## Android Fundamentals Project Self-Evaluation

**Instructions:** Once you’ve completed your Final Project, please respond to the questions below. 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!

# Questions about Required Components

## Permissions

**Please elaborate on why you chose the permissions in your app.**

|  |
| --- |
| *<!-- This permission is necessary in order for Brewski to perform network access. -->* <uses-permission android:name="android.permission.INTERNET" /> *<!-- Permissions required by the sync adapter -->* <uses-permission android:name="android.permission.READ\_SYNC\_SETTINGS" /> <uses-permission android:name="android.permission.WRITE\_SYNC\_SETTINGS" /> <uses-permission android:name="android.permission.AUTHENTICATE\_ACCOUNTS" /> |

## Content Provider

**What is the name of your Content Provider, and how is it backed? (For example, Sunshine’s Content Provider is named WeatherProvider backed by an SQLite database, with two tables: weather and location.)**

|  |
| --- |
| The name of my Content Provider is BrewskiContentProvider.java.  It is backed by 3 tables:   * beer * brewery * style |

**What backend does it talk to? (For example, Sunshine talks to the OpenWeatherMap API.)**

|  |
| --- |
| The backend talks to the BreweryDB API. Which can be found at:  http://www.brewerydb.com/developers/docs |

**If your app uses a SyncAdapter, what is it called? What mechanism is used to actually talk over the network? (For example, Sunshine uses HttpURLConnection to talk to the network, but your app may use a third-party library to do the talking.)**

|  |
| --- |
| My SyncAdapter is called BrewskiSyncAdapter and it also uses HttpURLConnection like the Sunshine app. |

**What loaders/adapters are used?**

|  |
| --- |
| There are CursorLoaders for each of my list and detail fragments. The fragments that use a CursorLoader are:   * BeerDetailFragment * BeerListFragment * BreweryDetailFragment * BreweryListFragment * StyleDetailFragment * StyleListFragment   There are 3 CursorAdapters in my project. They are:   * BeerListAdapter * BreweryListAdapter * StyleListAdapter |

## User/App State

**Please elaborate on how/where your app correctly preserves and restores user or app state. (See rubric for examples on this question)**

|  |
| --- |
| This is accomplished in the same manner that Sunshine achieved it, as I modeled my code directly off of it. With that being said, I’ve been handling this in my fragments via the methods onActivityCreated and onSaveInstanceState, in which I have overridden both.  I also use onPause and onResume in BeerListFragment, BreweryListFragment, and StyleListFragment for registering and unregistering my BroadcastReceivers. |

# Questions about Optional Components

Answer the questions that are applicable to your final project

## Notifications

**Please elaborate on how/where you implemented Notifications in your app:**

|  |
| --- |
| Did not implement. |

## ShareActionProvider

**Please elaborate on how/where you implemented ShareActionProvider:**

|  |
| --- |
| I implemented a SharActionProvider in each of the detail views so that the user would be able to share the individual item that they are looking at. The fragments where they are located are:   * BeerDetailFragment * BreweryDetailFragment * StyleDetailFragment |

## Broadcast Events

**Please elaborate on how/where you implemented Broadcast Events:**

|  |
| --- |
| I didn’t broadcast events in the sense that the project rubric outlined, but I do broadcast intents within my app. While a user is scrolling in the listviews, if the user gets within 25 items of the bottom of the list another call is fired off to the sync adapter and a Boolean is set to true to lock the interface from firing off subsequent calls. When the sync adapter is done loading and placing the new data into the database it broadcasts and event that the listviews are listening for so that they can flip that Boolean again and refresh the Loader. |

## Custom Views

**Please elaborate on how/where you implemented Custom Views:**

|  |
| --- |
| I implemented a Custom View for the Dashboard that is loaded when the app is first loaded up. The file is DashboardLayout and it extends ViewGroup. It then uses the onMeasure and onLayout functions to layout the buttons that are in the content portion of the dashboard.  Then, the actionbarLayout.xml, dashboardLayout.xml, and footerLayout.xml are included in the activity\_dashboard.xml. |