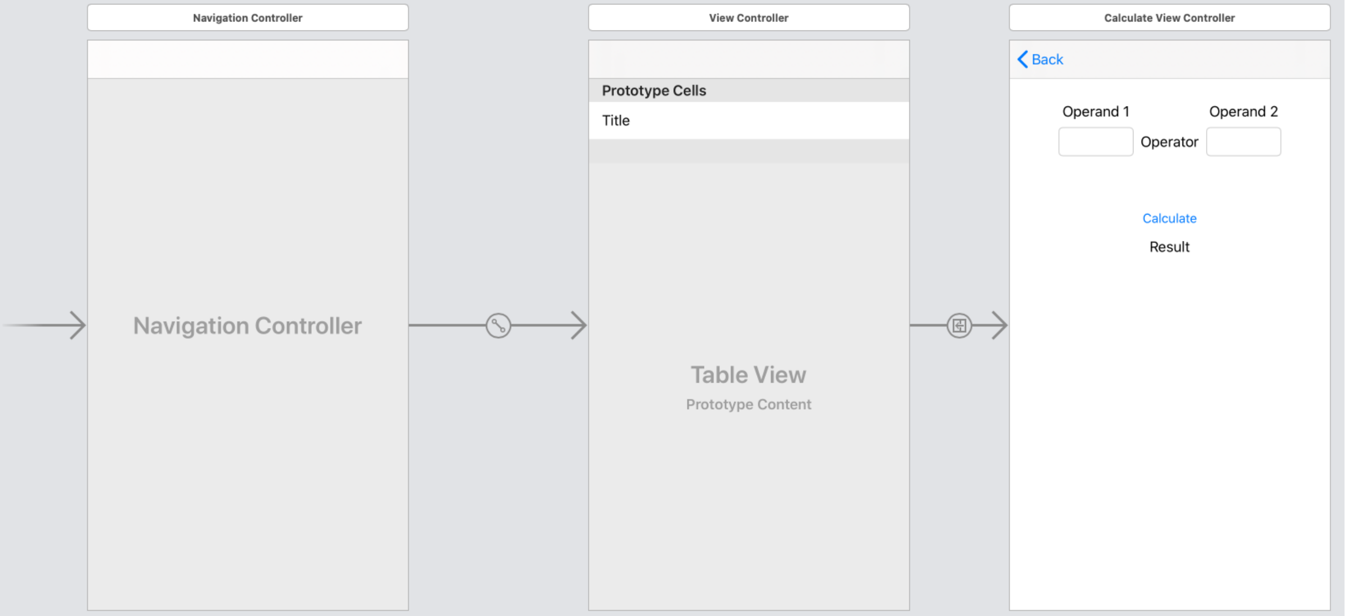
CS 329E: Bulko
Programming Assignment 4
Table and Navigation View Controllers

Your goal in this assignment is to create a rudimentary calculator using multiple view controllers. The app that you create will include a Navigation VC, a Table VC, and a plain VC.



1 Detailed instructions:

- Create a Single View application project named <lastName><firstName>-HW4.
- Add a Table View to the default View Controller. Pin the edges so that it occupies the entire screen.
- Embed the default VC in a Navigation Controller.
- Set the number of Prototype Cells to 1, and the Table View cell style to `Basic`.
- Add the `UITableViewDataSource` and `UITableViewDelegate` protocols to the `ViewController` class.
- Create a "data source": an array containing the four strings `"Add"`, `"Subtract"`, `"Multiply"`, and `"Divide"`.
- Implement the two methods required for a `UITableViewDataSource`.

- Create a generic ViewController containing seven elements:
  1. two text fields with labels “Operand 1” and “Operand 2”.
  2. a label in between that displays the operator chosen from the table, converted into the appropriate symbol `"+"`, `"-"`, `"*"`, or `"/"`.
  3. a button named “Calculate”.
  4. a label to display the result of the calculation.
- Create a segue by ctrl-dragging from the prototype text cell to the new VC.
- Create a custom class (in a new file) to handle things in the new VC.
- Add code to the new class for `ViewWillAppear`, which handles what happens when the VC is displayed. This should include displaying the operator chosen from the table.
- Once the user enters two operands and clicks the button, calculate the result of the operation and display it in the result label. Users can type any combination of integers or floats into the text fields. If both are integers, the result should also be an integer; otherwise, the result should be a float.
- Add code to the original VC for `prepare(for segue:)`.

2 Grading criteria

1. You have UI components and a data model as defined. (20%)
2. The Table View screen looks like the one pictured. (30%)
3. The calculation screen looks like the one pictured. (30%)
4. The application behaves as expected. (20%)
5. **If the app does not build and run, ZERO points will be given.**
6. The Coding Standard is followed. One point deducted for each violation.

3 General criteria

1. I will be looking for good documentation, descriptive variable names, clean logical structure, and adherence to all coding conventions expected of an experienced programmer, as well as those outlined in the Coding Standard document. There will be penalties for failure to meet these standards.
2. Your code must compile and run before submission.
3. Xcode will automatically generate standard headers to your .swift files. Add two lines to each Swift file so that the header includes the following:

   // Project: LastnameFirstname-HW4
   // EID: xxxxxx
   // Course: CS329E