a maximum of 256 lectures per day and is modifiable. Visual Studio interface was used to create the Windows form of the system and it functions with java class plug-in as the back end of the application.

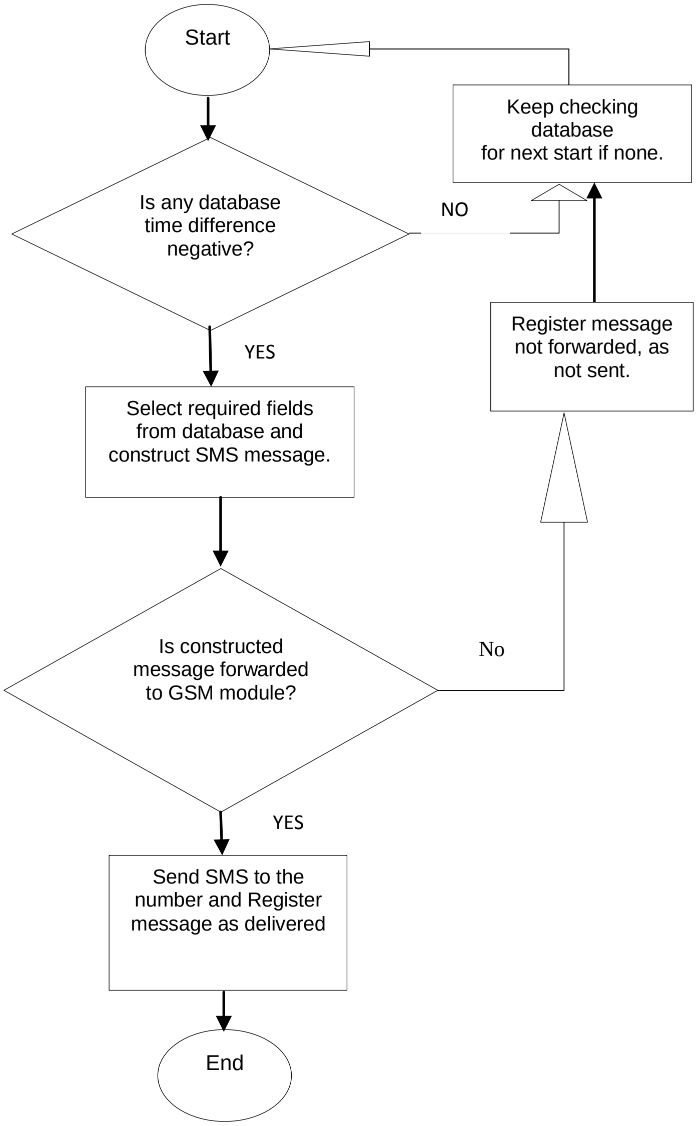

Start
Keep checking database for next start if none.
Is any database time difference negative?
NO
YES
Register message not forwarded, as not sent.
Select required fields from database and construct SMS message.
Is constructed message forwarded to GSM module?
No
YES
Send SMS to the number and Register message as delivered
End

**Figure5:** System flowchart

Since legacy ports are no longer provided on modern PC boards, virtual serial ports objects were created in software to interface with VxD (Virtual Device Drivers) packaged with the target Windows OS. This abstraction of the communication interface away from the user GUI is another important consideration in the utilization of the K700i engine.Accessing the serial modem device was implemented by instantiating the serial port class, setting its properties and accessing its methods to effect data transfer. For robustness, the modem was tested for operational conformity at every step of code execution. Port opening/closing was done via the serial port class methods. Determination of the modem's electrical presence on the virtual serial port was implemented by successively opening the available serial ports returned in a buffer populated by the OS hardware abstraction layer using the Getportnames() function invoked by the get_port_id() code block. The port addresses returned by the get_port_id function are opened successively, and the modem probed by sending the "AT" attention command. Resource deadlocks are prevented by making sure that the modem is correctly identified by the software before data is transmitted to it. Messages to be forwarded to recipients had to be converted into a meta-data format specific to the phone, or are otherwise rejected. The modem establishes connection with the provider network on which the Subscriber Identity Module (SIM) was registered. This is necessary for operation on the GSM communications network. Communication with the modem was effected using the Java TX/RX serial communication class. This class abstracts the hardware layer away from the higher level code block, providing an easily-upgradeable and reusable software component. An interface for testing the presence or otherwise of the connected modem was provide on the GUI, this enables any compatible modem interfaced with the GUI to be easily tested for functionality. Due to the proprietary nature of the modem, slight modifications have to be made if the GUI is to work seamlessly with other modems. Primarily, the K700i does not implement data format conversion as specified in the GSM 07.07 ETSI protocol in that the metadata sent as packets from the GUI is not strictly Hex-ASCII converted-septet PDU, but rather the Hex-ASCII equivalent of the IRA characters comprising the SMS body. In this format, each ASCII character is converted into its two-byte Hex-ASCII components and forwarded to the modem for onward transmission to the target recipient whose mobile number is pulled from the database. An example SMS Submit data packet is given below:

**Table 1: An SMS Submit Data Packet**

| 00:01:00:0B:81:7060903409F6:00:04:12:537769746368204E6F2E31206973206F6E21 | |
|---|---|
| HEX-ASCII | FUNCTION |
| 00 | Number of octets in Meta data, 0 means that modem should use stored Meta. |
| 01 | First octet in SMS-SUBMIT |
| 00 | Message reference, 00 means that the modem sets the reference number. |
| 0B | address length: 0x0B=11 |
| 81 | number type: national type |
| 7060903409F6 | Address of receiver |
| 00 | protocol identifier |
| 04 | data coding scheme |
| 12 | septet count: 0x12= 18 septets |
| 537769746368:E21 | user-defined text |

The SMS packet is pre-appended with the data length of the ASCII string less the size of the header metadata when forwarded to the modem. The actual SMS transmission is made possible using the AT command: AT+CMGS=<length of the ASCII string-2>. A successful transmission of the

SMS is reported by the modem with a +CMGS: <mr> response (mr is the message reference number that is used by the modem for identifying a particular submitted SMS when a delivery status report is returned from the network. In the GUI, the software only waits to receive the delivery reports for 30 seconds, after which it exits. If the SMS is reported delivered by the network, a checkbox against the particular lecture fixture for which the notification was sent is checked to indicate successful SMS delivery, otherwise it is left unchecked. Command-response interface between the modem as PC was made possible using AT command set for compatible modems [9]. Files were accessed (created, read, written, modified, or deleted) based on the file type. Binary files were accessed using the Binary reader/Binary writer instantiations, while text files were accessed via the Stream Reader/Stream Writer classes. The main line code responsible for maintaining periodic checks (every 60 seconds) on the lecture fixtures runs an infinite loop in which the database is accessed and relevant information extracted and the mainloop code sets up relevant portions of the GUI and populates some variable fields used by other code blocks. A subroutine keeps a running count of the SMS dispatch retries for each lecture record using pointers referencing a RAM array used to hold the SMS dispatch status of each record. The subroutine calls two main functions, get_lecture(…) and send_alert(…). Get_lecture_day returns true if there is lecture set for that day (moday,Tuesday….Sunday), otherwise false. the arguments passed to the function are accesed via pointers, and are used by the send_alert function. Get_lecture function also determines if the lecures due for the system day are within the SMS alert window of 10 minutes by comparing the lecture start time with the system time in a separate function. The send_alert function takes the lecture(s) within the SMS alert window and sequentially acceses their records, extracts the data that make up the SMS body and references a separate code block that transforms the ASCII data into the modem-specified format, along with the necessary encapsulations. To adapt the plain text ASCII SMS to modem-acceptable format, software re-formatting was employed. The re-formatting required of the GUI involved:

- The reversal of the recipient's mobile number address as specified by the GSM protocol,
- Appending the recipient's mobile number with an ASCII 'F' if it is an odd-length number,
- Addition of manufacturer-specific data to the string
- Conversion of the plain text SMS message into two-byte Hex-Ascii representations.

Arguments passed to the function includes the RAM space designated to hold the formatted header data, the recipient's mobile number, and the length of the actual user-readable portion of the SMS message, that is the message displayed on the recipient's mobile phone. The function returns the length of the header packet to the caller routine. The header packet is concatenated with the user message re-formatted as a Hex string and passed to the write_modem function that takes as arguments the header data, and its length.

## 4. IMPLEMENTATION/TESTING

Once the automated lecture alert management system has been installed, and the connection between the GSM module and PC/Server tested, launched from the Program Menu, the GUI interface of the system shown in Figure 4.2 below comes up.

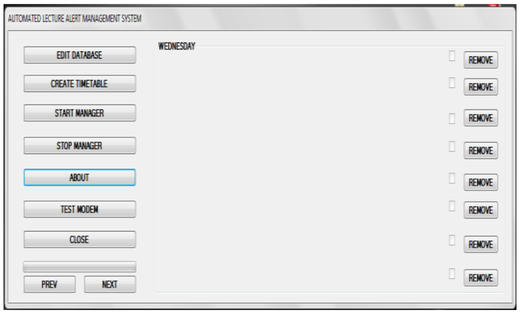
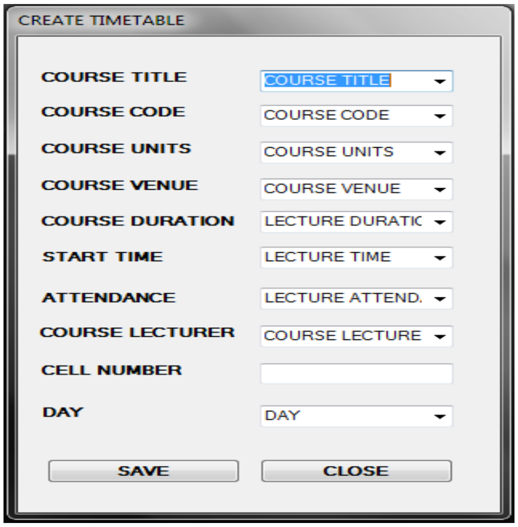


**Figure 6:** Program GUI

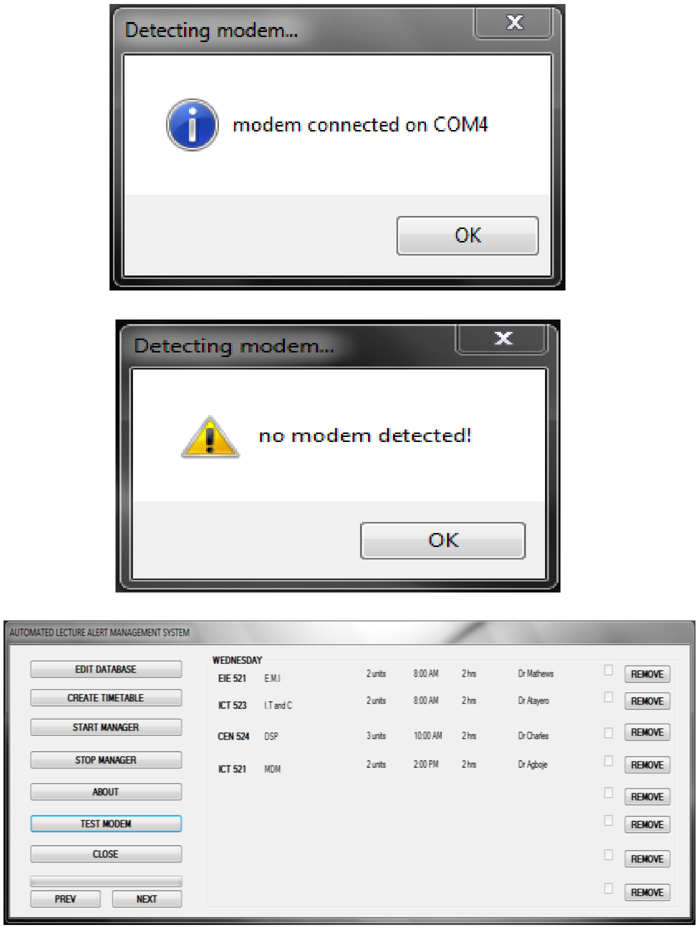
The "EDIT DATABASE" button provides you with the ability to configure the database and populate the various tables of the database with information. It is under this interface that the start time for each lecture and the days in which they are to hold can be selected. Also, the number of students that have registered and are expected to being in attendance can be selected by the administrator to be sent to the course lecturer using the inputted phone number.



| | |
|---|---|
| COURSE TITLE | COURSE TITLE |
| COURSE CODE | COURSE CODE |
| COURSE UNITS | COURSE UNITS |
| COURSE VENUE | COURSE VENUE |
| COURSE DURATION | LECTURE DURATIC |
| START TIME | LECTURE TIME |
| ATTENDANCE | LECTURE ATTEND. |
| COURSE LECTURER | COURSE LECTURE |
| CELL NUMBER | |
| DAY | DAY |

**Figure 7:** Create Timetable Interface

Clicking the "START MANAGER" button initiates the looping process of the manager to begin checking for the next lecture and the "STOP MANAGER" button ends this process.The "TEST MODEM" button enables you to determine whether or not a connection has been established between the modem and the program. Once a connection has been established through a USB to serial cable, it would inform the administrator and also specify which of the ports was used to establish the connection. The GUI works tightly bound to the modem's technical specifications. Hence, using a different modem will almost certainly necessitate modifying some portions of the manager software to accommodate differences in modem's hardware platforms.



Detecting modem...

modem connected on COM4

Detecting modem...

no modem detected!

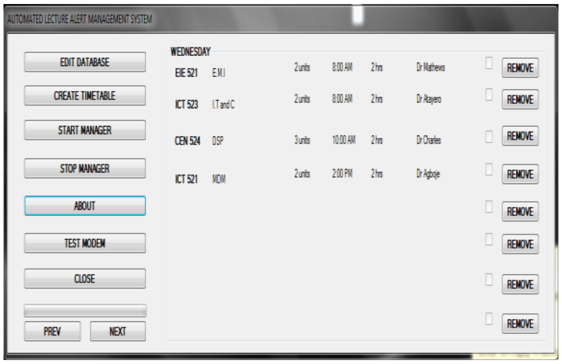
AUTOMATED LECTURE ALERT MANAGEMENT SYSTEM

WEDNESDAY

| | | | | | | |
|---|---|---|---|---|---|---|
| EIE 521 | E.M.I | 2 units | 8:00 AM | 2 hrs | Dr Mathews | □ |
| ICT 523 | I.T and C | 2 units | 8:00 AM | 2 hrs | Dr Atayero | □ |
| CEN 524 | DSP | 3 units | 10:00 AM | 2 hrs | Dr Charles | □ |
| ICT 521 | MDM | 2 units | 2:00 PM | 2 hrs | Dr Agboje | □ |

**Figure 8:** Testing Connection to Modem Interface

Once the database has been populated with information and the time table has been created, it would appear in the form shown below. Aside from displaying all the lectures scheduled to hold for that day, this interface also shows whether the alert has been sent to the lecturer by marking the field as sent once it receives a delivery report. The binary reader/binary writer class was used for accessing the binary portion of the database as some entries had to be saved as binary files. These entries include the SMS sent/delivery status.



AUTOMATED LECTURE ALERT MANAGEMENT SYSTEM

WEDNESDAY

| | | | | | | |
|---|---|---|---|---|---|---|
| EIE 521 | E.M.I | 2 units | 8:00 AM | 2 hrs | Dr Mathews | □ |
| ICT 523 | I.T and C | 2 units | 8:00 AM | 2 hrs | Dr Atayero | □ |
| CEN 524 | DSP | 3 units | 10:00 AM | 2 hrs | Dr Charles | □ |
| ICT 521 | MDM | 2 units | 2:00 PM | 2 hrs | Dr Agboje | □ |

**Figure 9:** Main GUI Form with Lectures Inputted

The software was tested exhaustively by creating plenty lecture fixtures with varying start times and monitoring the dispatch of SMS to the intended recipients through the GSM module. The SMS were successfully delivered with only very few exceptions as a result of network error in which case the software would attempt to resend the SMS. The SMS send retry window was set at 2, that is if the SMS is not sent at the first attempt; its sending would be retried on the next loop within the ten minutes time bracket and before the start of the lecture. If after the second attempt the sending fails, the entry is tagged as sent because it was forwarded by the program but not delivered because the GSM module did not receive a delivery report after which the GUI gives up retrying.

## 5. CONCLUSION

There is an increase in the need for automated reminder/alert system which can on its own generate alerts and send to specific recipients in many organisations. It was explained how time triggered commands were generated by the system to send SMS messages to recipients and how the database of the system functions and the system was interfaced to function with a Sony Ericcson K700i phone which was the selected GSM module used. Also, successful delivery or failure of the SMS sent via the module. However, future research could be to see the system work with its own time reading and not depend upon the PC for its time readings.Provisions could be made to enable users (recipients of SMS) query the system for information at any time and then the system reply them instead of having to wait for the next loop. Meanwhile, in comparison with SMS sent from mobile phones, an average time difference in delay of 20s was observed for the SMS delivered under the same coverage area and the same service provider while it took averagely 50s to receive delivery feedback from a different service provider network. This delay variation can be accounted for as processing delay from the PC and the GSM module.

## 6. REFERENCES

[1]. Now SMS/MMS Gateway June 8, 2011 Edition http://www.nowsms.com

[2]. Heikki Kaaranen, et al, UMTS Networks Architecture Mobility and Services, 2nd Edition 2005, Publisher: John Wiley.

[3]. SMS Centre API, Multi Messaging Software Server SMS/MMS/IVR/WAP Push, May, 2004

[4]. Gwenael Le Bodic, Mobile Messaging Technologies and Services: SMS, MMS, EMS, ISBN: 978-0-470-01143-0, February 2005.

[5]. Simon So, "The development of a SMS-based teaching and learning system". Journal of EducationalTechnology Development and Exchange, 2(1), 113-124, (2009).

[6]. P.J. Deitel and H.M. Deitel,JAVA How to Program, 7th Edition, Upper addle River, New Jersey 07458.

[7]. Yonan Getachew , “Design and implementation of Crisis and Emergency Reporting and Response system using SMS” Addis Ababa University Digital Library, June 2009

[8]. John Paul Ambu, Sridevi Jetty, “Control Alert System Using SMS: A Testimony of two Collaborative Projects in Africa and Asis”, Proceedings and reports of the 6th UbuntuNet Alliance, annual conference, ISSN 2223-7062, 2013, pp5-12.

[9]. Sony Ericsson Developers Guidelines Online Reference, August 2005

[10]. Robert Virkus**,** Pro J2ME Polish: Open Source Wireless Java Tools Suite, August, 2005

[11]. Todd Herman, Allen Jones and Matthew MacDonald, Visual Basic 2008 Recipes, a Problem-Solution Approach, 2008.