## 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.**

|  |
| --- |
| 1) INTERNET : The app uses internet access to communicate with 2 API's (Google Places and Google Directions).  2) ACCESS\_NETWORK\_STATE : The app detects the availability of an internet connection. If not available, the app relies on more limited data in an internal SQLite-database. |

## 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.)**

|  |
| --- |
| Its name is TaxiProvider. It's backed by a SQLite-database, which gets initialized upon first use of the app by inserting the content of 2 csv files, shipped with the app in the “assets”-folder.  Two tables are created :  1) poi : this table holds basic info of a list of 'points of interest' in Lima (spanish name/english name/address). These are frequently visited places in the city. For any combination of 2 of these places, the app can return taxi fare information without an internet connection.  2) rate : this table holds information on taxi fare, distance and duration between any 2 of the available points of interest in Lima. |

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

|  |
| --- |
| When an internet connection is available, the app allows calculating taxi fares between random addresses in Lima (as opposed to just points of interest). For automated address completion, it interacts with the Google Places API. For distance (and subsequent rate) calculation, it interacts with the Google Directions API. |

**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.)**

|  |
| --- |
| SyncAdapter not applicable.  The Google Places API is accessed through HttpURLConnection.  The Google Directions API is accessed in an AsyncTask, through DefaultHttpClient. |

**What loaders/adapters are used?**

|  |
| --- |
| SimpleCursorAdapter with a Loader<Cursor> to fetch the points of interest in the SQLite-database and show them as dropdown-items in a spinner.  GooglePlacesAutocompleteAdapter (extends ArrayAdapter implements Filterable) for loading the selectable address options in the autocomplete fields as letters get typed by the user. |

## 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)**

|  |
| --- |
| Currency preference is saved in SharedPreferences and thus always properly saved (and checked by the app).  In the MainFragment 4 booleans are saved upon onSaveInstanceState, to keep track of which fields have been selected (a point of interest or an address) and thus are highlighted with a border.  In the ResponseFragment the contents of the TextViews which hold the response data (rate, distance, duration, addresses used) are saved upon onSaveInstanceState, in order not to lose their contents upon rotation or upon temporarily leaving the app. |

# 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:**

|  |
| --- |
| Not implemented. |

## ShareActionProvider

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

|  |
| --- |
| Not implemented. |

## Broadcast Events

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

|  |
| --- |
| Not implemented. |

## Custom Views

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

|  |
| --- |
| Not implemented. |