iOS Assessment for PlanRadar vacancy – READ CAREFULLY

Task: Create a simple weather app

Technologies that must be used:

Objective-C **and** latest Swift, CoreData, AFNetworking, latest XCode

Dependency Management: CocoaPods

CocoaPods that must be used:

AFNetworking (https://github.com/AFNetworking/AFNetworking)
SVProgressHUD (https://github.com/SVProgressHUD/SVProgressHUD)

Weather API: https://openweathermap.org
API Key: f5cb0b965ea1564c50c6f1b74534d823

Endpoints to be used:

1) GET: api.openweathermap.org/data/2.5/weather?q=<CITY_NAME>
2) GET: http://openweathermap.org/img/w/<ICON_ID>.png

Specification

The user should be able to maintain a list of cities where he can get the current weather status as well as historical weather data. With historical weather data is meant all weather requests the user has made with the app.

Create a UITableViewController (Rootcontroller of app) where the user can add/remove City names (I have tested with London, Vienna, Paris -> all giving valid responses for Endpoint 1).

After Click on an added City Name show a Detail Controller displaying the following information from the response of endpoint 1:

- weather.description
- main.temp (response is kelvin -> convert to celsius)
- main.humidity (in percent)
- wind.speed

The request must be asynchron, during the request show a progress HUD.

Save every response for each city to CoreData (in addition to weather data save request date and time for the list of historical data). Show all historical received weather data by clicking on a UITableViewCellAccessoryDetailDisclosureButton that has to be added in the UITableViewCells of the city names.

Create a CoreData model with the 2 related Entities City and WeatherInfo.

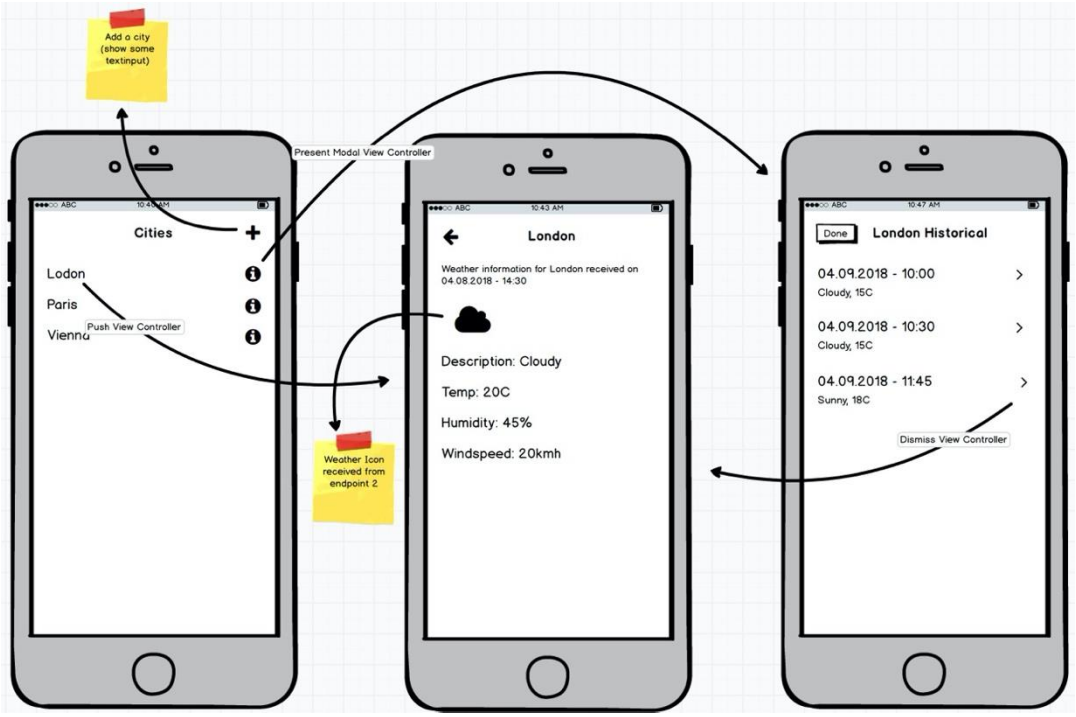
Create a simple Class that uses AFNetworking and deals with the 2 endpoints and create a model encapsulating the needed response data.

Create a local git repo and add meaningful commit messages during your work on the assessment (Git logs are checked on evaluation). Do everything on master branch.

A bonus is if you don't use interfacebuilder or storyboards, but do UI programmatically

As an architectural Design Pattern MVVM must be used. You can decide wich code parts you do with Swift and what you do with Objective-C – it is a must that both languages are used.

Find UI specs and assets in /UI_specs/dark or /UI_specs/light, check the index.html – You can decide if you implement light or darkmode (or both



Assessment Grading

- **App Architecture & Codestyle/quality & Documenation of Code**
- **User Interface**
- Error Handling
- Interpretation of specification
- Tests

The best assessment wins the Job! For any Questions on the assessment:
d.candotti@planradar.com

Delivery

Zip File containing the git repo. Do not include Pods, just the podfile we will hit pod install.
Zip File naming: <firstname>_<lastname>.zip