Sudan University of Since and Technology

College of Graduate Studies

Short messages technique in health level 7(HL7)

تقنية الرسائل القصيرة فى النظام الطبى

A thesis Submitted in Partial Fulfillment for the Requirement of the Degree of MSC in Biomedical Engineering

By:

Manahel bushra Hassan

Supervised by:

Dr .Abd Alrasool Al-Zebaidi

2014

Dedication

To my Mother,

Your kind support provided us to achieve our dreams

Brothers &sisters,

Friends &colleges,

With my love

Acknowledge

I would like to thanks all those whom assist me to complete this work

Special thanks to ***Dr.Abd Arasool AL zebidi*** *who supervised this project to his support and efforts.*

Abbreviations

Abbreviation	Meaning
HL7	Health level 7
OSI	Open system interconnection
DFT	Detailed financial transport
HIE	Health information exchange
MSH	Message header
PID	Patient identification
OBR	Observation report
OBX	Observation /result
GSM	Global system for mobile communication
HER	Electronic health record
CEN13606	Committee European norm
CDA	Clinical document architecture
RIM	Reference information model
VB	Visual basic language
NEHTA	National E-Health Transition Authority

List of figures

المستخلص

أوضحت الدراسات أن إستخدام تكنولوجيا المعلومات فى صناعة الصحة قد أرفدت المجال الصحى بالعديد من الفوائد أهمها التحسن النوعى بالعناية الطبية وتقليل الاخطاء الطبية ,على الرغم من أن إستخدام تكنولوجيا المعلومات فى هذا المجال جاء مؤخرا ويعزى هذا التاخير لعدة عوامل أهمها مدى الاستفادة من التقنية مقارنة مع تكلفتها.

إحدى التقنيات هو hl7وهو برنامج يعنى بتبادل المعلومات الصحية بين مؤسسات الرعاية الصحية وسنركز هنا على ارسال واستقبال المعلومات الاولية للمريض.

فى هذا البحث قدمنا كيفية ارسال واستقبال هذة المعلومات حيث تم استخدام GSM modemبه شريحة تستخدم شبكة الاتصالات.

يتم إرسال وإستقبال الرسالة الطبية باستخدام شبكة الاتصالات وقد تم عمل قاعدة بيانات تشتمل على المعلومات الاولية للمريض والمراد ارسالها وتمتلك قاعدة البيانات هذه السماحية باضافة المرضى الجدد ومعلوماتهم والتعديل فى بياناتهم .

تم إهمال عامل المسافة بين نقطتى الإرسال والإستقبال وذلك للإعتماد على شبكة الاتصالات.

Abstract

Proper management of health information is an important in providing safe and efficient patient care. Recent studies shown that adoption of IT (information technology) in health care participate in improving the quality of care, preventing of medical errors, reducing of health care cost. The slaw adoption of it in health care can be result of many factors like doubt of benefit vs. cost of it.

One of information technology systems is hl7 (health level 7) which is a software program responsible for exchanging medical informatics data in form of messaging here the data shall be send is patient identification.

In this research we present the sending and receiving process how this message has been sending and how receiver deals with the message.

We use GSM modem system this system has built in GSM modem and including SIM card using telecommunication network. Also we made data base for the system including patient information must be send this data base has the ability of addition the newer patient.

Distance factor shall be ignored because of the dependency of mobile network.

Table of content

Chapter tow

Chapter three

Chapter four

Chapter five

Chapter six

Chapter one

Introduction

Introduction

Employment of information technology in health care is essential and play important role in health care development. Now a days many software programs are developed to support communications in health care fields one of this programs is health level7 (HL7) which produce a standard for hospital information systems.

Health level 7 is series of pre defined logical format for packaging health care data in the form of messaging to be transmitting among computer systems has a mission to provide standard for the exchange ,management and integration of data that supports clinical patient care. Thus it is a messaging protocol specifically developed to exchange health, medical patient information between information systems.

These standards define how information is packaged and communicated from one party to another, setting the language, structure and data types required for seamless integration between systems. HL7 standards support clinical practice, management delivery, evaluation of health services, hl7 recognized as the most commonly medical data exchange software used in the world.

Health level 7(HL7) developed to provide a common language for Health/Medical-based information systems to use when sending information to each other.

HL7 stands for Health Level Seven. The Level Seven refers to the seventh layer of the open system interconnection (OSI) communications model as shown in figure 1.1.

1.1.1 Health level 7 layers

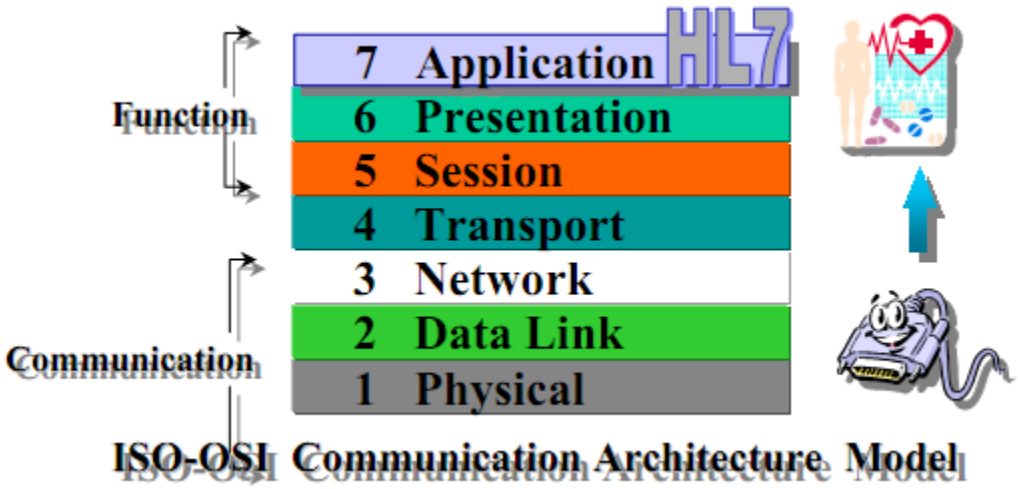

Figure 1.1 hl7 layers

Physical layer It provides the hardware means of sending and receiving data on a carrier.

Data -link layer This layer provides synchronization for the physical level. It furnishes transmission protocol knowledge and management.

Network layer This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does routing and forwarding.

Transport layer This layer manages the end-to-end control (for example, determining whether all packets have arrived).

Session layer This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with

session and connection coordination -checking. It ensures complete data transfer.

Presentation layer This is a layer, usually part of an operating system that converts incoming and outgoing data from one presentation format to another(2)

Application layer This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is *not* the application itself, although some applications may perform application layer functions [2].

As we know hospitals and other healthcare provider organizations typically have many different computer systems used for everything from billing records to patient tracking. All of these systems should communicate with each other (or "interface") when they receive new information but not all do so. Hl7specifies a number of flexible standards, guidelines, and methodologies by which various healthcare systems can communicate with each other. Such guidelines or data standards are a set of rules that allow information to be shared and processed in a uniform and consistent manner. These data standards are meant to allow healthcare organizations to easily share clinical information. Theoretically, the ability to exchange information should help to minimize the tendency for medical care to be geographically isolated and highly variable.

Hear we focus on HL7 version 2 which defines a series of electronic messages to support administrative, logistical, financial as well as clinical processes .it deal with sending package of information in the form of message from one station and receive it in other station .in our project we ignore the distance factor because we depend on mobile network.

1.1.2 Common use of hl7

HL7 has different and multiple uses that lead to improve health care as shown in figure 1.2.

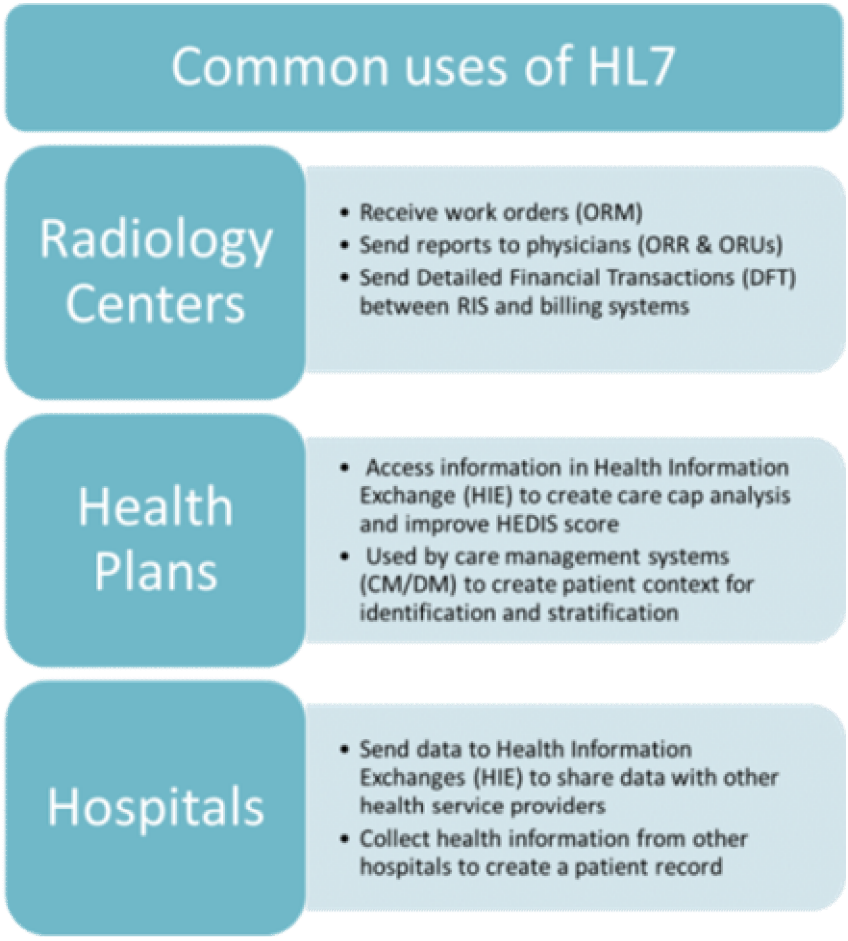

Figure 1.2 hl7 common uses

1.1.3 HL7 Technical Details

1.1.3.1 HL7 Protocol Characteristics

HL7 is a structured protocol that is used as a means of communicating between healthcare applications. The following descriptions provide an outline that defines the characteristics of the HL7 protocol [4].

1.1.3.1.1 Event-Driven

Real-world events, such as the admission of a patient, cause messages to flow between applications. In other words, an application that encounters a real-world event sends a message to other applications that need to be aware of this event [5].

1.1.3.1.2 Application-to-Application

HL7 defines a communication between two independent applications, rather than between closely coupled, client/server applications. The scope of interest for HL7 is the message exchange between the applications, rather than the specific role of each application in the healthcare delivery process [6].

1.1.3.1.3 Point-to-Point

HL7 is a messaging format that is independent of its transport method. However, it is typically used in a client/server environment for employing some form of a point-to-point protocol [7].

1.1.3.1.4 OSI Level 7

OSI Level 7 indicates that HL7's scope is the format and content of the data exchanged between the applications rather than how the messages are transmitted between computers or networks. In other words, HL7 does not specify how messages will be delivered between the applications [8].

1.1.3.1.5 Data Exchange

HL7 specifies the way data exchange between applications will be accomplished. It does not specify how applications store or process this data. It would be advantageous for an application developer if their application's

database structure coincided with HL7 message definitions, but this is not mandatory [9].

1.1.4 Hl7 message

HL7 Messages are used to transfer electronic data between disparate healthcare systems. Each HL7 message sends information about a particular event such as a patient admission [10].

1.1.5 Type of segment

Each HL7 message is of a particular message type. This HL7 message type indicates what health-related information is being provided in this message. The message type also determines what segments can be included as part of the message.

MSH this contain information about the sender and receiver of the message, the type of the message, time stamps, etc.

PID patient identification this contain patient identify and demographic information about the patient such as name, date of birth, sex.

OBR observation report this contain information about the observation order.

OBX observation /result this contain information of singles, indivisible units of observation or report and its repeatable [11].

There are some other segments which serve the other purpose.

Here we will focus on the patient identification segment (**PID)** and we will send it from one health care point to other health care point. We will send basic information about the patient such as name, old, insurance and basic diagnostic.

1.1.6 Why health level 7 for

1.1.6.1 Need for integrated systems

As shown in figure 1.3 bellow the need for communication between doctors and their patients.

Figure 1.3 needs for integration system

1.2 Problem statement

As we know and mentioned before the real need of implement the information technology in medical care industry, this information technology is costly but it has great benefits and supports the medical care industry by this benefits.

Health level 7 needs seven layers in building from physical layer to application layer which lead to high cost in the software design, actually it is very expensive to build all the layers of the system for this issue the need to simulate the function of hl7 is appear to minimize the cost and to pick up the benefit of the ordinary software.

1.3 Objective

The General objective of the research is to :

1-To minimize the cost of important communication software by developing new method with cheapest component.

2- To reduce the hardware means and cables required for building the network by depending on telecommunication networks.

3-To employ the familiar language to make communication.

Specific objective:

1-To create messaging technique in hl7.

2-To Support communication system in health field.

3-To enable medical informatics data to be transfer between health units.

4-To Safe time by depending on software in communication rather than man work.

5-To connect all hospital parts with network with information sharing.

6-To connect two or more hospital with each other for medical record transfer.

1.4 Thesis layout

Chapter one will contain the Introduction of the topic then the previous studies shall be introduced in chapter two with the title literature review then the system identification appear in chapter three the system component and flow chart. The second part of the thesis contain the result and discussion in chapter four, conclusion and recommendation in chapter five and finally in chapter six the references and appendix.

Chapter two

Literature Review

Literature Review

Here two studies are introduced .the first study compare between three standards HL7 (health level7), CEN 13606(Committee European Norm) and EHR (electronic health record).the second study compare between the structures of hl7 two type of interface are introduced the point to point interface and the interface engine.

2.1 The relationship between CEN 13606, HL7, and open EHR

2.1.1 The importance of interoperability

The development and adoption of national and international standards for EHR interoperability is essential for sharing patient health information between health professionals in a multi-disciplinary shared- care environment also it is important for Interoperability between organisations within an enterprise, a regional or national health system also it is participate in supporting interoperability between software from different vendors.

2.1.2 Open EHR electronic health record architecture

2.1.2.1 Open EHR and standards

Open EHR is an open, detailed, and tested specification for a comprehensive interoperable health information computing platform for the EHR and other major services such as terminology it is based on 15 years research, focussed engineering design and real-world implementation experience [13].

2.1.2.2 Open EHR and interoperability

The open EHR model and specifications provide both functional interoperability and the far more demanding semantic interoperability

computers can understand and automatically process transmitted health information).semantic interoperability is essential to enable local processing of the shared data; this is a pre-requisite for intelligent decision support and care planning which are arguably the most important added-value applications of the EHR.

2.1.3 The scope of CEN 13606

The CEN/ISO EN13606 is a European norm from the European Committee for Standardization (CEN) also approved as an international ISO standard. It is designed to achieve semantic interoperability in the electronic health record communication.

It is important to note that CEN 13606(committee European Norm) is a specification for exchange of EHR Extracts; there are many more requirements for an EHR system, including version management, workflow management, interfaces to other systems etc. Some of these requirements impact upon the Extract. For example, because the CEN EN13606 Extract does not yet cater properly for version information, it cannot be used between open EHR systems, or any other systems wanting to preserve versioning information. Also, the specification is somewhat of a "lowest common denominator", which in a sense all standards are. 13606 is not the place to expect to see all the requirements one has for EHR systems, only those relating to moving pieces of the EHR from one system to another. Open EHR not only provides a specification for the communication of EHR Extracts of various levels of complexity but also a full specification for the creation, storage maintenance, and querying of EHRs. Neither CEN 13606 nor the HL7 RIM (reference information model) defines a reference model for an HER, and now includes some features that are very important for implementation but not included in CEN 13606[13].

2.1.4 Health level 7

2.1.4.1HL7 Clinical Document Architecture (CDA)

The HL7 Clinical Document Architecture (CDA) is HL7's current main strategy for EHR interoperability, the clinical statement is a similar initiative for use with HL7v3 messages. Neither of these are an EHR specification however, both correspond to an important component of the EHR. At present neither supports archetypes, although there is now an international effort to overcome this. HL7 CDA is approximately a subset of the 13606 EHR Extract, limited to one version of one composition, with some minor differences.

HL7 version 2 is not based on any underlying reference model and is therefore not suitable for EHR architecture. However, it is appropriate for transmitting clinical and necessary demographic/administrative information from source clinical information systems to a Shared-EHR system HL7v2.x messages have been recommended for this purpose in the NEHTA report, at least in the short- to -medium term.

in compare with hl7 version 3 Whilst HL7v3 is based on a reference model (the RIM) it is designed primarily for the messaging environment and does not currently have a specification for an EHR messaging environment and does not currently have a specification for an EHR architecture. It was not recommended in the NEHTA Shared EHR standards report because it would be more complex and costly [13].

2.1.5 Comparison between CEN 13606, HL7, and open EHR

HL7v2.x messaging is an appropriate standard, at least for the short to medium term, for transmission of information from source clinical information systems to a Shared-EHR system HL7 CDA may also be suitable for this purpose at

some later stage. CEN EN13606 is an appropriate standard for the exchange of Shared EHR Extracts between different nodes of a multi node Shared-EHR system (e.g. the original national Health Connect concept) or between different Shared-EHR systems [13].

However, there is currently no standard available for a Shared-EHR system which supports the creation, storage, maintenance, and querying of Share EHR.

Open EHR is the only open specification currently available which is a candidate for this purpose. It has been suggested that CEN EN13606 could be extended to achieve this purpose but whilst this may be possible, there is currently no plan within CEN to do this accordingly [13].

The second paper compare between the design of hl7 itself in terms of interface design the point to point interface and interface engine.

2.2 The need of interface engine

The typical clinical facility or hospital has several HL7 enabled applications and devices to reduce data entry time and increase overall efficiency of the facility, these applications or devices need to communicate with each other.

There are two basic ways this can accomplished:

1. Point-to-point where each pair of applications communicates independently of other applications.

2. Using an interface engine that is placed between all the applications to aid in information exchange and monitoring [14].

2.2.1 Point-to-Point Communications

The point-to-point communications model has each pair of applications communicating independently of the other applications in the environment through the use of export and import endpoints as shown in figure 2.1. These endpoints are attached to an application to allow it to send and receive hl7 message .

Typically, the export and import end point are required to:

- Filter data being sent or received.
- Change the format of data to match the receiving applications needs.
- Write the data into the receiving application database.

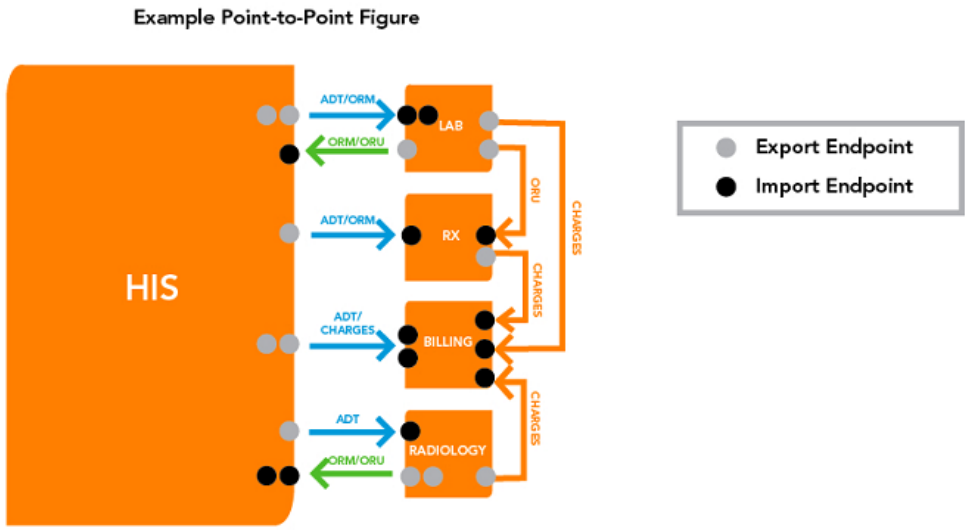

Figure 2.1 point to point interface [14]

2.2.2 Communication with an interface engine

An interface engine is designed to simplify connection, maintaining, monitoring and sharing data between interfaces. An interface engine can take data from

sending application and filter it or change the format of the data to match each individual application needs as shown in figure 2.2 [14].

This study with the title why do I need interface engine it is an evaluation between two approach points to points and interface engine.

The presence of an hl7 interface engine in a health care environment give more control to organizations and save money and time by:

- Reducing the required number of export and import end points
- Allowing for reuse of data between application
- Providing an easier method to interface anew or replace application
- Providing the ability to proactively notify interested persons using visual display and e-mail, when problem arise [14].

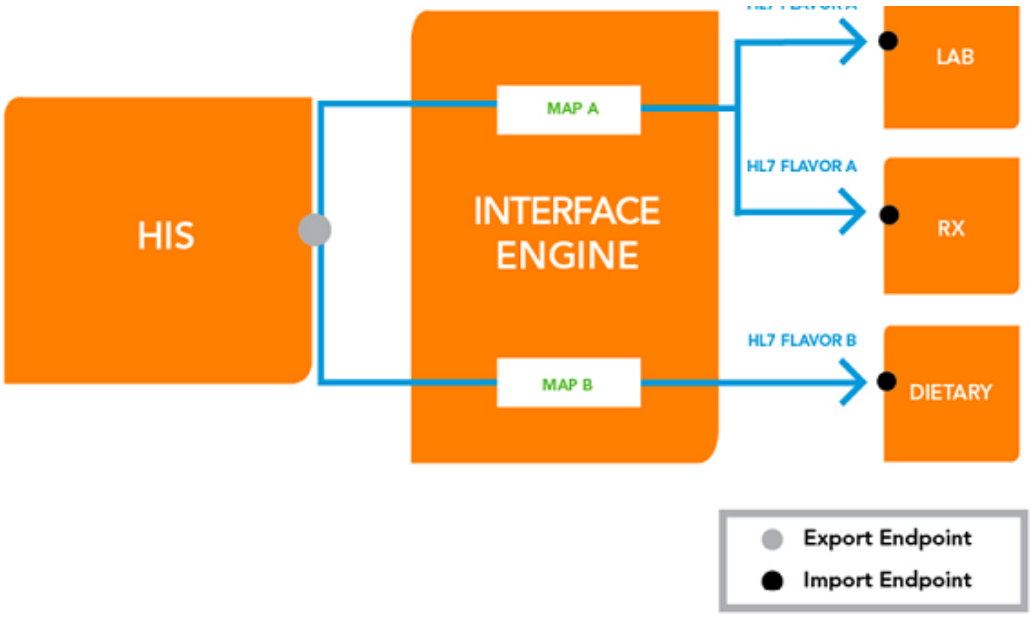

Figure 2.2 Engine filtering and mapping data [14]

The purpose of this white paper provides details on how an HL7 interface engine provides more control and saves time and money in a clinical or healthcare environment it includes a comparison of an interfaced environment using point-to-point communications and an interfaced environment using an interface engine.

2.2.3 Creating communication

To facilitate communication between two HL7 points like HIS and lab as shown in figure 2.3 enabled two applications to interface and communicate hl7 interface includes:

1-An export end point for sending application

2-An import end point for the receiving application

3- A method of moving data between the two end points.

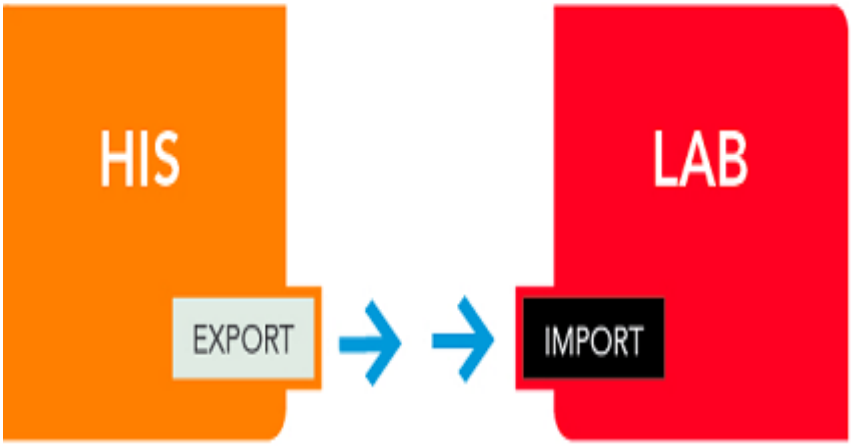

Figure 2.3 hl7 interface [14]

2.2.4 Comparison between interface engine and point to point interface

2.2.4.1 Facilities with point-to-point communication

Facilities that use point to point model find that:

1-It is very expensive to initially implement communication.

2-It very expensive to add new or replace existing application.

3- They spend considerable time and money to monitor and maintain the connection over the life of the application.

2.2.4.2 Facilities with interface engines

1-facilitise that use interface engine find that it is much less expensive and take less time to initially implement an interface because an engine allows for leveraging and it is flexible in it is acceptance of data.

2- Cost and time required to add new or replace existing application is frequently less than half that required in point to point model.

3-It required considerably less time and money to maintain and monitor the interface because of the availability centralized of monitor [14].

Chapter three

System component

System Component

The system is consist of two parts hardware and software the hardware consist of GSM (system or modem), SIM card and computer system. Software part employ visual basic language which is the communicating language employed to create the link between transmitter and receiver.

3.1 GSM System

GSM (Global system for mobile communication) is a wireless system has built in GSM modem.

3.1.1 Configuration of the base

There are 10 connector outside the back of base:

(**GND; SIREN; RELAY1; RELAY2; SPEAKER; O2; O1; I3; I2; I1**)

I1; I2; I3, this 3 point for line input, every one point can be connected with ground or open to make alarm out.

O1; O2; these 2 point for output, you can call in or send the SMS to set it. If this point output goes high, the lamp of OUT 1 or OUT 2 will light in the panel.

SPEAKER; this point for voice output, it connect to the speaker .the other point of speaker connect to ground. You can control it by call in or send the SMS. If this function avails, the lamp of OUT 3 will light in the panel.

RELAY1; RELAY2, this two point will close 3 minute when alarm happened .you can use this two point to start the power supply of the camera when alarm happened, but you can disable this function by set 16#1#,you can also make this relay close or open anytime by send the SMS or dial (94#1# or 94#0#).

SIREN, this point can output siren tone alarm, this connect to the siren .the other point of siren connect to ground.

GND, power ground

This GSM has built in GSM modem and it has SIM card inside it which responsible of the use of telecommunication network sending and receiving message is done via this SIM card and it is net work [15].

3.1.2 Setup

1. Delete all telephone numbers, messages or other data from the SIM card memory if it is not a new card and insert the SIM card.

2. Connect the antenna.

3. Insert the 1000mA, 9V—12V DC power supply. The signal status LED will turn red for 20 seconds, indicating that the unit is waiting for wireless detectors. , After 20 seconds, the signal status LED will flash orange, indicating the unit is checking the SIM card and searching for a GSM network. When it's finished its check the signal status LED will begin to flash green slowly. This indicates that the master phone number has been saved, and is armed. If the signal status LED goes solid green, it indicates that the master phone number has not been saved and the base is disarmed. You must dial in or send an SMS to program the phone number. Use the hand held controller to arm or disarm [15]

3.1.3 SOME TECHNICAL PARAMETER

1- Power: 9V-12V DC

2- Wireless frequency: 315/433/868/915MHZ

3- Maximum wireless distance: 100 M

Figure 3.1 GSM systems [15]

3.2 GSM modem

AGSM modem is type of modem which accepts a SIM card. When a GSM is connected to computer these allow the computer to use the GSM modem to use the GSM modem to communicate over the mobile network while these GSM modems are most frequently used to provide mobile internet connectivity, many of them are used for sending and receiving SMS and MMS messages.

A GSM modem can be can be dedicated modem device with a serial, USB or Bluetooth connection, the term GSM modem is used as generic term to refer to any modem that support one or more of the protocol in the GSM family thus GSM expose an inter face that allow application such as SMS to send and receive message over the modem interface.

GSM modem can be quick and efficient way to get start With SMS in most parts of the world ,GSM modem are cost effective solution for receiving SMS message [16]

Figure 3.2 GSM modem

3.3 SIM card

SIM (Subscriber Identity Module) card is a small, electronic memory unit used in conjunction with cell phones that use the GSM wireless signal. This SIM card houses the contact information, as well as other data you may have elected to save to the card, such as text messages and address listings [17].

Figure 3.3 SIM card

3.4 Computer

A computer is a general purpose device that can be programmed to carry out a set of arithmetic or logical operations. Since a sequence of operations can be readily changed, the computer can solve more than one kind of problem [18].

The computer can be used has this technical specification:

- Processor Pentium 4 or higher.
- Ram 128 MB or higher.
- Free memory space 2 MB.

- Operating system windows XP or higher.

3.5 Visual basic

The interface program it is made of visual basic language (visual basic.c sharp) which is designed to be easily learned and used by beginner programmers.

The language not only allows programmers to create simple GUI (graphical user interface) applications, but to also develop complex applications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue [19].

3.6 System flowchart

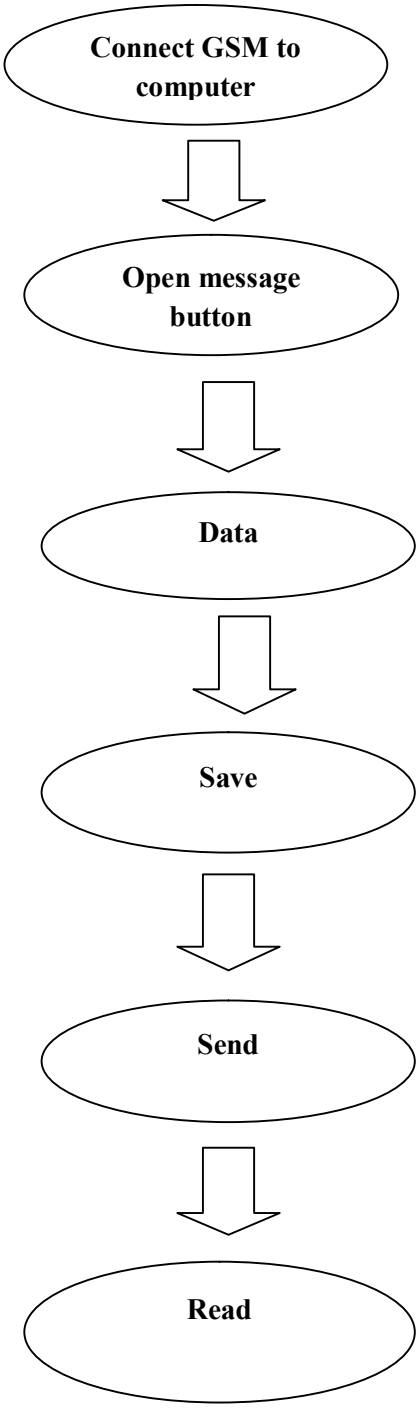

Figure 3.4 System flowchart

Chapter four

Results and Discussion

4.1 Results

Sending and receiving package of data or message especially medical informatics data is not an easy thing because medical data must be accurate always this accuracy leads for right medical decisions and reduce medical errors.

Here we search to find method for transferring (sending and receiving) medical informatics data these data can be medical images or medical reports or any other data here our targeting is PID (patient identification).

The transferring method that used in these research is the telecommunication network with the GSM modems and we have be able to get the most benefit from telecommunication network and employ it in medical care industry, this employment of information technology in medical care provide accurate and timely service for the patient which lead to improve the quality of medical care.

We successes in simulating hl7 software by using GSM and telecommunication network to send and receive messages and reducing the high costly building of hl7 by GSM system. Figure4.1.1 show the GSM when connected to computer which is the first step of sending operation must be ensured after right connection to the USB port.

The next step is opening database to fill the data must be send figure4.1.2 illustrate the above mentioned step.

Figure 4.1.1 GSM when connected

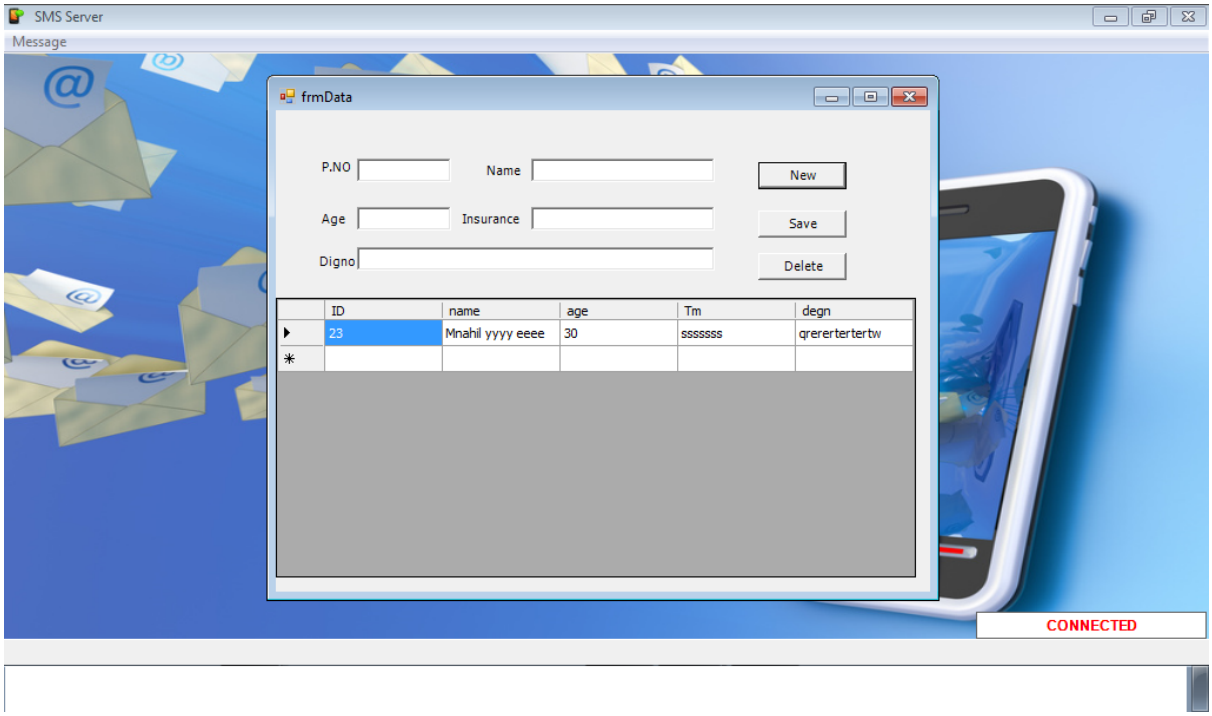

Figure 4.1.2 Opening database

The next step is sending operation which is done after saving the information in the above step figure 4.1.3 shown this step.

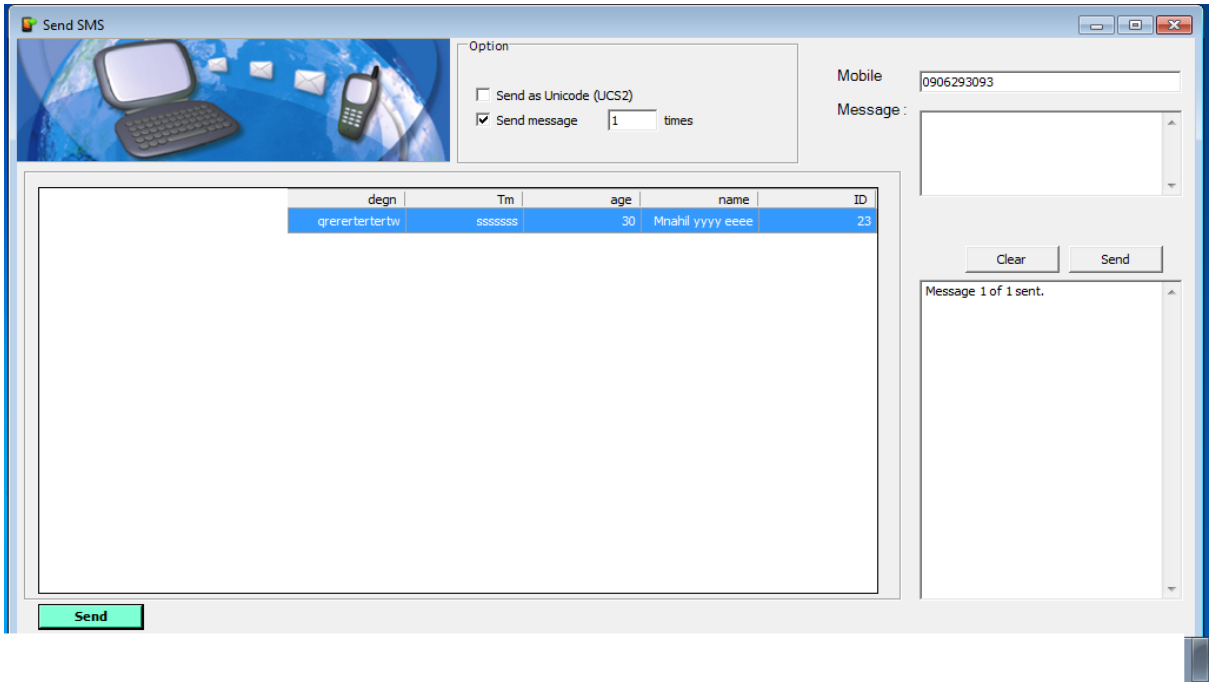

Figure 4.1.3 Sending operation

4.2 Discussion

Hl7 is large software as shown in it is construction and it has different version and segments which may make the user be confuse.

Different version of hl7 is one of it is problem for example if someone is sending you data in V2.5.1 that you cannot read in V2.3 also hl7 is coded value for example if one systems uses code A while another system uses code B then someone must translate the code .

Another problem with hl7 is mismatch segment usage which related to the difference in the format for example some systems ask for report information in the Note (NTE) segment while others want it in the Observation (OBX) segment. Someone has to change the message to meet these nuances in order for the receiver to view the report.

The HL7 specification may clearly note that a segment or field is optional, but some systems require the inclusion of the data in order to successfully exchange information. If you are unaware that a receiver's system requires an entry in a certain field, however, then message may not be received [20].

Chapter five

Conclusion and Recommendations

5.1 Conclusion

Financial issues always stopped the largest projects however this projects has a big value when done .our target is to reduce the cost of this projects specially the need of these project in developed countries like Sudan is considerable urgent.

The software program successes in achieve the linking between two or more points. This study illustrates how this message is transmitting and how patients will get the benefits from sending their information's before they arrive to the transmitted hospitals.

5.2 Recommendations

After the complete of sending and receiving operation we successes in transmitting messages among computer systems and we recommended by the following:

5.2.1 Send medical reports and medical images and check the network in term of speed.

5.2.2 Give the hospital servers some care and tension to distribute the data to all device need for exchangeable data.

5.2.3 Creating central points (servers) in the ministry of health to monitor patients transferring between hospitals to follow and ensure the timely and proper service.

5.2.4 Hospitals and health care units can send their requirements to the mentioned above server in the ministry of health to reduce the movements of their employers outside the hospitals on the other hand ministry of health can employ an employer to monitor the server and report the ministry of the receiving messages from hospitals.

5.2.5 Sending reminder messages (from hospitals servers) to patients on their mobiles to remind them the time of routine investigations they must do and to inform them when there is need to change time.

Chapter six

References and appendix

6.1 References

[1] Introduction to health level 7 w .E d Hammond Duke University medical center, ANSI accredits standards developer, member 2000.

[2] Health Insurance Reform: Announcement of Designated Standard Maintenance Organizations. Department of Health and Human Services, Health Care Financing Administration. August 17, 2000.

[3] Shaver, Dave. The HL7 Evolution Comparing HL7 Versions 2 and 3. Core point Health. Retrieved 16 February 2012.

[4] California Clinical Data Project: CALINX Standards". California HealthCare Foundation. March 2011.

[5] Health Insurance Reform: Announcement of Designated Standard Maintenance Organizations. Department of Health and Human Services, Health Care Financing Administration. August 17, 2000.

[6] HL7 2012 strategy Initiative . HL7. 2012.

[7]HLH Team Member Participation in HL7 plenary meeting for signing agreement of HL7 Pakistan with HL7 International at Cambridge, US". Pakistan: National University of Sciences & Technology.

[8] About Health Level Seven. Health Level Seven.

[9] http://www.interfaceware.com/data_exchange.html.

[10] http://www.interfaceware.com/understanding_hl7_messages.htm.

[11] Health Saskatchewan 2011-12 Annual Report (pdf). *health Saskatchewan*. Government of Saskatchewan. July 31, 2012. Retrieved 31 January 2014.

[12] Grant m wood, hl7 basic overview, Himss, lasvegas, 21.february 2012.

[13] Peter Schloeffel, Thomas Beale, George Hayworth, Sam Heard, Heather Leslie, the relationship between CEN 13606, HL7, and open EHR, Health Informatics Society of Australia Ltd (HISA), 2006.

[14] http://www.corepointhealth.com/whitepapers/why-do-i-need-hl7-interface-engine.

15 -Who created GSM. Stephen Temple. Retrieved 7 April 2013. "Before GSM, Europe had a disastrous mishmash of national analogue standards in phones and TV, designed to protect national industries but instead creating fragmented markets vulnerable to big guns from abroad.

[16] GSM d – Open moko. Wiki.openmoko.org. 8 February 2010. Retrieved 22 April 2010.

[17] Types of Smart Card. *Smart Card Basics*. Card Logix Corporation / evolis SCM Microsystems. Dynamic Application Card Operating System (section). Retrieved 15 July 2013.

[18] Felt, Dorr E. (1916). *Mechanical arithmetic, or the history of the counting machine*. Chicago: Washington Institute.

[19] Support Statement for Visual Basic 6.0 on Windows Vista, Windows Server 2008, Windows 7, and Windows 8". Msdn.microsoft.com. Retrieved 2013.

[20] http://www.netsoft-usa.com/2013/07/7-common-problems-with-hl7/ Harry Collins July 8, 2013.

[21]http://www.netsoft-usa.com/we-speak-your-language-hl7.

6.2 Appendix

Software

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.OleDb;
using GsmComm.PduConverter;
using GsmComm.GsmCommunication;
using SMS.Properties;
using System.Threading;


namespace SMS
{
	/// <summary>
	/// Summary description for Send.
	/// </summary>
	public class Send : System.Windows.Forms.Form
	{
		private System.Windows.Forms.PictureBox pictureBox1;
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.TextBox txt_destination_numbers;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.TextBox txt_text_remaining;
		private System.Windows.Forms.Button btnSendMessage;
		private System.Windows.Forms.Button BtnClear;
		private System.Windows.Forms.TextBox txt_message;
		private System.Windows.Forms.TextBox txtOutput;
		private System.Windows.Forms.CheckBox chkUnicode;
		private System.Windows.Forms.CheckBox chkAlert;
		private System.Windows.Forms.GroupBox groupBox1;
		private System.Windows.Forms.Label label19;
		private System.Windows.Forms.TextBox txtSendTimes;
		private System.Windows.Forms.CheckBox chkMultipleTimes;
	  private GroupBox groupBox2;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;

		private delegate void SetTextCallback(string text);

	  public string year, Result,ResultM;

	  string mesg;
	  DataSet ds = new DataSet();
	  public string ConString;
	  public string ServerName;
	  public string DBpassword;
	  public OleDbConnection con;
	  public bool Dcon;
```

```
        public Form CurrentForm;
        public string userName;
        public string password;
        private DataGridView GrdA;
        public string loginName;
        internal Button button1;
        private CheckBox chkMother;
        private ProgressBar prSend;
        Settings s = new Settings();

            public Send()
            {
                  //
                  // Required for Windows Form Designer support
                  //
                  InitializeComponent();

                  //
                  // TODO: Add any constructor code after
InitializeComponent call
                  //
            }

            /// <summary>
            /// Clean up any resources being used.
            /// </summary>
            protected override void Dispose( bool disposing )
            {
                  if( disposing )
                  {
                        if(components != null)
                        {
                              components.Dispose();
                        }
                  }
                  base.Dispose( disposing );
            }

            #region Windows Form Designer generated code
            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Send));
            System.Windows.Forms.DataGridViewCellStyle
dataGridViewCellStyle4 = new System.Windows.Forms.DataGridViewCellStyle();
            System.Windows.Forms.DataGridViewCellStyle
dataGridViewCellStyle5 = new System.Windows.Forms.DataGridViewCellStyle();
            System.Windows.Forms.DataGridViewCellStyle
dataGridViewCellStyle6 = new System.Windows.Forms.DataGridViewCellStyle();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.label1 = new System.Windows.Forms.Label();
            this.txt_destination_numbers = new
System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.txt_message = new System.Windows.Forms.TextBox();
            this.txt_text_remaining = new System.Windows.Forms.TextBox();
```

```
            this.btnSendMessage = new System.Windows.Forms.Button();
            this.BtnClear = new System.Windows.Forms.Button();
            this.txtOutput = new System.Windows.Forms.TextBox();
            this.chkUnicode = new System.Windows.Forms.CheckBox();
            this.chkAlert = new System.Windows.Forms.CheckBox();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.chkMother = new System.Windows.Forms.CheckBox();
            this.txtSendTimes = new System.Windows.Forms.TextBox();
            this.chkMultipleTimes = new System.Windows.Forms.CheckBox();
            this.label19 = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.GrdA = new System.Windows.Forms.DataGridView();
            this.button1 = new System.Windows.Forms.Button();
            this.prSend = new System.Windows.Forms.ProgressBar();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.groupBox1.SuspendLayout();
            this.groupBox2.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.GrdA)).BeginInit();
            this.SuspendLayout();
            //
            // pictureBox1
            //
            this.pictureBox1.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("pictureBox1.BackgroundImage"))
);
            this.pictureBox1.Location = new System.Drawing.Point(0, 0);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(368, 110);
            this.pictureBox1.TabIndex = 0;
            this.pictureBox1.TabStop = false;
            //
            // label1
            //
            this.label1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 10F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(695, 25);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(72, 24);
            this.label1.TabIndex = 1;
            this.label1.Text = "Mobile";
            //
            // txt_destination_numbers
            //
            this.txt_destination_numbers.Location = new
System.Drawing.Point(767, 29);
            this.txt_destination_numbers.Name = "txt_destination_numbers";
            this.txt_destination_numbers.Size = new
System.Drawing.Size(223, 20);
            this.txt_destination_numbers.TabIndex = 2;
            //
            // label2
            //
            this.label2.Font = new System.Drawing.Font("Microsoft Sans
Serif", 10F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label2.Location = new System.Drawing.Point(695, 55);
            this.label2.Name = "label2";
```

```
            this.label2.Size = new System.Drawing.Size(72, 28);
            this.label2.TabIndex = 3;
            this.label2.Text = "Message :";
            //
            // txt_message
            //
            this.txt_message.Location = new System.Drawing.Point(767, 64);
            this.txt_message.Multiline = true;
            this.txt_message.Name = "txt_message";
            this.txt_message.ScrollBars =
System.Windows.Forms.ScrollBars.Both;
            this.txt_message.Size = new System.Drawing.Size(224, 77);
            this.txt_message.TabIndex = 4;
            this.txt_message.TextChanged += new
System.EventHandler(this.textBox1_TextChanged);
            //
            // txt_text_remaining
            //
            this.txt_text_remaining.BackColor =
System.Drawing.SystemColors.Control;
            this.txt_text_remaining.Enabled = false;
            this.txt_text_remaining.Location = new
System.Drawing.Point(950, 148);
            this.txt_text_remaining.Name = "txt_text_remaining";
            this.txt_text_remaining.Size = new System.Drawing.Size(40, 20);
            this.txt_text_remaining.TabIndex = 5;
            this.txt_text_remaining.Text = "160";
            this.txt_text_remaining.Visible = false;
            //
            // btnSendMessage
            //
            this.btnSendMessage.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.btnSendMessage.Location = new System.Drawing.Point(894,
184);
            this.btnSendMessage.Name = "btnSendMessage";
            this.btnSendMessage.Size = new System.Drawing.Size(80, 24);
            this.btnSendMessage.TabIndex = 16;
            this.btnSendMessage.Text = "Send";
            this.btnSendMessage.Click += new
System.EventHandler(this.btnSendMessage_Click);
            //
            // BtnClear
            //
            this.BtnClear.BackColor = System.Drawing.Color.Cyan;
            this.BtnClear.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.BtnClear.Location = new System.Drawing.Point(806, 184);
            this.BtnClear.Name = "BtnClear";
            this.BtnClear.Size = new System.Drawing.Size(80, 24);
            this.BtnClear.TabIndex = 17;
            this.BtnClear.Text = "Clear";
            this.BtnClear.UseVisualStyleBackColor = false;
            this.BtnClear.Click += new
System.EventHandler(this.BtnClear_Click);
            //
            // txtOutput
            //
            this.txtOutput.Location = new System.Drawing.Point(767, 214);
            this.txtOutput.Multiline = true;
```

```
            this.txtOutput.Name = "txtOutput";
            this.txtOutput.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical;
            this.txtOutput.Size = new System.Drawing.Size(224, 284);
            this.txtOutput.TabIndex = 56;
            //
            // chkUnicode
            //
            this.chkUnicode.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.chkUnicode.Location = new System.Drawing.Point(16, 39);
            this.chkUnicode.Name = "chkUnicode";
            this.chkUnicode.Size = new System.Drawing.Size(208, 24);
            this.chkUnicode.TabIndex = 58;
            this.chkUnicode.Text = "Send as Unicode (UCS2)";
            //
            // chkAlert
            //
            this.chkAlert.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.chkAlert.Location = new System.Drawing.Point(16, 17);
            this.chkAlert.Name = "chkAlert";
            this.chkAlert.Size = new System.Drawing.Size(176, 24);
            this.chkAlert.TabIndex = 57;
            this.chkAlert.Text = "Request immediate display (alert)";
            this.chkAlert.Visible = false;
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.chkMother);
            this.groupBox1.Controls.Add(this.chkUnicode);
            this.groupBox1.Controls.Add(this.chkAlert);
            this.groupBox1.Controls.Add(this.txtSendTimes);
            this.groupBox1.Controls.Add(this.chkMultipleTimes);
            this.groupBox1.Controls.Add(this.label19);
            this.groupBox1.Location = new System.Drawing.Point(374, 0);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(291, 112);
            this.groupBox1.TabIndex = 59;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Option";
            //
            // chkMother
            //
            this.chkMother.ForeColor = System.Drawing.Color.Red;
            this.chkMother.Location = new System.Drawing.Point(17, 82);
            this.chkMother.Name = "chkMother";
            this.chkMother.Size = new System.Drawing.Size(114, 24);
            this.chkMother.TabIndex = 63;
            this.chkMother.Text = "للام رساله ارسال";
            this.chkMother.Visible = false;
            //
            // txtSendTimes
            //
            this.txtSendTimes.Location = new System.Drawing.Point(128, 63);
            this.txtSendTimes.Name = "txtSendTimes";
            this.txtSendTimes.Size = new System.Drawing.Size(43, 20);
            this.txtSendTimes.TabIndex = 61;
            this.txtSendTimes.Text = "1";
            //
```

```
            // chkMultipleTimes
            //
            this.chkMultipleTimes.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.chkMultipleTimes.Location = new System.Drawing.Point(16,
61);
            this.chkMultipleTimes.Name = "chkMultipleTimes";
            this.chkMultipleTimes.Size = new System.Drawing.Size(104, 24);
            this.chkMultipleTimes.TabIndex = 60;
            this.chkMultipleTimes.Text = "Send message";
            //
            // label19
            //
            this.label19.FlatStyle = System.Windows.Forms.FlatStyle.System;
            this.label19.ForeColor = System.Drawing.Color.Black;
            this.label19.Location = new System.Drawing.Point(176, 66);
            this.label19.Name = "label19";
            this.label19.Size = new System.Drawing.Size(42, 23);
            this.label19.TabIndex = 62;
            this.label19.Text = "times";
            this.label19.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.GrdA);
            this.groupBox2.Location = new System.Drawing.Point(6, 113);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(746, 385);
            this.groupBox2.TabIndex = 60;
            this.groupBox2.TabStop = false;
            //
            // GrdA
            //
            this.GrdA.AllowUserToAddRows = false;
            this.GrdA.BackgroundColor = System.Drawing.Color.White;
            dataGridViewCellStyle4.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft;
            dataGridViewCellStyle4.BackColor =
System.Drawing.SystemColors.Control;
            dataGridViewCellStyle4.Font = new System.Drawing.Font("Tahoma",
8F);
            dataGridViewCellStyle4.ForeColor =
System.Drawing.SystemColors.WindowText;
            dataGridViewCellStyle4.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
            dataGridViewCellStyle4.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;
            dataGridViewCellStyle4.WrapMode =
System.Windows.Forms.DataGridViewTriState.True;
            this.GrdA.ColumnHeadersDefaultCellStyle =
dataGridViewCellStyle4;
            this.GrdA.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            dataGridViewCellStyle5.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft;
            dataGridViewCellStyle5.BackColor =
System.Drawing.SystemColors.Window;
            dataGridViewCellStyle5.Font = new System.Drawing.Font("Tahoma",
8F);
```

```
            dataGridViewCellStyle5.ForeColor = System.Drawing.Color.Black;
            dataGridViewCellStyle5.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
            dataGridViewCellStyle5.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;
            dataGridViewCellStyle5.WrapMode =
System.Windows.Forms.DataGridViewTriState.False;
            this.GrdA.DefaultCellStyle = dataGridViewCellStyle5;
            this.GrdA.Location = new System.Drawing.Point(12, 19);
            this.GrdA.Name = "GrdA";
            this.GrdA.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
            dataGridViewCellStyle6.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft;
            dataGridViewCellStyle6.BackColor =
System.Drawing.SystemColors.Control;
            dataGridViewCellStyle6.Font = new System.Drawing.Font("Tahoma",
8F);
            dataGridViewCellStyle6.ForeColor =
System.Drawing.SystemColors.WindowText;
            dataGridViewCellStyle6.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
            dataGridViewCellStyle6.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;
            dataGridViewCellStyle6.WrapMode =
System.Windows.Forms.DataGridViewTriState.True;
            this.GrdA.RowHeadersDefaultCellStyle = dataGridViewCellStyle6;
            this.GrdA.RowHeadersVisible = false;
            this.GrdA.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
            this.GrdA.Size = new System.Drawing.Size(714, 360);
            this.GrdA.TabIndex = 36;
            this.GrdA.Visible = false;
            //
            // button1
            //
            this.button1.BackColor = System.Drawing.Color.Aquamarine;
            this.button1.Font = new System.Drawing.Font("Tahoma", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(0)));
            this.button1.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            this.button1.Location = new System.Drawing.Point(18, 501);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(90, 23);
            this.button1.TabIndex = 137;
            this.button1.Text = "Send";
            this.button1.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
            this.button1.UseVisualStyleBackColor = false;
            this.button1.Click += new
System.EventHandler(this.button1_Click);
            //
            // prSend
            //
            this.prSend.ForeColor =
System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(192)))),
((int)(((byte)(0)))));
            this.prSend.Location = new System.Drawing.Point(167, 504);
            this.prSend.Name = "prSend";
            this.prSend.Size = new System.Drawing.Size(506, 23);
```

```
            this.prSend.Style =
System.Windows.Forms.ProgressBarStyle.Marquee;
            this.prSend.TabIndex = 138;
            this.prSend.Visible = false;
            //
            // Send
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(1001, 532);
            this.Controls.Add(this.prSend);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.groupBox2);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.txtOutput);
            this.Controls.Add(this.BtnClear);
            this.Controls.Add(this.btnSendMessage);
            this.Controls.Add(this.txt_text_remaining);
            this.Controls.Add(this.txt_message);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.txt_destination_numbers);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.pictureBox1);
            this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.Name = "Send";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "Send SMS";
            this.Load += new System.EventHandler(this.Send_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.groupBox2.ResumeLayout(false);

((System.ComponentModel.ISupportInitialize)(this.GrdA)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

            }
            #endregion

            private void textBox1_TextChanged(object sender,
System.EventArgs e)
            {
                  int remaining=int.Parse(txt_text_remaining.Text.Trim());
                  remaining-=1;
                  txt_text_remaining.Text=remaining.ToString();
            }

            private void BtnClear_Click(object sender, System.EventArgs e)
            {
                  txt_message.Text="";
                  txt_message.Focus();
            }

            private void btnSendMessage_Click(object sender,
System.EventArgs e)
            {
            sendSMS(txt_destination_numbers.Text);
```

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

        private void sendSMS(string SMSM,string mobile)
        {
            Cursor.Current = Cursors.WaitCursor;

            try
            {
                // Send an SMS message
                SmsSubmitPdu pdu;
                bool alert = chkAlert.Checked;
                bool unicode = chkUnicode.Checked;

                if (!alert && !unicode)
                {
                    // The straightforward version
                    pdu = new SmsSubmitPdu(SMSM, mobile, "");  // ""
indicate SMSC No
                }
                else
                {
                    // The extended version with dcs
                    byte dcs;
                    if (!alert && unicode)
                        dcs = DataCodingScheme.NoClass_16Bit;
                    else if (alert && !unicode)
                        dcs = DataCodingScheme.Class0_7Bit;
                    else if (alert && unicode)
                        dcs = DataCodingScheme.Class0_16Bit;
                    else
                        dcs = DataCodingScheme.NoClass_7Bit; // should
never occur here

                    pdu = new SmsSubmitPdu(SMSM, mobile, "", dcs);
                }

                // Send the same message multiple times if this is set
                int times = chkMultipleTimes.Checked ?
int.Parse(txtSendTimes.Text) : 1;

                // Send the message the specified number of times
                for (int i = 0; i < times; i++)
                {
                    CommSetting.comm.SendMessage(pdu);
                    Output("Message {0} of {1} sent.", i + 1, times);
                    Output("");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
//-----------------------------------------------------------------
        private void sendSMS(string mobile)
        {
            Cursor.Current = Cursors.WaitCursor;
```

```
            try
            {
                // Send an SMS message
                SmsSubmitPdu pdu;
                bool alert = chkAlert.Checked;
                bool unicode = chkUnicode.Checked;

                if (!alert && !unicode)
                {
                    // The straightforward version
                    pdu = new SmsSubmitPdu(txt_message.Text, mobile, "");
// "" indicate SMSC No
                }
                else
                {
                    // The extended version with dcs
                    byte dcs;
                    if (!alert && unicode)
                        dcs = DataCodingScheme.NoClass_16Bit;
                    else if (alert && !unicode)
                        dcs = DataCodingScheme.Class0_7Bit;
                    else if (alert && unicode)
                        dcs = DataCodingScheme.Class0_16Bit;
                    else
                        dcs = DataCodingScheme.NoClass_7Bit; // should
never occur here

                    pdu = new SmsSubmitPdu(txt_message.Text, mobile, "",
dcs);
                }

                // Send the same message multiple times if this is set
                int times = chkMultipleTimes.Checked ?
int.Parse(txtSendTimes.Text) : 1;

                // Send the message the specified number of times
                for (int i = 0; i < times; i++)
                {
                    CommSetting.comm.SendMessage(pdu);
                    Output("Message {0} of {1} sent.", i + 1, times);
                    Output("");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

     //************----------------------------------------------------------
--------------------
        //public void Main()
        //{
         //Thread t = new Thread(sendReuslt);
         //t.Start();
        //}

         public void sendReuslt(string msg,string numder)
        {
         if(year ==null || Result==null || year =="" || Result=="")
```

```
          getParam();

              ResultM="";
        ds=ManipulatData("select ssname,Mark from vwResult where
ResultNo='"+Result+"' and Year='"+year+"' and StudNO='"+msg+"'");
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            ResultM = ResultM + ds.Tables[0].Rows[i]["ssname"].ToString() +
" " + ds.Tables[0].Rows[i]["Mark"].ToString()+",";

        }
     //-----------------send reuslt--------------

        try
        {
            // Send an SMS message
            SmsSubmitPdu pdu;
            bool alert = chkAlert.Checked;
            bool unicode = chkUnicode.Checked;

            if (!alert && !unicode)
            {
                // The straightforward version
                pdu = new SmsSubmitPdu(ResultM, numder, "");  // ""
indicate SMSC No
            }
            else
            {
                // The extended version with dcs
                byte dcs;
                if (!alert && unicode)
                    dcs = DataCodingScheme.NoClass_16Bit;
                else if (alert && !unicode)
                    dcs = DataCodingScheme.Class0_7Bit;
                else if (alert && unicode)
                    dcs = DataCodingScheme.Class0_16Bit;
                else
                    dcs = DataCodingScheme.NoClass_7Bit; // should never
occur here

                pdu = new SmsSubmitPdu(txt_message.Text, numder, "", dcs);
            }

            // Send the same message multiple times if this is set
            int times = chkMultipleTimes.Checked ?
int.Parse(txtSendTimes.Text) : 1;

            // Send the message the specified number of times
            for (int i = 0; i < times; i++)
            {
                CommSetting.comm.SendMessage(pdu);
                Output("Message {0} of {1} sent.", i + 1, times);
                Output("");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
     //------------------------------
```

```
        }

         public void getParam()
         {
             ds = ManipulatData("select YEAR,Result from SMSseeting");
             year = ds.Tables[0].Rows[0]["YEAR"].ToString();
             Result = ds.Tables[0].Rows[0]["Result"].ToString();
         }

     //-------------------------------------------------------------------
-------

        public void PrepaerConnection()
        {
            //ConString = "Integrated Security=False;Persist Security
Info=False;User ID=sa; Password=" + s.DBpassword  + "; Initial
Catalog=SchoolsDB;Data Source=" + s.serverName;
            ConString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
Application.StartupPath + "\\DB.mdb" + ";Persist Security Info=False";
            //sqlserver2008
            con = new OleDbConnection(ConString);

        }

        public  DataSet ManipulatData(string s)
        {
            PrepaerConnection();
            ds.Tables.Clear();
            OleDbCommand cmd = new OleDbCommand(s, con);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            da.Fill(ds);
            return ds;
        }

            private void Output(string text)
            {
                  if (this.txtOutput.InvokeRequired)
                  {
                        SetTextCallback stc = new SetTextCallback(Output);
                        this.Invoke(stc, new object[] { text });
                  }
                  else
                  {
                        txtOutput.AppendText(text);
                        txtOutput.AppendText("\r\n");
                  }
            }

            private void Send_Load(object sender, System.EventArgs e)
            {
                  chkMultipleTimes.Checked=true;
            ds = ManipulatData("select * from ptab");
            GrdA.DataSource = ds.Tables[0];
            GrdA.Visible = true;
            }

            private void Output(string text, params object[] args)
            {
                  string msg = string.Format(text, args);
                  Output(msg);
```

```
            }

        public static string SQLDate(DateTime myDate)
        {
            return " {d '" + String.Format(myDate.Year + "-" +
String.Format("{0:00}", myDate.Month) + "-" + "{0:00}", myDate.Day) + "'}";
        }
        public static string SQLDate(DateTime myDate, int i)
        {
            return String.Format(myDate.Year + "/" +
String.Format("{0:00}", myDate.Month) + "/" + "{0:00}", myDate.Day);
        }
        public static string SQLDate(DateTime myDate, string t)
        {
            return " {d '" + String.Format(myDate.Year + "-" +
String.Format("{0:00}", myDate.Month) + "-" + "{0:00}", myDate.Day) + "'}";
        }

        private void button2_Click(object sender, EventArgs e)
        {
            //try
            //{
            //   GrdAps.Visible = true;



            //   GrdAps.Rows.Clear();
            //   DataSet ds = ManipulatData("select
SID,Name,Fname,date,Class,SmsNo,Mtel,s='true' from vwAps where date=" +
SQLDate(dateTimePicker1.Value.Date));
            //     for(int i=0;i<ds.Tables[0].Rows.Count;i++)
            //
GrdAps.Rows.Add(ds.Tables[0].Rows[i]["SID"].ToString(),
ds.Tables[0].Rows[i]["Name"].ToString(),
ds.Tables[0].Rows[i]["Fname"].ToString(),
SQLDate(DateTime.Parse(ds.Tables[0].Rows[i]["date"].ToString()),0),
ds.Tables[0].Rows[i]["SmsNo"].ToString(),ds.Tables[0].Rows[i]["Mtel"].ToStr
ing(), true);

            //}
            //catch (Exception ex)
            //{
            //}
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                string M = "P.NO " +
GrdA.CurrentRow.Cells[0].Value.ToString() + " Name: " +
GrdA.CurrentRow.Cells[1].Value.ToString() + " Age: " +
GrdA.CurrentRow.Cells[2].Value.ToString() + " Insurance " +
GrdA.CurrentRow.Cells[3].Value.ToString() + " Digno " +
GrdA.CurrentRow.Cells[4].Value.ToString();
                sendSMS(M,txt_destination_numbers.Text );

                Cursor.Current = Cursors.Default;
            }
            catch (Exception ex)
```

```
            {
            }
        }


        private void button3_Click(object sender, EventArgs e)
        {
            //"SELECT FID,Name,GID,Gval,Gdate,SmsNO FROM vwGstRep where
Gdate>= " + SQLDate(dtpFrom.Value.Date) + " and Gdate<=" +
SQLDate(dtpTo.Value.Date) + " ");
                //try
            //{
            //   //grdgst.Visible = true;
            //   //grdF.Visible = false;
            //   //GrdAps.Visible = false;

            //   //grdgst.Rows.Clear();
            //   //DataSet ds = ManipulatData("SELECT
FID,Name,GID,Gval,Gdate,SmsNO,Mtel FROM vwGstRep where  Gdate>= " +
SQLDate(dtpFrom.Value.Date) + "  and Gdate<=" + SQLDate(dtpTo.Value.Date) +
" ");
            //   //  for(int i=0;i<ds.Tables[0].Rows.Count;i++)
            //   //
grdgst.Rows.Add(ds.Tables[0].Rows[i]["FID"].ToString(),
ds.Tables[0].Rows[i]["Name"].ToString(),
ds.Tables[0].Rows[i]["Gval"].ToString(),
SQLDate(DateTime.Parse(ds.Tables[0].Rows[i]["Gdate"].ToString()),0),
ds.Tables[0].Rows[i]["SmsNo"].ToString(),ds.Tables[0].Rows[i]["Mtel"].ToStr
ing(), true);

            //}
            //catch (Exception ex)
            //{
            //}
        }

        private void grdgst_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {
            //try
            //{
            //    grdF.Visible = true;
            //    grdgst.Visible = false;
            //    GrdAps.Visible = false;

            //    grdF.Rows.Clear();
            //    DataSet ds = ManipulatData("select FID,Name,SmsNO,Mtel
from Stdf where send='true'");
            //    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
            //        grdF.Rows.Add(ds.Tables[0].Rows[i]["FID"].ToString(),
ds.Tables[0].Rows[i]["Name"].ToString(),ds.Tables[0].Rows[i]["SmsNo"].ToStr
ing(),ds.Tables[0].Rows[i]["Mtel"].ToString(),true);

            //}
            //catch (Exception ex)
```

```
            //{
            //}
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            //if (GrdAps.Visible)
            //      {
            //        for (int i = 0; i < GrdAps.Rows.Count; i++)
            //           {
            //               GrdAps.Rows[i].Cells[6].Value=chk.Checked ;
            //            }
            //        }

            //if (grdF.Visible)
            //{
            //    for (int i = 0; i < grdF.Rows.Count; i++)
            //    {
            //        grdF.Rows[i].Cells[4].Value = chk.Checked;
            //    }
            //}


            //if (grdgst.Visible)
            //{
            //    for (int i = 0; i < grdgst.Rows.Count; i++)
            //    {
            //        grdgst.Rows[i].Cells[6].Value = chk.Checked;
            //    }
            //}
        }

        private void grdF_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void button5_Click(object sender, EventArgs e)
        {

        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SMS
{
    public partial class frmData : Form
    {
        DataSet ds = new DataSet();
        public string ConString;
        public string ServerName;
        public string DBpassword;
        public OleDbConnection con;

        public frmData()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }

        public void PrepaerConnection()
        {
            //ConString = "Integrated Security=False;Persist Security
Info=False;User ID=sa; Password=" + s.DBpassword + "; Initial
Catalog=SchoolsDB;Data Source=" + s.serverName;
            ConString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
Application.StartupPath + "\\DB.mdb" + ";Persist Security Info=False";
            //sqlserver2008
            con = new OleDbConnection(ConString);

        }

        public DataSet ManipulatData(string s)
        {
            PrepaerConnection();
            ds.Tables.Clear();
            OleDbCommand cmd = new OleDbCommand(s, con);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            da.Fill(ds);
            return ds;
        }

        private void frmData_Load(object sender, EventArgs e)
        {
            try
            {
                // TODO: This line of code loads data into the
'dBDataSet.Ptab' table. You can move, or remove it, as needed.
                ds = ManipulatData("select * from ptab");
                dataGridView1.DataSource = ds.Tables[0];
                //bindingNavigator1.
```

```
            }
            catch (Exception ex)
            {
            }

        }

        private void fillByToolStripButton_Click(object sender, EventArgs
e)
        {
            try
            {
                this.ptabTableAdapter.FillBy(this.dBDataSet.Ptab);
            }
            catch (System.Exception ex)
            {
                System.Windows.Forms.MessageBox.Show(ex.Message);
            }

        }

        private void ID_Click(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            txtID.Clear();
            txtName.Clear();
            txtTm.Clear();
            txtDeg.Clear();
            txtAge.Clear();

        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            try
            {
                ds = ManipulatData("insert into ptab values(" + txtID.Text
+ ",'" + txtName.Text + "','" + txtAge.Text + "','" + txtTm.Text + "','" +
txtDeg.Text + "')");
                ds = ManipulatData("select * from ptab");
                dataGridView1.DataSource = ds.Tables[0];
            }
            catch (Exception ex)
            {
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (txtID.Text == "")
                return;
            ds = ManipulatData("delete  from ptab where ID ="+txtID.Text);


        }
    }
```

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using GsmComm.GsmCommunication;
using SMS.Properties;
using System.Data;
using System.Data.SqlClient;

namespace SMS
{
	/// <summary>
	/// Summary description for frmConnection.
	/// </summary>
	public class frmConnection : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Label lblTimeout;
		private System.Windows.Forms.Label lblBaudRate;
		private System.Windows.Forms.ComboBox cboTimeout;
		private System.Windows.Forms.Label lblPort;
		private System.Windows.Forms.ComboBox cboPort;
		private System.Windows.Forms.ComboBox cboBaudRate;
		private System.Windows.Forms.Button btnOK;
		private System.Windows.Forms.Button btnCancel;
		private System.Windows.Forms.Button btnTest;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;

		private int port;
		private int baudRate;
	  private GroupBox groupBox1;
	  private Button button1;
	  private TextBox txtServer;
	  private TextBox txtPass;
	  private Label label1;
	  private Label label2;
		private int timeout;

	  public SqlConnection con;
	  private Button button2;
	  Settings S = new Settings();

		public frmConnection()
		{
			//
			// Required for Windows Form Designer support
			//
			InitializeComponent();

			//
			// TODO: Add any constructor code after
InitializeComponent call
			//
		}

		/// <summary>
		/// Clean up any resources being used.
```

```
		/// </summary>
		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if(components != null)
				{
					components.Dispose();
				}
			}
			base.Dispose( disposing );
		}

		#region Windows Form Designer generated code
		/// <summary>
		/// Required method for Designer support - do not modify
		/// the contents of this method with the code editor.
		/// </summary>
		private void InitializeComponent()
		{
		this.lblTimeout = new System.Windows.Forms.Label();
		this.lblBaudRate = new System.Windows.Forms.Label();
		this.cboTimeout = new System.Windows.Forms.ComboBox();
		this.lblPort = new System.Windows.Forms.Label();
		this.cboPort = new System.Windows.Forms.ComboBox();
		this.cboBaudRate = new System.Windows.Forms.ComboBox();
		this.btnOK = new System.Windows.Forms.Button();
		this.btnCancel = new System.Windows.Forms.Button();
		this.btnTest = new System.Windows.Forms.Button();
		this.groupBox1 = new System.Windows.Forms.GroupBox();
		this.button1 = new System.Windows.Forms.Button();
		this.txtServer = new System.Windows.Forms.TextBox();
		this.txtPass = new System.Windows.Forms.TextBox();
		this.label1 = new System.Windows.Forms.Label();
		this.label2 = new System.Windows.Forms.Label();
		this.button2 = new System.Windows.Forms.Button();
		this.SuspendLayout();
		//
		// lblTimeout
		//
		this.lblTimeout.Location = new System.Drawing.Point(33, 72);
		this.lblTimeout.Name = "lblTimeout";
		this.lblTimeout.Size = new System.Drawing.Size(100, 23);
		this.lblTimeout.TabIndex = 4;
		this.lblTimeout.Text = "Ti&meout (ms):";
		this.lblTimeout.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
		//
		// lblBaudRate
		//
		this.lblBaudRate.Location = new System.Drawing.Point(33, 48);
		this.lblBaudRate.Name = "lblBaudRate";
		this.lblBaudRate.Size = new System.Drawing.Size(100, 23);
		this.lblBaudRate.TabIndex = 2;
		this.lblBaudRate.Text = "&Baud rate:";
		this.lblBaudRate.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
		//
		// cboTimeout
		//
```

```
            this.cboTimeout.Location = new System.Drawing.Point(145, 72);
            this.cboTimeout.Name = "cboTimeout";
            this.cboTimeout.Size = new System.Drawing.Size(104, 21);
            this.cboTimeout.TabIndex = 5;
            //
            // lblPort
            //
            this.lblPort.Location = new System.Drawing.Point(33, 24);
            this.lblPort.Name = "lblPort";
            this.lblPort.Size = new System.Drawing.Size(88, 23);
            this.lblPort.TabIndex = 0;
            this.lblPort.Text = "&COM-Port:";
            this.lblPort.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            //
            // cboPort
            //
            this.cboPort.Location = new System.Drawing.Point(145, 24);
            this.cboPort.Name = "cboPort";
            this.cboPort.Size = new System.Drawing.Size(104, 21);
            this.cboPort.TabIndex = 1;
            //
            // cboBaudRate
            //
            this.cboBaudRate.Location = new System.Drawing.Point(145, 48);
            this.cboBaudRate.Name = "cboBaudRate";
            this.cboBaudRate.Size = new System.Drawing.Size(104, 21);
            this.cboBaudRate.TabIndex = 3;
            //
            // btnOK
            //
            this.btnOK.DialogResult = System.Windows.Forms.DialogResult.OK;
            this.btnOK.FlatStyle = System.Windows.Forms.FlatStyle.System;
            this.btnOK.Location = new System.Drawing.Point(46, 251);
            this.btnOK.Name = "btnOK";
            this.btnOK.Size = new System.Drawing.Size(75, 23);
            this.btnOK.TabIndex = 7;
            this.btnOK.Text = "OK";
            this.btnOK.Click += new System.EventHandler(this.btnOK_Click);
            //
            // btnCancel
            //
            this.btnCancel.DialogResult =
System.Windows.Forms.DialogResult.Cancel;
            this.btnCancel.FlatStyle =
System.Windows.Forms.FlatStyle.System;
            this.btnCancel.Location = new System.Drawing.Point(145, 251);
            this.btnCancel.Name = "btnCancel";
            this.btnCancel.Size = new System.Drawing.Size(75, 23);
            this.btnCancel.TabIndex = 8;
            this.btnCancel.Text = "Cancel";
            //
            // btnTest
            //
            this.btnTest.FlatStyle = System.Windows.Forms.FlatStyle.System;
            this.btnTest.Location = new System.Drawing.Point(16, 112);
            this.btnTest.Name = "btnTest";
            this.btnTest.Size = new System.Drawing.Size(75, 23);
            this.btnTest.TabIndex = 6;
            this.btnTest.Text = "&Test";
```

```
            this.btnTest.Click += new
System.EventHandler(this.btnTest_Click);
            //
            // groupBox1
            //
            this.groupBox1.Location = new System.Drawing.Point(-8, 141);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(296, 10);
            this.groupBox1.TabIndex = 9;
            this.groupBox1.TabStop = false;
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(4, 214);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(103, 23);
            this.button1.TabIndex = 10;
            this.button1.Text = "Test Connection";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Visible = false;
            this.button1.Click += new
System.EventHandler(this.button1_Click);
            //
            // txtServer
            //
            this.txtServer.Location = new System.Drawing.Point(113, 160);
            this.txtServer.Name = "txtServer";
            this.txtServer.Size = new System.Drawing.Size(136, 20);
            this.txtServer.TabIndex = 11;
            this.txtServer.Visible = false;
            //
            // txtPass
            //
            this.txtPass.Location = new System.Drawing.Point(113, 186);
            this.txtPass.Name = "txtPass";
            this.txtPass.PasswordChar = '*';
            this.txtPass.Size = new System.Drawing.Size(136, 20);
            this.txtPass.TabIndex = 12;
            this.txtPass.Visible = false;
            this.txtPass.TextChanged += new
System.EventHandler(this.textBox2_TextChanged);
            //
            // label1
            //
            this.label1.Location = new System.Drawing.Point(19, 157);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(88, 23);
            this.label1.TabIndex = 13;
            this.label1.Text = "Serve rName:";
            this.label1.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            this.label1.Visible = false;
            //
            // label2
            //
            this.label2.Location = new System.Drawing.Point(19, 184);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(88, 23);
            this.label2.TabIndex = 14;
            this.label2.Text = "DB Password";
```

```
            this.label2.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            this.label2.Visible = false;
            //
            // button2
            //
            this.button2.Location = new System.Drawing.Point(4, 0);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(34, 23);
            this.button2.TabIndex = 15;
            this.button2.Text = "...";
            this.button2.UseVisualStyleBackColor = true;
            this.button2.Click += new
System.EventHandler(this.button2_Click);
            //
            // frmConnection
            //
            this.AcceptButton = this.btnOK;
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.CancelButton = this.btnCancel;
            this.ClientSize = new System.Drawing.Size(282, 287);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.txtPass);
            this.Controls.Add(this.txtServer);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.btnTest);
            this.Controls.Add(this.btnCancel);
            this.Controls.Add(this.btnOK);
            this.Controls.Add(this.lblTimeout);
            this.Controls.Add(this.lblBaudRate);
            this.Controls.Add(this.cboTimeout);
            this.Controls.Add(this.lblPort);
            this.Controls.Add(this.cboPort);
            this.Controls.Add(this.cboBaudRate);
            this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
            this.MaximizeBox = false;
            this.MinimizeBox = false;
            this.Name = "frmConnection";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "Connection settings";
            this.Load += new System.EventHandler(this.frmConnection_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

            }
            #endregion

            public void SetData(int port, int baudRate, int timeout)
            {
                  this.port = port;
                  this.baudRate = baudRate;
                  this.timeout = timeout;
            }
```

```
            public void GetData(out int port, out int baudRate, out int
timeout)
            {
                  port = this.port;
                  baudRate = this.baudRate;
                  timeout = this.timeout;
            }

            private bool EnterNewSettings()
            {
                  int newPort;
                  int newBaudRate;
                  int newTimeout;

                  try
                  {
                        newPort = int.Parse(cboPort.Text);
                  }
                  catch(Exception)
                  {
                        MessageBox.Show(this, "Invalid port number.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        cboPort.Focus();
                        return false;
                  }

                  try
                  {
                        newBaudRate = int.Parse(cboBaudRate.Text);
                  }
                  catch(Exception)
                  {
                        MessageBox.Show(this, "Invalid baud rate.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        cboBaudRate.Focus();
                        return false;
                  }

                  try
                  {
                        newTimeout = int.Parse(cboTimeout.Text);
                  }
                  catch(Exception)
                  {
                        MessageBox.Show(this, "Invalid timeout value.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        cboTimeout.Focus();
                        return false;
                  }

                  SetData(newPort,newBaudRate,newTimeout);

                  return true;
            }

            private void btnOK_Click(object sender, System.EventArgs e)
            {
            SaveSetting();
            if (!EnterNewSettings())
                DialogResult = DialogResult.None;
```

```
        }

        private void frmConnection_Load(object sender, System.EventArgs
e)
        {
              cboPort.Items.Add("1");
              cboPort.Items.Add("2");
              cboPort.Items.Add("3");
              cboPort.Items.Add("4");
              //cboPort.Text = port.ToString();

              cboBaudRate.Items.Add("9600");
              cboBaudRate.Items.Add("19200");
              cboBaudRate.Items.Add("38400");
              cboBaudRate.Items.Add("57600");
              cboBaudRate.Items.Add("115200");
              //cboBaudRate.Text = baudRate.ToString();

              cboTimeout.Items.Add("150");
              cboTimeout.Items.Add("300");
              cboTimeout.Items.Add("600");
              cboTimeout.Items.Add("900");
              cboTimeout.Items.Add("1200");
              cboTimeout.Items.Add("1500");
              cboTimeout.Items.Add("1800");
              cboTimeout.Items.Add("2000");
              //cboTimeout.Text = timeout.ToString();
        lodSetting();
        }

        private void btnTest_Click(object sender, System.EventArgs e)
        {
              if (!EnterNewSettings())
                    return;

              Cursor.Current = Cursors.WaitCursor;
              GsmCommMain comm = new GsmCommMain(port, baudRate,
timeout);
              try
              {
                    comm.Open();
                    while (!comm.IsConnected())
                    {
                          Cursor.Current = Cursors.Default;
                          if (MessageBox.Show(this, "No phone
connected.", "Connection setup",
                                MessageBoxButtons.RetryCancel,
MessageBoxIcon.Exclamation) == DialogResult.Cancel)
                          {
                                comm.Close();
                                return;
                          }
                          Cursor.Current = Cursors.WaitCursor;
                    }

                    comm.Close();
              }
              catch(Exception ex)
              {
```

```
                        MessageBox.Show(this, "Connection error: " +
ex.Message, "Connection setup", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                        return;
                    }
                    MessageBox.Show(this, "Successfully connected to the
phone.", "Connection setup", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            SaveSetting();

            }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        public void TestConnection()
        {

            string ConString = "Integrated Security=False;Persist Security
Info=False;User ID=sa; Password=" + txtPass.Text +"; Initial
Catalog=SchoolsDB;Data Source=" + txtServer.Text;
            //ConString = "Integrated Security=False;Persist Security
Info=False;User ID=sa; Password=qwerty123*; Initial Catalog=SchoolsDB;Data
Source=MUNTASER-4F4F31";
            //sqlserver2008
            con = new SqlConnection(ConString);
            try
            {
                con.Open();
                MessageBox.Show(this, "Successfully connected to the
DataBase", "Connection setup", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            catch (Exception ex)
            {
                MessageBox.Show(this, "Connection error: " + ex.Message,
"Connection setup", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }


        }


        private void button1_Click(object sender, EventArgs e)
        {
            TestConnection();
            SaveSetting();

        }

        private void SaveSetting()
        {
            S.serverName = txtServer.Text;
            S.DBpassword = txtPass.Text;

            S.ComPort = cboPort.Text;
            S.Reate = cboBaudRate.Text;
            S.RT = cboTimeout.Text;
```

```
            S.Save();
        }


        private void lodSetting()
        {
               txtServer.Text=S.serverName;
              txtPass.Text=S.DBpassword;
             cboPort.Text=S.ComPort;
            cboBaudRate.Text=S.Reate;
           cboTimeout.Text= S.RT ;

        }

        private void button2_Click(object sender, EventArgs e)
        {
            lodSetting();
        }
    }
}
```

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

using GsmComm.GsmCommunication;
using GsmComm.PduConverter;


using System.Threading;

namespace SMS
{
	/// <summary>
	/// Summary description for Form1.
	/// </summary>
	public class Form1 : System.Windows.Forms.Form
	{

		private System.Windows.Forms.MainMenu mainMenu1;
		private System.Windows.Forms.MenuItem menuItem1;
		private System.Windows.Forms.Label lbl_phone_status;
		private System.Windows.Forms.TextBox txtOutput;
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.MenuItem mnu_send;
		private System.Windows.Forms.MenuItem mnu_read;

		private delegate void SetTextCallback(string text);
		private CommSetting comm_settings=new CommSetting();
	  private System.Windows.Forms.MenuItem mnudelete;
	  private StatusStrip statusStrip1;
	  public TextBox txtMessage;
	  private MenuItem menuItem2;
	  private IContainer components;

		public Form1()
		{
			//
			// Required for Windows Form Designer support
			//
			InitializeComponent();

			//
			// TODO: Add any constructor code after
InitializeComponent call
			//
		}

		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if (components != null)
				{
					components.Dispose();
```

```
                    }
                }
                base.Dispose( disposing );
            }

            #region Windows Form Designer generated code
            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.mainMenu1 = new
System.Windows.Forms.MainMenu(this.components);
            this.menuItem1 = new System.Windows.Forms.MenuItem();
            this.mnu_send = new System.Windows.Forms.MenuItem();
            this.mnu_read = new System.Windows.Forms.MenuItem();
            this.mnudelete = new System.Windows.Forms.MenuItem();
            this.menuItem2 = new System.Windows.Forms.MenuItem();
            this.lbl_phone_status = new System.Windows.Forms.Label();
            this.txtOutput = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.statusStrip1 = new System.Windows.Forms.StatusStrip();
            this.txtMessage = new System.Windows.Forms.TextBox();
            this.SuspendLayout();
            //
            // mainMenu1
            //
            this.mainMenu1.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
            this.menuItem1});
            //
            // menuItem1
            //
            this.menuItem1.Index = 0;
            this.menuItem1.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
            this.mnu_send,
            this.mnu_read,
            this.mnudelete,
            this.menuItem2});
            this.menuItem1.Text = "Message";
            //
            // mnu_send
            //
            this.mnu_send.Index = 0;
            this.mnu_send.Text = "Send";
            this.mnu_send.Click += new
System.EventHandler(this.mnu_send_click);
            //
            // mnu_read
            //
            this.mnu_read.Index = 1;
            this.mnu_read.Text = "Read";
            this.mnu_read.Click += new
System.EventHandler(this.mnu_read_click);
            //
```

```
            // mnudelete
            //
            this.mnudelete.Index = 2;
            this.mnudelete.Text = "Delete";
            this.mnudelete.Click += new
System.EventHandler(this.mnudelete_Click);
            //
            // menuItem2
            //
            this.menuItem2.Index = 3;
            this.menuItem2.Text = "Data";
            this.menuItem2.Click += new
System.EventHandler(this.menuItem2_Click);
            //
            // lbl_phone_status
            //
            this.lbl_phone_status.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bot
tom | System.Windows.Forms.AnchorStyles.Right)));
            this.lbl_phone_status.BackColor = System.Drawing.Color.White;
            this.lbl_phone_status.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
            this.lbl_phone_status.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.lbl_phone_status.ForeColor = System.Drawing.Color.Red;
            this.lbl_phone_status.Location = new System.Drawing.Point(701,
372);
            this.lbl_phone_status.Name = "lbl_phone_status";
            this.lbl_phone_status.Size = new System.Drawing.Size(196, 23);
            this.lbl_phone_status.TabIndex = 54;
            this.lbl_phone_status.Text = "NO PHONE CONNECTED";
            this.lbl_phone_status.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            this.lbl_phone_status.UseMnemonic = false;
            //
            // txtOutput
            //
            this.txtOutput.Enabled = false;
            this.txtOutput.Location = new System.Drawing.Point(12, 147);
            this.txtOutput.Multiline = true;
            this.txtOutput.Name = "txtOutput";
            this.txtOutput.ScrollBars =
System.Windows.Forms.ScrollBars.Vertical;
            this.txtOutput.Size = new System.Drawing.Size(357, 191);
            this.txtOutput.TabIndex = 55;
            this.txtOutput.Visible = false;
            //
            // label1
            //
            this.label1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 10F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(17, 123);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(104, 24);
            this.label1.TabIndex = 56;
            this.label1.Text = "Received SMS :";
            this.label1.Visible = false;
            //
```

```
            // statusStrip1
            //
            this.statusStrip1.Location = new System.Drawing.Point(0, 395);
            this.statusStrip1.Name = "statusStrip1";
            this.statusStrip1.Size = new System.Drawing.Size(899, 22);
            this.statusStrip1.TabIndex = 58;
            this.statusStrip1.Text = "statusStrip1";
            //
            // txtMessage
            //
            this.txtMessage.Location = new System.Drawing.Point(0, 372);
            this.txtMessage.Name = "txtMessage";
            this.txtMessage.Size = new System.Drawing.Size(200, 20);
            this.txtMessage.TabIndex = 59;
            this.txtMessage.Visible = false;
            //
            // Form1
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
            this.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
            this.ClientSize = new System.Drawing.Size(899, 417);
            this.Controls.Add(this.txtMessage);
            this.Controls.Add(this.statusStrip1);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.txtOutput);
            this.Controls.Add(this.lbl_phone_status);
            this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.Menu = this.mainMenu1;
            this.Name = "Form1";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "SMS Server";
            this.Load += new System.EventHandler(this.Form1_Load);
            this.Closing += new
System.ComponentModel.CancelEventHandler(this.Form1_Closing);
            this.ResumeLayout(false);
            this.PerformLayout();

            }
            #endregion

            /// <summary>
            /// The main entry point for the application.
            /// </summary>
            [STAThread]
            static void Main()
            {
                  Application.Run(new Form1());
            }


            private void Form1_Load(object sender, System.EventArgs e)
            {
                  // Prompt user for connection settings
                  int port = GsmCommMain.DefaultPortNumber;
```

```
                int baudRate = 9600; // We Set 9600 as our Default Baud
Rate
                int timeout = GsmCommMain.DefaultTimeout;

                frmConnection dlg = new frmConnection();
                dlg.StartPosition = FormStartPosition.CenterScreen;
                dlg.SetData(port, baudRate, timeout);

                if (dlg.ShowDialog(this) == DialogResult.OK)
                {
                      dlg.GetData(out port, out baudRate, out timeout);
                      CommSetting.Comm_Port=port;
                      CommSetting.Comm_BaudRate=baudRate;
                      CommSetting.Comm_TimeOut=timeout;
                }
                else
                {
                      Close();
                      return;
                }

                Cursor.Current = Cursors.WaitCursor;
                CommSetting.comm = new GsmCommMain(port, baudRate,
timeout);
                Cursor.Current = Cursors.Default;
                CommSetting.comm.PhoneConnected += new
EventHandler(comm_PhoneConnected);
                CommSetting.comm.MessageReceived+=new
MessageReceivedEventHandler(comm_MessageReceived);

                bool retry;
                do
                {
                      retry = false;
                      try
                      {
                            Cursor.Current = Cursors.WaitCursor;
                            CommSetting.comm.Open();
                            Cursor.Current = Cursors.Default;
                      }
                      catch(Exception)
                      {
                            Cursor.Current = Cursors.Default;
                            if (MessageBox.Show(this, "Unable to open the
port.", "Error",
                                  MessageBoxButtons.RetryCancel,
MessageBoxIcon.Warning) == DialogResult.Retry)
                                  retry = true;
                            else
                            {
                                  Close();
                                  return;
                            }
                      }
                }
                while(retry);

          }
```

```
            private delegate void ConnectedHandler(bool connected);

            private void OnPhoneConnectionChange(bool connected)
            {
                  lbl_phone_status.Text="CONNECTED";
            }


            private void comm_MessageReceived(object sender,
GsmComm.GsmCommunication.MessageReceivedEventArgs e)
            {
                  MessageReceived();

            }

            private void comm_PhoneConnected(object sender, EventArgs e)
            {
                  this.Invoke(new
ConnectedHandler(OnPhoneConnectionChange), new object[] { true });
            }


            private string GetMessageStorage()
            {
                  string storage = string.Empty;
                  storage = PhoneStorageType.Sim;

                  if (storage.Length == 0)
                        throw new ApplicationException("Unknown message
storage.");
                  else
                        return storage;
            }



//---------------------------------------------------------------------


//---------------------------------------------------------------------

            private void MessageReceived()
            {
                  Cursor.Current = Cursors.WaitCursor;
                  string storage = GetMessageStorage();

                  DecodedShortMessage[] messages =
CommSetting.comm.ReadMessages(PhoneMessageStatus.ReceivedUnread, storage);
                  foreach(DecodedShortMessage message in messages)
                  {
                Output(string.Format("Message status = {0}, Location =
{1}/{2}",
                  StatusToString(message.Status),     message.Storage,
message.Index));
                //ShowMessage(message.Data);
                message.Data.UserDataText.ToString();

                MsgOutput(message.Data.UserDataText);

//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

```
                //Send frm=new Send();
                //frm.sendReuslt(message.Data.UserDataText, "0902656159");


//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        Output("");
                    }

                    Output(string.Format("{0,9} messages read.",
messages.Length.ToString()));
                    Output("");
            }

        private void setmessage(string msg)
        {
            txtMessage.Text = msg;
        }

            private string StatusToString(PhoneMessageStatus status)
            {
                  // Map a message status to a string
                  string ret;
                  switch(status)
                  {
                        case PhoneMessageStatus.All:
                              ret = "All";
                              break;
                        case PhoneMessageStatus.ReceivedRead:
                              ret = "Read";
                              break;
                        case PhoneMessageStatus.ReceivedUnread:
                              ret = "Unread";
                              break;
                        case PhoneMessageStatus.StoredSent:
                              ret = "Sent";
                              break;
                        case PhoneMessageStatus.StoredUnsent:
                              ret = "Unsent";
                              break;
                        default:
                              ret = "Unknown (" + status.ToString() + ")";
                              break;
                  }
                  return ret;
            }

        private void MsgOutput(string text)
        {
            if (this.txtMessage.InvokeRequired)
            {
                SetTextCallback stc = new SetTextCallback(Output);
                this.Invoke(stc, new object[] { text });
            }
            else
            {
                txtMessage.AppendText(text);
                txtMessage.AppendText("\r\n");
            }
        }
```

```
        private void Output(string text)
            {
                  if (this.txtOutput.InvokeRequired)
                  {
                        SetTextCallback stc = new SetTextCallback(Output);
                        this.Invoke(stc, new object[] { text });
                  }
                  else
                  {
                        txtOutput.AppendText(text);
                        //txtOutput.AppendText("\r\n");
                  }
            }


            private void ShowMessage(SmsPdu pdu)
            {
                  if (pdu is SmsSubmitPdu)
                  {
                        // Stored (sent/unsent) message
                        SmsSubmitPdu data = (SmsSubmitPdu)pdu;
                        //Output("SENT/UNSENT MESSAGE");
                        //Output("Recipient: " + data.DestinationAddress);
                        //Output("Message text: " + data.UserDataText);
                    Output(data.UserDataText);
                        //Output("----------------------------------------
--------------------------");
                        return;
                  }
                  if (pdu is SmsDeliverPdu)
                  {
                        // Received message
                        SmsDeliverPdu data = (SmsDeliverPdu)pdu;
                        //Output("RECEIVED MESSAGE");
                        //Output("Sender: " + data.OriginatingAddress);
                        //Output("Sent: " + data.SCTimestamp.ToString());
                        Output(data.UserDataText);

                        //Output("----------------------------------------
--------------------------");
                        return;
                  }
                  if (pdu is SmsStatusReportPdu)
                  {
                        // Status report
                        SmsStatusReportPdu data = (SmsStatusReportPdu)pdu;
                        Output("STATUS REPORT");
                        Output("Recipient: " + data.RecipientAddress);
                        Output("Status: " + data.Status.ToString());
                        Output("Timestamp: " +
data.DischargeTime.ToString());
                        Output("Message ref: " +
data.MessageReference.ToString());
                        Output("----------------------------------------
-----------------------");
                        return;
                  }
```

```
                  //Output("Unknown message type: " +
pdu.GetType().ToString());
            }

            private void mnu_send_click(object sender, System.EventArgs e)
            {
                  Send send_sms=new Send();
                  send_sms.Show();
            }


            private void mnu_read_click(object sender, System.EventArgs e)
            {
                  Receive receice_sms=new Receive();
                  receice_sms.Show();
            }


            private void Form1_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
            {
                  // Clean up comm object
                  if (CommSetting.comm != null)
                  {
                        // Unregister events
                        CommSetting.comm.PhoneConnected -= new
EventHandler(comm_PhoneConnected);
                        CommSetting.comm.MessageReceived -= new
MessageReceivedEventHandler(comm_MessageReceived);

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

                        CommSetting.comm = null;
                  }
            }

            private void mnudelete_Click(object sender, System.EventArgs e)
            {
                  Delete delete=new Delete();
                  delete.Show();
            }

        private void menuItem2_Click(object sender, EventArgs e)
        {
            frmData frm = new frmData();
           // frm.MdiParent = this;
            frm.Show();
        }

      }

}
```