Code quality report

Mentorship System

by Bartek Pacia Google Code-in 2019

This document lists issues found in mentorship-android project by Codacy and CodeClimate, online automated code review tools.

Overview

From an architectural point of view, mentorship-android is a great example of well-done project. It follows Model-View-ViewModel pattern recommended by Android Team and rules like separation of concerns.
However, inspections performed by Codacy and CodeClimate revealed that the codebase contains a noticeable amount of code smells, which are parts of code that are written in unclean way and/or doesn't follow best practices. Even though it doesn't impact end-user experience, it decreases comfort of work of programmers working on the project. That's why such issues should be fixed, because they contribute to technical debt.

Lack of documentation

In general, the project is very well documented, but they are some places missing description, namely: `utils/Constant.kt`, `activities/AboutActivity.kt` and `activities/SettingsActivity.kt`. Adding it won't be a lot of work.

Duplicated error handling logic

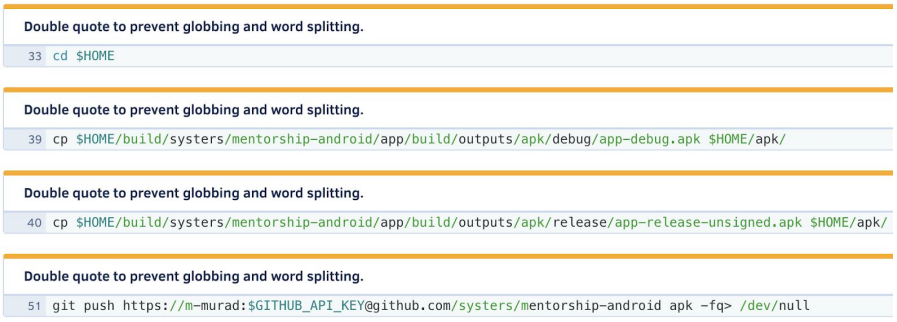
The app extensively uses RxJava, which introduces, together with undeniable benefits such as reactivity, a lot of boilerplate. Most of it is error handling code which is scattered all over the `ViewModel`s. It occurs 11 times and always looks the same:

```
override fun onError(throwable: Throwable) {
    when (throwable) {
        is IOException -> {
            message = MentorshipApplication.getContext()
                    .getString(R.string.error_please_check_internet)
        }
        is TimeoutException -> {
            message = MentorshipApplication.getContext()
                    .getString(R.string.error_request_timed_out)
        }
        is HttpException -> {
            message = CommonUtils.getErrorResponse(throwable).message
        }
        else -> {
            message = MentorshipApplication.getContext()
                    .getString(R.string.error_something_went_wrong)
            Log.e(TAG, throwable.localizedMessage)
        }
    }
    successful.value = false
}
```

As a solution I will list Pull Request #404 where I rewrote the app to use Kotlin Coroutines, entirely removing RxJava. While doing that, Anna Bauza came up with a solution to reduce error handling boilerplate – she suggested creating an extension method on Android's Context class. I added that method in this commit and implemented it everywhere in the next one.
It resulted in a great reduction of complexity and duplication which troubled ViewModels.

Shell script problems

Script generate-apks.sh contains 4 mistakes, which may lead to unexpected behavior on some systems. They are related to a sh concept known as globbing.



All it takes to fix those mistakes is to wrap paths with double quotes.

Formatting problems

This is something that was raised many times in Pull Requests made by Google Code-in participants. Oftentimes, developer has auto-formatting enabled in their IDE and whenever they edit a file, everything is formatted. If developer is inattentive, it leads to huge diffs, even when making small changes.
To fix this problem, I suggest establishing a uniform code style guide for the project. Official Kotlin Style Guide is the best option, because, well, it's official (and it just works well).

Short information about formatting should be adding to the project's README with the information on how to enable it in Android Studio:
Settings > Editor > Code Style > Kotlin > Set from > Predefined Style > Kotlin Style Guide

Summary

Mentorship Android app doesn't contain a lot of code smell and issues. Those which exist are rather quick and easy to fix. I hope this document will be shared publicly or that issues listed here will be made into GCI tasks, so that all those ugly pieces of code will be fixed.

Quality report on Codacy can be found here.
Quality report on CodeClimate can be found here.