**Project 1: Founders Directory**

**Objective**

In this assignment I want you to get more hands-on experience, including generally getting more experience with Xcode and Swift. Take a look at the app prototype that one of the Rollins Center student employees built a while ago: <https://projects.invisionapp.com/share/Y98HJ9Y3X#/screens/185914892>

In Project 1, I’d like you to think about overall app design, build an app framework, and then concentrate on implementing the “view profile” and “edit profile” MVC units. Here are the prototype’s screens for these two features:

  

**View Profile Edit Profile Edit Profile (scrolled down)**

In addition to View Profile and Edit Profile, you’ll want some kind of app framework that includes either a collection view or a table view giving a list of Founders in the directory. You can code up a static table view if you’d like, and have all clicks take you to a View Profile with the same dummy data each time. Or you can get more creative and do something more comprehensive. In a future iteration, we’ll get the list of Founders from a web API. This prototype is just that: one version of how this app could work. Please be creative and think about the user experience. What do **you** think will work well? The prototype does **not** conform to Apple’s HIG standards. You’ll need to construct your own design for this app.

**Required Tasks**

1. Create a Founders Directory app with a storyboard that contains the following scenes:
   1. A list of Founders
   2. A Founder profile page (e.g. to be displayed when you tap on a Founder)
   3. An edit profile page (e.g. to be displayed when you edit your own profile)
2. Supply your own data for a list of names, photos, phone numbers, etc. Make the data look reasonable and plausible (e.g. an email address should have the format “name@domain.com”, and a phone number should look like “801-555-1212”).
3. Include the following elements on the view profile page:
   1. Photo
   2. Name
   3. Company name
   4. Phone (may be unlisted)
   5. Email (may be unlisted)
   6. Spouse name (if available)
   7. Business profile (multi-line description of their business background)
   8. Optional: home and work contact info, year joined Founders Organization, Founder status (Bronze, Silver, Gold, Platinum, Diamond)
4. On the view profile page, include links that allow the user to call, text, or email the Founder if the corresponding contact information is listed.
5. Code up the view-profile function to call a Founder.
6. Code up the view-profile function to text a Founder.
7. Code up the view-profile function to email a Founder.
8. On the edit profile page, include the ability to edit the same information you display on the view page. Minimally this must include photo, name, company name, phone, email, spouse name, and business profile.
9. On the edit profile page, let the user choose whether to make phone and/or email unlisted. The default should be to have the contact info listed.
10. On the edit profile page, let the user choose an image either from the camera roll or from the camera (i.e. taking a new photo). You’ll want to learn about Apple’s UIImagePickerController to do this.

**Rules**

1. You are not required to use Autolayout for this assignment. You may assume a particular device size and orientation for this project if you prefer. A future project will require you to use Autolayout. But you *may* use Autolayout if you’d like.
2. You may load all your image assets into the project manually. Required Task #10 asks you to let the user choose an image or take a photo, but you’re not required to save the photo. The new image is only required to be temporarily present in the edit page’s UIImageView control. When you leave that scene, it can revert to the default image you’ve provided in the app. I’ll show you how to save it permanently later.

**Hints**

These hints are not required tasks. They are completely optional. Following them may make the assignment a little easier (no guarantees though!).

1. Calling, texting, or emailing from your app is a lot easier than you might fear. Google around for help on these subjects, and you’ll find some one-liners to help.
2. Taking a picture or choosing from the camera roll is also easier than you might fear. It’s not a one-liner, but here’s an example of a tutorial that walks you through the process: <http://bit.ly/2cukq0u>
3. Economy is valuable in coding: the easiest way to ensure a bug-free line of code is not to write the line of code at all. This assignment requires more lines of code than the homework has so far, but still not an excessive amount. If you find yourself writing many dozens of lines of code for some feature, you are probably on the wrong track.

**Evaluation**

In all of the assignments, writing quality code that builds without warnings or errors, and then testing the resulting application and iterating until it functions properly is the goal.

Here are the most common reasons assignments are marked down:

* Project does not build.
* Project does not build without warnings.
* One or more Required Tasks was not completed correctly.
* A fundamental concept was not understood.
* Code is sloppy and hard to read (e.g. indentation is not consistent).
* Your solution is difficult for someone reading the code to understand due to poor variable/method names, poor solution structure, etc.
* UI is a mess. Things should be lined up and appropriately spaced to “look nice.” Xcode gives you those dashed blue guidelines so there should be no excuse for things not being lined up, etc. Get in the habit of building aesthetically balanced UIs from the start of this course.
* Assignment was turned in late.

A common question is, “How much commenting of my code do I need to do?” The answer is that your code must be easily and completely understandable by anyone reading it. You can assume that the reader knows iOS, but you should not assume that they already know *the* (or *a*) solution to the problem.

**Need More of a Challenge?**

Design a mechanism to require the user to authenticate (first time? every time?) in order to use the app. Or integrate the app with other social networking features of iOS (Twitter, Facebook, LinkedIn, …). Or figure out how to save the image the user selects when editing their profile.