

# 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
<b>Standard Design</b>		X
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
<b>Navigation</b>		
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
<b>Permissions</b>		
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
<b>Please elaborate on why you chose these permissions:</b>		
<b>Performance and Stability</b>		X
App does not crash, force close, freeze, or otherwise function abnormally on any targeted device.		X
<b>ContentProvider</b>		X
App retrieves and caches data from a server using a ContentProvider.		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.	X	
<b>Please elaborate on how/where you implemented a ContentProvider and SyncAdapter:</b>		
<p>I implements a ContentProvider to persist post from the Server into SQLite Database.</p> <p>We implements this two ContentProviders that encapsulates the data, and provides mechanisms for defining data security:</p> <p>a) ar.com.sccradiomobile.storage.provider.PostContentProvider</p> <p>b) ar.com.sccradiomobile.storage.provider.FeedContentProvider</p> <p>We don't send data to a web service, only consume. We are using <code>com.android.volley</code></p> <p>To handle the Request and Response to the Server.</p>		

<b>User/App State</b>		X
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
<b>Please elaborate on how/where your app correctly preserves and restores user or app state:</b>		
We restore the app state using the method "onResume()" and saving the information into databases in some context.		

## 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).

<b>Criteria</b>	<b>Does Not Exceed Specifications</b>	<b>Exceeds Specifications</b>
<b>Notifications</b>		
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	
<b>Please elaborate on how/where you implemented Notifications in your app:</b> No implementation.		
<b>ShareActionProvider</b>		
Uses ShareActionProvider to share content with an outside application.	x	
Makes use of Intent Extras to send rich content.	x	
<b>Please elaborate on how/where you implemented ShareActionProvider:</b> No implementation.		
<b>Broadcast Events</b>		
App intercepts broadcast events.		X
App responds to Broadcast events in a meaningful way.		X
<b>Please elaborate on how/where you implemented Broadcast Events:</b> We have a BroadcastReceiver that executes itself by interval times, and go to the server, looking for new info.		X

<p>In the file PostNoticiasFragment we implements MyAlarmReceiver.OnScheduleUpdateListener , and inside we have the "private BroadcastReceiver mAlarmReceiver".</p> <p>In the method "onResume()" we Register the receiver, and after receiver registering, we can schedule the update from the Server.</p> <p>We Schedule AlarmManager to invoke MyAlarmReceiver and cancel any existing current PendingIntent.</p>		
<b>Custom Views</b>		
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
<p><b>Please elaborate on how/where you implemented Custom Views:</b></p> <p>We use a custom font in a TextView. The way to achieve this is to create our custom class that extends TextView.</p> <p>First we place our fonts in the Asset folder. In our case we se the ROBOTO font, and put the font under the fonts folder in the assets.</p> <p>Then we create our custom class. I have place it under ar.com.sccradiomobile.utils. And I have call it RobotoTextView.</p> <p>We have override the setTypeface method to be able to change the font between bold, italic and normal in the layout without having to create another custom class.</p>		