**Android Project Self-Evaluation**

**Instructions for Use**

Enter your name and submission number in the header. Complete the following rubric as a self-evaluation of your project for each of your three submissions. Remember that this self/peer/instructor-evaluation document must be submitted with your assignment to receive a grade. The following steps should be completed by each of a) yourself, b) two of your classmates (different people for each of the three submissions), and c) your instructor.

1. Enter a *Score* from zero to three for each row (component) of the rubric, based on the project specifications and point scale descriptions for that component.
2. Multiply the score by the weight for that component and enter the result in the *Score Earned* column. For rubric component rows with a weight greater than one, you may make minor adjustments (i.e. adjustments of less than the weight of that component) to the score earned to reflect your assessment on the degree to which you achieved the component.
3. Enter the sum of the scores earned in the *Total Earned* row at the bottom.
4. Complete the comments section at the bottom.

Your instructor will determine your grade on your third and final submission. Scoring less than 100% on your first two submissions is expected! Use your assessment as well as feedback from your peers and instructor to identify where you should focus your attention when making improvements (correcting defects, adding required components, new features to improve usefulness and UX, code organization, etc.).

**Rubric for Android Project**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Rubric Component** | **Point Scale** | | | | **Score**  **(0-3)** | | **Weight** | | | **Score Earned** | |
| **3** | **2** | **1** | **0** |
| *General* | Exemplary code organization/structure and efficiency | Adequate code organization/ structure and efficiency | Needs improvement in terms of code organization/ structure and efficiency | Inadequate commenting, poor organization/structure and code efficiency | You | 2 | | 1 | | | 2 |
| P1 |  | |  |
| P2 |  | |  |
| Ins. |  | |  |
| *User Interface and Event-Handling* | All user interface and event-handling requirements are met; very minor, or no defects identified | Most requirements are met, but there are a few notable defects | Some requirements are met, but there are numerous defects/errors | Incomplete/does not meet minimum level of performance | You | 3 | | 1 | | | 3 |
| P1 |  | |  |
| P2 |  | |  |
| Ins. |  | |  |
| *Data Persistence* | All data persistence requirements are met; very minor, or no defects identified | Most requirements are met, but there are a few notable defects | Some requirements are met, but there are numerous defects/errors | Incomplete/does not meet minimum level of performance | You | 3 | | 2 | | | 6 |
| P1 |  | |  |
| P2 |  | |  |
| Ins. |  | |  |
| *Connectivity* | The app performs Internet/web or another acceptable form of connectivity, to a relatively high degree of complexity. The processing is efficient (i.e. a separate thread is used) and application lifecycle is appropriately considered. | The app performs Internet/web or another acceptable form of connectivity, but not to a very high degree of complexity. | The app performs Internet/web or another acceptable form of connectivity at a basic level. There are lifecycle-related issues and/or threading (asynchrounous processing) is not properly employed. | The app does not effectively achieve the required connectivity requirements. | You | 3 | | 2 | | | 6 |
| P1 |  | |  |
| P2 |  | |  |
| Ins. |  | |  |
| *Usefulness and usability* | The app solves a problem, is useful, and the user experience (UX) is compelling. Required techniques (data persistence, connectivity) are used appropriately and are employed usefully within the context of the purpose of the app. | The app solves a problem, though the user experience could use some improvement. Required techniques are mostly used appropriately and usefully. | The usefulness and UX of the app could be improved. Required techniques are not all effectively employed. | The app is incomplete, does not have a well-defined purpose, offers an unintuitive UX, or does not employ the required features effectively. | You | 3 | | 1 | | | 3 |
| P1 |  | |  |
| P2 |  | |  |
| Ins. |  | |  |
| **Total Earned (max 21)** | | | | | | | | | 20 | | |

**App Description**

IWantThat is a shopping re-centralization app, which allows you to make requests for items you want to buy and have local and online stores searched to help you make your choice. The app will also allow you to compile a shopping list and try to figure out the best possible deals in regards to time, cost and travel distance.

**Your Comments**

Honestly im left rather disheartened at the moment. I current have 3 things that are on the testing build that were cleaned from this release so that it could run. A walmart API, current location of device for transit planning purposes, and a few more advanced google maps features. All that is currently in this release is a searchbar main menu (for which the only viable search entry is “PS4”), that leads to an example of what output will eventually look like when I start adding store API’s and the ability to find your way to said stores from my house using google maps destination API. But it seems they might have already locked out my free uses of that one so it might not work until I can get a billing option going. If all my daily uses are done with it will just black screen while it loads the map and then after it times out simply show a map with no icons or meaningful information. In the end I've wound up with a lot more knowledge of how my API’s function and a large sense of frustration at how difficult it is to get your own location on a phone. Hopefully I will have far more ready for you by the next checkpoint Jody.

Checkpoint 2:

Alright got almost all my API’s sorted and implemented a basic SQLite DB with a few basic electronic searches. Next up is adding the orderlist, sorting out my credit card issues with google so my directions code works, and hopefully finding more stores that will let me use their API.

Since my application with BestBuy was denied the app currently only works with Walmart inputs, all pricing is now live.

Example inputs for testing:

Playstation

iPod

Xbox

Telescope

Switch

Checkpoint 3:

Alright, full usability has been added to shopping list and the google issue has been sorted out. Everything should be fully functional and easily

navigable. My only regret is I wasen’t able to get another API to show my expansible framework for the code. The settings menu is also there and has a basic darkmode switch and a button toggle. While my UI might not be the prettiest I did go out of my way to make it as functional and user friendly as possible so please look at it in that light. Lastly I noticed that there might still be some small delays with the google api after selecting “I want that” they seem to have gone away after setting up a proper billing line with google but there might be some remaining slowness there.

**Peer Comments**  
For each submission you need to have *at least* two people – and different people for each of your three submissions -- complete this rubric for the assignment and enter their comments here:

e.g. Reviewer: John B. (submission 1): I think the app is coming along well so far. The layout and screens are well designed already. The connectivity component isn’t well defined yet and there are some errors saving data in some cases. I think a settings screen which allows saving of favourite locations would be a big UX improvement in terms of usability.

Reviewer: Niko H (submission 1): I think the app has a good starting point. There isn’t any connectivity yet. The basic framework is in place.

Niko H (submission 2): The search function is implemented quite well. Good Job!

**Instructor Comments**