## 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.**

|  |
| --- |
| **android.permission.INTERNET**, used to access the internet and fetch data from Google Books API & Good Reads API. |

## 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.)**

|  |
| --- |
| I have created 2 Content Providers to access 2 distinct databases:  **ResultProvider** backed by ResultDBHelper, with table *result* that stores the similar books.  **BookProvider** backed by BookDBHelper, with table *books*that stores saved books.(favorites) |

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

|  |
| --- |
| **Google Books API** – Used to do the searching of the book  **Good Reads API** – Used to fetch similar books and more detailed information. |

**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.)**

|  |
| --- |
| I am using 2 **IntentService (**BookIntentService & MainIntentService**)** called on BookFragment & MainFragment. |

**What loaders/adapters are used?**

|  |
| --- |
| I am using a **CursorLoader** attached to my **CursorAdapter** (ResultAdapter) on ResultFragment.  I am also using another **CursorAdapter** on populating the saved books on SaveFragment |

## 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 application remains as it is if it is not destroyed, onPause and then onStop will store the app state and will be called once the activity is not visible on the screen but running in the background waiting for its chance to become visible again, once that happens, onResume then onStart will be called resuming what the last app state was. |

# 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:**

|  |
| --- |
|  |

## ShareActionProvider

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

|  |
| --- |
|  |

## Broadcast Events

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

|  |
| --- |
|  |

## Custom Views

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

|  |
| --- |
|  |