WEEK 5 - DEMO CODE

Wireless Messaging API – SMS, Binary & MMS

```
** @author George Nguyen <George.Nguyen@rmit.edu.vn>

public class W6SMSDemo extends MIDlet implements CommandListener, Runnable {

    MessageConnection conn;
    private Form fr = new Form("Send SMS");
    private TextField dest = new TextField("Message", "", 16, TextField.ANY);
    private Command exit = new Command("Exit", Command.EXIT, 0);
    private Command send = new Command("Send", Command.OK, 1);
    private Command receive = new Command("Receive", Command.OK, 2);
    private Command stop = new Command("Stop", Command.OK, 3);
    private StringItem label = new StringItem("Status:", "");
    static StringItem receiveSMS = new StringItem("Received SMS:", "None");
    private Thread t = null;
    private SMSReceiver u = null;

    public void startApp() {
        Display mDisplay = Display.getDisplay(this);
        mDisplay.setCurrent(fr);

        fr.append(dest);
        fr.append(label);
        fr.append(receiveSMS);

        fr.addCommand(send);
        fr.addCommand(exit);
        fr.addCommand(receive);
        fr.addCommand(stop);

        fr.setCommandListener(this);

    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
```

```
public void commandAction(Command c, Displayable d) {

    if (c.getLabel().equals("Send")) {
        t = new Thread(this);
        t.start();
    } else {
        if (c.getLabel().equals("Receive")) {
            receiveSMS.setText("Start receiving ...");
            u = new SMSReceiver();
            u.start();
        } else {
            if (c.getLabel().equals("Stop")) {
                if (u != null) {
                    u.setReceiving(false);
                }
            } else {
                notifyDestroyed();
            }
        }
    }
}

public void run() {
    try {
        String port = "1234";
        String recipient = "+5550001";
        String protocol = "sms://" + ":" + port;
        String add = "sms://" + recipient;

        conn = (MessageConnection) Connector.open(protocol);
        TextMessage txt = (TextMessage) conn.newMessage(MessageConnection.TEXT_MESSAGE);
        txt.setAddress(add);
        txt.setPayloadText(dest.getString());
        label.setText("Sending SMS...");
        conn.send(txt);
        label.setText("SMS Sent sucessfully...");
    } catch (IOException ioe) {
        System.out.println("IO");

    } finally {
        try {
            conn.close();
```

```
            } catch (IOException ex) {
                System.out.println("Cannot close the connection !");
            }
        }
    }
}

class SMSReceiver extends Thread {

    private String port = "1234";
    private String protocol = "sms://" + ":" + port;
    private String message = "";
    private TextMessage txt;
    private MessageConnection conn;
    private boolean receiving = true;

    public void setReceiving(boolean receiving) {
        this.receiving = receiving;
    }

    public void run() {

        try {
            conn = (MessageConnection) Connector.open(protocol);
            while (receiving) {
                txt = (TextMessage) conn.receive();
                message = txt.getPayloadText();
                W6SMSDemo.receiveSMS.setText(message);
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                conn.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }

        }

    }

}
```