## 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.**

|  |
| --- |
| Only two permissions are requested in the app:  <**uses-permission android:name="android.permission.INTERNET"** /> <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />  The bulk of functionality of this application depends on interacting with the Parse cloud database, which requires an active connection to the internet. The app also requires access to the network state, to check that the app is connected to the internet before making a request, and displaying a toast error when there is no internet connection. |

## 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 custom Content Provider is DatesAvailListProvider, and is backed by an SQLite database set up in the class DatesAvailDatabase with one table: dates.db. |

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

|  |
| --- |
| The SQLite database and the Content Provider is used in the activity ProfileActivity. The app queries Parse for the user’s uploaded dates available and downloads it to the SQLite database. |

**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 app uses an AsyncTask implemented in FriendsDisplayFragment to run Facebook API method GraphRequest.newMyFriendsRequest, which allows the user to view his Facebook friends which have the app installed. |

**What loaders/adapters are used?**

|  |
| --- |
| The loader in ProfileActivity then queries this database via the CONTENT\_URI in the Content Provider, and connects it to a custom CursorAdapter, DatesAvailCursorAdapter. This CursorAdapter is then used to display the results of the Content Provider query to the UI via a ListView.  Loaders are also used in FriendsDisplayFragment, FriendsDetailFragment, PendingActivity to asynchronously download and push data to the ListViews. Each ListView in the above fragments/activities also implements its own custom ArrayAdapters and custom button listeners based on the options available for the user to interact with the data displayed. |

## 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)**

|  |
| --- |
| To ensure the same app’s state is preserved on rotation, the following line was added to the AndroidManifest in the relevant activities:  **android:configChanges="keyboardHidden|orientation|screenSize"**  Most of the app’s state is preserved immediately on the Parse database. Relaunching the app, or calling it after sleep calls the onResume() method which requeries the data and preserves all the changes that the user has made. |

# 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:**

|  |
| --- |
|  |