Project Overview

In this capstone project, you will create an app of your own design that showcases your iOS development chops. The app should demonstrate solid proficiency in these four areas:

- Building sophisticated and polished user interfaces with UIKit components
- Downloading data from network resources
- Persisting state on the device
- Researching and integrating new libraries

You are encouraged to find an interesting restful API to interact with. Social media sites, weather, sports, photo repositories, municipal data, and imdb are provide examples of APIs you can use. The app should provide some kind of interactive experience with the data. The user interface should contain multiple view controllers. We anticipate that most user interfaces will use navigation controllers to structure multiple pages of content, and will include tables, collections, and or tabs. But any use interface that provides a useful and interesting way to interact with the data is acceptable.

The project should have some requirement for persistent state. If the app is shut down and restarted, it should not need to refresh its state entirely from the network data source.

The app should also be submitted with a readme.txt file that describes experience that the user is expected to have, and any particular instructions a use might need.

Important Note for International Students!

For your capstone project, if you intend to build an app that has a user interface in a language other than English, or that uses a data source which is not accessible internationally, please check with us before you begin work on the app. Email **ios-project@udacity.com** and include the following information:

- A brief description of the app
- What language the user of the app is assumed to know, and/or
- What data source is being used.

This will enable us to verify whether we have a reviewer to whom your project submission can be assigned, and to prevent delays in the review process.

Why this project?

This project is open-ended, so it provides an opportunity for you to showcase your iOS abilities. It allows you to make expressive, engaging user interfaces, and to gather and generate compelling data.

What will you learn?

- Researching and leveraging a new framework or library
- How to design and build an app from the ground up
- Finding sources of networked data

Why is this project meaningful to my career?

Landing a job as an iOS Developer requires more than technical chops; it also requires a certain amount of creativity and innovation. Having an app 100% of your own creation in your portfolio demonstrates a passion for the field that will set you apart from your competition.

App Specification: Capstone App

iOS Developer Nanodegree

[Note that this is an informal app description. It will give you an idea how the app should work, but when you submit your app it will be rated based on the Rubric.]

The fifth portfolio app is open-ended. Let your own interests guide you. You will invent the app, then design and build it. We hope that you choose an idea that is motivating and challenging.

The app should showcase your iOS skillset. Your app should have a complexity similar to the On The Map app and the Virtual Tourist apps, and should include code from the following areas:

User Interface

Your app should demonstrate that you can combine the essential UIKit components in effective ways. It should include the following common UI features:

- More than one view controller
- A table or collection view
- Navigation and modal presentation
- Image assets in 1x, 2x, and 3x formats. Or in vector format.

Networking

Your app should incorporate data from a networked source:

- Choose an API and integrate downloaded data into the app
- Give users feedback around network activity, displaying activity indicators and/or progress bars when appropriate, and an alert in case of connection failures
- Encapsulate networking code in a class to reduce detail in View Controllers

Persistence

Your app should incorporate data that needs to be persisted between runs of the app.

- Include an object graph that can be persisted in Core Data
- Manage the Core Data Stack outside of your view controllers, either in the App Delegate or in a separate Core Data Stack manager class
- Aside from your primary app state, you should find some additional state that can be stored outside of Core Data, either in NSUserDefaults, or in the documents directory using an NSKeyedArchiver

README

The app should also include a README file within the project's directory. The README should accomplish two goals:

- Describe the intended user experience
- Include all specific actions and/or commands necessary for the reviewer to compile, run, and access any aspect of the project

To learn more about creating README files, check out Udacity's course on Writing READMEs.

App Functionality

The app should function as described in the README. No crashes or other runtime errors should be evident.

PROJECT SPECIFICATION

You Decide!

README file

CRITERIA	MEETS SPECIFICATIONS
Does the app contain a README file that describes the intended user experience in the app?	The app contains a README that fully describes the intended user experience. After reading the document, a user can easily use the app.
Does the README detail all steps needed for the reviewer to build, run, and access the app?	The README provides all necessary information to enable the reviewer to build, run, and access the app.

User Interface

CRITERIA	MEETS SPECIFICATIONS
Does the app contain a user interface with multiple pages of content?	The app contains multiple pages of interface in a navigation controller or tab controller, *or* a single view controller with a view that shows and hides significant new content.
Does the app use more than one type of native control in its user interface?	The user interface includes more than one type of control.

Networking

CRITERIA	MEETS SPECIFICATIONS
Does the app interact with a networked API that provides the app with data?	The app includes data from a networked source.
Is the networking code in the app encapsulated into networking classes?	The networking code is encapsulated in its own classes.

CRITERIA	MEETS SPECIFICATIONS
Does the user interface clearly indicate network activity, particularly in situations where the user is waiting for data?	The app clearly indicates network activity, displaying activity indicators and/or progress bars when appropriate.
Does the app alert the user in case of network failure?	The app displays an alert view if the network connection fails.

Persistent State

CRITERIA	MEETS SPECIFICATIONS
Does the app store persistent state using Core Data or a service with local persistence capabilities (e.g. Firebase or Realm)?	The app has a persistent state that is stored using Core Data or a service with local persistence capabilities (e.g. Firebase or Realm).

App Functionality

CRITERIA	MEETS SPECIFICATIONS
Does the app function as described in the README, without crashes or other runtime errors?	The app functions as described in the README, without crashes or other runtime errors.

Suggestions to Make Your Project Stand Out!

- Use some custom UI controls. There are hundreds of custom classes available on GitHub for you to try out; just remember to attribute them correctly. If you are feeling adventurous, try writing your own!
- Combine data from two or more networked sources in your app. For example, if your app can show walking directions to get to a restaurant, and the distance is more than 2km or so, it would be nice to present the user with the option to use a ride-sharing service such as Lyft or Uber.
- Persist a sophisticated data model for your app so the user gets a great experience even if there's no internet connection.
