Homework 4

The purpose of this assignment is to show your understanding of data sources and delegates, along with their applications in UIKit: picker views, table views, and collection views.

In order to successfully complete this assignment, you must successfully implement all of the Functional Requirements below. As with all apps that you submit for this class, you should also:

- Ensure that your app does not crash or behave in an unstable manner
- Make use of good design patterns (e.g. MVC) and structure your app accordingly
- Follow the Human Interface Guidelines when designing your app's UI
- Test your app on an actual device running the latest version of iOS

Functional Requirements

The app should consist of two view controllers that the user can switch between with a tab bar. Each view controller should have a title bar with the title "Birthdays." Both view controllers should show a different view of the same data: a series of names and corresponding birthdays. The first should use a UITableView and the second should use a UICollectionView.

Table View Controller

The table view in the first controller should have a single section with one row per birthday. Each cell should place the name on the left in bold black text and the birthday on the right in a lighter detail font. When displaying the birthday, only display the month, day, and year; do not display the time. Entries should be sorted alphabetically by name.

In the title bar for the table view controller, display two buttons: a standard "Edit" button on the right and a standard "Add" (usually styled as "+") button on the left. The Edit button should exhibit all the standard system behavior: when pressing it, the table view should show a delete control in each row, and the Edit button should become a Done button. When pressing the Done button, the table view should revert to its normal appearance.

When the Add button is pressed, the table view should present a modal controller that contains three items: a "Name" label, a text field for entering a name, and a date picker for selecting a birthday. The modal controller should also have a title bar that contains only a "Done" button in the upper right; when pressed, the Done button should save the birthday and dismiss the modal controller to reveal the table view, complete with the new entry, sorted in its proper alphabetical position.

Collection View Controller

The collection view should display the same set of names and birthdays as the table view; any updates to the table view should be reflected in the collection view immediately. Each cell in the collection view should be 80pts tall and occupy half the collection view's width (in any orientation). There should be no space between any cells; instead, each cell should have a 1pt gray border. (This may make it appear that adjacent cells have 2pts of border between them; this is OK.)

In each cell, the name should appear centered above the birthday, also centered; again, the name should appear in a bold font, while the birthday should be in a lighter detail font. All the cells should appear alphabetically by name, just like the table view. The user should not be able to edit any data in the collection view.

Bonus Opportunities

You can receive 10 bonus points for each of the following additional features.

Custom Birthday Ordering

Instead of sorting birthdays alphabetically, default to adding new birthdays at the end of the list. When the table is in editing mode, show reordering controls on the right of each cell, and store the order so that both the table view and the collection view display birthdays in that order.

Multiple Sort Orders

On both the table and collection view controllers, show a segmented control in the top bar just below the title. The control should have two segments: "Name" and "Birthday." When the user taps a segment, change the sort order of the table or collection to match the selected segment. Preserve the selection across tabs. (If you also implement "Custom Birthday Ordering," add a third segment for the custom order.)

View Sections

Introduce sections into both the table and collection views. Provide one section for each letter in the alphabet. Only show a section if there is a name to go in that section. Show the letter as the label for its section. (If you also implement "Multiple Sort Orders," section the birthday sort based on month. If you also implement "Custom Birthday Ordering," you can provide only one section and cannot get this bonus without "Multiple Sort Orders.")

Submitting Your Work

To submit your work, upload a .zip file to the appropriate drop box that contains your entire Xcode project directory, including:

- Your .xcodeproj bundle and all its contents
- All your source files, including code, .xib files, and any image resources
- Any additional files that your app requires to run

Name this zip file "UW_HW4_*<UW NetID>*.zip", where *<UW NetID>* is the username assigned to you by UW. (For example, the instructor's submission would be named UW_HW4_tekl.zip.)

Your submission should compile cleanly on the first try, throwing absolutely no errors, warnings, or static analyzer problems. You may lose points if your solution does not compile cleanly.

Your submission should, once compiled, run well on an actual device running the latest version of iOS. You may choose to support past versions of iOS, but all testing will be done on the newest version available on the due date.