iOS Swift Programming Challenge - Junior

Functional requirements

Create a simple iOS app that interacts with the Internet Chuck Norris Database API (**http://www.icndb.com/api/**)

The Challenge consists of three tasks, each has a button on the home screen:

1) Random Joke

When the 'Random Joke' button is pressed, the app fetches one random joke from the Chuck Norris server and displays the random joke in a popup with an OK button.

Use the following end point

http://api.icndb.com/jokes/random?exclude=[explicit]

2) Never-ending Joke list

When the 'Never-ending Jokes' button is pressed, the app opens a new view that contains a list of random jokes. Jokes should be requested asynchronously, in batches from the server. When the user scrolls to the bottom of the list, the list shows a loading message indicating that more jokes are being fetched.

Since the jokes are random, it is fine in this simple task to have duplicate jokes in the list.

Use the following end point

http://api.icndb.com/jokes/random?exclude=[explicit]

Technical requirements

The application should be written entirely in Swift 4.x+ and able to run on any iPhone running iOS 10.0 or greater.

The app layer should support iPhone (required) and iPad (nice to have, but not required).

Evaluation criteria

- The solution should be of high quality, suitable for production. This means that it should be stable, account for edge cases, unhappy scenarios and be responsive and fluid. Make sure you are paying attention to detailIf you are unsure about anything, use your own judgment to create the best solution. This is just a test to show your approach to problem solving, design and software development

- We would like to see business logic covered by **Unit tests.**

  Use any testable architecture (eg, MVVM, MVVM+C Viper or clean architecture) however, we required you to rely solely on the native XCTest framework (no external libraries are allowed for unit tests)

- Your code should demonstrate best-practices both from an iOS and a general software development perspective.

- For design/UI, you can use the Coop application as a reference.

  .

Deliverables

The completed project can be delivered as a zipped file or in a GitHub repository. A readme file should be included detailing how to simply build the app.