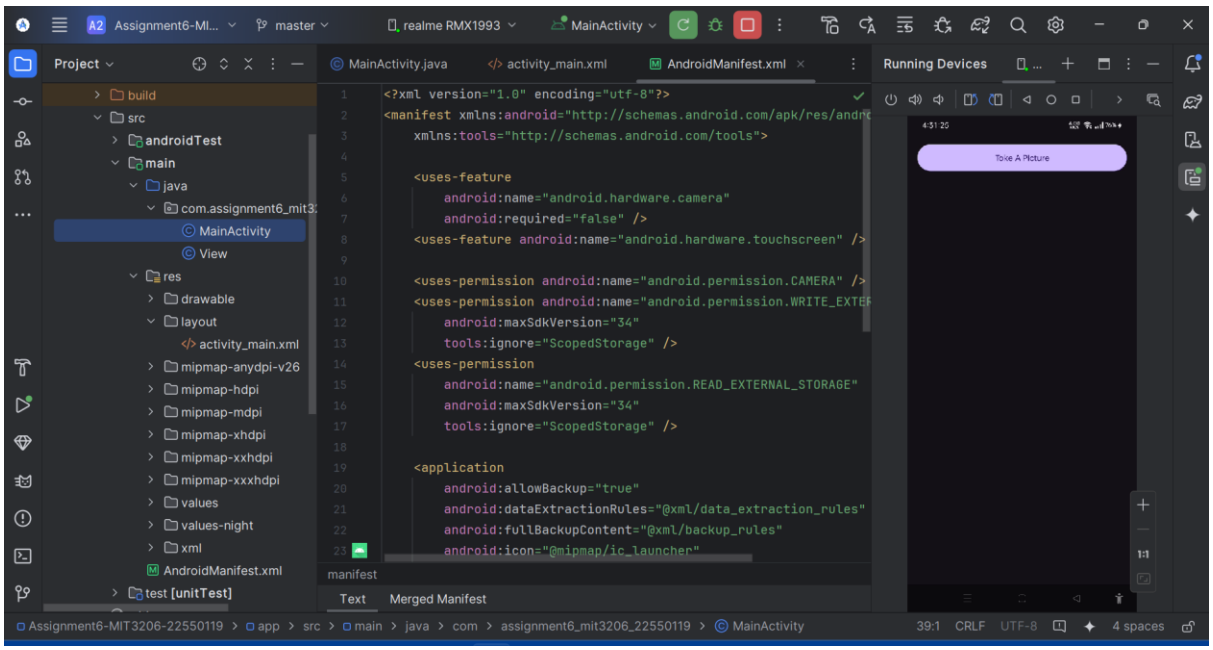
- **Course Module : MIT3206 – Mobile Computing**
- **Course Lecturer : Senior Lecturer Gihan P. Seneviratne Sir**

- **Assignment 6 : Camera Demo**
- **Used Android Studio : Android Studio Koala | 2024.1.1**
- **GitHub Private Repository Link : https://github.com/NavinduMadusanka/Assignment6-MIT3206-22550119.git**

- **Student Name : Kumarage Navindu Madusanka Dias (K.N.M. Dias)**
- **Student Index No : 22550119**
- **Student Registration No : 2022/MIT/011**
- **Email Address : navindu09@gmail.com**
- **Contact No : +94702678624**



Assignment6-MIT3206-22550119

Assignment 6 : Camera Demo

Below is a summary of what I have learned and focused on in this assignment.

1. Android Components

- **API level**

Android Version	API Level	Version Name
Android 7.0	24	Nougat

- **Permissions**

  uses-permission - READ_EXTERNAL_STORAGE

  uses-permission - WRITE_EXTERNAL_STORAGE

  uses-permission - ACCESS CAMERA

- **Newly learned key points in this assignment**
  - Granting user permissions to access headwear (External storage & built-in camera)
  - Reducing picture file capacity

- **tools:targetApi="34" in AndroidMainfest.xml**

- **uses-feature android:name="android.hardware.touchscreen"**

  uses-feature android:name="android.hardware.camera"

2. Functionality of the mobile application

- The app opens with a "Take A Picture" button to take a photo.
- Pressing the "Take A Picture" button opens the camera and gives a chance to the photo.

- Once the photo is taken, an image of it is displayed.

  If the photo is not up to standard, you will have the opportunity to try again, at the touch of the "Retry = (x)" button.

  If the photo is at the appropriate level, you can press the "Ok = ✔ " button to save it.
- After pressing the "Ok" button, its capacity is reduced and it is saved in the external storage.

- **Functions descriptions**

In this assignment I will mention the special activity functions that I have just learned.

No	Function	Description
1	onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)	This function is a callback for the result of the permission request. If the permission is granted, it opens the camera application. If not, it shows a message to the user (although this part is not implemented in the provided code).
2	onActivityResult(int requestCode, int resultCode, Intent data)	This function is a callback for the result of the camera activity. If the result is OK, it retrieves the captured image from the intent, resizes it, and saves it to the SD card.
3	saveImageToSDCard(Bitmap bitmap)	This function saves a bitmap to the SD card. It resizes the bitmap, creates a file in the Pictures directory, and writes the bitmap to the file.

3. Running the Application on my android mobile device

I was running the android app for testing in my android mobile device.

My android mobile device is Realme X2 RMX1993.

Below is a photo of my android mobile device (Realme X2 RMX1993) while the app was running.

My android mobile device display setting is set as dark mode option.

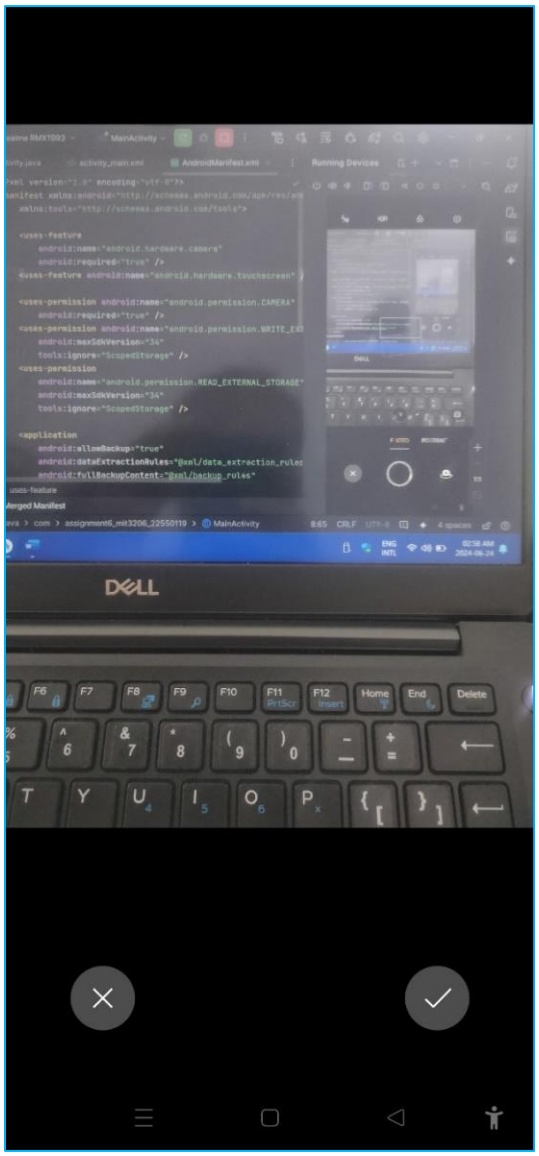


Photo 1 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

Photo 2 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

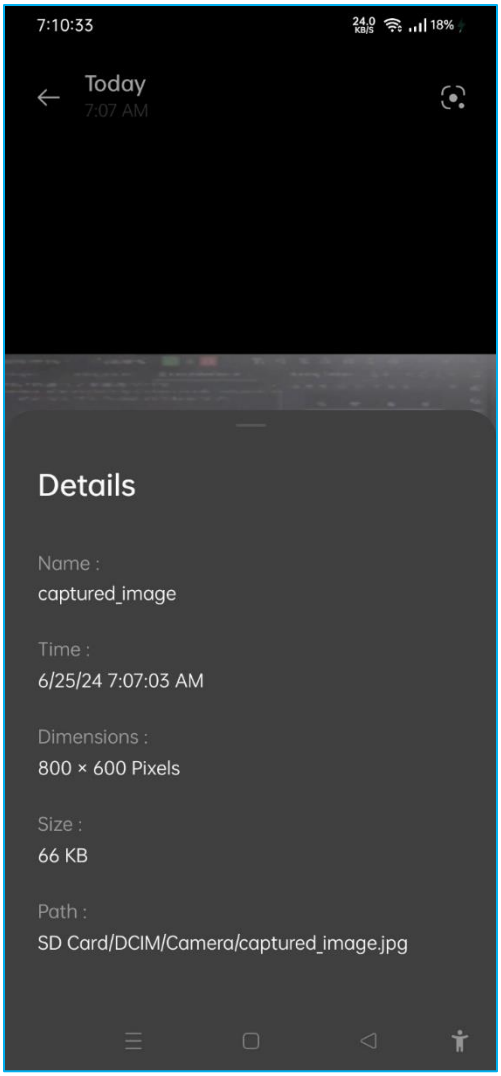


Photo 3 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

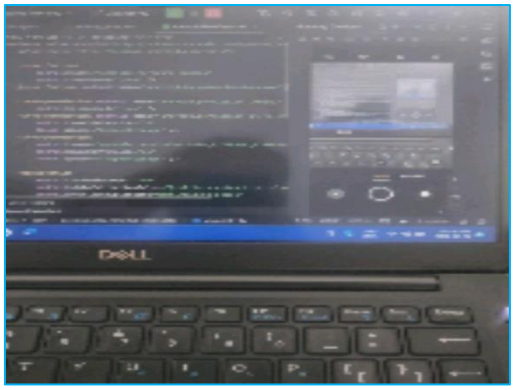


Photo 4 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

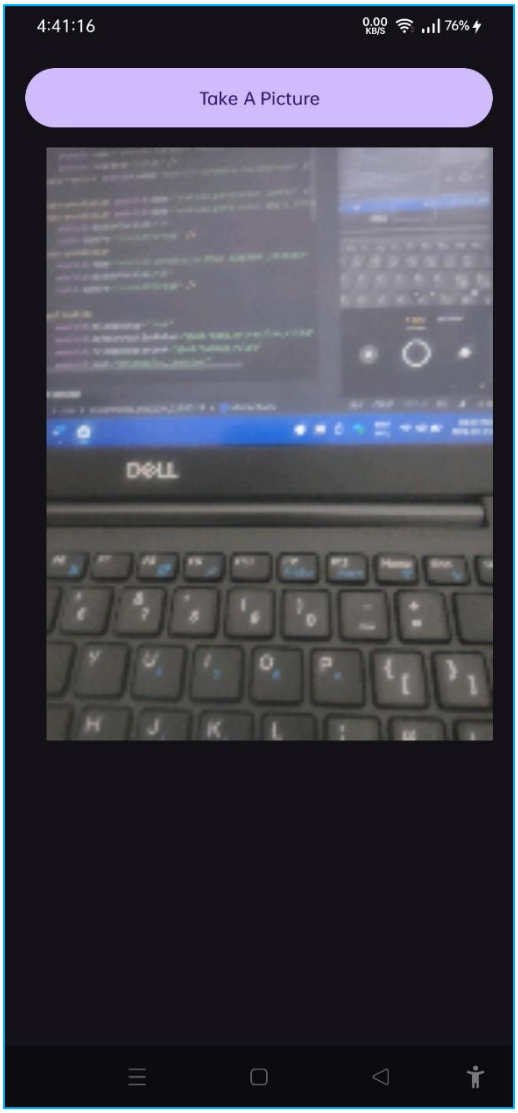


Photo 5 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

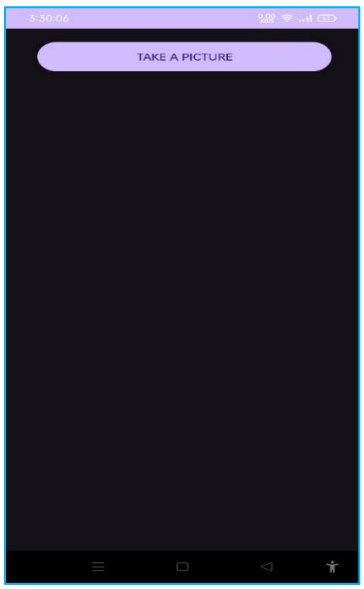


Photo 6 : Assignment6-MIT3206-22550119 in Realme X2 RMX1993

4. Main Coding files

- **MainActivity.java**

```
package com.assignment6_mit3206_22550119;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.ImageView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    public static final int REQUEST_CAMERA_PERMISSION_CODE = 1;
    private static final int REQUEST_IMAGE_CAPTURE = 2;
    private ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        imageView = findViewById(R.id.image_view);
    }

    // Check for permission at runtime
    public void captureImage(View view) {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.CAMERA},
REQUEST_CAMERA_PERMISSION_CODE);
        } else {
            openCamera();
        }
    }

    // Open camera application
    private void openCamera() {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);
        }
    }
```

```
    @Override
    public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_CAMERA_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                openCamera();
            } else {
                // Permission denied, show a message to the user
            }
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode ==
RESULT_OK) {
            Bundle extras = data.getExtras();
            Bitmap imageBitmap = (Bitmap) extras.get("data");
            imageView.setImageBitmap(imageBitmap);

            // Resize the image
            Bitmap resizedBitmap = resizeBitmap(imageBitmap, 800, 800);

            // Save the resized image to SD card
            saveImageToSDCard(resizedBitmap);
        }
    }

    private Bitmap resizeBitmap(Bitmap original, int width, int height) {
        return Bitmap.createScaledBitmap(original, width, height, true);
    }

    private void saveImageToSDCard(Bitmap bitmap) {
        // Reduce the size of the captured image
        Bitmap resizedBitmap = Bitmap.createScaledBitmap(bitmap, 800, 600,
true);

        // Create a file to save the image
        File imageFile = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PI
CTURES), "captured_image.jpg");

        try {
            // Save the image to the SD card
            FileOutputStream fos = new FileOutputStream(imageFile);
            resizedBitmap.compress(Bitmap.CompressFormat.JPEG, 90, fos);
            fos.close();

            // Notify the user that the image has been saved
            Uri imageUri = Uri.fromFile(imageFile);
            sendBroadcast(new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE,
imageUri));
        } catch (IOException e) {
            e.printStackTrace();
```

```
        }
    }
}
```

- **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:padding="15dp"
    tools:context=".MainActivity">

    <!-- add Camera Button to open the Camera-->
    <Button
        android:id="@+id/camera_button"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginTop="25dp"
        android:onClick="captureImage"
        android:text="Take A Picture" />

    <!-- add ImageView to display the captured image-->
    <ImageView
        android:layout_marginTop="70dp"
        android:layout_width="350dp"
        android:layout_marginLeft="15dp"
        android:layout_height="450dp"
        android:id="@+id/image_view"
        android:layout_marginBottom="10dp"/>
</RelativeLayout>
```

- **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">

    <uses-feature
        android:name="android.hardware.camera"
        android:required="false" />
    <uses-feature android:name="android.hardware.touchscreen" />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        android:maxSdkVersion="34"
        tools:ignore="ScopedStorage" />
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"
        android:maxSdkVersion="34"
        tools:ignore="ScopedStorage" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
```

```
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Assignment6-MIT3206-22550119"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Assignment6MIT320622550119"
        tools:targetApi="34">
        <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>
```

- **View.java**

```
package com.assignment6_mit3206_22550119;

public class View {
}
```