## 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.**

|  |
| --- |
| As required in the Rubric app requests only the absolute minimum permissions that it needs to support core functionality. 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.  As well as app pulls data from web it requires permission to check network state and permission to access the Internet. One more permission is for accessing to the vibrator, to let the phone vibrate when receiving notification. |

## 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.)**

|  |
| --- |
| My Content Provider named DebtaProvider. It backed by SQLite database, with four tables: User, DGroup, DCheck, DCheckItem. |

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

|  |
| --- |
| Debta talks to Parse.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.)**

|  |
| --- |
| My SyncAdapter named DebtaSyncAdapter. The app uses third-party library Parse SDK, to talk to the network. |

**What loaders/adapters are used?**

|  |
| --- |
| Debta uses several custom loaders (DGroupLoader, DCheckLoader, DCheckItemLoader, DCheckEditLoader, UserLoader,UserSearchLoader), which extend AsyncTaskLoader.  To populate listviews used custom adapters (DGroupListAdapter, DCheckListAdapter, DCheckItemsListAdapter, UserListAdapter, UserSearchListAdapter), which extends ArrayAdapter.  One more adapter (DGroupPagerAdapter) used to populate ViewPager.  App also contains SyncAdapter. |

## 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)**

|  |
| --- |
| Debta correctly preserves and restores user or app state When an activity is displayed, the same activity appears on rotation. User text input is preserved on rotation. 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. 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:**

|  |
| --- |
| Debta’s SyncAdapter configured for automatically syncing after a specified period of time. After sync finish app check is there any unseen DChecks. If such available user receives notification. |

## 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:**

|  |
| --- |
| Debta contains custom ListView. This was done to put that ListView into ScrollView. |