UDACITY

Back to Android Basics Nanodegree by Google

Musical Structure App

REVIEW

CODE REVIEW 7

HISTORY

Meets Specifications

Hello Dear Student!

This was an excellent effort! Congratulations on successfully completing this project!
Keep up the good work and all the best for your future projects!

Best Regards
Ali

App Design

The app's structure is suitable for a music player app. A similarly structured app which focuses on audiobooks, podcasts, or other audio media is also acceptable.

The purpose of each activity is easy for a user to understand through the UI design and feature labeling.

The app contains 2 to 6 activities

Layout

The app contains multiple activities, each labelled, which together make a cohesive music app.

Features in the app are clearly defined either by labelling or images. For example, a button to play music could use a universally recognized triangular "Play" symbol or could have the text label "Play".

Each activity contains button(s) which link it to other activities a user should be able to reach from that activity. For instance, a 'Library' activity might contain a button to move to the 'Now Playing' activity.

The code adheres to all of the following best practices:

- Text sizes are defined in sp
- Lengths are defined in dp
- Padding and margin is used appropriately, such that the views are not crammed up against each other.

Functionality

The code runs without errors

Each button's behavior is determined by an OnClickListener in the Java code rather than by the `android:onClick` attribute in the XML Layout.

Each button properly opens the intended activity using an explicit Intent.

App uses a custom adapter to populate the layout with views based on instances of the custom class.

Information about instances of the custom class are stored in an appropriate data structure (e.g. ArrayList, Array).

When the information needs to be displayed, it is efficiently retrieved (e.g. Looping).

Data about each song (or equivalent audio media such as podcast episode) should be stored in a custom class that contains at least 2 states (e.g. Song Name, Artist Name)

If images are included (e.g. Album Art), they are stored as drawables. All drawables are stored at multiple densities. Images are not required.

Code Quality

The code is properly formatted:

- No unnecessary blank lines
- No unused variables or methods
- No commented out code

The code also has proper indentation when defining variables and methods.

Code is easily readable so that a fellow programmer can understand the purpose of the app.

All variables, methods, and resource IDs are descriptively named so that another developer reading the code can easily understand their function.

DOWNLOAD PROJECT

7 CODE REVIEW COMMENTS ›

RETURN TO PATH