INTRODUCTION

INTRODUCTION TO ANDROID

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a start-up of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team).

Google wanted Android to be open and free; hence, most of the Android code was released under the open source Apache License, which means that anyone who wants to use Android can do so by downloading the full Android source code. Moreover, vendors (typically hardware manufacturers) can add their own proprietary extensions to Android and customize Android to differentiate their products from others. This simple development model makes Android very attractive and has thus piqued the interest of many vendors. This has been especially true for companies affected by the phenomenon of Apple's IPhone, a hugely successful product that revolutionized the smart phone industry. Such companies include Motorola and Sony Ericsson, which for many years have been developing their own mobile operating systems. When the IPhone was launched, many of these manufacturers had to scramble to find new ways of revitalizing their products. These manufacturers see Android as a solution — they will continue to design their own hardware and use Android as the operating system that powers it.

The main advantage of adopting Android is that it offers a unified approach to application development. Developers need only develop for Android, and their applications should be able to run on numerous different devices, as long as the devices are powered using Android. In the world of smart phones, applications are the most important part of the success chain. Device manufacturers therefore see Android as their best hope to challenge the onslaught of the IPhone, which already commands a large base of applications.

1.1 Android

Android has gone through quite a number of updates since its first release.

- Alpha
- Beta
- Cupcake (1.5)
- Donut (1.6)
- Eclair (2.0–2.1)
- Froyo (2.2–2.2.3)
- Lolipop(5.0-5.1.1)
- Marshmallow(6.0-6.0.1)
- Nougat(7.0-7.1.2)
- Oreo(8.0-8.0.1)
- Pie(9.0)

Applications written for versions of Android prior to 3.0 are compatible with Android 3.0 devices, and they run without modifications. Android 3.0 tablet applications that make use of the newer features available in 3.0, however, will not be able to run on older devices. To ensure that an Android tablet application can run on all versions of devices, you must programmatically ensure that you only make use of features that are supported in specific versions of Android.

In October 2011, Google released Android 4.0, a version that brought all the features introduced in Android 3.0 to smart phones, along with some new features such as facial recognition unlock, data usage monitoring and control, Near Field Communication (NFC), and more.

Features of Android

Because Android is open source and freely available to manufacturers for Customization, there are no fixed hardware or software configurations. However, Android itself supports the following features:

➤Storage — Uses SQLite,a lightweight relational database, for data storage.

➤Connectivity — Supports GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE, and WiMAX.

➤Messaging — supports both SMS and MMS.

➤Web browser — Based on the open source WebKit, together with Chrome's V8 JavaScript engine

➤Media support — Includes support for the following media: H.263, H.264 (In 3GP or MP4container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4or3GP container), MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF, and BMP

➤Hardware support — Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor and GPS

➤Multi-touch — Supports multi-touch screens

➤Multi-tasking — Supports multi-tasking applications

➤Flash support — Android 2.3 supports Flash 10.1.

➤Tethering — supports sharing of Internet connections as a wired/wireless hotspot

Architecture of Android

The Android OS is roughly divided into five sections in four main layers:

➤Linux kernel — this is the kernel on which Android is based. This layer contains all the low level device drivers for the various hardware components of an Android device.

➤Libraries — these contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it forData storage. The WebKit library provides functionalities for web browsing.

➤Android runtime — at the same layer as the libraries, the Android runtime provides a setof core libraries that enable developers to write Android apps using the Java programminglanguage. The Android runtime also includes the Dalvik virtual machine, which enables every
Android application to run in its own process, with its own instance of the Dalvikvirtualmachine (Android applications are compiled into Dalvikexecutables). Dalvik is a specialized
Virtual machine designed specifically for Android and optimized for battery-powered mobileDevices with limited memory and CPU.

➤Application framework — exposes the various capabilities of the Android OS to applicationdevelopers so that they can make use of them in their applications.

➤Applications —at this top layer, you will find applications that ship with the Android device
(Such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.

1.2 Introduction on Android SQLite Database

The Android SQLite Database requires very little memory (around 250kb), which is available on all android devices. Every device has an inbuilt support for SQLite database, which is automatically managed on android right from its creation, execution to querying up process.SQLite is an open source database, available on every android database. It supports standard relations database features, like SQL syntax, transactions & SQL statements. SQLite is considerably, the lighter version of SQL database, where most of the SQL commands don't run on SQLite database. Once SQLite is in place, it is important to ensure that a feature or command is available in SQLite; only then can it be executed.

The Basic Advantages of SQLite:

- It's a light weight database
- Requires very little memory
- An Automatically managed database

The SQLite supports only 3 Data types:

- Text(like string) – for storing datatype store
- Integer(like int) – for storing integer primary key
- Real(like double)- for storing long values

Basically SQLite does not validate datatypes by itself. In other words, whatever datatypes are used, they are termed as valid.For example, in this case, and the database of a cable operator has been discussed. Here, a new table is added with 'text' in the name field and fieldname box carrying 'textfield'. A random value datatype has been created. The end result is a test table with an invalid datatype, which shows that SQLite doesn't validate data type.

INTRODUCTION TO ANDROID STUDIO

Android is a mobile operating system currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. And as we said before, Android offers a unified approach to application development for mobile devices. Android is an open-source operating system named Android.

Google has made the code for all the low-level "stuff" as well as the needed middleware to power and use an electronic device, and gave Android freely to anyone who wants to write code and build the operating system from it. There is even a full application framework included, so third-party apps can be built and installed, then made available for the user to run as they like.

It's an Android focused IDE, designed specially for the Android development. It was launched on 16th May 2013, during Google I/O 2013 annual event. Android studio contains all the Android sDK tools to design, test, debug and profile the app. One great thing is that it depends on the IntelliJ Idea IDE which is proved itself a great IDE and has been using by most all the Android engineers.

Features of Android Studio :-

Here are the features:

- Powerful code editing (smart editing, code re-factoring)
- Rich layout Editor (As you soon as you drag and drop views on the layout, it shows you preview in all the screens including Nexus 4, Nexus 7, Nexus 10 and many other resolutions. Layout designing can be done much faster way as compared to eclipse.)
- Gradle-based build support
- Maven Support
- Templated based wizards
- Lint tool analysis (The Android lint tool is a static code analysis tool that checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.)

It provides ability to capture directly generate a screenshot of your application. the SDK but Android Studio provides something more:

- Device frame (As frames for many Nexus devices are available, you can capture screenshot in whiever frame you like most)
- Drop shadow
- Screen glare

Android Studio contains tools such as the Android Virtual Device Manager and the Android Device Monitor. It also contains Gradle, which helps you configure your Android application seamlessly. Some of the interesting features of Android Studio include the following:

- Support for a fast emulator
- Support for Gradle
- Support for plenty of code templates and GitHub integration
- Support for Google Cloud Platform
- Support for template-based wizards for creating Android designs and components
- Support for rich layout editor
- Support for deep code analysis
- Support for extensive set of tools and frameworks

ENVIRONMENT SPECIFICATION

2.1 HARDWARE REQUIREMENTS

MONITOR : LCD Monitor

PROCESSOR : Pentium 2 onwards

RAM : 4 GB

HARD DISK : 64 GB

MOUSE : Logitech Compatible

2.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM : Windows 7

FRONT END : Android Studio

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Although there are few inbuilt filters already present, you may want to create and customize one specific to your need and show your creativity. For that you would require to know how all the Sub filters can be used. Let me take you through all of them. Nowadays image filters are quite common in lot of android apps. Instagram is famous for its popular filters feature and probably the first app to introduce image filters to android world.

There are lot of other image editing apps provides image filters and image editing features. Usually image processing operations will be done in native C/C++ languages. In android, you can write your library in C or C++ and use JNI (Java Native Interface) to make the functions accessible via java code. You can also consider using popular image processing libraries like open CV to make your own filters library.

3.2 PROPOSED SYSTEM.

This app is very easy and user friendly to use. It provides variety of filters that can improve the image as per the user's choice. Since Instagram is becoming very popular these days we have chosen to develop this app that can improve our skills in android and have overall idea about photo editors.

Zomato, Andorid Hive. This library offers basic image operations like controlling Brightness, Saturation and Contrast and few image filters. This app improvises the library and hosted it on a public maven repository so that we can integrate in our projects very easily.

This library is very basic and we can achieve great filters like Instagram using this. We have used number of features that currently Instagram doesn't consists of. So, the motive is the current Instagram can also have such kind of features that will help the users to create different styles of editing pictures.

Most of the people do not use photo editor because they lack in editing their photos. So instead they can directly go on Instagram and choose the filters and features however they want to use. In this way this helps people in saving their memory space on mobile phones rather than installing several editing apps on their phones.

3.3 FEASIBILITY STUDY

Feasibility study is a high-level capsule version of the system and deign process. The objective is to determine wether or not the proposed system is feasible.Three tests of feasibility have been carried out:
Technical Feasibility
Operational Feasibility
Economic Feasibility

TECHNICAL FEASIBILITY

In technical easibility study, we test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using Android platform. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence the solution is technically feasible.

OPERATIONAL FEASIBILITY

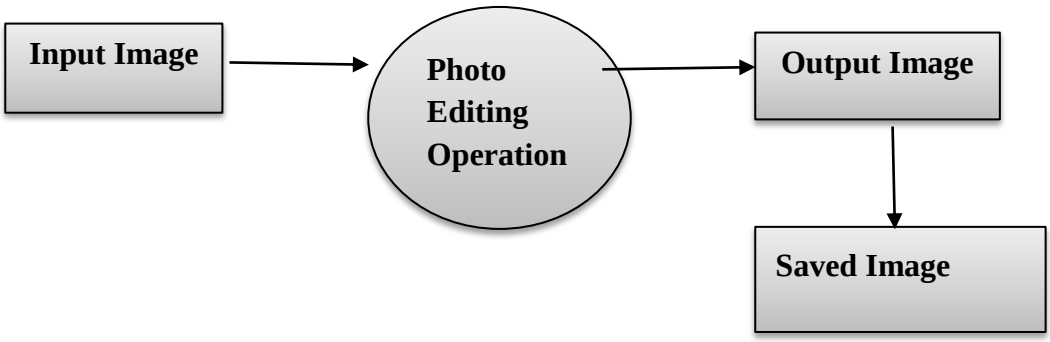
In operational feasibility study, we check whether the network infrastructure is in place along with a centralized server with administrative staff. Most organisations are equipped with sufficient computers for each employee with intranet facility. The proposed system is acceptable to users. Hence it is operationally feasible.

ECONOMIC FEASIBILITY

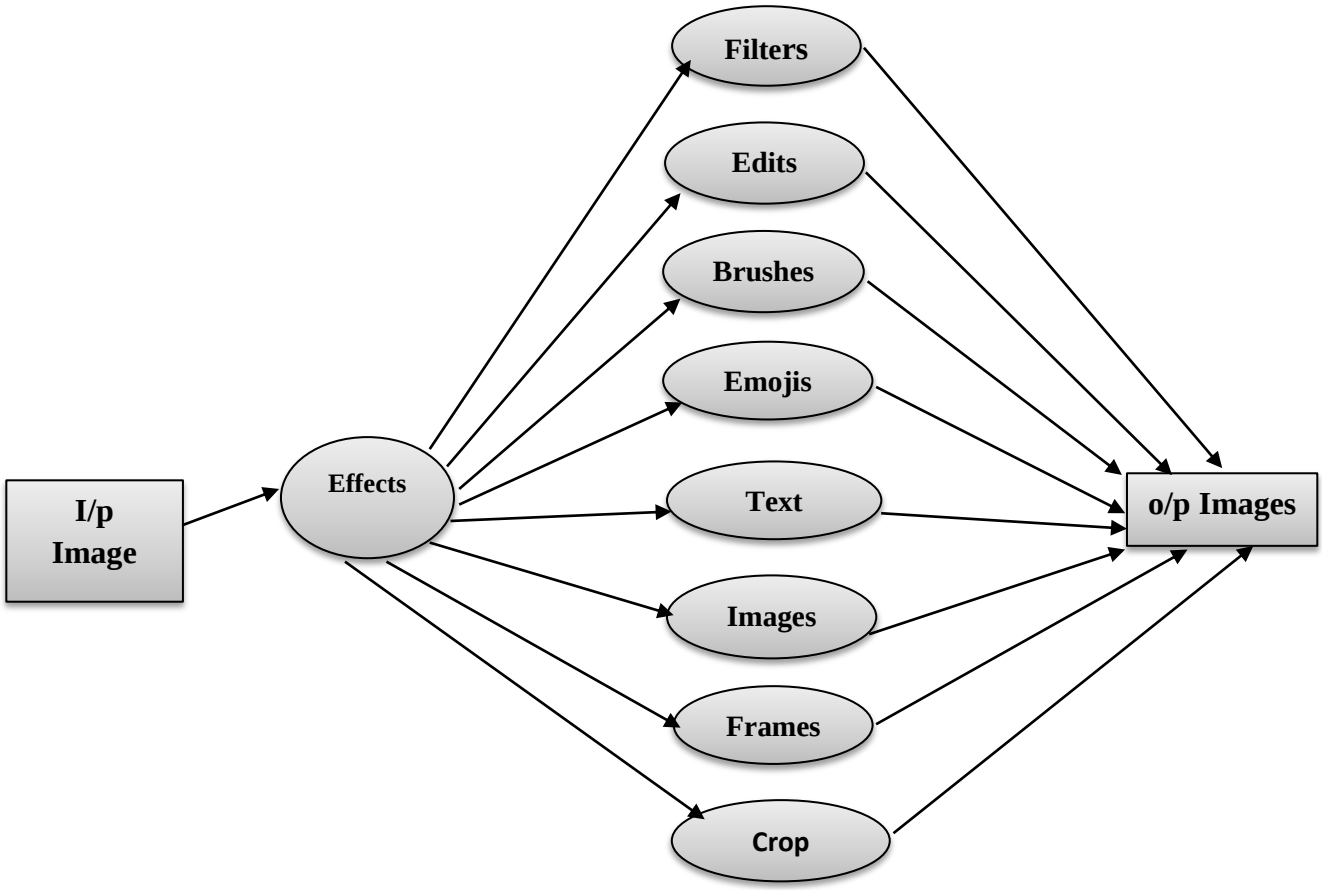
As a part of economic feasibility study, the costs and benefits associated with the proposed system are compared. Heme the tangible and intangible benefits outweight the system development cost. So the proposed system is economically feasible.

SYSTEM DESIGN

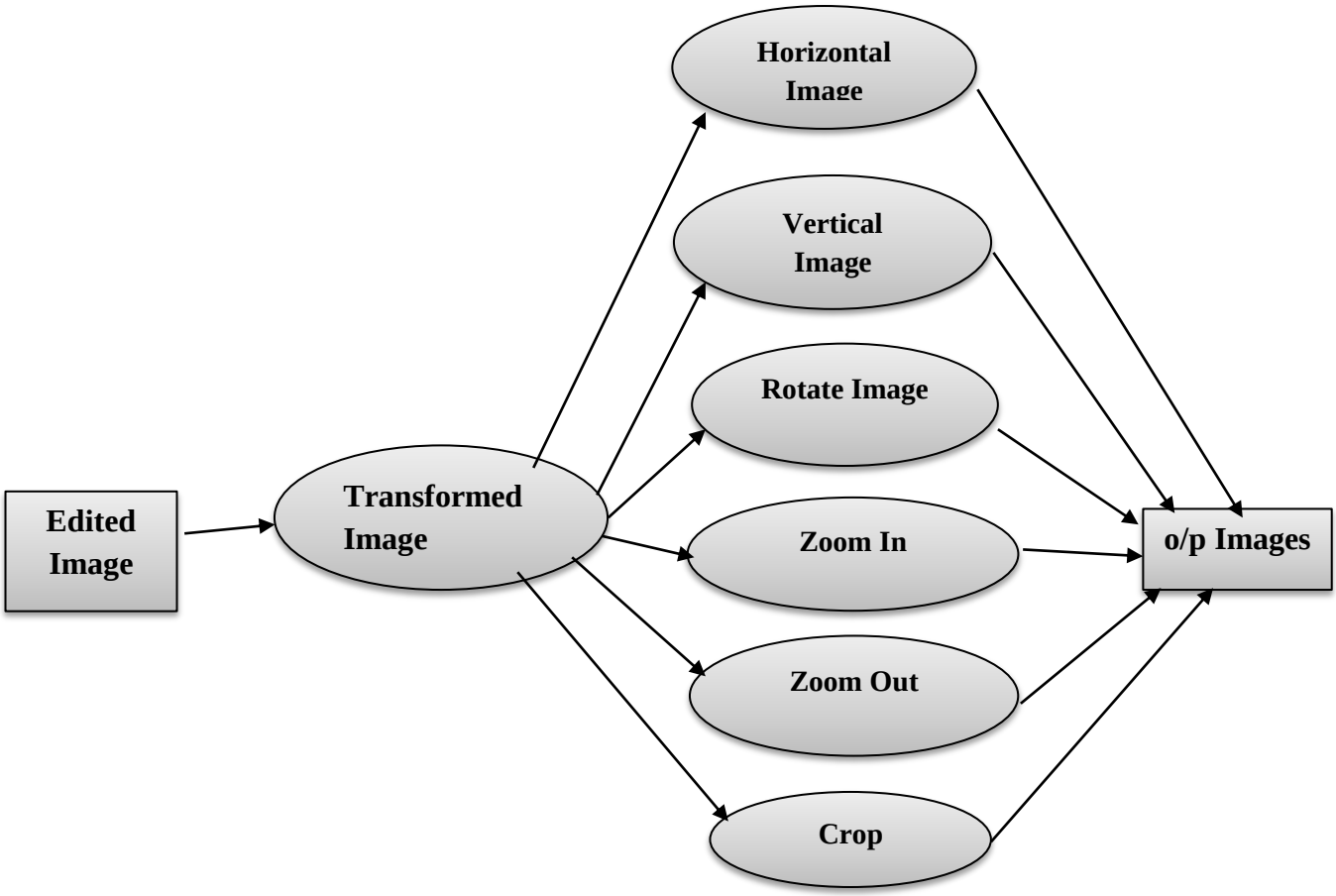
4.1 DATA FLOW DIAGRAM (DFD)



Level 1 :-

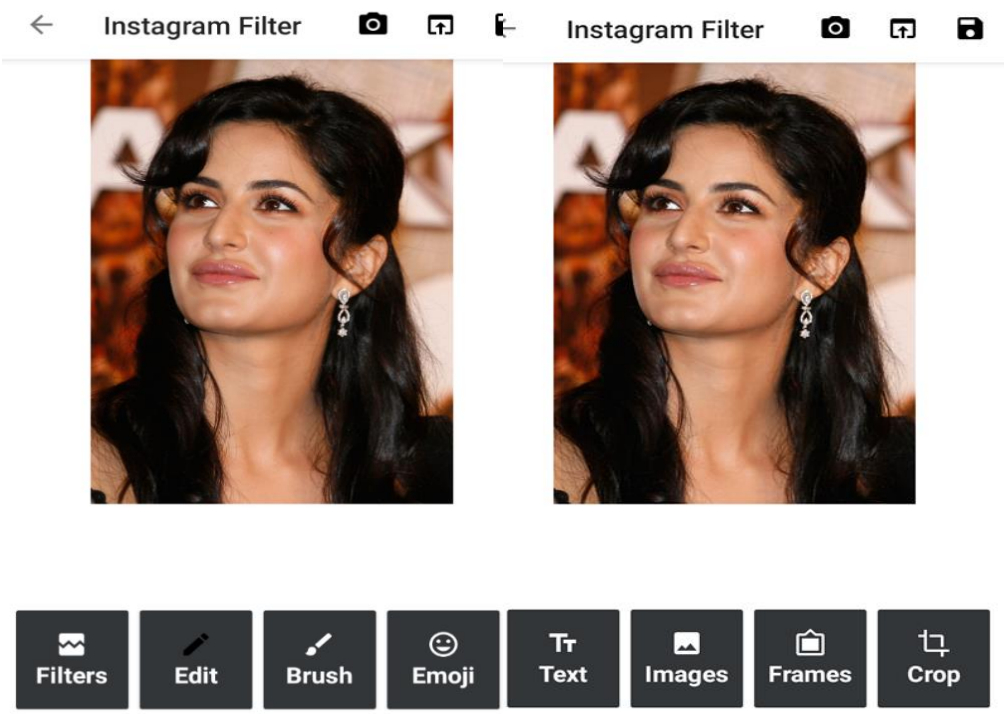


Level 2 :-

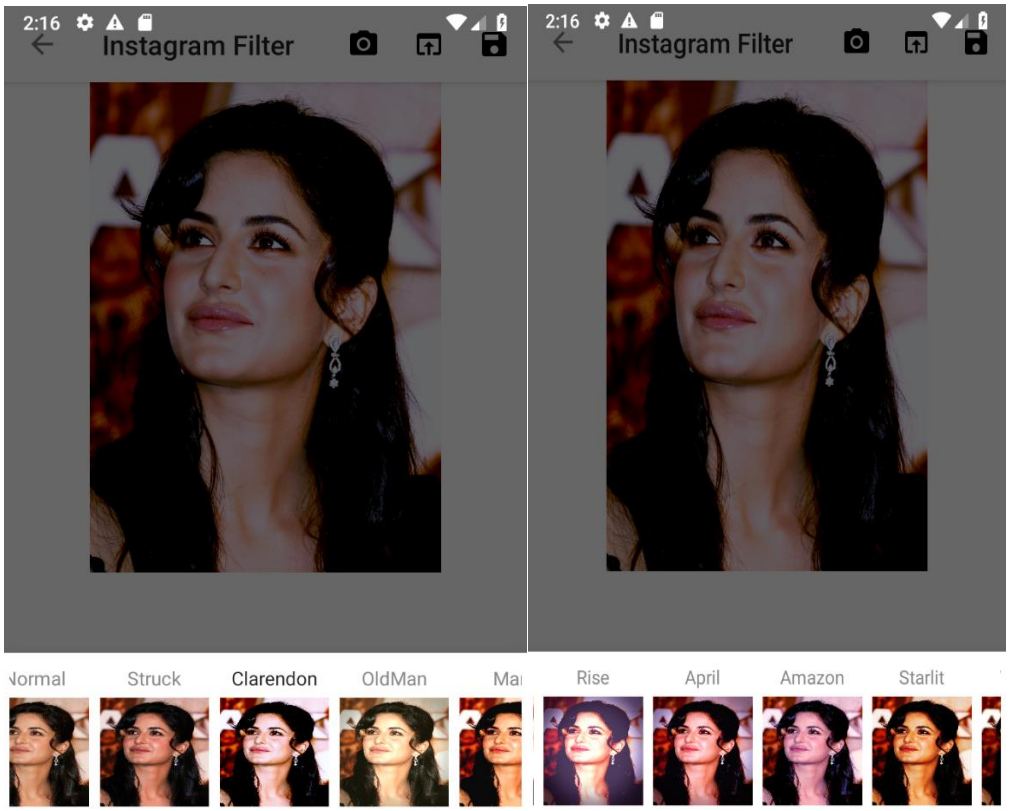

Horizontal Image
Vertical Image
Rotate Image
Edited Image
Transformed Image
Zoom In
o/p Images
Zoom Out
Crop

Input/Output Design

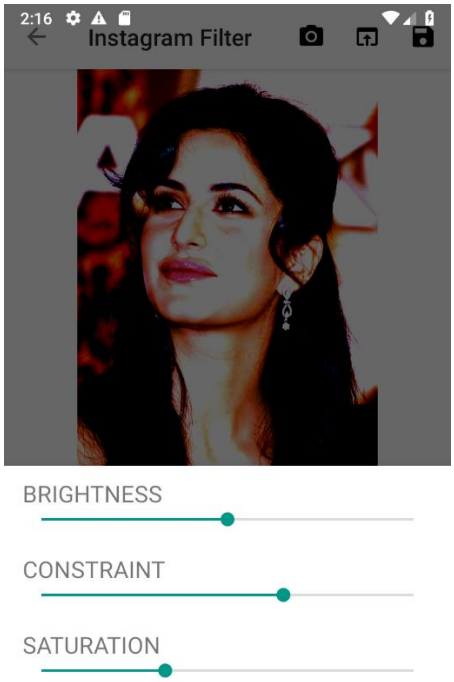
Instagram Filter App :-



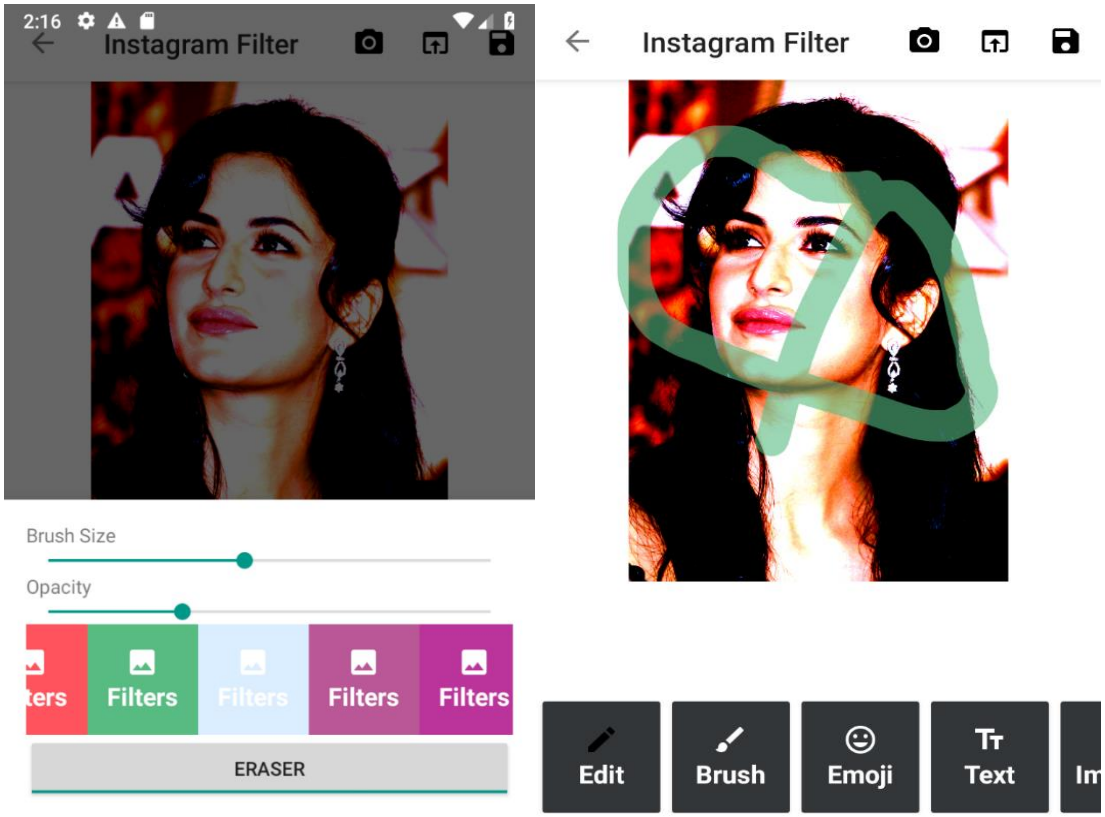
Filters :-



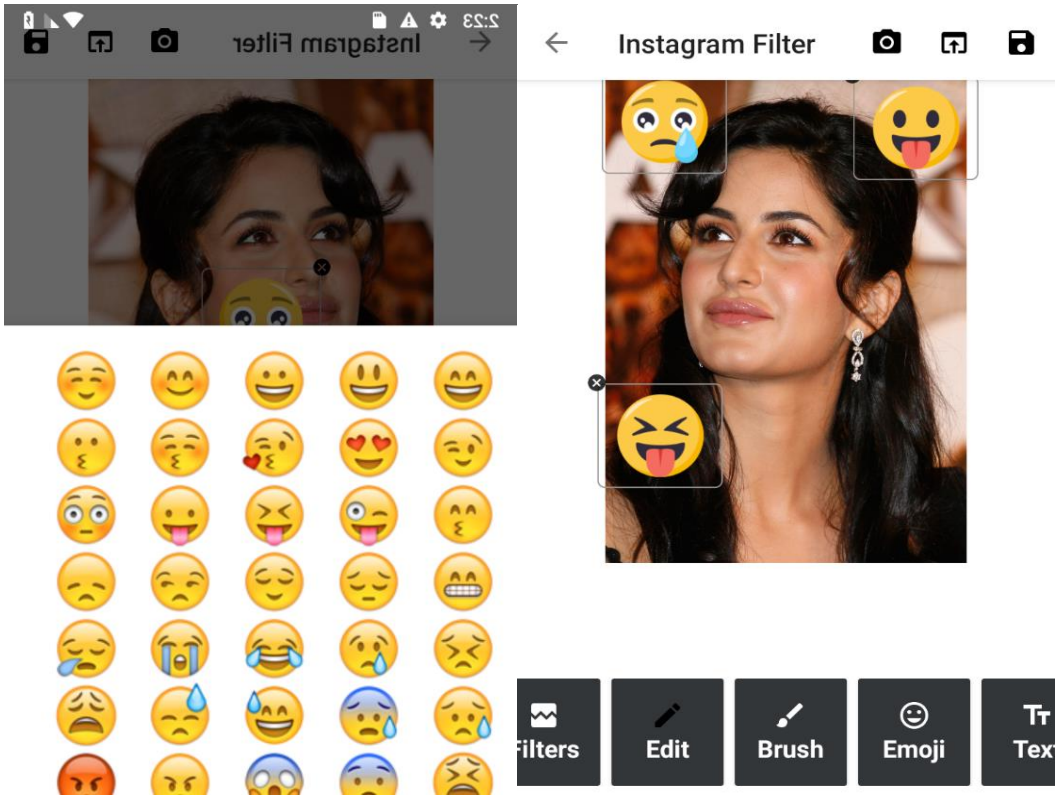
Brightness, Contrast, Saturation:-



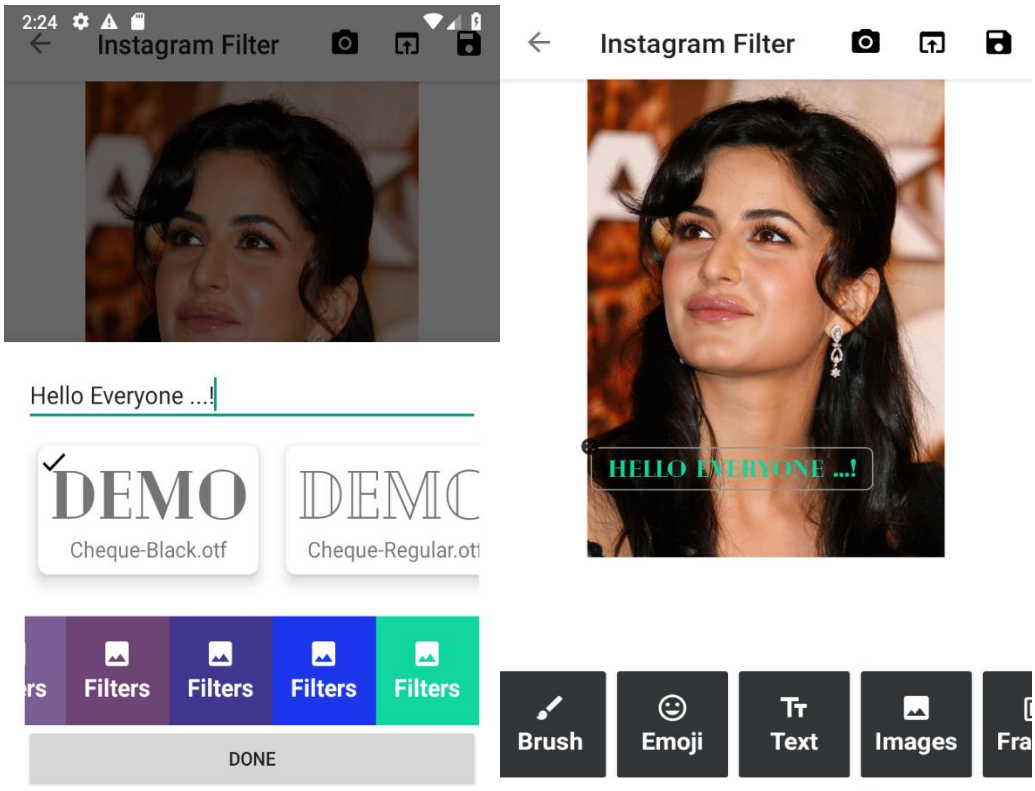
Brushes :-



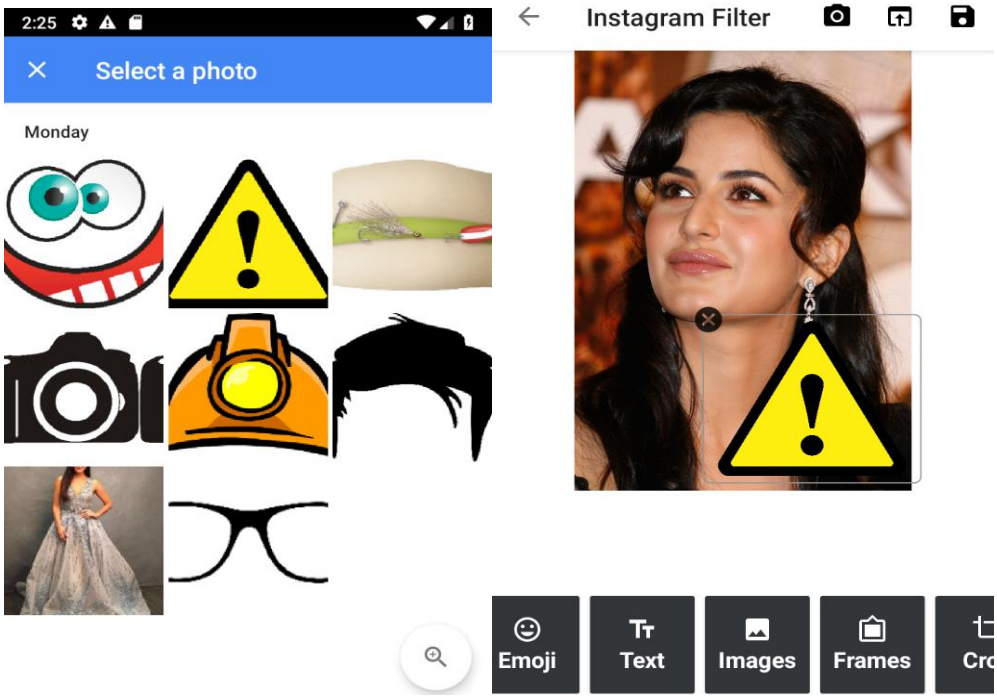
Emojies :-



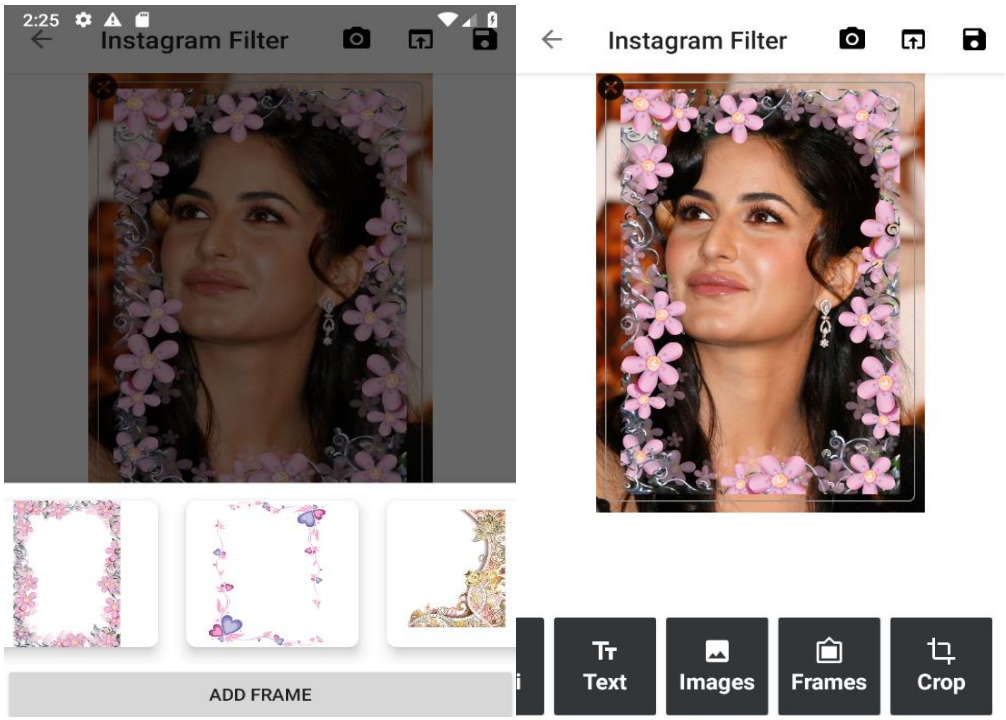
Adding Text :-



Add Imgaes From Gallary

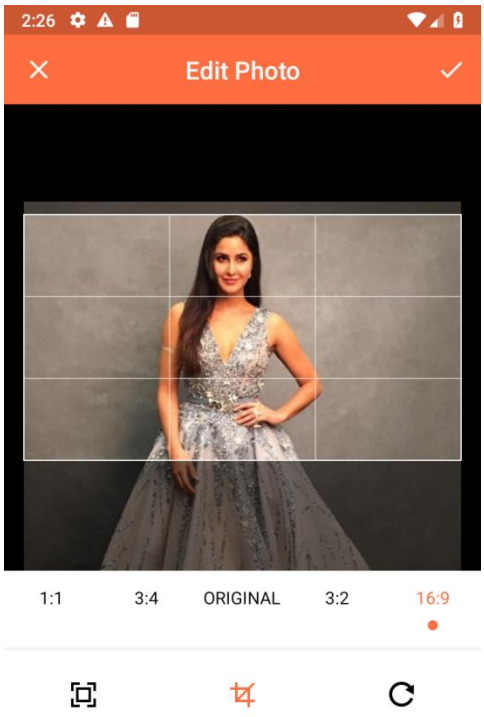


Add Frames :-

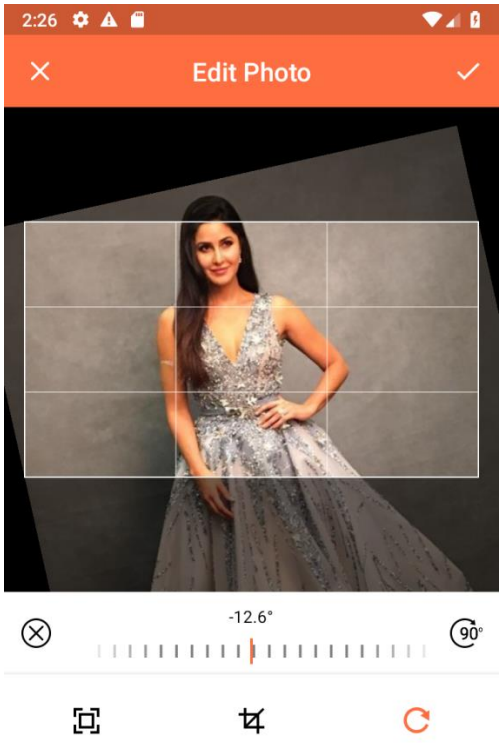


Crop image / Roate image / Zoom in :-

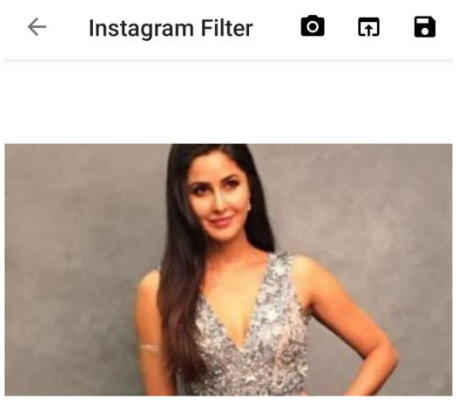
Crop :-



Rotate :-

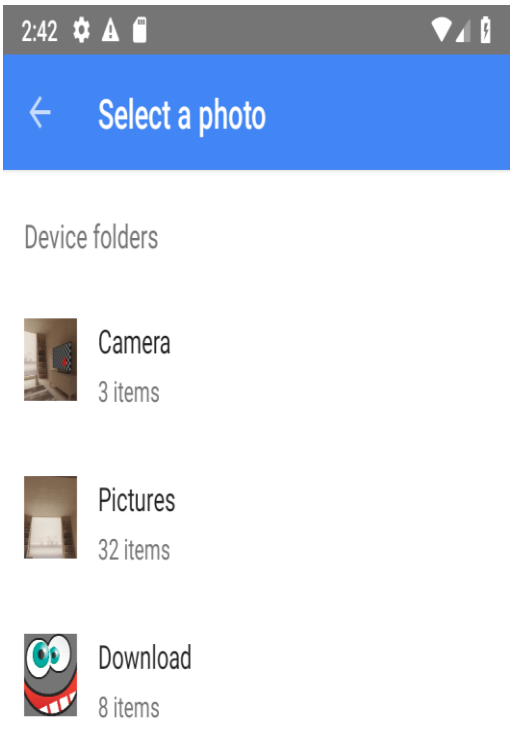


Zoom in :-



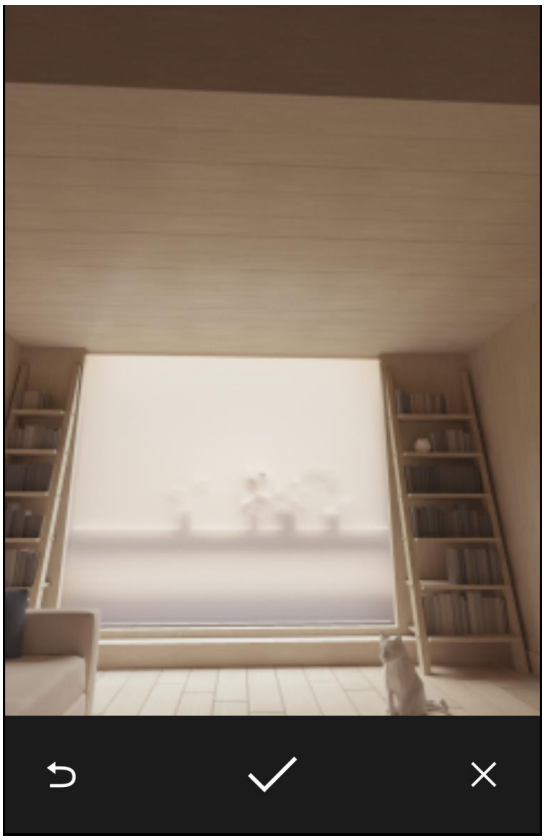
Save image to Gallery :-

Open image from Gallery :-

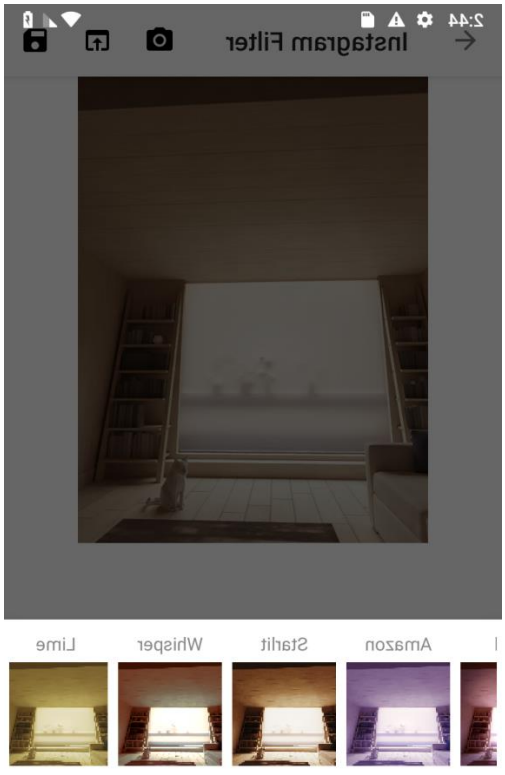


Camera :-

Save image to Gallery :-



Apply Filters :-



PROJECT DESCRIPTION

PROJECT DESCRIPTION

New Technologies of the images nowadays have drastically changed the way picture editors work. They do their research mostly on the internet and have to browse a never-ending flow of images. Photo editing is the changing of images.

These images can be digital photographs, illustrations, prints, etc. Photo editing is done for many reasons. Many of the models are used to remove blemishes and make the model better. This is usually called retouching, airbrushing or photo shopping even if Photoshop or airbrush are not used.

Photo editing is also used to make completely new images. Photo editing is sometimes called photo manipulation. In 2011, the first photo editing app was released on App Store. The most used Instagram filters are Normal, Clarendon, Juno, Ludwig, and Lark, etc.

The photo editor SDK is a powerful and multifaceted tool which enables you to equip your android applications with high performance photo editing capacity. The photo editor SDK is written in java and can easily be customized to entirely. The SDK large number of filters, covering all the art style and setting that can be previewed in real time.

Pacakages:

Zomato package:

implementation'info.androidhive:imagefilters:1.0.7'

Zomato, Andorid Hive. This library offers basic image operations like controlling brightness, saturation and contrast and few image filters. This app improvises the library and hosted it on a public maven repository so that we can integrate in our projects very easily.

This library is very basic and we can achieve great filters like Instagram using this. We have used number of features that currently Instagram doesn't consists of. So, the motive is the current Instagram can also have such kind of features that will help the users to create different styles of editing pictures.

Implementation 'com.android.support:recyclerview-v7:27.1.1'

Android dependency 'com.android.support:recyclerview-v7' has different version for the compile and runtime classpath. The recyclerView is a new ViewGroup that is prepared to render any adapter-based view in a similar way. It is supposed to be the successor of ListView and GridView and it can be found in the latest support-v7 version. One of the reason is that RecyclerView has more extensible framework, especially since it provides the ability to implement both horizontal and vertical layouts.

Implementation 'com.android.support:design:27.1.1'

Toolbar works well with apps targeted to API and above. However, Android has updated the AppCompat support libraries so the Toolbar can be used on lower Android OS devices as well.

Implementation 'com.karumi:dexter:4.1.0'

Dexter is an Android library that simplifies the process of requesting permissions at runtimeAndroid Marshmallow includes a new functionality to letusers grant or deny permissions running an app instead of granting them all when installing it. This approach gives the user more control over applications but require developers to add lots of code to support it.

Implementation 'ja.burhanrashid52:photoeditor:0.2.1'

Features: Drawing on image with option to change its brush's color, size, opacity and erasing

Applying Filter Effect on image on image using Media Effect

Adding/editing text with option to change its color with custom fonts.

FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

- This app can further be implemented to develop collage images
- Different font languages..
- Compress the large number of image files to send from one location to another.
- Can be also used to save images in different format like jpeg, png, GIF, etc.

Photo editing is also used to make completely new images. Photo editing is sometimes called photo manipulation. In 2011, the first photo editing app was released on App Store. The most used Instagram filters are Normal, Clarendon, Juno, Ludwig, and Lark, etc.

The photo editor SDK is a powerful and multifaceted tool which enables you to equip your android applications with high performance photo editing capacity. The photo editor SDK is written in java and can easily be customized to entirely. The SDK has large number of filters, covering all the art style and setting that can be previewed in real time.

CONCLUSION

CONCLUSION

In the ever shrinking world of Information Technology, our project is only a humble joint venture to satisfy a small part of the Image Processing. The system is highly flexible and can be modified to use in any photo studios & all of us .We have tried to make the system user friendly. Security is one main consideration in the project. The system is protected from any unauthorized access. We hope the entire objection to the system is rectified and the users will accept the system. There is no claim of this product being perfect, or anything near that. This is only a humble attempt made under trying circumstances. This system has been designed in an attractive manner. So that, even a user with minimum knowledge can operates the system easily.

The software is developed with scalability in mind. Additional modules can easily add when necessary. The software is developed with the modular approach. All modules in this system have been tested separately and put together to form the main system. Finally the system is tested with the real data and everything worked successfully. Thus the system has fulfilled all the objectives.

BIBLIOGRAPHY

BIBLOGRAPHY

References:
www.androidhive.com/

http://github.android.com/

http://developer.android.com

http://stackoverflow.com/

www.imagemagick.org

www.adobe.com

APPENDIX

MainActivity.java

```
package com.example.mca.androidinstagramfilters;

import android.Manifest;
import android.content.ContentValues;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Typeface;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.annotation.Nullable;
import android.support.design.widget.CoordinatorLayout;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.CardView;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

import com.example.mca.androidinstagramfilters.Adapter.ViewPagerAdapter;
import com.example.mca.androidinstagramfilters.Interface.AddFrameListener;
import com.example.mca.androidinstagramfilters.Interface.AddTextFragmentListener;
import com.example.mca.androidinstagramfilters.Interface.BrushFragmentListener;
import com.example.mca.androidinstagramfilters.Interface.EditImageFragmentListener;
import com.example.mca.androidinstagramfilters.Interface.EmojiFragmentListener;
import com.example.mca.androidinstagramfilters.Interface.FiltersListFragmentListener;
import com.example.mca.androidinstagramfilters.Utils.BitmapUtils;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;
import com.yalantis.ucrop.UCrop;
import com.zomato.photofilters.imageprocessors.Filter;
import com.zomato.photofilters.imageprocessors.subfilters.BrightnessSubFilter;
import com.zomato.photofilters.imageprocessors.subfilters.ContrastSubFilter;
```

```
import com.zomato.photofilters.imageprocessors.subfilters.SaturationSubfilter;

import java.io.File;
import java.io.IOException;
import java.util.List;
import java.util.UUID;

import ja.burhanrashid52.photoeditor.OnSaveBitmap;
import ja.burhanrashid52.photoeditor.PhotoEditor;
import ja.burhanrashid52.photoeditor.PhotoEditorView;

//here all the classes wil be initialize
public class MainActivity extends AppCompatActivity implements
FiltersListFragmentListener,EditImageFragmentListener, BrushFragmentListener,
EmojiFragmentListener, AddTextFragmentListener, AddFrameListener {
    public static final String pictureName = "flash.jpg";
    public static final int PERMISSION_PICK_IMAGE = 1000;
    public static final int PERMISSION_INSERT_IMAGE = 1001;
    private static final int CAMERA_REQUEST = 1002;

    PhotoEditorView photoEditorView;
    PhotoEditor photoEditor;
    CoordinatorLayout coordinatorLayout;

   // the final image after applying brightness, saturation, contrast
    Bitmap originalBitmap,filteredBitmap,finalBitmap;

    FiltersListFragment filtersListFragment;
    EditImageFragment editImageFragment;

    CardView
btn_filters_list,btn_edit,btn_brush,btn_emoji,btn_add_text,btn_add_image,btn_add_frame,bt
n_crop;

    // modified image values
    int brightnessFinal = 0;
    float saturationFinal = 1.0f;
    float constrantFinal = 1.0f;
    Uri image_selected_uri;
    //load native image filters library and is called to initialize native library
    static {
        System.loadLibrary("NativeImageProcessor");
    }

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

        Toolbar toolbar = findViewById(R.id.toolBar);
```

```
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Instagram Filter");
        //View
       photoEditorView = (PhotoEditorView) findViewById(R.id.image_preview);
       photoEditor = new PhotoEditor.Builder(this,photoEditorView)
              .setPinchTextScalable(true)
              .setDefaultEmojiTypeface(Typeface.createFromAsset(getAssets(),"emojione-
android.ttf"))
              .build();
        coordinatorLayout = (CoordinatorLayout) findViewById(R.id.coordinator);


        btn_edit = (CardView)findViewById(R.id.btn_edit);
        btn_filters_list = (CardView)findViewById(R.id.btn_filters_list);
        btn_brush = (CardView)findViewById(R.id.btn_brush);
        btn_emoji = (CardView)findViewById(R.id.btn_emoji);
        btn_add_text = (CardView)findViewById(R.id.btn_add_text);
        btn_add_image = (CardView)findViewById(R.id.btn_add_image);
        btn_add_frame = (CardView)findViewById(R.id.btn_add_frame);
        btn_crop = (CardView)findViewById(R.id.btn_crop);

        btn_crop.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
              startCrop(image_selected_uri);
           }
        });

        btn_filters_list.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
              if(filtersListFragment != null)
              {

filtersListFragment.show(getSupportFragmentManager(),filtersListFragment.getTag());
              }
              else
              {
                 FiltersListFragment filtersListFragment = FiltersListFragment.getInstance(null);
                 filtersListFragment.SetListener(MainActivity.this);

filtersListFragment.show(getSupportFragmentManager(),filtersListFragment.getTag());
              }
           }
        });

        btn_edit.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
```

```
            EditImageFragment editImageFragment = EditImageFragment.getInstance();
            editImageFragment.setListener(MainActivity.this);

editImageFragment.show(getSupportFragmentManager(),editImageFragment.getTag());
        }
    });

    btn_brush.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //Enable brush  mode

            photoEditor.setBrushDrawingMode(true);
            BrushFragment brushFragment = BrushFragment.getInstance();
            brushFragment.setListener(MainActivity.this);
            brushFragment.show(getSupportFragmentManager(),brushFragment.getTag());
        }
    });

    btn_emoji.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
         EmojiFragment emojiFragment = EmojiFragment.getInstance();
         emojiFragment.setListener(MainActivity.this);
         emojiFragment.show(getSupportFragmentManager(),emojiFragment.getTag());
        }
    });

    btn_add_text.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           AddTextFragment addTextFragment = AddTextFragment.getInstance();
            addTextFragment.setListener(MainActivity.this);

addTextFragment.show(getSupportFragmentManager(),addTextFragment.getTag());
        }
    });

    btn_add_image.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            addImageToPicture();

        }
    });

    btn_add_frame.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            FrameFragment frameFragment = FrameFragment.getInstance();
```

```
            frameFragment.setListener(MainActivity.this);
            frameFragment.show(getSupportFragmentManager(),frameFragment.getTag());
        }
      });
      loadImage();
   }

   private void startCrop(Uri uri) {
      String destinationFileName = new
StringBuilder(UUID.randomUUID().toString()).append(".jpg").toString();

      UCrop ucrop = UCrop.of(uri,Uri.fromFile(new
File(getCacheDir(),destinationFileName)));

      ucrop.start(MainActivity.this);
   }

   private void addImageToPicture() {
      Dexter.withActivity(this)
            .withPermissions(Manifest.permission.READ_EXTERNAL_STORAGE,
                  Manifest.permission.WRITE_EXTERNAL_STORAGE)
            .withListener(new MultiplePermissionsListener() {
               @Override
               public void onPermissionsChecked(MultiplePermissionsReport report) {
                  if(report.areAllPermissionsGranted())
                  {
                     Intent intent = new Intent(Intent.ACTION_PICK);
                     intent.setType("image/*");
                     startActivityForResult(intent,PERMISSION_INSERT_IMAGE);
                  }
               }

               @Override
               public void onPermissionRationaleShouldBeShown(List<PermissionRequest>
permissions, PermissionToken token) {
                     Toast.makeText(MainActivity.this,"Permission Denied",
Toast.LENGTH_SHORT).show();
               }
            }).check();
   }

   // load the default image from assets on app launch
   private void loadImage() {
      originalBitmap = BitmapUtils.getBitmapFromAssets(this,pictureName,300,300);
      filteredBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888,true);
      finalBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888,true);
     photoEditorView.getSource().setImageBitmap(originalBitmap);
   }

   private void setupViewPager(ViewPager viewPager) {
```

```
        ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());

        //adding filters class and edit class
        filtersListFragment = new FiltersListFragment();
        filtersListFragment.SetListener(this);

        editImageFragment = new EditImageFragment();
        editImageFragment.setListener(this);

        adapter.addFragment(filtersListFragment,"FILTERS");
        adapter.addFragment(editImageFragment,"EDIT");

        viewPager.setAdapter(adapter);
    }

    // dis methods are called when a seekbar value changed in editimagefragment
    @Override
    public void onBrightnessChanged(int brightness) {
        brightnessFinal = brightness;
        Filter myFilter = new Filter();
        myFilter.addSubFilter(new BrightnessSubFilter(brightness));

photoEditorView.getSource().setImageBitmap(myFilter.processFilter(finalBitmap.copy(Bitm
ap.Config.ARGB_8888,true)));
    }

    @Override
    public void onSaturationChanged(float saturation) {

        saturationFinal = saturation;
        Filter myFilter = new Filter();
        myFilter.addSubFilter(new SaturationSubfilter(saturation));

photoEditorView.getSource().setImageBitmap(myFilter.processFilter(finalBitmap.copy(Bitm
ap.Config.ARGB_8888,true)));
    }

    @Override
    public void onConstrantChanged(float constrant) {
        constrantFinal = constrant;
        Filter myFilter = new Filter();
        myFilter.addSubFilter(new SaturationSubfilter(constrant));

photoEditorView.getSource().setImageBitmap(myFilter.processFilter(finalBitmap.copy(Bitm
ap.Config.ARGB_8888,true)));
    }

    @Override
    public void onEditStarted() {
```

```
}

// once the editing is done i.e seekbar is drag is completed,
// apply the values on to filtered image
@Override
public void onEditCompleted() {
    Bitmap bitmap = filteredBitmap.copy(Bitmap.Config.ARGB_8888,true);

    Filter myFilter = new Filter();
    myFilter.addSubFilter(new BrightnessSubFilter(brightnessFinal));
    myFilter.addSubFilter(new SaturationSubfilter(saturationFinal));
    myFilter.addSubFilter(new ContrastSubFilter(constrantFinal));


    finalBitmap = myFilter.processFilter(bitmap);
}

@Override
public void onFilterSelected(Filter filter) {
    //resetControl();
    //applying the selected filter
    filteredBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888,true);
    // preview filtered image
    photoEditorView.getSource().setImageBitmap(filter.processFilter(filteredBitmap));
    finalBitmap = filteredBitmap.copy(Bitmap.Config.ARGB_8888,true);
}

//Resets image edit controls to normal when new filter is selected
private void resetControl() {
    if(editImageFragment != null)
        editImageFragment.resetControls();
    brightnessFinal=0;
    saturationFinal = 1.0f;
    constrantFinal = 1.0f;
}


@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main,menu);
    return true;

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if(id == R.id.action_open)
    {
        openImageFromGallery();
```

```
        return true;
    }
    else if(id == R.id.action_save)
    {
        saveImageToGallery();
        return true;
    }
    else if(id == R.id.action_camera)
    {
        openCamera();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

private void openCamera() {
    Dexter.withActivity(this)
            .withPermissions(Manifest.permission.CAMERA,
                    Manifest.permission.WRITE_EXTERNAL_STORAGE)
            .withListener(new MultiplePermissionsListener() {
                @Override
                public void onPermissionsChecked(MultiplePermissionsReport report) {
                    if(report.areAllPermissionsGranted())
                    {
                        ContentValues values = new ContentValues();
                        values.put(MediaStore.Images.Media.TITLE,"New Picture");
                        values.put(MediaStore.Images.Media.DESCRIPTION,"From Camera");
                        image_selected_uri =
getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
                                values);
                        Intent cameraIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,image_selected_uri);
                        startActivityForResult(cameraIntent,CAMERA_REQUEST);
                    }
                    else
                    {
                        Toast.makeText(MainActivity.this, "Permission Denied",
Toast.LENGTH_SHORT).show();
                    }
                }

                @Override
                public void onPermissionRationaleShouldBeShown(List<PermissionRequest>
permissions, PermissionToken token) {
                    token.continuePermissionRequest();


                }
            })
```

```
            .check();

    }

    private void saveImageToGallery() {

        Dexter.withActivity(this)
                .withPermissions(Manifest.permission.READ_EXTERNAL_STORAGE,
                        Manifest.permission.WRITE_EXTERNAL_STORAGE)
                .withListener(new MultiplePermissionsListener() {
                    @Override
                    public void onPermissionsChecked(MultiplePermissionsReport report) {
                        if(report.areAllPermissionsGranted())
                        {
                            photoEditor.saveAsBitmap(new OnSaveBitmap() {
                                @Override
                                public void onBitmapReady(Bitmap saveBitmap) {
                                    try {

                                     photoEditorView.getSource().setImageBitmap(saveBitmap);
                                      final String path = BitmapUtils.insertImage(getContentResolver(),
                                       saveBitmap,
                                        System.currentTimeMillis()+"_profile.jpg",
                                        null);
                                        if(!TextUtils.isEmpty(path))
                                        {
                                          Snackbar snackbar = Snackbar.make(coordinatorLayout,
                                           "Image saved to gallery!",
                                            Snackbar.LENGTH_LONG)
                                             .setAction("OPEN", new View.OnClickListener() {
                                               @Override
                                               public void onClick(View view) {
                                                        openImage(path);
                                                    }
                                                });
                                            snackbar.show();
                                        }
                                        else
                                        {
                                            Snackbar snackbar = Snackbar.make(coordinatorLayout,
                                                    "Unable to save image!",
                                                    Snackbar.LENGTH_LONG);
                                            snackbar.show();
                                        }
                                    } catch (IOException e) {
                                        e.printStackTrace();
                                    }
                                }

                                @Override
```

```
                    public void onFailure(Exception e) {

                    }
                });
            }
            else
            {
                Toast.makeText(MainActivity.this, "Permission Denied",
Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onPermissionRationaleShouldBeShown(List<PermissionRequest>
permissions, PermissionToken token) {
            token.continuePermissionRequest();


        }
    })
    .check();


}
// opening image in default image viewer app
private void openImage(String path) {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    intent.setDataAndType(Uri.parse(path), "image/*");
    startActivity(intent);
}


private void openImageFromGallery() {
    Dexter.withActivity(this)
        .withPermissions(Manifest.permission.READ_EXTERNAL_STORAGE,
            Manifest.permission.WRITE_EXTERNAL_STORAGE)
        .withListener(new MultiplePermissionsListener() {

            @Override
            public void onPermissionsChecked(MultiplePermissionsReport report) {
                if(report.areAllPermissionsGranted())
                {
                    Intent intent = new Intent(Intent.ACTION_PICK);
                    intent.setType("image/*");
                    startActivityForResult(intent,PERMISSION_PICK_IMAGE);
                }
                else
                {
                    Toast.makeText(MainActivity.this,"Permission denied",
```

```
Toast.LENGTH_SHORT).show();
                    }

                }

                @Override
                public void onPermissionRationaleShouldBeShown(List<PermissionRequest>
permissions, PermissionToken token) {

                    token.continuePermissionRequest();
                }

            })
            .check();

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        if(resultCode == RESULT_OK)
        {
            if( requestCode == PERMISSION_PICK_IMAGE) {
                Bitmap bitmap = BitmapUtils.getBitmapFromGallery(this, data.getData(), 800,
800);

                image_selected_uri = data.getData();
                //clear bitmap gmemory
                originalBitmap.recycle();
                finalBitmap.recycle();
                filteredBitmap.recycle();

                originalBitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
                finalBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888, true);
                filteredBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888, true);
                photoEditorView.getSource().setImageBitmap(originalBitmap);
                bitmap.recycle();


                //fix crash
                filtersListFragment = FiltersListFragment.getInstance(originalBitmap);
                filtersListFragment.SetListener(this);
            }
            if( requestCode == CAMERA_REQUEST) {
                Bitmap bitmap = BitmapUtils.getBitmapFromGallery(this, image_selected_uri,
800, 800);

                //clear bitmap gmemory
                originalBitmap.recycle();
                finalBitmap.recycle();
                filteredBitmap.recycle();
```

```
            originalBitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
            finalBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888, true);
            filteredBitmap = originalBitmap.copy(Bitmap.Config.ARGB_8888, true);
            photoEditorView.getSource().setImageBitmap(originalBitmap);
            bitmap.recycle();


            //fix crash
            filtersListFragment = FiltersListFragment.getInstance(originalBitmap);
            filtersListFragment.SetListener(this);
        }
        else   if( requestCode == PERMISSION_INSERT_IMAGE)
        {
            Bitmap bitmap = BitmapUtils.getBitmapFromGallery(this,data.getData(),200,250);
            photoEditor.addImage(bitmap);
        }
        else  if(requestCode == UCrop.REQUEST_CROP)
            handleCropResult(data);
    }
    else if(resultCode == UCrop.RESULT_ERROR)
        handleCropError(data);
}

private void handleCropError(Intent data) {
    final Throwable cropError = UCrop.getError(data);
    if(cropError != null)
    {
        Toast.makeText(this,""+cropError.getMessage(), Toast.LENGTH_SHORT).show();
    }
    else
    {
        Toast.makeText(this,"Unexpected Error", Toast.LENGTH_SHORT).show();
    }
}

private void handleCropResult(Intent data) {
    final Uri resultUri = UCrop.getOutput(data);
    if(resultUri != null) {
        photoEditorView.getSource().setImageURI(resultUri);
        //Fix error return original images after crop and use Filters
        Bitmap bitmap =
((BitmapDrawable)photoEditorView.getSource().getDrawable()).getBitmap();
        originalBitmap = bitmap.copy(Bitmap.Config.ARGB_8888,true);
        filteredBitmap = originalBitmap;
        finalBitmap = originalBitmap;
    }
    else
        Toast.makeText(this,"Cannot retrieve crop image", Toast.LENGTH_SHORT).show();
}
```

```
    @Override
    public void onBrushSizeChangedListener(float size) {
        photoEditor.setBrushSize(size);
    }

    @Override
    public void onBrushOpacityChangedListener(int opacity) {
        photoEditor.setOpacity(opacity);
    }

    @Override
    public void onBrushColorChangedListener(int color) {
       photoEditor.setBrushColor(color);
    }

    @Override
    public void onBrushStateChangedListener(boolean isEraser) {
        if(isEraser)
            photoEditor.brushEraser();
        else
           photoEditor.setBrushDrawingMode(true);
    }

    @Override
    public void onEmojiSelected(String emoji) {
        photoEditor.addEmoji(emoji);
    }


    @Override
    public void onAddTextButtonClick(Typeface typeface, String text, int color) {
        photoEditor.addText(typeface,text,color);
    }

    @Override
    public void onAddFrame(int frame) {
        Bitmap bitmap = BitmapFactory.decodeResource(getResources(),frame);
        photoEditor.addImage(bitmap);
    }
}
```

FiltersListFragment.java

```
package com.example.mca.androidinstagramfilters;


import android.graphics.Bitmap;
import android.os.Bundle;
```

```
import android.support.annotation.Nullable;
import android.support.design.widget.BottomSheetDialogFragment;
import android.support.v4.app.Fragment;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.example.mca.androidinstagramfilters.Adapter.ThumbnailAdapter;
import com.example.mca.androidinstagramfilters.Interface.FiltersListFragmentListener;
import com.example.mca.androidinstagramfilters.Utils.BitmapUtils;
import com.example.mca.androidinstagramfilters.Utils.SpacesItemDecoration;
import com.zomato.photofilters.FilterPack;
import com.zomato.photofilters.imageprocessors.Filter;
import com.zomato.photofilters.utils.ThumbnailItem;
import com.zomato.photofilters.utils.ThumbnailsManager;

import java.util.ArrayList;
import java.util.List;


/**
 * A simple {@link Fragment} subclass.
 */
public class FiltersListFragment extends BottomSheetDialogFragment implements
FiltersListFragmentListener {
    RecyclerView recyclerView;
    ThumbnailAdapter adapter;
    List<ThumbnailItem> thumbnailItems;

    FiltersListFragmentListener listener;

   static FiltersListFragment instance;
   static Bitmap bitmap;

    public static FiltersListFragment getInstance(Bitmap bitmapSave) {
        bitmap = bitmapSave;
        if(instance == null) {
            instance = new FiltersListFragment();

        }
        return instance;
    }

    public void SetListener(FiltersListFragmentListener listener)
    { // this method provides callback method to mainactivity wnv new filter is selected
        this.listener = listener;
```

```
    }

    public FiltersListFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View itemView = inflater.inflate(R.layout.fragment_filters_list, container, false);
        thumbnailItems = new ArrayList<>();
        adapter = new ThumbnailAdapter(thumbnailItems, this, getActivity());


        recyclerView = (RecyclerView) itemView.findViewById(R.id.recycler_view);
        recyclerView.setLayoutManager(new
LinearLayoutManager(getActivity(),LinearLayoutManager.HORIZONTAL,false));
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        int space = (int)
TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP,8,getResources().getDispl
ayMetrics());
        recyclerView.addItemDecoration(new SpacesItemDecoration(space));
        recyclerView.setAdapter(adapter);

        displayThumbnail(bitmap);
        return itemView;
    }

    public void displayThumbnail(final Bitmap bitmap) {
        Runnable r = new Runnable() {
            @Override
            public void run() {
                Bitmap thumbImg;
                if (bitmap == null)

                    thumbImg = BitmapUtils.getBitmapFromAssets(getActivity(),
MainActivity.pictureName, 100, 100);

                else

                    thumbImg = Bitmap.createScaledBitmap(bitmap, 100, 100, false);

                if(thumbImg == null)
```

```
            return;
        ThumbnailsManager.clearThumbs();
        thumbnailItems.clear();

        // add normal bitmap first
        ThumbnailItem thumbnailItem = new ThumbnailItem();
        thumbnailItem.image = thumbImg;
        thumbnailItem.filterName = "Normal";
        ThumbnailsManager.addThumb(thumbnailItem);

        List<Filter> filters = FilterPack.getFilterPack(getActivity()); // provides list of all
filters from liabrary

        for (Filter filter : filters) {
            ThumbnailItem tI = new ThumbnailItem(); //object
            tI.image = thumbImg; //image
            tI.filter = filter; //filtertype
            tI.filterName = filter.getName();//filtername
            ThumbnailsManager.addThumb(tI); // manager will handle this
        }

        thumbnailItems.addAll(ThumbnailsManager.processThumbs(getActivity()));
        //each thumbnail item is added to ThumbnailManager to manage dem, this process
thumbnails are added back to thumbnailitem

        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                adapter.notifyDataSetChanged(); // all this done in background thread and we
shouldnot block the main thread
            }
        });

    }

};
new Thread(r).start();
}


@Override
public void onFilterSelected(Filter filter) {
    if(listener != null)

        listener.onFilterSelected(filter);

}
}
```

EditImageFragment.java

```
package com.example.mca.androidinstagramfilters;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.BottomSheetDialogFragment;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SeekBar;

import com.example.mca.androidinstagramfilters.Interface.EditImageFragmentListener;


/**
 * A simple {@link Fragment} subclass.
 */

//this fragment is used to display the image controls like b,s,c
public class EditImageFragment extends BottomSheetDialogFragment implements
SeekBar.OnSeekBarChangeListener {

    private EditImageFragmentListener listener;
    SeekBar seekbar_brightness,seekbar_constrant,seekbar_saturation;
    //seekbar widget is used to control b,s,c


    //listner
    public void  setListener(EditImageFragmentListener listener)
    {
        this.listener = listener;
    }

    static EditImageFragment instance;

    public static EditImageFragment getInstance() {
        if(instance == null)
            instance = new EditImageFragment();
        return instance;
    }


    public EditImageFragment() {
        // Required empty public constructor
    }
```

```
@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View itemView = inflater.inflate(R.layout.fragment_edit_image, container, false);

    seekbar_brightness = (SeekBar)itemView.findViewById(R.id.seekbar_brightness);
    seekbar_constrant = (SeekBar)itemView.findViewById(R.id.seekbar_constraint);
    seekbar_saturation = (SeekBar)itemView.findViewById(R.id.seekbar_saturation);

    //b values can be btwn -100 to +100
    seekbar_brightness.setMax(200);
    seekbar_brightness.setProgress(100);

    //contrast  and saturation in the form of float values
    // keeping contrast value b/w 1.0 - 3.0
    seekbar_constrant.setMax(20);
    seekbar_constrant.setProgress(0);

    // keeping saturation value b/w 0.0 - 3.0
    seekbar_saturation.setMax(30);
    seekbar_saturation.setProgress(10);

    seekbar_saturation.setOnSeekBarChangeListener(this);
    seekbar_constrant.setOnSeekBarChangeListener(this);
    seekbar_brightness.setOnSeekBarChangeListener(this);

    return itemView;
}

@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
    if(listener != null)
    {
        if(seekBar.getId() == R.id.seekbar_brightness)
        {
            // brightness values are b/w -100 to +100
            listener.onBrightnessChanged(progress-100);
        }
        else if(seekBar.getId() == R.id.seekbar_constraint)
        {
            // converting int value to float
            // contrast values are b/w 1.0f - 3.0f
            // progress = progress > 10 ? progress : 10;
            progress+=10;
```

```
            float value = .10f*progress;
            listener.onConstrantChanged(value);
        }
        else if(seekBar.getId() == R.id.seekbar_saturation)
        {
            // converting int value to float
            // saturation values are b/w 0.0f - 3.0f
            float value = .10f*progress;
            listener.onSaturationChanged(value);
        }

    }
  }

  @Override
  public void onStartTrackingTouch(SeekBar seekBar) {
    if(listener!=null)
        listener.onEditStarted();

  }

  @Override
  public void onStopTrackingTouch(SeekBar seekBar) {
    if(listener != null)
        listener.onEditCompleted();
  }

  public void resetControls()
  {
    seekbar_brightness.setProgress(100);
    seekbar_constrant.setProgress(0);
    seekbar_saturation.setProgress(10);

  }
}
```

Activity-Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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/coordinator"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

  <android.support.design.widget.AppBarLayout
```

```
    android:theme="@style/AppTheme.AppBarOverlay"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.v7.widget.Toolbar
      android:id="@+id/toolBar"
      android:background="@android:color/white"
      app:popupTheme="@style/AppTheme.PopupOverlay"
      android:layout_width="match_parent"
      android:layout_height="?attr/actionBarSize">


    </android.support.v7.widget.Toolbar>

  </android.support.design.widget.AppBarLayout>

   <include layout="@layout/content_main"/>

</android.support.design.widget.CoordinatorLayout>
```

Fragment-Edit-Image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:gravity="center_vertical"
   android:orientation="vertical"
   android:paddingLeft="16dp"
   android:paddingRight="16dp"
   tools:context=".EditImageFragment">

   <LinearLayout
      android:orientation="vertical"
      android:paddingBottom="10dp"
      android:paddingTop="10dp"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <TextView
         android:text="BRIGHTNESS"
         android:textSize="20sp"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content" />
      <SeekBar
         android:id="@+id/seekbar_brightness"
         android:layout_width="match_parent"
         android:layout_height="wrap_content" />
```

```
    </LinearLayout>
    <LinearLayout
        android:orientation="vertical"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:text="CONSTRAINT"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <SeekBar
            android:id="@+id/seekbar_constraint"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />


    </LinearLayout>

    <LinearLayout
        android:orientation="vertical"
        android:paddingBottom="10dp"
        android:paddingTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:text="SATURATION"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <SeekBar
            android:id="@+id/seekbar_saturation"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />


    </LinearLayout>


</LinearLayout>
```

Content-Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
   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:layout_width="match_parent"
   android:background="@android:color/white"
   android:orientation="vertical"
   app:layout_behavior="@string/appbar_scrolling_view_behavior"
   android:layout_height="match_parent"
   tools:context=".MainActivity"
   tools:showIn="@layout/activity_main">

   <ja.burhanrashid52.photoeditor.PhotoEditorView
      android:id="@+id/image_preview"
      android:scaleType="centerCrop"
      android:layout_width="match_parent"
      android:layout_height="360dp" />

   <HorizontalScrollView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_alignParentStart="true"
      android:fillViewport="true"
      android:measureAllChildren="false"
      android:scrollbars="none">

      <LinearLayout
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:orientation="horizontal"
         android:padding="10dp">

         <android.support.v7.widget.CardView
            android:id="@+id/btn_filters_list"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:padding="16dp"
            app:cardBackgroundColor="@color/card_background_color">


            <LinearLayout
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_gravity="center_vertical|center_horizontal"
               android:orientation="vertical">


               <ImageView
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_broken_image_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Filters"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_edit"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_edit_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Edit"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />
```

```
    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_brush"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_brush_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Brush"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_emoji"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_insert_emoticon_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Emoji"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_add_text"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_text_fields_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Text"
```

```
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_add_image"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_image_white_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Images"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_add_frame"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">
```

```
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_filter_frames_black_24dp" />


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Frames"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/btn_crop"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:cardBackgroundColor="@color/card_background_color">


    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical|center_horizontal"
        android:orientation="vertical">


        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/ic_crop_black_24dp" />
```

```
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Crop"
                android:textAlignment="center"
                android:textColor="@android:color/white"
                android:textSize="18sp"
                android:textStyle="bold" />

        </LinearLayout>
    </android.support.v7.widget.CardView>

</LinearLayout>

</HorizontalScrollView>

</RelativeLayout>
```