Android Fundamentals Project Self-Evaluation

Required Components

To “meet specifications”, your app must fulfill all of the criteria listed in this section of the rubric.

Criteria	Does Not Meet Specifications	Meets Specifications
Standard Design		
App does not redefine the expected function of a system icon (such as the Back button).		X
App does not replace a system icon with a completely different icon if it triggers the standard UI behavior.		X
App does not redefine or misuse Android UI patterns, such that icons or behaviors could be misleading or confusing to users.		X
App includes a tablet layout which takes advantage of the additional space (if possible).		X
App includes at least two distinct views and uses intents properly to move between these views.		X
Navigation		
App supports standard system Back button navigation and does not make use of any custom, on-screen "Back button" prompts.		X
All dialogs are dismissible using the Back button.		X
Pressing the Home button at any point navigates to the Home screen of the device.		X

Permissions		
App requests only the absolute minimum permissions that it needs to support core functionality.		X
App does not request permissions to access sensitive data or services that can cost the user money, unless related to a core capability of the app.		X
Please elaborate on why you chose these permissions: ACCESS_FINE_LOCATION: This permission is used to access location data, which is (if enabled by the used) stored as a part of the Metadata of DriveFiles created by the app. The location is then used to provide location-based quick access to documents. INTERNET: This permission is needed to load the user's profile and cover photo.		
Performance and Stability		
App does not crash, force close, freeze, or otherwise function abnormally on any targeted device.		X
ContentProvider		
App implements a ContentProvider to access locally stored data.		X
If it regularly pulls or sends data to/from a web service or API, app updates data in its cache at regular intervals using a SyncAdapter. If it needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), app uses an IntentService to do so.		X
App uses a Loader to move its data to its views.		X
1) What's the content provider called, and how is it backed? The content provider is called PreferencesProvider and is used to store and load preferences. The preferences are stored in a SQLite database. I am aware that this is not practical, but I had already implemented most of the app when I realized that a content provider is needed. According to the Forum it is allowed to use the content provider for such a minor task.		

2) What backend does it talk to? What is the SyncAdapter called? What mechanism is used to actually talk over the network? Since all the data is stored on the user's Google Drive and the local SQLite database, the app does not require a backend. Hence, there is no need for a SyncAdapter. The only things directly fetched from the web are the user's profile and cover photo. **3) What loaders/adaptors are used?** A CursorLoader is used to access the preferences in MainActivity.		
User/App State		
App correctly preserves and restores user or app state.		X
When the app is resumed after the device wakes from sleep (locked) state, the app returns the user to the exact state in which it was last used.		X
When the app is relaunched from Home or All Apps, the app restores the app state as closely as possible to the previous state.		X
Please elaborate on how/where your app correctly preserves and restores user or app state: Since I refrained from using Fragments (IMHO they're more trouble than they're worth), the app restores most Activities without a hitch. The only Activity that needs active saving/restoring of the app state is the EditorActivity. Because it combines a document viewer and editor, it has to saved if the edit mode is enabled and whether the toolbar is expanded or collapsed.		

Optional Components

To receive “exceeds specifications”, your app must fully implement all of the criteria listed under at least two of the four categories below (e.g. Notifications, ShareActionProvider, Broadcast Events, and Custom Views).

Criteria	Does Not Exceed Specifications	Exceeds Specifications
Notifications		
Notifications do not contain advertising or content unrelated to the core function of the app.	X	
Notifications are persistent only if related to ongoing events (such as music playback or a phone call).	X	
Multiple notifications are stacked into a single notification object, where possible.	X	
App uses notifications only to indicate a context change relating to the user personally (such as an incoming message).	X	
App uses notifications only to expose information/controls relating to an ongoing event (such as music playback or a phone call).	X	
Please elaborate on how/where you implemented Notifications in your app: No notification were used.		
ShareActionProvider		
Uses ShareActionProvider to share content with an outside application.	X	

Makes use of Intent Extras to send rich content (i.e. a paragraph of content-specific text, a link and description, an image, etc).	X	
Please elaborate on how/where you implemented ShareActionProvider: I did not implement a ShareActionProvider.		
## Broadcast Events		
App intercepts broadcast events.		X
App responds to Broadcast events in a meaningful way.		X
Please elaborate on how/where you implemented Broadcast Events: Broadcasts are used for file operations (e.g. creating a new file/folder, deleting it, ...) as well as for modifications of documents (e.g. changing the style of a text block, altering the content, ...)		
## Custom Views		
App creates and uses a custom View.	X	
App uses a novel View that couldn't sufficiently be satisfied by the core Views in Android.	X	
Please elaborate on how/where you implemented Custom Views: No Custom Views were implemented, aside from RecyclerView items.		