Abstract

In this project i developed an android custom camera software which is highly useful for any android phone or tablet. As modern civilization requires digitalization, as a result the need of software is increasing everywhere, it not only saves our time but also gives fastest output.

I analyzed user’s overview and developed this software for the sake of both user and developer.

The android framework includes support for various cameras and camera features available on devices, allowing us to capture pictures in our applications. This document discusses a quick, simple approach to image capture and outlines an advanced approach for creating custom camera experiences for our users.

Declaration

I hereby declare that, this project has been done under ETE-498 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

Tauseef Aziz

2012-2-55-058

Department of ECE

East West University

Letter of Acceptance

We hereby declare that this project is from the student’s own work and best effort of mine, and all other source of information used have been acknowledge. This project has been submitted with our approval.

Dr.Nahid Akter Jahan

Supervisor

Assistant Professor

Department of ECE

East West University

Dr. M. Mofazzol Hossain

Chairperson

Professor and Chairperson

Department of ECE

Department of Electronics and Communications Engineering

Acknowledgement

First of all, I would like to thank almighty Allah for giving me the strength & proper knowledge to complete my project work.

I would like to express my deep sense of gratitude and sincere thanks to my honorable supervisor Dr.**Nahid Akter Jahan** Assistant Professor, Department of ECE, East West University, Aftabnagar, Dhaka, Bangladesh for his at most direction, encouragement, kind guidance, sharing knowledge and constant inspiration throughout this project work.

My sincere gratefulness for the faculty of ECE whose friendly attitude and enthusiastic support that has given me for four years.

I am very grateful for the motivation and stimulation from my friends and seniors.

Finally my most heartfelt gratitude goes to my beloved parents for their endless support, continuous inspiration, great contribution and perfect guidance from the beginning to the end.

Table of Contents

Chapter 1

Introduction

- It is a open source software platform and operating system for mobile devices.

- Based on the Linux kernel.

- Developed by Google and later the Open Handset Alliance (OHA).

- Allows writing managed code in the Java language.

- Android has its own virtual machine i.e. DVM (Dalvik Virtual Machine), which is used for executing the android application.

- Google purchased the initial developer of the software, android incorporated in 2005.

- Most Android devices have at least one camera. Some devices have a front and a back facing camera.

- Using the camera on the Android device can be done via the integration of existing camera application. In this case you would start the existing Camera application via an *intent* and use the return data of the application to access the result.

- Alternatively you can also directly integrate the camera into your application via the Camera API.

1.1 Objective of the project

The android framework includes support for various cameras and camera features available on devices, allowing us to capture pictures in our applications. This document discusses a quick, simple approach to image capture and outlines an advanced approach for creating custom camera experiences for our users.

Chapter 2

Software Requirement Specification:

2.1 Tools and technology

- Android studio is the official integrated development environment (IDE) for android app development, based on intellij idea. on top of intellij's powerful code editor and developer tools, android studio offers even more features that enhance our productivity when building android apps, such as:

- A flexible gradle-based build system.

- A fast and feature-rich emulator.

- A unified environment where a developer can develop for all android devices.

- Instant run to push changes to our running app without building a new APK.

- Code templates and github integration to help us build common app features and import sample code.

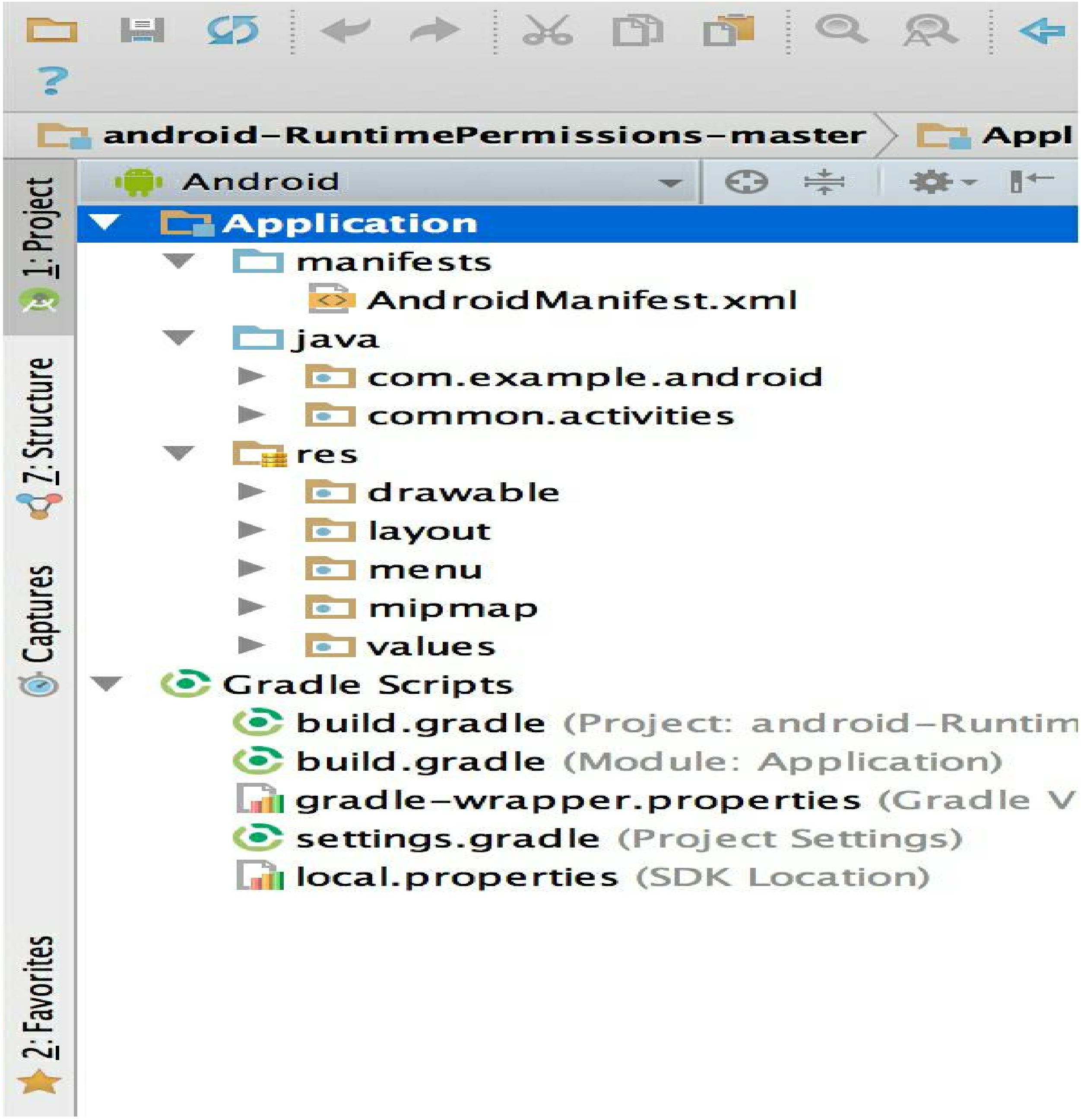
- Extensive testing tools and frameworks.

- Lint tools to catch performance, usability, version compatibility, and other problems.

- C++ and NDK support.

- Built-in support for google cloud platform, making it easy to integrate google cloud messaging and app engine.

2.2 The project file in android view



```
android-RuntimePermissions-master > Appl
Android
▼ Application
    ▼ manifests
          AndroidManifest.xml
    ▼ java
        ► com.example.android
        ► common.activities
    ▼ res
        ► drawable
        ► layout
        ► menu
        ► mipmap
        ► values
▼ Gradle Scripts
      build.gradle (Project: android-Runtim
      build.gradle (Module: Application)
      gradle-wrapper.properties (Gradle V
      settings.gradle (Project Settings)
      local.properties (SDK Location)
```

2.3 Requirments

RAM: 512MB or above.

Memory: 50MB or above.

OS: API level higher or kitkat 4.4(minimum).

2.4 Functionality

- By opening the android camera app there will be a initial photo with a button TAKE PHOTO.
- When clicking the TAKE PHOTO button camera will be turn on.
- After taking the captured photo we will see RETRY & OK option. Clicking RETRY the camera will turn on again for better caption and OK will lead us next user interface.
- Then we will see the captured photo with a SAVE PHOTO button.
- Clicking the SAVE PHOTO button there will be a toast for notifying the user where is the picture saved and SAVE PHOTO button turns to TAKE PHOTO button at a time for the next caption.

2.5 User Characteristics

Anyone can use it for the betterment of life and save their memories.

2.6 Description:

The android framework supports capturing images through the android.hardware.camera2API or camera intent. Here are the relevant classes:
This package is the primary API for controlling device cameras. It can be used to take pictures or videos when you are building a camera application.

- Camera

This class is the older deprecated API for controlling device cameras.

- Surface view

This class is used to present a live camera preview to the user.

- Intent

An intent action type of MediaStore.ACTION IMAGE CAPTURE can be used to capture images without directly using the camera object.

Before starting development on the application with the camera API, we should make sure our manifest has the appropriate declarations to allow use of camera hardware and other related features.

- The application must request permission to use a device camera.

- For saving images to the device's external storage (SD card), we must also specify this in the manifest.

The camera class is used to set image capture settings, start/stop preview, snap pictures. This class is a client for the camera server, which manages the actual camera hardware. To access the device camera, we must declare the camera permission in android manifest. Also be sure to include the <users-feature> manifest element to declare camera features used by the application. For example, if anyone use the camera and auto-focus feature, the Manifest should include the following:

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

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

<uses-feature android:name="android.hardware.camera.autofocus" />

To take pictures with this class, use the following steps:

- Obtain an instance of Camera from open(int).

- Get existing (default) settings with getParameters().

- If necessary, modify the returned Camera.Parameters object and callsetParameters(Camera.Parameters).

- Call setDisplayOrientation(int) to ensure correct orientation of preview.

- **Important**: Pass a fully initialized SurfaceHolderto setPreviewDisplay(SurfaceHolder). Without a surface, the camera will be unable to start the preview.

- **Important**: Call startPreview() to start updating the preview surface. Preview must be started before we can take a picture.

- When we want, call takePicture(Camera.ShutterCallback, Camera.PictureCallback, Camera.PictureCallback, Camera.PictureCallback) to capture a photo. Wait for the callbacks to provide the actual image data.

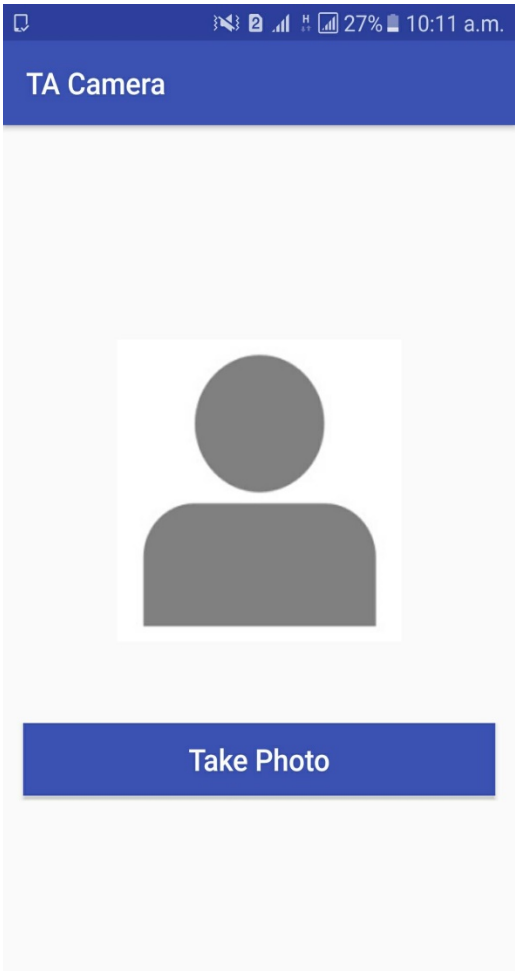
- After taking a picture, preview display will have stopped. To take more photos, call startPreview() again first.

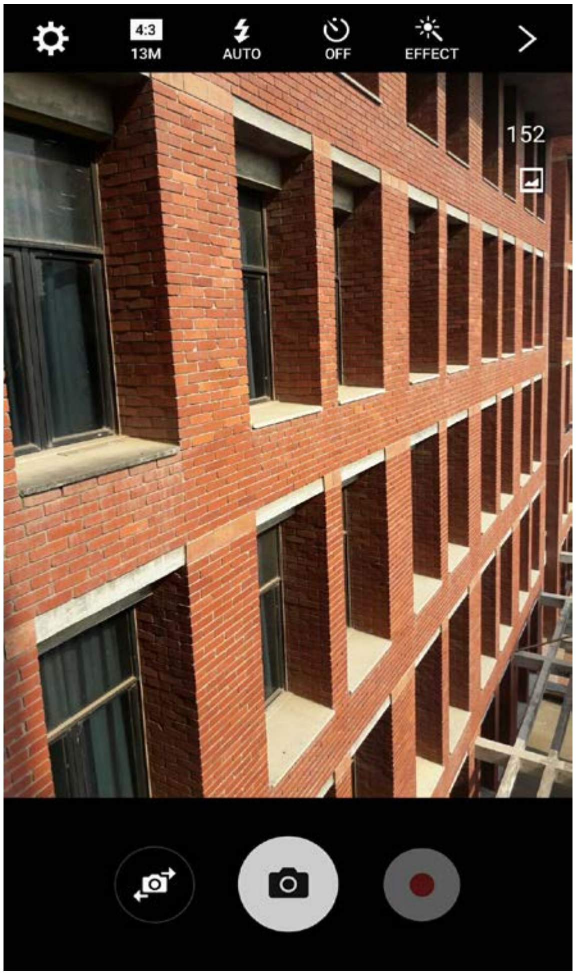
- Call stopPreview() to stop updating the preview surface.

- **Important:** Call release() to release the camera for use by other applications. Applications should release the camera immediately in onPause() (and re-open() it in onResume()).

Chapter 3

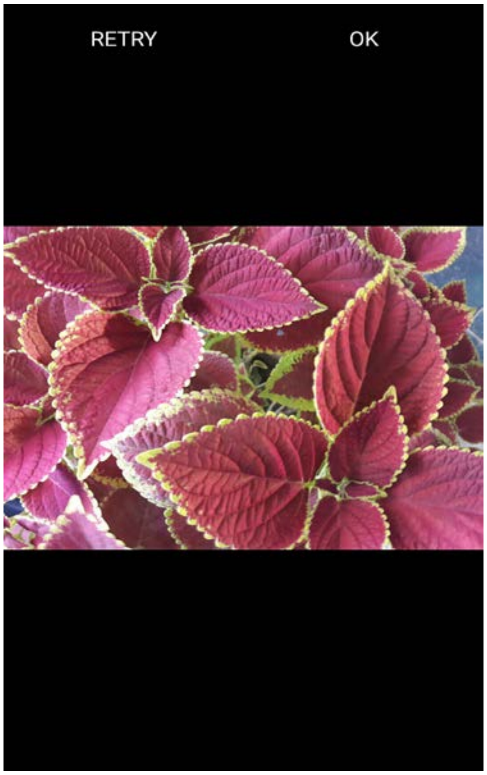
3.1 Operational screen-shots

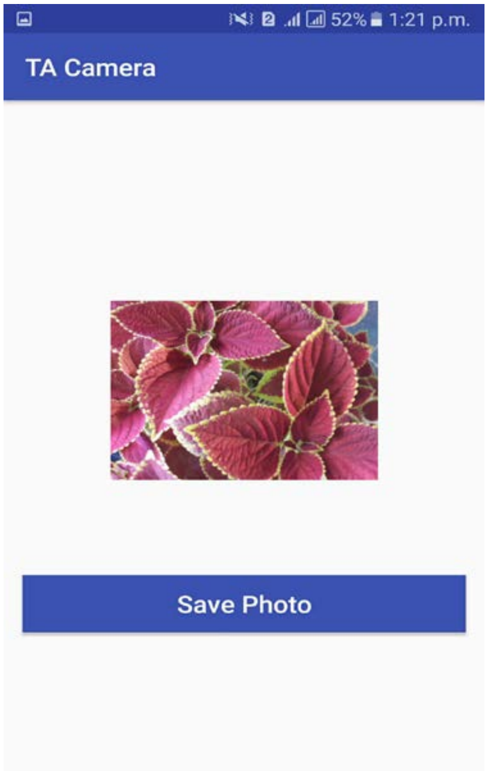




Step-1

Step-2





Step-3

Step-4

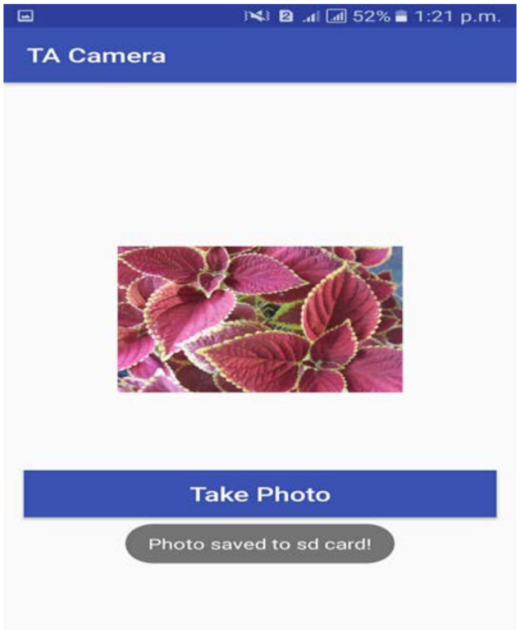
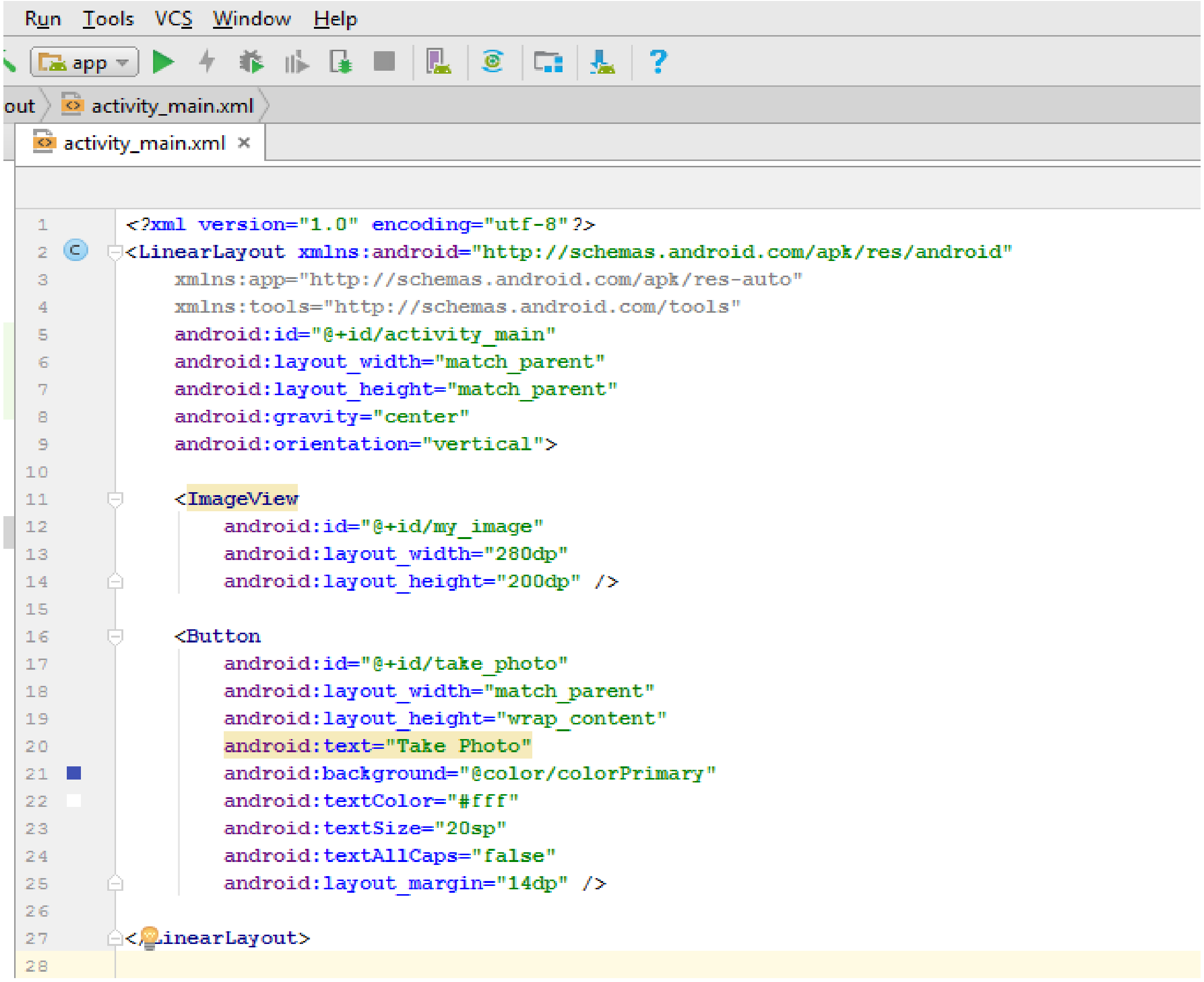


Photo saved to sd card!

Step-5

3.2 Screenshot and Source Code



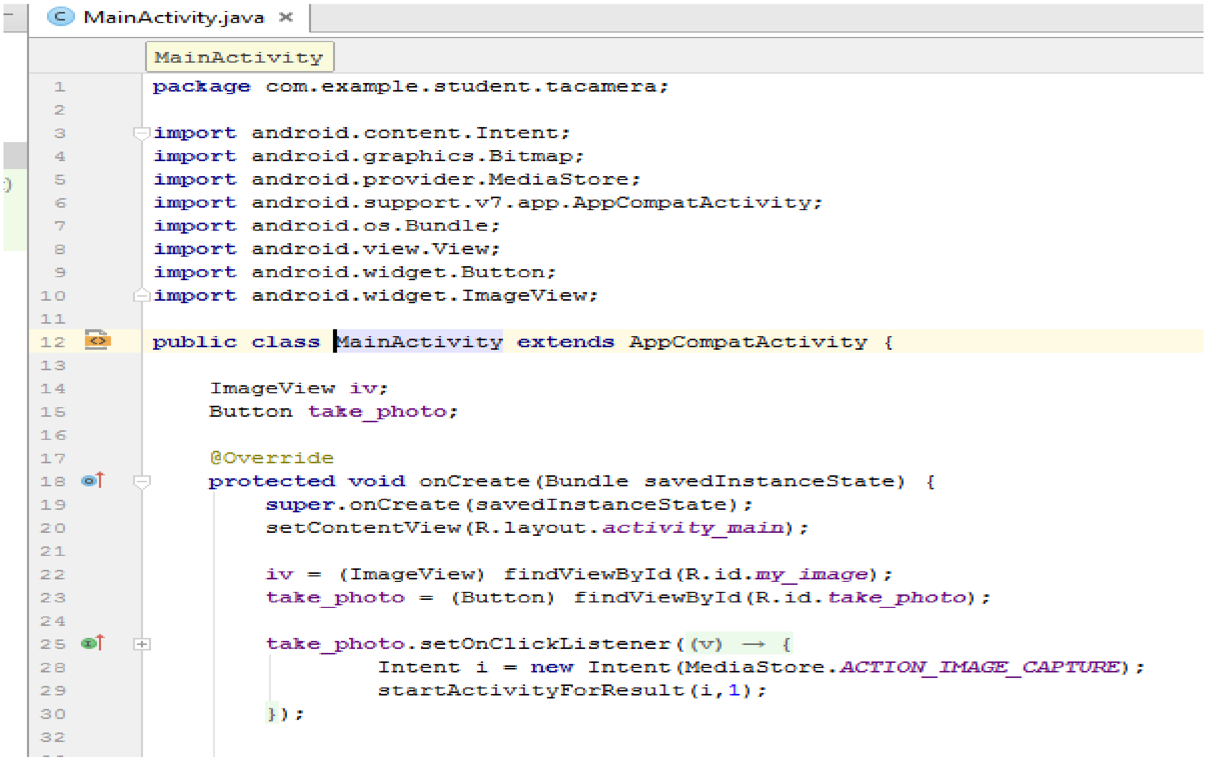
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/my_image"
        android:layout_width="280dp"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/take_photo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Take Photo"
        android:background="@color/colorPrimary"
        android:textColor="#fff"
        android:textSize="20sp"
        android:textAllCaps="false"
        android:layout_margin="14dp" />

</LinearLayout>
```

activity_main.xml

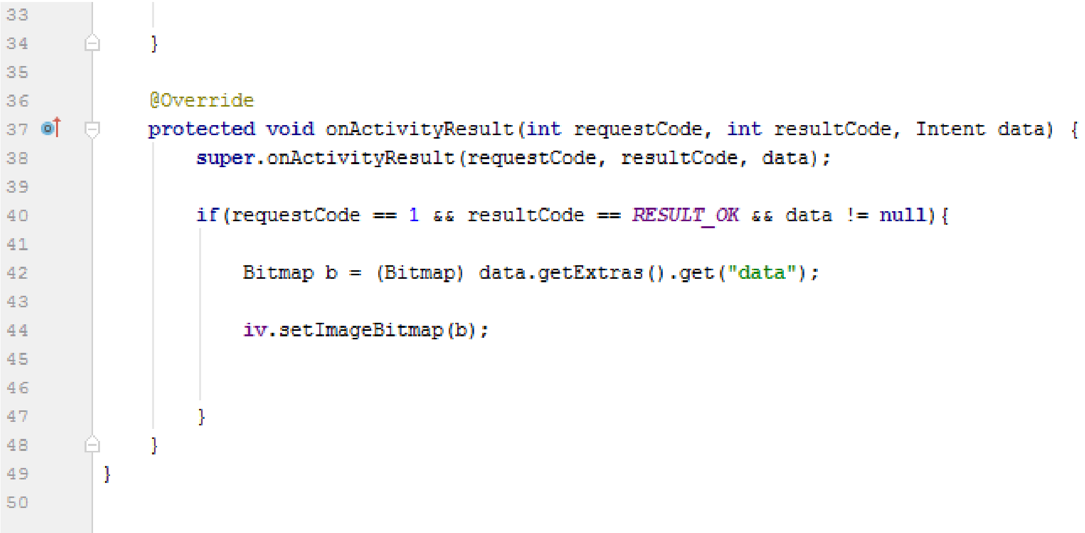
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"



   android:layout_height="match_parent"
    android:gravity="center"
```

```
android:orientation="vertical">

    <ImageView
        android:id="@+id/my_image"
        android:layout_width="280dp"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/take_photo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Take Photo"
        android:background="@color/colorPrimary"
        android:textColor="#fff"
        android:textSize="20sp"
        android:textAllCaps="false"
        android:layout_margin="14dp" />
</LinearLayout>
```

MainActivity.java



```
package com.example.student.tacamera;

import android.content.Intent;
import android.graphics.Bitmap;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    ImageView iv;
    Button take_photo;

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

        iv = (ImageView) findViewById(R.id.my_image);
        take_photo = (Button) findViewById(R.id.take_photo);

        take_photo.setOnClickListener((v) → {
                Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(i,1);
        });
```

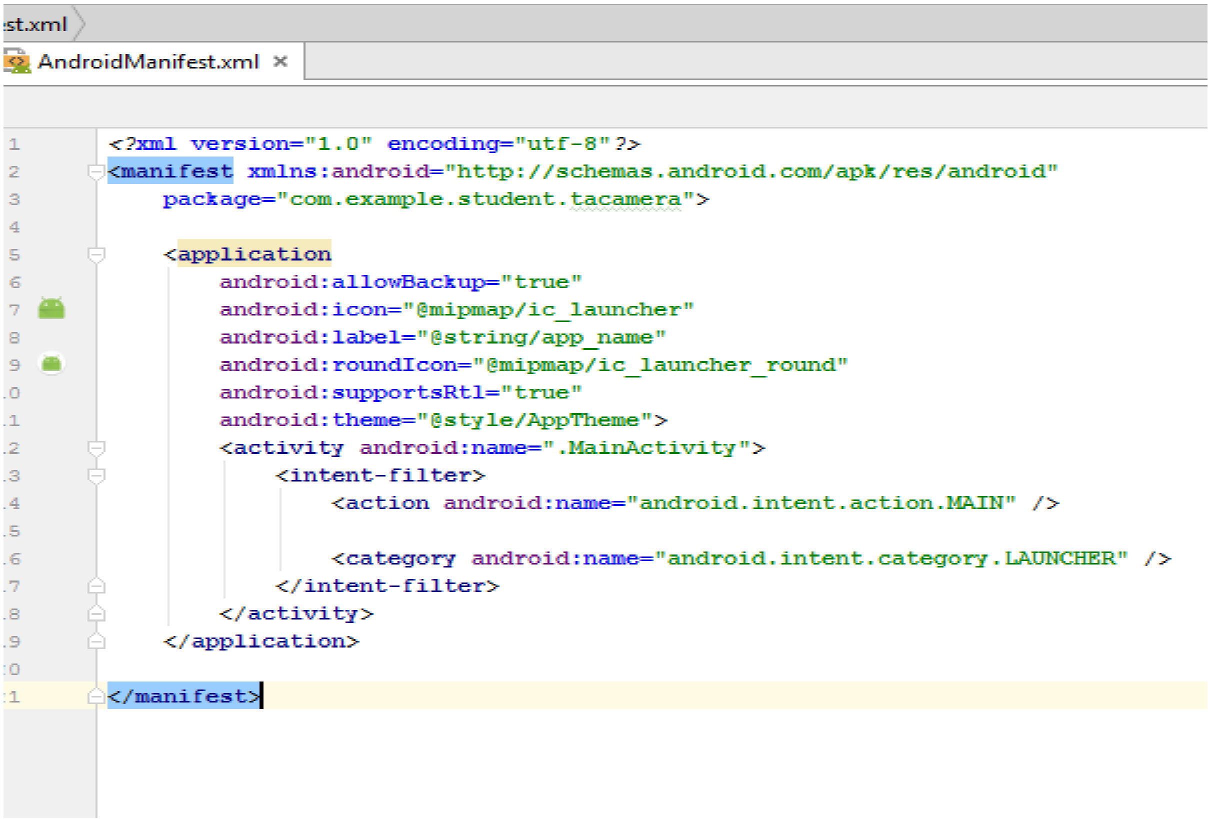


```

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 1 && resultCode == RESULT_OK && data != null){

            Bitmap b = (Bitmap) data.getExtras().get("data");

            iv.setImageBitmap(b);


        }
    }
}

```

MainActivity.java

```
package com.example.student.tacamera;
import android.content.Intent;
import android.graphics.Bitmap;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

   ImageView iv;
   Button take_photo;

   @Override
   protected void onCreate(Bundle savedInstanceState) {

       super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
```

```
        iv = (ImageView) findViewById(R.id.my_image);
        take_photo = (Button) findViewById(R.id.take_photo);

        take_photo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(i,1);
            }
        });


    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 1 && resultCode == RESULT_OK && data != null){

            Bitmap b = (Bitmap) data.getExtras().get("data");

            iv.setImageBitmap(b);
        }
    }
}
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.student.tacamera">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.student.tacamera">

    <application


        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">

        <intent-filter>
                <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

</manifest>
```

Chapter 4

4.1 Conclusion:

- Android is now becoming the best among all the mobile operating systems.   Requirments for android users:

    - Ram: minimum 512mb.
    - Memory: minimum 50mb space.
    - OS: API level higher or kitkat 4.4(minimum).

- Android camera is being used in 170 models of different 27 mobile companies.
- It surely takes lot of time for other os to create such a rapid revolution.

4.2 References

1. www.Google.com

2. www.android.com

3. www.java.com

4. www.w3schools.com