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

**App Description**

I built a Gallery/Unsplash.com browsing application that gives users the ability to capture their own pictures and save them in their device’s storage. The Unsplash portion pulls data from the Unsplash API and integrates SQLite. I’m using SQLite to store user information retained from my application’s login authentication system. SQLite is also being used to store each photo a user marks as their favourite.

The application includes one setting, which is the screen mode. (Dark Mode, Light Mode) In order to set your changes, you have to go to the homepage and click the refresh button on the toolbar to reload the application.

**Your Comments**

**TO GET THE GALLERY PORTION WORKING, YOU HAVE TO SET THE PATH INSIDE OF THE showImages() METHOD INSIDE OF THE GalleryActivity TO THE PATH YOU’RE SAVING THE PICTURES IN WHEN USING THE CAMERA APPLICATION. (**I had to manually set mine to my phone’s path for my final demo**)**

Overall, this was a great experience. I wish I had more time to implement other features I had in mind. Thanks Jody!

**Peer Comments**

**Fred B.:**

App looks great!

**Danny:**

Nice work!

**Instructor Comments**