

Stevens Institute of Technology
Hoboken, New Jersey



Bluetooth Enabled Point of Sale System

EE/CPE 423
Group # 10

December 6, 2005

Group Members:
Lauren Hollahan
Melissa Biel
Matthew Kelly
Liza Robles

Faculty Technical Advisor : Yu-Dong Yao

Signed : _____

We pledge our honor that we have abided by the Stevens Honor System.

Table of Contents

I.	Abstract	p 3
II.	Project Proposal	
	a. Introduction	p 4
	b. Design Requirements	p 5
	c. System Diagram	p 6
	d. Design Approach	p 6
	e. Financial Budget	p 11
	f. Project Schedule	p 12
III.	Conclusion	p 13
IV.	References	p 14
V.	Appendices	
	a. Bluetooth Connection Flowchart	p 15
	b. Working source code for BluCoupon Server	p 16

Abstract

Originally, cellular telephones were introduced to allow users to send and receive calls wherever they may be. As cellular telephone technology increases, these telephones are becoming smaller, lighter, and more feature laden. What used to be for merely sending and receiving calls now acts as a camera, planner, video game console, web browser, and even an alarm clock. Cellular telephones are thought of as a way to make life easier.

GenBlu set out to make cellular telephones even more of a convenience and time saver. The group set out to make the in store shopping experience more informative and convenient. All a customer has to do is download the GenBlu software onto his cellular phone, and he is ready to begin saving time and money with GenBlu's innovative software. With the GenBlu software on a user's phone, the user will be able to roam the aisles of a store and receive valuable coupons automatically from the Bluetooth access points strategically placed around the store. The program will also allow a user to input and save his credit card information, which can be used to complete a transaction in a GenBlu enabled store. This is the one drawback of the project, that users must have a Bluetooth enabled cellular telephone. However, nearly all new cellular telephones have Bluetooth as a standard feature. This makes the GenBlu project marketable to anyone with a cellular telephone and a credit card.

Introduction

Cellular telephones have become more advanced since their inception in 1973. Originally, cellular telephones were large and bulky objects, the first cellular telephone weighed over two pounds. Over the years, cellular telephones have become smaller, more compact, and more powerful. Now they are almost an integral part of daily life, and many people cannot leave their homes without their cellular telephone.

As the technology used in cellular telephones advanced, their features did as well. Today one can do a number of things on his or her cellular telephone. Cellular telephones are now fully equipped with cameras, calendars, games, and can even play mp3s. Beyond that, the majority of cellular telephones today are also equipped with Bluetooth technology.

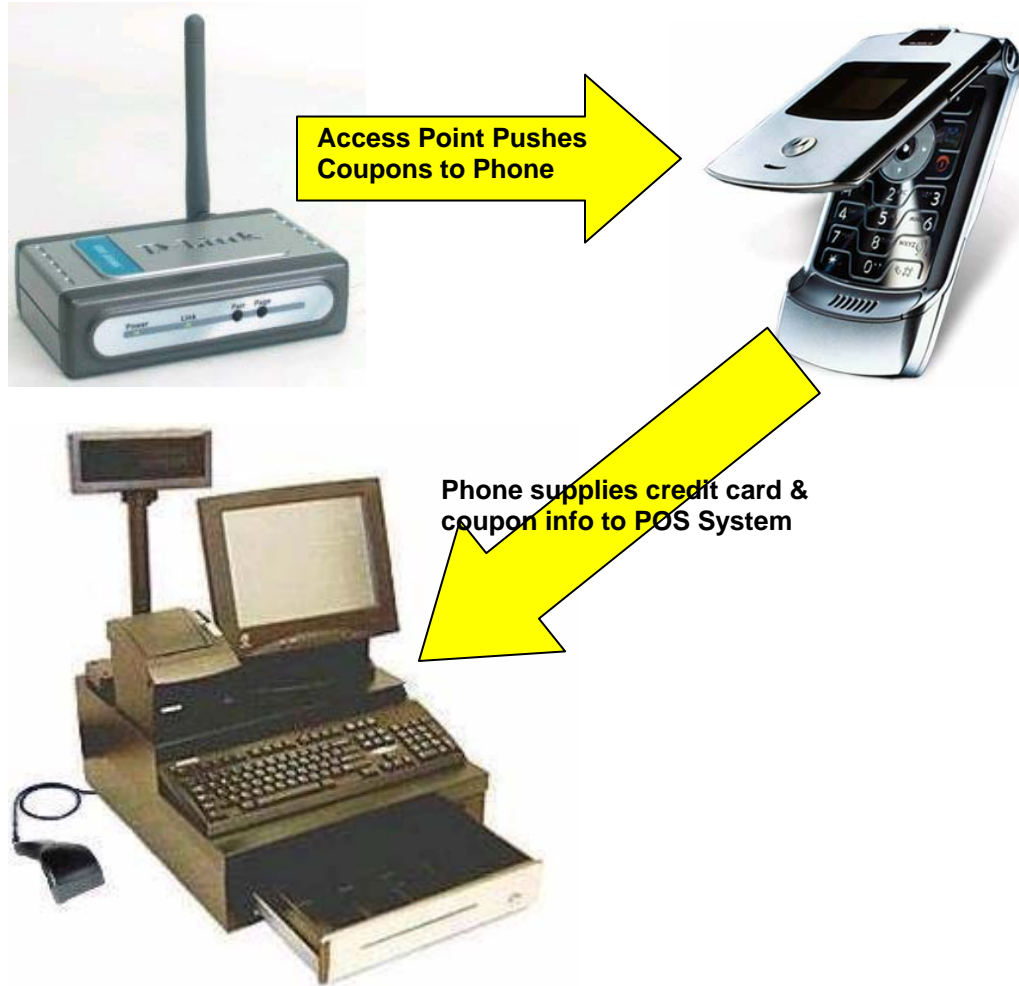
The group plans to take advantage of the new technology of cellular telephones before anyone else. The project, GenBlu, plans to empower cellular telephone users in a whole new way by adding a new convenience to their everyday lives. GenBlu intends to redefine the cellular telephone by adding the ability to make purchases and receive coupons, all with minimal effort on the consumer's behalf. Users will be able to make purchases using their pre-configured cellular telephone, eliminating the need to carry cash or credit cards.

Design Requirements

Using standard Bluetooth technology, a Java applet will be created for use on the cellular telephone. To create this applet, the group will utilize the Java 2 Micro Edition platform, more commonly known as J2ME. For testing and development purposes, a Motorola Razr V3 telephone will be used. This applet serves two purposes. First, it will allow the user to enter credit or debit card information. The user will have a pin number that they will use when making purchases (similar to how one enters a debit card pin today) in order to ensure that they are being charged for the correct purchases, and only the correct purchases. In addition, the application will also automate the connection process to the Bluetooth access points, both on the register, as well as on the coupon access points. The Bluetooth access points will use BlueZ, the official Linux Bluetooth protocol stack, which uses OBEX to transfer the coupons and other information from the access points to the cellular telephones.

The register will consist of a personal computer running Linux and cash drawer, along with a Point of Sale program loaded into memory that will control the cash drawer as well as the scanning of bar codes. It will be identical in appearance and function to ordinary registers, as to keep the overhead of the stores that will be implementing this product to a minimum. The last thing anyone wants to do is require stores to re-train all of their personnel. A Java application will be integrated into the Point of Sale program that will control the automated Bluetooth connection to the cellular telephone, and OBEX once again will handle the transfer of data between the cellular telephone and Bluetooth device, this time the register.

System Diagram



Design Approaches

Currently, there is development being conducted by major corporations on a product that will allow a customer to pay for his or her purchases via a cellular telephone. However, most of the techniques being developed require the customer to buy a new cellular telephone equipped with RFID or other technologies. GenBlu set out to implement the same functionality as these products, but using common cellular

telephone features and technologies. The group then decided to expand upon this idea, and include access points around a store that would distribute ads and coupons to the customers' cellular telephones.

The most obvious wireless standard to choose for this product was Bluetooth. Bluetooth is a common feature on almost all new telephones, and because it has the desirable short connection distance, Bluetooth became the basis of the communications for this project. RFID was investigated briefly, but since it was not a common component of cellular telephones, Bluetooth prevailed.

Bluetooth access points would have to be implemented throughout the store in strategic locations in order to push the ads and coupons out as well. Initially, these access points were to be designed and constructed by the group, but due to time and monetary constraints, this idea was abandoned. The access points will now be laptop computers with access points attached to them via a USB cable. This allows easier programming and debugging of the software and eases the design process significantly.

In addition to the access points, the group also needed a cellular telephone for development. The group chose the Motorola Razr V3 for a variety of reasons. First and foremost, it has Bluetooth capabilities. In addition, the Motorola Razr V3 uses a standard USB port for both data transfer and charging. Although you can transfer limited data via Bluetooth with the Razr, V3 you cannot add applications. The USB port, combined with the ease of use of Motorola Mobile Phone Tools will ensure that the

group will not run into problems in the last mile of the project. While the Razr V3 only has 5 megabytes of memory (which many may see as a detrimental feature), that makes it an even better candidate for testing. Due to this, the developers of the group will be forced to develop the java applet within these memory constraints, or ideally, utilizing even less memory.

The original plan for the connection between the cellular telephone and register was to have the register initiate a connection to the cellular telephone. Then the cellular telephone would accept the connection and send the data wirelessly to the register. However, due to security concerns and the inability to connect automatically to another Bluetooth device without a known passkey, it became evident that this arrangement would not be sufficient.

The current strategy for the wireless connections is to have software residing on the access points, register, and cellular telephone. These pieces of software, developed on Linux and utilizing OBEX, will coincide with each other and provide automated, seamless connections between the devices. The software that controls the access points and register will be written in Java, and the cellular telephone will use either Java, more specifically J2ME as the programming language. The program flowchart can be viewed in Chart I-A of the appendix.

The register, which consists of a computer running a Linux operating system, will be running a Point of Sale program. A Java program will be integrated into the Point of

Sale software that will control the Bluetooth connection to the cellular telephone. The method of payment (cash, credit card, debit card, or cellular telephone) is controlled by the cashier. The program will also allow the cashier to select the correct telephone for the current customer, if cellular telephone payment is selected. The credit or debit card information will then be downloaded from the cellular device to the register for use. At this point, the customer will have to enter the preset credit card PIN number (which was entered in the cellular telephone program).

The access point will consist of a Compaq Evo N800c laptop running Linux. The laptop will have a Java program residing on it that will push out the coupons to the cellular device utilizing OBEX. The Bluetooth adapters are made by Psion and are pre-release versions. The adapters will be connected to the laptop via the USB port. These access points will initiate and automate the connection to the cellular telephones as well.

Bluetooth operates at the 2.4 GHz frequency. While this should not be an issue, there is always the chance that another device will cause interference with the transaction because many devices (i.e., cordless phones) operate at this frequency. To make sure no erroneous data is received, a checksum will be also be sent by the phone to the register. This checksum will allow the register to check the data it has received against the data the phone sent and ensure that it is the same.

Bluetooth implements a fast frequency-hopping spread spectrum (FHSS), which was designed to prevent data from being stolen while in transmission. However, FHSS has been compromised by hackers, in this case with a BlueSniper Rifle. The BlueSniper Rifle allows a hacker to intercept data traveling over Bluetooth from a distance, even through walls.

Bluetooth, as with all wireless technology, has been determined to be a less than secure medium for data transfer. Since the GenBlu software will be passing credit card information between devices via Bluetooth, security measures will have to be taken. The first means of protecting users' information will be by breaking up credit card and other pertinent information into smaller fragments and then encrypting each of those pieces of data. An initial connection is made between the two devices over a predefined channel. This connection will be strongly encrypted and will determine the channels that data will be sent over and the order in which that data should be recompiled. Other random channels will send garbage data over them so that anyone trying to intercept the data will not know which data is real and which data is random. All channels will of course be very strongly encrypted. The garbage data will not be registered by the point of sale program, so it will not interfere with the transaction at hand.

Financial Budget

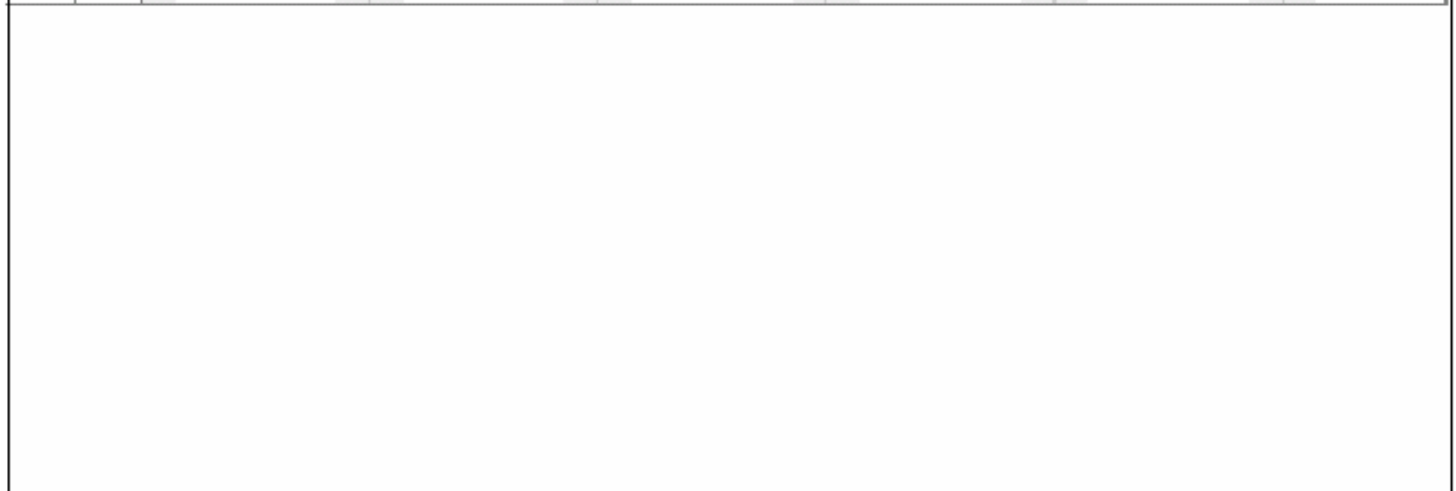
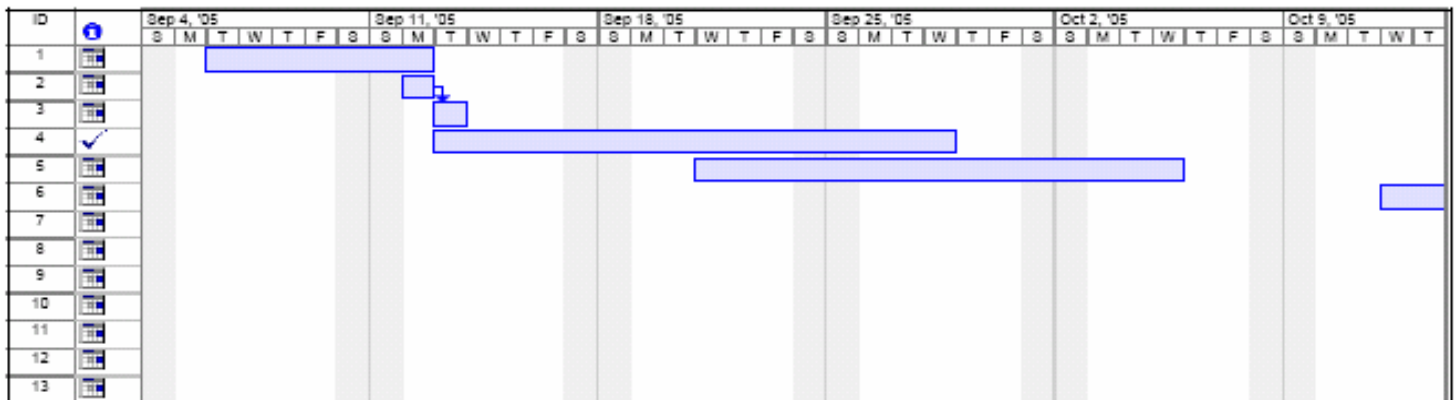
Total Labor Costs

	Labor Charge	Total Cost
Research & Development	\$20/hour	
Fabrication	\$30/hour	
Testing	\$30/hour	
Packaging	\$20/hour	
Technical Installers	\$60/hour	
Management Staff	\$70 and up/hour	
Price for Retail Stores		
-Initial Cost		\$2000
-Monthly Charge		\$200
-Cost / transaction		\$0.50
Cellular Telephone Software	FREE	FREE
Tech Support	\$30/hour	
Various Bills		\$3000
Marketing & Advertising		\$1000

The prices listed above are approximations for future sales. GenBlu will charge an initial fee for parts and set up to each store of \$2000, which will include ten access points with their installation and the installation of the Java applet program and Bluetooth program into the database for access. Each month the store will be charged a fee of \$200 which will include coupon updates and software upgrades as applicable. The store will also be charged \$0.50 for each Bluetooth transaction that is completed at the register. This charge is standard for most POS software. The GenBlu software will be free to all consumers who wish to install it on their cellular telephone. In addition to having the software a customer will also have to have Bluetooth enabled on their cell phone, which most newer phones already have.

Project Schedule

ID	Task Name	Duration	Start	Finish
1	Research Ideas	5 days?	Tue 9/6/05	Mon 9/12/05
2	Define Project	1 day?	Mon 9/12/05	Mon 9/12/05
3	Obtain Advisor	1 day?	Tue 9/13/05	Tue 9/13/05
4	Research and obtain need	12 days?	Tue 9/13/05	Wed 9/28/05
5	Set up / Testing of parts	11 days?	Wed 9/21/05	Wed 10/5/05
6	Project Proposal	5 days?	Wed 10/12/05	Tue 10/18/05
7	Proposal Presentation	1 day?	Tue 12/6/05	Tue 12/6/05
8	Proposal Report	1 day?	Tue 12/6/05	Tue 12/6/05
9	Develop Mobile Software	90 days?	Wed 10/19/05	Tue 2/21/06
10	Develop POS software	90 days?	Wed 10/19/05	Tue 2/21/06
11	Testing	1 day?	Wed 3/8/06	Wed 3/8/06
12	Design Report	5 days?	Tue 5/9/06	Mon 5/15/06
13	Final Presentation	1 day?	Tue 5/9/06	Tue 5/9/06



Project: GenBlu Date: Sun 10/16/05	Task		Rolled Up Task		External Tasks	
	Progress		Rolled Up Milestone		Project Summary	
	Milestone		Rolled Up Progress		Group By Summary	
	Summary		Split		Deadline	
	Page 1					

Conclusion

The GenBlu project is geared towards making a consumer's life easier. Using Bluetooth, a technology available in the majority of new cellular telephones, a consumer will now be able to make purchases without even taking anything out of his or her pockets. The cellular telephone can reside in a customer's purse or pocket, and the transaction can be completed quickly and efficiently without the exchange of cash or the swiping of a credit card. Coupons will no longer need to be clipped, fumbled, and scanned, thus providing a significant decrease in the time a customer waits in line. For the same reasons this technology will make the store clerk's jobs easier and shorten wait time. All of these points, which result from the implementation of GenBlu's package, will provide a more convenient and enjoyable shopping experience.

References

1. Bluetooth Security White Paper.
<http://64.233.179.104/search?q=cache:6HqIN2zxgYAJ:www.bluetooth.com/upload/24Security_Paper.PDF+bluetooth+security&hl=en>
2. BlueZ – Linux Bluetooth Protocol. <<http://www.bluez.org>>
3. Bluetooth and Bluetooth Internet Access Points White Paper.
<<http://www.pico.net/download/whitepaper.pdf>>
4. TuxMobil: Linux and Laptops, PDA's, and Cell Phones.
http://tuxmobil.org/bluetooth_linux.html
5. Open OBEX – Free source implementation of the Object Exchange (OBEX) Protocol
<http://triq.net/obex/>
6. Bluetooth-cracking gun
<http://www.esato.com/news/article.php/id=226>
7. How to: Building a BlueSniper Rifle
<http://www.tomsnetworking.com/Sections-article106.php>

Appendices

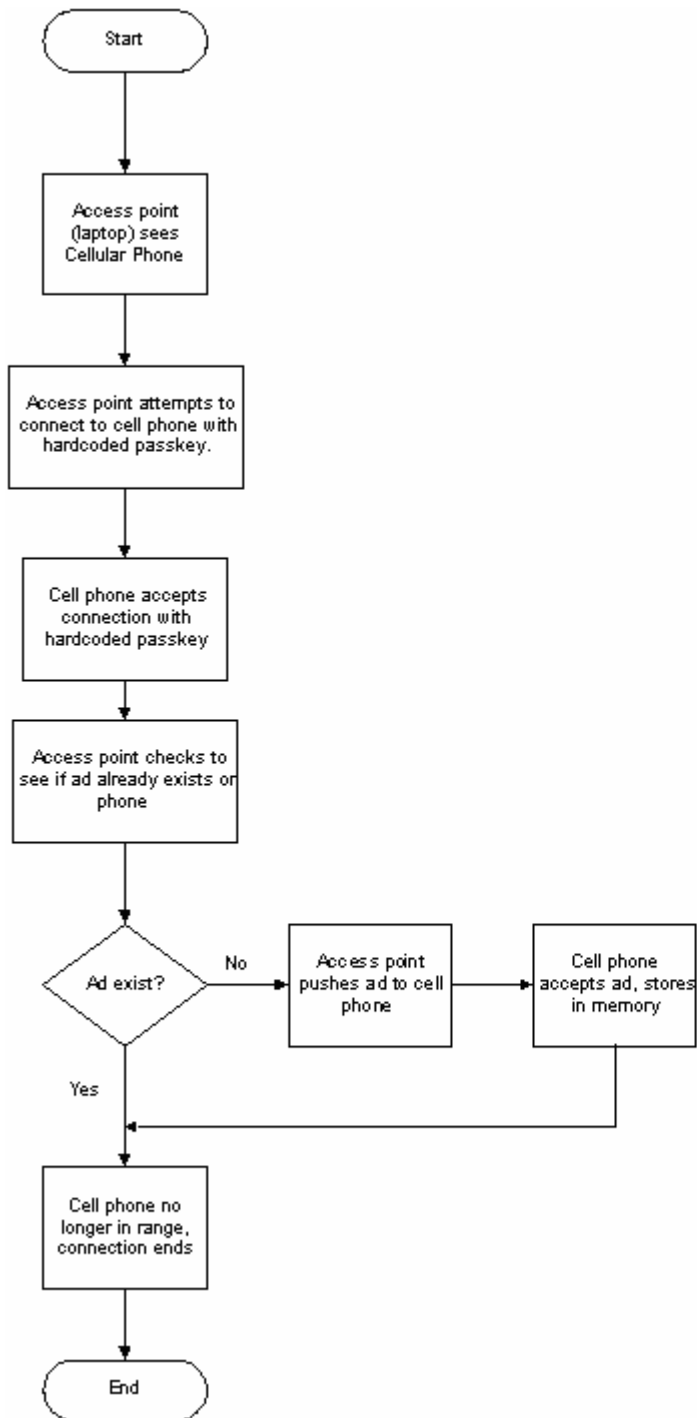


Chart I-A – Bluetooth Connection Flowchart

BluCoupon Server Source Code (In Progress):

UUIDGenerator.java

```

package CouponServer;
import java.util.Random;

public class UUIDGenerator
{
    private static final String hexChars = "0123456789abcdef";
    private static final byte INDEX_TYPE = 6;
    private static final byte INDEX_VARIATION = 8;
    private static final byte TYPE_RANDOM_BASED = 4;

    private Random rnd;

    public UUIDGenerator()
    {
        rnd = new Random(System.currentTimeMillis());
    }

    public String generateUUID()
    {
        // Generate 128-bit random number
        byte[] uuid = new byte[16];
        nextRandomBytes(uuid);

        // Set various bits such as type
        uuid[INDEX_TYPE] &= (byte) 0x0F;
        uuid[INDEX_TYPE] |= (byte) (TYPE_RANDOM_BASED << 4);
        uuid[INDEX_VARIATION] &= (byte) 0x3F;
        uuid[INDEX_VARIATION] |= (byte) 0x80;

        // Convert byte array into a UUID formatted string
        StringBuffer b = new StringBuffer(36);
        for (int i=0; i<16; i++)
        {
            // Mod by renyi
            //if (i==4 || i==6 || i==8 || i==10) b.append('-');
            int hex = uuid[i] & 0xFF;
            b.append(hexChars.charAt(hex >> 4));
            b.append(hexChars.charAt(hex & 0x0F));
        }

        // Return UUID
    }
}

```

```

        return b.toString();
    }

    private void nextRandomBytes(byte[] bytes)
    {
        int numRequested = bytes.length;
        int numGot = 0, rand = 0;
        while (true)
        {
            for (int i=0; i<4; i++)
            {
                if (numGot == numRequested)
                    return;
                rand = (i==0 ? rnd.nextInt() : rand>>8);
                bytes[numGot++] = (byte)rand;
            }
        }
    }
}

```

CouponServer.java

```

package CouponServer;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileOutputStream;

import javax.bluetooth.BluetoothStateException;
import javax.bluetooth.LocalDevice;
import javax.bluetooth.UUID;
import javax.microedition.io.Connector;
import javax.microedition.io.StreamConnection;
import javax.microedition.io.StreamConnectionNotifier;

import CouponServer.fileManager;

public class CouponServer {
    private boolean freeThread = true;

    public static void main(String[] args) {
        printMsg("");
        printMsg("Starting CouponServer v1.0.....");
        writeLog("Starting CouponServer v1.0.....");
        CouponServer cs = new CouponServer();
    }
}

```

```

        System.exit(0);
    }

    public CouponServer(){
        Runnable sthread = new startService();
        Thread thread = new Thread(sthread);

        try {
            for (int i=0; i<5; i++ ) {
                while (!freeThread) { synchronized (this) { wait (1000); } }
                thread.start();
            }
        } catch (Exception e) { printMsg (e.toString()); }
    }

    public class startService implements Runnable {
        public void run() {
            freeThread=false;
            printMsg("Running");
        }
    }

    public void sendList(fileManager f, StreamConnection c){
        String[] tmpFile = f.getFileList();
        String tmpF = tmpFile[0];

        for (int i=1; i<tmpFile.length; i++)
            tmpF = tmpF.concat(",").concat(tmpFile[i]);

        try {
            DataOutputStream out = new DataOutputStream(c.openOutputStream());
            out.writeUTF(tmpF);
            out.flush();
            out.close();
            printMsg("Done.");
        } catch (Exception e) {
            printMsg(e.toString());
        }
    }

    public static void printMsg(String m) {
        System.out.println(m);
    }

    public static void writeLog(String m) {
        try {
            File logFile = new File(System.getProperty("user.dir") + File.separator + "log.txt");
            FileOutputStream fout = new FileOutputStream(logFile,true);

```

```

        DataOutputStream wout = new DataOutputStream(fout);
        wout.writeUTF(m);
        wout.flush();
        wout.close();
        fout.close();
    } catch (Exception e) { System.out.println(e.toString()); }
}
}

```

FileMgr.java

```

package CouponServer;
import java.io.File;

public class FileMgr {

    private String fileList[];
    private String filePath;
    private int fileCount;

    public FileMgr() {
        filePath = System.getProperty("user.dir");
        filePath = filePath.concat(File.separator);
        filePath = filePath.concat("files");

        File temp = new File(filePath);
        fileList = temp.list();
        fileCount = 0;

        for (int i = 0; i < fileList.length; i++){
            temp = new File(filePath + File.separator + fileList[i]);
            if (temp.isFile()) {
                fileList[fileCount] = fileList[i];
                fileCount++;
            }
        }
    }

    public String[] getFileList(){
        return fileList;
    }

    public String getFilePath() {
        return filePath;
    }
}

```

```

    public int getFileCount(){
        return fileCount;
    }
}

```

GetServices.java

```

package CpuonServer;

import javax.bluetooth.DeviceClass;
import javax.bluetooth.DiscoveryListener;
import javax.bluetooth.RemoteDevice;
import javax.bluetooth.UUID;
import javax.bluetooth.DiscoveryAgent;
import javax.bluetooth.LocalDevice;
import javax.bluetooth.ServiceRecord;
import javax.bluetooth.BluetoothStateException;

public class GetServices implements DiscoveryListener {

    private String url = null;
    private int transID;
    private boolean searchCompleted = false;

    public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) { }

    public void inquiryCompleted(int discType) { }

    public void serviceSearchCompleted(int transID, int respCode) {
        searchCompleted = true;
    }

    public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
        url = servRecord[0].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT, false );
    }

    public GetServices (String addr, String uuid, UUID[] uuids) {
        try {
            DiscoveryAgent da = LocalDevice.getLocalDevice().getDiscoveryAgent();

            transID = da.searchServices(null, uuids, new RemoteDevice (addr), this);

            while (!searchCompleted) {

```

```

        synchronized (this) { wait(100); }
    }

    }

    catch (BluetoothStateException be) {}
    catch (InterruptedException ie) {}
}

public String getUrl () {
    return url;
}

public int getTransID () {
    return transID;
}
}

```

PushFile.java

```

package CouponServer;

import java.io.ByteArrayOutputStream;
import java.io.OutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;

import javax.bluetooth.UUID;
import javax.microedition.io.Connector;
import javax.obex.ClientSession;
import javax.obex.HeaderSet;
import javax.obex.Operation;
import javax.obex.ResponseCodes;

import CouponServer.UUIDGenerator;

public class PushFile {
    private UUIDGenerator newUUID = new UUIDGen();

    public PushFile(String f, String c){
        File tempFile = null;
        tempFile = new File(System.getProperty("user.dir") + File.separator + "files" +
File.separator + f);
        if (!tempFile.exists()) {                return;                }
        byte[] file2send = null;
        byte[] buff = new byte[1024];
    }
}

```

```

ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);

try {
    FileInputStream fin = new FileInputStream(tempFile);
    while (true) {
        int length = fin.read(buff);
        if (length == -1) break;
        baos.write(buff,0,length);
    }
    file2send = baos.toByteArray();
} catch (IOException e) {}

UUID[] uuids = { new UUID(0x1106) };
GetServices gs = new findServices(c,newUUID.generateUUID(),uuids);

try {
    ClientSession session = (ClientSession) Connector.open(gs.getUrl());
    HeaderSet head = session.connect(null);

    int responseCode = head.getResponseCode();
    if (responseCode != ResponseCodes.OBEX_HTTP_OK) {
        throw new IOException();
    }

    HeaderSet headers = session.createHeaderSet();
    headers.setHeader(HeaderSet.LENGTH, new Long(file2send.length));
    headers.setHeader(HeaderSet.NAME,f);
    headers.setHeader(HeaderSet.TYPE,"binary");
    Operation operation = session.put(headers);

    OutputStream out = operation.openOutputStream();
    out.write(file2send);
    out.flush();

    session.disconnect(null);
    session.close();
    out.close();
    operation.close();

    System.out.println("Done.");
} catch (Exception e) {
    System.out.println(e.toString());
}
}
}

```