CS 329E - Spring 2018 - Homework 4

Due date: **2/20/18 by 11:59pm**, late work is not accepted
Points: 20 points
Submit: A zip file of your entire project folder.
Name your zip file: **<last-name><first-name><dash>hw4.zip**
Example: for Joe Smith SmithJoe-hw4.zip

Description: This will be an exercise in creating an iOS application which includes:
- Navigation controller
- Table View controller
- Custom Table View Cells
- Alert Controller

General:
- Create a new Single-View iOS project named **<last-name><first-name><dash>hw4.**
- Alternate the table view cells between a **NameTableViewCell** and an **AddressTableViewCell**, starting with a NameTableViewCell.
- Make the height of the NameTableViewCell **50** and the height of the AddressTableViewCell **55**. Do not set this value in Interface Builder - instead override the appropriate method to set the cell height.
- Make the Navigation Bar title “People List”.
- Set appropriate constraints so that the UI looks good/correct when run on an **iPhone 8** or **iPhone 8** plus in portrait orientation.
- The **NameTableViewCell** button handler must be in the NameTableViewCell class.
- Since only view controllers can ‘present’, the logic to create the alert controller and present it must be in the view controller.

Details:
- Delete the initial view controller class and related controller in the storyboard.
  - The storyboard should be empty at this point.
- Create a class named **Person:**
  - Include the following public attributes:
    - name: firstName, type: String
    - name: lastName, type: String
    - name: age, type: Int
    - name: street, type: String
    - name: city, type: String
    - name: state, type: String
    - name: zip, type: Int
  - Include an **init** method with arguments for all the attributes, and initialize them with the values passed in.
- Create a data model with at least **9** Person objects, which will be used to populate the table view cells. See the data model data below for the values to set in each of the Person objects. Create a Table View Controller derived class named **ContactTableViewController** and associated storyboard table view controller, and tie them together.
- Create 2 Table View Cell derived classes named **NameTableViewCell** and **AddressTableViewCell** and associated storyboard table view cells, and tie them together.
  - Populate the table view cells with UI elements per the screen shots below.

- Make sure to create the necessary connections (outlet/action) for each UI element.
- Create a button handler for the button in the NameTableViewCell that displays an alert view per the screen shot below. It should contain the first name, last name and age for the cell in which the button exists.

- Add a navigation controller to the storyboard, making sure it is the initial view controller.

Grading criteria:

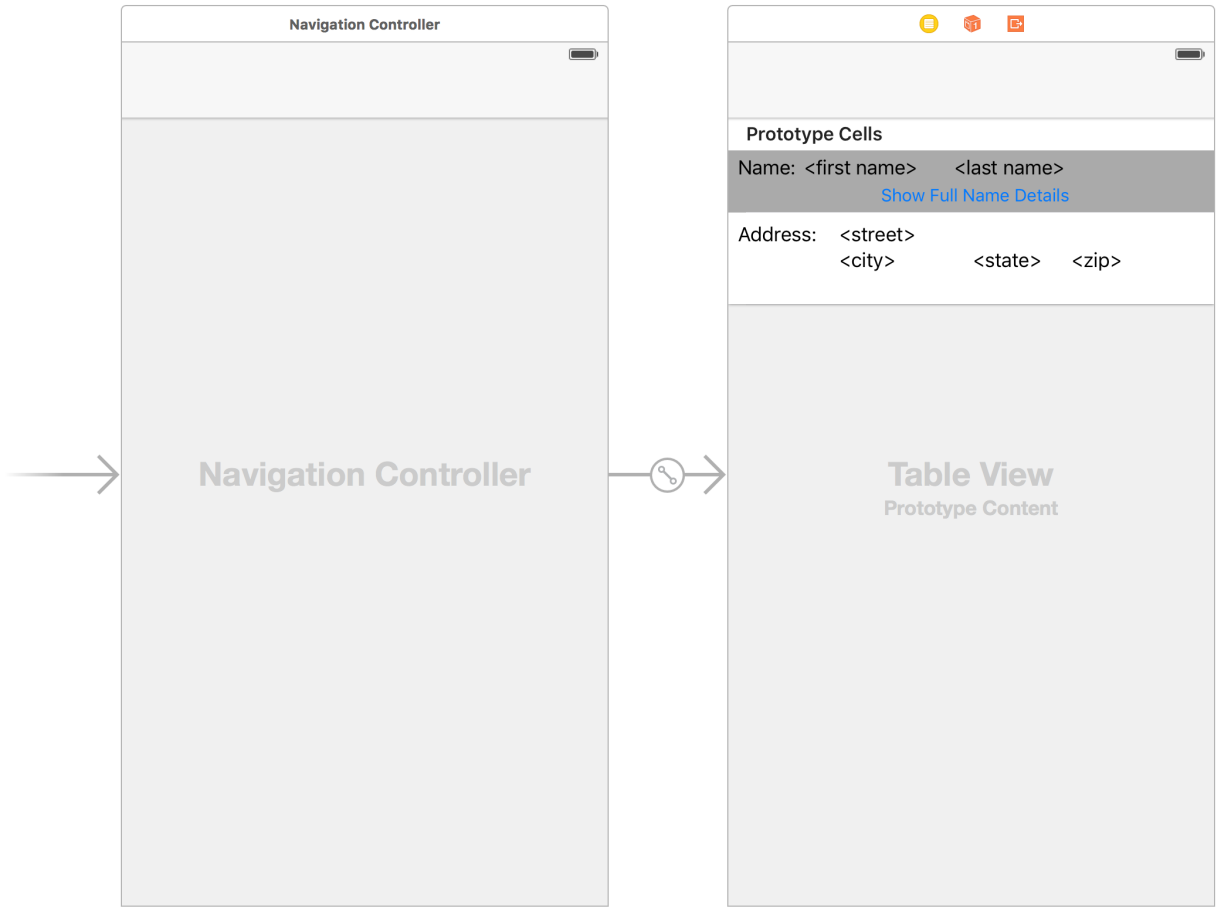
1. The application builds and runs. (1 point)
2. The data model is as defined. (3 points)
3. The user interface is as defined. (6 points)
4. The user interface displays correctly on both **iPhone 8** and **iPhone 8 plus**. (4 points)
5. The application behaves as described. (6 points)
6. The coding standard is followed. (1 point deducted for each kind of violation)

Data Model data:

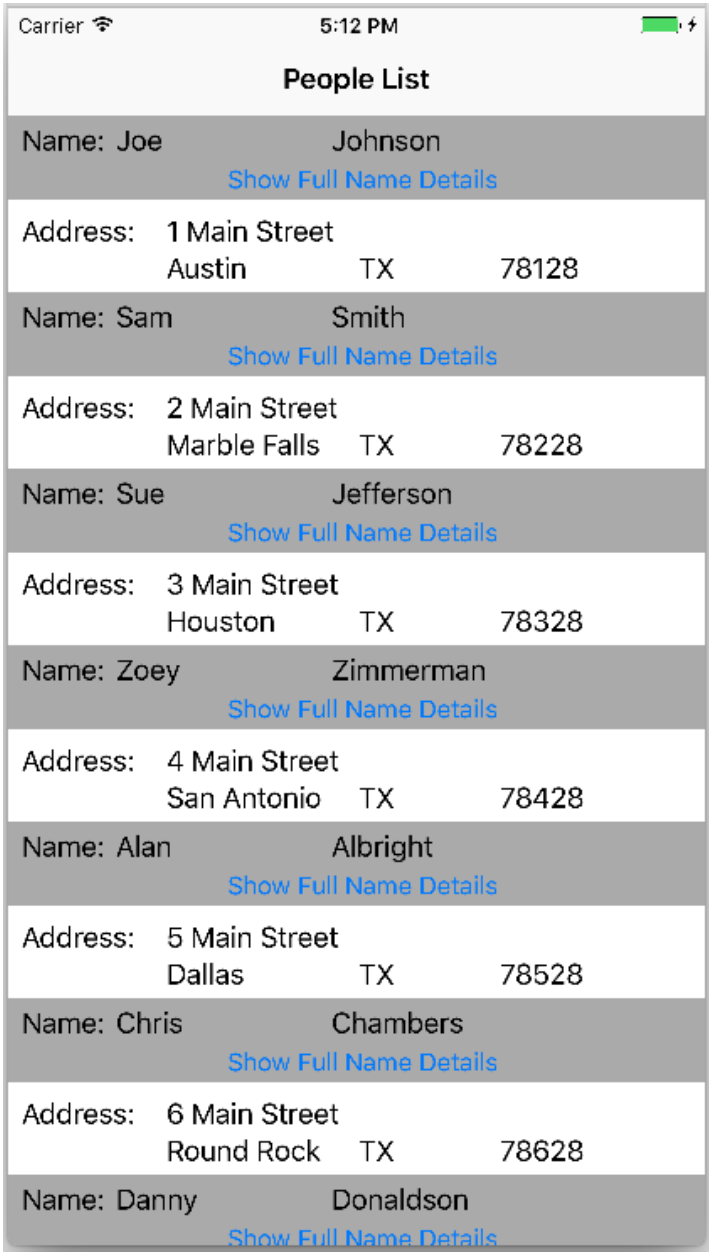
First Name	Last Name	Age	Street	City	State	Zip
Joe	Johson	35	1 Main Street	Austin	TX	78128
Sam	Smith	27	2 Main Street	Marble Falls	TX	78228
Sue	Jefferson	52	3 Main Street	Houston	TX	78328
Zoey	Zimmerman	17	4 Main Street	San Antonio	TX	78428
Alan	Albright	83	5 Main Street	Dallas	TX	78528
Chris	Chambers	33	6 Main Street	Round Rock	TX	78628
Danny	Donaldson	6	7 Main Street	Cedar Park	TX	78728
Eli	Edgerton	10	8 Main Street	Leander	TX	78828
Frank	Farmer	100	9 Main Street	Webster	TX	78928

This is what your storyboard should look like:

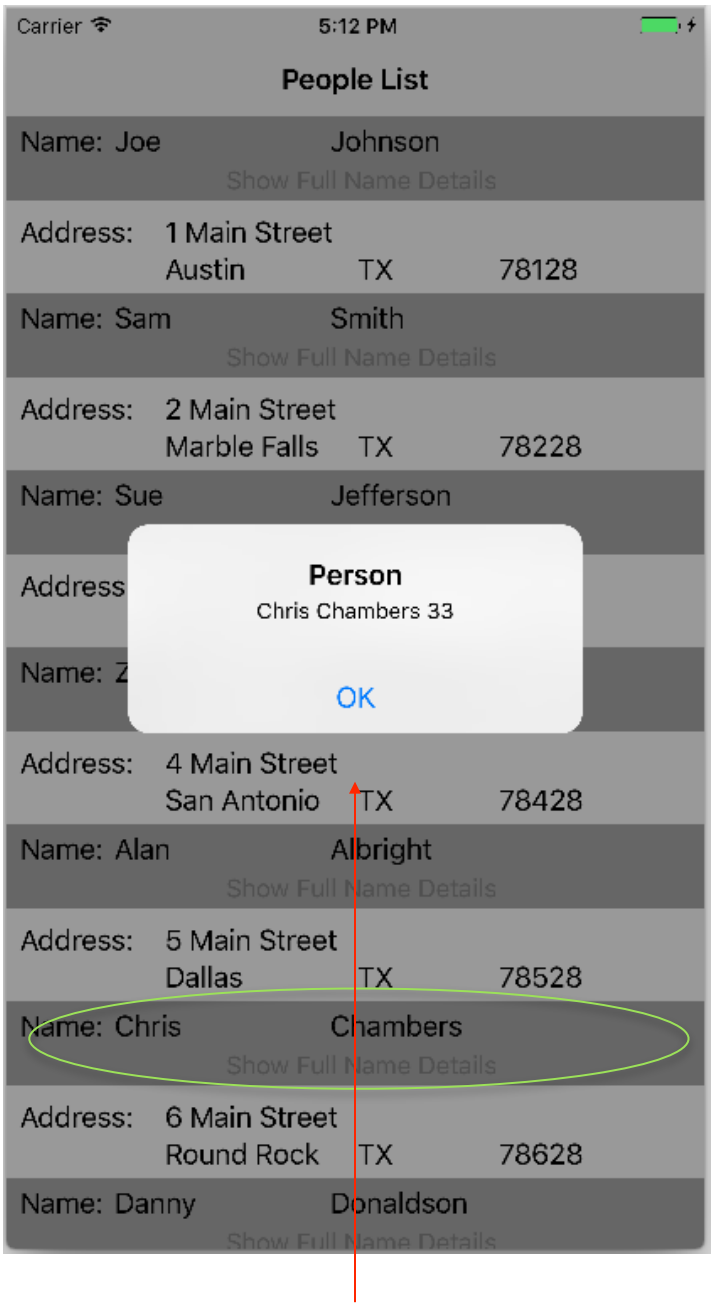
- Notice the two custom table view cells



When the application first starts - the first screen (the table view) should look like this - a total of 18 cells. A pair of cells per Person object in the data model. The first cell of the pair containing a Person's first name and last name and a 'Show Full Name Details' button. The second cell of the pair containing the related address data.



When the user touches the ‘Show Full Name Details’ button in one of the ‘name’ table cells, an alert view should display with the layout shown below (first name, last name, age):



For example, this is what should show when touching the ‘Show Full Name Details’ button in the table cell for Chris Chambers.