## 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 |
| App includes a tablet layout which takes advantage of the additional space (if possible). |  | x |
| App includes at least two distinct views and uses intents properly to move between these views. |  | 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:**  The following permissions were used:  android.permission.INTERNET – The app uses internet connection to search for other users on the platform. This permission enables such functionality to be possible  android.permission.READ\_SYNC\_SETTINGS - I needed to perform some update periodically. Updates like caching users new profile image locally from the server, checking if the user device has successfully register for push notification. SyncAdapter was used in implementing this functionality which requires the android.permission.READ\_SYNC\_SETTINGS,android.permission.WRITE\_SYNC\_SETTINGS,android.permission.AUTHENTICATE\_ACCOUNTS  android:name="android.permission.RECEIVE\_SMS – A way of verifying a user device is phone number a pin. The application intercepts incoming mail that matches a pattern, strip the pin to autofill the pin field and send the request to the server. To be able to receive broadcast from incoming SMS, I have to yse this permission.  android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES – The App uses Google play services. With the help of google plus sign in API, I am able to get users email and image url on google plus. The user have to provide this details as a means of authentication in situation where a change want to be made to a an account when their contact may not be available. The information is never shared with other users via the application. Using Google play services to implement this functionality, I have add android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES,android:name="android.permission.GET\_ACCOUNTS,android:name="android.permission.USE\_CREDENTIALS  android:name="android.permission.READ\_CONTACTS – The user may want to alert some group of friends about changes in their account. Finding an intent that allow multiple select for contact proved difficult. I have to use this permission to read the contact from the users phone and display to the user, enabling them to select more than one contact if they want to.  android:name="android.permission.WAKE\_LOCK – When a user receives a notification, the app has to wake the device in case the device was sleeping/idle.  android:name="android.permission.CALL\_PHONE – The application uses intent to call make call. Using the call intent requires this permission  android:name="com.google.android.c2dm.permission.RECEIVE- As a way of alerting uses about current updates from their friends on the platform, the app uses push notification. This permission enables the application to be able to make Google cloud messaging request, |  |  |
| **Performance and Stability** |  |  |
| App does not crash, force close, freeze, or otherwise function abnormally on any targeted device. |  | x |
| **ContentProvider** |  |  |
| App implements a ContentProvider to access locally stored data. |  | 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.  If it needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), app uses an IntentService to do so. |  | x |
| App uses a Loader to move its data to its views. |  | x |
| 1. **What's the content provider called, and how is it backed?**   The name of the content provider is ContactedEternalProviders   1. **What backend does it talk to? What is the SyncAdapter called? What mechanism is used to actually talk over the network?**   The sync adapter uses an image url to get an image and cache it locally. Also, if it is the first time it was able to obtain the app id of the device for push notification, it uses send it to the server at <http://my-eterna.appspot.com/registerPush> . It does this by using http connections in AbstractThreadedSyncAdapter.  **3) What loaders/adaptors are used?**  **CursorAdapter** |  |  |
| **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:**  The application correctly preserve app state on the mainactity. It has the ability to remember users last search. If a user was searching for someone before hitting the back button, on relaunching the app, the editText view still retain the text. |  |  |

### 

### 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. |  | x |
| 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:**  App uses notification to alert user about a friends update. The notification uses InboxStyle to help display more than one update to the user the same time. The implementation of this can be found on the GCM\_Handler package of the project |  |  |
| ShareActionProvider |  |  |
| Uses ShareActionProvider to share content with an outside application. |  | x |
| Makes use of Intent Extras to send rich content (i.e. a paragraph of content-specific text, a link and description, an image, etc). |  | x |
| **Please elaborate on how/where you implemented ShareActionProvider:**  The user uses the shareAction provider to tell friends about the name he/she have claimed on the platform. It simply shares a text and a link. This can be found on the callFragmentActivity of the application |  |  |
| 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:**  App uses both local broadcast receiver and intercepts broadcast event from system like the messages.  App intercepts messages when waiting for verification pin from the server. It only makes use of messages that matches a pattern.  App users localbroadcast receiver to make changes to the UI of an activity. The localbroadcast is registered when the activity start and unregistered when the activity is describe. When a background task completes and there is a need to update the a view eg, user image, it broadcast the event locally. If the broadcast is currently registered, an update is made. |  |  |
| **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. |  | x |
| **Please elaborate on how/where you implemented Custom Views:**  App uses a circular imageview to display user profile image and the notification icon.  The circularImageView extends the default android imageview, the view was redrawn by overriding the onDraw(Canvas canvas). Other attributes that specifies border, shadow were also included. |  |  |