This is the first oblig, **deadline Monday, 9th February**. Please submit your solution as a link to a public Git repository (e.g. on Codeberg, Gitlab, BitBucket, GitHub, ...) in groups of 2 or 3 students. We will review the app and your code with your group in one of the lab sessions after the deadline, and you will have the chance to revise your app once if anything doesn't work. Please do ask for help on time if you get stuck somewhere! *This assignment will be marked as passed when you've passed the review in the lab between the 9th and 16th.*

-Volker

The Quiz app

This little app should implement an easy game where you have to match names and photos (or a random selection of cute animal pictures). There are two core activities, which the user should be able to choose from when the application starts:

- the "gallery": it shows all names & pictures, if necessary, letting the user scroll through the list. There should be buttons for adding a new entry, removing an existing entry (e.g. by clicking an image), and sorting all existing entries alphabetical order or reverse order (from A to Z or from Z to A).
- the "quiz": When users click on this activity, the app will randomly select a photo from the gallery, and shows it on the screen. The app should present the right name for the photo and two wrong names in random order, and the user has to select the one they think is correct. After submission, there should be an indication by the app if the name was correct or not. If not, the app should show the correct name. After that, the whole process repeats until the user decides to leave this activity. The app should keep track of the score (the number of correct answers vs all attempts) and show it on the screen during the quiz.
- The "gallery" has the "add entry"-functionality: Here the user can add a new entry (i.e., a pair of a photo and the associated name). Please allow the user to choose an existing photo from his/her phone or enable the user to take a photo using his/her camera (using the camera is an optional feature, but you must at least let the user choose from the media gallery). The name/photo pair should then of course be available to the "gallery" and the "quiz".

Other remarks:

- you must design the "main menu" using an XML-based layout. Use Compose for all the other UIs.

- don't immediately try to use one of the fancy databases such as SQLite or Rooms! Use a simple datastructure from the Collections interface to maintain photos & names! Use the Application-class (see below) to share this datastructure throughout the app.
- add *three* photos (at least) and names to the app through the resource folder, and use it to initialize your database when the app starts! That is, load the image data and put it into your datastructure. (Make sure that the images are not too large, because it will also be in Git -- you can also of course use a cat-pic instead of your real face.)
- do not worry about persistently storing new entries (or the score) on the phone. We will add this functionality in the next oblig, for now it is okay if your app "forgets" everything except for the builtin-photos above when we restart the app.
- Make sure navigating back from an activity works correctly (common mistakes: internal data structure not updating correctly when adding/removing, gallery not updating after adding/deleting, memory leak when dealing with image files).
- Document your code!
- Please use git "properly", that is, only store the Android Studio project, not generated files like JARs and class-files.

In future obligs, we will work on storing the data (with new entries) on the phone, writing tests, and integration with other services on the phone.

Some Useful Links

- Use e.g the ACTION_GET_CONTENT/ACTION_OPEN_DOCUMENT-intent to let the user choose an existing image -- https://developer.android.com/guide/topics/providers/document-provider
- Lenker til eit ekstern område.
- , https://developer.android.com/training/data-storage/shared/documents-files#bitmap
- Lenker til eit ekstern område.
- Sharing state throughout an application (one of many possible ways) -- https://developer.android.com/reference/kotlin/android/app/Application
- Lenker til eit ekstern område.
-
- An alternative in Kotlin to RecyclerView: Lazy list