## Android Fundamentals Project Self-Evaluation

**Instructions:** Once you’ve completed your Final Project, please evaluate it against the components of the rubric below. For each criteria that you met, put an “X” in either the “Does Not Meet Specifications” or the “Meets Specifications” box. For some criteria, we ask you to provide an explanation of where and how it was implemented in your app. 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!

### Required Components

To “meet specifications”, your app must fulfill all of the criteria listed in this section of the rubric.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Does Not Meet Specifications** | **Meets Specifications** |
| Standard Design |  |  |
| App does not redefine the expected function of a system icon (such as the Back button). |  | X |
| App does not replace a system icon with a completely different icon if it triggers the standard UI behavior. |  | X |
| App does not redefine or misuse Android UI patterns, such that icons or behaviors could be misleading or confusing to users. |  | X |
| **Navigation** |  |  |
| App supports standard system Back button navigation and does not make use of any custom, on-screen "Back button" prompts. |  | X |
| All dialogs are dismissible using the Back button. |  | X |
| Pressing the Home button at any point navigates to the Home screen of the device. |  | X |
| **Permissions** |  |  |
| App requests only the absolute minimum permissions that it needs to support core functionality. |  | X |
| App does not request permissions to access sensitive data or services that can cost the user money, unless related to a core capability of the app. |  | X |
| **Please elaborate on why you chose these permissions:**  INTERNET – to access the internet, GET\_ACCOUNTS – required by sync adapter, WAKE\_LOCK –use by Gcm broadcast receiver, c2dm.permission.RECEIVE – used to receive GCM messages, READ\_SYNC\_SETTINGS, WRITE\_SYNC\_SETTINGS\_AUTHENTICATE\_ACCOUUNTS – are all required for sync adapter implementation. READ\_SYNC\_STATS – is used to listen for the status of the sync adapter, in order to appropriately use the progress bar |  |  |
| **Performance and Stability** |  |  |
| App does not crash, force close, freeze, or otherwise function abnormally on any targeted device. |  | X |
| **ContentProvider** |  |  |
| App retrieves and caches data from a server using a ContentProvider. |  | X |
| If it regularly pulls or sends data to/from a web service or API, app updates data in its cache at regular intervals using a SyncAdapter. |  | X |
| **Please elaborate on how/where you implemented a ContentProvider and SyncAdapter:**  My content provider is implemented in the com.tasteofuganda.app.provider package. My sync adapter is implemented in the com.tasteofuganda.sync package.  I manually trigger the sync adapter only when the app is opened for the first time, and it fetches all the required data. The device also tries to register itself on my GCM server on the first launch.  Once the device has been registered on my GCM server, all future syncs are triggered via GCM. My server sends tickles instructing the device to download new data if there is any, in order to save on the user’s data usage. |  |  |
| **User/App State** |  |  |
| App correctly preserves and restores user or app state. |  | X |
| When the app is resumed after the device wakes from sleep (locked) state, the app returns the user to the exact state in which it was last used. |  | X |
| When the app is relaunched from Home or All Apps, the app restores the app state as closely as possible to the previous state. |  | X |
| **Please elaborate on how/where your app correctly preserves and restores user or app state:**  I preserve state in the Recipe Activity and the Recipe Detail activity. The Recipe Activity contains the Recipe Fragment. The activity passes the category id in the arguments to the fragment. I set it as an argument, such that the fragment is independent of the activity.  My Recipe Activity also contains a spinner which is what determines which category I passed to the Recipe Fragment. I save the selected spinner position in the onsaveinstancestate method of the Activity, and try to access the savedstate in the on resume and oncreate methods.  I also save the selected list position in my recipe fragment in the on saveinstancestate method, and check if the state was saved in the oncreateview method of the fragment, and subsequently set the selection in the load finished method of the cursor loader.  I also pass the recipe id to the RecipeDetailFragment arguments, and try to restart the loader from the arguments in the onresume method.  Also in the RecipeDetailActivity, I saved the recipe id and color in the onsaveinstancestate because these come from the calling intent and can be lost if the activity is recreated. |  |  |

### 

### Optional Components

To receive “exceeds specifications”, your app must fully implement all of the criteria listed under at least two of the four categories below (e.g. Notifications, ShareActionProvider, Broadcast Events, and Custom Views).

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Does Not Exceed Specifications** | **Exceeds Specifications** |
| Notifications |  |  |
| Notifications do not contain advertising or content unrelated to the core function of the app. |  | X |
| Notifications are persistent only if related to ongoing events (such as music playback or a phone call). |  | X |
| Multiple notifications are stacked into a single notification object, where possible. |  |  |
| App uses notifications only to indicate a context change relating to the user personally (such as an incoming message). |  | X |
| App uses notifications only to expose information/controls relating to an ongoing event (such as music playback or a phone call). |  | X |
| **Please elaborate on how/where you implemented Notifications in your app:**  I implemented notifications in my GcmIntentService in the com.tasteofuganda.gcm package. In the onHandleIntent method, if the message received from the GCM server is a notification, the app displays a notification to the user using the sendNotification method in the GcmIntentService. My server implementation sends a Your Recipe for the Week notification to all registered devices every week.  To try this out, simply hit the url <http://tasteofuganda.appspot.com/cron/notify> in a browser once the app has initialized on your device, and successfully registered the device on the GCM server. You should receive a “Your recipe for the week notification” |  |  |
| ShareActionProvider |  |  |
| Uses ShareActionProvider to share content with an outside application. |  | X |
| Makes use of Intent Extras to send rich content. |  | X |
| **Please elaborate on how/where you implemented ShareActionProvider:**  The action item is created in the oncreateoptionsmenu method.  I implemented a share action provider in my RecipeDetailFragment using a shareActionProvider. The createShareIntent method creates the share intent and adds the message as intent extras. The message “I just checked out <Recipe Name> + <Recipe share hashtag> is shared” |  |  |
| Broadcast Events |  |  |
| App intercepts broadcast events. |  | X |
| App responds to Broadcast events in a meaningful way. |  | X |
| **Please elaborate on how/where you implemented Broadcast Events:**  My app makes use of a WakefulBroadCastReceiver to respond to messages from my GCM server. The server sends either tickles or notifications which are then handled by the GcmIntentService. These are both in the com.tasteofuganda.gcm package. The broadcast events either trigger the sync adapter or create a notification. |  |  |
| **Custom Views** |  |  |
| App creates and uses a custom View. |  | X |
| App uses a novel View that couldn’t sufficiently be satisfied by the core Views in Android. |  |  |
| **Please elaborate on how/where you implemented Custom Views:**  My app uses a custom views for the detail fragment layout and for the recipe list item in the RecipeFragment. |  |  |