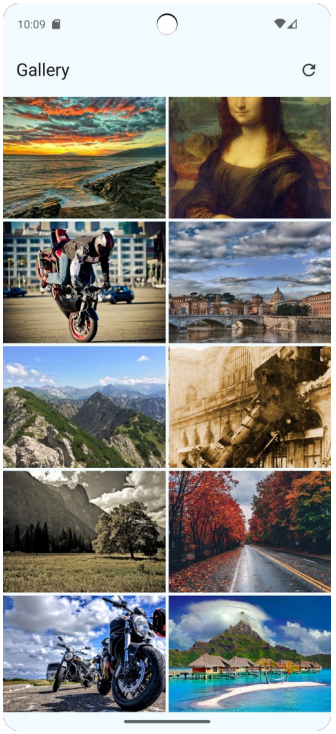
159.336 Assignment 2

Due 19th September 2025



For this assignment you need to write a simple gallery application for Android to view all the photos stored on the device. The gallery must show a scrollable grid of all photos. When you click on a photo it must open in a separate activity which shows only that photo. Your app needs to implement all the behaviour demonstrated in the video on Stream including similar animations.

Your app must use a **LazyVerticalGrid** (see lecture 12) of thumbnails with a **LaunchedEffect** (see lectures 14 and 15) to load photos using a coroutine with the IO dispatcher. When the app starts (and resumes) use the **MediaStore** content provider to get a list of all photos ordered by date added with the most recent first (see lectures 15 and 16), for each photo you will need the id, orientation, width and height. You **must** use a targetSdk of 36 and a minSdk of 26 and you will need to get the **READ_MEDIA_IMAGES** or **READ_EXTERNAL_STORAGE** permission (depending on api level).

To get an **InputStream** for a photo use:

```
contentResolver.openInputStream(Uri.withAppendedPath(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,id))
```

Where **id** is a string obtained from the **MediaStore _ID** column

You must use **BitmapFactory.decodeStream** with **inSampleSize** options to load a low resolution thumbnail as described here:

https://developer.android.com/topic/performance/graphics/load-bitmap

You may need to rotate the thumbnail depending on the orientation. To speed up loading, you should use a cache of thumbnails. You are **not** allowed to use an image library such as Coil, Glide, Fresco, Picasso or Photo Picker (submissions using any of these will get 0 marks) but you can use a memory cache as described here:

https://developer.android.com/topic/performance/graphics/cache-bitmap

The single photo activity image must have a higher resolution than the thumbnail. You must add pinch to zoom to the photo viewer and gallery (see lecture 13). Make sure your app behaves correctly when the device is rotated and when photos are added or deleted. Make sure your app works with large images, i.e. up to at least 24MP. Some suitable test photos are available on Stream, transfer them to /sdcard/Pictures on the emulator using the Android Studio device explorer. You will need to do a cold boot (from the 3 dot menu for the device in device manager) for them to appear in MediaStore.

Submit your assignment on Stream. You must submit a zip file containing a clean source tree. To do this, use “File..Export to Zip File...” in Android Studio. You do not need to submit an apk. Do not use any libraries other than those included in the latest Android SDK. You **must** use **Kotlin** and **Compose** for this assignment, submissions using Java and/or Views will not be marked. You will lose marks if you submit a zip which was not created by Android Studio or any other format (such as rar or 7z). Marks will be deducted for zip files which do not build correctly. Do not change the top level gradle build file. We will build your app using the gradle command line so do not change anything in the gradle folder or the gradlew scripts.

This is an individual assignment, you can not work in groups. Marks will be awarded for well written apps. Use comments in your code to document it. Marks will be subtracted for plagiarism, late submission and bad documentation.

This assignment is worth 20% of the total marks for the course.

M Johnson 2025