**APPENDIX 1-SAMPLE CODING**

**//NEW WEB SERVICE**

**@WebService(serviceName = "NewWebService")**

**public class NewWebService {**

**/\*\***

**\* Web service operation**

**\*/**

**@WebMethod(operationName = "signin")**

**public String signin(@WebParam(name = "username") String username, @WebParam(name = "password") String password) {**

**try {**

**Common\_DB cd=new Common\_DB();**

**ResultSetrs=Common\_DB.LoginCheck("psjav05", "login","username","password", username, password);**

**if(rs.next()) {**

**return "success";**

**}**

**else {**

**return "username or password is invalid"; }**

**} catch (Exception ex) {**

**Logger.getLogger(NewWebService.class.getName()).log(Level.SEVERE, null, ex);**

**return "server temporarily not available"; }**

**}**

**/\*\*Web service operation**

**@WebMethod(operationName = "signup")**

**public String signup(@WebParam(name = "username") String username, @WebParam(name = "password") String password, @WebParam(name = "email") String email) { try {**

**Common\_DB cd=new Common\_DB();**

**intrs=Common\_DB.InsertTable("psjav05", "INSERT INTO login(username,password,email) VALUES('"+username+"','"+password+"','"+email+"')");**

**if(rs>0) {**

**return "success"; }**

**else {**

**return "username is already available"; }**

**} catch (Exception ex) {**

**Logger.getLogger(NewWebService.class.getName()).log(Level.SEVERE, null, ex);**

**return "server temporarily not available"; } }**

**//TRANSCODER**

**public static void main(String a[]) { try {**

**File source=new File("E:/Wildlife.wmv");**

**File target=new File("E:/my.mp4");**

**AudioAttributesaattrib=new AudioAttributes();**

**aattrib.setCodec("libfaac");**

**aattrib.setBitRate(new Integer(256000));**

**aattrib.setSamplingRate(new Integer(65100));**

**aattrib.setChannels(new Integer(2));**

**VideoAttributesvattrib=new VideoAttributes();**

**vattrib.setCodec("mpeg4");**

**vattrib.setBitRate(new Integer(3200000));**

**vattrib.setFrameRate(new Integer(18));**

**vattrib.setSize(new VideoSize(620, 480));**

**EncodingAttributesattrs = new EncodingAttributes();**

**attrs.setFormat("mp4");**

**attrs.setAudioAttributes(aattrib);**

**attrs.setVideoAttributes(vattrib);**

**Encoder encoder = new Encoder();**

**encoder.encode(source, target, attrs);**

**} catch (IllegalArgumentException ex) {**

**Logger.getLogger(TransCoder.class.getName()).log(Level.SEVERE, null, ex);**

**} catch (InputFormatException ex) {**

**Logger.getLogger(TransCoder.class.getName()).log(Level.SEVERE, null, ex);**

**} catch (EncoderException ex) {**

**Logger.getLogger(TransCoder.class.getName()).log(Level.SEVERE, null, ex);**

**}**

**}**

**}**

**/CALL SERVICES**

**public static String getVideosFromMobileInfo(String bandwidth,Stringphonetype,Stringnetworkname,Stringsimstate,Stringnetworktype,Stringosversion,Stringusername,String method) {**

**String list = null;**

**SoapObject soap=new SoapObject(NAMESPACE,method);**

**soap.addProperty("bandwidth",bandwidth);**

**soap.addProperty("networkname",networkname);**

**soap.addProperty("phonetype",phonetype);**

**soap.addProperty("simstate",simstate);**

**soap.addProperty("networktype",networktype);**

**soap.addProperty("osversion",osversion);**

**soap.addProperty("username",username);**

**SoapSerializationEnvelope envelope=new SoapSerializationEnvelope(SoapEnvelope.VER11);**

**envelope.setOutputSoapObject(soap);**

**HttpTransportSE http=new HttpTransportSE(URL);**

**try {**

**http.call(NAMESPACE+method, envelope);**

**SoapPrimitivepri=(SoapPrimitive) envelope.getResponse();**

**list=pri.toString();**

**} catch (IOException e) {**

**e.printStackTrace();**

**return list;**

**} catch (XmlPullParserException e) {**

**e.printStackTrace();**

**return list;**

**} return list; }**

**//MAIN ACTIVITY**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**pb=(ProgressBar)findViewById(R.id.progressBar2);**

**bn=(Button)findViewById(R.id.bt);**

**bw=(TextView)findViewById(R.id.bandwidth);**

**pt=(TextView)findViewById(R.id.phone\_type);**

**nt=(TextView)findViewById(R.id.network\_type);**

**ss=(TextView)findViewById(R.id.sim\_state);**

**nn=(TextView)findViewById(R.id.network\_name);**

**uname=(TextView)findViewById(R.id.uname);**

**username=getIntent().getStringExtra("username");**

**uname.setText("welcome "+username+",");**

**icon=R.drawable.ic\_launcher;**

**bn.setOnClickListener(new OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**// Get the telephony system service to find out network details**

**finalTelephonyManager tm = (TelephonyManager) getSystemService(TELEPHONY\_SERVICE);**

**// Update text views with readable values.**

**updateViews(tm);**

**// Since these attributes can change, we will register a**

**// {@code PhoneStateListener} to listen for these changes and**

**// update the view.**

**tm.listen(new PhoneStateListener() {**

**@Override**

**public void onServiceStateChanged(ServiceStateserviceState) {**

**// Update our TextViews**

**updateViews(tm);**

**}**

**@Override**

**public void onDataConnectionStateChanged(int state) {**

**// A change in data connection state may be due to**

**// of a different network type**

**updateViews(tm);**

**}**

**},**

**PhoneStateListener.LISTEN\_SERVICE\_STATE | PhoneStateListener.LISTEN\_DATA\_CONNECTION\_STATE);**

**// calculate bandwidth**

**Downloadfiledf=new Downloadfile();**

**df.execute } ); }**

**// @Override**

**// protected void onResume() {**

**// super.onResume();**

**// final TelephonyManager tm = (TelephonyManager) getSystemService(TELEPHONY\_SERVICE);**

**//**

**// // Update text views with readable values.**

**// updateViews(tm);**

**// }**

**@Override**

**publicbooleanonCreateOptionsMenu(Menu menu) {**

**// Inflate the menu; this adds items to the action bar if it is present.**

**getMenuInflater().inflate(R.menu.main, menu);**

**return true;**

**}**

**/\*\***

**\* Update text views with telephony attributes.**

**\*/private final void updateViews(TelephonyManager tm) {**

**// The telephony system service returns integer constants for various**

**// telephony attributes.**

**simstate=tm.getSimSerialNumber();**

**phonetype=mapDeviceTypeToName(tm.getPhoneType());**

**networkname=tm.getNetworkOperatorName();**

**networktype= mapNetworkTypeToName(tm.getNetworkType());**

**osversion=Build.VERSION.RELEASE;**

**ss.setText("SIM State: " + simstate);**

**nt.setText("Network Type: " +networktype);**

**pt.setText("Phone Type: " + phonetype);**

**nn.setText("Network Operator: " +networkname);**

**}**