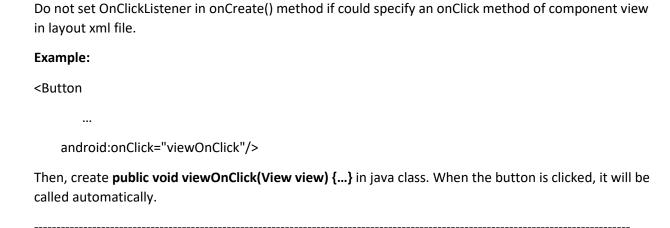
## **Project Naming Conventions and Professional Practices**

Instance fields of view components in java file should be named the same as their id.
Format: Camel case. "layout name + field name + component type".
Button -> Button,
All components that allow user to type (AutoCompleteTextView, EditText, etc.) -> Input
All components that don't allow user to type (TextView, etc) -> Text
<b>Example</b> : For EditText of first name field in add employee layout:
<pre>empFstNameInput = findViewById(R.id.empFstNameInput);</pre>
Define content description attribute for image view.
Format: Camel case. "layout name + field name + functionality".
Example: The arrow that select department in add employee activity can be named as empDeptSelector.
Do not use hardcoded text. Extract string resources and refer from <b>res/values/strings.xml</b> .
<b>Format</b> : Resource name should be the same as resource value, but with all lower case and separated by underscore.
<b>Example</b> : If a string is "First Name", the resource name should be <i>first_name</i> .
Do not declare instance fields that will only be used by one method. (Declare them as local variable)
All instance fields in java class should be declared as private except constant variables that are not subject to change and will be shared among other classes.
Format: For constant variables, all upper case, separated by underscore. For other variables, camel case.
<b>Example</b> : public static final String <b>TASK_TABLE_NAME</b> = "task";
private int <i>currDay, currMonth, currYear</i> ;



## **Important**

If there exists any case that this doc doesn't cover. Please find the existing naming convention of similar scenarios in the project in order to keep coding style consistent, which helps peer review and reduce confusion.