

PROJECT TITLE : A mobile based photo editing application with all standard image editing features

Under the Supervision of Mr. Neeraj Khanna

By

:- Nikhil Mishra(181500426)

:- Ved Prakash Chaubey(181500785)

ACKNOWLEDGEMENT

It gives me immense pleasure in presenting project report on the topic **A mobile based photo editing application with all standard image editing features** . Apart from the efforts of me, the success of my project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to my project in-charge, Mr. Neeraj Khanna. I can't say thank you enough for the tremendous support and help. I feel motivated and encouraged every time I attend her meeting. Without her encouragement and guidance this project work would not have materialized.

Date: 25/11/2020

:- Nikhil Mishra(181500426)

:- Ved Prakash Chaubey(181500785)

ABSTRACT

Photo editing can be a challenging task, and it becomes even more difficult on the small, portable screens of mobile devices that are now frequently used to capture and edit images. To address this problem I present Photo Editor, a photo editing interface for direct manipulation. Through this application user can easily and quickly edit there pictures with the help of the features provided in the application. Some of the features of the application are: - One tap Auto Enhance, Ability to Crop, rotate and straighten your photo, Adjust brightness, contrast and saturation, adding effects like blur, snowy, emboss, engrave, etc. All the coding has been done in JAVA language using a plugin of Android IDE i.e. Android Development Tools (ADT).

ADT is designed to provide an integrated environment in which to build Android applications. ADT extends the capabilities to let developers set up new Android projects, create an application UI, add packages based on the Android Framework API, debug their applications using the Android SDK tools, and export signed (or unsigned) .apk files in order to distribute their applications. It is a freeware available to download. Through this software we can run the code either directly in our android device or by using AVD (Android virtual device) manager to create an AVD.

INTRODUCTION

The project “PHOTO EDITING ANDROID APPLICATION” is developed using Android IDE and Android SDK manager using JAVA language. This project has been developed for busy customers, Photo Editor is an easy to use Android application through which customers can easily edit photos using various features provided in the application. Its graphical user interface is designed in a manner to attract wide variety of people varying from age group of 15 years to 55 years. The editing methods are optimized in such a way so that user can quickly and easily edit their photographs without wasting much of their time. Hence it's quick, easy and efficient

Problem Statement

To develop a Photo editing Android application with a quick and easy way to edit photos having both basic and advance level features to edit your photographs.

A lot of photo editing, applications which work either online or offline and both, a lot of applications are available which is system dependent they do not work different systems and a lot of photo editing application is available which is provide an edited photograph with watermark or logo. Brightness has always been an issue, most often pictures turn out darker than expected, and correcting brightness can be an issue. Lighting as monitoring is very important to highlight the product in an image. Brightness correction along with enhancing and manipulating is done in Photo editing services. Some of the other services are red-eye removal, perspective correction, removing lens distortion, and adding special effects. While color enhancement works on many photographs, but a lot of photo editing applications are not adjusted to clean and over the whole photograph.

There are also a number of online photo editing applications tools that allow you to upload a photograph and perform a number of artistic manipulations on the image, including cropping, adding borders, adding text, changing the background, adding objects to the background, etc, but all the photo editing applications can't do essay work for photo editing, user required more knowledge to do a better photograph with the help of the photo editing applications.

WORKING ENVIRONMENT

About Android:-

Android is an open source operating system, created by Google, and available to all kinds of developers with various expertise levels, ranging from rookie to professional. From a developer's perspective, Android is a Linux-based operating system for smartphones and tablets. It includes a touch screen user interface, widgets, camera, network data monitoring and all the other features that enable a cell phone to be called a smartphone. Android is a platform that supports various applications, available through the Android Play Store. The Android platform also allows end users to develop, install and use their own applications on top of the Android framework. The Android framework is licensed under the Apache License, with Android application developers holding the right to distribute their applications under their customized license.

Understanding android

To begin development on Android even at the application level, it is paramount to understand the basic internal architecture. Knowing how things are arranged inside helps us understand the application framework better, so we can design the application in a better way. Android is an OS based on Linux. Hence, deep inside, Android is pretty similar to Linux. To understand Android internals, let us look at an architectural diagram.

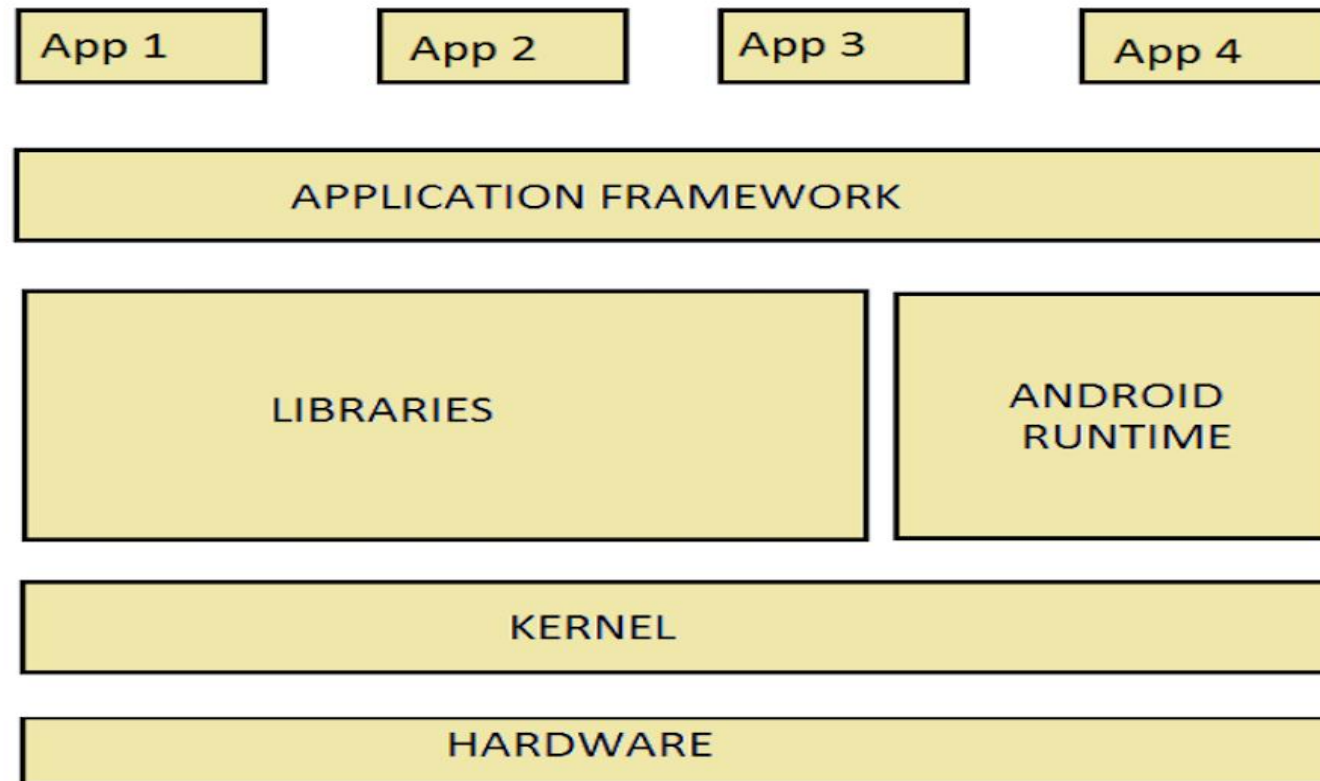


Fig 1.1

The above diagram illustrates the Android architecture. As you can see, it is a software stack above the hardware that is provided by the OEMs. Let's start with the topmost layer, i.e., the applications.

Applications :-

The diagram shows four basic apps (App 1, App 2, App 3 and App 4), just to give the idea that there can be multiple apps sitting on top of Android. These apps are like any user interface you use on Android; for example, when you use a music player, the GUI on which there are buttons to play, pause, seek, etc is an application. Similarly, is an app for making calls, a camera app, and so on. All these apps are not necessarily from Google. Anyone can develop an app and make it available to everyone through Google Play Store. These apps are developed in Java, and are installed directly, without the need to integrate with Android OS.

Libraries:-

This layer holds the Android native libraries. These libraries are written in C/C++, java and kotlin offer capabilities similar to the above layer, while sitting on top of the kernel. A few of the major native libraries include

- ❑ Surface Manager: manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.
- ❑ System C Libraries: Standard C library like libc targeted for ARM or embedded devices.
- ❑ OpenGL ES Libraries : These are the graphics libraries for rendering 2D and 3D graphics.
- ❑ SQLite : A database engine for Android.

Android Runtime :-

The Android runtime consists of the Dalvik Virtual Machine. It is basically a virtual machine for embedded devices, which like any other virtual machine, is a bytecode interpreter. When we say it is for embedded devices, it means it is low on memory, comparatively slower and runs on battery power. Besides the Dalvik Virtual Machine, it also consists of the core libraries, which are Java libraries and are available for all devices.

Kernel :-

The Android OS is derived from Linux **Kernel 5.7** and is actually created from Linux source, compiled for mobile devices. The memory management, process management etc. are mostly similar. The kernel acts as a Hardware Abstraction Layer between hardware and the Android software stack

- New ARM Features And Device Support. v5.7 introduces several new enhancements to 64-bit ARM architecture such as ARM Activity Monitors (AMU) extension support and in-kernel pointer authentication which was earlier restricted to userspace.
- Better HDR/OLED Display Support. For AMDGPU Linux driver that deals with modern OLED and HDR displays, Linux 5.
- Improved Linux I/O Interface IO_uring. IO_uring is a new high-performance I/O interface for Linux that was introduced with the release of Linux 5.

LITERATURE SURVEY

In order to fully understand the impact of this photo viewer and editor in an industrial scenario, it is important to first take a back-seat and ensure what a photo editor conjures up with, by having feasibility study and evaluating various scenarios, and why it should be implemented and from what perspective it will be accountable? Lastly, the conceptual literature review will focus on how an innovative and practical application can be implemented alongside the pre-existing applications in the modern trends of an industrial scenario, keeping in mind both the subject world and the usage world. The facets that this photo editor deals with to innovate in designing this application which would help us in image viewing and editing. The existence of this very model will furnish individuals with a sense of confidence to experiment and examine with various facets and genres of image viewing from the app store perspective along with image editing.

Terminology and Terms Used

Image Rearrangement :-

Image rearrangement consists of moving an object to a new image location, or deleting part of the image, while keeping some of the content of the image unchanged. The user selects a region to move, and specifies the location at which the selected region will be placed. A new image is generated satisfying this constraint. This application was demonstrated and gave impressive results in many cases. In image rearrangement pixels can be relocated by a large displacement, creating a possible computational complexity

Image Retargeting :-

Image retargeting is the change of image size, which is typically done in only a single direction in order to change the image aspect ratio. We will assume that the change is in image width, but we could also address changing both image dimensions.

Methodology

A good interface can be understood by the user. after being observed, it can be investigated how to construct a workflow for creating high quality image editing on mobile application. The way is quite simple, users just open a photo, then edits and save.

Instructions

This activity is meant to introduce you to your software editing program in a more in-depth way. At the end, you should be able to easily repeat the steps necessary to open up a given tool at any time.

1. Open your photo editing Application program.
2. Open up a photo file (it's easier to open up tools when you have a photo loaded into your program).
3. With your photo editing journal open beside you, find and open each of the following tools. For each tool, write down the path of how you got there. For instance, the best way to do it is like this: "Brightness: Edit> Adjust> Brightness", etc.

Hardware & Software to be used

Hardware:

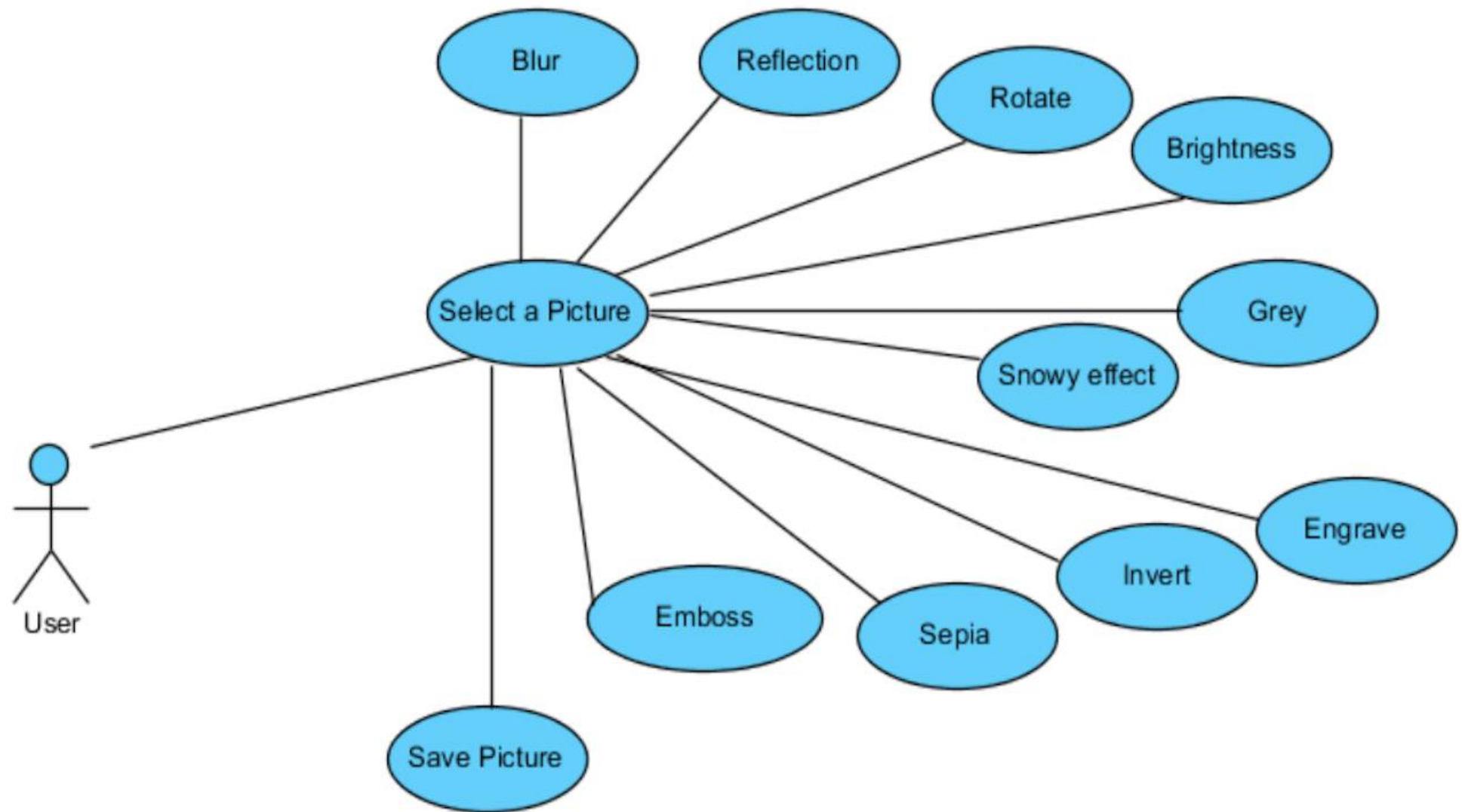
- 1vCPU/cores, 2 GB RAM
- **JVM (is not a real hardware machine but a software layer which resembles a hardware platform.)**

Software:

1. Android devices with Android 7+
2. Java Coding/Decoding Environment/Compiler
3. Android Studio

Use-Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the case and will often be accompanied by other types of diagrams as well.




```
object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier.name))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exact")
```

```
-- OPERATOR CLASSES -----
```

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

IMPLEMENT ATION

PhotoEditor18

app

src

main

AndroidManifest.xml

Project

PhotoEditor18

~/.Downloads/PhotoEditor18

.gitignore

.idea

app

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

ic_photo_filter.xml

ic_redo.xml

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.nikhilved.imageeditor">
4
5     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
6     <uses-permission android:name="android.permission.INTERNET" />
7
8     <application
9         android:name="com.nikhilved.imageeditor.PhotoApp"
10        android:allowBackup="true"
11        android:icon="@mipmap/ic_launcher"
12        android:label="@string/app_name"
13        android:roundIcon="@mipmap/ic_launcher_round"
14        android:supportsRtl="true"
15        android:theme="@style/AppTheme">
16        <activity
17            android:name="com.nikhilved.imageeditor.EditImageActivity"
18            android:screenOrientation="portrait"
19            android:theme="@style/AppTheme.NoActionBar">
20            <intent-filter>
21                <action android:name="android.intent.action.MAIN" />
22
23                <category android:name="android.intent.category.LAUNCHER" />
24            </intent-filter>
25        </activity>
26        <meta-data
27            android:name="preloaded_fonts"
28            android:resource="@array/preloaded_fonts" />
29    </application>
30
31 </manifest>
32
```

manifest > application

Text Merged Manifest

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

Gradle sync finished in 6 s 657 ms (from cached state)

29:1 LF UTF-8 4 spaces Git: master

app

src

main

java

com

nikhilved

1: Project

Resource Manager

2: Structure

Build Variants

2: Favorites

PhotoEditor18

~/Downloads/PhotoEditor18

.gitignore

.idea

app

.gitignore

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

BaseActivity

BaseFragment

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

AndroidManifest.xml

BaseActivity.java

1

2

3

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

```
package com.nikhilved.imageeditor.base;

import ...

public class BaseActivity extends AppCompatActivity {

    public static final int READ_WRITE_STORAGE = 52;
    private ProgressDialog mProgressDialog;

    public boolean requestPermission(String permission) {
        boolean isGranted = ContextCompat.checkSelfPermission(this, permission) == PackageManager.PERMISSION_GRANTED;
        if (!isGranted) {
            ActivityCompat.requestPermissions(
                this,
                new String[]{permission},
                READ_WRITE_STORAGE);
        }
        return isGranted;
    }

    public void isPermissionGranted(boolean isGranted, String permission) {
    }

    public void makeFullScreen() {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        switch (requestCode) {
            case READ_WRITE_STORAGE:
                isPermissionGranted(grantResults[0] == PackageManager.PERMISSION_GRANTED, permissions[0]);
                break;
        }
    }

    protected void showLoading(@NonNull String message) {
        mProgressDialog = new ProgressDialog(this);
    }
}
```

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

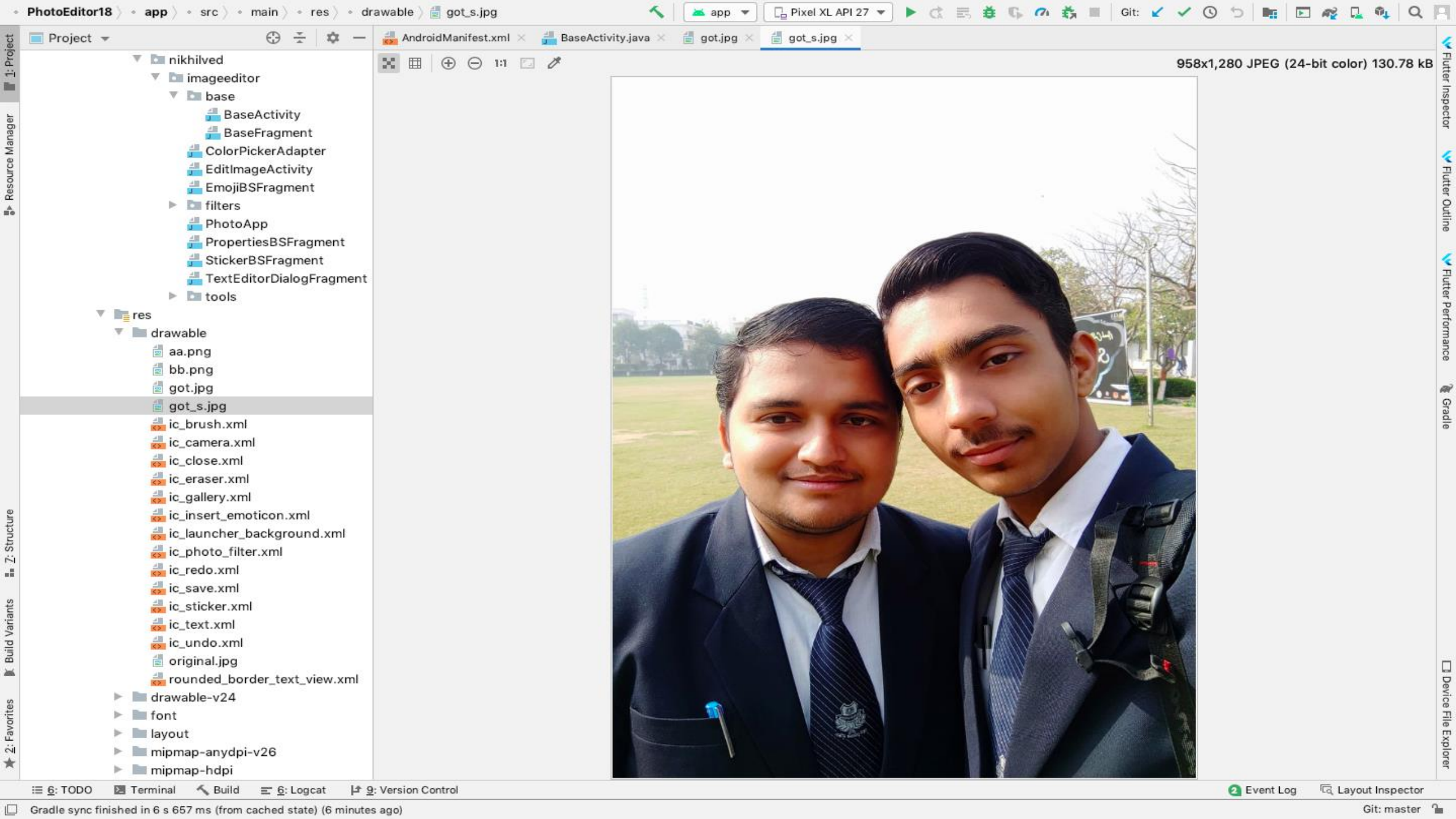
Gradle sync finished in 6 s 657 ms (from cached state) (4 minutes ago)

17:14 LF UTF-8 4 spaces Git: master

The screenshot displays the Android Studio interface. On the left, the 'Project' view shows the file structure of 'PhotoEditor18', including folders like 'src', 'main', 'assets', 'java', 'com', 'nikhilved', 'imageeditor', 'base', and 'res'. The 'BaseActivity' class is highlighted in the 'base' folder. The main editor shows the code for 'BaseActivity.java', which includes methods for making the screen full, handling permissions, showing/hiding a loading dialog, and showing a snackbar. The code is as follows:

```
38 public void makeFullScreen() {
39     requestWindowFeature(Window.FEATURE_NO_TITLE);
40     getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
41 }
42
43 @Override
44 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
45     switch (requestCode) {
46         case READ_WRITE_STORAGE:
47             isPermissionGranted(grantResults[0] == PackageManager.PERMISSION_GRANTED, permissions[0]);
48             break;
49     }
50 }
51
52 protected void showLoading(@NonNull String message) {
53     mProgressDialog = new ProgressDialog(this);
54     mProgressDialog.setMessage(message);
55     mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
56     mProgressDialog.setCancelable(false);
57     mProgressDialog.show();
58 }
59
60 protected void hideLoading() {
61     if (mProgressDialog != null) {
62         mProgressDialog.dismiss();
63     }
64 }
65
66 protected void showSnackbar(@NonNull String message) {
67     View view = findViewById(android.R.id.content);
68     if (view != null) {
69         Snackbar.make(view, message, Snackbar.LENGTH_SHORT).show();
70     } else {
71         Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
72     }
73 }
74 }
75
```

The bottom status bar indicates the current class is 'BaseActivity'.



itor18

app

src

main

java

com

nikhilved

imageeditor

EditImageActivity

app

Pixel XL API 27

Project

PhotoEditor18

~/Downloads/PhotoEditor18

.gitignore

.idea

app

.gitignore

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

ic_photo_filter.xml

ic_redo.xml

EditImageActivity.java

1

package com.nikhilved.imageeditor;

2

3

import ...

42

43

public class EditImageActivity extends BaseActivity implements OnPhotoEditorListener,

44

View.OnClickListener,

45

PropertiesBSFragment.Properties,

46

EmojiBSFragment.EmojiListener,

47

StickerBSFragment.StickerListener, EditingToolsAdapter.OnItemSelected, FilterListener {

48

49

private static final String TAG = EditImageActivity.class.getSimpleName();

50

public static final String EXTRA_IMAGE_PATHS = "extra_image_paths";

51

private static final int CAMERA_REQUEST = 52;

52

private static final int PICK_REQUEST = 53;

53

private PhotoEditor mPhotoEditor;

54

private PhotoEditorView mPhotoEditorView;

55

private PropertiesBSFragment mPropertiesBSFragment;

56

private EmojiBSFragment mEmojiBSFragment;

57

private StickerBSFragment mStickerBSFragment;

58

private TextView mTxtCurrentTool;

59

private Typeface mWonderFont;

60

private RecyclerView mRvTools, mRvFilters;

61

private EditingToolsAdapter mEditingToolsAdapter = new EditingToolsAdapter(this);

62

private FilterViewAdapter mFilterViewAdapter = new FilterViewAdapter(this);

63

private ConstraintLayout mRootView;

64

private ConstraintSet mConstraintSet = new ConstraintSet();

65

private boolean mIsFilterVisible;

66

67

68

@Override

69

protected void onCreate(Bundle savedInstanceState) {

70

super.onCreate(savedInstanceState);

71

makeFullScreen();

72

setContentView(R.layout.activity_edit_image);

73

74

initViews();

75

76

mWonderFont = Typeface.createFromAsset(getAssets(), "beyond_wonderland.ttf");

77

78

mPropertiesBSFragment = new PropertiesBSFragment();

79

mEmojiBSFragment = new EmojiBSFragment();

80

mStickerBSFragment = new StickerBSFragment();

81

mStickerBSFragment.setStickerListener(this);

EditImageActivity

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

Gradle sync finished in 6 s 657 ms (from cached state) (7 minutes ago)

43:14 LF UTF-8 4 spaces Git: master

PhotoEditor18

src

main

java

com

nikhilved

imageeditor

base

EditImageActivity

PhotoEditor18

~/Downloads/PhotoEditor18

.gitignore

.idea

app

.gitignore

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

ic_photo_filter.xml

ic_redo.xml

EditImageActivity.java

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

makeFullScreen();

setContentView(R.layout.activity_edit_image);

initViews();

mWonderFont = Typeface.createFromAsset(getAssets(), "beyond_wonderland.ttf");

mPropertiesBSFragment = new PropertiesBSFragment();

mEmojiBSFragment = new EmojiBSFragment();

mStickerBSFragment = new StickerBSFragment();

mStickerBSFragment.setStickerListener(this);

mEmojiBSFragment.setEmojiListener(this);

mPropertiesBSFragment.setPropertiesChangeListener(this);

LinearLayoutManager llmTools = new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false);

mRvTools.setLayoutManager(llmTools);

mRvTools.setAdapter(mEditingToolsAdapter);

LinearLayoutManager llmFilters = new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false);

mRvFilters.setLayoutManager(llmFilters);

mRvFilters.setAdapter(mFilterViewAdapter);

//Typeface mTextRobotoTf = ResourcesCompat.getFont(this, R.font.roboto_medium);

//Typeface mEmojiTypeFace = Typeface.createFromAsset(getAssets(), "emojione-android.ttf");

mPhotoEditor = new PhotoEditor.Builder(this, mPhotoEditorView)

.setPinchTextScalable(true) // set flag to make text scalable when pinch

//setDefaultTextTypeface(mTextRobotoTf)

//setDefaultEmojiTypeface(mEmojiTypeFace)

.build(); // build photo editor sdk

mPhotoEditor.setOnPhotoEditorListener(this);

//Set Image Dynamically

// mPhotoEditorView.getSource().setImageResource(R.drawable.color_palette);

}

private void initViews() {

EditImageActivity

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

Gradle sync finished in 6 s 657 ms (from cached state) (7 minutes ago)

43:14

LF

UTF-8

4 spaces

Git: master

app

src

main

java

com

nikhilved

imageeditor

filters

FilterViewAdapter

app

Pixel XL API 27

Git

Project

PhotoEditor18

~/Downloads/PhotoEditor18

.gitignore

.idea

app

.gitignore

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

FilterListener

FilterViewAdapter

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

FilterViewAdapter.java

1 package com.nikhilved.imageeditor.filters;

2

3 import ...

24

25

26 public class FilterViewAdapter extends RecyclerView.Adapter<FilterViewAdapter.ViewHolder> {

27

28 private FilterListener mFilterListener;

29 private List<Pair<String, PhotoFilter>> mPairList = new ArrayList<>();

30

31 public FilterViewAdapter(FilterListener filterListener) {

32 mFilterListener = filterListener;

33 setupFilters();

34 }

35

36 @NonNull

37 @Override

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

39 View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.row_filter_view, parent, false);

40 return new ViewHolder(view);

41 }

42

43 @Override

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

45 Pair<String, PhotoFilter> filterPair = mPairList.get(position);

46 Bitmap fromAsset = getBitmapFromAsset(holder.itemView.getContext(), filterPair.first);

47 holder.mImageFilterView.setImageBitmap(fromAsset);

48 holder.mTxtFilterName.setText(filterPair.second.name().replace("_", " "));

49 }

50

51 @Override

52 public int getItemCount() { return mPairList.size(); }

53

54

55

56 class ViewHolder extends RecyclerView.ViewHolder {

57 ImageView mImageFilterView;

58 TextView mTxtFilterName;

59

60 ViewHolder(View itemView) {

61 super(itemView);

62 mImageFilterView = itemView.findViewById(R.id.imgFilterView);

63 mTxtFilterName = itemView.findViewById(R.id.txtFilterName);

64 itemView.setOnClickListener(new OnClickListener() {

65 mFilterListener.onFilterSelected(mPairList.get(getLayoutPosition()).second);

66 }

67 }

FilterViewAdapter

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

Gradle sync finished in 6 s 657 ms (from cached state) (11 minutes ago)

26:14

LF

UTF-8

4 spaces

Git: master

app

src

main

java

com

nikhilved

imageeditor

filters

FilterViewAdapter

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

FilterListener

FilterViewAdapter

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

Project

PhotoEditor18

~/Downloads/PhotoEditor18

.gitignore

.idea

app

.gitignore

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

FilterListener

FilterViewAdapter

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

ic_launcher_background.xml

FilterViewAdapter.java

68

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

}};

}

}

private Bitmap getBitmapFromAsset(Context context, String strName) {

AssetManager assetManager = context.getAssets();

InputStream istr = null;

try {

istr = assetManager.open(strName);

return BitmapFactory.decodeStream(istr);

} catch (IOException e) {

e.printStackTrace();

return null;

}

}

private void setupFilters() {

mPairList.add(new Pair<("filters/original.jpg", PhotoFilter.NONE));

mPairList.add(new Pair<("filters/auto_fix.png", PhotoFilter.AUTO_FIX));

mPairList.add(new Pair<("filters/brightness.png", PhotoFilter.BRIGHTNESS));

mPairList.add(new Pair<("filters/contrast.png", PhotoFilter.CONTRAST));

mPairList.add(new Pair<("filters/documentary.png", PhotoFilter.DOCUMENTARY));

mPairList.add(new Pair<("filters/dual_tone.png", PhotoFilter.DUE_TONE));

mPairList.add(new Pair<("filters/fill_light.png", PhotoFilter.FILL_LIGHT));

mPairList.add(new Pair<("filters/fish_eye.png", PhotoFilter.FISH_EYE));

mPairList.add(new Pair<("filters/grain.png", PhotoFilter.GRAIN));

mPairList.add(new Pair<("filters/gray_scale.png", PhotoFilter.GRAY_SCALE));

mPairList.add(new Pair<("filters/lomish.png", PhotoFilter.LOMISH));

mPairList.add(new Pair<("filters/negative.png", PhotoFilter.NEGATIVE));

mPairList.add(new Pair<("filters/posterize.png", PhotoFilter.POSTERIZE));

mPairList.add(new Pair<("filters/saturate.png", PhotoFilter.SATURATE));

mPairList.add(new Pair<("filters/sepia.png", PhotoFilter.SEPIA));

mPairList.add(new Pair<("filters/sharpen.png", PhotoFilter.SHARPEN));

mPairList.add(new Pair<("filters/temprature.png", PhotoFilter.TEMPERATURE));

mPairList.add(new Pair<("filters/tint.png", PhotoFilter.TINT));

mPairList.add(new Pair<("filters/vignette.png", PhotoFilter.VIGNETTE));

mPairList.add(new Pair<("filters/cross_process.png", PhotoFilter.CROSS_PROCESS));

mPairList.add(new Pair<("filters/b_n_w.png", PhotoFilter.BLACK_WHITE));

mPairList.add(new Pair<("filters/flip_horizontal.png", PhotoFilter.FLIP_HORIZONTAL));

mPairList.add(new Pair<("filters/flip_vertical.png", PhotoFilter.FLIP_VERTICAL));

mPairList.add(new Pair<("filters/rotate.png", PhotoFilter.ROTATE));

}

}

FilterViewAdapter

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

6: TODO

Terminal

Build

6: Logcat

9: Version Control

Event Log

Layout Inspector

Gradle sync finished in 6 s 657 ms (from cached state) (11 minutes ago)

26:14

LF

UTF-8

4 spaces

Git: master

PhotoEditor18

• app

• src

• main

• com

nikhilved

imageeditor

PhotoApp

app

build.gradle

proguard-rules.pro

src

main

AndroidManifest.xml

assets

java

com

nikhilved

imageeditor

base

ColorPickerAdapter

EditImageActivity

EmojiBSFragment

filters

FilterListener

FilterViewAdapter

PhotoApp

PropertiesBSFragment

StickerBSFragment

TextEditorDialogFragment

tools

res

drawable

aa.png

bb.png

got.jpg

got_s.jpg

ic_brush.xml

ic_camera.xml

ic_close.xml

ic_eraser.xml

ic_gallery.xml

ic_insert_emoticon.xml

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

47

import ...

public class PhotoApp extends Application {

private static PhotoApp sPhotoApp;

private static final String TAG = PhotoApp.class.getSimpleName();

@Override

public void onCreate() {

super.onCreate();

sPhotoApp = this;

/* FontRequest fontRequest = new FontRequest(

com.google.android.gms.fonts",

com.google.android.gms",

Noto Color Emoji Compat",

R.array.com_google_android_gms_fonts_certs);

EmojiCompat.Config config = new FontRequestEmojiCompatConfig(this, fontRequest)

.setReplaceAll(true)

// .setEmojiSpanIndicatorEnabled(true)

// .setEmojiSpanIndicatorColor(Color.GREEN)

.registerInitCallback(new EmojiCompat.InitCallback() {

@Override

public void onInitialized() {

super.onInitialized();

Log.e(TAG, "Success");

}

@Override

public void onFailed(@Nullable Throwable throwable) {

super.onFailed(throwable);

Log.e(TAG, "onFailed: " + throwable.getMessage());

}

});

// BundledEmojiCompatConfig bundledEmojiCompatConfig = new BundledEmojiCompatConfig(this);

EmojiCompat.init(config);*/

}

public static PhotoApp getPhotoApp() { return sPhotoApp; }

Flutter Inspector

Flutter Outline

Flutter Performance

Gradle

Device File Explorer

4: Run

6: TODO

Terminal

Build

Profiler

6: Logcat

9: Version Control

Event Log

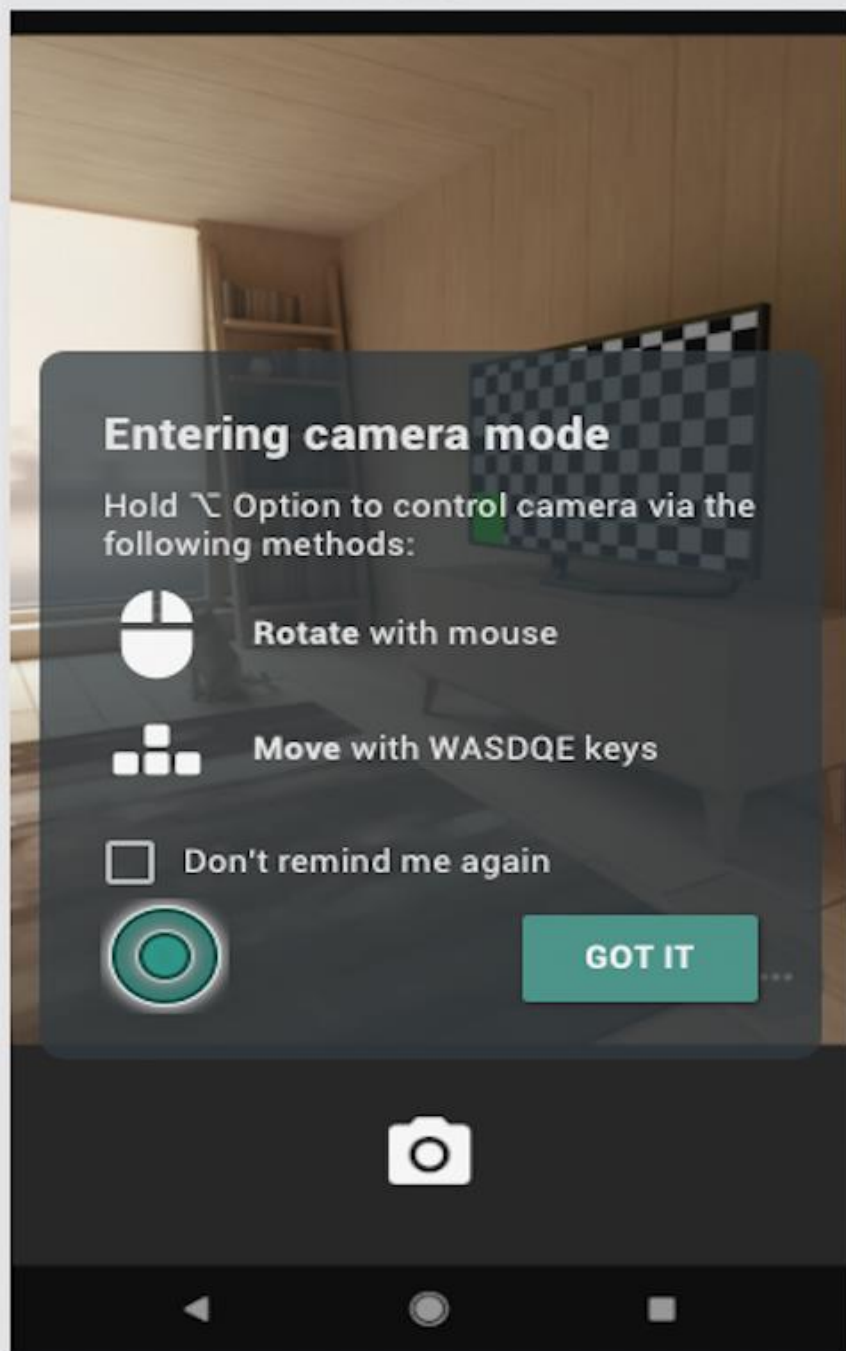
Layout Inspector

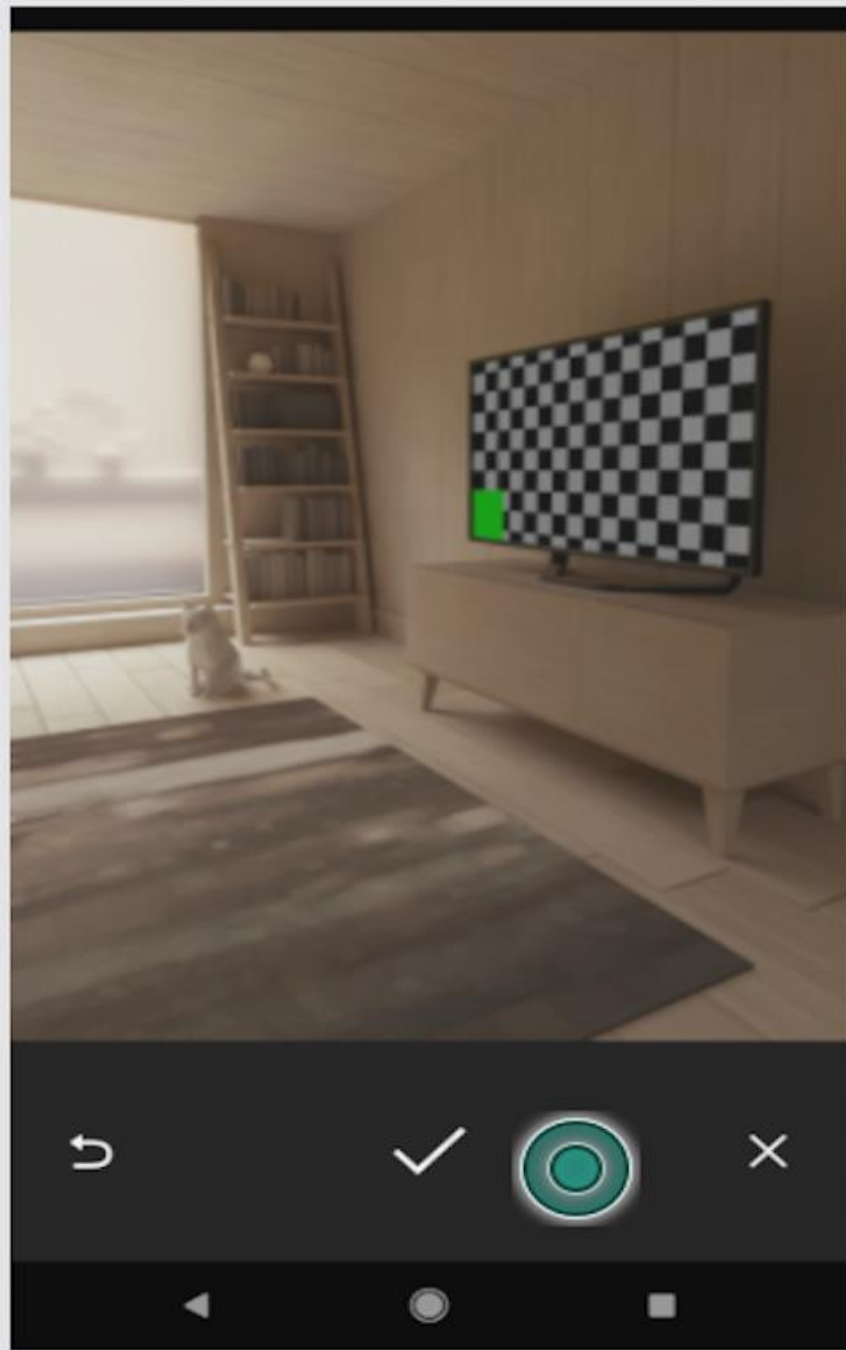
Install successfully finished in 595 ms.

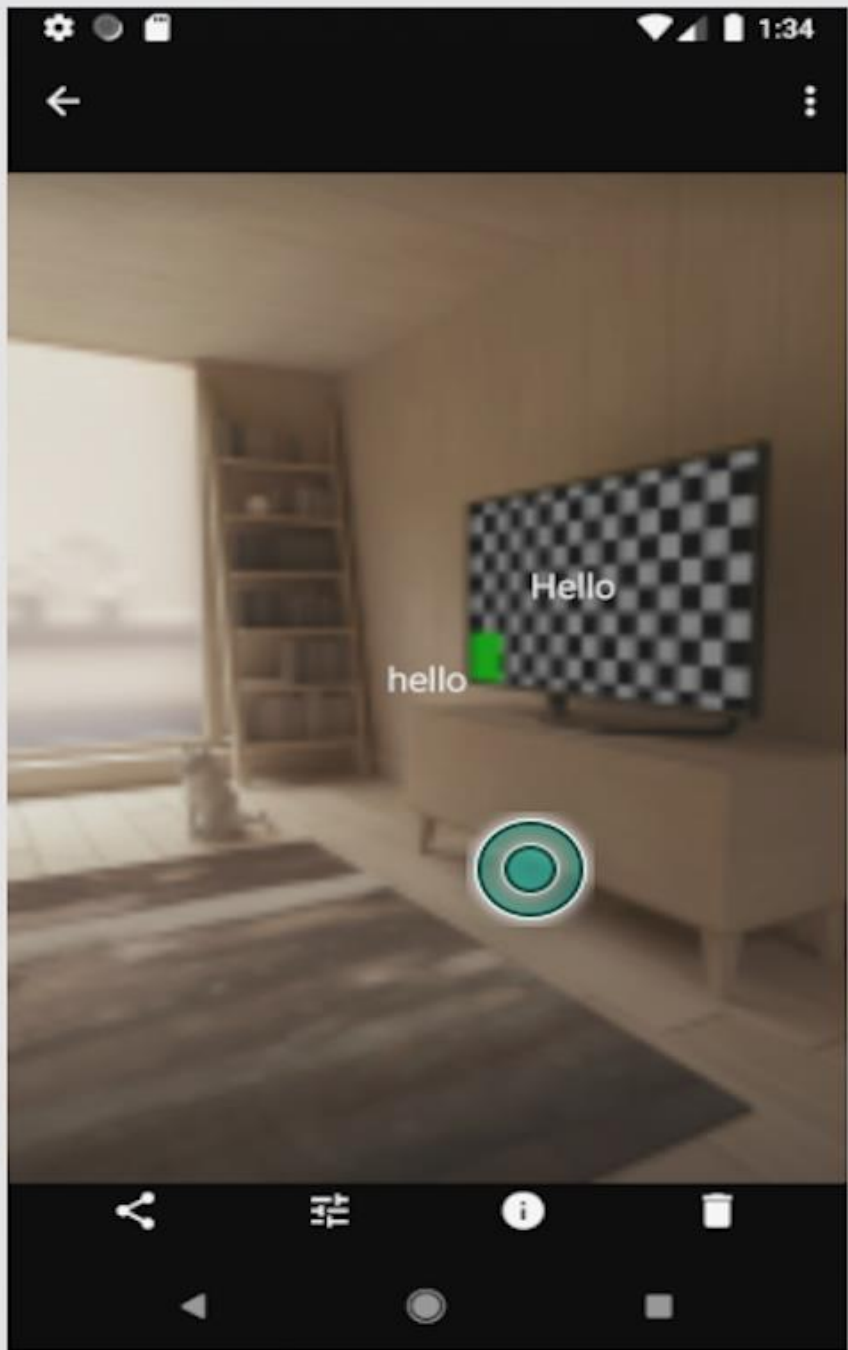
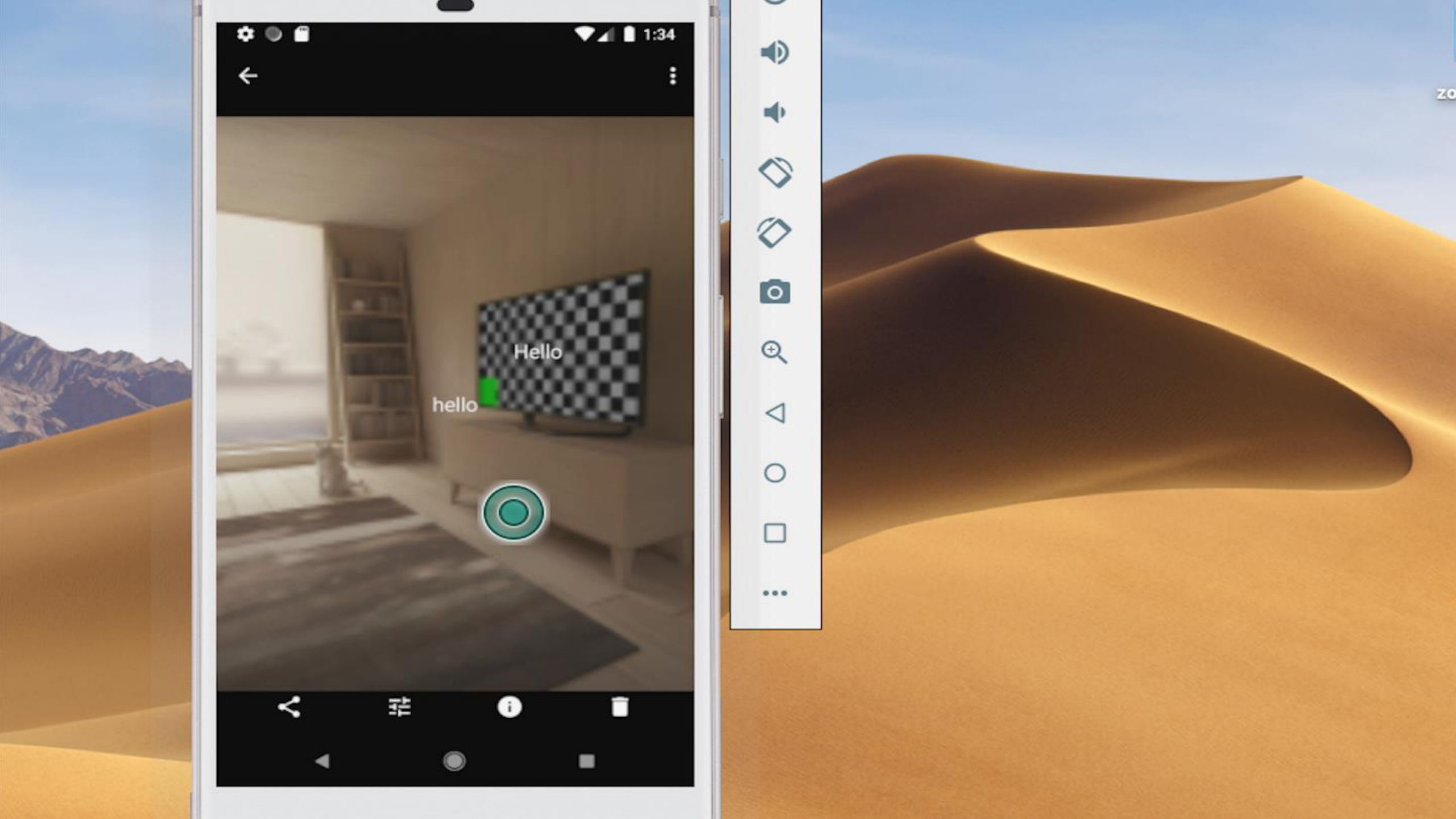
App restart successful without requiring a re-install.

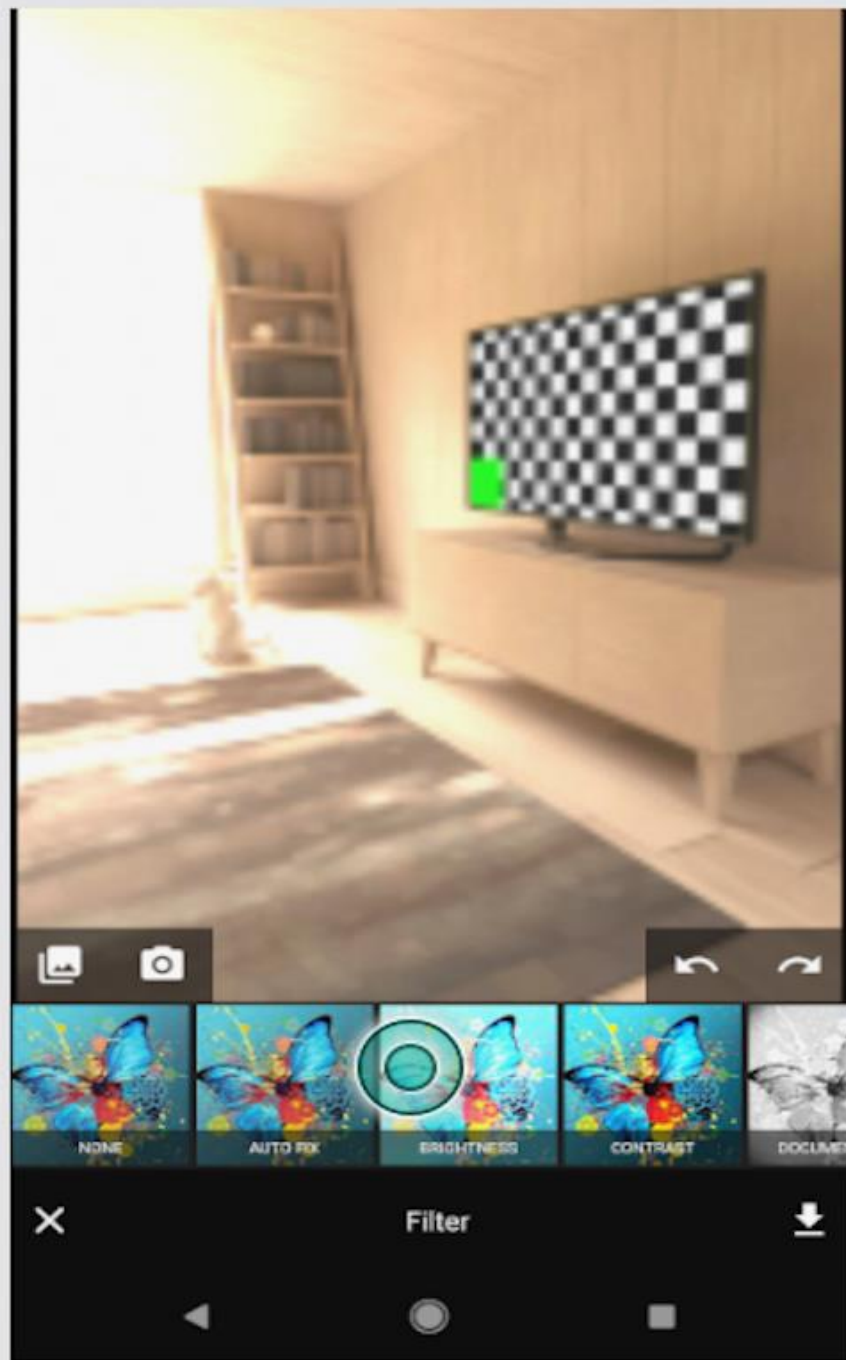
Install successfully finished in 595 ms.: App restart successful without requiring a re-install. (moments ago)

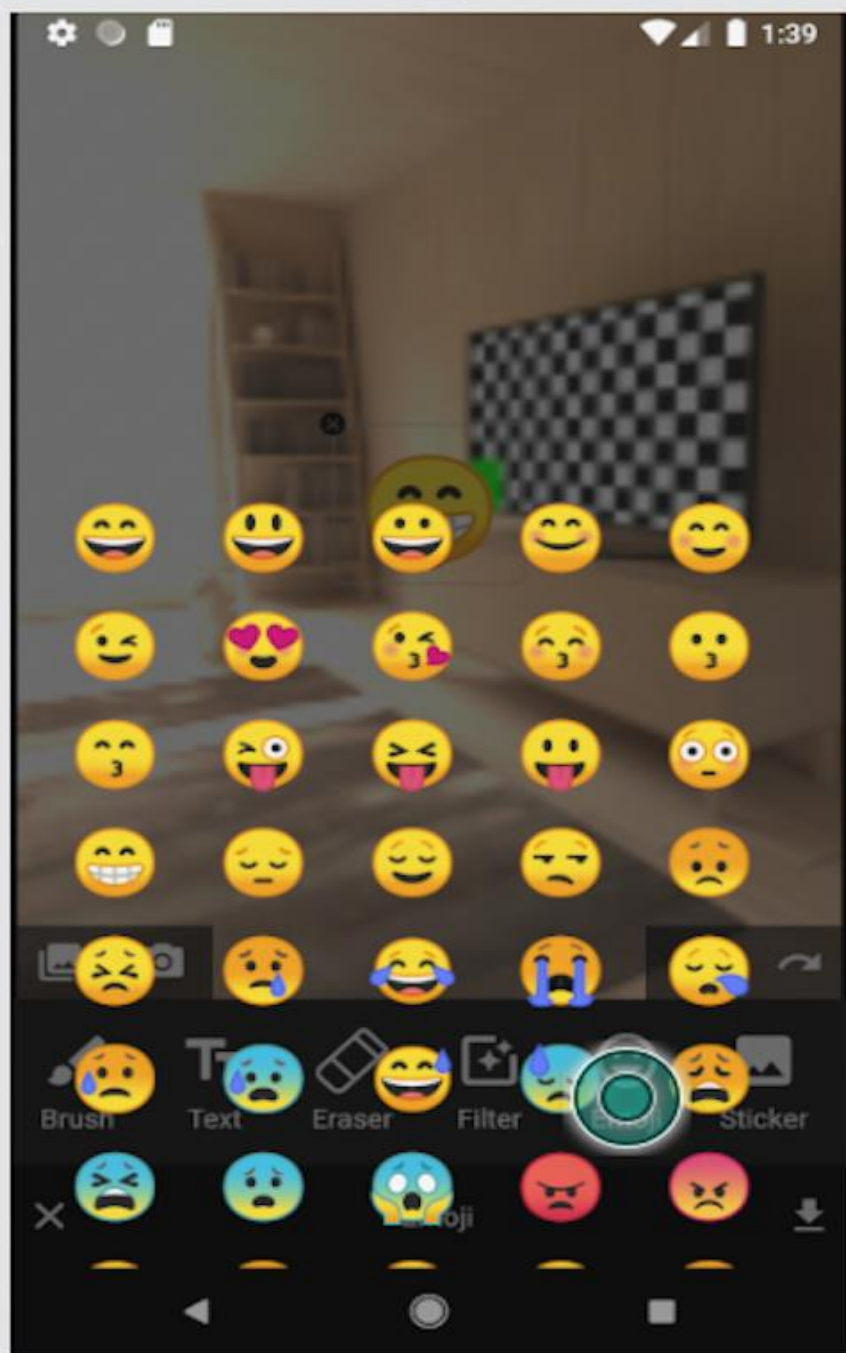
8:14 LF UTF-8 4 spaces Git: master













Brush



Text



Eraser



Filter



Emoji



Sticker

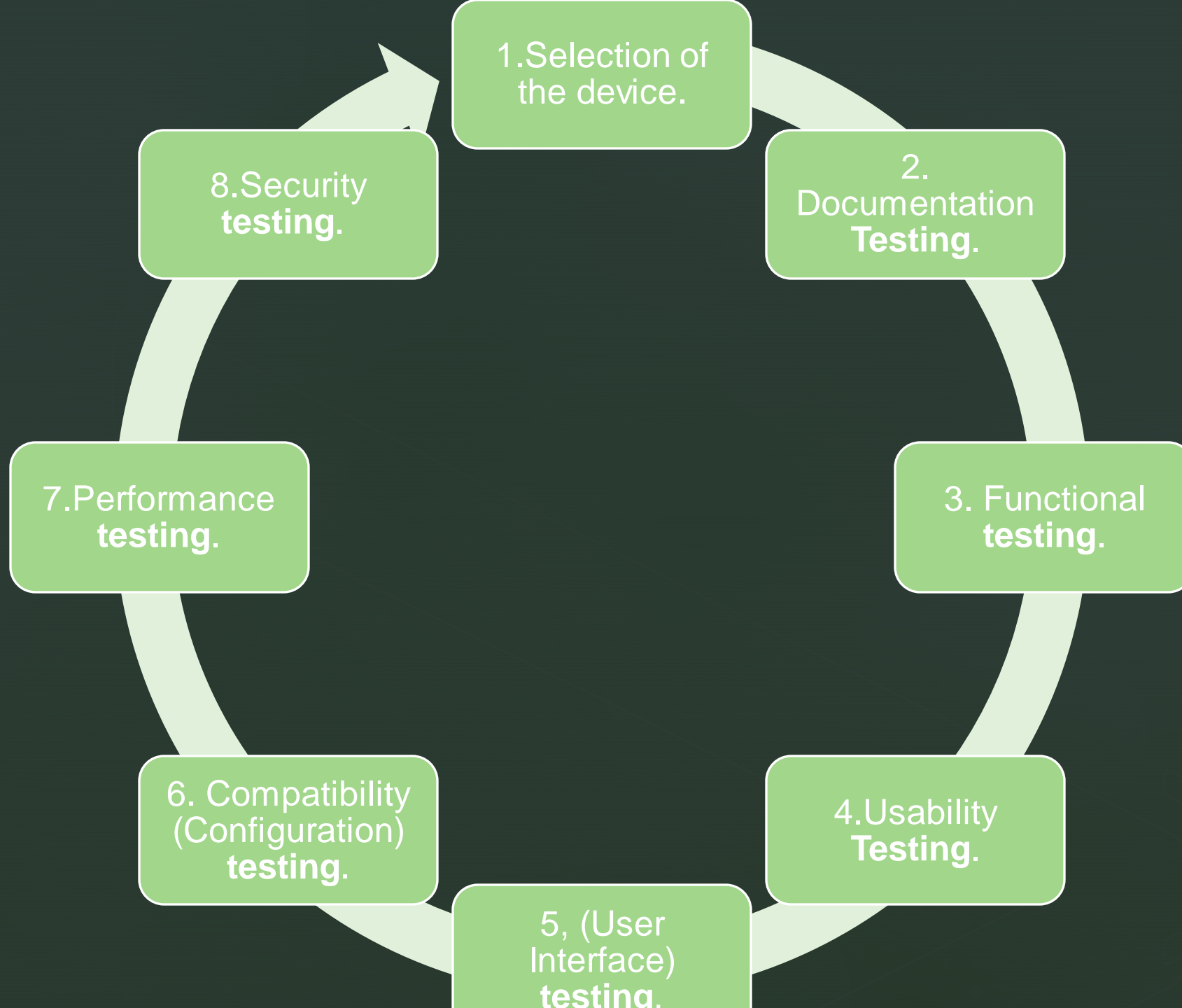


Emoji





Testing



Scope for
extension into a
major project



This application is accessible on any system and any platform. This application can be used to change the platform like Flutter or more advance technology can be used for enhancement of the project into the major project then after photo editing application is applicable to run on different platforms like Windows, Android , and ios, which is very useful to edit photo similar to the professional photo.

Future Work:-

The Future Work for the Application will include the following:

1. Adding more photo editing packages.
2. Integration with social networking websites like Facebook, Twitter etc.
3. Layers functionality like in that of Adobe Photoshop.

Conclusion

This is an image editor with various image editing functionality that allows you to crop, zoom, transform, adjust brightness, and apply more such transformations over an image. This software system first allows the user to choose an image from your local computer. A menu bar on the top of the screen provides this functionality.

The menu bar allows the user to open a new image and even save an edited image in your computer. The application also allows you to zoom in and out of the image.

The user may apply various transformations over an image. The user may adjust image brightness and contrast. Users can even apply gamma correction to the image.

It also allows the user to crop and rotate the image as needed. The application also allows users to add reddish, bluish, and greenish effects to the image. Finally, the user may save the edited file as needed.

This app gives user the power to edit their picture easily and efficiently. Its an application which can be used by people of all ages who knows how to use a smartphone. The application uses minimum CPU memory and doesn't compromise with its performance. The editing is fast and smooth and its GUI is easy to use. There are many features which can be added to the app and those features will be added time to time with its regular updates.



GLA
UNIVERSITY
MATHURA
Established vide U.P. Act 21 of 2010.

Thank You