Topic : PC Remote (via using phone)

INDEX

Certificate Of Acknowledgment

This is to certify that ,on this day of 25 july,2014 , Sumit Bansal has successfully completed their project on topic “Pc Remote (via using phone)”.
Languages use : java and android.

Acknowledgment

We take this opportunity to express our profound gratitude and deep regards to our guide and teacher ER. SURENDRA CHOUDHARY for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her has played a major role in making our project a success.

We also take this opportunity to express a deep sense of gratitude to Company Mentors for their cordial support, valuable information and guidance, which helped me in completing this task through various stages.

PC Remote (via android phone) :

There is a two languages code , one for android phone(android language) and one for laptop or pc (java language) . Both are connected through wifi .

By phone we control the pc for example - shutdown , restart , lock screen and display any message on pc screen. This app is short version of pc cmd (command prompt).

App Icon :

CODE:

Pc code :-

```
/*
Topic :  pc controller via wifi (Server code)
Language:  java
Author:  Sumit Bansal
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;
import java.io.File;
import javax.swing.JOptionPane;

public class server {

        private static ServerSocket serverSocket;
```

```
private static Socket clientSocket;
private static InputStreamReader inputStreamReader;
private static BufferedReader bufferedReader;
private static String message,db;

public static void main(String[] args) {
	try {
		serverSocket = new ServerSocket(4444); // Server socket

	} catch (IOException e) {
	//	System.out.println("Could not listen on port: 4444");
	}

	//System.out.println("Server started. Listening to the port 4444");

	while (true) {
		try {

			clientSocket = serverSocket.accept(); // accept the client connection
		inputStreamReader = new InputStreamReader(clientSocket.getInputStream());
	bufferedReader = new BufferedReader(inputStreamReader); // get the client message
				message = bufferedReader.readLine();
				PrintWriter writer = new PrintWriter("test.bat", "UTF-8");
				//System.out.println(message);
				if(message==null)
				message="cls";
				else if(message.substring(0, 1).equals("?"))
```

```
                {
                db =message.substring(1);
                JOptionPane.showMessageDialog(null,db);
                message = "cls";
                }
                else if(message.equals("up"))
                {
                writer.println("/*");
                writer.println("cscript /e:jscript \"%~f0\" */");
        message = "var shl = new ActiveXObject(\"WScript.Shell\");for (var i=0; i<2; i++)
{shl.SendKeys(String.fromCharCode(0xAF));}";
                }
                else if(message.equals("down"))
                {
                writer.println("/*");
                writer.println("cscript /e:jscript \"%~f0\" */");
        message = "var shl = new ActiveXObject(\"WScript.Shell\");for (var i=0; i<2; i++)
        {shl.SendKeys(String.fromCharCode(0xAE));}";
                }

                writer.println(message);
                writer.close();
                try{
                        Process p = Runtime.getRuntime()
                                .exec("rundll32 url.dll,FileProtocolHandler test.bat");
                        p.waitFor();
                        } catch(Exception e){
        //                System.out.println("operation is failed.");
```

```
                }


            try {
                Thread.sleep(1000);
                } catch(InterruptedException ex) {
                    Thread.currentThread().interrupt();
                }



            try{
                File file = new File("test.bat");
                if(file.delete()){
            //          System.out.println(file.getName() + " is deleted!");
                }else{
            //          System.out.println("Delete operation is failed.");
                    }
                }catch(Exception e){
                        e.printStackTrace();
                }


            inputStreamReader.close();
            clientSocket.close();


        } catch (IOException ex) {
        //      System.out.println("Problem in message reading");
        }

}}}
```

Android code:

MainActivity.java

```
package com.example.pcremote;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {

        Button bconnect;

        EditText textField;

        public static String ip ;

public void onCreate(Bundle savedInstanceState) {

         super.onCreate(savedInstanceState);

         setContentView(R.layout.activity_main);

         textField = (EditText) findViewById(R.id.editText1);

         bconnect = (Button) findViewById(R.id.button1);

         bconnect.setOnClickListener(this);

        }

        public void onClick(View v) {

                ip = textField.getText().toString();

                Intent intent=new Intent(this,Client.class);

                startActivity(intent);

                }       }
```

Client.java

```
package com.example.pcremote;


import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

import java.net.UnknownHostException;




import android.os.AsyncTask;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;



public class Client extends Activity implements OnClickListener {


        private Socket client;

        private PrintWriter printwriter;

        private String message,str;

        EditText command,msg;

        Button b1,b2,exe,up,down,b6,b7,b8,b9;
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_client);
 command = (EditText) findViewById(R.id.editText1);
 msg = (EditText) findViewById(R.id.editText2);
 b1 = (Button) findViewById(R.id.button1);
 b2 = (Button) findViewById(R.id.button2);
 b6 = (Button) findViewById(R.id.button6);
 b7 = (Button) findViewById(R.id.button7);
 b8 = (Button) findViewById(R.id.button8);
 b9 = (Button) findViewById(R.id.button9);
 up = (Button) findViewById(R.id.button4);
 down = (Button) findViewById(R.id.button5);
 exe = (Button) findViewById(R.id.button3);
 b1.setOnClickListener(this);
 b2.setOnClickListener(this);
 b6.setOnClickListener(this);
 b7.setOnClickListener(this);
 b8.setOnClickListener(this);
 b9.setOnClickListener(this);
 exe.setOnClickListener(this);
 up.setOnClickListener(this);
 down.setOnClickListener(this);
 MainActivity obj = new MainActivity();
 str = obj.ip;
}
```

```
public void onClick(View v) {


            switch (v.getId()) {

            case R.id.button1:

            message = "shutdown.exe /s /t 00";

             break;

            case R.id.button2:

   message = "shutdown.exe /r /t 00";

             break;

            case R.id.button4:

             message = "up";

             break;

            case R.id.button5:

             message = "down";

             break;

            case R.id.button6:

                        message = "Rundll32.exe User32.dll,LockWorkStation";

                        break;

            case R.id.button7:

                        message = "rundll32.exe powrprof.dll,SetSuspendState 0,1,0";

                        break;

            case R.id.button8:

                        message = "rundll32.exe PowrProf.dll,SetSuspendState";

                        break;

            case R.id.button3:

                        message = command.getText().toString();
```

```
                break;
            case R.id.button9:
                message ="?"+msg.getText().toString();
                break;

            }

  SendMessage sendMessageTask = new SendMessage();
  sendMessageTask.execute();
}


        private class SendMessage extends AsyncTask<Void, Void, Void> {

            @Override
            protected Void doInBackground(Void... params) {
                try {
                    //String str = new String("116.202.91.218");
                    client = new Socket(str, 4444); // connect to the server

                    printwriter = new PrintWriter(client.getOutputStream(), true);
                    printwriter.write(message); // write the message to output stream
                    printwriter.flush();
                    printwriter.close();
                    client.close(); // closing the connection

                } catch (UnknownHostException e) {
```

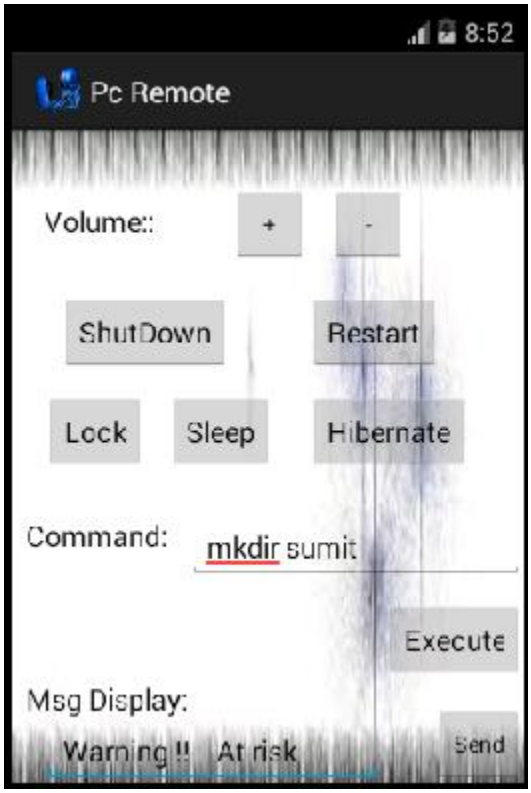
```
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                return null;
            }

        }

}
```

SnapShots:


8:50
Pc Remote
Connect


8:52
Pc Remote
Volume::
+
-
ShutDown
Restart
Lock
Sleep
Hibernate
Command:
mkdir sumit
Execute
Msg Display:
Warning !! At risk
Send

BIBLIOGRAPHY

1. Java Doc file
2. Wikipedia
3. NIIT material
4. Android development and oracle websites.