**Final Milestone - Friends Networking App**

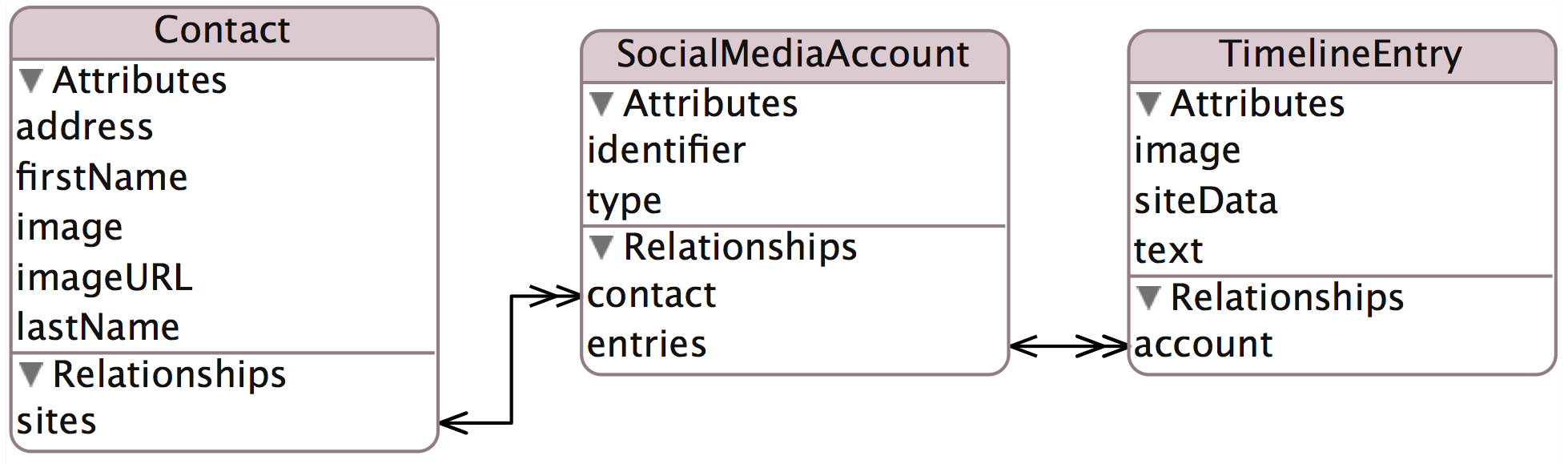
***Xcode Project "Friends"***

Create a new Xcode project consisting of a Swift iOS Application called **Friends**. When you set up the project, make sure the Company Identifier is set to **au.edu.griffith.ict.prog** and Devices is set to either **Universal**. After clicking **Next**, select the **Desktop** folder to create your project in and make sure you select **Create local git repository for this project**, before clicking **Create**.

***GIT Repository***

Make sure you commit to your git repository at least once for every meaningful piece you add to (or modify in) your app.  E.g., commit after you have created your model, before and after the unit tests, and after every view controller you have added.  Also commit after each modification (remember the rule of thumb: whenever you have successfully compiled your program, you should commit!).

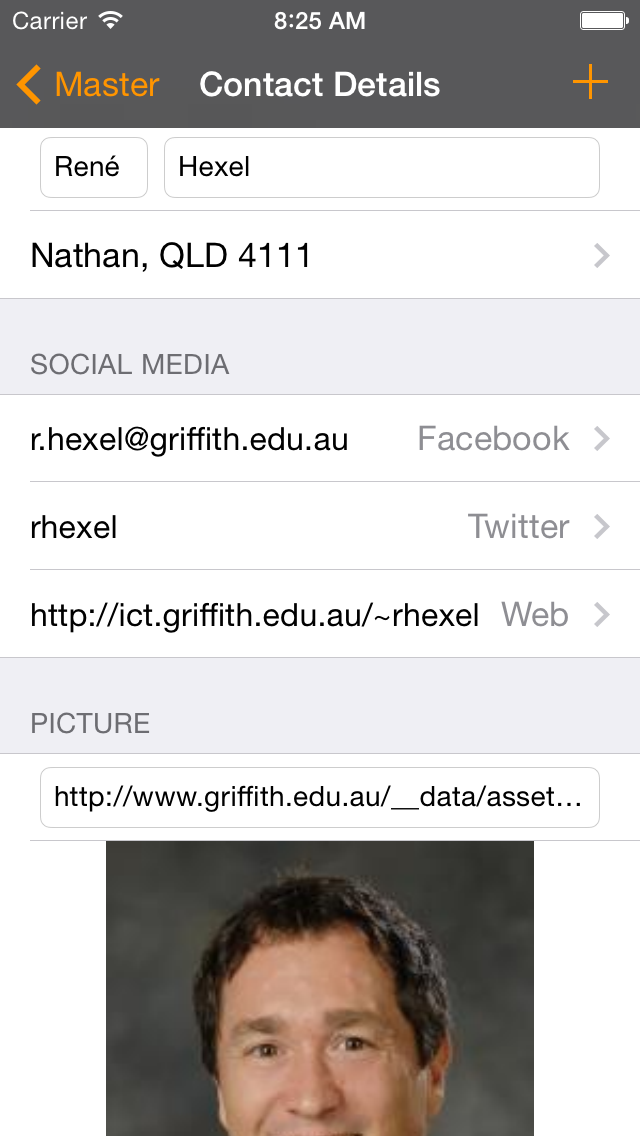
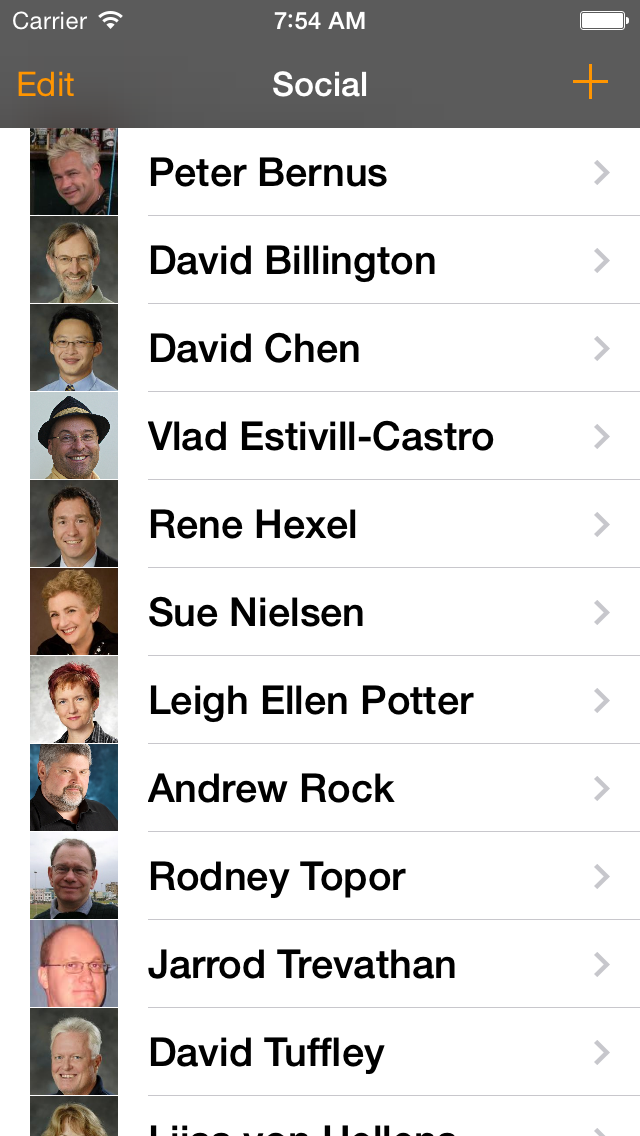
***Model and Unit Testing***

The model for this application needs to contain all the data required to handle the functionality described below.  All data need to be **persistent** (i.e. loaded and saved appropriately), either by using NSPropertylistSerialization, Core Data,or NSJSONSerialization, and be key/value coding compliant (use **key/value observation** (KVO) in your view controllers below).  You can use files from your own earlier milestones for both assignments (but don't forget to copy them into your new project folder) and modify them accordingly.  Here is a recommendation for what your data model should look like (in this example a Contact is what would go into a contacts array or ContactList model class):

Create **unit tests** for your data model and the social networking APIs that you use (see below).  Create your tests before your view controllers (and commit) to make sure everything will work in your view controllers!

***Contacts Master View and Detail View***

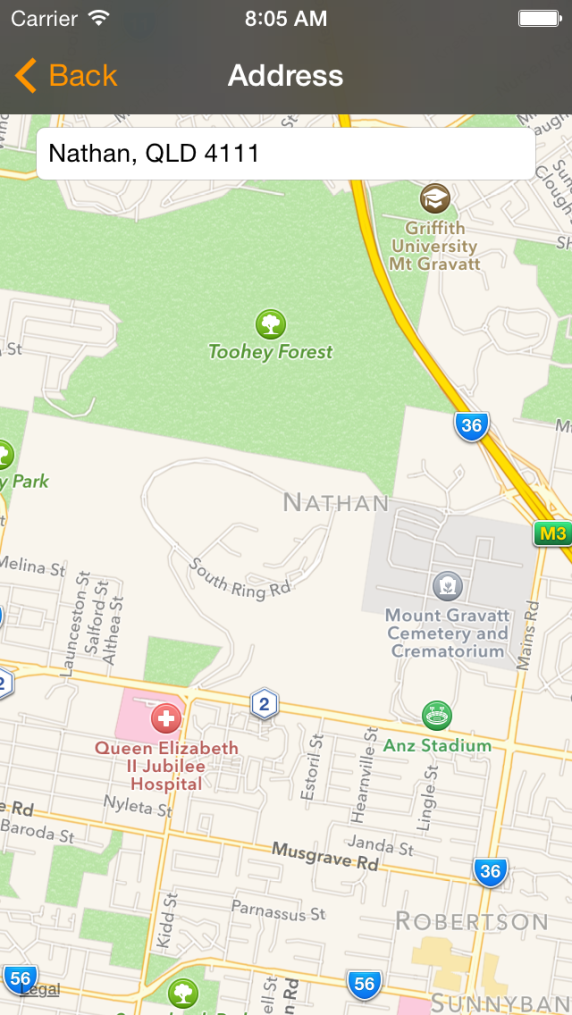
Edit your main story board to have a **master** view that contains a list of friends showing each person's first and last name, and a thumbnail image.  At the top of the *master view*, there should be a [+] button that takes the user to the detail view to **add** a new friend and an [Edit] button that allows removing a friend.  Tapping on an existing friend in the table should also take the user to the friend's detail view.  The **detail** view needs to show and allow to edit at least the following details: first name, last name, address, social networking info, picture URL (anywhere on the web, not necessarily on Flickr).  Make sure you can add social networking information (e.g. through an add button in the top right).  Your app should provide for at least *two different* social networks (e.g.,[Facebook](https://developers.facebook.com/docs/ios/graph), [Flickr](http://www.flickr.com/services/api/), [Twitter](https://dev.twitter.com/docs/api), [Yammer](https://developer.yammer.com/docs), and [Weibo](http://open.weibo.com/wiki/API%E6%96%87%E6%A1%A3/en), a user web page, or a manually maintained list of photos, their full-screen views and their details as per the week 8 milestone).  Here is an image of what your master and detail view User Interfaces could look like:



Make sure you link up the necessary actions and outlets!  Tapping some of the detail view cells (labels or arrows) should lead to further views below.

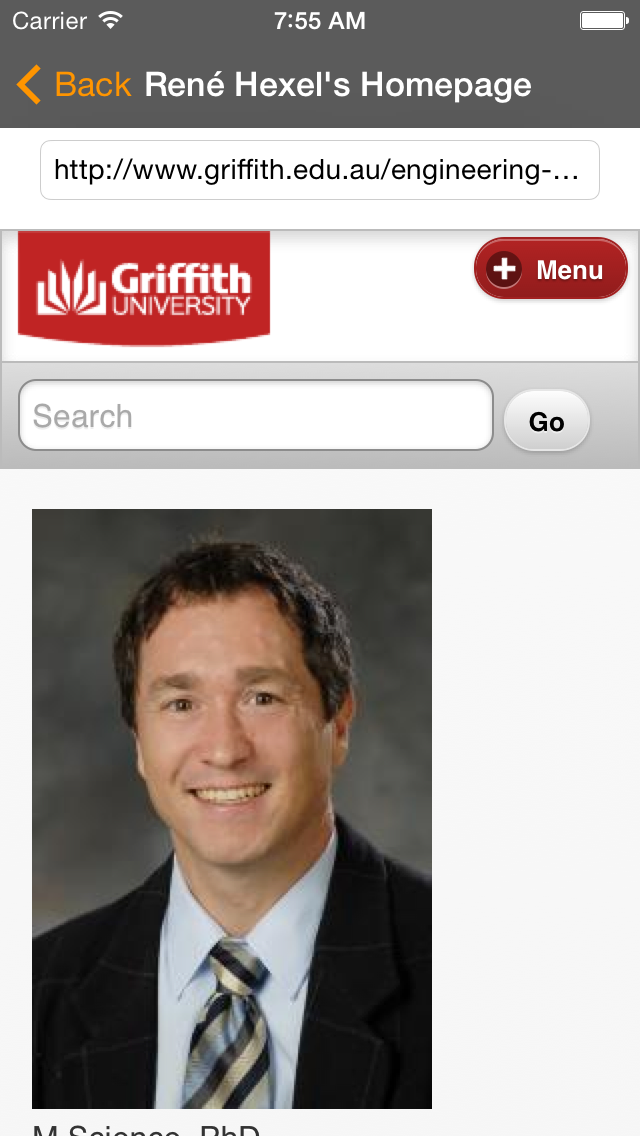
***Address: Map View***

Tapping on the Address cell of the Detail View should trigger a segue into another view that displays the address in a map view (with an editable text field), e.g.:



***Home Page: Web View***

Tapping on the Web Page cell in the Friends Media section of the Detail View should trigger a segue into another view that displays the home page in a web view:



***Friends Media Timeline***

Tapping on any social media cell of the Detail View (other than the web page above) should trigger a segue into another view that displays the user's social media stream (e.g. the public photo stream for Flickr, user updates or time line for Facebook, tweets for Twitter, etc.).  Make sure all information is downloaded in the background without blocking the user interface on the main thread!  Thumbnail photos (if available) and the title should be displayed in a table or collection view.  Tapping on an entry should segue into a view controller displaying the full details, e.g. for Flickr:



Read the [iOS Friends Media framework documentation](https://developer.apple.com/library/ios/documentation/Social/Reference/Social_Framework/_index.html#//apple_ref/doc/uid/TP40012233) for details on how to access social networking sites.  You will also need to read the API information on the social networking for details on how to access information from within your app (e.g., here are the links to [Flickr](http://www.flickr.com/services/api/), [Facebook](https://developers.facebook.com/docs/ios/graph), [Twitter](https://dev.twitter.com/docs/api), [Yammer](https://developer.yammer.com/docs), and [Weibo](http://open.weibo.com/wiki/API%E6%96%87%E6%A1%A3/en)).  Keep in mind that *not* all social networking sites are integrated into the operating system's Friends.framework.  In this case, you have to access the web site yourself through URL requests (make sure you encapsulate that in a separate class).  I have attached a FlickrAPI example (see Assignment Files (either below or at the top, depending on which BBLearn page you are viewing this from)).  Also, for most social networking sites, you need to acquire an API key that you have to use in your App in order to access the site!

***Code Quality and Commenting***

Use best practices when writing your code.  E.g., use unit tests, subclassing, delegation, and other object-oriented design patterns as appropriate.  Make sure all source files (.swift files) have a comment header that contains your name as the author and your student ID.  Put SwiftDoc style comments in front of your classes, properties, and methods, describing what each of these do!  For methods, make sure you document parameters and return types as well as pre- and post-conditions!  Make sure your code is well formatted and readable, and uses consistent indentation!

***Submission***

Zip up the folder that contains your project (e.g. if your project is called Friends, you should find a Friends folder on the Desktop). On the Mac, you can easily create a ZIP file of a folder by holding the Control key, clicking on the folder you want to zip up, and then select Compress "Friends" to compress).  Submit the ZIP file to BBlearn (click on the link that reads Milestone 6 at the top of this milestone page, then attach your ZIP file, and submit).