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.**

**INTERNET, ACCESS\_NETWORK\_STATE** - The app works with the internet

**USE\_CREDENTIALS, AUTHENTICATE\_ACCOUNTS, READ\_SYNC\_SETTINGS, WRITE\_SYNC\_SETTINGS** - For the sync adapter

**VIBRATE** - For notifications

**com.google.android.c2dm.permission.RECEIVE, com.ymgeva.doui.permission.C2D\_MESSAGE, com.ymgeva.doui.permission.C2D\_MESSAGE, android.permission.WAKE\_LOCK** - For push notifications receiver

**GET\_ACCOUNTS, READ\_PROFILE, READ\_CONTACTS** - Used for login with Facebook and for autocomplete email address. Both features are not implemented at the moment.

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.)**

**DoUIContentProvider**, backed up by SQLite database with 2 tables - task\_items and shopping\_items (there’s also a third table - general\_items which is not used atm).

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

**parse.com**

**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 sync adapter is called **DoUISyncAdapter**.

The actual network service is done by **parse.com** and is implemented in **DoUIParseSyncAdapter**.

**What loaders/adapters are used?**

I’m using a CursorLoader and CursorAdapter for **ShoppingListAdapter** and **TaskListAdapter**.

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)**

In **TaskListFragment** the activated position is saved. Also in **EditTaskActivity** the details are saved on rotation (specifically the reminder CheckBox).

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:**

**NotificationsService** handles a number of notification events - reminders for tasks, notification for urgent tasks and shopping items.

ShareActionProvider

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

Broadcast Events

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

**NotificationsService** listens to an AlarmManager broadcasts to send reminder notifications.

**DoUIPushBroadcastReceiver** receives push notifications from parse.com and handles them according to a push code - for example it initiates a sync to fetch new urgent data.

**SyncDoneReceiver** receives notification for when a sync is done and acts according to a push code - it might send notifications to the user, or send ask the backend to send a push to another user.

Custom Views

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

**CheckableImageView** is an image view that handles “onCheck” events. It changes the alpha of the presented image according to the state. It is used in all “check boxes” and also with the “important” exclamation mark icon in the shopping list.