|  |  |  |
| --- | --- | --- |
| **AREAS FOR IMPROVEMENT** | **CRITERIA** | **AREAS EXCEEDING EXPECTATIONS** |
| Good. Just make sure that you try to use GitHub markdown formatting in future projects when necessary for organization and readability (headings are helpful, and using code formatting is good when referencing methods etc.) | REQUIREMENTS  All stated assignment-specific requirements have been met, including correct use of GitHub repositories and a well-documented README.md file. |  |
| Generally good, though I made some suggestions to keep in mind for future projects (companion objects on the top, private methods on the bottom of the class) | CODE ORGANIZATION  Code is sensibly laid out in individual files. Code is properly formatted and readable with good whitespacing. |  |
| Also good – especially in your Kotlin code. Note that it is more idiomatic to use snake\_case for view IDs in XML. | NAMING CONVENTIONS  Care has been taken to name variables and other constructs appropriately. File names are appropriate to code content and best practices. |  |
| Good, though with some repeated code that should be addressed in future assignments. | CODE TECHNIQUE  All code (both Kotlin and XML) is idiomatic and demonstrates techniques presented in readings and tutorials. There is no unnecessarily repeated code. |  |
|  | UI ORGANIZATION  The user interface is well-organized with a clean architecture. |  |
| Just watch out for hardcoded width and height values for views. This can be a problem on smaller screens. | UI USABILITY  The user interface is readable and is easy to navigate. |  |
| Good. Though all string warnings should be addressed in future assignments, rather than ignored. | VERIFICATION  The app compiles and runs properly with no significant errors. |  |