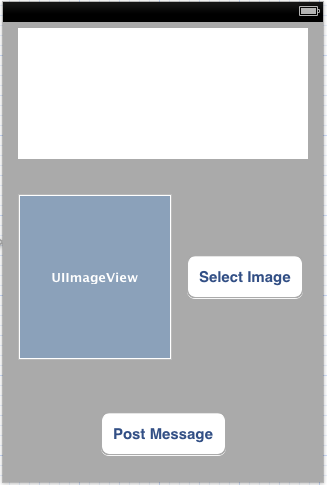
We will now use the UIActivityViewController class to spice up our social apps.

**Part I the setup:**

1. **Create another Single View App. If you want to be consistent with what I’ve done, call it SocialApp.**
2. **Design this interface with a Text View, Image View and two Buttons**

****

**3. You’ll need to add two frameworks: The Mobile Core Services and the Social Frameworks…(Remember Link Binaries)**

**Part II Create Actions and Outlets**

1. **Configure the textarea connection as an Outlet named postText**
2. **Create an outlet for the Image View object named postImage.**

**The application will require three actions. One for each of the button objects and one for the background view that will be used to hide the keyboard when the user has finished entering text.**

1. **Create an Action for Select Image button named selectImage.**
2. **Now add an action for the Post Message button, this time naming the action sendPost.**

**This is new: In order to assign an action to the View object, it will need to be changed so that it is a subclass of UIControl. Click on the background of the view, display the Identity Inspector (View -> Utilities -> Show Identity Inspector) and change the object’s class from UIView to UIControl.**

1. **Create an action connection named hideKeyboard for the Touch Down event.**
2. **Now select the SocialAppViewController.h file and further modify it to configure the class to act as an image picker delegate and to add some imports that will be required later in the tutorial:**

#import <UIKit/UIKit.h>

#import <Social/Social.h>

#import <MobileCoreServices/MobileCoreServices.h>

@interface SocialAppViewController : UIViewController

<UIImagePickerControllerDelegate, UINavigationControllerDelegate>

@property (strong, nonatomic) IBOutlet UITextView \*postText;

@property (strong, nonatomic) IBOutlet UIImageView \*postImage;

- (IBAction)selectImage:(id)sender;

- (IBAction)sendPost:(id)sender;

- (IBAction)hideKeyboard:(id)sender;

@end

**Part III: Implementing the selectImage and Delegate Methods**

**This will be your selectImage method:**

- (IBAction)selectImage:(id)sender {

if ([UIImagePickerController isSourceTypeAvailable:

UIImagePickerControllerSourceTypeSavedPhotosAlbum])

{

UIImagePickerController \*imagePicker =

[[UIImagePickerController alloc] init];

imagePicker.delegate = self;

imagePicker.sourceType =

UIImagePickerControllerSourceTypePhotoLibrary;

imagePicker.mediaTypes = [NSArray arrayWithObjects:

(NSString \*) kUTTypeImage,

nil];

imagePicker.allowsEditing = NO;

[self presentViewController:imagePicker animated:YES completion:nil];

}

}

**Now we need add the other image picker delegate methods so that the picker is dismissed when the user has made a selection:**

#pragma mark -

#pragma mark UIImagePickerControllerDelegate

-(void)imagePickerController:(UIImagePickerController \*)picker

didFinishPickingMediaWithInfo:(NSDictionary \*)info

{

NSString \*mediaType = [info

objectForKey:UIImagePickerControllerMediaType];

[self dismissViewControllerAnimated:YES completion:nil];

if ([mediaType isEqualToString:(NSString \*)kUTTypeImage]) {

UIImage \*image = [info

objectForKey:UIImagePickerControllerOriginalImage];

\_postImage.image = image;

}

}

**Dismiss Picker:**

-(void)imagePickerControllerDidCancel:(UIImagePickerController \*)picker

{

[self dismissViewControllerAnimated:YES completion:nil];

}

**Hide the Keyboard**

- (IBAction)hideKeyboard:(id)sender {

[\_postText resignFirstResponder];

}

**Post to Social Media**

**/\*The code simply creates an array of items to be included in the post (in this case the text entered by the user and an image in the event that one was selected), creates a UIActivityViewController instance initialized with that array and presents the controller to the user.\*/**

- (IBAction)sendPost:(id)sender {

NSArray \*activityItems;

if (\_postImage.image != nil) {

activityItems = @[\_postText.text, \_postImage.image];

} else {

activityItems = @[\_postText.text];

}

UIActivityViewController \*activityController =

[[UIActivityViewController alloc]

initWithActivityItems:activityItems

applicationActivities:nil];

[self presentViewController:activityController

animated:YES completion:nil];

}