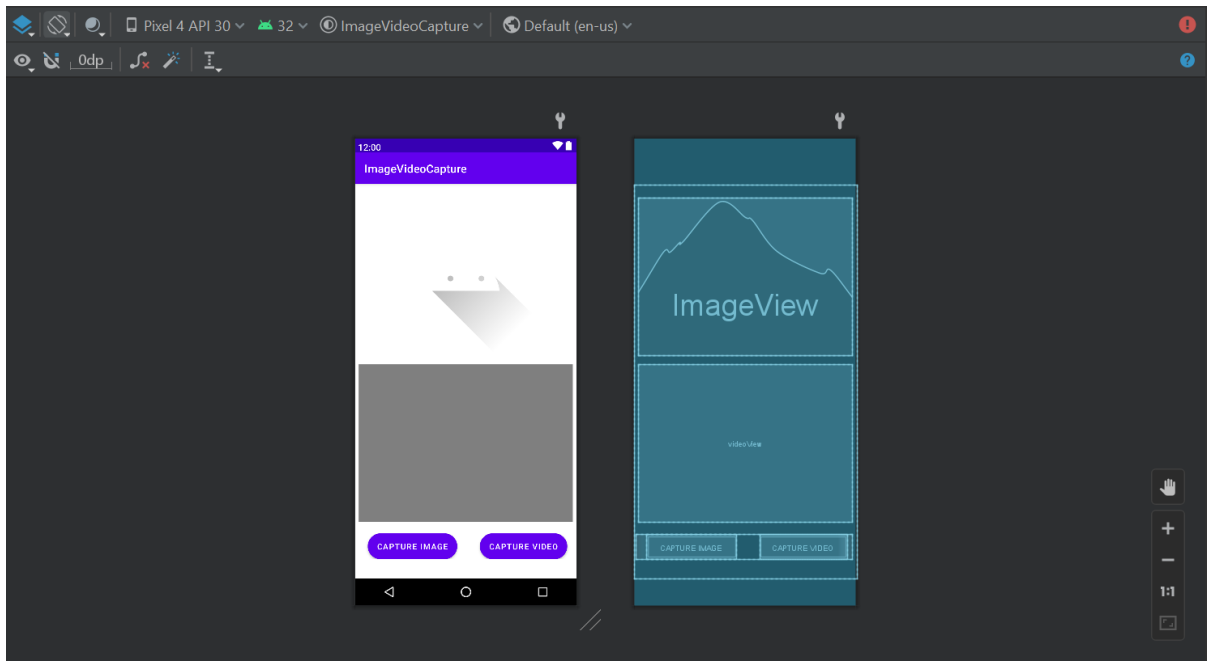


```
activity_main.xml x MainActivity.java x
Code Split Design
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <ImageView
10         android:id="@+id/imageView"
11         android:layout_width="377dp"
12         android:layout_height="235dp"
13         android:layout_marginStart="16dp"
14         android:layout_marginTop="24dp"
15         android:layout_marginEnd="16dp"
16         android:contentDescription="T000"
17         app:layout_constraintEnd_toEndOf="parent"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:srcCompat="@drawable/ic_launcher_foreground"
21         tools:ignore="VectorDrawableCompat,ImageContrastCheck" />
22
23     <VideoView
24         android:id="@+id/videoView"
25         android:layout_width="371dp"
26         android:layout_height="248dp"
27         android:layout_marginStart="16dp"
28         android:layout_marginTop="16dp"
29         android:layout_marginEnd="16dp"
30         app:layout_constraintEnd_toEndOf="parent"
31         app:layout_constraintHorizontal_bias="0.51"
32         app:layout_constraintStart_toStartOf="parent"
33         app:layout_constraintTop_toBottomOf="@+id/imageView" />
34
35     <LinearLayout
36         android:layout_width="409dp"
37         android:layout_height="50dp"
38         android:layout_marginStart="1dp"
39         android:layout_marginTop="8dp"
40         android:layout_marginEnd="1dp"
41         android:orientation="horizontal"
42         app:layout_constraintEnd_toEndOf="parent"
43         app:layout_constraintHorizontal_bias="0.0"
44         app:layout_constraintStart_toStartOf="parent"
45         app:layout_constraintTop_toBottomOf="@+id/videoView">
46
47         <Button
48             android:id="@+id/button_image"
49             android:background="@drawable/white_grey_border_bottom"
50             android:layout_width="wrap_content"
51             android:layout_height="wrap_content"
52             android:layout_marginStart="20dp"
53             android:layout_marginEnd="40dp"
54             android:layout_weight="0"
55             android:text="Capture Image" />
56
57         <Button
58             android:id="@+id/button_video"
59             android:background="@drawable/white_grey_border_bottom"
60             android:layout_width="wrap_content"
61             android:layout_height="wrap_content"
62             android:layout_weight="0"
63             android:text="Capture Video" />
64     </LinearLayout>
65 </androidx.constraintlayout.widget.ConstraintLayout>
```

```
activity_main.xml x MainActivity.java x
Code Split Design
23 <VideoView
24     android:id="@+id/videoView"
25     android:layout_width="371dp"
26     android:layout_height="248dp"
27     android:layout_marginStart="16dp"
28     android:layout_marginTop="16dp"
29     android:layout_marginEnd="16dp"
30     app:layout_constraintEnd_toEndOf="parent"
31     app:layout_constraintHorizontal_bias="0.51"
32     app:layout_constraintStart_toStartOf="parent"
33     app:layout_constraintTop_toBottomOf="@+id/imageView" />
34
35 <LinearLayout
36     android:layout_width="409dp"
37     android:layout_height="50dp"
38     android:layout_marginStart="1dp"
39     android:layout_marginTop="8dp"
40     android:layout_marginEnd="1dp"
41     android:orientation="horizontal"
42     app:layout_constraintEnd_toEndOf="parent"
43     app:layout_constraintHorizontal_bias="0.0"
44     app:layout_constraintStart_toStartOf="parent"
45     app:layout_constraintTop_toBottomOf="@+id/videoView">
46
47     <Button
48         android:id="@+id/button_image"
49         android:background="@drawable/white_grey_border_bottom"
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:layout_marginStart="20dp"
53         android:layout_marginEnd="40dp"
54         android:layout_weight="0"
55         android:text="Capture Image" />
56
57     <Button
58         android:id="@+id/button_video"
59         android:background="@drawable/white_grey_border_bottom"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:layout_weight="0"
63         android:text="Capture Video" />
64 </LinearLayout>
65 </androidx.constraintlayout.widget.ConstraintLayout>
```

```
activity_main.xml x MainActivity.java x
Code Split Design
41 android:orientation="horizontal"
42 app:layout_constraintEnd_toEndOf="parent"
43 app:layout_constraintHorizontal_bias="0.0"
44 app:layout_constraintStart_toStartOf="parent"
45 app:layout_constraintTop_toBottomOf="@+id/videoView">
46
47 <Button
48     android:id="@+id/button_image"
49     android:background="@drawable/white_grey_border_bottom"
50     android:layout_width="wrap_content"
51     android:layout_height="wrap_content"
52     android:layout_marginStart="20dp"
53     android:layout_marginEnd="40dp"
54     android:layout_weight="0"
55     android:text="Capture Image" />
56
57 <Button
58     android:id="@+id/button_video"
59     android:background="@drawable/white_grey_border_bottom"
60     android:layout_width="wrap_content"
61     android:layout_height="wrap_content"
62     android:layout_weight="0"
63     android:text="Capture Video" />
64 </LinearLayout>
65 </androidx.constraintlayout.widget.ConstraintLayout>
```



```
24
25 public class MainActivity extends AppCompatActivity {
26
27     private Button buttonImage;
28     private Button buttonVideo;
29
30     private VideoView videoView;
31     private ImageView imageView;
32
33     private static final int REQUEST_ID_READ_WRITE_PERMISSION = 99;
34     private static final int REQUEST_ID_IMAGE_CAPTURE = 100;
35     private static final int REQUEST_ID_VIDEO_CAPTURE = 101;
36
37     @Override
38     protected void onCreate(Bundle savedInstanceState) {
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_main);
41
42         this.buttonImage = (Button) this.findViewById(R.id.button_image);
43         this.buttonVideo = (Button) this.findViewById(R.id.button_video);
44         this.videoView = (VideoView) this.findViewById(R.id.videoView);
45         this.imageView = (ImageView) this.findViewById(R.id.imageView);
46
47         this.buttonImage.setOnClickListener(new Button.OnClickListener() {
48             @Override
49             public void onClick(View v) { captureImage(); }
50         });
51     }
52 }
```

```

MainActivity.java x
53
54     this.buttonVideo.setOnClickListener(new Button.OnClickListener() {
55         @Override
56         public void onClick(View v) { askPermissionAndCaptureVideo(); }
59     });
60 }
61
62 private void captureImage() {
63
64     // Create an implicit intent, for image capture.
65     Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
66
67     // Start camera and wait for the results.
68     this.startActivityForResult(intent, REQUEST_ID_IMAGE_CAPTURE);
69 }
70
71 private void askPermissionAndCaptureVideo() {
72
73     // With Android Level >= 23, you have to ask the user
74     // for permission to read/write data on the device.
75     if (android.os.Build.VERSION.SDK_INT >= 23) {
76
77         // Check if we have read/write permission
78         int readPermission = ActivityCompat.checkSelfPermission( context: this,
79             Manifest.permission.READ_EXTERNAL_STORAGE);
80         int writePermission = ActivityCompat.checkSelfPermission( context: this,
81             Manifest.permission.WRITE_EXTERNAL_STORAGE);
82
83         if (writePermission != PackageManager.PERMISSION_GRANTED ||
84             readPermission != PackageManager.PERMISSION_GRANTED) {
85             // If don't have permission so prompt the user.
86             this.requestPermissions(
87                 new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
88                     Manifest.permission.READ_EXTERNAL_STORAGE},
89                 REQUEST_ID_READ_WRITE_PERMISSION
90             );
91             return;
92         }
93     }
94     this.captureVideo();
95 }
96
97 private void captureVideo() {
98     try {
99         // Create an implicit intent, for video capture.
100         Intent intent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
101
102         // The external storage directory.
103         File dir = Environment.getExternalStorageDirectory();
104         if (!dir.exists()) {
105             dir.mkdirs();
106         }
107         // file:///storage/emulated/0/myvideo.mp4

```

```

108         String savePath = dir.getAbsolutePath() + "/myvideo.mp4";
109         File videoFile = new File(savePath);
110         Uri videoUri = Uri.fromFile(videoFile);
111
112
113         intent.putExtra(MediaStore.EXTRA_OUTPUT, videoUri);
114         intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
115
116         this.startActivityForResult(intent, REQUEST_ID_VIDEO_CAPTURE); // (**)
117
118     } catch (Exception e) {
119         Toast.makeText(context: this, text: "Error capture video: " + e.getMessage(),
120             Toast.LENGTH_LONG).show();
121         e.printStackTrace();
122     }
123 }
124
125
126 // When you have the request results
127 @Override
128 public void onRequestPermissionsResult(int requestCode,
129     String permissions[], int[] grantResults) {
130
131     super.onRequestPermissionsResult(requestCode, permissions, grantResults);
132     //
133     switch (requestCode) {
134         case REQUEST_ID_READ_WRITE_PERMISSION: {
135
136             // Note: If request is cancelled, the result arrays are empty.
137             // Permissions granted (read/write).
138             if (grantResults.length > 1
139                 && grantResults[0] == PackageManager.PERMISSION_GRANTED
140                 && grantResults[1] == PackageManager.PERMISSION_GRANTED) {
141
142                 Toast.makeText(context: this, text: "Permission granted!", Toast.LENGTH_LONG).show();
143
144                 this.captureVideo();
145
146             }
147             // Cancelled or denied.
148             else {
149                 Toast.makeText(context: this, text: "Permission denied!", Toast.LENGTH_LONG).show();
150             }
151             break;
152         }
153     }
154 }
155
156 // When results returned
157 @Override
158 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
159     super.onActivityResult(requestCode, resultCode, data);
160

```

```

160     if (requestCode == REQUEST_ID_IMAGE_CAPTURE) {
161         if (resultCode == RESULT_OK) {
162             Bitmap bp = (Bitmap) data.getExtras().get("data");
163             this.imageView.setImageBitmap(bp);
164         } else if (resultCode == RESULT_CANCELED) {
165             Toast.makeText( context: this, text: "Action canceled", Toast.LENGTH_LONG).show();
166         } else {
167             Toast.makeText( context: this, text: "Action Failed", Toast.LENGTH_LONG).show();
168         }
169     } else if (requestCode == REQUEST_ID_VIDEO_CAPTURE) {
170         if (resultCode == RESULT_OK) {
171             Uri videoUri = data.getData();
172             Log.i( tag: "MyLog", msg: "Video saved to: " + videoUri);
173             Toast.makeText( context: this, text: "Video saved to:\n" +
174                 videoUri, Toast.LENGTH_LONG).show();
175             this.videoView.setVideoURI(videoUri);
176             this.videoView.start();
177         } else if (resultCode == RESULT_CANCELED) {
178             Toast.makeText( context: this, text: "Action Cancelled.",
179                 Toast.LENGTH_LONG).show();
180         } else {
181             Toast.makeText( context: this, text: "Action Failed",
182                 Toast.LENGTH_LONG).show();
183         }
184     }
185 }
186 }

```

