Innovative Systems Design and Engineering
ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol.4, No.13, 2013

System Design for Mobile Phone Data Backup

Kehinde Aregbesola[1*], Oluwaseyitan Osunade[2]

1. College of Information and Communication Technology, Salem University, Lokoja, Kogi State, Nigeria
2. Department of Computer Science, University of Ibadan, Ibadan, Oyo State, Nigeria

* E-mail of the corresponding author: kennie747@yahoo.com, o.osunade@mail.ui.edu.ng

Abstract

This study briefly looked into how mobile phones work. It then considered the design of the system (graphical user interface, the application and the database) used for mobile phone data backup. This system backs up mobile phone and Subscribers Identification Module (SIM) data (contacts and SMS) on the computer storage. It also allows for easy retrieval and restoration of the data to phone memory when the need arises. The objectives considered in the design include: user-friendly graphical user interface (GUI) for user interaction; modular subroutines, methods and functions that retrieves data from the phone and stores it on the computer memory; modular subroutines, methods and functions that retrieves data from the SIM and stores it on the computer memory; modular subroutines and functions that send data from the computer back to the phone; portable database for storing and retrieving the data. The design employs several intuitive system design tools (mainly of graphical nature) to make the concept being discussed as lucid as possible. Some of the tools used include context diagram, data flow diagram, state diagram, use case diagram, deployment diagram, entity relation model etc.

Keywords: Mobile Phone, Data Backup, System Design, GSM, Phone Data.

1. Introduction

Mobile phones come in different shapes and sizes. Owners of GSM phones are sometimes classed by the type of phone(s) they own and the functionalities the phones can perform. The demand for functionalities that come with GSM phones has hence increased significantly. The major striking features that most users clamor for apart from the phone having digital camera/recording facilities is the ability of their phones to store more data – or at least have some sort of extended memory - , display a high level of security and provide for backup of phone data (in case of any mishap to the phone). As more and more functionalities are added to these phones, the more expensive they become. Hence these days, we find phones costing hundreds of thousands of naira. Not all can afford phones this expensive. Everyone wants to maximize and improve on what he already has. The average man is in search of a means by which he can extend the functionalities that port naturally with his phone.

There is the risk of losing all of one's contacts and valuable text messages in situations of Phone/SIM loss, damage, or erasure by mistake due to careless use. It is hence very necessary to back up the data in the phonebook of a mobile phone/SIM.

One can manually write out all these data from his phone. This would however be very time consuming and tedious. Besides, it would be extremely difficult to keep this manual record up to date since the user will need to browse through the entire contents of the phone each time he plans to make an update of his record.

It is for this reason that this data backup tool was designed and developed to automate the backup of mobile phone data. The details of the system design is discussed in subsequent chapters of this work.

2. Literature Review

In spite of its current popularity, mobile phones have existed long before GSM was conceived. In telecommunication, a **wireless phone, cellular mobile**, **cell phone** or **mobile phone**, is a mobile communications system that uses a combination of radio wave transmission and conventional telephone switching to permit telephone communication to and from mobile users within a specified area.

The term does not comprise the so-called *portable phone* or *cordless phone*, which is associated with a fixed telephone landline and can only be operated close to (less than 100 meters of) its base station, such as in and around the house. The term *cell phone* applies specifically to mobile phones that use a cellular network. **Satellite phones** are also mobile phones, but not cellular (David, 1991; Feldmann and Rissen, 1993b).

In cellular mobile systems, large geographical areas are segmented into many smaller areas, *i.e.*, cells, each of which has its own radio transmitters and receivers and a single controller interconnected with the public switched telephone network. Each cell site has a range of 3-5 miles and overlaps other cell sites. All of the cell sites are connected to one or more cellular switching exchanges which can detect the strength of the signal received from the telephone. As the telephone user moves or roams from one cell area to another, the exchange automatically switches the call to the cell site with the strongest signal (David, 1991; Feldmann and Rissen, 1993b).
All cell phones have special codes associated with them. These codes are used to identify the phone, the phone's owner and the service provider (Maduka, 2005). They are:

- Electronic Serial Number (ESN)
- Mobile Identification Number (MIN)
- System Identification Code (SID)

While the ESN is considered a permanent part of the phone, both the MIN and SID codes are programmed into the phone when you purchase a service plan and have the phone activated (David, 1991; Feldmann and Rissen, 1993b).

3. Requirement Analysis and Design

A series of tools and methodologies were used in the design of the system. The overall system design, organizational flow and database structure are clearly represented.

3.1 Analysis of Existing Systems

Most people that currently have a backup of their mobile phone data acquired these data manually from the phone. By manually, it is meant that they scroll through the mobile phone and with the aid of a pen and paper, copy out the data displayed on the Liquid Crystal Display (LCD). Many of the other existing systems are mere modifications of this manual technique. These are besides some of the more expensive phones that port with their own data backup systems, which are usually very restrictive and would not work for any other phone.
These technique obviously are either expensive or highly time consuming and stressful. Thus, most people would rather go without a backup for the mobile phone data than go through the stress of backing up the phone data.

3.2 Requirement Analysis of the New System

The system satisfied certain quirements before it was considered satisfactory. The requirements set for the system prior to its developement are as follows:
The system should first and foremost be able to fully identify any mobile phone attatched to it. The system should then be able to read the mobile phones phonebook which consist of contact names and their phone numbers. This would then be transfered into the computer memory. The system should also be able to read text messages received through the short message service (SMS), and transfer them into the computer memory. The data read from the phone must be displayed for the user to see, and saved only if the user chooses to do so. The data stored on the computer memory should be retrievable and readable in the form of a phonebook on the computer. It should also be possible to have direct access to the list for both contacts and text messages. The user of the system should be able to transfer these data stored on the computer unto a mobile phone when he wants to. Other functionalities expected of the system is that it should be able to allow the user to send text messages from the system to any network of his choice using SMS offered by the subscribers' network. It should also support initiation and termination of calls as well as supporting multiple users.

3.3 Design of the New System

This section will be considering the design of the system using different design (Pfleeger, 2001) models to project the system from different angles of view. The different models used will allow the entire system to be projected when viewed with respect to the perspective represented. Naturally, different models are usually best suited to different views of a system under development.

3.3.1 Context Diagram (CD)

The CD is used to represent pictorially, the scope and boundaries of the area under study. The purpose is to identify what is to be included in the study. The area to be studied is shown as a single circle in the center of the

diagram. The system is not isolated but interacts with other entities, which are shown as rectangles on the context diagram as shown in Fig. 1.

3.3.2 Data Flow Diagram (DFD)

A DFD is a graphical representation of the system that shows data flows to, from and within the system. Processing functions that change the data in some manner, and the storage of the data are also depicted. The data flow diagram can be used to represent systems at different levels of depth. The level 0 (zero) data flow diagram as shown in Fig. 2 is the topmost level at which a system can be represented and has the least complexity.

3.3.3 The Unified Modeling Language (UML) Design of the system

The Unified Modeling Language is a notational approach that is popular for describing object-oriented (OO) solutions. It can be tailored to fit different development situations and software life cycles (Stevens and Pooley, 1999; Oestereich, 1999; Bennett, McRobb and Farmer, 1999). UML can be used to visualize, specify, or document a problem. UML diagrams include the dynamic view of the system, lists of activities, interactions, sequences, collaboration, and state diagrams. The UML diagrams also show the static view of the system depicted by class, package and deployment diagrams.

3.3.3.1 The State Diagram

The state diagram as show in Fig 3 & 4 shows all possible states that an object can take. The change from one state to another is triggered by a message (representing an event) that is sent from one object to another. It is usually necessary when a class has many state changes (Stevens and Pooley, 1999; Oestereich, 1999).

3.3.3.3 The Use Case Diagram

The sequence diagram describes the set of sequence of actions including variants, that a system performs that yields an observable result of value to a particular actor. A use case describes what a system (Subsystem, class, or interface) does, but it does not specify how it does it. This use case represents a functional requirement of the system as a whole (Oestereich, 1999). See Fig. 5. The interactions among classes are illustrated using interaction diagrams. The interaction diagrams are of two types; Sequence and Collaboration Diagrams.

3.3.3.2 Class Diagram of the New System

The UML class diagrams are used to document the static structure of the system. That is, it shows the classes (McKelvy, 1996) being used and how they are related, but not how they interact to achieve particular behaviors. A class diagram also shows other aspects of static structure such as packages (Stevens and Pooley, 1999). See Fig. 6.

3.3.3.4 The Package Diagram

Package diagram as shown in Fig 7 gives a high-level overview of the system and notes the high-level dependencies (Broch, Rambaugh and Jacob, 1991). The package diagram below shows how the classes are logically divided into modules. The dashed arrows show the package dependencies.

3.3.3.5 The Sequence Diagram

The Sequence diagram shows how messages flow from one object to another. It shows the sequence in which activities or behaviors occur. See Fig 8.

3.3.3.6 The Collaboration Diagram

The collaboration Diagram uses objects and sequence information to show how the objects are connected statically.

3.3.3.7 The Activity Diagram

The activity diagrams below display all activities that can occur in the system as the values of an object change (Oestereich, 1999). See Fig 10.

3.3.3.8 The Component Model

The component model shows the dependencies between parts of the code. It is primarily of interest to designers and maintainers of the system, and forms part of the development view. See Fig 11.

3.3.3.9 The Deployment Diagram

The deployment model shows the structure of the runtime system: which parts run on which processors and how the hardware is configured to provide necessary resources. It contributes to both the physical view and the process view (Fowler and Scott, 2000). The deployment diagram shows:

- The physical communication links between hardware items (machines and other resources)
- The relationship between physical machine and processes that run on them

The physical system consists of nodes with association between them. A node may be a processor or some other devices which provide services such as the mobile phone in this case. See Fig 12.

3.3.4 Database Design of the System

Here, we discuss the design of the database used for the storing of acquired data for the system.

3.3.4.1 Entity Relation Model

The entity relation model which shows the relationship between entities and the relationships between then is as shown in Fig 13.

3.3.4.2 Relational Model

The relational model is more or less a tabular representation of the entity relation model. It shows the graphical exact of the database interface when it is ready for data input. Table 1, 2, and 3.

3.3.4.3 Relational Translation

The relational translation for the system is as shown below.

Person (RegistrationNo, PersonName, Password)
Phonebook (IdentificationNo, PhoneNo, ContactName, RegistrationNo)
Text_Message (MessageNo, Message, Sender, RegistrationNo)

3.3.4.4 Graph of Relation

The Graph of Relation shows the relationship between different entities. Arrow-head lines are used to show how attributes of one entity are linked with attributes of some other entity (David, 2004). See Fig. 14.

2.3.4.5 Dictionary of Attributes

The Dictionary of attribute is a form of documentation that shows all attribute, their data types, the entity to which they belong and a description (David, 2004). See Table 4.

3.4 Input of the System

The system input would be from the memory of the SIM or mobile phone connected to the computer, or straight from the keyboard through the graphical user interface of the system resident on the computer. The mobile phone can be connected to the system at any stage of operation. Once the phone is connected, the commands issued out by pressing buttons on the graphical user interface are immediately executed. If no phone is connected, commands related to the mobile phone will be ignored.

3.5 Output of the System

The output of the system is the phonebook backup and text message backup that are retrieved from the mobile phone, and stored on the computer memory. Copies of these backup can latter be transferred to the mobile phone when the need arises. Also as output is the text messages that are and sent from the system through the short

message service offered by the network operator that supplied the SIM. Finally is the "flashing" (make and break calls) capability which is also through the network operator that supplied the SIM.

4. Summary and Conclusion

4.1 Summary of the Project

This application has been carefully designed. It is highly scalable and robust. Using the system, mobile phone users can easily and very efficiently store and retrieve the contacts and text messages on their mobile phones. At their convenienc, users can easily type text messages using the keyboard which is much faster and convenient. They can also decide to send SMS and/or simply save the message to the phone. With this system as well, 'Flashing' has been made more accurate. This work has been strategically laid out and organized to facilitate understanding of the basic concepts. Different design models such as the context diagram, dataflow diagram, and the UML design which consists of several different parts were considered. The database structure of the system was also presented using the relational database design approach.

4.2 Conclusion

This system has undergone a series of unit testing, integration testing, and installation testing. It has been found to effectively perform all the tasks laid out at the requirement analysis phase. All functions, subroutines, modules, classes and objects are fully functional and produce the expected results. It was designed in such a way as to allow for easy expansion or upgrade. Once new classes are developed, allowances have been provided within the application where they can easily be plugged and made functional. The system has been implemented, and it runs perfectly well. Contributions are very much welcomed.

References

Bennett, S., McRobb, S. & Farmer, R., (1999), Object Oriented systems analysis and design using UML, McGraw-Hill.

Broch, G., Rambaugh, J. & Jacobs, I., (1991), The Unified Modeling Language (UML): User Guide, Addison-Wesley.

David, A., (2004), Data Modeling and Relational Model. Summer School at the Department of Computer Science, University of Ibadan.

David, C., (1991), The pan-European cellular mobile radio system. In R.C.V. Macario, Personal and Mobile Radio Systems. Peter Peregrinus, London.

Feldmann, M. and Rissen J.P., (1993b) GSM Network Systems and Overall System Integration. Electrical Communication.

Fowler, M. & Scott, K., (2000), UML Distilled, Addison Wesley.

Maduka I., (2005), Development of a Phonebook Software, Thesis submitted to the Department of Computer Science, University of Ibadan.

McKelvy, M., (1996), MCSD: Visual Basic 6 Desktop Application Study Guide. Gira Sound.

Oestereich, B., (1999), Developing Software with UML, Addison Wesley.

Pfleeger, S. L., (2001), Software Engineering Theory and Practice.

Stevens, P. & Pooley, R., (1999), Using UML Software Engineering with Objects and Components. Addisson-Wesley.

Kehinde Aregbesola had his secondary education at Lagelu Grammar School, Agugu, Ibadan, where he was the Senior Prefect (2000). He obtained his first and second degrees in computer science from the prestigious University of Ibadan - a former college of the University of London – (2006, 2009). He is an experienced solutions developer with several years in the industry. He has been involved in the development of diverse kinds of applications currently in use in different organizations, as well as a few tools currently in use by other software developers. He has implemented projects with a few prominent ICT companies including LITTC, Microsolutions Technology, Farsight Consultancy Services, Chrome Technologies, Infoworks, etc. His focus is to be a pure blend of academic excellence and industrial resourcefulness. He is a member of the Computer Professionals of Nigeria CPN (2008), Nigeria Computer Society NCS (2011), Nigerian Institute of Management NIM (2008), ISACA (2012), and Project Management Institute PMI (2012). He is a certified Project Management Professional PMP (2012), and currently a Lecturer at Salem University, Lokoja, Kogi State, Nigeria.

Oluwaseyitan Osunade
OSUNADE Oluwaseyitan is a member of IEEE and the Nigeria Computer Society**.** He has a PhD in Computer Science from University of Ibadan, Nigeria. He specializes in computer networks and data communication systems

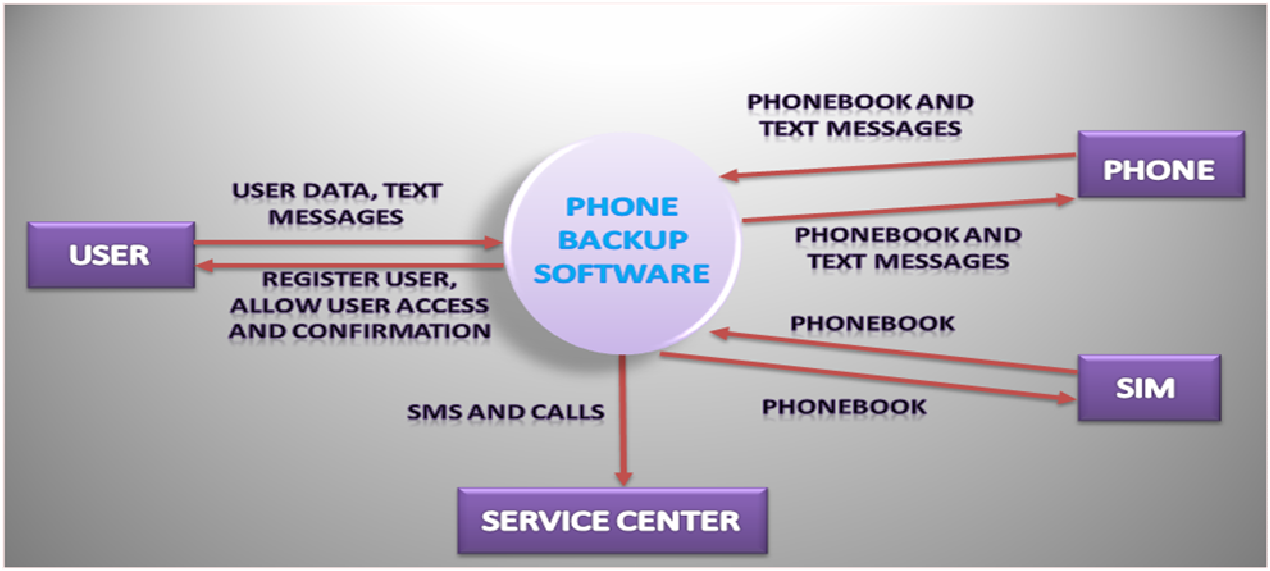

FIG. 1: THE CONTEXT DIAGRAM OF THE SYSTEM

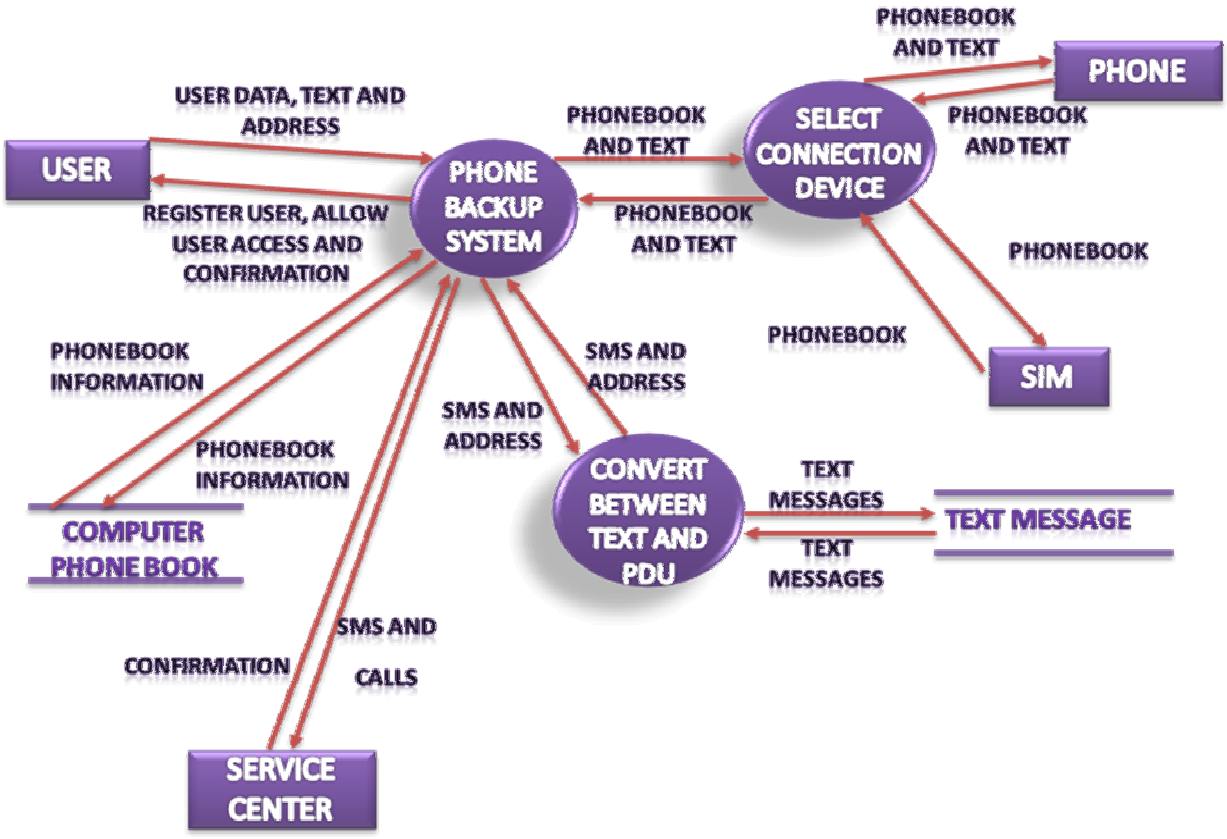

FIG. 2: THE LEVEL 0 DATA FLOW DIAGRAM (TOP LEVEL) FOR THE SYSTEM

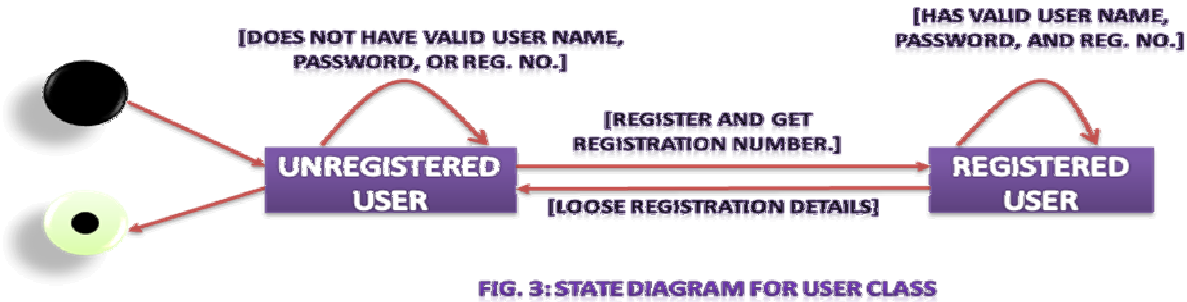

FIG. 3: STATE DIAGRAM FOR USER CLASS

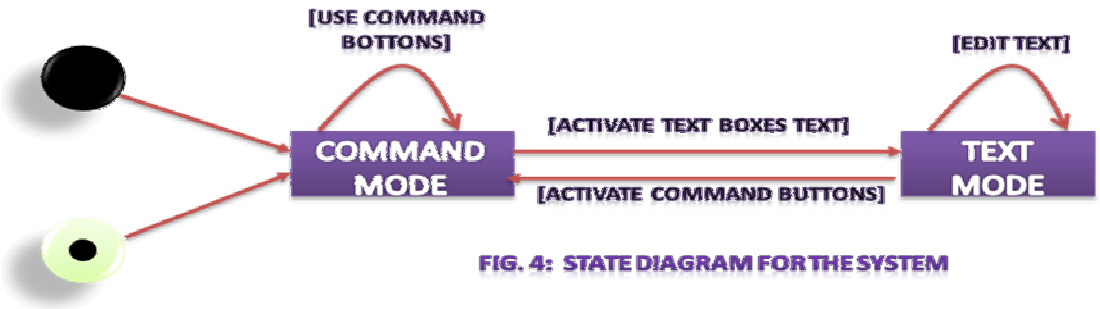

FIG. 4: STATE DIAGRAM FOR THE SYSTEM

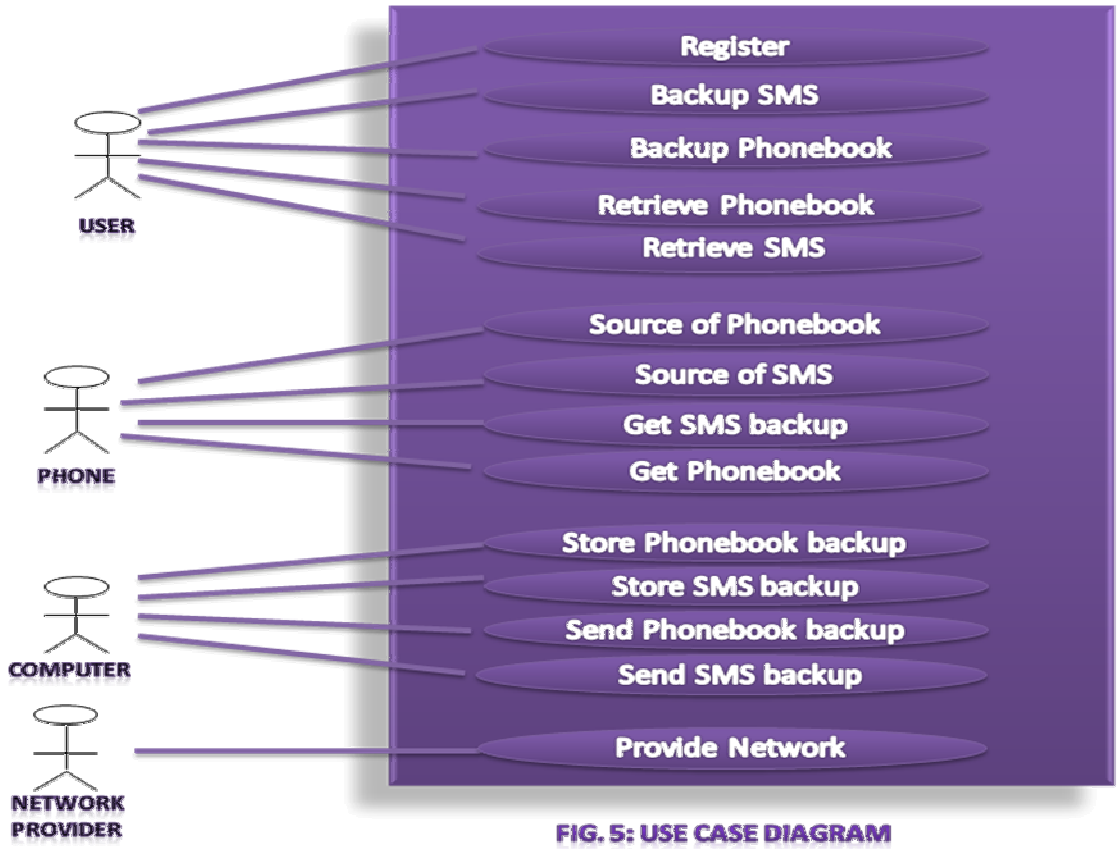

FIG. 5: USE CASE DIAGRAM

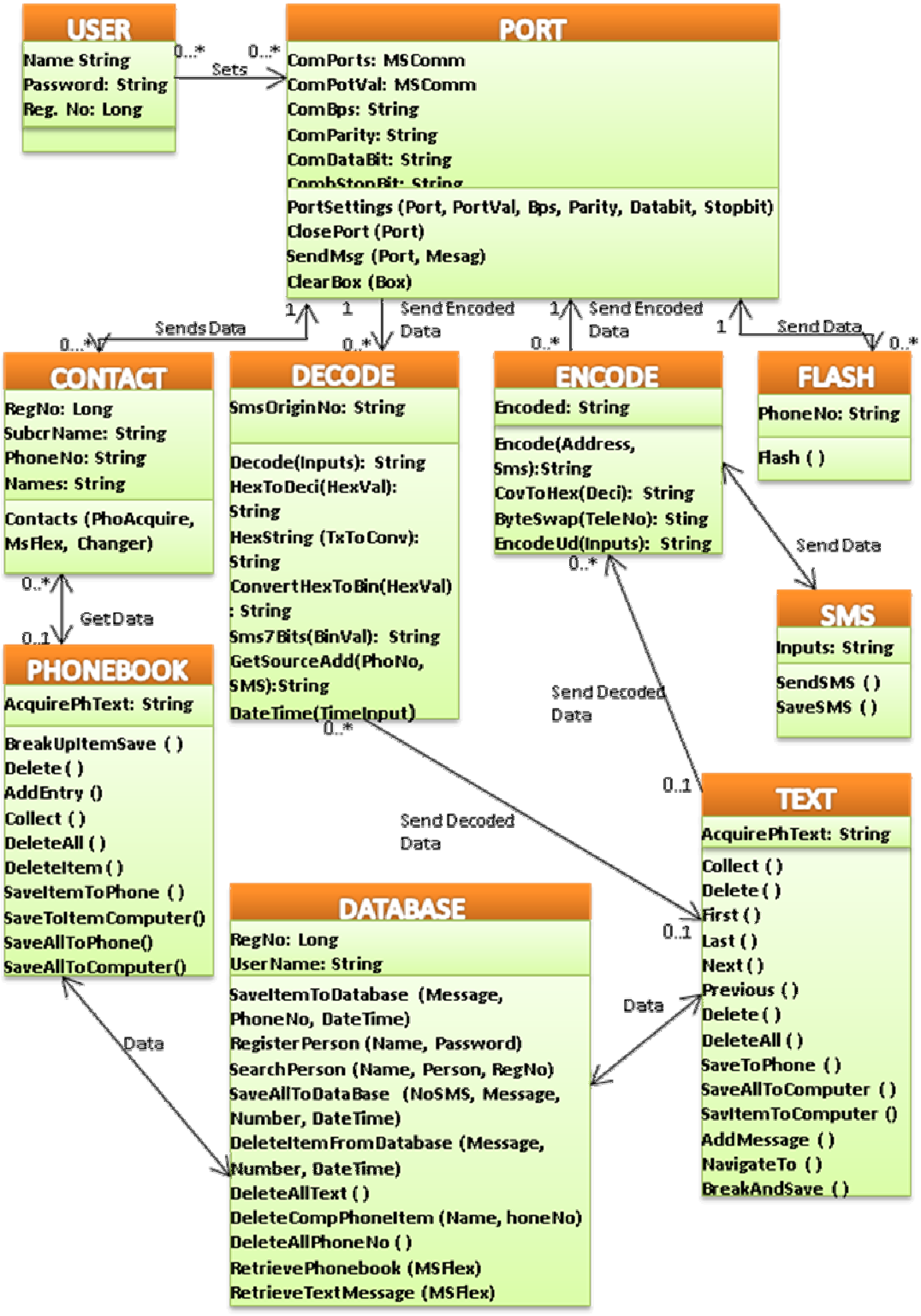

Fig. 6: Class Diagram

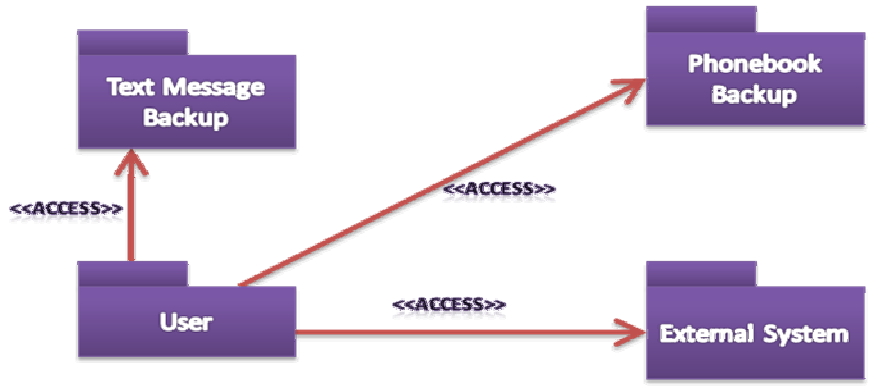

FIG 7: PACKAGE DIAGRAM FOR THE SYSTEM

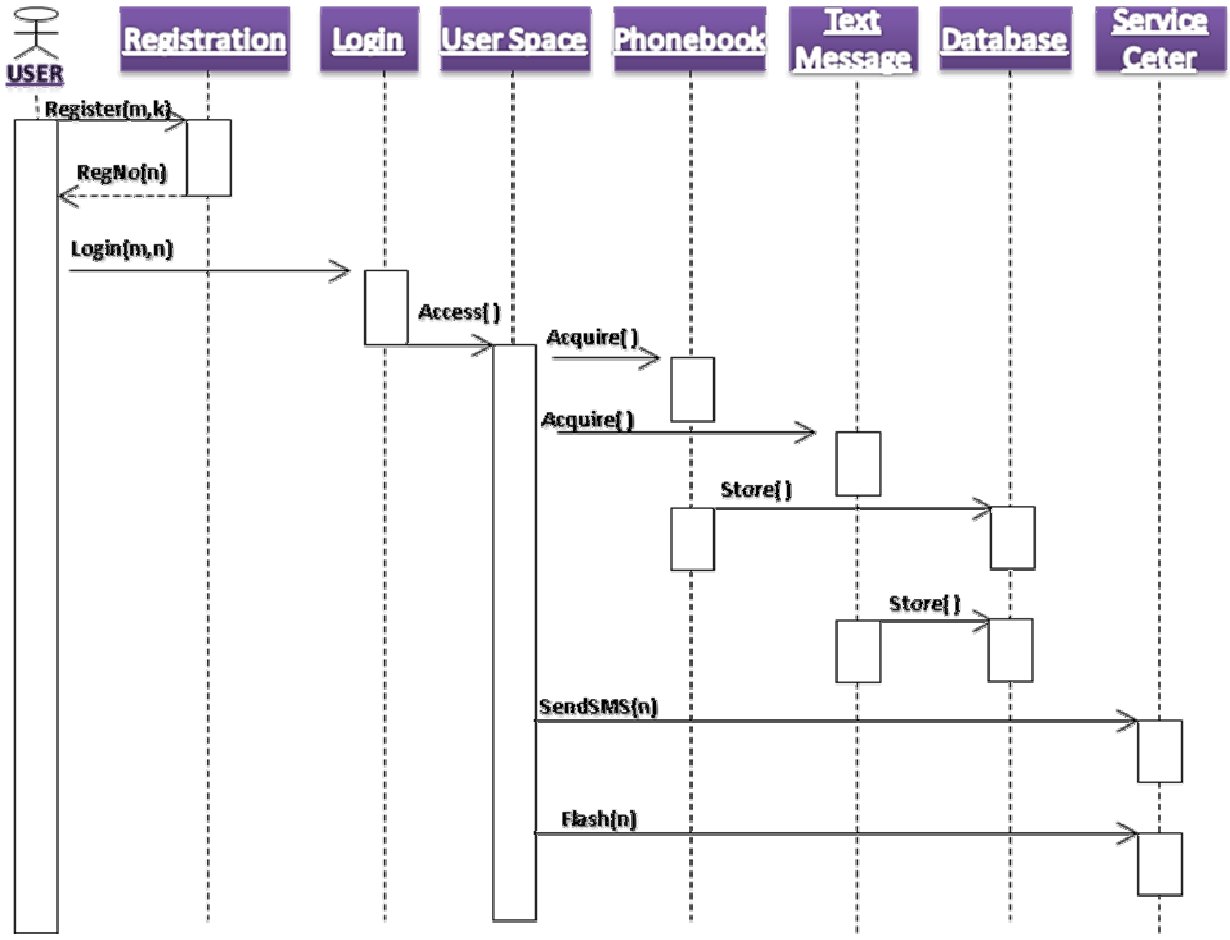

FIG. 8: THE SEQUENCE DIAGRAM

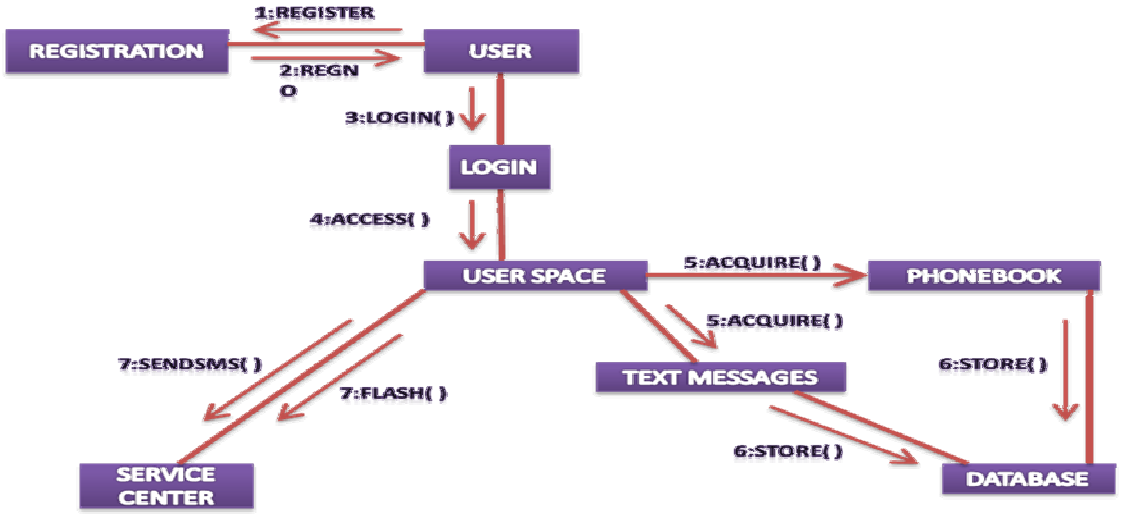

FIG. 9: COLLABORATION DIAGRAM

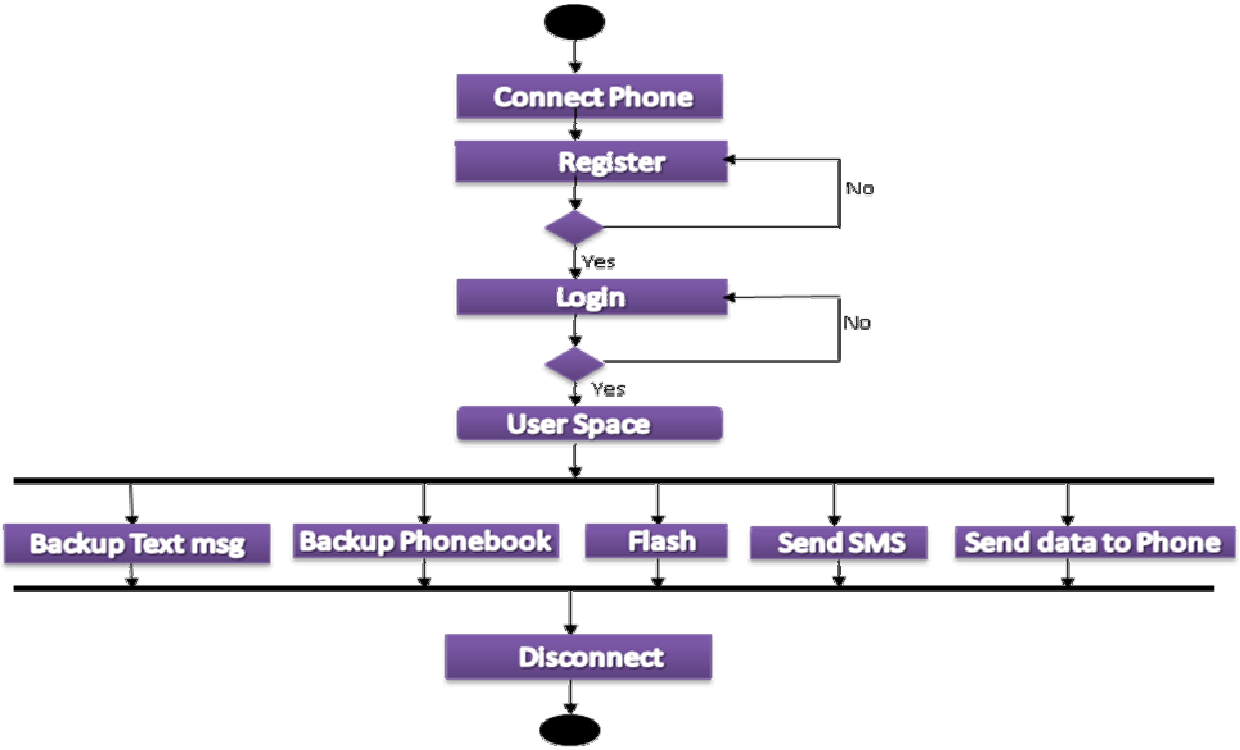

FIG 10: ACTIVITY DIAGRAM FOR THE SYSTEM

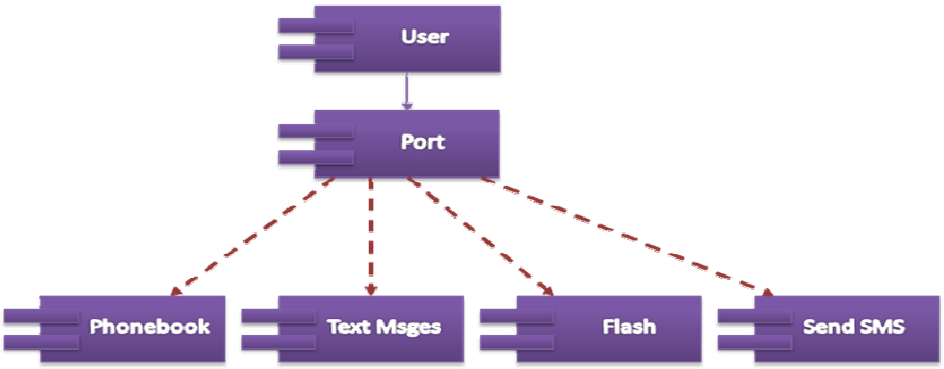

FIG. 11: COMPONENTMODELS

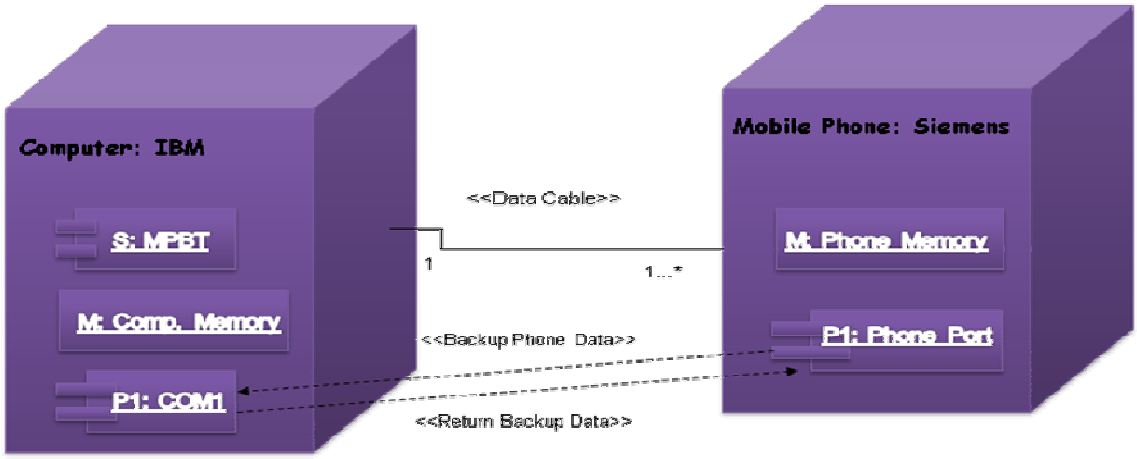

FIG 12: DEPLOYMENT DIAGRAM

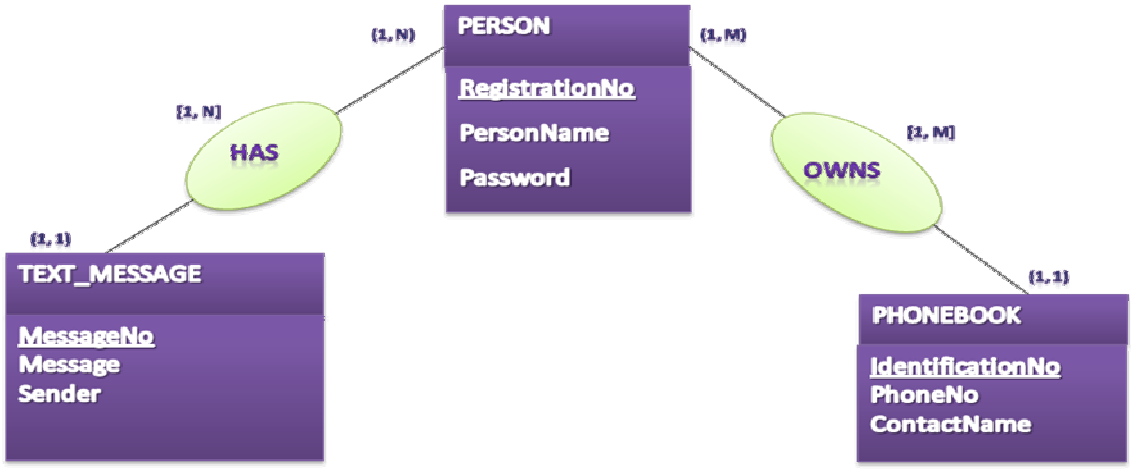

FIG 13: ENTITY RELATION MODEL

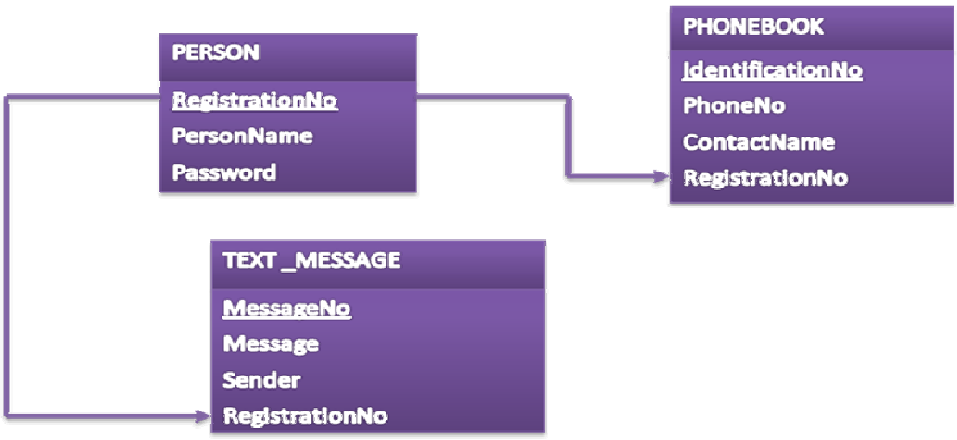

FIG 14: GRAPH OF RELATION

TABLE 1: RELATIONAL MODEL - PERSON

RegistrationNo	PersonName	Password

TABLE 2: RELATIONAL MODEL – PHONEBOOK

IdentificationNo	PhoneNo	ContactName	RegistrationNo

TABLE 3: RELATIONAL MODEL – TEXT MESSAGE

MessageNo	Message	Sender	RegistrationNo

Table 4: The dictionary of Attributes for the system

S/N	Attribute Name	Data Type (Domain)	Entity	Description
1	RegistrationNo	Number	Person	Identifies each subscriber
2	PersonName	Text(20)	Person	Subscriber name
3	Password	Text(20)	Person	Subscriber password
4	MessageNo	Number	Text_Message	Identifies each text message
5	Message	Text(160)	Text_Message	The message
6	Sender	Number	Text_Message	The number that sent the message
7	IdentificationNo	Number	Phonebook	Identifies each phone book entry
8	PhoneNo	Number	Phonebook	The contact phone numbers
9	ContactName	Text(20)	Phonebook	The contact names