## 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.**

|  |
| --- |
| In order to use Parse, we needed the permissions “**ANDROID\_NETWORK\_STATE**,” as well as “**INTERNET**.” In order to use the Android Maps API, we needed the permissions “**MAP\_RECIEVE,**” “**ACCESS\_COARSE\_LOCATION**,” “**ACCESS\_FINE\_LOCATION**,” “**WRITE\_EXTERNAL\_STORAGE,”** and “**INTERNET.**”All of these permissions can be seen in our AndroidManifest file. |

## 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.)**

|  |
| --- |
| As our Content Provider, we used Parse. Parse allows users to set up a local database and pull/push objects into that database. Our objects that we pushed/pulled were each of the pins that were dropped. Parse stores all of this information for us, and each of the objects (pins) were set up on a table with the following categories: **Activity, Index, Start\_Time, End\_Time, Date, Description, Latitude,** and **Longitude.**  \*Index is a number between 0-5 which defines what color the pin should be. |

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

|  |
| --- |
| Our app talks to the Android Maps API which required Google Play Service permission. |

**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.)**

|  |
| --- |
| We used Parse in order to get the data to and from the network and allow multiple users to pull down the same event pins. |

**What loaders/adapters are used?**

|  |
| --- |
| We used Parse to load the data (see three questions above). |

## 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)**

|  |
| --- |
| When you close out the app, the app will open up to your current location again. The idea is that you want to know the events that are going on around, and near you, so it opens up to your current location. Any pins you add to the map will be there when you re-open it, as well as if a different user opens up the app on their phone- all pins will be there. |

# 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:**

|  |
| --- |
|  |