## 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:**  As I had develop app which use web service so it is necessary for the app to have Internet connection, so I give possible network permission so my app can work properly. |  |  |
| **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?**  **2) What backend does it talk to? What is the SyncAdapter called? What mechanism is used to actually talk over the network?**  **3) What loaders/adaptors are used?**  1)it is called EquestProvider in my APP , and it is backed with SQL Lite with EquestDB.  2)It is used to connect a Database developed through SQL Lite , I don’t use sync adapter. I have used HTTP request through ASYNC Thread over Intent services and Activites.  3)Not Used |  |  |
| **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:**  Well I don’t do anything to get these functionalities , nor I have change the app cycle expect form some part where I have to restrict user to go back(As my Test taking Application restrict user to do so).  I override the Backbutton functionality to prompt the user that if he want to exit the app if yes ,the app finishes if not then he will be remain in same screen (which will be the test taking screen) |  |  |

### 

### 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:**  Notification is used over a service to notify user about Updated Results for offline version , the service which call notification is NotificationService and it is called in showAllUsers Activity.  I used NotificationCompat.Builder class to implement Notification my just seting text in it and activate it through my activity. |  |  |
| 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:**  Not used |  |  |
| 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:**  Not used |  |  |
| **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:**  As my App is rich with Custom UI interface , a simple view cannot able to handle the view which I want to create.  I had implemented Custom Text, Button, Edit and List view to implement a UI friendly and eye catching layouts.  I used android pre built classed and used them to implement my own custom views , like I had extend text view and set font feature in it to apply my own fonts same case for Edit text and Button View.  As for Custom list view , I had made my custom Adapters  These are custom views are named as:  CustEditView  CustomButtonView  CustTextView  TempListAdapter |  |  |