

## Style Guide for Flower Finder

James Wang, Luke Stein, Mingrui Chen, William Ye

Consensus reached 6/23/2020

This guide, written collectively by the entire development team, is designed to give a uniform appearance of our code written by different engineers. This helps in readability and maintainability, thus reducing the time it takes to detect and fix errors and increasing the efficiency for all in the group. This also ensures quality code and promotes good programming practices. These guidelines also apply to JUnit testing.

1. Do not use global variables wherever possible - there should be no reason to use global variables in our application.
2. As a subset of point 1, always consider scope of functions/methods; methods should be private. This helps with security of our program.
3. Consider efficiency of all algorithms. Mobile applications are restricted in computation and physical power. Reduce complexity when possible.
4. Name all functions/objects/variables adequately descriptively with standard camelcase in the Java/Android app. Start with lower-case and capitalize each additional word. For constants, use standard all-caps and underscores notation. Functions should start with a verb (e.g. `getPicture(String path)`).

For the Python/PyTorch model training script, due to the widespread use of underscore naming conventions used in PyTorch, use all lowercase naming with underscores.

```
//Status and permission codes
private static int RESULT_LOAD_IMAGE = 1;
private static int CAMERA_PERMISSION_CODE = 100;
```

```
//UI Components
private Button loadImageButton;
private Button takePictureButton;
```

5. Follow JavaDoc conventions for every function in the code. This should include a description of the function, the parameters, and what it returns. The first sentence should always be a summary of the function. Include any information about anonymous inner classes if used. Note that the notes are in html format.

```
/**
 * Function to get the model
 * @param modelName is the name of the model to get a path to
 * @return the model path as a String
 */
public String fetchModelFile(String modelName) {
```

6. Comment any variable or algorithm that may be unclear to a reader in the group. Anything that might possibly be ambiguous should be reconsidered for clarity and at a minimum, well documented/commented.
7. Catch any exception that represents an error that can be solved by the user - e.g. issues arising from permissions.
8. Use @Todo tags if there is any component not completed, meaning not ready for release.
9. Ensure uniform indentation. Set all tabs to the equivalent of four whitespaces.
10. Limit the length of each line to 80 characters long. Break long statements into multiple lines in a logical fashion.
11. Start all conditional blocks with a starting parenthesis on that line.