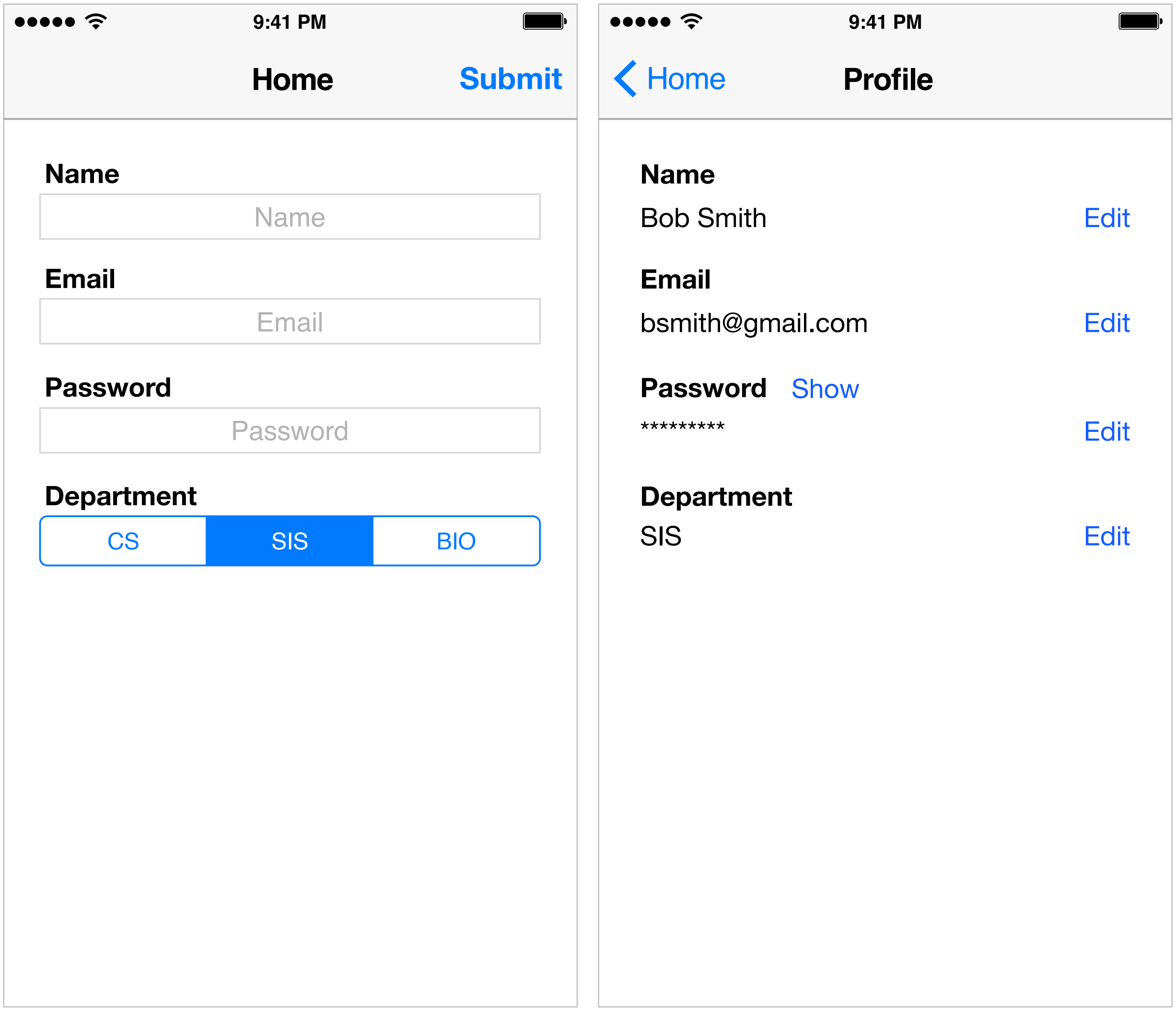
ITIS/ITCS 6010 – Advanced Topics for Mobile App Development
In Class Assignment 4

Basic Instructions:

1. In every file submitted you MUST place the following comments:
    a. Assignment #.
    b. File Name.
    c. Student Full Name.
2. This is an individual assignment, each student is expected to work alone and submit their own work.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Create a zip file which includes all the project folder, any required libraries, and your presentation material.
6. Submission details:
    a. You should submit the assignment through canvas: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 4 (100 Points)

In this assignment you will build an iOS application that uses navigation controllers and investigates communication and data passing between the different view controllers.



Name
Bob Smith

Email
bsmith@gmail.com

Password
****

Department
SIS

(a) Home View Controller **(b) Profile View Controller**

Figure 1: Application Wireframe

Part 1 : Home View Controller (20 Points)
The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

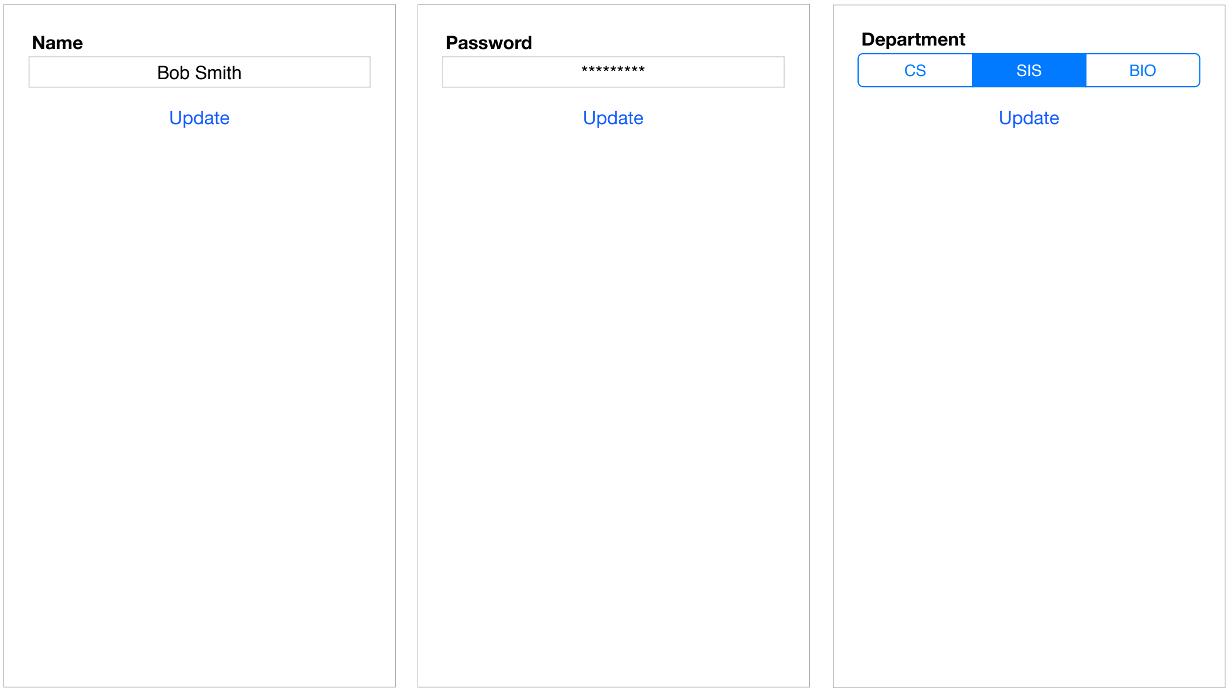
1. The home view controller should be embedded in a navigation controller to provide the transition to the Profile View Controller.
2. Upon clicking “Submit”
   a. If any of the user missed any of the profile fields display an alert dialog indicating the missing field.

b. If all the fields are provided correctly, then store the provided data in a User Object. Then pass the User object to the Profile View Controller and then display the Profile View Controller.

Part 2 : Profile View Controller (40 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. Upon loading the Profile View Controller, it should display the content of the User object as shown in Figure 1(b).
2. Note that the Password Label is displayed as “****”, the number of stars should match the number of characters in the password field.
3. If the “Show” button is clicked the Password Label should show the actual password field. Then the “Show” button text should be changed to “Hide”, upon clicking “Hide” the password field should be hidden using the “*****” as done when loading the View Controller, and then the “Hide” button text should be changed to “Show”.
4. Clicking the “Edit” button beside each field should open the corresponding Edit View Controller using a “Present Modally” Segue.



(a) Edit Name **(b) Edit Password** **(c) Edit Name**

Figure 2: Application Wireframe

Part 3 : Edit View Controllers (40 Points)

The interface should be created to match the UI presented in Figure 2. Feel free to create multiple view controllers to provide the different functionality required for this part. The requirements are as follows:

1. The view controller should populate the Text View with the passed current value of the parameter to be changed. For example, if the user name is “Bob Smith” the view controller should show “Bob Smith” which can be edited by the user. When the user clicks the “Update” button, the view controller should pass the updated value to the Profile View Controller, and should dismiss the Edit View Controller.
2. Note that the Edit View controller is displayed using “Present Modally”, you should consider using “Unwind Segues” to pass data back to the Profile View Controller.
3. Upon returning to the Profile View Controller, the updated value should be updated and displayed in the Profile View Controller.