Design Task - iOS Application Development

Guidelines

- The candidate should capture the results of the task in a feasible format and share them with the Xgrid team member. Please note that the spirit of this test is to give you interesting technical challenges.
- Be prepared to explain your approach and outcome, if asked by an Xgrid team member
- Do not hesitate to ask questions if clarification is needed.
- Timelines: You have 8 days from the date of issue of the test to complete it. You are welcome to submit your results early if you have completed the test. We understand that you will be completing these tasks while managing other work and personal commitments. So, feel free to work with your Xgrid contact, if additional time is required. Please keep in mind that the general assumption we make in your evaluation is that you asked for the task and worked on it during your free time.
- Good luck and have fun! We hope you will enjoy the tasks as they are a brief overview of what you could be doing if you join our team at Xgrid.

Task

The candidate is required to design and develop an iOS mobile application based on the UI wireframes provided in the Appendix section of this document.

Description of Mobile Application

1. The mobile application needs to be designed for an admin of an Xgrid Football League i.e., XFL.

2. The goal of the application is to provide the admin, an interface to view league information and make changes at run time.
3. Admin can
   a. Login with username and password (Login screen should be as per high fidelity mockups)
   b. View Fixtures (should be as per high fidelity mockup)
   c. View Match details
   d. Start, Pause & End the match
      i. These buttons should not be available once the match has ended
   e. View Remaining time for the ongoing match
   f. Update the goals for the ongoing match
      i. Admin should not be able to update the goals once the match has ended

4. User Authentication should be done from the server, and all the data will be updated on the live server if the internet is connected. You can choose **Firebase** for data storage and retrieval, or you can create your own API server. Being a senior developer, you need to take special consideration when creating the architecture for the network/service layer.
   a. You should easily be able to replace the networking SDK implementation e.g., Alamofire with AFNetworking.
   b. Your network layer should handle all the requests and responses. Do not expose the headers and body for any request at the view/view model layer. Network layer should parse and return the response that can be seamlessly used at the view/view model e.g., an array of matches.
   c. In future if you want to remove the network calls and get the data from the local database (e.g., Core Data), you should be able to do that without making any change in the view/view model.

    d. Your code should be reusable

5. The included wireframes have around 3 screens and as you will observe the app has further flows (not shown in the wireframes), you are not expected to complete further flows. You may however add data points and flows if you see fit.

6. While the wireframes show various navigation options, feel free to get creative and use gesture, etc based navigation if you see fit.
7. The end result needs to be a completed iOS application (Swift), we are not looking for a web app developed for the mobile ecosystem.

Additional Requirements

Note that the design aspect is intentionally left open-ended instead of dictating specifics. We would like to see what kind of design decisions and conventions you end up adhering to. You will be specifically judged on the basis of your aesthetic sense, UX design, and UI implementation. Please note that architecture is very important for this assignment. As you have wireframes attached along with the color palette, free to choose basic button designs etc that are easily available.

All the design decisions/explanatory parts of the solution should be part of a ".md" file on the Github project. The goal is to create a working demo with the necessary build setup. Prioritize showing a working demo of the setup over completing all the needed features. Create README/installation instruction for the tools being used.