Discussion-5 How can you incorporate images, audio, and video into an Android app and handle related operations

Discussion Topic:

How can you incorporate images, audio, and video into an Android app and handle related operations?

My Post:

Hello Class,

Incorporating images, audio, and video into an Android app often involves implementing permissions to display images and play videos. This is usually necessary to meet the project's requirements or enhance the user's experience. This post provides an overview on how to integrate images, audio, and video into Android apps.

Permissions

To integrate images, audio, and video within an Android app, handling permissions is essential because multimedia functionalities require access to sensitive resources like the camera and external storage. Thus, understanding what the necessary permissions are and how to implement them is crucial. Additionally, before they can be implemented, the permissions have to be allowed by the user.

To implement a camera permission, for example, to capture or record videos, an app needs to request access to the camera (Android Developers. n.d.a). This request can be declared in the *AndroidManifest.xml* file as follows:

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

To implement storage permissions to access media files stored on the device, the storage-related permissions need to be declared, and the files must reside in the *ediaStore.Images*, *MediaStore.Video*, or *MediaStore.Audio* (Android Developers. n.d.b).
For apps used in Android 10 (API level 29) and higher, scoped storage is the recommended. With scoped storage, apps have access to their own app-specific directories and can access media files through the *MediaStore* API. For Android 9 and lower, or not wanting to use scoped storage, the *READ_EXTERNAL_STORAGE* (read only) and the *WRITE_EXTERNAL_STORAGE* (write only) permissions need to be declared in the AndroidManifest.xml file as follows:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Image, Audio, and Video

To display images using XML, the *ImageView* element is used to display images from various sources such as from drawable, local storage, and networks. To use *ImageView* with a drawable, simply drag the *ImageView* widget into your activity's layout, and then a pop-up dialog will appear, allowing you to choose from the available drawables (Rishu_mishra, 2025).

When using Composable, the *Image composable* provides a way to display images. You can load images from different sources, including drawables, bitmaps, and painters (Android Developers. n.d.c).

To play audio, the *MediaPlayer* class allows to play audio. It supports various audio sources, including local files, streams, and resources (Adityamshidlyali, 2024)
To play videos, the *VideoView* is a UI element allows playing video content. It supports videos from various sources, including local files, resources, and network URLs (Google Developers, n.d.).
Another alternative to *MediaPlayer* and *VideoView,* Jetpack Media3 is an Android library for media that enables apps to display rich audio and visual experiences (Android Developers. n.d.d).

To summarize, integrating into an Android app, images, audio, and video requires handling permissions and the use of different tools. Permissions like *CAMERA, READ_EXTERNAL_STORAGE*, and *WRITE_EXTERNAL_STORAGE* need to be declared in the AndroidManifest.xml file and allowed by the user. For displaying images, *ImageView* in XML layouts or *the Image composable* can be used for that purpose. Audio can be managed by using the *MediaPlayer* class, while for video playing, *VideoView* can be used. An alternative for audio and video playing is to use the Jetpack Media3 library.

-Alex

References:

Adityamshidlyali (2024, August 12). *MediaPlayer class in Android.* GeeksForGeeks. https://www.geeksforgeeks.org/mediaplayer-class-in-android/

Android Developers (n.d.a). *Capture an image*. Google. https://developer.android.com/media/camera/camerax/take-photo

Android Developers (n.d.b). *Access media files from shared storage*. Google. https://developer.android.com/training/data-storage/shared/media

Android Developers (n.d.c). *Add images to your Android app*. Google. https://developer.android.com/codelabs/basic-android-kotlin-compose-add-images#0

Android Developers. (n.d.d). *Introduction to Jetpack Media3*. Google. https://developer.android.com/media/media3

Google Developers (n.d.). 13.1: Playing video with VideoView. *Advanced Android development*. Google. https://google-developer-training.github.io/android-developer-advanced-course-practicals/

Rishu_mishra (2025, January 28). ImageView in Android with example. GeeksForGeeks. https://www.geeksforgeeks.org/imageview-in-android-with-example/