

CHAPTER 1

INTRODUCTION OF PROJECT

The advancement of Internet services and various storage technologies made video piracy as an increasing problem particularly with the proliferation of media sharing through the internet. Thus, research in copyright protection mechanisms and content authentication, where one of which includes video watermarking has been receiving an increasing interest from scientists especially in designing a seamless algorithm for effective implementation.

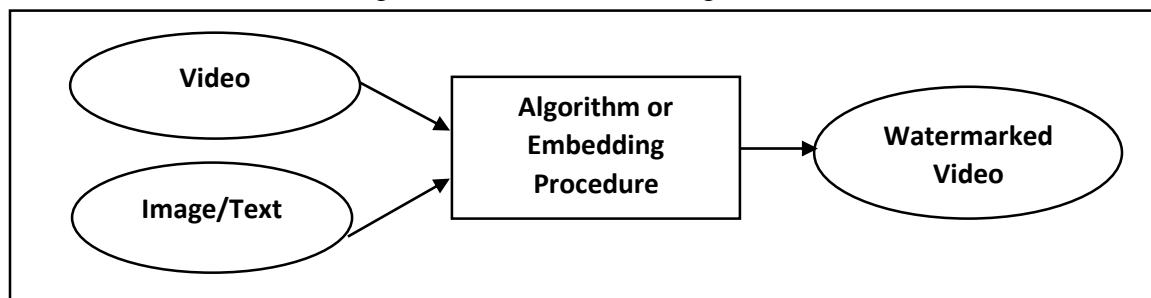
Basically video watermarking involves embedding secret symbols known as watermarks within video data which can be used later for copyright detection and authentication verification purposes.

Need for Video Watermarking:

- Protect copyright of a data.
- Video Watermarking can help
 - Prove ownership
 - Identify a misappropriating person.

Watermarks may be used in two-fold — to discourage unauthorized usage, and also act as an advertisement.

Fig 1.1 Video Watermarking Process



The Fig 1.1 represents video watermarking process where video and image/text are provided as input and output is shown in form of watermarked video.

The algorithm or embedding procedure works on frame manipulation to embed image or text into video frames.

CHAPTER 2

MOTIVATION

The main motivation comes for this project after studying various issues related to video and media legacy and piracy in modern digital era as like discussed in following session.

2.1 IN THE PAST

Conventionally, in analog world, a painting is signed by the artist to attest the copyright, an identity card is stamped by the steel seal to avoid forgery, and the paper money are identified by the embossed portrait. Such kind of hand-written signatures, seals and watermarks have been used from ancient times as a way to identify the source, creator of a document or a picture. For example, a priceless painting of the 11th century in National Palace Museum named "Travelers on a Mountain Path" had not been identified as the genuine work of Fan Kuan until Fan's signature is found between the woods behind a group of travelers of the painting.

Fig 2.1 Travelers on a Mountain

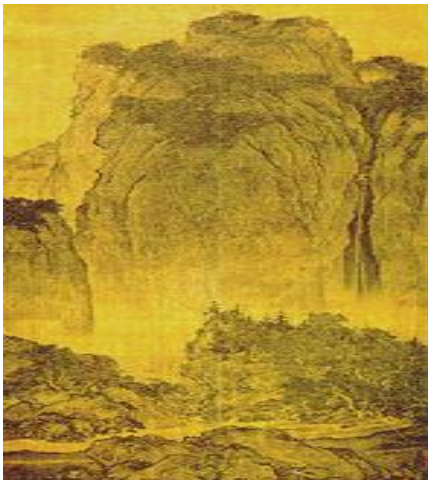


Fig 2.2 Fan's signature “范宽”



2.2 IN DIGITAL WORLD

However, in the digital world, digital technology for manipulating images and video has make it difficult to distinguish the visual truth. Besides, the characteristics of digitization bring significant hangs in copyright issues, which create an urgent need to intellectual property protection on the digitally recorded information.

Digital watermarking has been proposed as a way to claim the ownership of the source and owner. Unlike encryption, watermarking does not restrict access to the data. Once the encrypted data is decrypted, the intellectual property rights are no longer protected.

Over the past few years, the technology of the digital watermarking has gained prominence and emerged as a leading candidate that could solve the fundamental problems of legal ownership and content authentications for digital multimedia data. A great deal of research efforts has been focused on video watermarking in recent years.

CHAPTER 3

FEASIBILITY STUDY

A feasibility study is a test of a system proposal according to its workability impact on the organization, ability to meet user needs and effective use of resources. Three key considerations are involved in the feasibility analysis: economic, technical, behavioral. Before entering into the procedure of system designing, we are obliged to study about feasibility of introducing a new computerized system. To replace the existing system with a new one is quite easy, but we have to vanish the drawbacks of current system and make the user able to enjoy the advantages of coming system. The proposed system must be evaluated from a technical, operational and economic feasibility of developing a computer system. The objective of the feasibility study is not only solve the problem but also to accurate the sense of scope. During the study, the problem is defined is crystallized and aspects of the problem to be included in the system are determined. Consequently, cost and benefits are estimated with greater accuracy at this stage.

Apparently any image watermarking technique can be extended to watermark videos, but in reality video watermarking techniques need to meet other challenges than that in image watermarking schemes such as large volume of inherently redundant data between frames, the unbalance between the motion and motionless regions, real-time requirements in the video broadcasting etc. Watermarked video sequences are very much susceptible to pirate attacks such as frame averaging, digital-analog (AD/DA) conversion, and lossy compressions.

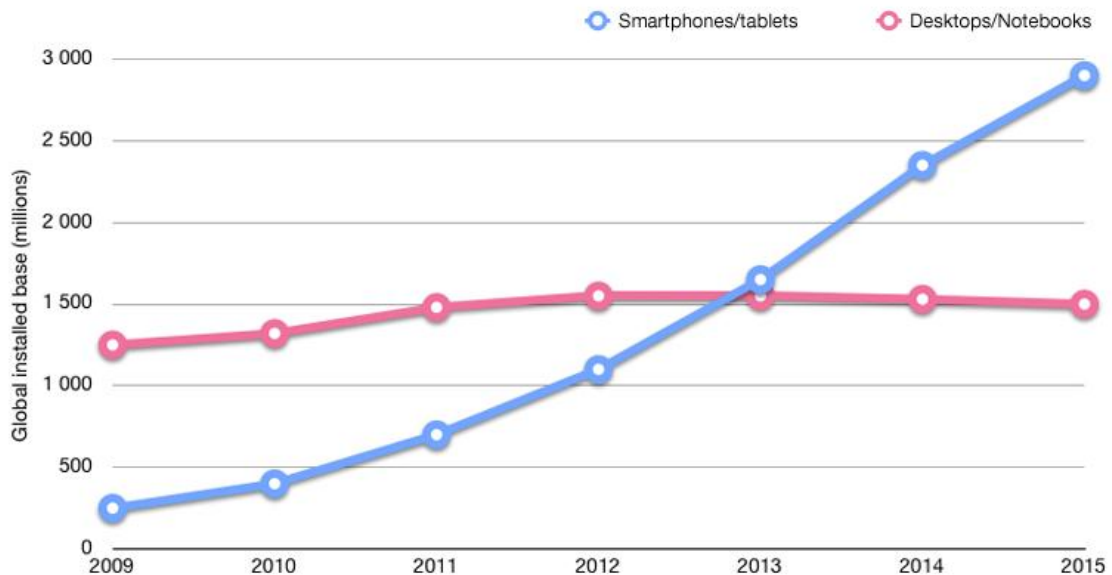
Video watermarking applications can be grouped as security related like Copy control, ownership, identification, authentication, etc. or value added applications like legacy system enhancement, video tagging, digital video broadcast monitoring, etc.

Based on all these studies economic, technical and operational feasibility is done relating to video watermarking android app development.

3.1 STATISTICS

Just a year ago, there was still nearly twice the percentage of desktop-only internet users (19.1 percent) as mobile-only users (10.8 percent). While the share of mobile-only users has climbed over the past year to 11.3 percent, the desktop-only population has drastically declined to just 10.6 percent. Hence mobile application is much closure to end user rather than desktop application.

Fig 3.1 Statistics on search engine usage in U.S.

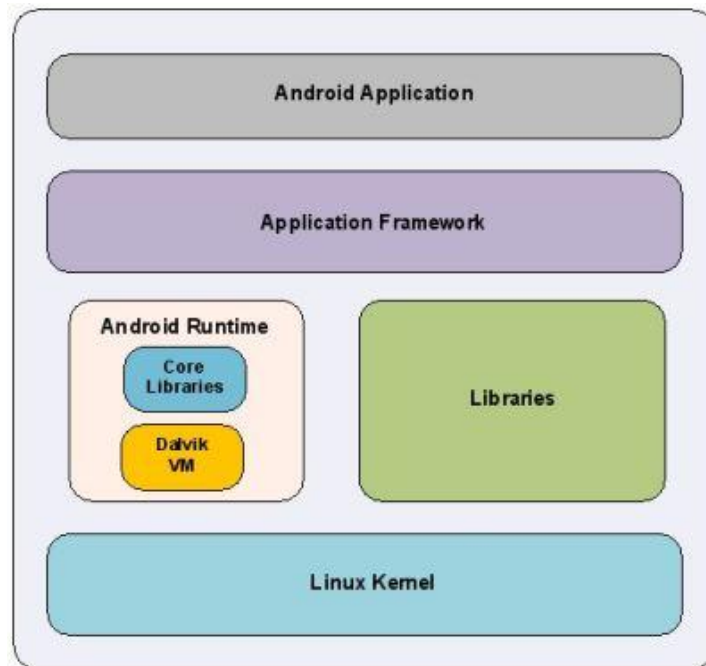


3.2 ANDROID ARCHITECTURE

Android platform has a software stack with operating system, middleware and key applications. The layers are the kernel, application framework layer, applications layer and libraries. The application layer is the location where all applications are present. The application framework layer provides APIs and managers to help the developer exploit the functions of Android. Android applications are written in the Java programming language. The Android SDK tools compile the code—along with any data and resource files—into

an Android package, an archive file with an .apk suffix. All the code in a single .apk file is considered to be one application and is the file that Android powered devices use to install the application.

Fig 3.2 Android Architecture



3.2.1 WHY ANDROID?

The ease of development, deployment and accessibility to maximum user simultaneously on to a mobile phone is the main reason of choosing Android to deploy the application.

3.3 ECONOMIC FEASIBILITY

The overall cost-benefit analysis shows it is economically feasible as it does not required any investment but have benefits from app user if they download from play store.

3.4 TECHNICAL FEASIBILITY

The requirements of this project involves Android Studio to develop app which is freely available on website. So this project is economically feasible.

3.5 OPERATIONAL FEASIBILITY

Due to simplicity and attractive interface the project is feasible in terms of operation feasibility.

CHAPTER 4

LITERATURE SURVEY

Literature survey is nothing but checking or reviewing the findings at surface level done through research or statistics.

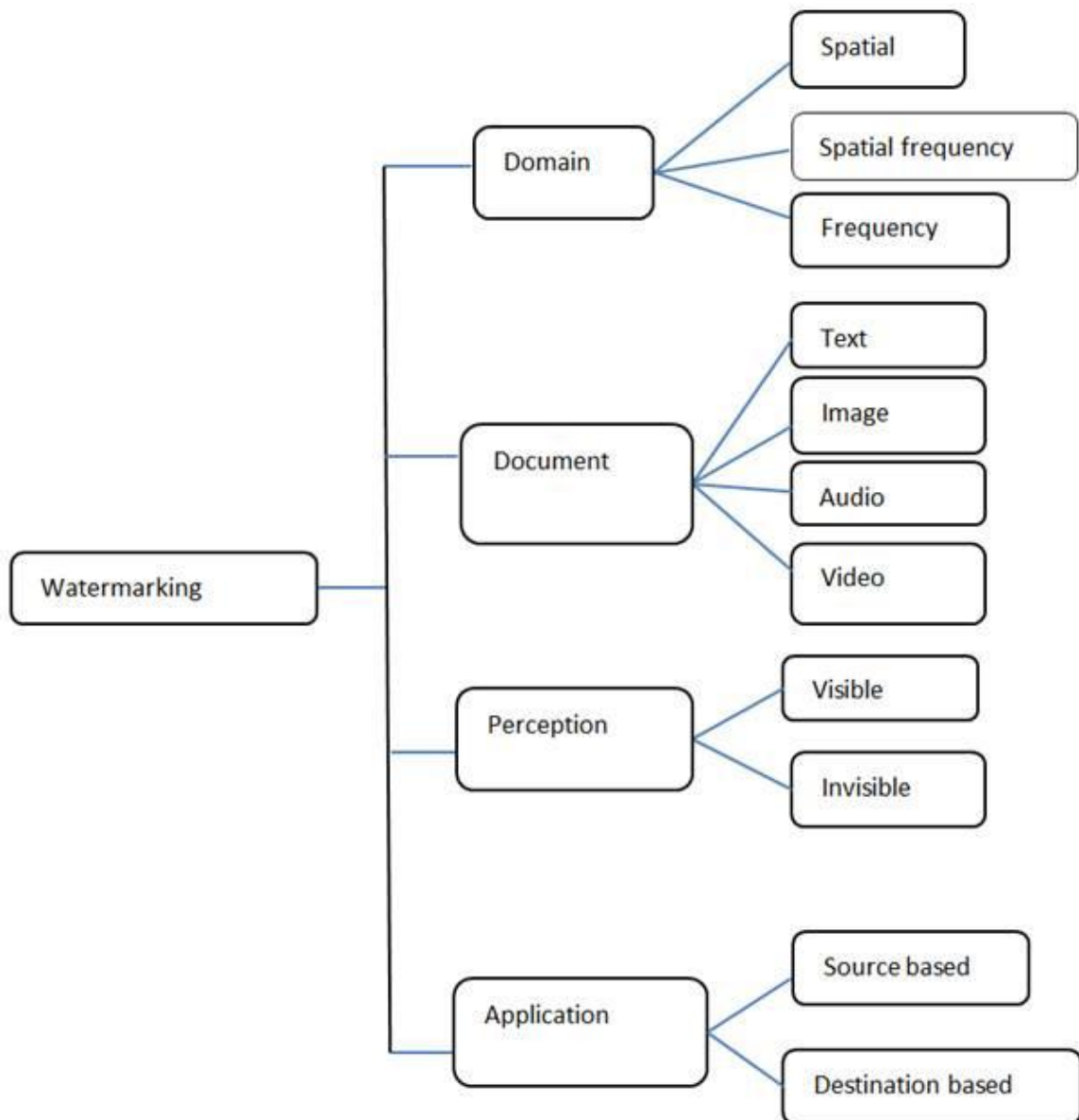
Due to the rapid development of the internet, the usages of multimedia data like audio, image and video become very popular. Transmission of digital contents is an easy task but maintaining the ownership is a difficult one, for this various techniques have been proposed. One of the main techniques to protect the copyright is digital watermarking. The concept of digital watermarking has been derived from steganography. Video watermarking is used to provide authentication for the video content. The various video watermarking techniques have been developed, but those techniques are withstand only basic attacks not resistant to video processing attacks like frame dropping, frame swapping and average and addition of noise etc.

The fundamental steps involved in the watermarking scheme is firstly the video bit streams are partially parsed and watermarked according to the information of a specific scene and image characteristics. The watermarked bit streams are then reconstructed and the framing information is added. Finally, the watermarked video is stored or delivered to the mainstream video network for further use.

The term "Video Watermark" was coined by Andrew Tirkel and Charles Osborne in December 1992. The first successful embedding and extraction of a steganographic spread spectrum watermark was demonstrated in 1993 by Andrew Tirkel, Charles Osborne and Gerard Rankin.

Watermarks are identification marks produced during the paper making process. The first watermarks appeared in Italy during the 13th century, but their use rapidly spread across Europe. They were used as a means to identify the papermaker or the trade guild that manufactured the paper. The marks often were created by a wire sewn onto the paper mold. Watermarks continue to be used today as manufacturer's marks and to prevent forgery.

Fig 4.1 Survey on Watermarking



CHAPTER 5

TOOLS AND TECHNOLOGY USED

Android application development can be started on either of the following operating systems –

- Microsoft® Windows® 10/8/7/Vista/2003 (32 or 64-bit)
- Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
- GNOME or KDE desktop.

Following tools and technology are required to build an android app:

- Java JDK5 or later version
- Java Runtime Environment (JRE) 6
- Android Studio

5.1 JDK

The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition; Java Platform, Enterprise Edition or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). On 17 November 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK

5.2 JRE

Java Runtime Environment (JRE) is a software package that contains what is required to run a Java program. It includes a Java Virtual Machine implementation together with an implementation of the Java Class Library. The Oracle Corporation, which owns the Java trademark, distributes a Java Runtime environment with their Java Virtual Machine called HotSpot.

5.3 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for the Android platform. It was announced on May 16, 2013 at the Google I/O conference.

Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, macOS and Linux and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

CHAPTER 6

METHODOLOGY

The lifecycle of mobile development is largely no different than the Software Development Life Cycle (SDLC) for web or desktop applications. As with those, there are usually 5 major portions of the process:

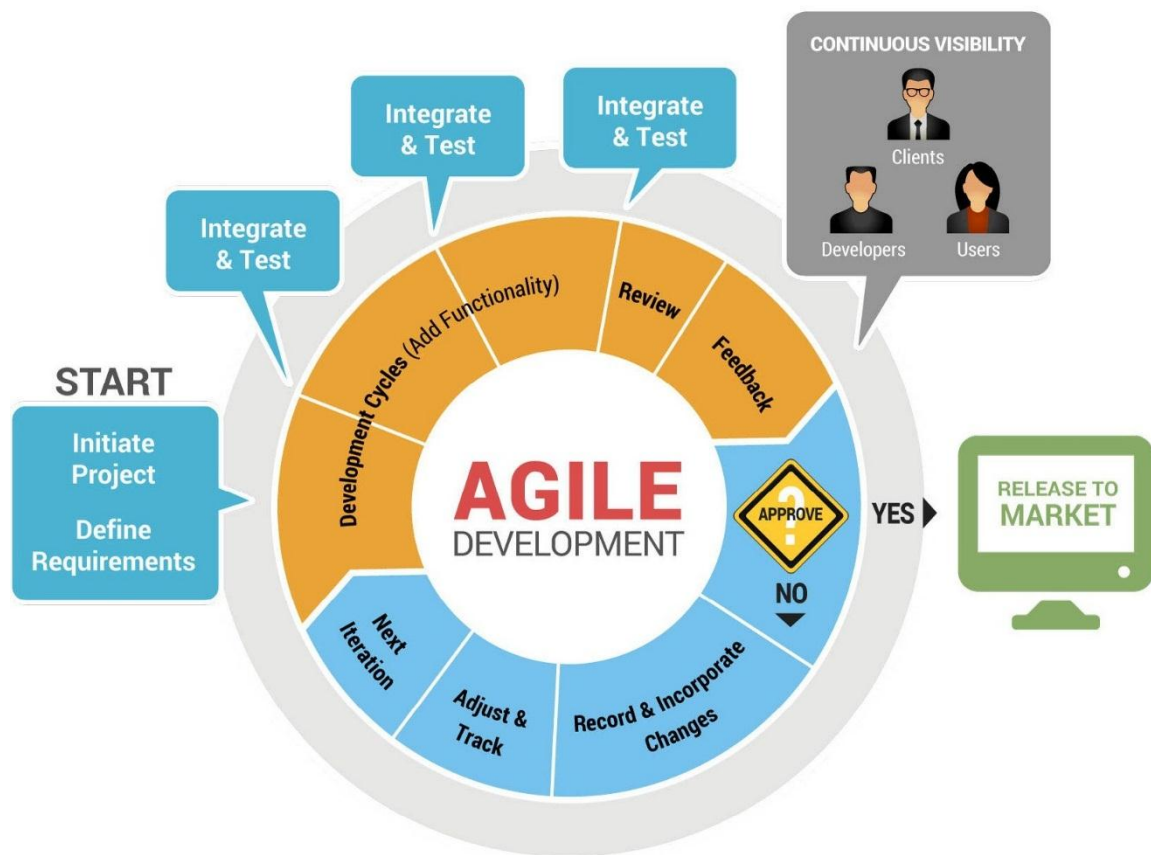
1. Inception – All apps start with an idea. That idea is usually refined into a solid basis for an application.
2. Design – The design phase consists of defining the app’s User Experience (UX) such as what the general layout is, how it works, etc., as well as turning that UX into a proper User Interface (UI) design, usually with the help of a graphic designer.
3. Development – Usually the most resource intensive phase, this is the actual building of the application.
4. Stabilization – When development is far enough along, QA usually begins to test the application and bugs are fixed. Often times an application will go into a limited beta phase in which a wider user audience is given a chance to use it and provide feedback and inform changes.
5. Deployment – Deployment is much easier.

6.1 AGILE MODEL

Agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods.

So, after seeing the requirements we came to conclusion for using agile model as shown in Fig 6.1.

Fig 6.1 Agile Model



It has following advantages-

- Is a very realistic approach to software development
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.

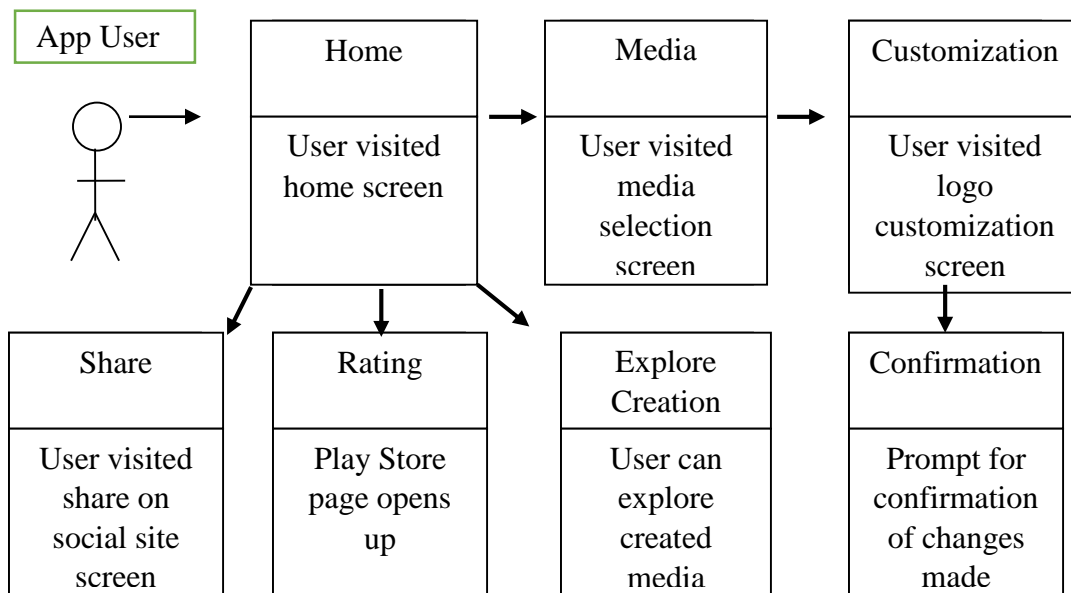
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

CHAPTER 7

IMPLEMENTATION DETAIL

The section describe the implementation detail for developing watermarking app. The project will be capable of running on android devices (Kitkat version and above).

Fig 7.1 Flow Chart



7.1 MODULES

7.1.1 HOME

Introduction: The home screen of the video watermarking app provides user to begin interactivity with app.

Details: The home screen will contain 4 buttons (start, my creation, share, and rate buttons) on clicking these buttons particular specified page opens up.

7.1.2 MEDIA

Introduction: The media screen of the video watermarking app provides is used for selecting image and video contents from mobile storage.

Details: It opens up when start button on home screen is pressed. The media screen will contain 2 buttons (image and video button) on clicking these buttons mobile gallery opens up prompting for media contents.

7.1.3 CUSTOMIZATION

Introduction: The customization screen of the video watermarking app decide the position where the logo to be placed.

Details: It opens up when user has selected both media content from media screen. The customization screen will contain play and pause video controlling buttons and seek bar to seek video to respective position. And 4 buttons for corner positioning of logo (i.e. bottom left corner, bottom right corner, top left corner and top right corner) and 1 center positioning button.

7.1.4 CONFIRMATION

Introduction: The confirmation screen of the video watermarking app prompt for confirmation of changes applied.

Details: It opens up when tick button on logo is pressed. The confirmation screen will contain yes and no button which decide whether to save changes or not. When user presses yes progress bar displays showing status of transcoding.

7.1.5 SHARE

Introduction: The share screen of the video watermarking app allow user to share his/her contents on social site.

Details: It opens up when share button on home screen is pressed. The share screen will contain 4 buttons mainly whatsapp, twitter, facebook and instagram buttons.

7.1.6 RATING

Introduction: The play store opens up.

Details: To rate and comment the play store opens up when user presses rate button on home screen.

7.1.7 EXPLORE CREATION

Introduction: The explore screen of the video watermarking app allows user to see their contents

Details: It opens up when my creation button is pressed on home screen. This allows user to play created contents of their own.

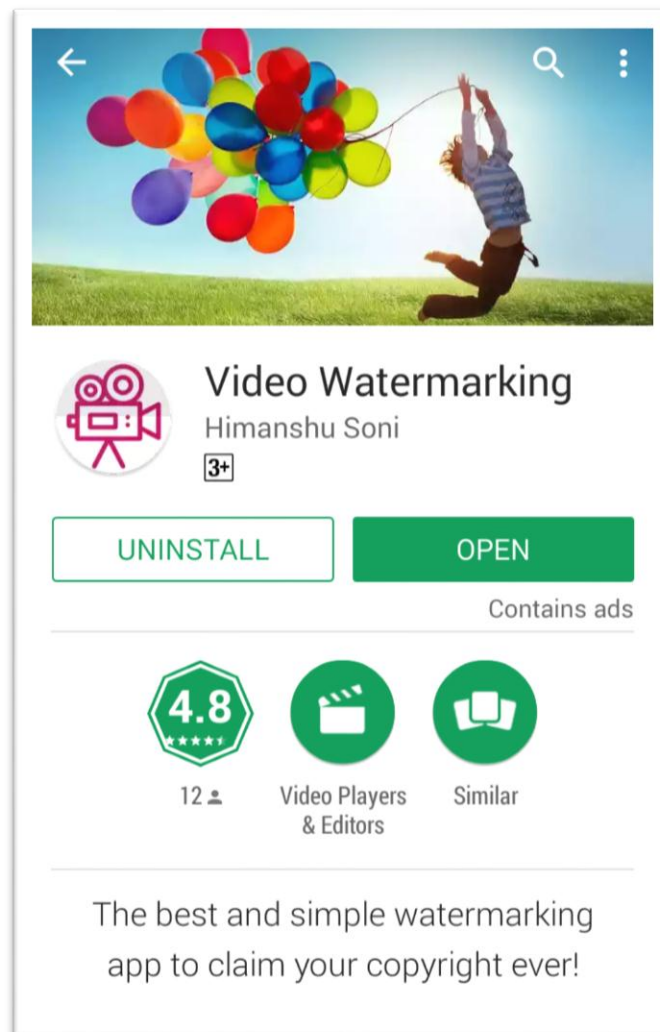
CHAPTER 8

RESULTS (SNAPSHOTS)

8.1 DOWNLOAD APP

Search and download video watermarking app from play store

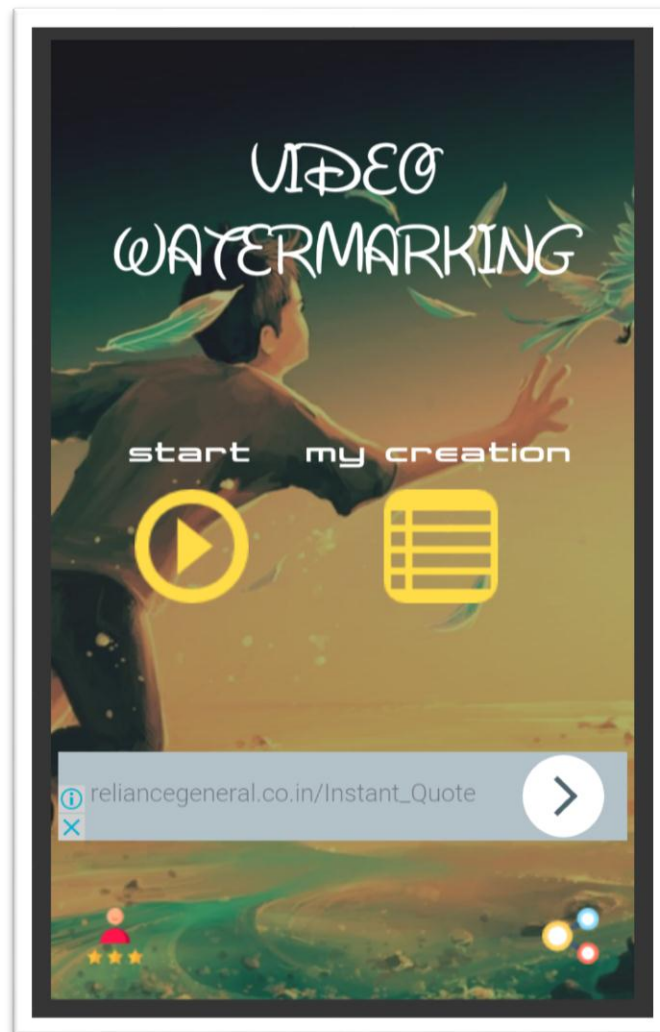
Fig 8.1 Download Page



8.2 HOME

Welcome to home screen.

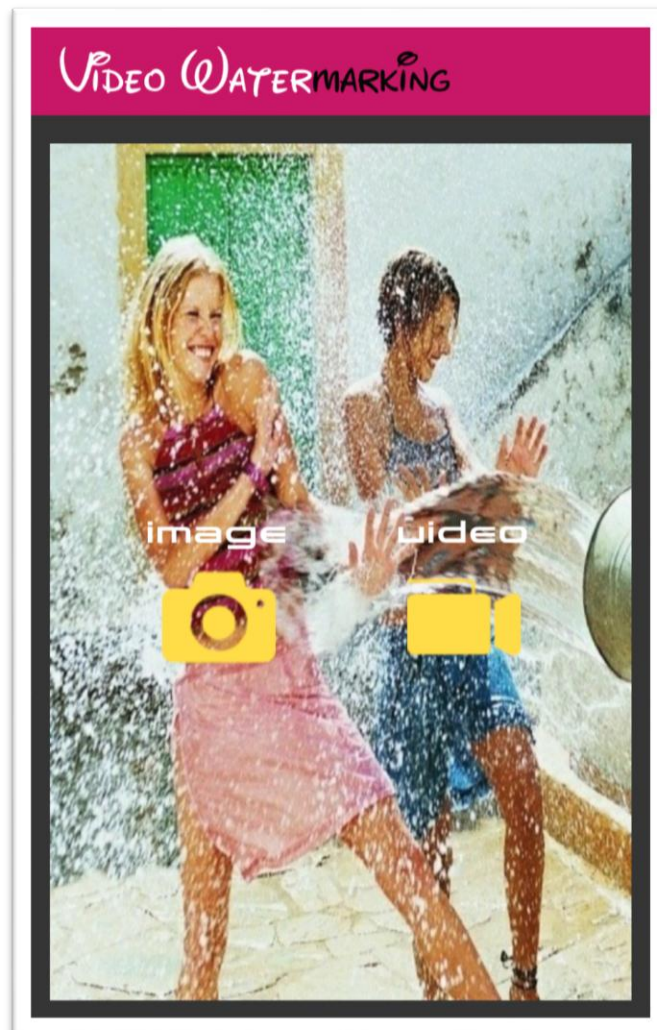
Fig 8.2 Home Screen



8.3 MEDIA

Open image/logo and video selection gallery here.

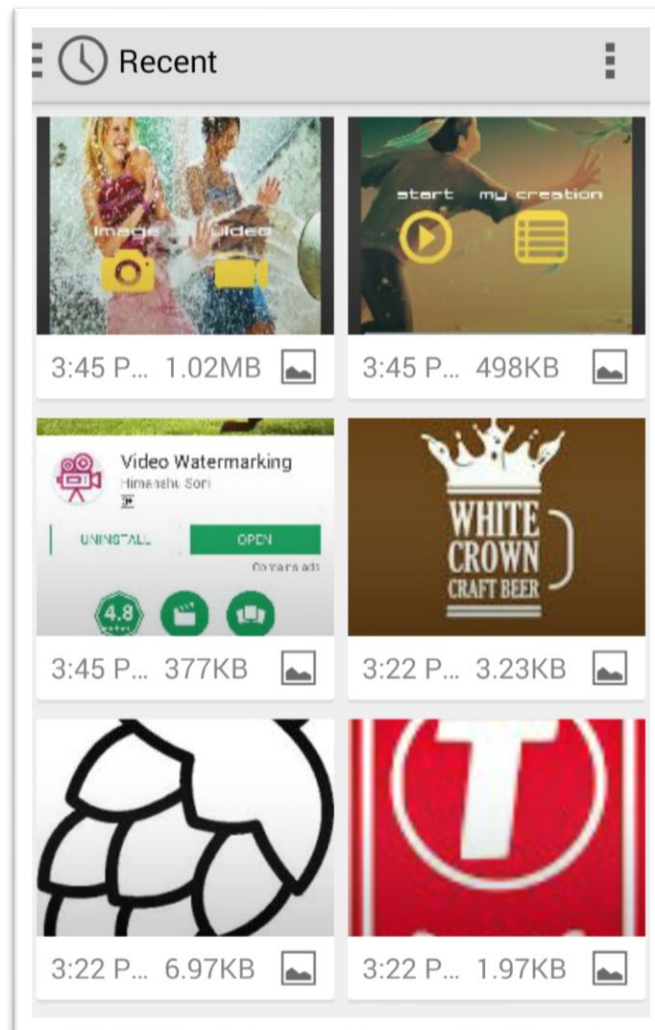
Fig 8.3 Media Page



8.3.1 IMAGE

Select image/logo here from gallery

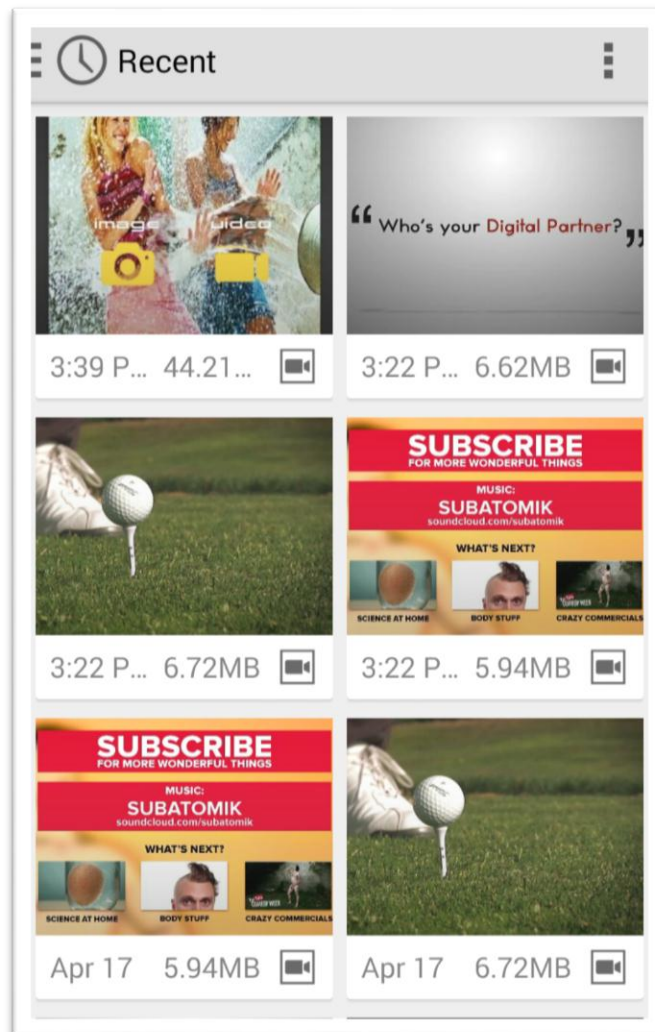
Fig 8.3.1 Image Gallery



8.3.2 VIDEO

Select video here from gallery.

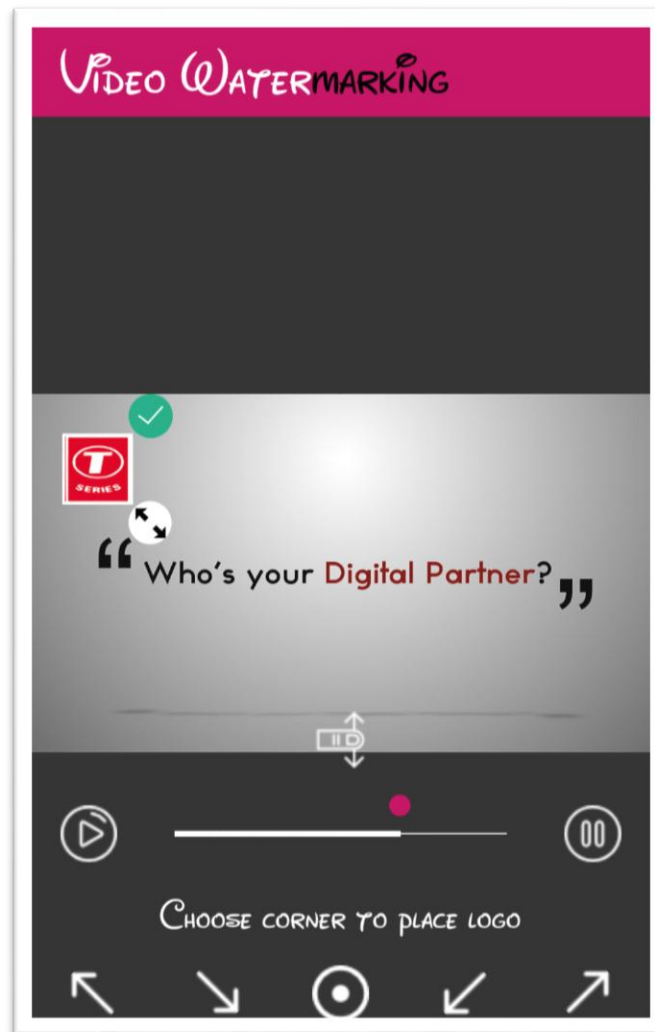
Fig 8.3.2 Video Gallery



8.4 CUSTOMIZATION

Customize logo here.

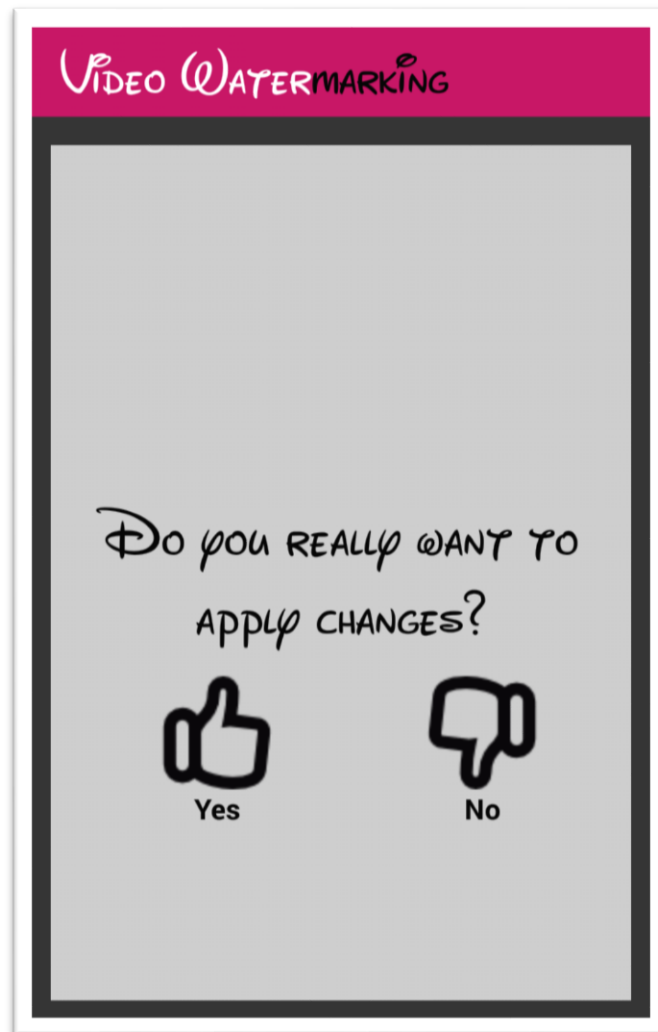
Fig 8.4 Customization Screen



8.5 CONFIRMATION

Prompt to apply changes or not.

Fig 8.5 Confirmation Screen



8.5.1 STATUS

Displays status of transcoding.

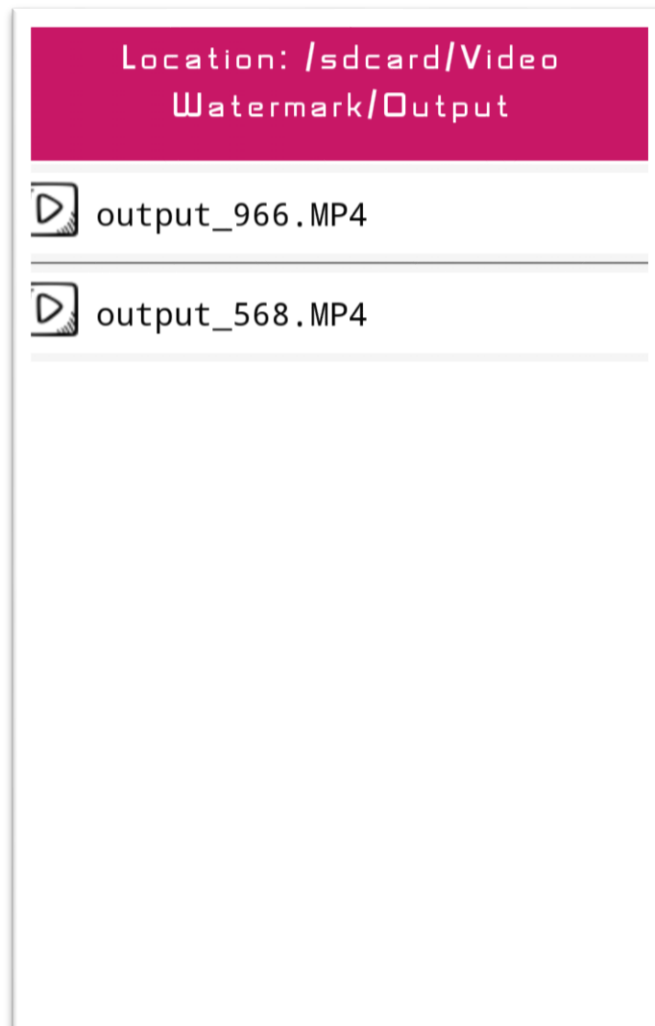
Fig 8.5.1 Status Dialog Screen



8.6 MY CREATION

Explore your creation.

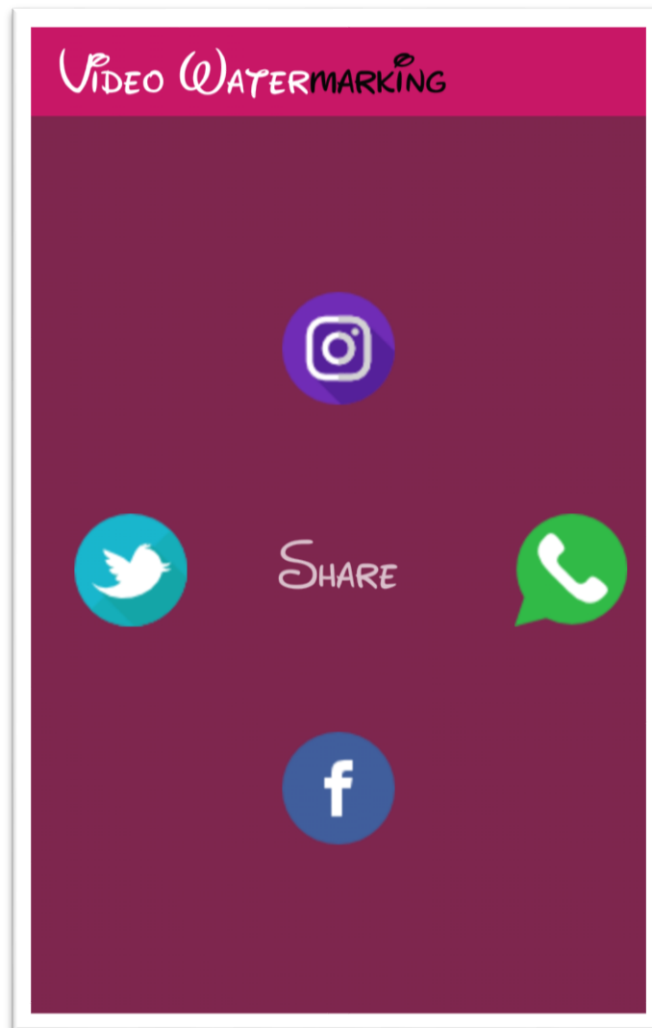
Fig 8.6 My Creation Screen



8.7 SHARE

Share your contents.

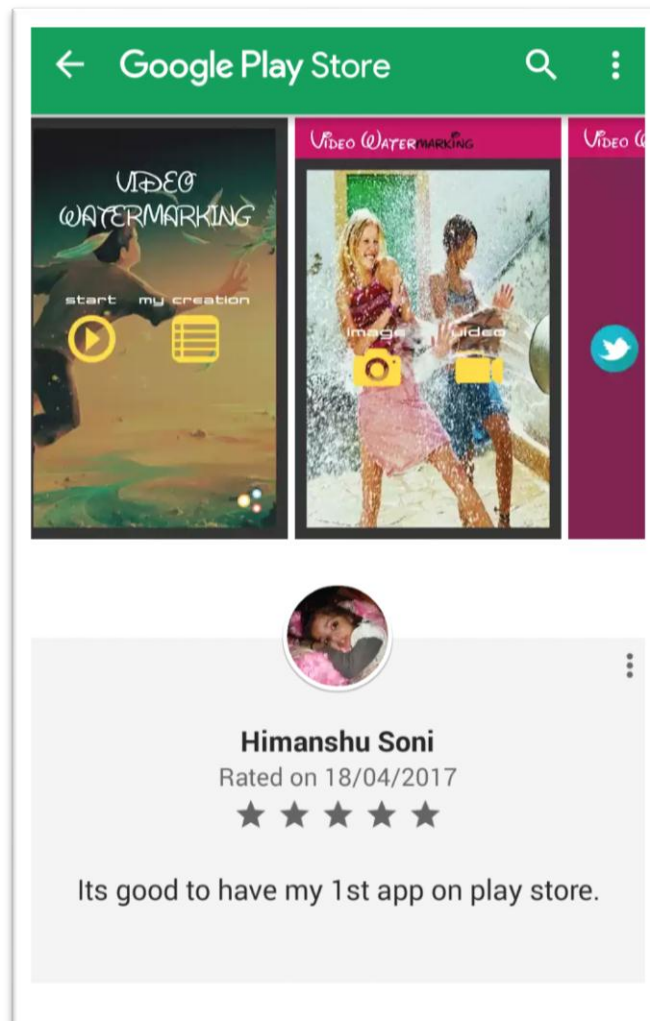
Fig 8.7 Share Screen



8.8 RATE

Provide rating to app.

Fig 8.8 Rate Screen



CHAPTER 9

FEATURES

Our app provides following features mainly:

1. Free of cost.
2. Easily accessible on play store.
3. Simplicity and attractive.
4. User can share and publish their content on social sites.
5. User can create their content without internet connectivity.
6. Small in size.
7. Faster than other similar apps.

Now user can protect their original video content, claim copyright, use it for authentication and verification purpose

This app is based on video processing using watermarks. Many app allow video creation and design using innovative design and graphic techniques. In order to prevent these videos to be reused without proper copyright purchase, watermarking is used. Watermarks are water stamps inflicted on images in order to display the entire videos with all its graphic features on public internet networks yet prevent these videos to be reused without proper copyrights. Today's its quite common to use video on public networks for personal use without purchasing proper copyrights for the video. In order to prevent such cases and avoid the reuse and reconstruction of images without copyright purchase the video watermarking techniques are used. A video watermark or water stamp is inflicted on the desired video. This water mark cannot be removed from the video easily. If removed it also takes away a part of video graphics. This video water marking app allows us to inflict watermarks on videos in order to prevent them from being copied for unauthorized use, yet display the entire video on public networks.

CHAPTER 10

CONCLUSION

The study of the watermark technology has become active since mid-1990s, and some technologies are already adopted in practical applications as a product or as proprietary services for enterprises.

Although this is a relatively new technology area, it quickly becomes a practical and effective solution in some application areas, and has great potential for some other areas as well.

The key to the successful implementation is to understand the advantages and the limitations of the watermark technology, and to use the watermark technology as a complimentary element to the existing security elements.

Video sequencing is a collection of consecutive and equally time spaced still images. Apparently any image watermarking technique can be extended to watermark videos, but in reality video watermarking techniques needs to meet other challenges. Watermarked video sequences are very much susceptible to pirate attacks such as frame averaging, frame swapping, statistical analysis, digital-analog (AD/DA) conversion, and lossy compressions. And proper encoding algorithm required as implemented in this project.

Hence for following behavior this solution can be used:

1. To protect ownership rights, mark this solution.
2. Provide security from attacks of your digital video contents.
3. Save and share your video on Facebook, WhatsApp, Instagram and Twitter.

CHAPTER 11

FUTURE ENHANCEMENTS

Table 11.1 Future enhancements

Features	Present	Future
Platform	Android	iOS
API	API level 19(Kitkat) Android 4.4 and above	-
Text Watermarking	Not Present	Can be done
Filters on Video and Image	Limited like position	Can be enhanced like color, effect, border padding etc.
Processing or Transcoding Speed	Currently works on FFMPEG library and is slows	Some alternative and paid library can be used to enhance speed.

CHAPTER 12

CODING

manifest.xml

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <activity
            android:name=".Home"
```

```

        android:launchMode="singleTask">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".WatermarkActivity" />
    <activity android:name=".AndroidExplorer" />
    <activity
        android:name=".ProgressBarWithNotification"
        android:configChanges="orientation|screenSize"
        android:screenOrientation="portrait" />
    <activity
        android:name=".Video"
        android:configChanges="orientation|screenSize"
        />
    <activity android:name=".Share"
        ></activity>
</application>

</manifest>

```

home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/home"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.projectwatermark.Home">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/home_page"
        android:id="@+id/imageView"
        android:scaleType="fitXY"
        android:alpha="0.6"
    />

    <TextView
        android:text="VIDEO WATERMARKING"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textWatermark"
        android:textAlignment="center"
        android:textStyle="bold"
```

```
    android:textSize="38dp"
    android:textColor="@color/white"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="50dp"
/>
```

```
<TableLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center">
```

```
<TableRow>
    <TextView
        android:text="Start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txtStart"
        android:textColor="@color/white"
        android:textSize="14dp"
        android:layout_column="1"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="30dp"
        android:textAlignment="center"
    />
```

```
<TextView
    android:text="My Creation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtCreation"
    android:textAlignment="center"
```

```
        android:textColor="@color/white"
        android:textSize="14dp"
        android:layout_column="2"
    />
```

```
</TableRow>
```

```
<TableRow>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnStart"
    android:src="@drawable/start_not_touched"
    android:background="@drawable/roundcorner"
    android:layout_column="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="30dp"
    android:layout_marginTop="12dp"
/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnMyWork"
    android:src="@drawable/mycreation_not_touched"
    android:background="@drawable/roundcorner"
    android:layout_column="2"
    android:layout_marginTop="12dp"
/>
```

```

</TableRow>
</TableLayout>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/rate_nt_touch"
    android:id="@+id/btnRate"
    android:layout_gravity="left|bottom"
    android:layout_marginLeft="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/roundcorner"/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/share_not_clicked"
    android:id="@+id/btnShare"
    android:layout_gravity="right|bottom"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="20dp"
    android:background="@drawable/roundcorner"/>
<com.google.android.gms.ads.AdView
    android:id="@+id/adView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="400dp"
    ads:adSize="BANNER"
    ads:adUnitId="@string/banner_ad_unit_id">
</com.google.android.gms.ads.AdView>
</FrameLayout>

```

Home.java

```
package com.projectwatermark;

import android.*;
import android.Manifest;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdView;
```

```

import com.google.android.gms.ads.MobileAds;
import com.projectwatermark.util.PlayOrShare;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Random;
import android.content.res.AssetManager;

public class Home extends AppCompatActivity implements View.OnTouchListener{
    ImageButton btShare, btStart, btCreation, btRate;
    TextView txStart, txCreation;
    String folderPath;
    File folder;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getSupportActionBar().hide();

        setContentView(R.layout.home);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new
String[]{ android.Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
            requestPermissions(new
String[]{ android.Manifest.permission.READ_EXTERNAL_STORAGE}, 1);

```



```

        requestPermissions(new String[]{android.Manifest.permission.INTERNET}, 1);
        requestPermissions(new String[]{android.Manifest.permission.WAKE_LOCK},
1);
    }

```

```

/*For Advertisement */
MobileAds.initialize(getApplicationContext(), "ca-app-pub-
4827826624748292/1726877160");
AdView mAdView = (AdView) findViewById(R.id.adView);
AdRequest adRequest = new AdRequest.Builder().build();
mAdView.loadAd(adRequest);

```

```

TextView label1=(TextView)findViewById(R.id.textWatermark);
btStart=((ImageButton)findViewById(R.id.btnStart));
btCreation=((ImageButton)findViewById(R.id.btnMyWork));
btShare=((ImageButton)findViewById(R.id.btnShare));
btRate=((ImageButton)findViewById(R.id.btnRate));
txStart=(TextView)findViewById(R.id.txtStart);
txCreation=(TextView)findViewById(R.id.txtCreation);

```

```

Typeface fontSpaceman1=Typeface.createFromAsset(getAssets(),"spaceman.ttf");
Typeface fontSpaceman2=Typeface.createFromAsset(getAssets(),"GodOfWar.ttf");
Typeface fontSpaceman3=Typeface.createFromAsset(getAssets(),"New Walt
Disney.ttf");

```

```

txStart.setTypeface(fontSpaceman1);
txCreation.setTypeface(fontSpaceman1);
label1.setTypeface(fontSpaceman3);

```

```

        btStart.setOnTouchListener(this);
        btCreation.setOnTouchListener(this);
        btShare.setOnTouchListener(this);
        btRate.setOnTouchListener(this);
        folderPath = createFolder();
    }
    public void myCreationClicked()
    {
        Intent i=new Intent(this,AndroidExplorer.class);
        startActivity(i);
        overridePendingTransition(R.anim.slide_from_right, R.anim.slide_to_left);
        new PlayOrShare(0);
    }

    public void startClicked() {
        Intent intent = new Intent(this,WatermarkActivity.class);
        startActivity(intent);
        overridePendingTransition(R.anim.slide_from_right, R.anim.slide_to_left);
    }

    public void shareClicked(){
        Intent i=new Intent(this,AndroidExplorer.class);
        startActivity(i);
        overridePendingTransition(R.anim.slide_from_right, R.anim.slide_to_left);
        new PlayOrShare(1);
    }

    private String createFolder()
    {
        String extStorageDirectory = Environment.getExternalStorageDirectory().toString();
        folder = new File(extStorageDirectory, "/Video Watermark/Output");
    }

```

```

if(!folder.exists())
{
    folder.mkdirs();
}
return folder.getPath().toString();
}

```

@Override

```

public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN: {
            if(btStart == v) {
                btStart.setImageResource(R.drawable.start_touched);
                txStart.setTextSize(18);
            }
            if (btCreation==v){
                btCreation.setImageResource(R.drawable.mycreation_touched);
                txCreation.setTextSize(18);
            }
            if (btShare==v){
                btShare.setImageResource(R.drawable.share_clicked);
            }
            if (btRate==v){
                btRate.setImageResource(R.drawable.rate_touch);
            }
            break;
        }
        case MotionEvent.ACTION_UP:

```

```

        if(btStart == v) {
            startClicked();
            resetStartButton();
        }
        if (btCreation==v){
            myCreationClicked();
            resetStartButton();
            finish();
        }
        if (btShare==v){
            shareClicked();
            resetStartButton();
        }
        if(btRate==v){
            rateClicked();
            resetStartButton();
        }
        break;
        // Your action here on button click

    case MotionEvent.ACTION_CANCEL: {
        break;
    }
}
return true;
}

public void rateClicked(){
    Uri uri = Uri.parse("market://details?id=" + getPackageName());
    Intent goToMarket = new Intent(Intent.ACTION_VIEW, uri);

```

```

// To count with Play market backstack, After pressing back button,
// to taken back to our application, we need to add following flags to intent.
goToMarket.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY |
    Intent.FLAG_ACTIVITY_NEW_DOCUMENT |
    Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
try {
    startActivity(goToMarket);
} catch (ActivityNotFoundException e) {
    startActivity(new
Intent(Intent.ACTION_VIEW,Uri.parse("http://play.google.com/store/apps/details?id="
+ getPackageName())));
}
}

@Override
public void onBackPressed() {
    new AlertDialog.Builder(this)
        .setIcon(android.R.drawable.ic_dialog_alert)
        .setTitle("Closing App")
        .setMessage("Are you sure you want to say bye?")
        .setPositiveButton("Yes", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String extStorageDirectory =
Environment.getExternalStorageDirectory().toString();
                folder = new File(extStorageDirectory, "/Video Watermark/temp");
                finish();
            }
        })
        .setNegativeButton("No", null)

```

```

        .show();
    }

    public void resetStartButton(){
        Handler handler = new Handler();
        handler.postDelayed(new Runnable(){
            @Override
            public void run() {
                btStart.setImageResource(R.drawable.start_not_touched);
                txStart.setTextSize(14);

                btCreation.setImageResource(R.drawable.mycreation_not_touched);
                txCreation.setTextSize(14);
                btRate.setImageResource(R.drawable.rate_nt_touch);
                btShare.setImageResource(R.drawable.share_not_clicked);
            }
        }, 200);
    }
}

```

watermark.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
    tools:context="com.projectwatermark.WatermarkActivity"
    android:id="@+id/scrollView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true" >
    <FrameLayout
        android:id="@+id/activity_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_vertical_margin"
        android:paddingRight="@dimen/activity_vertical_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
    >
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/sum"
        android:scaleType="fitXY"
        android:alpha="1.0"
    />
    <LinearLayout
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_marginTop="180dp"
        android:layout_gravity="center_horizontal"
```

```

        android:padding="30dp"
    >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="50dp"
        android:drawableBottom="@drawable/p_not_touched"
        android:background="@drawable/roundcorner"
        android:id="@+id/bt_imgWatermark"
        android:drawablePadding="5sp"
        android:textColor="#FFFFFF"
        android:text="Image"
        android:textSize="14sp"
        android:textAllCaps="false"></Button>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawableBottom="@drawable/v_not_touched"
        android:background="@drawable/roundcorner"
        android:id="@+id/bt_txtWatermark"
        android:drawablePadding="5sp"
        android:textColor="#FFFFFF"
        android:text="Video"
        android:textSize="14sp"
        android:textAllCaps="false"></Button>
</LinearLayout><![CDATA[
]]>
</FrameLayout>
</ScrollView>

```


WatermarkActivity.java

```
package com.projectwatermark;

import android.app.Activity;
import android.content.ContentUris;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Typeface;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.provider.DocumentsContract;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.TextView;
import android.widget.VideoView;

public class WatermarkActivity extends AppCompatActivity implements
View.OnTouchListener{

    public String pathImage=null;
    public String pathVideo=null;
```

```

Uri uriImage = null;
public String m_Text = "Hello";
private static final int READ_REQUEST_CODE1 = 41;
private static final int READ_REQUEST_CODE2 = 42;
VideoView videoView;
Button buttonImg, buttonTxt;
FrameLayout frameLayout;
StickerView iv_sticker;
Uri videoUri;
static int a=0,b=0;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.watermark);
    createCutomActionBarTitle();
    buttonImg=(Button)findViewById(R.id.bt_imgWatermark);
    buttonTxt=(Button)findViewById(R.id.bt_txtWatermark);
    buttonImg.setOnClickListener(this);
    buttonTxt.setOnClickListener(this);
    a=0;
    b=0;
    Typeface fontSpaceman1=Typeface.createFromAsset(getAssets(),"spaceman.ttf");
    buttonTxt.setTypeface(fontSpaceman1);
    buttonImg.setTypeface(fontSpaceman1);
}
@Override
public void onBackPressed() {
    a=0;
    b=0;
    Intent i=new Intent(this,Home.class);
    startActivity(i);
}

```

```

        finish();
        overridePendingTransition(R.anim.slide_from_left, R.anim.slide_to_right);
    }

```

```

private void createCutomActionBarTitle(){
    this.getSupportActionBar().setDisplayShowCustomEnabled(true);
    this.getSupportActionBar().setDisplayShowTitleEnabled(false);
    LayoutInflator inflater = LayoutInflater.from(this);
    View v = inflater.inflate(R.layout.custom_action_bar, null);
    Typeface tf = Typeface.createFromAsset(getAssets(),"New Walt Disney.ttf");
    ((TextView)v.findViewById(R.id.titleFragment1)).setTypeface(tf);
    ((TextView)v.findViewById(R.id.titleFragment2)).setTypeface(tf);
    //assign the view to the actionbar
    this.getSupportActionBar().setCustomView(v);
}

```

@Override

```

public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN: {
            if(buttonImg == v) {
                Drawable bottom = getResources().getDrawable(R.drawable.p_touched);
                buttonImg.setCompoundDrawablesWithIntrinsicBounds(null, null , null,
bottom);
                buttonImg.setTextSize(12);
            }
            if (buttonTxt==v){
                Drawable bottom = getResources().getDrawable(R.drawable.v_touched);
                buttonTxt.setCompoundDrawablesWithIntrinsicBounds(null, null , null,
bottom);
                buttonTxt.setTextSize(12);
            }
        }
    }
}

```

```

    }
    break;
}
case MotionEvent.ACTION_UP:
    if(buttonImg == v) {
        resetButton();
        imageButtonClicked();
    }
    if (buttonTxt==v){
        resetButton();
        videoButtonClicked();
    }
    break;
case MotionEvent.ACTION_CANCEL: {
    break;
}
}
return true;
}

public void resetButton(){
    Handler handler = new Handler();
    handler.postDelayed(new Runnable(){
        @Override
        public void run() {
            Drawable bottom1 = getResources().getDrawable(R.drawable.p_not_touched);
            buttonImg.setCompoundDrawablesWithIntrinsicBounds(null, null , null,
bottom1);
            buttonImg.setTextSize(14);

            Drawable bottom2 = getResources().getDrawable(R.drawable.v_not_touched);
            buttonTxt.setCompoundDrawablesWithIntrinsicBounds(null, null , null,

```

```

bottom2);
        buttonTxt.setTextSize(14);
    }
}, 200);
}

public void videoButtonClicked(){
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    intent.setType("video/*");
    startActivityForResult(intent, READ_REQUEST_CODE1);
    overridePendingTransition(R.anim.slide_from_right, R.anim.slide_to_left);
}

public void imageButtonClicked() {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    intent.setType("image/*");
    startActivityForResult(intent, READ_REQUEST_CODE2);
    overridePendingTransition(R.anim.slide_from_right, R.anim.slide_to_left);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent resultData) {
    if (requestCode == READ_REQUEST_CODE2 && resultCode ==
Activity.RESULT_OK) {
        if (resultData != null) {
            uriImage = resultData.getData();
            pathImage=ExactFilePath.getPath(this,uriImage);
            a=1;
        }
    }
}

```

```

    }
    if (requestCode == READ_REQUEST_CODE1 && resultCode ==
Activity.RESULT_OK) {

        if (resultData != null) {
            videoUri = resultData.getData();
            pathVideo=ExactFilePath.getPath(this,videoUri);
            b=1;
        }
    }

    if(a==1 && b==1){
        Intent i=new Intent(this,Video.class);
        i.putExtra("videoPath",pathVideo);
        i.putExtra("imagePath",pathImage);
        startActivity(i);
    }
}

```

ExactFilePath.java

```
class ExactFilePath
{
    public static String getPath(final Context context, final Uri uri)
    {
        //check here to KITKAT or new version
        // DocumentProvider
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT &&
DocumentsContract.isDocumentUri(context, uri)) {

            // ExternalStorageProvider
            if (isExternalStorageDocument(uri)) {
                final String docId = DocumentsContract.getDocumentId(uri);
                final String[] split = docId.split(":");
                final String type = split[0];

                if ("primary".equalsIgnoreCase(type)) {
                    return Environment.getExternalStorageDirectory() + "/" + split[1];
                }
            }
            // DownloadsProvider
            else if (isDownloadsDocument(uri)) {

                final String id = DocumentsContract.getDocumentId(uri);
                final Uri contentUri = ContentUris.withAppendedId(
                    Uri.parse("content://downloads/public_downloads"), Long.valueOf(id));

                return getDataColumn(context, contentUri, null, null);
            }
        }
    }
}
```

```

// MediaProvider
else if (isMediaDocument(uri)) {
    final String docId = DocumentsContract.getDocumentId(uri);
    final String[] split = docId.split(":");
    final String type = split[0];

    Uri contentUri = null;
    if ("image".equals(type)) {
        contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
    } else if ("video".equals(type)) {
        contentUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
    } else if ("audio".equals(type)) {
        contentUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
    }

    final String selection = "_id=?";
    final String[] selectionArgs = new String[] {
        split[1]
    };
    return getDataColumn(context, contentUri, selection, selectionArgs);
}

// MediaStore (and general)
else if ("content".equalsIgnoreCase(uri.getScheme())) {
    // Return the remote address
    if (isGooglePhotosUri(uri))
        return uri.getLastPathSegment();
    return getDataColumn(context, uri, null, null);
}

// File
else if ("file".equalsIgnoreCase(uri.getScheme())) {

```



```

        return uri.getPath();
    }
    return null;
}

/**
 * Get the value of the data column for this Uri. This is useful for
 * MediaStore Uris, and other file-based ContentProviders.
 *
 * @param context The context.
 * @param uri The Uri to query.
 * @param selection (Optional) Filter used in the query.
 * @param selectionArgs (Optional) Selection arguments used in the query.
 * @return The value of the _data column, which is typically a file path.
 */
public static String getDataColumn(Context context, Uri uri, String selection,
                                   String[] selectionArgs) {

    Cursor cursor = null;
    final String column = "_data";
    final String[] projection = {
        column
    };

    try {
        cursor = context.getContentResolver().query(uri, projection, selection,
selectionArgs,
        null);
        if (cursor != null && cursor.moveToFirst()) {
            final int index = cursor.getColumnIndexOrThrow(column);
            return cursor.getString(index);
        }
    }
}

```

```

        }
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return null;
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is ExternalStorageProvider.
 */
public static boolean isExternalStorageDocument(Uri uri) {
    return "com.android.externalstorage.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is DownloadsProvider.
 */
public static boolean isDownloadsDocument(Uri uri) {
    return "com.android.providers.downloads.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is MediaProvider.
 */
public static boolean isMediaDocument(Uri uri) {
    return "com.android.providers.media.documents".equals(uri.getAuthority());
}

```

```
/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is Google Photos.
 */
public static boolean isGooglePhotosUri(Uri uri) {
    return "com.google.android.apps.photos.content".equals(uri.getAuthority());
}
}
```

activity_video.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_video"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.projectwatermark.Video">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
    >
        <FrameLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/frameVideo"
        >
            <VideoView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/videoV"
                android:layout_gravity="center"
            />
        </FrameLayout>
    </RelativeLayout>
    <SlidingDrawer
        android:id="@+id/SlidingDrawer"
        android:layout_width="wrap_content"
        android:layout_height="180dip"
        android:content="@+id/contentLayout"
```

```

        android:handle="@+id/slideButton"
        android:orientation="vertical"
        android:layout_alignParentBottom="true"
    >
    <ImageButton
        android:id="@+id/slideButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:focusable="true"
        android:src="@drawable/swipe"
        android:rotation="90"
        android:onClick="slide"
        android:background="@drawable/roundcorner"
    >
</ImageButton>
<LinearLayout
    android:id="@+id/contentLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="5dip" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:layout_weight="1">
        <ImageButton
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/btnPlay"
            android:src="@drawable/p"

```

```

        android:layout_weight="0.5"
        android:background="@drawable/roundcorner"/>
<SeekBar
    android:layout_width="220dp"
    android:id="@+id/seekBar"
    android:layout_height="match_parent"
    android:layout_gravity="center"
/>

<ImageButton
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/pause"
    android:background="@drawable/roundcorner"
    android:id="@+id/btnPause"
    android:layout_weight="0.5"/>
</LinearLayout>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtLabel"
    android:text="Choose corner to place logo"
    android:textSize="20sp"
    android:textColor="@color/white"
    android:layout_gravity="center"
    android:textStyle="bold|italic"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="120dp"
    android:paddingTop="15dp"
    android:orientation="horizontal"
    android:layout_weight="1">

```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnTL"
    android:src="@drawable/arrow"
    android:rotation="45"
    android:background="@drawable/roundcorner"
    android:layout_weight="1"/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnBR"
    android:background="@drawable/roundcorner"
    android:src="@drawable/arrow"
    android:rotation="225"
    android:layout_weight="1"/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnC"
    android:background="@drawable/roundcorner"
    android:src="@drawable/circle"
    android:layout_weight="1"/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnBL"
    android:src="@drawable/arrow"
    android:rotation="315"
    android:background="@drawable/roundcorner"
```

```
android:layout_weight="1"/>
```

```
<ImageButton
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/btnTR"
```

```
    android:src="@drawable/arrow"
```

```
    android:rotation="135"
```

```
    android:background="@drawable/roundcorner"
```

```
    android:layout_weight="1"
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</SlidingDrawer>
```

```
</RelativeLayout>
```


Video.java

```
package com.projectwatermark;

import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.PixelFormat;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.Drawable;
import android.graphics.drawable.StateListDrawable;
import android.media.Image;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.MediaController;
import android.widget.RelativeLayout;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.VideoView;
import java.io.File;
import java.io.FileOutputStream;
import java.util.Random;

public class Video extends AppCompatActivity implements
View.OnTouchListener,SeekBar.OnSeekBarChangeListener {
    FrameLayout frmLayout;
    StickerView iv_sticker;
    VideoView videoView;
    ImageButton bTopLeft,bTopRight,bCenter,bBottomLeft,bBottomRight, bPlay,
bPause;
    int padX,padY;
    Bitmap bmp;
    View vv;
    SeekBar seekBar;
    Handler mHandler = new Handler();
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_video);
    createCutomActionBarTitle();
    videoView=(VideoView)findViewById(R.id.videoV);
    frmLayout=(FrameLayout)findViewById(R.id.frameVideo);
    bTopLeft=(ImageButton)findViewById(R.id.btnTL);
    bTopRight=(ImageButton)findViewById(R.id.btnTR);
    bCenter=(ImageButton)findViewById(R.id.btnC);
    bBottomLeft=(ImageButton)findViewById(R.id.btnBL);
    bBottomRight=(ImageButton)findViewById(R.id.btnBR);
    bPlay=(ImageButton)findViewById(R.id.btnPlay);
    bPause=(ImageButton)findViewById(R.id.btnPause);
    seekBar=(SeekBar)findViewById(R.id.seekBar);
    Typeface tf = Typeface.createFromAsset(getAssets(),"New Walt Disney.ttf");
    ((TextView)findViewById(R.id.txtLabel)).setTypeface(tf);
    bTopLeft.setOnTouchListener(this);
    bTopRight.setOnTouchListener(this);
    bCenter.setOnTouchListener(this);
    bBottomLeft.setOnTouchListener(this);
    bBottomRight.setOnTouchListener(this);
    bPlay.setOnTouchListener(this);
    bPause.setOnTouchListener(this);
    Bundle bundle = getIntent().getExtras();
    Uri uriV=Uri.parse(bundle.getString("videoPath"));
    Media m=null;
    showVideo(uriV);
    String imagePath = bundle.getString("imagePath");
    m= new Media(imagePath, bundle.getString("videoPath"));
    File imgFile = new File(imagePath);
    bmp = BitmapFactory.decodeFile(imgFile.getAbsolutePath());
}

```

```

        iv_sticker = new StickerView(this);
        iv_sticker.setImageBitmap bmp);
        frmLayout.addView(iv_sticker);
        new PositionWatermarkCorner(5);
        new VideoFrame(getWindowManager().getDefaultDisplay().getWidth());
        seekBar.setOnSeekBarChangeListener(this);
        seekBar.getProgressDrawable().setColorFilter(Color.WHITE,
PorterDuff.Mode.SRC_IN);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {
            seekBar.getThumb().setColorFilter(Color.parseColor("#C51162"),
PorterDuff.Mode.SRC_IN);
        }
    }
}

private void createCutomActionBarTitle(){
    this.getSupportActionBar().setDisplayShowCustomEnabled(true);
    this.getSupportActionBar().setDisplayShowTitleEnabled(false);
    LayoutInflater inflater = LayoutInflater.from(this);
    View v = inflater.inflate(R.layout.custom_action_bar, null);
    Typeface tf = Typeface.createFromAsset(getAssets(),"New Walt Disney.ttf");
    ((TextView)v.findViewById(R.id.titleFragment1)).setTypeface(tf);
    ((TextView)v.findViewById(R.id.titleFragment2)).setTypeface(tf);
    this.getSupportActionBar().setCustomView(v);
}

@Override
public void onBackPressed() {
    frmLayout.removeView(iv_sticker);
    Intent i=new Intent(this,WatermarkActivity.class);
    startActivity(i);
    finish();
    return;
}

```

```

    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {

        if(event.getAction()==MotionEvent.ACTION_DOWN){
            if(v==bPlay){
                videoView.start();
                updateProgressBar();
                bPlay.setImageResource(R.drawable.pl);
            }else if(v==bPause){
                videoView.pause();
                updateProgressBar();
                bPause.setImageResource(R.drawable.pa);
            }
        }

        if(event.getAction()==MotionEvent.ACTION_UP)
        {
            if(v==bTopLeft){
                frmLayout.removeView(iv_sticker);
                iv_sticker.setX(0);
                iv_sticker.setY(0);
                new PositionWatermarkCorner(1);
                frmLayout.addView(iv_sticker);
            }else if(v==bTopRight){
                frmLayout.removeView(iv_sticker);
                iv_sticker.setX(videoView.getWidth()-iv_sticker.getWidth());
                iv_sticker.setY(0);
                new PositionWatermarkCorner(2);
                frmLayout.addView(iv_sticker);
            }else if(v==bCenter){

```

```

        frmLayout.removeView(iv_sticker);
        iv_sticker = new StickerImageView(this);
        iv_sticker.setImageBitmap bmp;
        new PositionWatermarkCorner(5);
        frmLayout.addView(iv_sticker);
    }else if(v==bBottomLeft){
        frmLayout.removeView(iv_sticker);
        iv_sticker.setX(0);
        iv_sticker.setY(videoView.getHeight()-iv_sticker.getHeight());
        new PositionWatermarkCorner(3);
        frmLayout.addView(iv_sticker);
    }else if(v==bBottomRight){
        frmLayout.removeView(iv_sticker);
        iv_sticker.setX(videoView.getWidth()-iv_sticker.getWidth());
        iv_sticker.setY(videoView.getHeight()-iv_sticker.getHeight());
        new PositionWatermarkCorner(4);
        frmLayout.addView(iv_sticker);
    }else if(v==bPlay){
        resetButton();
    }else if(v==bPause){
        resetButton();
    }
}
return true;
}

public void showVideo(Uri uri){
    videoView.setVideoURI(uri);
    videoView.requestFocus();
    videoView.pause();
}

```

```

public void resetButton(){
    Handler handler = new Handler();
    handler.postDelayed(new Runnable(){
        @Override
        public void run() {
            bPlay.setImageResource(R.drawable.p);
            bPause.setImageResource(R.drawable.pause);
        }
    }, 200);
}

private void updateProgressBar() {
    mHandler.postDelayed(updateTimeTask, 100);
}

private Runnable updateTimeTask = new Runnable() {
    public void run() {
        seekBar.setProgress(videoView.getCurrentPosition());
        seekBar.setMax(videoView.getDuration());
        mHandler.postDelayed(this, 100);
    }
};

@Override
public void onProgressChanged(SeekBar seekbar, int progress,boolean fromTouch) {

}

@Override
public void onStartTrackingTouch(SeekBar seekbar) {

```

```
        mHandler.removeCallbacks(updateTimeTask);
    }
    @Override
    public void onStopTrackingTouch(SeekBar seekbar) {
        mHandler.removeCallbacks(updateTimeTask);
        videoView.seekTo(seekbar.getProgress());
        updateProgressBar();
    }
}
```


activity_progress_bar_with_notification.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_progress_bar_with_notification"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.projectwatermark.ProgressBarWithNotification">
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:alpha="0.75"
            android:background="@color/white"
            android:theme="@style/Theme.AppCompat.NoActionBar"
        />
        <TextView
            android:text="Do you really want to apply changes?"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/txtMsg"
            android:textColor="@color/black"
            android:gravity="center"
```

```

        android:textStyle="bold"
        android:textSize="32sp"/>
    <Button
        android:text="Yes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnYes"
        android:drawableTop="@drawable/u"
        android:background="@drawable/roundcorner"
        android:layout_marginTop="300dp"
        android:layout_marginLeft="50dp"
        android:textAllCaps="false"
        android:textColor="@color/black"
        android:textStyle="bold"
    />
    <Button
        android:text="No"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnNo"
        android:layout_marginTop="300dp"
        android:layout_marginLeft="200dp"
        android:drawableTop="@drawable/d"
        android:background="@drawable/roundcorner"
        android:textAllCaps="false"
        android:textColor="@color/black"
        android:textStyle="bold"
    />
</FrameLayout>
</ScrollView>

```

ProgressBarWithNotification.java

```
package com.projectwatermark;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Typeface;
import android.graphics.drawable.ColorDrawable;
import android.os.Environment;
import android.os.Handler;
import android.os.Message;
import android.os.PowerManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.netcompss.ffmpeg4android.CommandValidationException;
import com.netcompss.ffmpeg4android.GeneralUtils;
import com.netcompss.ffmpeg4android.Prefs;
import com.netcompss.ffmpeg4android.ProgressCalculator;
```

```

import com.netcompss.loader.LoadJNI;

import java.io.File;
import java.io.FileOutputStream;
import java.util.Random;
import java.util.Timer;

public class ProgressBarWithNotification extends AppCompatActivity {

    public ProgressDialog progressBar;
    String fname;
    String workFolder = null;
    String demoVideoFolder = null;
    String demoVideoPath = null;
    String vkLogPath = null;
    LoadJNI vk;
    private final int STOP_TRANSCODING_MSG = -1;
    private final int FINISHED_TRANSCODING_MSG = 0;
    private boolean commandValidationFailedFlag = false;
    String inputVideo="";
    String image="";
    String outputVideo="";
    String overlayPositionCommand="overlay=(W-w)/2:(H-h)/2";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        Log.i(Prefs.TAG, "onCreate ffmpeg4android ProgressBarExample");
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_progress_bar_with_notification);
        createCutomActionBarTitle();
        demoVideoFolder = Environment.getExternalStorageDirectory().getAbsolutePath()

```

```

+ "/Video watermark/temp/";
    demoVideoPath = demoVideoFolder + "in.mp4";
    Log.i(Prefs.TAG, getString(R.string.app_name) + " version: " +
GeneralUtils.getVersionName(getApplicationContext()) );
    Button yes = (Button)findViewById(R.id.btnYes);
    Button no = (Button)findViewById(R.id.btnNo);
    no.startAnimation(AnimationUtils.loadAnimation(this, R.anim.bounce1));
    yes.startAnimation(AnimationUtils.loadAnimation(this, R.anim.bounce2));

    yes.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v){
            Log.i(Prefs.TAG, "run clicked.");

            File imgFile = new File(Media.getImage());
            Bitmap imageBitmap =
BitmapFactory.decodeFile(imgFile.getAbsolutePath());
            // int w = (int) (Size.getWidth() * imageBitmap.getWidth() /
VideoFrame.getWidth());
            int w=(int)Size.getWidth();
            Bitmap newBitmap = Bitmap.createScaledBitmap(imageBitmap, w, w,
false);
            image = saveImage(newBitmap);
            inputVideo=Media.video;
            Random generator = new Random();
            int n = 1000;
            n = generator.nextInt(n);
            outputVideo = "/sdcard/Video watermark/Output/output_" + n +
inputVideo.substring(inputVideo.indexOf('.'),inputVideo.length());
            Log.wtf("mmmmmmmmmmmmmm",outputVideo);
            if(PositionWatermarkCorner.getCorner()==1) {
                overlayPositionCommand = "overlay=5:5";
            }
        }
    });

```

```

    }else if(PositionWatermarkCorner.getCorner()==2){
        overlayPositionCommand="overlay=W-w-5:5";
    }else if(PositionWatermarkCorner.getCorner()==3){
        overlayPositionCommand="overlay=5:H-h-5";
    }else if(PositionWatermarkCorner.getCorner()==4){
        overlayPositionCommand="overlay=W-w-5:H-h-5";
    }else{
        overlayPositionCommand="overlay=(W-w)/2:(H-h)/2";
    }
    runTranscoding();
}
});

```

```

no.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v){
        finish();
    }
});

```

```

workFolder = getApplicationContext().getFilesDir() + "/";
//Log.i(Prefs.TAG, "workFolder: " + workFolder);
vkLogPath = workFolder + "vk.log";
GeneralUtils.copyLicenseFromAssetsToSDIfNeeded(this, workFolder);
GeneralUtils.copyDemoVideoFromAssetsToSDIfNeeded(this, demoVideoFolder);
int rc = GeneralUtils.isLicenseValid(getApplicationContext(), workFolder);
Log.i(Prefs.TAG, "License check RC: " + rc);
}

```

```

public String saveImage(Bitmap finalBitmap) {
    String root = Environment.getExternalStorageDirectory().toString();
    File myDir = new File(root + "/Video Watermark/temp/");

```

```

myDir.mkdirs();
Random generator = new Random();
int n = 10000;
n = generator.nextInt(n);
fname = "Image-" + n + ".jpg";
File file = new File(myDir, fname);
if (file.exists()) file.delete();
try {
    FileOutputStream out = new FileOutputStream(file);
    finalBitmap.compress(Bitmap.CompressFormat.JPEG, 90, out);
    out.flush();
    out.close();

} catch (Exception e) {
    e.printStackTrace();
}
return root + "/Video Watermark/temp/"+fname;
}

private void createCutomActionBarTitle(){
    this.getSupportActionBar().setDisplayShowCustomEnabled(true);
    this.getSupportActionBar().setDisplayShowTitleEnabled(false);
    LayoutInflater inflater = LayoutInflater.from(this);
    View v = inflater.inflate(R.layout.custom_action_bar, null);
    Typeface tf = Typeface.createFromAsset(getAssets(),"New Walt Disney.ttf");
    ((TextView)v.findViewById(R.id.titleFragment1)).setTypeface(tf);
    ((TextView)v.findViewById(R.id.titleFragment2)).setTypeface(tf);
    this.getSupportActionBar().setCustomView(v);
    ((TextView)findViewById(R.id.txtMsg)).setTypeface(tf);
}

```

```

private void runTranscodingUsingLoader() {
    Log.i(Prefs.TAG, "runTranscodingUsingLoader started...");

    PowerManager powerManager =
(PowerManager)ProgressBarWithNotification.this.getSystemService(Activity.POWER_
SERVICE);
    PowerManager.WakeLock wakeLock =
powerManager.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,
"VK_LOCK");
    Log.d(Prefs.TAG, "Acquire wake lock");
    wakeLock.acquire();

    String complexCommand[];
    complexCommand = new String[]{"ffmpeg", "-y", "-i", inputVideo, "-strict",
"experimental", "-vf", "movie=" + image + " [watermark]; [in][watermark] " +
overlayPositionCommand + " [out]", "-r", "30", "-b", "15496k", "-vcodec", "mpeg4", "-
ab", "48000", "-ac", "2", "-ar", "22050", outputVideo};

    vk = new LoadJNI();
    try {
        vk.run(complexCommand, workFolder, getApplicationContext());
        Log.i(Prefs.TAG, "vk.run finished.");
        GeneralUtils.copyFileToFolder(vkLogPath, demoVideoFolder);

    } catch (CommandValidationException e) {
        Log.e(Prefs.TAG, "vk run exeption.", e);
        commandValidationFailedFlag = true;
        Toast.makeText(this, "Invalid media content. Please try
again.", Toast.LENGTH_LONG).show();

    } catch (Throwable e) {

```



```

        Log.e(Prefs.TAG, "vk run exeption.", e);
        Toast.makeText(this, "Invalid media content. Please try
again.", Toast.LENGTH_LONG).show();
    }
    finally {
        if (wakeLock.isHeld()) {
            wakeLock.release();
            Log.i(Prefs.TAG, "Wake lock released");
        }else{
            Log.i(Prefs.TAG, "Wake lock is already released, doing nothing");
        }
    }
}

// finished Toast
String rc = null;
if (commandValidationFailedFlag) {
    rc = "Command Vaidation Failed";
}
else {
    rc = GeneralUtils.getReturnCodeFromLog(vkLogPath);
}
final String status = rc;
ProgressBarWithNotification.this.runOnUiThread(new Runnable() {
    public void run() {
        if(status.equalsIgnoreCase("Transcoding Status: Finished OK")){
            Toast.makeText(ProgressBarWithNotification.this, "Operation Successfully
Performed", Toast.LENGTH_LONG).show();
            Intent i=new Intent(ProgressBarWithNotification.this,Home.class);
            startActivity(i);
        }
        else

```

```

        Toast.makeText(ProgressBarWithNotification.this, status,
Toast.LENGTH_LONG).show();
        if (status.equals("Transcoding Status: Failed")) {
            Toast.makeText(ProgressBarWithNotification.this, "Oops Error Occured.
Check: " + vkLogPath + " for more information.", Toast.LENGTH_LONG).show();
        }
    }
});
}

```

```

private Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        Log.i(Prefs.TAG, "Handler got message");
        if (progressBar != null) {
            progressBar.dismiss();
            // stopping the transcoding native
            if (msg.what == STOP_TRANSCODING_MSG) {
                Log.i(Prefs.TAG, "Got cancel message, calling fexit");
                vk.fExit(getApplicationContext());
            }
        }
    }
};

```

```

public void runTranscoding() {
    progressBar = new
ProgressDialog(ProgressBarWithNotification.this,R.style.Theme_MyDialog);
    progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
    // progressBar.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressBar.setTitle("Applying Watermark Please Wait!");
}

```

```

progressBar.setMax(100);
progressBar.setProgress(0);
progressBar.setIndeterminate(false);
progressBar.setCancelable(false);
progressBar.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        handler.sendMessage(STOP_TRANSCODING_MSG);
    }
});

progressBar.show();

new Thread() {
    public void run() {
        Log.d(Prefs.TAG, "Worker started");
        try {
            //sleep(5000);
            runTranscodingUsingLoader();
            handler.sendMessage(FINISHED_TRANSCODING_MSG);

        } catch (Exception e) {
            Log.e("threadmessage", e.getMessage());
        }
    }
}.start();

// Progress update thread
new Thread() {
    ProgressCalculator pc = new ProgressCalculator(vkLogPath);

```

```

public void run() {
    Log.d(Prefs.TAG,"Progress update started");
    int progress = -1;
    try {
        while (true) {
            sleep(300);
            progress = pc.calcProgress();
            if (progress != 0 && progress < 100) {
                progressBar.setProgress(progress);
            }
            else if (progress == 100) {
                Log.i(Prefs.TAG, "==== progress is 100, exiting Progress update
thread");
                pc.initCalcParamsForNextInter();

                break;

            }
        }

        } catch(Exception e) {
            Log.e("threadmessage",e.getMessage());
        }
    }.start();
}
}

```

explorer_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:background="@drawable/border_shadow"
>
    <ImageView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_marginTop="10dp"
        android:id="@+id/textIcon"
        android:src="@drawable/media"
        android:background="@drawable/roundcorner"
    />
    <TextView
        android:id="@+id/textR"
        android:padding="10dp"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@color/black"
        android:typeface="monospace"
    />
</LinearLayout>
```

explorer.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:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/white">
    <TextView
        android:id="@+id/path"
        android:layout_width="fill_parent"
        android:layout_height="80dp"
        android:layout_marginTop="2dp"
        android:layout_marginBottom="2dp"
        android:background="@color/myColor"
        android:textColor="@color/white"
        android:textSize="18sp"
        android:typeface="serif"
        android:textStyle="bold"
        android:gravity="center"
        android:padding="4dp"
        android:lineSpacingMultiplier="1.8" />
    <ListView
        android:id="@+id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/white" />
    <TextView
        android:id="@android:id/empty"
        android:layout_width="fill_parent"
```

```

        android:layout_height="wrap_content"
        android:text="No Data"
    />
</LinearLayout>

```

custom_action_bar.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/actionBar"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:paddingLeft="5dp"
    android:paddingTop="7dp"
    android:orientation="horizontal" >
    <TextView
        android:id="@+id/titleFragment1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Video Water"
        android:textSize="30sp"
        android:textColor="@color/white" />
    <TextView
        android:id="@+id/titleFragment2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:text="marking"
        android:textColor="@color/black"/>
</LinearLayout>

```

CustomListAdapter.java

```
package com.projectwatermark;

/**
 * Created by HDMP on 16-Feb-17.
 */

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.projectwatermark.util.PlayOrShare;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class CustomListAdapter extends BaseAdapter {

    private List<String> item = null;
    private List<String> path = null;
    private String root = "/sdcard/Video Watermark/Output";
    private TextView myPath;
    Context context;
```



```

int imageId;
private static LayoutInflater inflater = null;

public CustomListAdapter(AndroidExplorer exActivity, int prgmImages) {
    // TODO Auto-generated constructor stub
    myPath = (TextView) exActivity.findViewById(R.id.path);
    getDir(root);
    context = exActivity;
    imageId = prgmImages;
    inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
}

@Override
public int getCount() {
    // TODO Auto-generated method stub
    return item.size();
}

@Override
public Object getItem(int position) {
    // TODO Auto-generated method stub
    return position;
}

@Override
public long getItemId(int position) {
    // TODO Auto-generated method stub
    return position;
}

```

```

public class Holder {
    TextView tv;
    ImageView img;
}

@Override

public View getView(final int position, final View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    Holder holder = new Holder();
    View rowView;
    rowView = inflater.inflate(R.layout.explorer_row, null);
    holder.tv = (TextView) rowView.findViewById(R.id.textR);
    holder.img = (ImageView) rowView.findViewById(R.id.textIcon);
    holder.tv.setText(item.get(position));
    holder.img.setImageResource(imageId);

    rowView.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            File file = new File(path.get(position));
            if(PlayOrShare.getFlag()==0) {
                Intent i = new Intent(Intent.ACTION_VIEW,Uri.fromFile(file));
                i.setDataAndType(Uri.fromFile(file),"video/*");
                context.startActivity(i);

            }else{
                Intent i=new Intent(context,Share.class);
                i.putExtra("uri",""+Uri.fromFile(file));
                context.startActivity(i);
            }
        }
    })
}

```

```

        }
    });
    return rowView;
}

private void getDir(String dirPath) {
    myPath.setText("Location: " + dirPath);
    item = new ArrayList<String>();
    path = new ArrayList<String>();
    File f = new File(dirPath);
    File[] files = f.listFiles();
    if (!dirPath.equals(root)) {
        item.add(root);
        path.add(root);
        item.add("../");
        path.add(f.getParent());
    }
    for (int i = 0; i < files.length; i++) {
        File file = files[i];
        path.add(file.getPath());
        if (!file.isDirectory())
        {
            String s=file.getName();
            if(s.endsWith(".mp4") || s.endsWith(".mkv") || s.endsWith(".3gp") ||
s.endsWith(".avi") || s.endsWith(".MP4") || s.endsWith(".MKV") || s.endsWith(".3GP") ||
s.endsWith(".AVI"))
                item.add(s);
        }
    }
}
}

```

AndroidExplorer.java

```
package com.projectwatermark;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.TextView;
import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdView;
import com.google.android.gms.ads.MobileAds;
import java.io.Serializable;

public class AndroidExplorer extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.explorer);
        CustomListAdapter adapter = new CustomListAdapter(this, R.drawable.media);
        ListView list = (ListView) findViewById(R.id.list);
        list.setAdapter(adapter);
        TextView path=(TextView)findViewById(R.id.path);
        Typeface fontSpaceman=Typeface.createFromAsset(getAssets(),"SF Fedora
Titles.ttf");
        path.setTypeface(fontSpaceman);
    }

    @Override
    public void onBackPressed() {
```

```
Intent i=new Intent(this,Home.class);
startActivity(i);
finish();
overridePendingTransition(R.anim.slide_from_left, R.anim.slide_to_right);
}
}
```

activity_pop_up.xml

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

    android:fillViewport="true"
    tools:context="com.projectwatermark.Share">

    <FrameLayout

        android:layout_width="match_parent"
        android:layout_height="match_parent">
        android:id="@+id/frmCon"

    <ImageView

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/myColor"
        android:alpha="0.5"/>

    <TextView

        android:layout_width="match_parent"
        android:layout_height="511dp"
        android:gravity="center"
        android:text="Share"
        android:id="@+id/txtSt"
        android:textSize="32sp"/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnInstagram"
    android:src="@drawable/insta_nt"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="100dp"
    android:background="@drawable/roundcorner"
```

```
/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/roundcorner"
    android:src="@drawable/twit_nt"
    android:id="@+id/btnTwitter"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="30dp"
```

```
/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:id="@+id/btnWhatsapp"
    android:background="@drawable/roundcorner"
    android:src="@drawable/whats_nt"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="280dp"/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:id="@+id/btnFb"
    android:layout_marginTop="350dp"
    android:layout_gravity="center_horizontal"
    android:background="@drawable/roundcorner"
    android:src="@drawable/fb_nt" />
```

```
</FrameLayout>
```

```
</ScrollView>
```


Share.java

```
package com.projectwatermark;

import android.animation.Animator;
import android.content.Intent;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.graphics.Typeface;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.view.animation.TranslateAnimation;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import static android.animation.Animator.*;

public class Share extends AppCompatActivity implements View.OnTouchListener {
```

```

ImageButton fb,twitter,whatsapp,instagram;
Animation anim;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    /* getSupportActionBar().hide();
    getWindow().setFlags(
        WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);*/
    setContentView(R.layout.activity_pop_up);
    createCutomActionBarTitle();
    whatsapp=(ImageButton)findViewById(R.id.btnWhatsapp);
    whatsapp.setOnTouchListener(this);
    instagram=(ImageButton)findViewById(R.id.btnInstagram);
    instagram.setOnTouchListener(this);
    fb=(ImageButton)findViewById(R.id.btnFb);
    fb.setOnTouchListener(this);
    twitter=(ImageButton)findViewById(R.id.btnTwitter);
    twitter.setOnTouchListener(this);
    instagram.startAnimation(AnimationUtils.loadAnimation(this,
R.anim.slide_from_left_view));
    twitter.startAnimation(AnimationUtils.loadAnimation(this,
R.anim.slide_from_right_view));
    whatsapp.startAnimation(AnimationUtils.loadAnimation(this,
R.anim.slide_from_left_view));
    fb.startAnimation(AnimationUtils.loadAnimation(this,
R.anim.slide_from_right_view));
}

private void createCutomActionBarTitle(){

```

```

this.getSupportActionBar().setDisplayShowCustomEnabled(true);
this.getSupportActionBar().setDisplayShowTitleEnabled(false);
LayoutInflater inflater = LayoutInflater.from(this);
View v = inflater.inflate(R.layout.custom_action_bar, null);
Typeface tf = Typeface.createFromAsset(getAssets(),"New Walt Disney.ttf");
((TextView)v.findViewById(R.id.titleFragment1)).setTypeface(tf);
((TextView)v.findViewById(R.id.titleFragment2)).setTypeface(tf);
//assign the view to the actionbar
((TextView)findViewById(R.id.txtSt)).setTypeface(tf);
this.getSupportActionBar().setCustomView(v);
}
@Override
public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN: {
            if(whatsapp == v) {
                whatsapp.setImageResource(R.drawable.whats_t);
            }
            if(instagram==v){
                instagram.setImageResource(R.drawable.insta_t);
            }
            if(fb==v){
                fb.setImageResource(R.drawable.fb_t);
            }
            if(twitter==v){
                twitter.setImageResource(R.drawable.twit_t);
            }

            break;
        }
        case MotionEvent.ACTION_UP:

```

```

String packageInfo="",url="",msg="";
if(whatsapp == v) {
    packageInfo="com.whatsapp";
    url="https://www.whatsapp.com/";
    msg="WhatsApp not Installed";

}
if(instagram == v) {
    packageInfo="com.instagram.android";
    url="https://www.instagram.com/";
    msg="Instagram not Installed";

}
if(twitter == v) {
    packageInfo="com.twitter.android";
    url="https://twitter.com/";
    msg="Twitter not Installed";
}
if(fb == v) {
    packageInfo="com.facebook.katana";
    url="https://www.facebook.com/";
    msg="Facebook not Installed";
}
resetButton();

Bundle bundle = getIntent().getExtras();
Uri uriV=Uri.parse(bundle.getString("uri"));
PackageManager pm=getPackageManager();
try {

    Intent intent = new Intent(Intent.ACTION_SEND);

```

```

        intent.setType("video/*");
        PackageInfo info=pm.getPackageInfo(packageInfo,
PackageManager.GET_META_DATA);
        intent.setPackage(packageInfo);
        intent.putExtra(Intent.EXTRA_STREAM, uriV);
        startActivity(Intent.createChooser(intent, "Share with"));

    } catch (PackageManager.NameNotFoundException e) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
        Uri uri = Uri.parse(url);
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
    break;
// Your action here on button click

case MotionEvent.ACTION_CANCEL: {
    whatsapp.clearAnimation();
    instagram.clearAnimation();
    twitter.clearAnimation();
    fb.clearAnimation();
    break;
}
}
return true;
}

public void resetButton(){
    Handler handler = new Handler();
    handler.postDelayed(new Runnable(){
        @Override

```

```
public void run() {  
    whatsapp.setImageResource(R.drawable.whats_nt);  
    instagram.setImageResource(R.drawable.insta_nt);  
    fb.setImageResource(R.drawable.fb_nt);  
    twitter.setImageResource(R.drawable.twit_nt);  
  
    }  
    }, 200);  
}  
}
```

Location.java

```
package com.projectwatermark;
```

```
public class Location {  
    public static float x ;  
    public static float y ;  
    public Location( float xx, float yy){  
        x=xx;  
        y=yy;  
    }  
    public static float getX() {  
        return x;  
    }  
  
    public static float getY() {  
        return y;  
    }  
}
```

Media.java

```
public class Media {  
    public static String image;  
    public static String video;  
  
    public Media(String img,String vid){  
        image=img;  
        video=vid;  
    }  
    public static String getImage(){
```

```

        return image;
    }
    public static String getVideo(){
        return video;
    }
}

```

Size.java

```

public class Size{

    public static float w ;
    public static float h ;

    public Size(float ww,float hh){
        w=ww;
        h=hh;
    }
    public static float getWidth(){
        return w;
    }
    public static float getHeight(){
        return h;
    }
}

```

VideoFrame.java

```

public class VideoFrame {
    public static int width=5;

```



```

    public VideoFrame(int w){
        width=w;
    }
    public static int getWidth(){
        return width;
    }
}

```

PositionWatermarkCorner.java

```

public class PositionWatermarkCorner {
    public static int corner=5;
    public PositionWatermarkCorner(int c){
        corner=c;
    }
    public static int getCorner(){
        return corner;    }
}

```

StickerView.java

```

package com.projectwatermark;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.util.AttributeSet;
import android.view.View;
import android.widget.ImageView;

```

```

public class StickerView extends StickerView {

    private String owner_id;
    public ImageView iv_main;
    public StickerView(Context context) {
        super(context);
    }

    public StickerView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public StickerView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
    }

    public void setOwnerId(String owner_id){
        this.owner_id = owner_id;
    }

    public String getOwnerId(){
        return this.owner_id;
    }

    @Override
    public View getMainView() {
        if(this.iv_main == null) {
            this.iv_main = new ImageView(getContext());
            this.iv_main.setScaleType(ImageView.ScaleType.FIT_XY);
        }
        return iv_main;
    }
}

```

```

    }
    public void setImageBitmap(Bitmap bmp){
        this.iv_main.setImageBitmap(bmp);
    }
    public void setImageResource(int res_id){
        this.iv_main.setImageResource(res_id);
    }
    public void setImageDrawable(Drawable drawable){
this.iv_main.setImageDrawable(drawable); }
        public Bitmap getImageBitmap(){ return
((BitmapDrawable)this.iv_main.getDrawable()).getBitmap() ; }

    }

```

StickerView.java

```

package com.projectwatermark;

import android.content.Context;
import android.content.Intent;
import android.content.res.Resources;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.util.AttributeSet;
import android.util.DisplayMetrics;

```

```
import android.util.Log;
import android.util.Size;
import android.view.Gravity;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.Toast;
```

```
public abstract class StickerView extends FrameLayout{

    public static final String TAG = "com.knef.stickerView";
    private BorderView iv_border;
    private ImageView iv_scale;
    private ImageView iv_save;
    public double changeWidth=0,changeHeight=0;

    // For scalling
    private float this_orgX = -1, this_orgY = -1;
    private float scale_orgX = -1, scale_orgY = -1;
    private double scale_orgWidth = -1, scale_orgHeight = -1;
    // For rotating
    private float rotate_orgX = -1, rotate_orgY = -1, rotate_newX = -1, rotate_newY = -1;
    // For moving
    private float move_orgX = -1, move_orgY = -1;

    private double centerX, centerY;

    private final static int BUTTON_SIZE_DP = 25;
```

```
private final static int SELF_SIZE_DP = 50;
```

```
public StickerView(Context context) {  
    super(context);  
    init(context);  
}
```

```
public StickerView(Context context, AttributeSet attrs) {  
    super(context, attrs);  
    init(context);  
}
```

```
public StickerView(Context context, AttributeSet attrs, int defStyle) {  
    super(context, attrs, defStyle);  
    init(context);  
}
```

```
private void init(final Context context){  
  
    this.iv_border = new BorderView(context);  
    this.iv_scale = new ImageView(context);  
    this.iv_save = new ImageView(context);  
    this.iv_scale.setImageResource(R.drawable.zoominout);  
    this.iv_save.setImageResource(R.drawable.sst);  
    this.setTag("DraggableViewGroup");  
    this.iv_border.setTag("iv_border");  
    this.iv_scale.setTag("iv_scale");  
    this.iv_save.setTag("iv_delete");  
    int margin = convertDpToPixel(BUTTON_SIZE_DP+20, getContext())/2;
```

```
int size = convertDpToPixel(SELF_SIZE_DP+20, getContext());
```

```
LayoutParams this_params =  
    new LayoutParams(  
        size,  
        size  
    );  
this_params.gravity = Gravity.CENTER;
```

```
LayoutParams iv_main_params =  
    new LayoutParams(  
        ViewGroup.LayoutParams.MATCH_PARENT,  
        ViewGroup.LayoutParams.MATCH_PARENT  
    );  
iv_main_params.setMargins(margin,margin,margin,margin);
```

```
LayoutParams iv_border_params =  
    new LayoutParams(  
        ViewGroup.LayoutParams.MATCH_PARENT,  
        ViewGroup.LayoutParams.MATCH_PARENT  
    );  
iv_border_params.setMargins(margin,margin,margin,margin);
```

```
LayoutParams iv_scale_params =  
    new LayoutParams(  
        convertDpToPixel(BUTTON_SIZE_DP, getContext()),  
        convertDpToPixel(BUTTON_SIZE_DP, getContext())  
    );  
iv_scale_params.gravity = Gravity.BOTTOM | Gravity.RIGHT;
```

```

LayoutParams iv_save_params =
    new LayoutParams(
        convertDpToPixel(BUTTON_SIZE_DP, getContext()),
        convertDpToPixel(BUTTON_SIZE_DP, getContext())
    );
iv_save_params.gravity = Gravity.TOP | Gravity.RIGHT;
this.setLayoutParams(this_params);
this.addView(getMainView(), iv_main_params);
this.addView(iv_border, iv_border_params);
this.addView(iv_scale, iv_scale_params);
this.addView(iv_save, iv_save_params);
this.setOnTouchListener(mTouchListener);
this.iv_scale.setOnTouchListener(mTouchListener);
this.iv_save.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        if(StickerView.this.getParent()!=null){
            Location l=new Location(StickerView.this.getX(),StickerView.this.getY());
            com.projectwatermark.Size s=new
com.projectwatermark.Size(iv_border.getWidth(),iv_border.getHeight());
            Intent i=new Intent(context,ProgressBarWithNotification.class);
            context.startActivity(i);
        }
    }
});
changeHeight=StickerView.this.getLayoutParams().height;
changeWidth=StickerView.this.getLayoutParams().width;

}

protected abstract View getMainView();
private OnTouchListener mTouchListener = new OnTouchListener() {

```

```

@Override
public boolean onTouch(View view, MotionEvent event) {

    if(view.getTag().equals("iv_scale")){
        switch (event.getAction()) {

            case MotionEvent.ACTION_DOWN:
                Log.v(TAG, "iv_scale action down");

                this_orgX = StickerView.this.getX();
                this_orgY = StickerView.this.getY();

                scale_orgX = event.getRawX();
                scale_orgY = event.getRawY();
                scale_orgWidth = StickerView.this.getLayoutParams().width;
                scale_orgHeight = StickerView.this.getLayoutParams().height;

                rotate_orgX = event.getRawX();
                rotate_orgY = event.getRawY();

                centerX = StickerView.this.getX()+
                    ((View)StickerView.this.getParent()).getX()+
                    (float)StickerView.this.getWidth()/2;

                int result = 0;
                int resourceId = getResources().getIdentifier("status_bar_height",
                    "dimen", "android");
                if (resourceId > 0) {
                    result = getResources().getDimensionPixelSize(resourceId);
                }
            }
        }
    }
}

```



```

double statusBarHeight = result;
centerY = StickerView.this.getY()+
        ((View)StickerView.this.getParent()).getY()+
        statusBarHeight+
        (float)StickerView.this.getHeight()/2;

break;
case MotionEvent.ACTION_MOVE:
    Log.v(TAG, "iv_scale action move");

    rotate_newX = event.getRawX();
    rotate_newY = event.getRawY();

    double angle_diff = Math.abs(
        Math.atan2(event.getRawY() - scale_orgY , event.getRawX() -
scale_orgX)
        - Math.atan2(scale_orgY - centerY, scale_orgX -
centerX))*180/Math.PI;

    Log.v(TAG, "angle_diff: "+angle_diff);

    double length1 = getLength(centerX, centerY, scale_orgX, scale_orgY);
    double length2 = getLength(centerX, centerY, event.getRawX(),
event.getRawY());

    int size = convertDpToPixel(SELF_SIZE_DP, getContext());
    if(length2 > length1
        && (angle_diff < 25 || Math.abs(angle_diff-180)<25)
        ) {
        //scale up
        double offsetX = Math.abs(event.getRawX() - scale_orgX);

```

```

double offsetY = Math.abs(event.getRawY() - scale_orgY);
double offset = Math.max(offsetX, offsetY);
offset = Math.round(offset);
StickerView.this.getLayoutParams().width += offset;
StickerView.this.getLayoutParams().height += offset;

changeHeight=StickerView.this.getHeight();
changeWidth=StickerView.this.getWidth();

onScaling(true);
}else if(length2 < length1
    && (angle_diff < 25 || Math.abs(angle_diff-180)<25)
    && StickerView.this.getLayoutParams().width > size/2
    && StickerView.this.getLayoutParams().height > size/2) {
//scale down
double offsetX = Math.abs(event.getRawX() - scale_orgX);
double offsetY = Math.abs(event.getRawY() - scale_orgY);
double offset = Math.max(offsetX, offsetY);
offset = Math.round(offset);
StickerView.this.getLayoutParams().width -= offset;
StickerView.this.getLayoutParams().height -= offset;

changeHeight=StickerView.this.getHeight();
changeWidth=StickerView.this.getWidth();

onScaling(false);
}

double angle = Math.atan2(event.getRawY() - centerY, event.getRawX() -
centerX) * 180 / Math.PI;

```

```

        Log.v(TAG, "log angle: " + angle);
        Log.v(TAG, "getRotation(): " + getRotation());
        onRotating();
        rotate_orgX = rotate_newX;
        rotate_orgY = rotate_newY;
        scale_orgX = event.getRawX();
        scale_orgY = event.getRawY();
        postInvalidate();
        requestLayout();
        break;
    case MotionEvent.ACTION_UP:
        Log.v(TAG, "iv_scale action up");
        break;
    }
}
return true;
}
};

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
}

private double getLength(double x1, double y1, double x2, double y2){
    return Math.sqrt(Math.pow(y2-y1, 2)+Math.pow(x2-x1, 2));
}

private float[] getRelativePos(float absX, float absY){
    Log.v("ken", "getRelativePos getX:"+((View)this.getParent()).getX());
    Log.v("ken", "getRelativePos getY:"+((View)this.getParent()).getY());
}

```

```

float [] pos = new float[]{
    absX-((View)this.getParent()).getX(),
    absY-((View)this.getParent()).getY()
};
Log.v(TAG, "getRelativePos absY:"+absY);
Log.v(TAG, "getRelativePos relativeY:"+pos[1]);
return pos;
}

public void setControlItemsHidden(boolean isHidden){
    if(isHidden) {
        iv_border.setVisibility(View.INVISIBLE);
        iv_scale.setVisibility(View.INVISIBLE);
        iv_save.setVisibility(View.INVISIBLE);
        // iv_flip.setVisibility(View.INVISIBLE);
    }else{
        iv_border.setVisibility(View.VISIBLE);
        iv_scale.setVisibility(View.VISIBLE);
        iv_save.setVisibility(View.VISIBLE);
        // iv_flip.setVisibility(View.VISIBLE);
    }
}

protected void onScaling(boolean scaleUp){ }

protected void onRotating(){ }

private class BorderView extends View{

    public BorderView(Context context) {
        super(context);
    }
}

```

```

public BorderView(Context context, AttributeSet attrs) {
    super(context, attrs);
}

public BorderView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    // Draw sticker border

    LayoutParams params = (LayoutParams)this.getLayoutParams();

    Log.v(TAG,"params.leftMargin: "+params.leftMargin);

    Rect border = new Rect();
    border.left = (int)this.getLeft()-params.leftMargin;
    border.top = (int)this.getTop()-params.topMargin;
    border.right = (int)this.getRight()-params.rightMargin;
    border.bottom = (int)this.getBottom()-params.bottomMargin;
    Paint borderPaint = new Paint();
    borderPaint.setStrokeWidth(6);
    borderPaint.setColor(Color.WHITE);
    borderPaint.setStyle(Paint.Style.STROKE);
    canvas.drawRect(border, borderPaint);

}
}

```

```
private static int convertDpToPixel(float dp, Context context){  
    Resources resources = context.getResources();  
    DisplayMetrics metrics = resources.getDisplayMetrics();  
    float px = dp * (metrics.densityDpi / 160f);  
    return (int)px;  
}  
}
```

REFERENCES

1. CHAN Pik-Wah, Techniques for Secure Multimedia Creation and Delivery, Thesis, 2005
2. Ponni (a) Sathya Sethuraman, Ramakrishnan Srinivasan. Survey of Digital Video Watermarking Techniques and Its Applications. Engineering Science. Vol. 1, No. 1, 2016, pp. 22-27. doi: 10.11648/j.es.20160101.14
3. Wei-Meng Lee, Beginning Android™ 4 Application Development, New Delhi: John Wiley, 2014.
4. Herbert Schildt, The Complete Reference Java™ Ninth Edition, New Delhi, McGraw Hill Education, 2014