



A Mobile System Implementation for Student Support Service in Java.

Frank Appiah

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

January 11, 2021

A Mobile System Implementation for Student Support Service in Java.

Frank Appiah , *Member, IEEE*

Abstract—With Java library, smslib a wireless data communication system is implemented for servicing students in their daily academic or semester activities with their mobile devices at campus. The activities of students include asking of vacation date, examination dates, general performance average or cumulative weighted average, classes timetable, current fee values and changed timetable. With this student support service, more can be implemented in future needs.

Index Terms—SMS, Mobile Service, Student Support Service, Wireless Data, Java.

-
- Frank Appiah is with Kwame Nkrumah University of Science and Technology, Kumasi, Ghana . E-mail: appiahnsiahfrank@gmail.com
-
- ◆

1 INTRODUCTION

In this present age, Humans are gradually being replaced by computers in order for tasks and processes to be easily done. To have a dynamic intelligent information retrieval system[21] whereby people can retrieve certain kinds of information based on their needs and wants at any convenient time, day or night is one area in which a computer technology will do well to replace a human but not completely.

Providing a system in an educational facility where students can make telephone calls or send sms text messages to access information about their Cumulative Weighted Average, trailed courses as well as courses for a particular semester, time table on specific days and others will reduce the burden students go through in order to access basic information. Short Messaging Service (SMS) is a technology that enables short messages of about 160 characters to be sent from one mobile phone to the other. It can be used to send regular text as well advanced content such as operator logos, ringing tones, phone configurations etc.

SMS[4] Appeared in Europe in 1991. The Global System for mobile communications included SMS from the outset. SMS is built on digital wireless standards such as code division multiple access (CDMA), time division multiple access (TDMA) and GSM. SMS provides a mechanism for transmitting data to and from wireless devices. This makes use of a short messaging servicing centre (SMSC). The SMSC acts as a store and forward

system for short messages. In contrast to other existing text messaging services such as alphanumeric-paging, the service elements are designed to provide guaranteed delivery of text messages to the destination. A very distinguishing aspect of SMS is that a mobile phone or handset is able to deliver or receive messages irrespective of whether a voice or data call is in session or not. SMS is characterized by out of band packet delivery and low bandwidth transfer which results in efficient means of transmitting short burst of data.

Modem is an acronym that stands for modulator demodulator. A modem is a simple device which is capable of converting analogue signals to digital ones and digital signals to analog. Modems used to be a separate hardware third party users that had to be purchased separately. However, today's computer manufacturers have made it one of the primary hardware components found on any computer. It is very rare to come across a computer without a modem.

This system has as its objectives to provide flexible dynamic access to basic information especially in this situation where most students do not have constant internet access.

This study work will describe the hardware interface along its logically requirements in the MobileSystem implementation. In terms of language code in java, I will code the actual Java-based language[14,15,16,17] in the support service.

2 MOBILE SYSTEM

The hardware interface is represented as a *MobileSystem* interface as listed in code 1. The hardware interface is a wireless modem[20](modulation demodulation) that physically connects to a computer system with a Sim Card from a network operator based in Ghana in this case study. With Short messaging service(SMS) support from sms center of the network operator, the wireless modem is located from Over the Air Programming Protocol (OTAP)[18, 19] to enable the student service to reply back to the sender via a notification handler in the Service class, *svr*. The modem is represented as a *ModemGateway* class in this implementation. The *mobile system* of the service has about 9 methods in the functional side of implementation. The *sendrequest* method works by sending text to a current student in request of information. This is listed in *MobileSystem* interface and implemented in the *MobileSystemImpl* class. *IsConnect* method does indicate on invocation whether the physical modem is really connected or not to the computer system. Mobile receiver like the physical wireless device cannot operate properly on low signal state and so it is important to check on the signal state if location adjustment is needed some times. The *lowSignalLevel* method will provide the indicator in the system implementation. The wireless modem has its own sim card to be able to receive messages. The phone number of the device is retrieved by *getPhoneNumber* method. The wireless modem is initiated in the *MobileSystem* interface by loading the connection properties. The properties include manufacturer, terminalname, modeltype, baudrate and port of connected device. The *MobileSystemImpl* uses the *ModemGateway* to implement the functionality of sending and receiving messages from end users. In order to send and receive, the inbound and outbound flags have to be set to true respectively. The wireless modem is logically instantiated as *Serial ModemGateway* with the terminal name, port, baudrate, manufacturer and a logger.

```
modem=new SerialModemGateway(terminal, port, baud, man,
    "", srv.getLogger());
modem.setInbound(true);
modem.setOutbound(true);
```

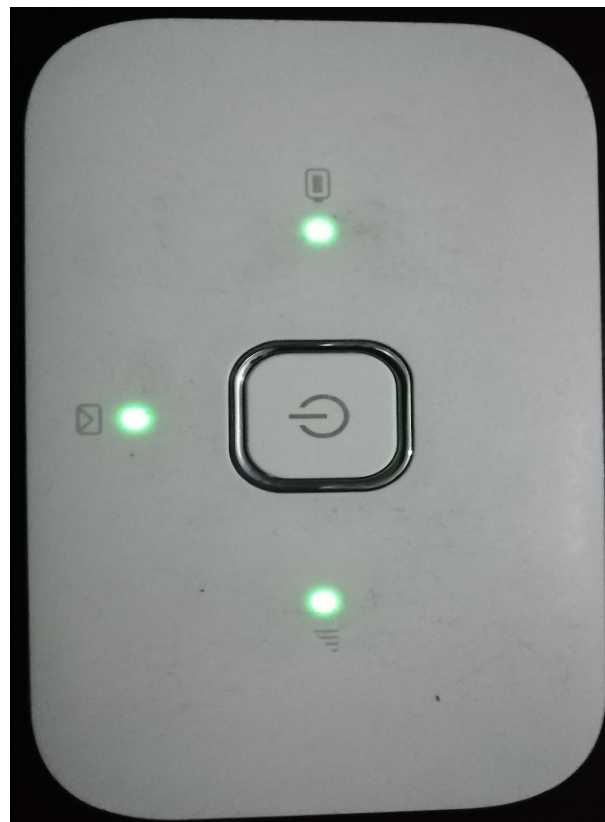
The USB powered wireless modem(Figure 1: off, Figure 2: on state) is termed as a *Serial ModemGateway*.

The construction of the *MobileSystemImpl* takes one argument which the console view. This console is implemented as *AINRSView* class. The fields of the implementation class includes service, baudrate, terminalname, model type, port, manufacturer, modem and console. The Console is used to log information on activity as it happens in the runtime.

Figure 1: Wireless Modem on Switch Off



Figure 2: Wireless Modem on Switch On State



2.1 Notifications

There are three main notifications in the messaging system built to support the services. These includes:

- **InboundMessageNotification** :This notifier is responsible for the incoming messages from the end users--The students.
- **OutboundMessageNotification** : This notifier is responsible for all outgoing messages from the mobile-2-system to the end users.
- **Callnotification** : This notifier is responsible for in calling times and it is normally used to cut calls from being accepted during service times.

After the instantiation of the serialmodem, the flags of inbound and outbound processing are enabled. Without that modem will be out of service. The inboundnotification is then instantiated by passing arguments *console* and *service, srv*. This inboundnotification is set to the Serialmodem. The OutboundMessageNotification is set also after the instantiation stage. Finally, the Callnotification is instantiated and set to the modem. The modem is finalised in preparation to send and receive messages. This serialmodem is set to the service as gateway. With the modem gateway set, the service will now start. If case of updation or troubleshooting, the service can be stopped and restarted.

3 MOBILE SYSTEM IMPLEMENTATION

This section shows the mobile system in actual implementation with smslib[22], java modem communication library. This modem communication is based on attention commands, Hayes commands set. There are only two code listing namely MobileSystem and the implementation, MobileSystemImpl.

3.1 Appendices

CODE LISTING : ainrs.telephonesystem

CODE 1: MobileSystem

```
package ainrs.telephonesystem;

import org.smslib.Service;
import org.smslib.gateway.ModemGateway;

/**
 *
 * @author frank appiah
 */
public interface MobileSystem {

    public void sendRequest(String text)
;

    public boolean isConnect();

    public boolean lowSignalLevel();
```

```
public String getPhoneNumber();

public void init();

public Service getSrv();

public String getConnParam();

public ModemGateway getModem();

public void stop();
}
```

CODE 2: MobileSystemImpl

```
package ainrs.telephonesystem;

import ainrs.AINRSView;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;
import java.util.Timer;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import org.smslib.Service;
import org.smslib.gateway.ModemGateway;
import org.smslib.gateway.SerialModemGateway;

/**
 *
 * @author frank appiah
 */
public class MobileSystemImpl
implements MobileSystem{
    Service srv;
    ModemGateway modem;
    static final int LOW=50;
    String man,model,terminal,port;
    int baud;
    AINRSView console;
    InboundMessageNotification
inboundnotification;
    OutboundMessageNotification
outboundnotification;

    public MobileSystemImpl(AINRSView
console) {
        srv=null;
        modem=null;
        man="";
        model="";
        terminal="";
        port="";
        baud=300;
        this.console=console;
    }

    public void init() {
```

```

    FileInputStream fis=null;

    try {
        fis = new
FileInputStream("Connect.properties");
        Properties prop=new
Properties();
        prop.load(fis);
        man=
prop.getProperty("manufacturer","Nokia
");
        //
System.out.println("manufacturer
:"+man);

model=prop.getProperty("modeltype","")
;
        // System.out.println("model
:"+model);

terminal=prop.getProperty("terminalnam
e","modem");

//System.out.println("terminal
:"+terminal);

port=prop.getProperty("port", "COM4");
        //System.out.println("port
:"+port);

baud=Integer.parseInt(prop.getProperty
("baudrate","300"));
        // System.out.println("baud
rate :"+baud);

        srv=new Service();

        modem=new
SerialModemGateway(terminal, port,
baud, man, "", srv.getLogger());
        modem.setInbound(true);
        modem.setOutbound(true);
        inboundnotification=new
InboundMessageNotification(console,
srv);
        outboundnotification=new
OutboundMessageNotification(console);

modem.setInboundNotification(inboundno
tification);

modem.setOutboundNotification(outbound
notification);

modem.setCallNotification(new
CallNotification(console));

        srv.addGateway(modem);
        srv.startService();
    } catch (Exception ex) {

JOptionPane.showMessageDialog(null,
ex);
    } finally {
        try {
            fis.close();
        } catch (IOException ex) {

```

```

JOptionPane.showMessageDialog(null,
ex);
        }
    }

    public void sendRequest(String
text){

    }

    public boolean isConnect() {
        return modem.isStarted();
    }

    public boolean lowSignalLevel(){
        try {
            Timer timer=new Timer();
            if (modem.getSignalLevel()
< LOW) {
                return true;
            } else {
                return false;
            }
        } catch (Exception ex) {

Logger.getLogger(MobileSystemImpl.clas
s.getName()).log(Level.SEVERE, null,
ex);
            return false;
        }
    }

    public String getPhoneNumber() {
        return "0244265562";
    }

    public ModemGateway getModem() {
        return modem;
    }

    public void setModem(ModemGateway
modem) {
        this.modem = modem;
    }

    public Service getSrv() {
        return srv;
    }

    public void setSrv(Service srv) {
        this.srv = srv;
    }

    public String getConnParam(){
        return
terminal+":"+port+":"+baud+":"+man+":"+
+model;
    }

    public void stop() {
        try {
            if (srv != null && modem
!= null) {
                srv.stopService();

```

```

    }
    } catch (Exception ex) {

JOptionPane.showMessageDialog(null,
ex);
    }
}
}

```

4 CONCLUSION

This study of student support service is a telecommunication application on computer system with wireless GSM[1,2,3] network operating on a network provider or operator. This service will render an intelligent answering on behalf of student's short message service identified by their mobile phone number.

REFERENCES

- [1] Holmes, H., Munro, R., Hall, R., & Kerr, J. (2000). U.S. Patent No. 6,134,432. Washington, DC: U.S. Patent and Trademark Office.
- [2] Kesanupalli, R., Akella, S., Lakshminarayanan, G., & Krishnamshetty, R. (2003). U.S. Patent No. 6,603,837. Washington, DC: U.S. Patent and Trademark Office.
- [3] Peersman, C., Cvetkovic, S., Griffiths, P., & Spear, H. (2000). The global system for mobile communications short message service. *IEEE Personal Communications*, 7(3), 15-23.
- [4] Trosby, F., Holley, K., & Harris, I. (2010). Short message service (SMS): the creation of personal global text messaging. John Wiley & Sons.
- [5] Prabhakaran, L., Chee, W. Y., Chua, K. C., Abisheganaden, J., & Wong, W. M. (2010). The use of text messaging to improve asthma control: a pilot study using the mobile phone short messaging service (SMS). *Journal of telemedicine and telecare*, 16(5), 286-290.
- [6] Gong, J., & Tarasewich, P. (2004, November). Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting* (pp. 3751-3756).
- [7] Yogish, D., Manjunath, T. N., & Hegadi, R. S. (2017, December). Survey on trends and methods of an intelligent answering system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)* (pp. 346-353). IEEE.
- [8] Feng, D., Shaw, E., Kim, J., & Hovy, E. (2006, January). An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th international conference on Intelligent user interfaces* (pp. 171-177).
- [9] Pudaruth, S., Boodhoo, K., & Goolbudun, L. (2016, March). An intelligent question answering system for ict. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 2895-2899). IEEE.
- [10] Mattsson, H., Nilsson, H., & Wikström, C. (1999, January). Mnesia—A distributed robust DBMS for telecommunications applications. In *International symposium on practical aspects of declarative languages* (pp. 152-163). Springer, Berlin, Heidelberg.
- [11] LaPadula, M. (2003). A comprehensive look at online student support services for distance learners. *The American Journal of Distance Education*, 17(2), 119-128.
- [12] Britto, M., & Rush, S. (2013). Developing and implementing comprehensive student support services for online students. *Journal of Asynchronous Learning Networks*, 17(1), 29-42.
- [13] Grant-Vallone, E., Reid, K., Umali, C., & Pohler, E. (2003). An analysis of the effects of self-esteem, social support, and participation in student support services on students' adjustment and commitment to college. *Journal of College Student Retention: Research, Theory & Practice*, 5(3), 255-274.
- [14] Arnold, K., Gosling, J., Holmes, D., & Holmes, D. (2000). *The Java programming language* (Vol. 2). Reading: Addison-wesley.
- [15] Farrell, J. (2011). *Java programming*. Cengage Learning.
- [16] Liang, Y. D. (2003). *Introduction to Java programming*. Pearson Education India.
- [17] Gosling, J., Holmes, D. C., & Arnold, K. (2005). *The Java programming language*.
- [18] Arakadakis, K., Charalampidis, P., & Fragkiadakis, A. (2020). Firmware over-the-air programming techniques for IoT networks—A survey. *arXiv preprint arXiv:2009.02260*.
- [19] Aschenbruck, N., Bauer, J., Bieling, J., Bothe, A., & Schwamborn, M. (2012, July). Selective and secure over-the-air programming for wireless sensor networks. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)* (pp. 1-6). IEEE.
- [20] Kumar, P., Bhrdwaj, V., Pal, K., Rathor, N. S., & Mishra, A. (2012). GSM based e-Notice Board: wireless communication. *International Journal of Soft Computing and Engineering*, 2(3), 601-605.
- [21] Appiah, FK (2017). *Automatic Information Retrieval System: Intelligent Computer Communication*.
- [22] Bingwen, C. S. W. (2011). *Design and Implementation of Data Acquisition System Based on SMSLib*. *Computer & Digital Engineering*, 12.



Dr. Frank Appiah. He is a holder of Bsc(Hon) from Kwame Nkrumah University of Science and Technology in 2018, Msc in Advanced Software Engineering from King's college London in 2010 and PhD in computer science and engineering from both KCL (2012/2014) and KNUST (2014) respectively. Frank Appiah has professional certificates in Management and engineering since 2011. He developed StreamEPS - Stream Event Processing System in 2011 which is hosted at Github. This work is a PhD candidate at KNUST, Department of Computer Engineering, Kumasi, Ghana. This work is awarded course leadership programme by Kwame Nkrumah University of Science and Technology.