INDIVIDUAL CODING EXERCISE – WEATHER APP

Summary:
Design and build a mobile application that shows relevant weather forecast to the user

Recommended Duration:
2 weeks

About the activity
This exercise is about building an app that meets a very simple, straightforward and specific use case – show the weather forecast based on the user's current or selected locations. The app's functionality is intentionally scoped this way so that the developer can focus on designing and implementing a creative user experience and make careful considerations about the application's architecture instead of worrying about implementing a large set of functionalities

Evaluating the Results
This document includes a set of very high level **functional, technical and design guidelines** meant to help the developer in building the app. Take this opportunity to dive deeply into the relevant aspects of iOS's core framework, UI design principles and architecture patterns and explore possible solutions needed to meet these requirements. After the activity, each developer will demonstrate the application and explain the design and technical choices made.

Tip: Focus on designing and building your app instead of creating icons and other image assets yourself. This app is not meant to be published so make use of generated, open source, public domain or royalty-free assets as appropriate

FUNCTIONAL

The user should be able to launch the application immediately know the weather information at the user's general location

The user should be able to manage (view, add, remove) a persistent list of other locations of interest

The user should be able to see the weather forecast at his current location for the next 5 days

The user must still be able to view the app's last received weather forecast for a location even while offline

The user should be able to search for other locations and check the current weather forecast at a selected location

The user should be able to set basic settings about the app: temperature scale (C or F) and app theme (standard or dark)

The user should be able to search for and see the 4 or 5 day weather forecast for a selected location

The app should request app permissions (e.g. location) only if needed by a feature requested by the user

DESIGN AND UX

The app's design will follow the Apple Human Interface Guidelines.

The app's UI design must make use of transitions, animations and motion to convey responsiveness and app feedback

The app should make use of appropriate graphics, imagery and iconography to convey information instead of text and labels

The app must have clear, visual indicators as to the state of the application and the freshness of the app's displayed information

The app must have a definitive color theme. The app must have both a standard and dark theme.

The must be optimized for smartform formfactor and supports portrait orientation only

TECHNICAL

The app must use information available from the free-tier APIs provided by OpenWeatherMap or Singapore Real Time APIs

The app must not exhibit any memory / resource leaks.

The app is allowed to use 3rd party services of other APIs as appropriate to support its functionalities (e.g. Google Maps)

The developer is encouraged to use native libraries to use as networking library. But should a networking library be needed, Alamofire is highly recommended

The app's architecture must use MVVM pattern. For data binding, it is recommended to use RxSwift

The app must make use of local data caching in order to reduce the need to make expensive network calls.

Strings, images, colors and other non-code resources should not be hard-coded and instead referenced as external resources

The app must be responsive, maintaining a smooth framerate of at least 60fps as measured by built in profilers

REFERENCES

Design
Weather App Designs Concepts 1
Weather App Designs Concepts 2

iOS Core
Apple Developer Guide
App Permissions
Alamofire Networking Library
MVVM Architecture

Data Sources
OpenWeatherMap
Singapore Real Time APIs