PBD 2023

Android Mini App 1

Summary

Your first assignment for this course consists in writing a simple Android app using Kotlin that allows users to create, delete and share memos. Each memo includes title and description (text added by the user) and optionally a photo (taken by the user using the smartphone's camera), together with a timestamp (marking the moment it was created). The user has the option to share memos via email.

NOTE: Please read the instructions carefully before you begin programming!

Deadline: 23:59, March 19th, 2023.

Features and Architecture

Your app must consist of one activity and three fragments:

1. MainActivity (of type AppCompatActivity), with layout file activity_main.xml;
2. ListFragment (of type Fragment), with layout file fragment_list.xml: it will show a RecyclerView populated with a list of the memos. For each memo, the photo, title and timestamp is displayed. Clicking on a memo will switch to the DetailsFragment. The layout will also include a floating action button with a plus ("+") sign; clicking on this button will allow adding a new memo (switching to the NewFragment);
3. DetailsFragment (of type Fragment), with layout file fragment_details.xml: it will show a detailed view of a memo: the photo, title, description and timestamp. It will also include two buttons: one for deleting the memo and one for sharing it via email (using an intent);
4. NewFragment (of type Fragment), with layout file fragment_new.xml: it will show an ImageView for the photo, two EditText elements for entering the title and the description, and two buttons: one for capturing a photo using the smartphone's camera and another for saving the memo;
5. Clicking on the take picture button within NewFragment must call the device camera for the user to take a picture. Once taken, the picture will be saved in the ImageView element of NewFragment. By default, ImageView will show a placeholder image;

6. Clicking on the Send email button in the DetailsFragment will lead to sending a new email message using an intent which will add the title of the memo as the subject of the message, the description + timestamp as the body text of the message, and the photo as an attachment to the message;
7. The layout of the RecyclerView item will be described in a layout file recycler_item_memo_model.xml;
8. MemoModel data class will be used for storing the data associated with a memo item.

Requirements

1. The app must target API version 31;
2. Use Android X (do not use the support libraries);
3. You must use Fragments (from Android X);
4. Do not use external libraries or frameworks, you must complete this assignment using only what the Android SDK and Android X offer you.

Implementation details

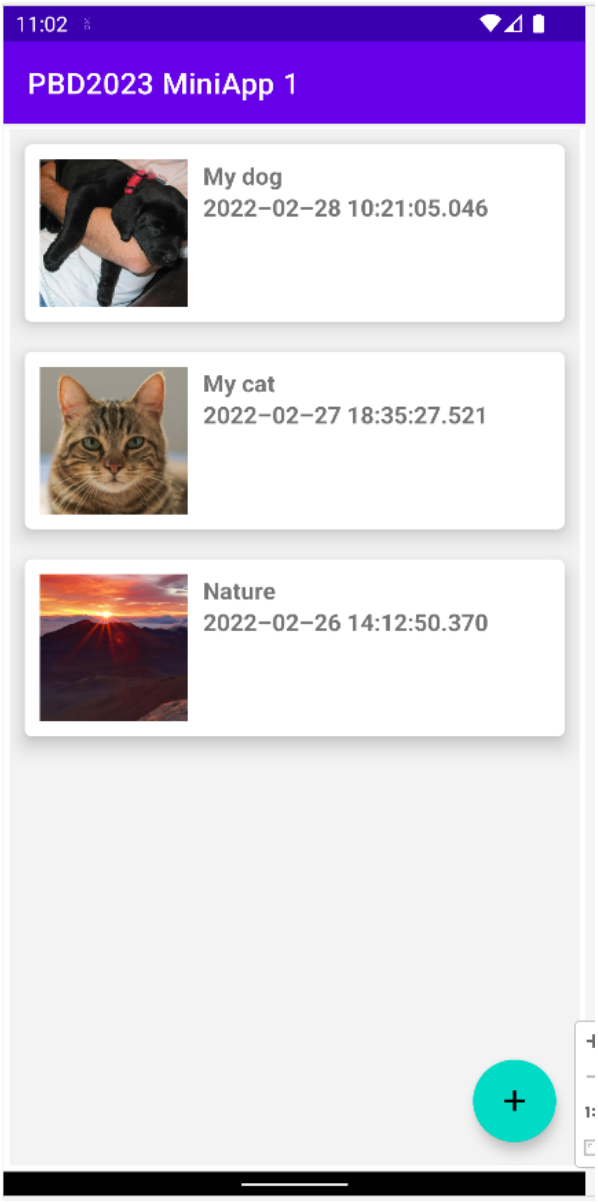
- The new memo's picture must be updated as soon as the user has taken a picture with the camera;
- You must use SharedPreferences to store the memos information. Suggestion: store each newly created memo as a JSON object in SharedPreferences. Load the JSON objects from SharedPreferences and use them to populate the RecyclerView. **Note:** SharedPreferences is not the optimal storage for large amounts of information, including images. We will learn about more efficient storage options in future course lectures. However, for the purpose of this app we will use SharedPreferences. In order to store the photo efficiently, it must be compressed before being saved (Hint: use the compress method from android.graphics.Bitmap);
- Your app must support screen orientation. For example, when adding a new memo (showing the NewFragment layout) if the screen orientation changes, the already entered information must be preserved;
- Implement error checking (for example, if you try to save a new memo with no title and/or description). Errors should be reported in a user-friendly manner;
- After adding a new memo, the app should return to showing the list of memos;
- The app should not crash in normal usage (nevertheless, you don't need to implement thorough and exotic testing scenarios);

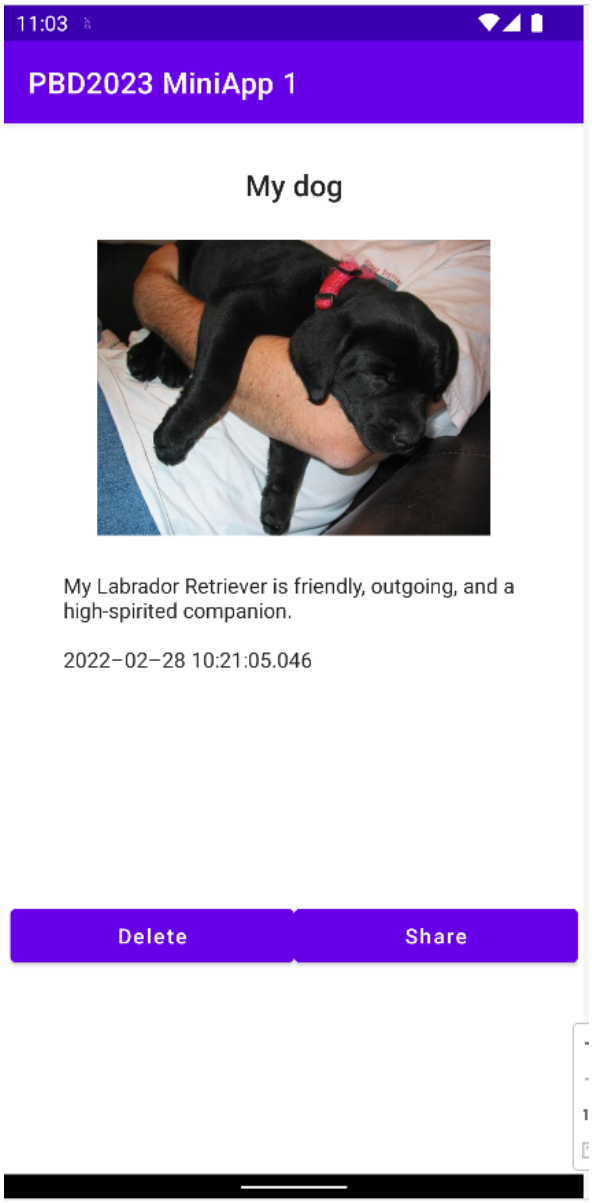
- Follow best-practice coding guidelines (for example, avoid using static methods and variables to pass information between fragments);
- Add meaningful comments to your code;
- Feel free to experiment with layout modifications (colors, structure, fonts): you are free to make the app look as good as you wish, as long as all the required functionality is there;

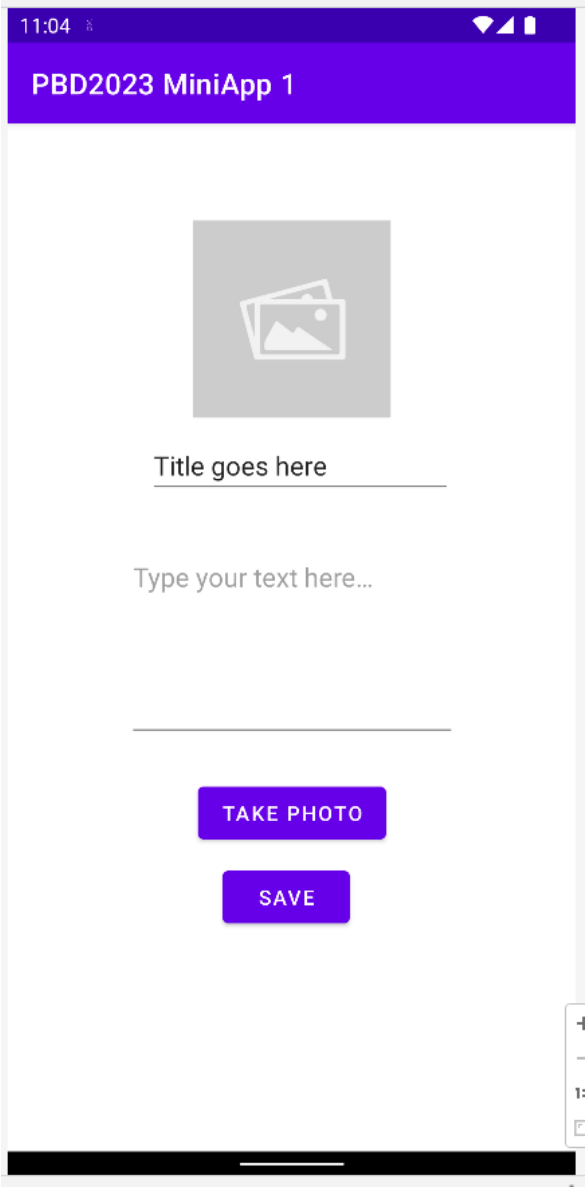
Important grading notes

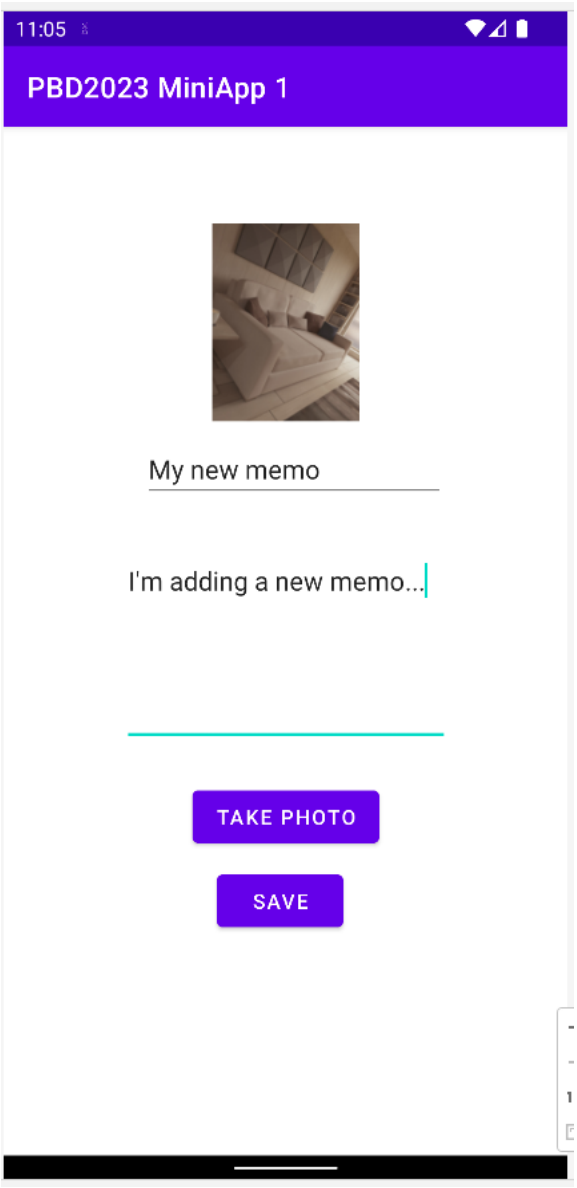
- You must submit your code in a repository titled PBD2023-MA-1 in your Bitbucket account. The repository must be private and the user “pbdfrita” (pbdfrita@gmail.com) must be added as a read-only member;
- Your project title must be: MiniApp1;
- Your classes must be named: MainActivity, ListFragment, DetailsFragment, NewFragment, MemoModel;
- Your layouts must be named: activity_main.xml, fragment_list.xml, fragment_details.xml, fragment_new.xml, recycler_item_memo_model.xml
- Your project package must be named: si.uni_lj.fri.pbd.miniapp1
- Your code must be fully anonymous - your email, name, or bitbucket user name must not be shown anywhere in the code/comments;
- Your comments, variable names, etc., must be in English

Example screenshots






11:04
PBD2023 MiniApp 1
Title goes here
Type your text here...
TAKE PHOTO
SAVE


11:05
PBD2023 MiniApp 1
My new memo
I'm adding a new memo...
TAKE PHOTO
SAVE