## 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.**

|  |
| --- |
| My app is a Bluetooth manager app hence it requires the following permissions:  \* BLUETOOTH  \* BLUETOOTH\_ADMIN  \* BROADCAST\_STICKY  \* BIND\_ACCESSIBILITY\_SERVICE  My app involves switching on Bluetooth on startup (Since app is a Bluetooth Manager) hence requires "BLUETOOTH\_ADMIN" permission. My app also connects to paired Bluetooth devices hence it requires "BLUETOOTH" , " BROADCAST\_STICKY " and " BIND\_ACCESSIBILITY\_SERVICE" 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.)**

|  |
| --- |
| My app is a standalone app and doesn't use any service providers of any sort. |

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

|  |
| --- |
| My app is a standalone app and doesn't talk to any backend. |

**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 app doesn't use a SyncAdapter |

**What loaders/adapters are used?**

|  |
| --- |
| My app doesn't use loaders/adapters |

## 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)**

|  |
| --- |
| My app stores the list of paired A2DP devices in the internal storage. files saved to the internal storage are private to the application and other applications cannot access them (nor can the user). When the user uninstalls the application, these files are removed. Ive accomplished this by using the following method :  \* Calling [openFileOutput()](http://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)) with the name of the file (FILENAME) and the operating mode(MODE\_PRIVATE) which returns a [FileOutputStream](http://developer.android.com/reference/java/io/FileOutputStream.html).  \* I have stored the data as a serialised Java data object with [write()](http://developer.android.com/reference/java/io/OutputStream.html#write(byte[])).  \* Close the stream with [close()](http://developer.android.com/reference/java/io/FileOutputStream.html#close()).  My app later reads this stored data using the following functions :   1. \* Calling [openFileInput()](http://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String)) and pass it the name of the file to read. This returns a [FileInputStream](http://developer.android.com/reference/java/io/FileInputStream.html). 2. \* Read bytes from the file with [read()](http://developer.android.com/reference/java/io/FileInputStream.html#read(byte[], int, int)) and deserialise that data to a java object. 3. \* Then close the stream with [close()](http://developer.android.com/reference/java/io/FileInputStream.html#close()). |

# 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 haven't implemented notifications in my app |

## ShareActionProvider

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

|  |
| --- |
| I haven't implemented ShareActionProvider in my app |

## Broadcast Events

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

|  |  |
| --- | --- |
| I haven't implemented Broadcast Events in my app |  |

## Custom Views

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

|  |
| --- |
| In the "Manage Bluetooth Priority" activity I've used a custom "Draggable List View". This helps in setting the priority of the paired devices by moving each element by long pressing them and dragging them over other devices in the list with ease and placing them at the required priority position (courtesy DevBytes). |