**IMAGE MANAGEMENT SYSTEM**  
  
**1. Dependencies**

* Build Gradle­­­­­­­­­
* AndroidManifest

1. **Layout Components(User Interface)**

* Activity Main
* Activity Image Detail
* Card Layout

1. **Main Components(Java)**

* Main Activity
* Recycler View Adapter(Module-1)
* Image Detail Activity(Module-2)
* Active Image Detail(Subclass)

1. **Server Side Components(PHP, MYSQL)**

* PHP, MYSQL

**DEPENDENCIES (User Interface)   
build.gradle**plugins {  
 id 'com.android.application'  
}  
android {  
 compileSdk 32  
 useLibrary 'org.apache.http.legacy'  
 defaultConfig {  
 applicationId "com.imagemanagementsystem"  
 minSdk 21  
 targetSdk 32  
 versionCode 1  
 versionName "1.0"  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 }  
 buildTypes {  
 release {  
 minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 }  
 }  
 compileOptions {  
 sourceCompatibility JavaVersion.VERSION\_1\_8  
 targetCompatibility JavaVersion.VERSION\_1\_8  
 }  
}

dependencies {  
 implementation 'androidx.appcompat:appcompat:1.5.1'  
 implementation 'com.google.android.material:material:1.6.1'  
 implementation 'com.squareup.picasso:picasso:2.71828'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
 implementation 'com.android.volley:volley:1.2.1'  
 testImplementation 'junit:junit:4.13.2'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
}

**AndroidManifest.xml**<? xml version="1.0" encoding="utf-8"?>

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

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

package="com.imagemanagementsystem">

<!-- permissions for reading external storage -->

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android: name="android.permission.ACCESS\_NETWORK\_STATE" />

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:hardwareAccelerated="false"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:largeHeap="true"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.ImageManagementSystem"

tools:targetApi="31">

<activity

android:name=".activity\_image\_detail"

android:exported="false" />

<activity

android:name=".ImageDetailActivity"

android:exported="false" />

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android: name="android.intent.action.MAIN" />

<category android: name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**LAYOUT COMPONENTS (User Interface)   
activity\_main.xml**<? xml version="1.0" encoding="utf-8"?>

<RelativeLayout

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

android:gravity="center"

android:orientation="vertical"

tools:context=".MainActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/idRVImages"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

/>

</RelativeLayout>

**activity\_image\_detail.xml**<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ImageDetailActivity">

<ImageView

android:id="@+id/idIVImage"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

android:layout\_centerInParent="true" />

</RelativeLayout>

**card\_layout.xml**<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView

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

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_margin="3dp"

android:elevation="8dp"

app:cardCornerRadius="8dp">

<!--Image view for displaying the image

in our card layout in recycler view-->

<ImageView

android:id="@+id/idIVImage"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_gravity="center"

android:scaleType="centerCrop" />

</androidx.cardview.widget.CardView>

**MAIN COMPONENTS (Java)   
MainActivity.java**package com.imagemanagementsystem;

import android.content.pm.PackageManager;

import android.database.Cursor;

import android.os.Bundle;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import androidx.recyclerview.widget.GridLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

import static android.Manifest.permission.READ\_EXTERNAL\_STORAGE;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

public class MainActivity extends AppCompatActivity {

private static final int PERMISSION\_REQUEST\_CODE = 200;

private ArrayList<String> imagePaths;

private RecyclerView imagesRV;

private RecyclerViewAdapter imageRVAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

requestPermissions();

imagePaths = new ArrayList<>();

imagesRV = findViewById(R.id.idRVImages);

prepareRecyclerView();

}

private boolean checkPermission() {

the result.

int result = ContextCompat.checkSelfPermission(getApplicationContext(), READ\_EXTERNAL\_STORAGE);

return result == PackageManager.PERMISSION\_GRANTED;

}

private void requestPermissions() {

if (checkPermission()) {

Toast.makeText(this, "Permissions Granted, Welcome to Image Management System.", Toast.LENGTH\_SHORT).show();

requestPermission();

imagePaths = new ArrayList<>();

imagesRV = findViewById(R.id.idRVImages);

prepareRecyclerView();

} else {

requestPermission();

}

}

private void requestPermission() {

//on below line we are requesting the read external storage permissions.

ActivityCompat.requestPermissions(this, new String[]{READ\_EXTERNAL\_STORAGE}, PERMISSION\_REQUEST\_CODE);

}

private void prepareRecyclerView() {

imageRVAdapter = new RecyclerViewAdapter(MainActivity.this, imagePaths);

GridLayoutManager manager = new GridLayoutManager(MainActivity.this, 4);

imagesRV.setLayoutManager(manager);

imagesRV.setAdapter(imageRVAdapter);

}

private void getImagePath() {

boolean isSDPresent = android.os.Environment.getExternalStorageState().equals(android.os.Environment.MEDIA\_MOUNTED);

if (isSDPresent) {

final String[] columns = {MediaStore.Images.Media.DATA, MediaStore.Images.Media.\_ID};

final String orderBy = MediaStore.Images.Media.\_ID;

Cursor cursor = getContentResolver().query(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, columns, null, null, orderBy);

int count = cursor.getCount();

for (int i = 1; i < count; i++) {

cursor.moveToPosition(i);

int dataColumnIndex = cursor.getColumnIndex(MediaStore.Images.Media.DATA);

imagePaths.add(cursor.getString(dataColumnIndex));

}

imageRVAdapter.notifyDataSetChanged();

cursor.close();

}

}

@Override

public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

// this method is called after permissions has been granted.

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

switch (requestCode) {

// we are checking the permission code.

case PERMISSION\_REQUEST\_CODE:

// in this case we are checking if the permissions are accepted or not.

if (grantResults.length > 0) {

boolean storageAccepted = grantResults[0] == PackageManager.PERMISSION\_GRANTED;

if (storageAccepted) {

// if the permissions are accepted we are displaying a toast message

// and calling a method to get image path.

Toast.makeText(this, "Permissions Granted..",

Toast.LENGTH\_SHORT).show();

getImagePath();

} else {

// if permissions are denied we are closing the app and displaying the toast message.

Toast.makeText(this, "Permissions denied, Permissions are required to use the app..", Toast.LENGTH\_SHORT).show();

}

}

break;

}

}

}

**RecyclerViewAdapter.java**package com.imagemanagementsystem;

import android.content.Context;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.squareup.picasso.Picasso;

import java.io.File;

import java.util.ArrayList;

public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.RecyclerViewHolder> {

private final Context context;

private final ArrayList<String> imagePathArrayList;

public RecyclerViewAdapter(Context context, ArrayList<String> imagePathArrayList) {

this.context = context;

this.imagePathArrayList = imagePathArrayList;

}

@NonNull

@Override

public RecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

// Inflate Layout in this method which we have created.

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.card\_layout, parent, false);

return new RecyclerViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull RecyclerViewHolder holder, int position) {

File imgFile = new File(imagePathArrayList.get(position));

if (imgFile.exists())

if(1==1) {

Picasso.get().load(imgFile).placeholder(R.drawable.ic\_launcher\_background).into(holder.imageIV);

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(context, ImageDetailActivity.class);

i.putExtra("imgPath", imagePathArrayList.get(position));

context.startActivity(i);

}

});

}

}

@Override

public int getItemCount() {

return imagePathArrayList.size();

}

public static class RecyclerViewHolder extends RecyclerView.ViewHolder {

private final ImageView imageIV;

public RecyclerViewHolder(@NonNull View itemView) {

super(itemView);

// initializing our views with their ids.

imageIV = itemView.findViewById(R.id.idIVImage);

}

}

}

**ImageDetailActivity.java**package com.imagemanagementsystem;

import android.os.Bundle;

import android.view.MotionEvent;

import android.view.ScaleGestureDetector;

import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

import com.squareup.picasso.Picasso;

import java.io.File;

public class ImageDetailActivity extends AppCompatActivity {

// creating a string variable, image view variable

// and a variable for our scale gesture detector class.

String imgPath;

private ImageView imageView;

private ScaleGestureDetector scaleGestureDetector;

// on below line we are defining our scale factor.

private float mScaleFactor = 1.0f;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_image\_detail);

imgPath = getIntent().getStringExtra("imgPath");

imageView = findViewById(R.id.idIVImage);

scaleGestureDetector = new ScaleGestureDetector(this, new ScaleListener());

File imgFile = new File(imgPath);

if (imgFile.exists()) { Picasso.get().load(imgFile).placeholder(R.drawable.ic\_launcher\_background).into(imageView);

}

}

@Override

public boolean onTouchEvent(MotionEvent motionEvent) {

// inside on touch event method we are calling on

// touch event method and passing our motion event to it.

scaleGestureDetector.onTouchEvent(motionEvent);

return true;

}

private class ScaleListener extends ScaleGestureDetector.SimpleOnScaleGestureListener {

// on below line we are creating a class for our scale

// listener and extending it with gesture listener.

@Override

public boolean onScale(ScaleGestureDetector scaleGestureDetector) {

mScaleFactor \*= scaleGestureDetector.getScaleFactor();

mScaleFactor = Math.max(0.1f, Math.min(mScaleFactor, 10.0f));

imageView.setScaleX(mScaleFactor);

imageView.setScaleY(mScaleFactor);

return true;

}

}

}

**activity\_image\_detail.java**package com.imagemanagementsystem;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class activity\_image\_detail extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_image\_detail);

}

}

**Server Components (PHP, MYSQL)   
GetImage.php**

<!DOCTYPE html>

<html>

<body>

<?php

$server\_name = "localhost";

$user\_name = "pgt";

$password = "";

$database = "image\_management\_system";

$isConnection = new mysqli($server\_name, $user\_name, $password,$database);

if($isConnection-> connect\_error)

die("Connection failed");

else

{

echo '<h1>Connection success!</h1>';

$images\_link = mysqli\_query($isConnection,"SELECT \* FROM images");

if($images\_link->num\_rows >0)

{

while($rows = mysqli\_fetch\_array($images\_link))

{

?>

<img width="75%" height="75%" src= "<?php echo $rows['image'] ?>" />

<?php

}

}

}

?>

</body>

</html>