## 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.**

|  |
| --- |
| The permissions granted for the app were minimum:   1. **Access to Internet**   <uses-permission android:name="android.permission.INTERNET" />   1. **For 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 the Content provider is MovieProvider, backed by a SQLite database with a table named movie. |

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

|  |
| --- |
| It talks to the Rotten Tomatoes API, (more details at <http://developer.rottentomatoes.com/> ) |

**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.)**

|  |
| --- |
| The SyncAdapter used is called MovieSyncAdapter. It uses the HttpURLConnection mechanism to talk over the network. |

**What loaders/adapters are used?**

|  |
| --- |
| The adapter used is called MovieAdapter. |

## 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)**

|  |
| --- |
| The app meets the following behavior:   * When an activity is displayed, the same activity appears on rotation. (For both the MainActivity and MovieDetailActivity) * When the app is relaunched from Home or All Apps, the app restores the app state as closely as possible to the previous state. |

# 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:**

|  |
| --- |
| Not implemented. |

## ShareActionProvider

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

|  |
| --- |
| The ShareActionProvider was implemented by using android.support.v7.widget.ShareActionProvider  The option to share is implemented in the MovieDetailFragment, so users are able to share via messaging or other social media, the name of the movie that is now playing on theaters.  The sharing is done via an intent using (ACTION\_SEND), which is created and past as argument to the ShareActionProvider. |

## Broadcast Events

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

|  |
| --- |
| Not implemented |

## Custom Views

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

|  |
| --- |
| Not implemented. |