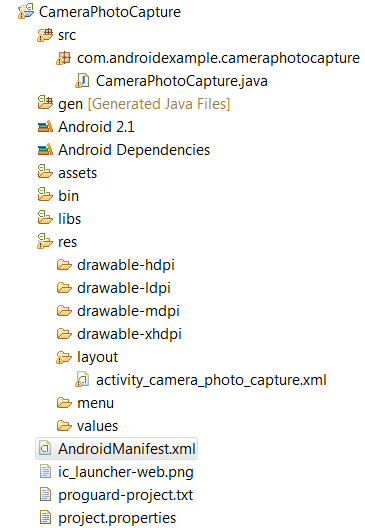
## Project Structure:



## CameraPhotoCapture.java file :

In this class

**1.**  Opening android phone inbuild camera  by camera intent.

**2.**   Capture photo and again return on CameraPhotoCapture.java  onActivityResult method.

**3.**   onActivityResult method  Convert image URI Path to  physical image path by convertImageUriToFile Method.

**4.**   Load Captured photo with AsyncTask  ( LoadImagesFromSDCard Class ).

public class CameraPhotoCapture extends Activity {

    final static int CAPTURE\_IMAGE\_ACTIVITY\_REQUEST\_CODE = 1;

    Uri imageUri                      = null;

    static TextView imageDetails      = null;

    public  static ImageView showImg  = null;

    CameraPhotoCapture CameraActivity = null;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_camera\_photo\_capture);

        CameraActivity = this;

        imageDetails = (TextView) findViewById(R.id.imageDetails);

        showImg = (ImageView) findViewById(R.id.showImg);

        final Button photo = (Button) findViewById(R.id.photo);

        photo.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {

              /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Camera Intent Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

                // Define the file-name to save photo taken by Camera activity

                String fileName = "Camera\_Example.jpg";

                // Create parameters for Intent with filename

                ContentValues values = new ContentValues();

                values.put(MediaStore.Images.Media.TITLE, fileName);

                values.put(MediaStore.Images.Media.DESCRIPTION,"Image capture by camera");

                // imageUri is the current activity attribute, define and save it for later usage

                imageUri = getContentResolver().insert(

                        MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, values);

                /\*\*\*\* EXTERNAL\_CONTENT\_URI : style URI for the "primary" external storage volume. \*\*\*\*/

                // Standard Intent action that can be sent to have the camera

                // application capture an image and return it.

                Intent intent = new Intent( MediaStore.ACTION\_IMAGE\_CAPTURE );

                 intent.putExtra(MediaStore.EXTRA\_OUTPUT, imageUri);

                 intent.putExtra(MediaStore.EXTRA\_VIDEO\_QUALITY, 1);

                startActivityForResult( intent, CAPTURE\_IMAGE\_ACTIVITY\_REQUEST\_CODE);

             /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Camera Intent End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

            }

        });

    }

     @Override

     protected void onActivityResult( int requestCode, int resultCode, Intent data)

        {

            if ( requestCode == CAPTURE\_IMAGE\_ACTIVITY\_REQUEST\_CODE) {

                if ( resultCode == RESULT\_OK) {

                   /\*\*\*\*\*\*\*\*\*\*\* Load Captured Image And Data Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

                    String imageId = convertImageUriToFile( imageUri,CameraActivity);

                   //  Create and excecute AsyncTask to load capture image

                    new LoadImagesFromSDCard().execute(""+imageId);

                  /\*\*\*\*\*\*\*\*\*\*\* Load Captured Image And Data End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

                } else if ( resultCode == RESULT\_CANCELED) {

                    Toast.makeText(this, " Picture was not taken ", Toast.LENGTH\_SHORT).show();

                } else {

                    Toast.makeText(this, " Picture was not taken ", Toast.LENGTH\_SHORT).show();

                }

            }

        }

     /\*\*\*\*\*\*\*\*\*\*\*\* Convert Image Uri path to physical path \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

     public static String convertImageUriToFile ( Uri imageUri, Activity activity )  {

            Cursor cursor = null;

            int imageID = 0;

            try {

                /\*\*\*\*\*\*\*\*\*\*\* Which columns values want to get \*\*\*\*\*\*\*/

                String [] proj={

                                 MediaStore.Images.Media.DATA,

                                 MediaStore.Images.Media.\_ID,

                                 MediaStore.Images.Thumbnails.\_ID,

                                 MediaStore.Images.ImageColumns.ORIENTATION

                               };

                cursor = activity.managedQuery(

                                imageUri,         //  Get data for specific image URI

                                proj,             //  Which columns to return

                                null,             //  WHERE clause; which rows to return (all rows)

                                null,             //  WHERE clause selection arguments (none)

                                null              //  Order-by clause (ascending by name)

                             );

                //  Get Query Data

                int columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.\_ID);

                int columnIndexThumb = cursor.getColumnIndexOrThrow(MediaStore.Images.Thumbnails.\_ID);

                int file\_ColumnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

                //int orientation\_ColumnIndex = cursor.

                //    getColumnIndexOrThrow(MediaStore.Images.ImageColumns.ORIENTATION);

                int size = cursor.getCount();

                /\*\*\*\*\*\*\*  If size is 0, there are no images on the SD Card. \*\*\*\*\*/

                if (size == 0) {

                    imageDetails.setText("No Image");

                }

                else

                {

                    int thumbID = 0;

                    if (cursor.moveToFirst()) {

                        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Captured image details \*\*\*\*\*\*\*\*\*\*\*\*/

                        /\*\*\*\*\*  Used to show image on view in LoadImagesFromSDCard class \*\*\*\*\*\*/

                        imageID     = cursor.getInt(columnIndex);

                        thumbID     = cursor.getInt(columnIndexThumb);

                        String Path = cursor.getString(file\_ColumnIndex);

                        //String orientation =  cursor.getString(orientation\_ColumnIndex);

                        String CapturedImageDetails = " CapturedImageDetails : \n\n"

                                                          +" ImageID :"+imageID+"\n"

                                                          +" ThumbID :"+thumbID+"\n"

                                                          +" Path :"+Path+"\n";

                        // Show Captured Image detail on activity

                        imageDetails.setText( CapturedImageDetails );

                    }

                }

            } finally {

                if (cursor != null) {

                    cursor.close();

                }

            }

            // Return Captured Image ImageID ( By this ImageID Image will load from sdcard )

            return ""+imageID;

        }

         /\*\*

         \* Async task for loading the images from the SD card.

         \*

         \* @author Android Example

         \*

         \*/

        // Class with extends AsyncTask class

     public class LoadImagesFromSDCard  extends AsyncTask<String, Void, Void> {

            private ProgressDialog Dialog = new ProgressDialog(CameraPhotoCapture.this);

            Bitmap mBitmap;

            protected void onPreExecute() {

                /\*\*\*\*\*\* NOTE: You can call UI Element here. \*\*\*\*\*/

                // Progress Dialog

                Dialog.setMessage(" Loading image from Sdcard..");

                Dialog.show();

            }

            // Call after onPreExecute method

            protected Void doInBackground(String... urls) {

                Bitmap bitmap = null;

                Bitmap newBitmap = null;

                Uri uri = null;

                    try {

                        /\*\*  Uri.withAppendedPath Method Description

                        \* Parameters

                        \*    baseUri  Uri to append path segment to

                        \*    pathSegment  encoded path segment to append

                        \* Returns

                        \*    a new Uri based on baseUri with the given segment appended to the path

                        \*/

                        uri = Uri.withAppendedPath(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, "" + urls[0]);

                        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Decode an input stream into a bitmap. \*\*\*\*\*\*\*\*\*/

                        bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(uri));

                        if (bitmap != null) {

                            /\*\*\*\*\*\*\*\*\* Creates a new bitmap, scaled from an existing bitmap. \*\*\*\*\*\*\*\*\*\*\*/

                            newBitmap = Bitmap.createScaledBitmap(bitmap, 170, 170, true);

                            bitmap.recycle();

                            if (newBitmap != null) {

                                mBitmap = newBitmap;

                            }

                        }

                    } catch (IOException e) {

                        // Error fetching image, try to recover

                        /\*\*\*\*\*\*\*\*\* Cancel execution of this task. \*\*\*\*\*\*\*\*\*\*/

                        cancel(true);

                    }

                return null;

            }

            protected void onPostExecute(Void unused) {

                // NOTE: You can call UI Element here.

                // Close progress dialog

                  Dialog.dismiss();

                if(mBitmap != null)

                {

                  // Set Image to ImageView

                   showImg.setImageBitmap(mBitmap);

                }

            }

        }

}

<RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

     xmlns:tools="<http://schemas.android.com/tools>"

     android:layout\_width="fill\_parent"

     android:layout\_height="fill\_parent"

     tools:context=".CameraPhotoCapture" >

             <Button android:text="Capture Photo"

                     android:id="@+id/photo"

                     android:layout\_width="wrap\_content"

                     android:layout\_height="wrap\_content" />

             <TextView

                 android:id="@+id/message"

                 android:layout\_below="@+id/photo"

                 android:text="Click on button to capture image"

                 android:textSize="14dp"

                 android:layout\_width="wrap\_content"

                 android:layout\_height="wrap\_content"

                 android:layout\_marginTop="10dp"

                 android:layout\_marginLeft="20dp"

                 />

             <TextView

                 android:id="@+id/imageDetails"

                 android:layout\_below="@+id/message"

                 android:layout\_width="wrap\_content"

                 android:layout\_height="wrap\_content"

                 android:layout\_marginTop="5dp"

                 android:layout\_marginLeft="20dp"

                 />

             <ImageView

                 android:id="@+id/showImg"

                 android:layout\_width="wrap\_content"

                 android:layout\_height="wrap\_content"

                 android:layout\_centerHorizontal="true"

                 android:layout\_centerVertical="true"

                 android:layout\_below="@+id/imageDetails"

                 />

</RelativeLayout>

Prepared By: Prof. Kamlesh A. Meshram