Android Fundamentals Project Self-Evaluation

Instructions: Once you've completed your Final Project, please evaluate it against the components of the rubric below. For each criteria that you met, put an "X" in either the "Does Not Meet Specifications" or the "Meets Specifications" box. For some criteria, we ask you to provide an explanation of where and how it was implemented in your app. This is a chance for you to briefly explain to the grader your thought-process during development. Once you are done, include this with the source code and accompanying files you are submitting. Then, give yourself a pat on the back for making a great app!

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: I chose these permissions for network access and for using a sync adapter.		X
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 RepoProvider and it is backed by a SQLite Database.** **2) What backend does it talk to? What is the SyncAdapter called? What mechanism is used to actually talk over the network?** **It talks to the GitHub API for fetching trending repos, its contributors, and additional information. The SyncAdapter is called SyncAdapter. The app interacts with a REST API over HTTP. It uses Retrofit for making the request and RxJava for chaining together multiple requests to different API endpoints.** **3) What loaders/adaptors are used?** **The app uses a CursorLoader to load data from the RepoProvider to the views. A RecyclerViewAdapter is used to create and serve the views to the RecyclerView itself.**		X

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: The app does not change any expected system behavior in terms of preserving user and app state on configuration changes. It retains, for example, the current list item position on configuration changes. There are no user inputs for the app to preserve.		X

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:	X	
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: The ShareActionProvider was implemented using the support library version. It was added in the menu resource file for the DetailActivity. A share intent was created and set on the ShareActionProvider in the onCreateOptionsMenu method. The Sunshine app was used as a reference for implementing this feature.		X
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: I implemented Broadcast events in the SyncAdapter for informing the RepoWidgetProvider that the app data has been updated. The RepoWidgetProvider then receives these events in its onReceive method and notifies the widget of the data change.		X
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:	X	