## 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.**

|  |
| --- |
| GiftWise uses the Contacts content provider. It requires READ\_CONTACTS access to allow users to import from Contacts. To create a RawContact in Contacts, we need WRITE\_CONTACTS and AUTHENTICATE\_ACCOUNTS permissions. |

## 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.)**

|  |
| --- |
| com.honu.giftwise.data.GiftContentProvider  GiftContentProvider is backed by SQLite with three tables:  1) gift  2) color  3) size |

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

|  |
| --- |
| GiftWise does not talk to a backend. All data is stored locally. It uses the Contacts content provider to allow users to import family and friends from their Contacts. |

**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.)**

|  |
| --- |
| GiftWise does not have a SyncAdapter today. |

**What loaders/adapters are used?**

|  |
| --- |
| Cursor loaders and adapters are used in multiple Activities. Gift ideas, color preferences, and size details all use CursorLoader and CursorAdapter. The InfoActivity includes a custom adapter with a ListView to display items from a List of Pair<String, String>. |

## 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)**

|  |
| --- |
| The EditGiftFragment and EditSizeFragment are examples where instance state is saved and restored on orientation change. Both activities are forms for user data entry, and the current values must be saved and restored. |

# 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:**

|  |
| --- |
| none |

## ShareActionProvider

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

|  |
| --- |
| ShareActionProvider is used by ViewGiftFragment and ContactActivity. The user can share a details of a single gift item from the ViewGiftFrament or a list of gift items from the ContactActivity. Both share content as text. |

## Broadcast Events

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

|  |
| --- |
| none |

## Custom Views

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

|  |
| --- |
| To provide some material design elements, I reused the Android example code for FloatingActionButton and SlidingTabLayout. My project also includes com.android.colorpicker as a module. In this module, I modified the ColorPickerDialog to support multi-select. |