

# 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.**

I used "android.permission.READ\_CONTACTS" in order to read user contacts data stored in the phone; I used "android.permission.VIBRATE" in order to give a feedback to the user while using the "send contact info" button: the overall process (search contact and choose details) is terminated and he have only to choose the messaging app.

### 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 Content Provider is named "DataProvider"; it is backed by an SQLite database, with one table: "data".

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

It talks to the internal Contacts Provider which manages several tables storing contacts info.

**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 doesn't use a SyncAdapter.

**What loaders/adapters are used?**

The app uses `CursorLoader` in order query the "DataProvider". A custom `CursorAdapter` is used working with a `ListView`.

## 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)

Basically the app uses `Shared Preferences` in order to store the main `ListView` status in the `MainActivity`: the app preserve the last appearance when is launched, preserve the status when the orientation is changed. The `Detail Activity` preserve the status only on orientation change and until the underlying fragment is destroyed: this is made by means of `saveInstanceState Bundle`. Of course, a conditional if (`saveInstanceState == null`) is used in the `onCreate` callback of each activity to "follow" the activity lifecycle.

## 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:**

I didn't use Notifications.

### ShareActionProvider

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

I didn't use ShareActionProvider.

### Broadcast Events

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

I didn't use Broadcast Events.

### Custom Views

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

I didn't use Custom Views.