Homework: Camera and Photos

The purpose of this assignment is to demonstrate your understanding of camera system on iOS by interacting with a UIImagePickerController. In addition, you will show your comprehension of the Photos framework by accessing photos on the device and in iCloud.

Since there is no camera hardware in the iOS Simulator and UIImagePickerController is limited in the number of photo sources it can access, this assignment will likely require that you do the majority of your development on an actual iOS device.

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

For this assignment, you will create an app that offers users the ability to both take new photos using the device's camera and to select photos from the photo library. The app should use a tab bar controller to separate these two abilities; it should present both a tab labeled "Camera" and a tab labeled "Photos."

On the "Camera" tab, the app should initially show a blank screen with only a button at the bottom labeled "Take Photo." When the button is tapped, the app should present a standard camera interface to the user.

Upon taking a picture, the app should display the picture in the formerly blank space above the Take Photo button. Ensure that any photo taken maintains its aspect ratio when displayed, and that the photo is saved to the user's photo library. (If a camera is unavailable on the device, tapping the Take Photo button should do nothing.)

On the "Photos" tab, the app should display a navigation controller containing a table view of Smart Albums available on the device. Each cell in the table should show a disclosure indicator on the right, illustrating to the user that the cell can show additional details if tapped.

Once tapped, each cell should cause the navigation controller to navigate to a new screen; this screen should show all the photos in the selected Smart Album in a collection. Each photo should occupy exactly 50x50 points onscreen, scaled and cropped to fit while preserving the photo's aspect ratio. There should be at least 10pts spacing between each photo, as well as between each row of photos. If there are more photos than fit onscreen, the collection should scroll vertically to reveal each photo. All photos shown on this screen must be retrieved and rendered using the Photos framework.

Your app should notice any changes that occur in the device's photo library, regardless of the currently selected tab. When such a change occurs (including changes provoked from the Camera tab), the relevant view should update as quickly as possible to include the newly added, changed, or removed photos.

Bonus Opportunities

You can receive up to 15 points for including a "poster image" in each row in the Smart Albums list. This image should be left-aligned, preceding the album title in the row. Select the image to use at random from the album. Each poster image should be of a fixed width; scale and crop it such that it fills this width and maintains its aspect ratio. If the album is empty, use a solid light gray fill in the poster image space instead. Ensure that your poster images update if needed when the photo library changes.

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 "UWHWCameraAndPhotos_*<UW NetID>*.zip", where *<UW NetID>* is the username assigned to you by UW. (For example, the instructor's submission would be named UWHWCameraAndPhotos_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.