**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 | 3 | 1 |  |
| P1 | 3 | 3 |
| P2 | 3 |  |
| 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 | 1 | 1 |  |
| P1 | 2 | 2 |
| P2 | 2 |  |
| 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 |  |
| P1 | 2 | 4 |
| P2 | 3 |  |
| 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 | 5 | 2 |  |
| P1 | 3 |  |
| P2 | 5 | 6 |
| 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 | 2 | 1 |  |
| P1 | 2 | 2 |
| P2 | 2 |  |
| Ins. |  |  |
| **Total Earned (max 21)** | | | | | | | |  |

**Your Comments**

Is there anything you are having trouble with? What do you think you did exceptionally well? What are you aiming to do for your next submission? Is there anything you would like your instructor to give special attention to when reviewing or evaluating your assignment?**:**

I am struggling a bit with firebase. It doesn’t perform as traditional database. Nothing I can’t handle,though.

Not sure what stands for “Connectivity - high degree of complexity”

**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.

P1: Jadon

* The app queries from multiple job sites very well, the ListView displays useful info and clicking redirects to the job page
* UI/UX could use some work - it’s a little plain at the moment
* Data persistence with Firebase works, but no relational complexity (typical for Firebase applications)
* Code is well organized and easy to read
* Looks good! This could be a very useful app when it’s completed.

P2: Minghua Lei

* Code is separated into multiple classes
* Very simple UI
* Shared preferences and firebase works
* App connects to firebase and RSS feed
* I think this will be useful in the future

**Instructor Comments**