Exercise #2 - Data-driven Views

100 Points Possible

Support

6/28/2025

Attempt 1

6/28/2025

NEXT UP: Review Feedback

Attempt 1 Score: N/A

Add Comment

Unlimited Attempts Allowed

6/2/2025 to 6/28/2025

Details

Overview

This exercise is focused to test you on your knowledge of topics from **Module 2.b: Basic Android Development (Data-driven Views [RecyclerView]) (https://dlsu.instructure.com/courses/214805/modules/973787)** .

Problem

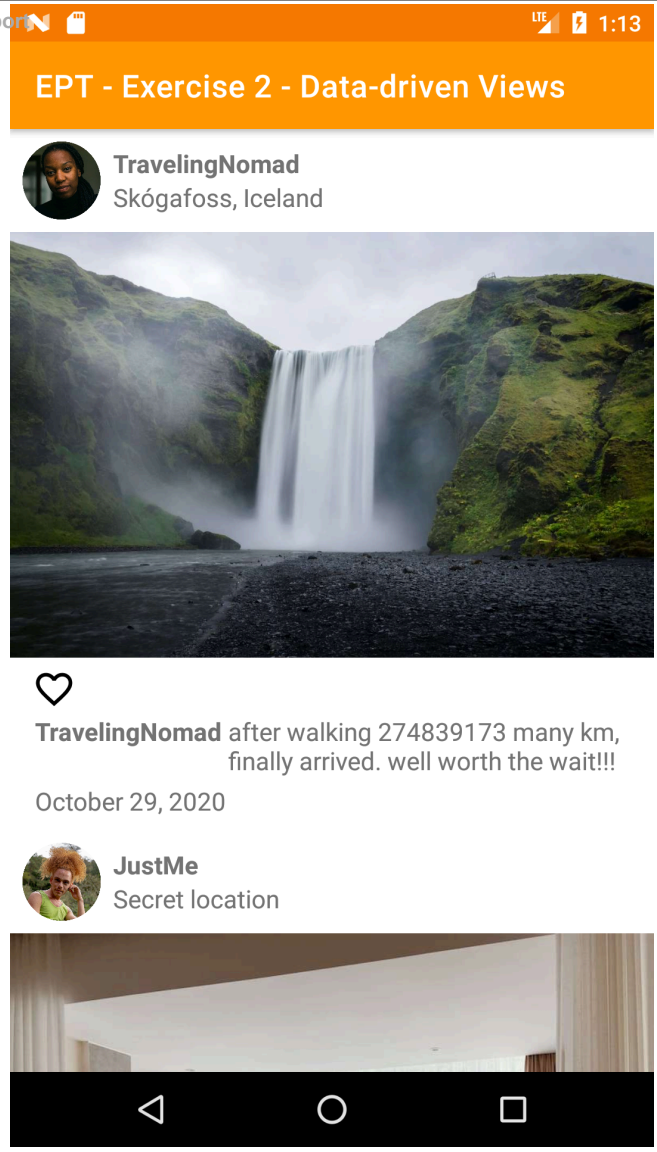
You are asked to recreate the following application:

https://youtu.be/DORDLzwCpw4 (https://youtu.be/DORDLzwCpw4)

(https://youtu.be/DORDLzwCpw4)

From the video, you should have noticed that there is only one activity/screen. Kindly see the following for details regarding implementation:

Activity / Screens	Details



- The single activity/screen is a list of posts styled somewhat like how Instagram is presented.
- In a post, you have the following information
  - User's username
  - User's image
  - Location
  - Image of the post
  - A like button
  - Caption of the post
  - Date posted
- Files are provided and you're expected to use these files. A zip file of all the files needed and a quick description is provided below this table.
- The list must be implemented using a RecyclerView.
- In case it isn't obvious from the video, here are some special scenarios with how the ViewHolder adjusts based on the data:
  - Caption...
    - All posts have a caption, but the caption can be null.
    - When the caption is present, it is displayed with the user's name. The username should be on the right and the caption should be on the left. Both should be under the like button.
    - When the caption is null, both the username and caption are not seen and do not take up space. This does not affect the like button or the date posted.
  - Location...
    - All posts have a location, but the value can be null.
    - When there is a location, the view is displayed under the username on top of the image.
    - When it is null, the view responsible for displaying the location should not be seen and does not take up space. This then centers the username with the user's image.
  - Some posts are automatically liked on initialization while others are not.
  - Whatever changes are many within the life of the app are not saved on close. There is no need to attempt to save the data.
  - The Like button can be clicked by the user to like or unlike a post.
- Lastly, kindly observe proper spacing between Views, as well as text size. Specifics aren't provided, so try to style your views accordingly.

- The image's width must span the entire width of the screen. Yes... it looks ugly for long photos.

Files Provided

You are provided a zip file containing the following:

- DataHelp contains a method that generates data.
- Post represents a post for our problem.
- Provided code is available in both Java and Kotlin.
- Images Folder contains all the image resources needed for the problem.

Link to zip file: **Exercise2-ProvidedFiles.zip (https://dlsu.instructure.com/courses/214805/files/26095788?wrap=1)** **(https://dlsu.instructure.com/courses/214805/files/26095788/download?download_frd=1)**

For reference, the images are taken from the following sources:

- fruits - **https://unsplash.com/photos/Kw0SidXP0Dw (https://unsplash.com/photos/Kw0SidXP0Dw)**
- furniture - **https://unsplash.com/photos/K8vwbVAwKko (https://unsplash.com/photos/K8vwbVAwKko)**
- race_car - **https://unsplash.com/photos/CpmHt-G1PuM (https://unsplash.com/photos/CpmHt-G1PuM)**
- waterfall - **https://unsplash.com/photos/54Pfu4EK4pc (https://unsplash.com/photos/54Pfu4EK4pc)**
- buildings - **https://unsplash.com/photos/GNMKDxDdhu8 (https://unsplash.com/photos/GNMKDxDdhu8)**
- work_desk - **https://unsplash.com/photos/UQxWuJLVmto (https://unsplash.com/photos/UQxWuJLVmto)**
- food - **https://unsplash.com/photos/HfjCJLCuTlo (https://unsplash.com/photos/HfjCJLCuTlo)**
- person1 - **https://unsplash.com/photos/-30gNkg-Abk (https://unsplash.com/photos/-30gNkg-Abk)**
- person2 - **https://unsplash.com/photos/llO3ll_7Rzo (https://unsplash.com/photos/llO3ll_7Rzo)**
- person3 - **https://unsplash.com/photos/jc0UM9z6G2w (https://unsplash.com/photos/jc0UM9z6G2w)**
- person4 - **https://unsplash.com/photos/LY4Qd8V_YLw (https://unsplash.com/photos/LY4Qd8V_YLw)**

Instructions

1. Create an Android project that fulfills the specifications of the problem, as described above.
2. When done, export your project as a zip file (please see instructions below).
3. 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

- When creating a project, please name the package name properly.
  - For reference, please follow the format specified:
    - **com.mobdeve.<section>.<lastname>.<first name>.exercise2**
    - e.g. com.mobdeve.s15.tighe.edward.exercise1
- 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**

Support (https://stackoverflow.com/questions/12870537/android-naming-convention)

- Convention helps in readability and documentation.

- Your instructor reserves the right to apply deductions for not following these instructions.

Rubric

Category	Points
Proper placement of image resources	5
RecyclerView implementation in MainActivity	5
Adapter general setup (passing of data, getItemCount method)	5
ViewHolder creation (onCreateViewHolder of Adapter, Constructor of ViewHolder)	15
ViewHolder binding (onBindViewHolder of Adapter, Setters of ViewHolder)	15
RecyclerView item layout	40
Like button logic	5
Null location handling	5
Null caption handling	5

View Rubric

Instagram App (Exercise 2) Rubric

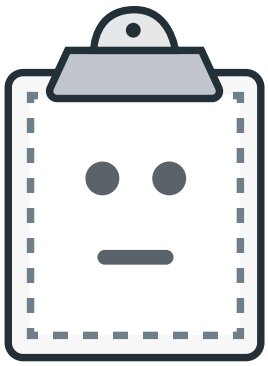
Criteria	Ratings			Pts
Proper placement of image resources	**5 pts** **Full Marks**	**2.5 pts** **w/ error(s)**	**0 pts** **No image resources used.**	/ 5 pts
RecyclerView implementation in MainActivity	**5 pts** **Full Marks**	**2.5 pts** **attempted, but with error OR with missing statements**	**0 pts** **No RecyclerView implementation in code attempted.**	/ 5 pts
Adapter general setup (passing of data, getItemCount method)	**5 pts** **Full Marks**	**2.5 pts** **Data instantiation and getItemCount were attempted but there's an error with either of them.**	**0 pts** **No attempt at either data instantiation or getItemCount**	/ 5 pts
ViewHolder creation (onCreateViewHolder of Adapter, Constructor of ViewHolder)	**15 to >14 pts** **Full Marks**	**14 to >0 pts** **with error or issues relating onCreateViewHolder or**	**0 pts** **No attempt at either data instantiation or getItemCount**	/ 15 pts

Support

Instagram App (Exercise 2) Rubric

Criteria	Ratings			Pts
		Constructor of ViewHolder		
ViewHolder binding (onBindViewHolder of Adapter, Setters of ViewHolder)	15 to >14 pts Full Marks	14 to >0 pts with error or issues relating with binding of data (onBindViewHolder or setters)	0 pts No attempt at either onBindViewHolder or ViewHolder's setters (or equivalent)	/ 15 pts
RecyclerView item layout	40 to >39 pts Full Marks	39 to >0 pts with error or issues	0 pts No layout file for the RecyclerView item found	/ 40 pts
Like button logic	5 pts Full Marks	2.5 pts w/ error(s)	0 pts No attempt at the like button logic	/ 5 pts
Null location handling	5 pts Full Marks	2.5 pts w/ error(s)	0 pts No attempt at null caption handling	/ 5 pts
Null caption handling	5 pts Full Marks	2.5 pts w/ error(s)	0 pts No attempt at null location handling	/ 5 pts

Total Points: 0



Preview Unavailable

S18ClementeDaniel-Exercise2.zip

Download

(https://dlsu.instructure.com/files/26788284/download?download_frd=1&verifier=1bjxbAVNJTncqjC0ToefJa0uiliru34Jsgc1MaUr)