**Project Specification**

**Solution Design and README**

|  |  |
| --- | --- |
| CRITERIA | MEETS SPECIFICATIONS |
| README | The README file provides instructions on how to setup the Android Project and run the app. This instruction will be used by the evaluator to review your project. |
| App design images | The student submission includes images of the app design (sketches on paper are fine). |

**Idea Definition**

|  |  |
| --- | --- |
| CRITERIA | MEETS SPECIFICATIONS |
| App concept | The app idea is clearly defined in a short (approx. 300 word) document included with the submission. |

**Code Readability**

|  |  |
| --- | --- |
| CRITERIA | MEETS SPECIFICATIONS |
| Code formatting | The code is properly formatted i.e. there are no unnecessary blank lines; there are no unused variables or methods; there is no commented out code.  The code also has proper indentation when defining variables and methods. |
| Code readability | Code is easily readable such that a fellow programmer can understand the purpose of the app. |
| Naming conventions | All variables, methods, and resource IDs are descriptively named such that another developer reading the code can easily understand their function. |

**Code Functionality**

|  |  |
| --- | --- |
| CRITERIA | MEETS SPECIFICATIONS |
| Use of control flow statements | The app contains at least one properly used if/else statement. |
| Use of methods | The methods in Java files:   * Are defined and called * Do what their names suggest * Have appropriate input parameters and return data types |
| Use of variables | Variables are declared and scoped appropriately. |
| Interactivity | All Button(s) have onClick events defined and function appropriately. |
| Layout best practices | The code adheres to all of the following best practices:   * Text sizes are defined in sp * Lengths are defined in dp * Padding and margin is used appropriately, such that the views are not crammed up against each other. |
| Use of Views | The app includes at least four of the following Views: TextView, ImageView, Button, Checkbox, EditText, LinearLayout, RelativeLayout, ScrollView.  If applicable, the app uses nested ViewGroups to reduce the complexity of the layout. |
| Functionality | The code runs without errors. |
| Implementation of idea | The code meets all the requirements listed in the idea definition. |

**Suggestions to make your Project stand out**

* Incorporate views that were not explored in the class
* As part of best practices implement a custom theme in your app (primary color, primary dark color, and accent color) using colors from the Material Design color palette. Keep all strings defined in the strings.xml file, with no hardcoded strings in the XML layout or Java files.
* Use comments including Javadoc style comments to explain all parts of the app.