1. Choose a design layout. Customize it to fit your needs.
2. Add your desired features. **Create an app** that reflects the right image for your brand.
3. Publish your **app**. Push it live on Android or iPhone **app** stores on-the-fly.

**CREATING AN RFID TAG READER USING:**

/\* Sample java code to read an RFID tag \*/

import java.io.\*;

import java.util.\*;

import javax.comm.\*;

public class RFIDTagRead implements Runnable, SerialPortEventListener {

static CommPortIdentifier portId;

static Enumeration portList;

InputStream inputStream;

SerialPort serialPort;

Thread readThread;

//Array consisting of SOH, length, command, data, BCC

static byte[] bytearray = {0x01, 0x02, 0x09, 0x32, 0x39};

static OutputStream outputStream;

static int n =0;

public static void main(String[] args) {

//Enumerate a list of available ports

portList = CommPortIdentifier.getPortIdentifiers();

// Identify the ports. I connected the reader with COM1

while (portList.hasMoreElements()) {

portId = (CommPortIdentifier) portList.nextElement();

if (portId.getPortType() == CommPortIdentifier.PORT\_SERIAL) {

if (portId.getName().equals("COM1")) {

System.out.println("The port is: " + portId.getName());

RFIDTagRead reader = new RFIDTagRead();

}

}

}

}

public RFIDTagRead() {

try {

//Open the COM1 port and name it MicroReader with timeout 2000ms

serialPort = (SerialPort) portId.open("SimpleReadApp", 2000);

} catch (Exception e) {System.out.println("Port Error");}

try {

outputStream = serialPort.getOutputStream();

// Write the stream of data conforming to PC to reader protocol

outputStream.write(bytearray);

outputStream.flush();

System.out.println("The following bytes are being written");

for(int i=0; i<bytearray.length; i++)

System.out.println(bytearray[i]);

System.out.println("Tag will be read when its in the

field of the reader");

} catch (IOException e) {}

// Set Serial Port parameter

try {

serialPort.setSerialPortParams(9600,

SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1,

SerialPort.PARITY\_NONE);

} catch (UnsupportedCommOperationException e) {}

try {

//Register an event listener object to the port

serialPort.addEventListener(this);

} catch (TooManyListenersException e)

{System.out.println("Too Many Listeners");

}

//Specify an event type. On data availability, triggers serialEvent method

serialPort.notifyOnDataAvailable(true);

try {

//Associate an InputStream object with this port.

inputStream = serialPort.getInputStream();

} catch (IOException e) {}

//Start a thread to handle the time-to-read the tag

readThread = new Thread(this);

readThread.start();

}

public void run() {

try {

Thread.sleep(56);

} catch (InterruptedException e) {}

}

//This method is called by notifyOnDataAvailabe()

public void serialEvent(SerialPortEvent event) {

switch(event.getEventType()) {

case SerialPortEvent.BI:

case SerialPortEvent.OE:

case SerialPortEvent.FE:

case SerialPortEvent.PE:

case SerialPortEvent.CD:

case SerialPortEvent.CTS:

case SerialPortEvent.DSR:

case SerialPortEvent.RI:

case SerialPortEvent.OUTPUT\_BUFFER\_EMPTY:

break;

case SerialPortEvent.DATA\_AVAILABLE:

n++; //to count the number of readings

System.out.println("The reading description of RFID Tag" + " " + n);

//array size must not be less than the number of bytes to be read

byte[] readBuffer = new byte[20]; // to store the read data

int numbyte = 0;

try {

while(inputStream.available() >0) {

// Read the RFID data and store in the byte array

numbyte = inputStream.read(readBuffer);

System.out.println("Number of Bytes read: " + numbyte);

}

} catch (IOException e) {}

if( readBuffer[0] == 1) /\*check if start bit is detected \*/

{

int length = readBuffer[1];

// Identify the Transponder type

switch(readBuffer[2]) {

case 12 :

{

System.out.print("RFID is RO:" + "\t");

break;

}

case 13 :

{

System.out.print("RFID is R/W:" + "\t");

break;

}

case 14:

{

System.out.print("RFID is MPT/SAMPT:" + "\t");

break;

}

case 15:

{

System.out.print("RFID is Other:" + "\t");

break;

}

}

// Write the actual tag reading in Hexadecimal

for( int m = length+1; m > 2; m--)

System.out.print(Integer.toHexString(readBuffer[m] & 255));

}

System.out.println(" ");

System.out.println("\t" + "Read Sucessful");

System.out.println("----------------------------------");

break;

}

}

}