## 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!

We were really impressed by your extra work, going the extra mile to write your own response to fit the requirements. This way you took a load from the JSON parsing. This was a nice approach, nevertheless, you will encounter situations where you will receive loads of data to parse and it will be easier to use a library, such as Volley. Below you’ll find feedback and some improvement recommendations.

### 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).  The sole purpose and functionality of the SearchView widget is to provide the user the possibility to search through a set of data. Please stay in this standards and rewrite the Action Bar Search functionality. Remove the validation and implicit addition of a car plate/event.  Our recommendation is to replace Search with a contextual Add button, +. That is, when the Car Plates list view is visible, tapping the + button will add a new Car Plate (you could additionally validate it before adding it). On the other hand, if the Detail view, event list, is in focus, + adds a new event. |  | **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.  The Navigation Drawer is a panel that displays the app’s main navigation options on the left edge of the screen. These options are usually static strings, actions that can change depending on the context. Please remove the Car Plates list from the content of the Navigation Drawer.  Our recommendation is to implement a ListFragment to represent the Car Plates list. |  | **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:**  <!-- access web service, download data -->  <uses-permission android:name="**android.permission.INTERNET**" />  <!-- verify network state before doing actual jobs -->  <uses-permission android:name="**android.permission.ACCESS\_NETWORK\_STATE**" />  <!-- reschedule any scheduled alarms at reboot -->  <uses-permission android:name="**android.permission.RECEIVE\_BOOT\_COMPLETED**" />  <!-- Vibrate at notification -->  <uses-permission android:name="**android.permission.VIBRATE**" /> |  | **X** |
| **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?**  Provider name: **EventProvider**   * Uses SQLite structured data to store car events   **2) What backend does it talk to? What is the SyncAdapter called? What mechanism is used to actually talk over the network?**   * The app pulls data from an API on request and uses an IntentService (CheckStatusService) to do so   **3) What loaders/adaptors are used?**   * Loaders : CursorLoader (CarEventsFragment, DetailsFragment, NavigationDrawerFragment) * Adapters : ArrayAdapter (CarEventsAdapter, CarSummaryAdapter, StatusEventAdapter) |  | **X** |
| **User/App State** |  |  |
| App correctly preserves and restores user or app state.  The data in the validation Activity like dialog is not preserved on rotate screen. |  | **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:**   * **NavigationDrawerFragment** - selected item, drawer state - preserved using saved instance state * **CarEventsFragment** - operations done on events - preserved using saved instance state (EventsContainer saved as part of the internal state and restored when the activity resumes) |  | **X** |

### 

### 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:**   * **NotificationManager** - events that need user attention are shown using notifications   (need user attention = an alarm was triggered)  - uses the notification service to show system notifications |  | **X** |
| 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:** | **X** |  |
| 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:**   * **CheckStatusService** : local broadcast the received data from API, StatusActivity has a BroadcastReceiver that can interpret the results sent by the service * **RebootReceiver** : listens for ACTION\_BOOT\_COMPLETED events and reschedules alarms for all existing car events |  | **X** |
| **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:** | **X** |  |

Here are a few suggested improvements for a better UI experience:

* The Navigation Drawer should only contain actions. For example, “Car Plates” to open the Car Plates list view, “Validate” to prompt the user to input a Car Plate number and “Settings”.
* Replace the Activity for validation like dialog with a real dialog.
* Remove the + Add event and the Save and Cancel buttons to the left and right of it.