**CODE SNIPPETS**

### SMS RECIEVER

**package** com.project.project;

**import** android.content.BroadcastReceiver;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.content.SharedPreferences;

**import** android.location.Criteria;

**import** android.location.Location;

**import** android.location.LocationListener; **import** android.location.LocationManager; **import** android.media.MediaPlayer;

**import** android.media.MediaRecorder;

**import** android.os.Bundle;

**import** android.preference.PreferenceManager;

**import** android.telephony.SmsManager; **import** android.telephony.TelephonyManager; **import** android.telephony.gsm.SmsMessage; **import** android.widget.Toast;

**import** java.io.File;

**import** java.io.IOException;

**public class** smsReceiver **extends** BroadcastReceiver { String incomingno;

**static** LocationManager locationManager;

**static** MyLocationListener locationListener; Context cntxt;

**static** LocationManager locationMgr; SharedPreferences sp;

SmsManager smanager;

**static** MyTrackListener MTListener;

**private static final** String AUDIO\_RECORDER\_FILE\_EXT\_3GP = **".3gp"**; **private static final** String AUDIO\_RECORDER\_FILE\_EXT\_MP4 = **".mp4"**; **private static final** String AUDIO\_RECORDER\_FOLDER = **"AudioRecorder"**; **private** MediaRecorder recorder = **null**;

**private int** currentFormat = 0;

**private int** output\_formats[] = {MediaRecorder.OutputFormat.MPEG\_4, MediaRecorder.OutputFormat.THREE\_GPP};

**private** String file\_exts[] = {AUDIO\_RECORDER\_FILE\_EXT\_MP4, AUDIO\_RECORDER\_FILE\_EXT\_3GP};

**public static** String BalanceWallet = **"200"**; **public static int** counter = 0;

@Override

**public void** onReceive(Context context, Intent intent) {

*// TODO Auto-generated method stub*

cntxt = context;

sp = PreferenceManager.getDefaultSharedPreferences(context); smanager = SmsManager.getDefault();

Bundle bundle = intent.getExtras(); SmsMessage[] msgs = **null**;

String str = **""**;

**if** (body.equalsIgnoreCase(**"loc"**)) {

**final** Criteria criteria = **new** Criteria(); locationManager = (LocationManager)

context.getSystemService(Context.LOCATION\_SERVICE);

**final** String bestProvider = locationManager.getBestProvider(criteria, **true**);

Toast.makeText(context, **""** + bestProvider, 600000).show(); locationListener = **new** MyLocationListener(); locationManager.requestLocationUpdates(

bestProvider, 0,

0,

locationListener);

}

**if** (body.equalsIgnoreCase(**"photo"**)) { Intent in = **new** Intent(context,

OneShotPreviewActivity.**class**);

in.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK); context.startActivity(in);

}

**if** (body.equalsIgnoreCase(**"track"**)) {

**final** Criteria criteria = **new** Criteria(); locationMgr = (LocationManager)

context.getSystemService(Context.LOCATION\_SERVICE);

**final** String bestProvider = locationMgr.getBestProvider(criteria, **true**);

Toast.makeText(cntxt, **""** + bestProvider, 600000).show();

MTListener = **new** MyTrackListener(); locationMgr.requestLocationUpdates(

bestProvider, 10000,

0,

MTListener);

}

*//*

**public void** stopListening(LocationManager locationManager) {

**try** {

**if** (locationManager != **null** && locationListener != **null**) { locationManager.removeUpdates(locationListener); locationManager = **null**;

locationListener = **null**;

}

} **catch** (**final** Exception ex) {

}

}

**private class** MyLocationListener **implements** LocationListener {

**public void** onLocationChanged(Location location) { String message = String

.format(**"12345 New Location of your friend \n Longitude:**

**%1$s \n Latitude: %2$s"**,

location.getLongitude(), location.getLatitude()); Toast.makeText(cntxt, **"got loc"**, 6000).show(); stopListening(locationManager);

smanager.sendTextMessage(

incomingno,

**null**,

**"loc1234 "** + location.getLatitude() + **","**

+ location.getLongitude(), **null**, **null**); String PoliceNo = sp.getString(**"Police No"**, **null**);

*/\*smanager.sendTextMessage( PoliceNo,*

*null,*

*"loc1234 " + location.getLatitude() + ","*

*+ location.getLongitude(), null, null);\*/*

}

**public void** onStatusChanged(String s, **int** i, Bundle b) {

}

**public void** onProviderDisabled(String s) {

}

**public void** onProviderEnabled(String s) {

}

}

**private class** MyTrackListener **implements** LocationListener {

**public void** onLocationChanged(Location location) { String message = String

.format(**"New Location of your friend \n Longitude: %1$s \n**

**Latitude: %2$s"**,

location.getLongitude(), location.getLatitude());

*// stopListening(locationManager);*

smanager.sendTextMessage(

incomingno,

**null**,

**"loc1234 "** + location.getLatitude() + **","**

+ location.getLongitude(), **null**, **null**);

}

**public void** onStatusChanged(String s, **int** i, Bundle b) {

}

**public void** onProviderDisabled(String s) {

}

**public void** onProviderEnabled(String s) {

}

#### PHOTO CAPTURE ACTIVITY

**package** com.project.project;

**import** android.app.Activity;

**import** android.content.Context;

**import** android.content.SharedPreferences; **import** android.graphics.ImageFormat; **import** android.graphics.Rect;

**import** android.graphics.YuvImage; **import** android.hardware.Camera; **import** android.hardware.Camera.Size; **import** android.os.Bundle;

**import** android.preference.PreferenceManager;

**import** android.util.Log;

**import** android.view.MotionEvent; **import** android.view.Surface; **import** android.view.SurfaceHolder; **import** android.view.SurfaceView; **import** android.view.Window; **import** android.view.WindowManager;

**import** java.io.ByteArrayOutputStream; **import** java.io.FileNotFoundException; **import** java.io.FileOutputStream; **import** java.io.IOException;

**import** java.util.List;

**public class** CapturePhotoActivity **extends** Activity {

*/\*\* Called when the activity is first created. \*/*

**private** Camera \_camera; **private** Activity \_activity; **byte**[] mydata;

**int** mPreviewImageFormat = ImageFormat.NV21; **static** CapturePhotoActivity os; SharedPreferences sp;

@Override

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

\_activity = **this**; **super**.onCreate(savedInstanceState); os = **this**;

sp = PreferenceManager.getDefaultSharedPreferences(**this**); getWindow().addFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN); requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(**new** CameraPreview(**this**));

}

**private** Camera.PreviewCallback mPreviewListener = **new** Camera.PreviewCallback()

{

@Override

**public void** onPreviewFrame(**byte**[] data, Camera camera) {

Camera.Parameters parameters = \_camera.getParameters(); Size size = parameters.getPreviewSize();

YuvImage image = **new** YuvImage(data, parameters.getPreviewFormat(), size.width, size.height, **null**);

ByteArrayOutputStream bos = **new** ByteArrayOutputStream();

image.compressToJpeg(

**new** Rect(0, 0, image.getWidth(), image.getHeight()), 90, bos);

**byte**[] myimage = bos.toByteArray(); FileOutputStream outStream = **null**;

**try** {

System.out.println(**"the data is to be "** + data); FileOutputStream outStream1 = **new** FileOutputStream(

**"/sdcard/daya.jpg"**);*// String.format("/sdcard/daya.jpg",*

*// System.currentTimeMillis()));* outStream1.write(myimage); outStream1.close();

Log.d(**"Camera"**, **"onPictureTaken - wrote bytes: "** + data.length);

} **catch** (FileNotFoundException e) { e.printStackTrace();

} **catch** (IOException e) { e.printStackTrace();

} **finally** {

}

finish();

*// takepicture();*

}

};

**public void** takepicture() {

\_camera.setOneShotPreviewCallback(mPreviewListener);

}

@Override

**public boolean** onTouchEvent(MotionEvent event) {

**if** (event.getAction() == MotionEvent.ACTION\_DOWN) {

\_camera.setOneShotPreviewCallback(mPreviewListener);

}

**return super**.onTouchEvent(event);

}

**public void** setCameraDisplayOrientation(Camera camera,

**int** cameraId) {

Camera.CameraInfo info = **new** Camera.CameraInfo(); Camera.getCameraInfo(cameraId, info);

**int** rotation = \_activity.getWindowManager().getDefaultDisplay()

.getRotation(); **int** degrees = 0; **switch** (rotation) {

**case** Surface.ROTATION\_0: degrees = 0;

**break**;

**case** Surface.ROTATION\_90: degrees = 90;

**break**;

**case** Surface.ROTATION\_180:

degrees = 180;

**break**;

**case** Surface.ROTATION\_270:

degrees = 270;

**break**;

}

**int** result;

**if** (info.facing == Camera.CameraInfo.CAMERA\_FACING\_FRONT) { result = (info.orientation + degrees) % 360;

result = (360 - result) % 360; *// compensate the mirror*

} **else** { *// back-facing*

result = (info.orientation - degrees + 360) % 360;

}

Log.d(**"camera"**, **"result= "** + result); camera.setDisplayOrientation(result);

}

**private** Camera openFrontFacingCameraGingerbread() {

**int** cameraCount = 0; Camera cam = **null**;

Camera.CameraInfo cameraInfo = **new** Camera.CameraInfo(); cameraCount = Camera.getNumberOfCameras();

**for** (**int** camIdx = 0; camIdx < cameraCount; camIdx++) { Camera.getCameraInfo(camIdx, cameraInfo);

**if** (cameraInfo.facing == Camera.CameraInfo.CAMERA\_FACING\_BACK) {

**try** {

cam = Camera.open(camIdx);

} **catch** (RuntimeException e) { Log.e(**"camera"**,

**"Camera failed to open: "** + e.getLocalizedMessage());

}

}

}

**return** cam;

}

**public class** CameraPreview **extends** SurfaceView **implements**

SurfaceHolder.Callback {

**private** SurfaceHolder holder; CameraPreview(Context context) {

**super**(context); holder = getHolder();

holder.addCallback(**this**); holder.setType(SurfaceHolder.SURFACE\_TYPE\_PUSH\_BUFFERS);

}

**public void** surfaceChanged(SurfaceHolder holder, **int** format, **int** width,

**int** height) {

configure(format, width, height);

*/\*try { Thread.sleep(1000);*

*\_camera.setOneShotPreviewCallback(mPreviewListener);*

*} catch (InterruptedException e) {*

*// TODO Auto-generated catch block e.printStackTrace();*

*}\*/*

}

**public void** surfaceDestroyed(SurfaceHolder holder) {

**if** (\_camera != **null**) {

\_camera.stopPreview();

\_camera.release();

\_camera = **null**;

}

}

**public void** surfaceCreated(SurfaceHolder holder) {

*// \_camera = Camera.open();*

\_camera = openFrontFacingCameraGingerbread();

**try** {

\_camera.setPreviewDisplay(holder);

} **catch** (IOException e1) {

*// TODO Auto-generated catch block*

e1.printStackTrace();

}

setCameraDisplayOrientation(\_camera, 0);

\_camera.startPreview();

}

**protected void** setPictureFormat(**int** format) { Camera.Parameters params = \_camera.getParameters();

List<Integer> supported = params.getSupportedPictureFormats();

**if** (supported != **null**) {

**for** (**int** f : supported) {

**if** (f == format) { params.setPreviewFormat(format);

\_camera.setParameters(params);

**break**;

}

}

}

}

**protected void** setPreviewSize(**int** width, **int** height) { Camera.Parameters params = \_camera.getParameters(); List<Size> supported = params.getSupportedPreviewSizes(); **if** (supported != **null**) {

**for** (Size size : supported) {

**if** (size.width <= width && size.height <= height) { params.setPreviewSize(size.width, size.height);

\_camera.setParameters(params);

**break**;

}

}

}

}

**public void** configure(**int** format, **int** width, **int** height) {

\_camera.stopPreview(); setPictureFormat(format); setPreviewSize(width, height);

\_camera.startPreview();

}

**}**

#### ONE SHOT PREVIEW ACTIVITY

**package** com.project.project;

**import** java.io.ByteArrayOutputStream; **import** java.io.FileNotFoundException; **import** java.io.FileOutputStream; **import** java.io.IOException;

**import** java.util.List;

**import** android.app.Activity;

**import** android.content.Context;

**import** android.content.SharedPreferences; **import** android.graphics.ImageFormat; **import** android.graphics.Rect;

**import** android.graphics.YuvImage; **import** android.hardware.Camera; **import** android.hardware.Camera.Size; **import** android.os.Bundle;

**import** android.preference.PreferenceManager;

**import** android.util.Log;

**import** android.view.MotionEvent; **import** android.view.Surface; **import** android.view.SurfaceHolder; **import** android.view.SurfaceView; **import** android.view.Window;

**import** android.view.WindowManager;

**import** android.widget.Toast;

**public class** OneShotPreviewActivity **extends** Activity {

*/\*\* Called when the activity is first created. \*/*

**private** Camera \_camera; **private** Activity \_activity; **byte**[] mydata;

**int** mPreviewImageFormat = ImageFormat.NV21; **static** OneShotPreviewActivity os; SharedPreferences sp;

@Override

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

\_activity = **this**; **super**.onCreate(savedInstanceState);

os = **this**;

sp = PreferenceManager.getDefaultSharedPreferences(**this**); getWindow().addFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN); requestWindowFeature(Window.FEATURE\_NO\_TITLE); setContentView(**new** CameraPreview(**this**));

}

**private** Camera.PreviewCallback mPreviewListener = **new** Camera.PreviewCallback()

{

@Override

**public void** onPreviewFrame(**byte**[] data, Camera camera) {

Camera.Parameters parameters = \_camera.getParameters(); Size size = parameters.getPreviewSize();

YuvImage image = **new** YuvImage(data, parameters.getPreviewFormat(), size.width, size.height, **null**);

ByteArrayOutputStream bos = **new** ByteArrayOutputStream();

image.compressToJpeg(

**new** Rect(0, 0, image.getWidth(), image.getHeight()), 90, bos);

**byte**[] myimage = bos.toByteArray(); FileOutputStream outStream = **null**;

**try** {

System.out.println(**"the data is to be "** + data); FileOutputStream outStream1 = **new** FileOutputStream(

**"/sdcard/daya.jpg"**);*// String.format("/sdcard/daya.jpg",*

*// System.currentTimeMillis()));* outStream1.write(myimage); outStream1.close();

Log.d(**"Camera"**, **"onPictureTaken - wrote bytes: "** + data.length);

} **catch** (FileNotFoundException e) { e.printStackTrace();

} **catch** (IOException e) { e.printStackTrace();

} **finally** {

}

**new** Thread() { @Override

**public void** run() {

*// new SendFromYahoo().SendMail();*

String email = sp.getString(**"emailid"**,

[**"dayanand.shine@gmail.com"**](mailto:dayanand.shine@gmail.com)); **try** { *// GMailSender sender =*

*// new* [*GMailSender("abhignadoijode@gmail.com",*](mailto:abhignadoijode@gmail.com) *"@bh!\_doll");*

GMailSender sender = **new** GMailSender(

[**"dayadad4390@gmail.com"**](mailto:dayadad4390@gmail.com), **"dayadad17daya"**);

sender.sendMail(**"photo"**,

**"this is photo of Android photo app"**, [**"dayadad4390@gmail.com"**](mailto:dayadad4390@gmail.com), email, **"/sdcard/daya.jpg"**);

Log.i(**"mail sent"**, **"mail sent"**);

} **catch** (**final** Exception e) { Log.e(**"SendMail"**, e.getMessage(), e); runOnUiThread(**new** Runnable() {

@Override

**public void** run() { *// TODO Auto-generated*

Toast.makeText(OneShotPreviewActivity.**this**,

**""** + e.getMessage(), 600000).show();

}

});

}

}

}.start(); finish();

*// takepicture();*

}

};

**public void** takepicture() {

\_camera.setOneShotPreviewCallback(mPreviewListener);

}

@Override

**public boolean** onTouchEvent(MotionEvent event) {

**if** (event.getAction() == MotionEvent.ACTION\_DOWN) {

*// \_camera.setOneShotPreviewCallback(mPreviewListener);*

}

**return super**.onTouchEvent(event);

}

**public void** setCameraDisplayOrientation(android.hardware.Camera camera,

**int** cameraId) { android.hardware.Camera.CameraInfo info = **new**

android.hardware.Camera.CameraInfo(); android.hardware.Camera.getCameraInfo(cameraId, info);

**int** rotation = \_activity.getWindowManager().getDefaultDisplay()

.getRotation(); **int** degrees = 0; **switch** (rotation) {

**case** Surface.ROTATION\_0:

degrees = 0;

**break**;

**case** Surface.ROTATION\_90:

degrees = 90;

**break**;

**case** Surface.ROTATION\_180:

degrees = 180;

**break**;

**case** Surface.ROTATION\_270:

degrees = 270;

**break**;

}

**int** result;

**if** (info.facing == Camera.CameraInfo.CAMERA\_FACING\_FRONT) { result = (info.orientation + degrees) % 360;

result = (360 - result) % 360; *// compensate the mirror*

} **else** { *// back-facing*

result = (info.orientation - degrees + 360) % 360;

}

Log.d(**"camera"**, **"result= "** + result); camera.setDisplayOrientation(result);

}

**private** Camera openFrontFacingCameraGingerbread() {

**int** cameraCount = 0; Camera cam = **null**;

Camera.CameraInfo cameraInfo = **new** Camera.CameraInfo(); cameraCount = Camera.getNumberOfCameras();

**for** (**int** camIdx = 0; camIdx < cameraCount; camIdx++) { Camera.getCameraInfo(camIdx, cameraInfo);

**if** (cameraInfo.facing == Camera.CameraInfo.CAMERA\_FACING\_BACK) {

**try** {

cam = Camera.open(camIdx);

} **catch** (RuntimeException e) { Log.e(**"camera"**,

**"Camera failed to open: "** + e.getLocalizedMessage());

}

}

}

**return** cam;

}

**public class** CameraPreview **extends** SurfaceView **implements**

SurfaceHolder.Callback {

**private** SurfaceHolder holder; CameraPreview(Context context) {

**super**(context);

holder = getHolder(); holder.addCallback(**this**);

holder.setType(SurfaceHolder.SURFACE\_TYPE\_PUSH\_BUFFERS);

}

**public void** surfaceChanged(SurfaceHolder holder, **int** format, **int** width,

**int** height) {

configure(format, width, height);

**try** { Thread.sleep(1000);

\_camera.setOneShotPreviewCallback(mPreviewListener);

} **catch** (InterruptedException e) {

*// TODO Auto-generated catch block*

e.printStackTrace();

}

}

**public void** surfaceDestroyed(SurfaceHolder holder) {

**if** (\_camera != **null**) {

\_camera.stopPreview();

\_camera.release();

\_camera = **null**;

}

}

**public void** surfaceCreated(SurfaceHolder holder) {

*// \_camera = Camera.open();*

\_camera = openFrontFacingCameraGingerbread();

**try** {

\_camera.setPreviewDisplay(holder);

} **catch** (IOException e1) {

*// TODO Auto-generated catch block*

e1.printStackTrace();

}

setCameraDisplayOrientation(\_camera, 0);

\_camera.startPreview();

}

**protected void** setPictureFormat(**int** format) {

Camera.Parameters params = \_camera.getParameters(); List<Integer> supported = params.getSupportedPictureFormats(); **if** (supported != **null**) {

**for** (**int** f : supported) {

**if** (f == format) { params.setPreviewFormat(format);

\_camera.setParameters(params);

**break**;

}

}

}

}

**protected void** setPreviewSize(**int** width, **int** height) { Camera.Parameters params = \_camera.getParameters(); List<Camera.Size> supported = params.getSupportedPreviewSizes(); **if** (supported != **null**) {

**for** (Camera.Size size : supported) {

**if** (size.width <= width && size.height <= height) { params.setPreviewSize(size.width, size.height);

\_camera.setParameters(params);

**break**;

}

}

}

}

**public void** configure(**int** format, **int** width, **int** height) {

\_camera.stopPreview(); setPictureFormat(format); setPreviewSize(width, height);

\_camera.startPreview();

}

}

}