IOS Developer - Case Study

Please first read the following document to get the intro about Lean Scale and how the case studies will be handled:

https://docs.google.com/document/d/1Ury8T09UrAKi8JsP2Fn6UkZhsA4qcHaUfmtPjtZouv8/edit?usp=sharing

After reading and answering Generic Questions, please see below the “Role - Specific Questions”. “Optional, **(O)**” and “Mandatory**(M)**” signs are relevant in the questions below as well.

Technical Test Case - Role Specific Questions:

Instructions:
1- Use git, commit often.
2- The solution should be a mobile application which was written via Swift 4.2 or higher versions.
3- Please solve it as it’s a real problem.
4- Please provide documentation for your code. Tell us what decisions you have had. (for example: Which architecture you preferred. Which design pattern you used. etc..)
5- There should not be any warnings or errors.
6- Your min ios version should be iOS11.
7- Please do not use 3rd party libraries as much as possible.
8- Please do not write the best code in the world. However, write your code by showing much sensitivity as if you are distributing this to production or as if this project is going to be evaluated by you.
9- The design should be the same as far as the design we have sent you.
10- Please support multiple lines for the label which needs it.
11- Please provide unit tests, and be sure they are all working.
12- The app should support both iPhone and iPads.

The Question:
Please provide us with a mobile application which has 3 screens.

The first screen should contain two tabs. One of them is a list of elements from the response. The other one is the favourites tab.

The first tab:

There should be a search bar. When a user types a string with more than 3 characters, you should search for that string. (Eg: GtaV should list gtav search content.)

Cells should contain image, name, genre(If more than one genres exist, please combine their names like “Action, Shooter”), metacritic ratings.

You should show a different background colour for the items’ cell that the user has already opened its detail. (use the color we have sent you on figma.)

Add pagination support for search screen.

When the user taps an item, it opens the detail screen. (push or present, no matter.)

Bonus: Offline first approach will let you get more points.

The second tab:

That’s the favourites tab.

There is no need for background colour change operation we mentioned for the first screen.

User can remove a favourited game using swipe to delete. (Please ask the user if it is sure.)

Favouriting a game does not need a service call. You can store that information anywhere on client-side.

The detail screen:

When the user taps an item, it opens the detail screen. Detail screen will need another service call which we will share below.

If the game has already been favourited, write “Favorited”, write “Favorite” vice versa on the top right bar button. Bar button should do the work what it’s been written on it.

There is no need for in-app browser for the links “Visit website”, “Visit Reddit”. You can open safari, directly.

Game description area can support 4 lines.

Bonus: Adding a “Read More” option for game description cell let you get more points.
Bonus: Use single column for portrait mode. Supporting double columns in lanscape mode will let you get more points.

API's:

https://api.rawg.io/api/games?page_size=10&page=1 (default request)
https://api.rawg.io/api/games?page_size=10&page=1&search=gtav (search for "gtav" string)
You can paginate nex page using the "next" field in the response. Or you can create a different way of it.
https://api.rawg.io/api/games/3498 (getting the details of the game)

Bonus: Managing api urls using a build configuration or target configuration will let you more points.

UI:

https://www.figma.com/file/WT769GcbH3biF1wiw82QVm/LS-Code-Case-UI?node-id=0%3A1

Documentation:

A few tips to help you documenting your code.

- How did you decide to use that design and architectural patterns?
- What should be the part of this app that needs more time to develop or improve?
- Which part did you like most in this app?
- Does this app ready to submit to store? If not, what should be done to achieve that?
- Do you have any comments to us?
- What are the things you think are missing or open in this assignment?