

Introduction

You are sitting in front of the project to which we have a couple of tasks. Each of them requires you to analyze the code and problem-solving skills

We have three tasks that verify your knowledge about the iOS platform and put to the test your developer skills:

1. Fix & enhance the loading of the collection of Avatars
 1. Knowledge of UIKit
 2. Optimization skills
 3. SOLID
2. Fix broken scene
 1. Knowledge about concurrency using Dispatch API
 2. Debugging skills
3. Cover an avatar downloader with unit tests.
 1. Swift language proficiency
 2. Experience with unit tests

Important when doing and finalizing tasks:

1. There are no limitations to change implementation to any other tools, methods, or frameworks. The only limitation is not intact logic which is being made by the current solution.
2. At the end of your work code must compile
3. In case of troubles with resolving any of the tasks don't hesitate to share your results even with solutions for not all of the tasks
4. Our recommendation is to tackle task number 4 whenever You finalize work on the previous ones
5. Tips are here to help You, it is not mandatory to follow them if You decide otherwise

Tasks details

1. Fix & enhance the loading of the collection of Avatars

Description:

The first task is related to the list of avatars. When a user is scrolling down the list a few problems are appearing:

1. Avatars might be displayed in a wrong place (assigned to different person)
2. Loading of the avatar seems to be not so optimized

Where to start: ListViewController and AvatarCell

Tips:

- Feel free to modify the code however you like. The code breaks many architectural rules, so you can apply your improvements.

2. Fix broken scene

After selecting the avatar on the list, a user is redirected to the details view. When the view is opened, there are a few problems:

1. The user expects to see details related to the selected avatar, like the number of repositories and followers, and instead of the details, they see a blank view.
2. The UI is not responsive.
3. Labels are loaded in random order.

Where to start: `DetailsViewController`

Tips:

- After diagnosis and fixing problems, how can it be improved and simplified using modern API, like Combine?

3. Cover an avatar downloader with unit tests

In order to assure our code does what we want, we need to cover it with unit tests.

Currently, the code is untestable. You need to modify the code so that you can verify different test cases.

Class to test: `AvatarDownloader`

Requirements:

Write tests only for downloaded image data.

Tips:

- How can you control what you get from the network?

4. Architecture challenge

This task requires You to improve the architecture of our project. You can use any tools, frameworks, or patterns to do that. You can choose a specific part of the app based on which You want to demonstrate your skills. You can focus on the specific layers or whole project.

Requirements:

1. Architecture of the project after changes is enhanced on any level
You will choose
2. Code after changes must compile
3. In the main controller we need a small note describing what You did and why in this way.

Scope of classes: the whole project

Tips:

- This is about showing architecture skills and the way You might think about it. It is not necessarily changing everything in the project.