AnimoSpace Support

Android Challenge #2

7/17/2025

100 Points Possible

Attempt 1

In Progress
NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed
7/17/2025 to 7/17/2025

Details

Overview

This assessment is focused to test you on your knowledge of topics mainly from Module 4.a: Storage (Local) (https://dlsu.instructure.com/courses/214805/modules/973790?wrap=1) .

Please note that while this assessment is available from 1610H to 1750H, this assessment is designed to be answered in an hour and thirty minutes or less. There is no time limit except for the due time of the assessment specified in Canvas. Please note that the extra time is meant to account for possible technical issues you might encounter. If you benefit from working on the problem with extra time, then good for you, but no submissions will be accepted after the 1750H deadline. If you encounter issues, please try to get in contact with your instructor as soon as you're able to.

If you have any questions, please check the following Google Docs file: https://docs.google.com/document/d/1HnIEbdg9xyVS_mGwTg6jIfeBtQT0yr--3RqC59aOIwE/edit?usp=sharing (https://docs.google.com/document/d/1HnIEbdg9xyVS_mGwTg6jIfeBtQT0yr--3RqC59aOIwE/edit?usp=sharing)

Please note that the file is read-only. Ask your question directly to your instructor and your instructor will update the Google Docs file. You should not have access to edit the file.

Problem

You're tasked to complete a dummy media playlist application that saves data to a local database. All media items consist of (1) an ID, (2) a title, (3) a media type, and (4) a media duration. The ID is used to reference the location of the entry in the database. By default, it is set to -1. It is also represented as an integer. The title, media type, and duration are all Strings. There is no error checking for title and media duration, but media type should only be 'Audio' or 'Video' (case insensitive). This is already provided in the application.

The user can create, modify, and delete media items through different means:

- Creation is done by clicking on the + button, which opens a dialogue box allowing for user input.
- Modification is done by clicking on an item. This opens the same dialogue box with the information already provided.
- Deletion is done by swiping an item in the RecyclerView to the left or right.

Additionally, the MainActivity should be responsible for querying the data so that the entries can be displayed in the RecyclerView.

For a visualization of the app, please see the following demo video:

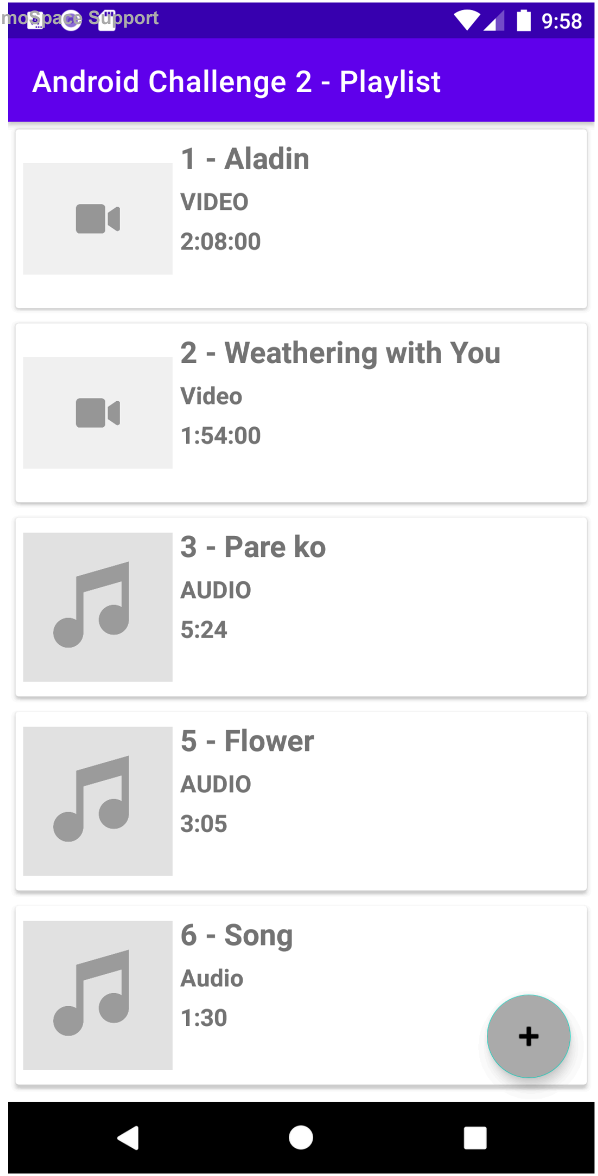
https://youtu.be/kFuJWuaiVww (https://youtu.be/kFuJWuaiVww)

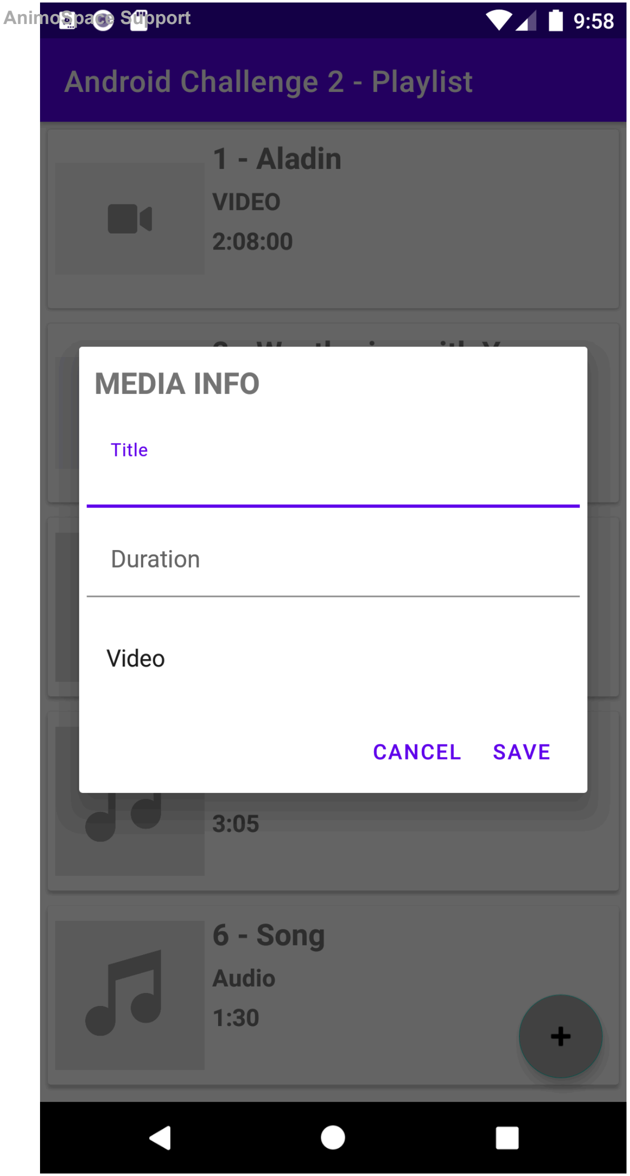
(https://youtu.be/kFuJWuaiVww)

Please make sure to read the details below so you can be certain of what is expected.

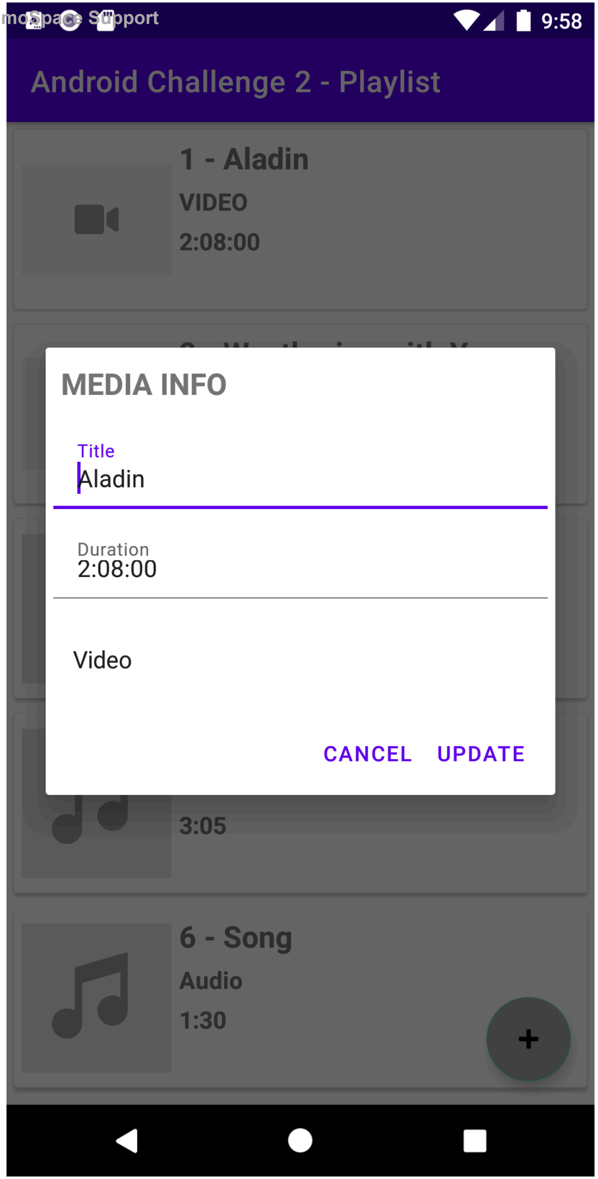
Activity / Screens	Details

AnimoSpace Support

9:58 Android Challenge 2 - Playlist 1 - Aladin VIDEO 2:08:00 2 - Weathering with You Video 1:54:00 3 - Pare ko AUDIO 5:24 5 - Flower AUDIO 3:05 6 - Song Audio 1:30 +	• This is the **main activity** that the user sees upon launch of the app. • The are only two main views: ◦ The **floating action button** for adding a media item ◦ The **recycler view** for showing the media items • On the **first start of the application**, there should be **five media items** that are **immediately loaded** into the DB. ◦ This implies that on the creation of the DB, these five items should be inserted into the database. ◦ You can modify the actual information, but if you don't want to think about what to use, please just use the specified information here. • On **load of the application** after the first launch, the **entries** in the database would be **loaded** accordingly. ◦ Please note that the number found in a view holder references the ID found in the database. It is not the order of the items, although it may look like it. In the example, notice how #4 is not present because it was previously deleted. • **Adding** an item is performed by clicking on the + floating action button. ◦ This opens a dialogue box, shown in the next row of this table. • **Modifying** an item is performed by clicking on a view holder. ◦ This opens the same dialogue box as with adding but the initial information is found in the box. This is shown in the last row of the table. • **Deleting** an item is performed by swiping a view holder to the left or right. ◦ This is not shown in the screenshots, but it is shown in the video above.



- This is what the dialogue box looks like when **adding** a media item.
- Clicking the save button triggers insertion of an entry into the DB.
- Clicking the cancel button does nothing.
- After clicking either update or cancel buttons, the dialogue box is removed and item should be found at the **bottom** of the RecyclerView.



Android Challenge 2 - Playlist

1 - Aladin
VIDEO
2:08:00

MEDIA INFO

Title
Aladin

Duration
2:08:00

Video

- This is what the dialogue box looks like when **modifying** a media item.
- Clicking the update button triggers a modification command of an entry in the DB.
- Clicking the cancel button does nothing.
- After clicking either update or cancel buttons, the dialogue box is removed and information in the RecyclerView should be **updated**.

Files Provided

You are provided an Android project that contains the following files:

- **Layouts (no layouts need to be modified; all are provided and work accordingly)**
  - activity_main (for the MainActivity)
  - dialogue_insert_media (for the dialogue box)
  - item_media (for the ViewHolder)
- **Classes**
  - adapter
    - PlaylistAdapter (a custom RecyclerView adapter class)
  - database
    - **DatabaseHandler** (a custom SQLiteHeleper class)
      - **HAS MISSING INFORMATION; MUST BE COMPLETED**
    - **PlaylistDatabase** (a wrapper class for the DatabaseHandler which handles most of the DB operations)
      - **HAS MISSING INFORMATION; MUST BE COMPLETED**
  - model
    - PlayListMediaItem (a model class for an individual media item)
  - **MainActivity** (the main activity class)
    - **HAS MISSING INFORMATION; MUST BE COMPLETED**
  - SwipeCallback (a class responsible for handling the swiping mechanism used by the RecyclerView)

Please note that both projects implement the ViewBinding feature.

Link to Java zip template: [AndroidChallenge2-Playlist (Java template).zip](https://dlsu.instructure.com/courses/214805/files/26095714?wrap=1) (https://dlsu.instructure.com/courses/214805/files/26095714?wrap=1) ↓ (https://dlsu.instructure.com/courses/214805/files/26095714/download?download_frd=1)

Link to Kotlin zip template: [AndroidChallenge2-Playlist (Kotlin template).zip](https://dlsu.instructure.com/courses/214805/files/26095717?wrap=1) (https://dlsu.instructure.com/courses/214805/files/26095717?wrap=1) ↓ (https://dlsu.instructure.com/courses/214805/files/26095717/download?download_frd=1)

Please note that the Java template was manually converted from the Kotlin template. Please inform your instructor if you see any issues with the translation.

Instructions

1. Ideally, use the Android project template provided as a template. Select the Java or Kotlin template based on your preference.
2. Complete the project to fulfill the specifications of the problem, as described above.
3. When done, kindly place a comment at the top of the DatabaseHandler and PlaylistDatabase classes that contains your name and section.
4. Export your project as a zip file (please see the instructions below for more information).
5. Upload the zipped project as your submission to this exercise.

How to Export Project as Zip (in Android Studio 4.x)?

In order to export an Android Studio Project as a zip file, from the IDE, go to File -> Export -> Export to Zip file. Select the appropriate directory to save the zip file. Please note that this is different from compressing the entire project folder. Performing the indicated steps results in a slightly more compressed version of the project -- lighter file, easier to upload.

Project / Settings / Considerations

- To avoid issues with the restructuring of your Android Project, you're discouraged from renaming the packages. Leave them alone so you can focus on the content.
- **However, you MUST place your name and section in a comment on each Java or Kotlin file of the project.**
  - Deductions will be applied if there are missing comments.
- Kindly set the minimum SDK support to either API 23
- For your own sake, please observe proper labeling of view ids.
  - There are different conventions, such as those found here: https://stackoverflow.com/questions/12870537/android-naming-convention (https://stackoverflow.com/questions/12870537/android-naming-convention)
  - The convention helps in readability and documentation.
- Your instructor reserves the right to apply deductions for not following these instructions.

Rubric

Category	Points
DatabaseHandler: Complete the inheritance of the class	6
DatabaseHandler: Complete the SQL statement to create the media item table in onCreate()	6
DatabaseHandler: Add five dummy data to the DB on the creation of the DB	7
MainActivity: Complete the logic for adding an item to the DB	22.5
PlaylistDatabase: Logic for updating a media item	18
PlaylistDatabase: Logic for deleting a media item	18

PlaylistDatabase: Logic for querying all media items in the DB + returning an ArrayList	22.5

View Rubric

Playlist App Rubric

Criteria	Ratings						Pts
DatabaseHandler: Complete the inheritance of the class	**6 pts** **Fully Functioning**		**3 pts** **w/ error**		**0 pts** **No Implementation**		/ 6 pts
DatabaseHandler: Complete the SQL statement to create the media item table in onCreate()	**6 pts** **Fully Functioning**		**3 pts** **w/ error**		**0 pts** **No Implementation**		/ 6 pts
DatabaseHandler: Add five dummy data to the DB on the creation of the DB	**7 pts** **Fully Functioning**		**3.5 pts** **w/ error**		**0 pts** **No Implementation**		/ 7 pts
MainActivity: Complete the logic for adding an item to the DB	**22.5 pts** **Fully Functioning**	**19 pts** **w/ minor error(s)** That do not crash the app	**15 pts** **w/ minor error(s)** That does crash the app. These errors should be easily fixable.	**8 pts** **w/ major error(s)** That does crash the app. These errors are not easily fixable. The logic, however, is moving in the right direction.	**4 pts** **Attempted** There are signs that the functionality was at least attempted, although the logic is severely lacking.	**0 pts** **No Implementation**	/ 22.5 pts
PlaylistDatabase: Logic for updating a media item	**18 pts** **Fully Functioning**	**15 pts** **w/ minor error(s)** That do not crash the app	**12 pts** **w/ minor error(s)** That does crash the app. These errors should be easily fixable.	**6 pts** **w/ major error(s)** That does crash the app. These errors are not easily fixable. The logic, however, is moving in the right direction.	**3 pts** **Attempted** There are signs that the functionality was at least attempted, although the logic is severely lacking.	**0 pts** **No Implementation**	/ 18 pts
PlaylistDatabase: Logic for deleting a media item	**18 pts** **Fully Functioning**	**15 pts** **w/ minor error(s)** That do not crash the app	**12 pts** **w/ minor error(s)** That does crash the app. These errors should be easily fixable.	**6 pts** **w/ major error(s)** That does crash the app. These errors are not easily fixable. The logic, however, is moving in the right direction.	**3 pts** **Attempted** There are signs that the functionality was at least attempted, although the logic is severely lacking.	**0 pts** **No Implementation**	/ 18 pts

AnimoSpace Support

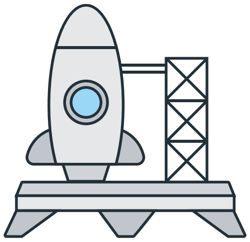
Playlist App Rubric

Criteria	Ratings						Pts
	22.5 pts **Fully Functioning**	**19 pts** **w/ minor error(s)** That do not crash the app	**15 pts** **w/ minor error(s)** That does crash the app. These errors should be easily fixable.	**8 pts** **w/ major error(s)** That does crash the app. These errors are not easily fixable. The logic, however, is moving in the right direction.	**4 pts** **Attempted** There are signs that the functionality was at least attempted, although the logic is severely lacking.	**0 pts** **No Implementation**	
PlaylistDatabase: Logic for querying all media items in the DB + returning an ArrayList							/ 22.5 pts

Total Points: 0

Choose a submission type

Upload ARC More



Choose a file to upload

File permitted: ZIP

or

Canvas Files

< (https://dlsu.instructure.com/courses/214805/modules/items/5850401)

Assignment > (https://dlsu.instructure.com/courses/214805/modules/items/5850405