## 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 app requests for the vibration permission from the user, which was implemented in the apps introductory activity to give the user a more in depth UX feel. |

## 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 Flash Study app’s content provider is named FlashProvider and it is backed by an SQLite database, with two tables: classes and cards. |

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

|  |
| --- |
| FlashStudy does not talk to any backend API. |

**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 FlashStudy app does not use a SyncAdapter since it does not implement the retrieval of data via internet or network. |

**What loaders/adapters are used?**

|  |
| --- |
| CursorLoaders and a customListViewAdapter were implemented in the FlashStudy app to move the app’s database data to its views. |

## 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)**

|  |
| --- |
| Although the FlashStudy app’s orientation was designed to remain portrait on phones and landscape on tablets, the app does implement the following had it been rotation was allowed:   * When a list item is selected, it remains selected on rotation * When a fragment is displayed, the same fragment appears on rotation * User text input is preserved on rotation   For the FlashStudy app, the user is returned to the exact state in which it was last used when the app is resumed after the device wakes from sleep.  When the app is relaunched from the Home or All Apps screen, the app restores the app state as closely as possible to the previous state if not exactly 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:**

|  |
| --- |
| Notification was not implemented in the FlashStudy app at this stage of development since it was not necessary. |

## ShareActionProvider

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

|  |
| --- |
| ShareActionProvider was not implemented in the FlashStudy app at this stage of development since it was not necessary. |

## Broadcast Events

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

|  |
| --- |
| Broadcast Event was not implemented in the FlashStudy app at this stage of development since it was not necessary. |

## Custom Views

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

|  |
| --- |
| A CustomTextView class was implemented that extends from the built in TextView class. This was done to change the typrface of the TextViews. Also, a CustomListViewAdapter, extending the BaseAdapter class was implemented to populate the ListView as seen in the ‘ClassDetailsFragment’. Two other Custom views were used to display a GIF image and the app intro of the FlashStudy app. |