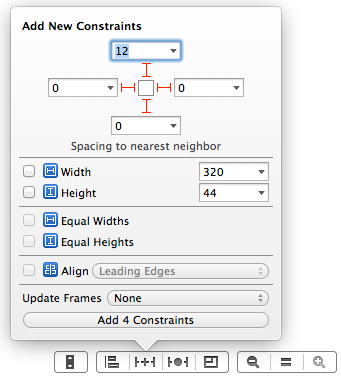
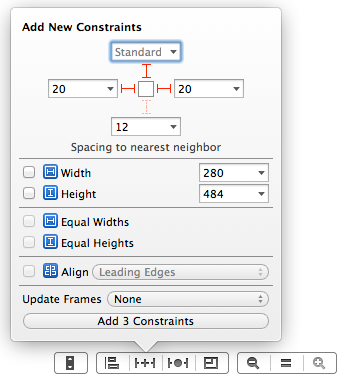
Camera for IOS 7

1. Single View Application
2. image view, a toolbar and two bar button items
3. Check out the flexible space bar button item
4. Create the following constraints on the tool bar:



1. Add the following constraints to the image view:



1. Using the size inspector, reduce the Vertical Content Compression Resistance Priority value to 100. This will prevent the image view from growing vertically and obscuring the toolbar when displaying large images.
2. Create an Outlet for the Image View. Call it whatever you’d like. I called mine imageView
3. Create Actions for your 2 toolbar buttons. I called mine useCamera and useCameraRoll

**Now for Coding**

9. Add a delegate to your h file. <UIImagePickerControllerDelegate,

UINavigationControllerDelegate>

10: Create a property for a BOOL datatype called newPic

@property BOOL newPic;

11. Add the Mobile Core Services framework to your library

**Your h file might look like this:**

#import <UIKit/UIKit.h>

#import <MobileCoreServices/MobileCoreServices.h>

@interface CameraViewController : UIViewController

<UIImagePickerControllerDelegate,

UINavigationControllerDelegate>

@property BOOL newPic;

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

- (IBAction)useCamera:(id)sender;

- (IBAction)useCameraRoll:(id)sender;

@end

12. Here is your useCamera method:

- (void) useCamera:(id)sender

{

if ([UIImagePickerController isSourceTypeAvailable:

UIImagePickerControllerSourceTypeCamera])

{

UIImagePickerController \*imagePicker =

[[UIImagePickerController alloc] init];

imagePicker.delegate = self;

imagePicker.sourceType =

UIImagePickerControllerSourceTypeCamera;

imagePicker.mediaTypes = @[(NSString \*) kUTTypeImage];

imagePicker.allowsEditing = NO;

[self presentViewController:imagePicker

animated:YES completion:nil];

\_newPic = YES;

}

}

13. Here is your useCameraRoll Method

- (void) useCameraRoll:(id)sender

{

if ([UIImagePickerController isSourceTypeAvailable:

UIImagePickerControllerSourceTypeSavedPhotosAlbum])

{

UIImagePickerController \*imagePicker =

[[UIImagePickerController alloc] init];

imagePicker.delegate = self;

imagePicker.sourceType =

UIImagePickerControllerSourceTypePhotoLibrary;

imagePicker.mediaTypes = @[(NSString \*) kUTTypeImage];

imagePicker.allowsEditing = NO;

[self presentViewController:imagePicker

animated:YES completion:nil];

\_newPic = NO;

}

}

14. Here are your delegate methods:

#pragma mark -

#pragma mark UIImagePickerControllerDelegate

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

didFinishPickingMediaWithInfo:(NSDictionary \*)info

{

NSString \*mediaType = info[UIImagePickerControllerMediaType];

[self dismissViewControllerAnimated:YES completion:nil];

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

UIImage \*image = info[UIImagePickerControllerOriginalImage];

\_imageView.image = image;

if (\_newPic)

UIImageWriteToSavedPhotosAlbum(image,

self,

@selector(image:finishedSavingWithError:contextInfo:),

nil);

}

else if ([mediaType isEqualToString:(NSString \*)kUTTypeMovie])

{

// Code here to support video if enabled

}

}

-(void)image:(UIImage \*)image

finishedSavingWithError:(NSError \*)error

contextInfo:(void \*)contextInfo

{

if (error) {

UIAlertView \*alert = [[UIAlertView alloc]

initWithTitle: @"Save failed"

message: @"Failed to save image"

delegate: nil

cancelButtonTitle:@"OK"

otherButtonTitles:nil];

[alert show];

}

}

15. If the user cancels a session:

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

{

[self dismissViewControllerAnimated:YES completion:nil];

}