

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

Example: For EditText of first name field in add employee layout:

```
empFstNameInput = findViewById(R.id.empFstNameInput);
```

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 **res/values/strings.xml**.

Format: Resource name should be the same as resource value, but with all lower case and separated by underscore.

Example: If a string is "First Name", the resource name should be ***first_name***.

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.

Example: public static final String ***TASK_TABLE_NAME*** = "task";

```
private int currDay, currMonth, currYear;
```

Do not set OnClickListener in onCreate() method if could specify an onClick method of component view in layout xml file.

Example:

```
<Button
```

```
...
```

```
    android:onClick="viewOnClick"/>
```

Then, create **public void viewOnClick(View view) {...}** in java class. When the button is clicked, it will be called automatically.

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.