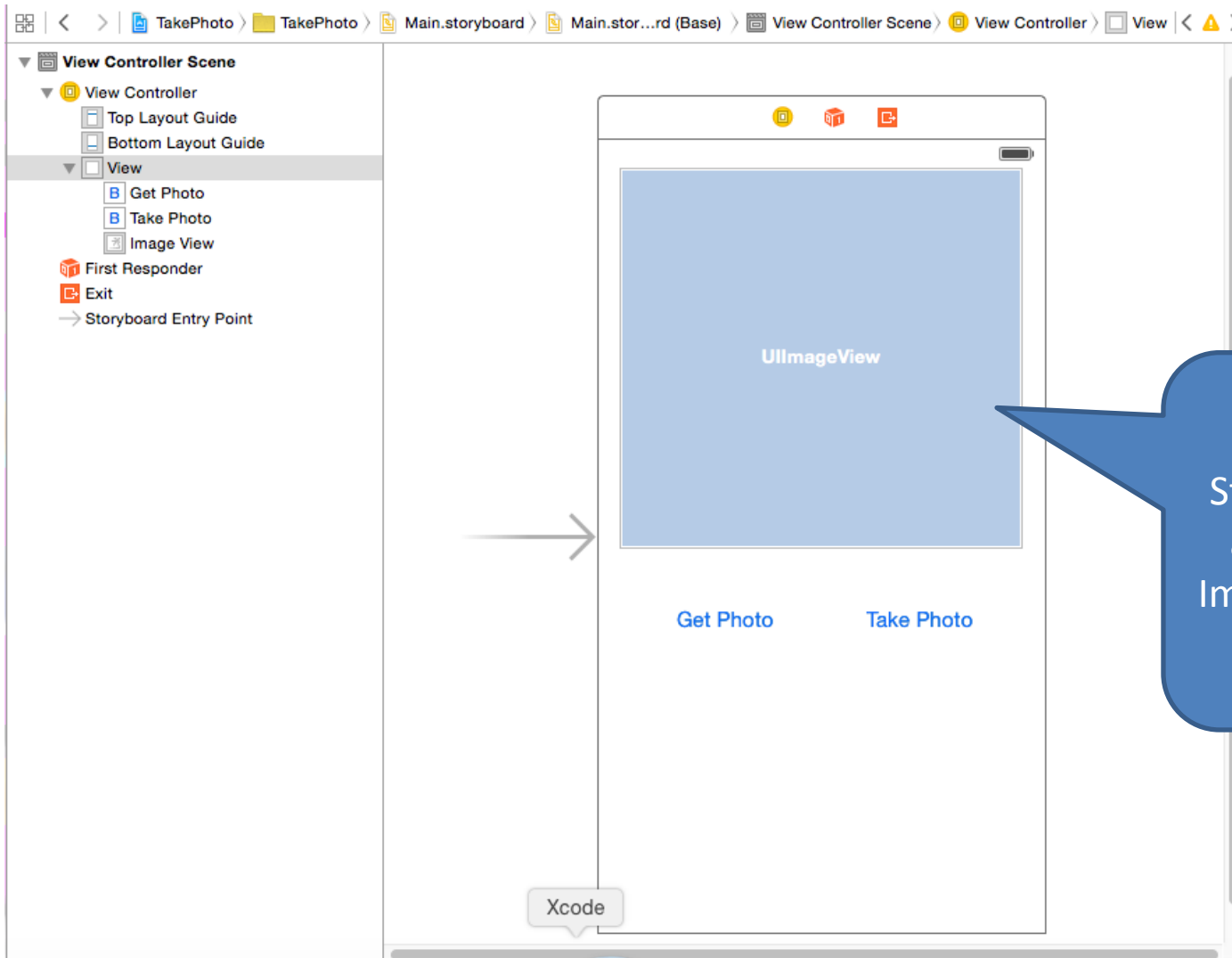


iPhone Apps Development using Objective C & Xcode (Lesson 9)

By Dannis Mok

Take Photo



Take Photo

```
TakePhoto > TakePhoto > ViewController.h > @interface ViewController

1 //
2 //  ViewController.h
3 //  TakePhoto
4 //
5 //  Created by Dannis Mok on 6/10/15.
6 //  Copyright (c) 2015 com.uec. All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10
11 @interface ViewController : UIViewController <UIImagePickerControllerDelegate,
    UINavigationControllerDelegate>
12
13
14 @property (weak, nonatomic) IBOutlet UIImageView *myImageView;
15
16
17 - (IBAction)getPhoto:(id)sender;
18
19
20 - (IBAction)takePhoto:(id)sender;
21
22
23 @end
24
25
```

Add the delegate
declarations

Make the necessary
outlets and actions

Take Photo

```
- (IBAction)getPhoto:(id)sender {
    UIImagePickerController *picker = [[UIImagePickerController alloc] init];
    picker.delegate = self;
    picker.allowsEditing = YES;
    picker.sourceType = UIImagePickerControllerSourceTypeSavedPhotosAlbum;
    [self presentViewController:picker animated:YES completion:nil];
}

- (IBAction)takePhoto:(id)sender {
    if([UIImagePickerController isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {
        UIImagePickerController *picker = [[UIImagePickerController alloc] init];
        picker.delegate = self;
        picker.allowsEditing = YES; // Must added for image showup
        picker.sourceType = UIImagePickerControllerSourceTypeCamera;
        [self presentViewController:picker animated:YES completion:nil];
    }
}
```

Get the photos by
using the
ImagePickerView
Controller

Take the photo and
then dismiss the
ImagePickerView
Controller

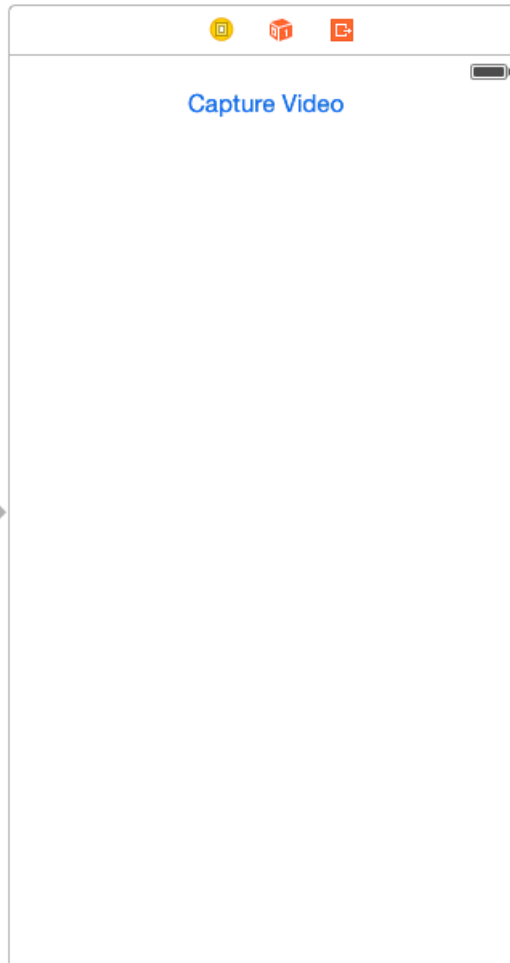
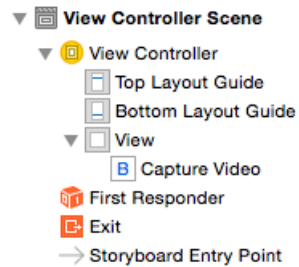
Take Photo

```
-(void)imagePickerControllerDidCancel:(UIImagePickerController *)picker {  
    [self dismissViewControllerAnimated:YES completion:nil];  
}  
  
-(void)imagePickerController:(UIImagePickerController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info {  
  
    UIImage *myimage = info[UIImagePickerControllerEditedImage];  
    _myImageView.image = myimage;  
    [self dismissViewControllerAnimated:YES completion:nil];  
}
```

These delegation methods will run when the user either the Cancel or Use Button in the ImagePicker

Show the photo in the ImageView

Record Video



Create a
Storyboard by
adding one
Button

Record Video

```
1 //  
2 // ViewController.h  
3 // TakeVideo  
4 //  
5 // Created by Dannis Mok on 6/10/15.  
6 // Copyright (c) 2015 com.uec. All rights reserved.  
7 //  
8  
9 #import <UIKit/UIKit.h>  
10 #import <MediaPlayer/MediaPlayer.h>  
11 #import <MobileCoreServices/MobileCoreServices.h>  
12  
13 @interface ViewController : UIViewController<UINavigationControllerDelegate,  
14     UIImagePickerControllerDelegate>  
15  
16 @property(strong, nonatomic) NSURL *videoURL;  
17 @property(strong, nonatomic) MPMoviePlayerController *videocontroller;  
18  
19  
20 @end  
21  
22 |
```

Still use the
UIImagePickerController to
record the video and so need
to add the delegation
declaration

Use the
MoviePlayerController to
control the playback of the
video

Record Video

```
//  
// ViewController.m  
// TakeVideo  
//  
// Created by Dannis Mok on 6/10/15.  
// Copyright (c) 2015 com.uec. All rights reserved.  
//  
  
#import "ViewController.h"  
  
@interface ViewController ()  
  
- (IBAction)captureVideo:(id)sender;  
  
@end  
  
@implementation ViewController  
  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
}  
  
- (void)didReceiveMemoryWarning {  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.  
}
```



Add the action
for the button

Record Video

```
- (IBAction)captureVideo:(id)sender {  
    if([UIImagePickerController isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera])  
    {  
        UIImagePickerController *picker = [[UIImagePickerController alloc] init];  
        NSArray *media = @[(NSString *)kUTTypeMovie];  
  
        picker.delegate = self;  
        picker.allowsEditing = YES;  
        picker.sourceType = UIImagePickerControllerSourceTypeCamera;  
        picker.mediaTypes = media;  
        [self presentViewController:picker animated:YES completion:nil];  
    }  
}
```

This array is used to inform the UIImagePickerController to take video rather than taking photo

Record Video

```
-(void)imagePickerController:(UIImagePickerController *)picker didFinishPickingMediaWithInfo:
(NSDictionary *)info {
    _videoURL = info[UIImagePickerControllerMediaURL];
    [picker dismissViewControllerAnimated:YES completion:nil];
    _videocontroller = [[MPMoviePlayerController alloc] init];
    [_videocontroller setContentURL:_videoURL];
    [_videocontroller.view setFrame:CGRectMake(0, 150, 320, 460)];
    [self.view addSubview:_videocontroller.view];
    [_videocontroller play];
}

-(void)imagePickerControllerDidCancel:(UIImagePickerController *)picker {
    [picker dismissViewControllerAnimated:YES completion:nil];
}
```

After taking the video, a VideoController is created and then add the video into its view to playback

Image Assets

Create a new image assets for storing different versions of image files under a common name

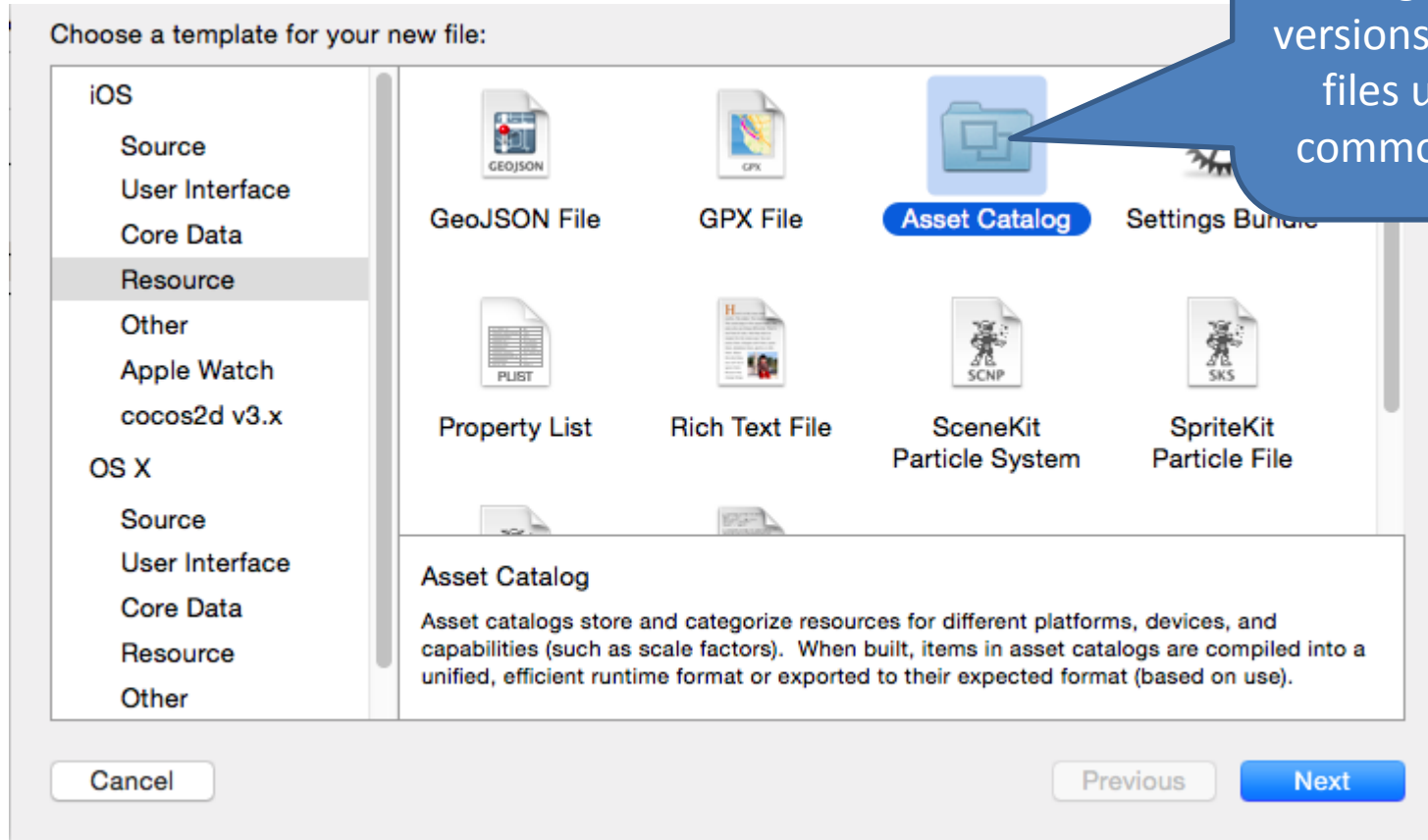


Image Assets and Icons

The screenshot shows the 'Save As' dialog in Xcode. The 'Save As:' text field contains the word 'flower'. Below it is an empty 'Tags:' text field. The 'Where:' section shows a folder icon and the text 'CollectionViewTest'. Below this is a 'Group' section with a folder icon and the text 'CollectionViewTest'. The 'Targets' section is a list box containing two items: 'CollectionViewTest' (checked with a blue checkmark and an icon) and 'CollectionViewTestTests' (unchecked with an empty checkbox and an icon). At the bottom right are 'Cancel' and 'Create' buttons.

Save As: flower

Tags:

Where: CollectionViewTest

Group CollectionViewTest

Targets

- ☒ CollectionViewTest
- ☐ CollectionViewTestTests

Cancel Create

Name the
asset file

Image Assets and Icons

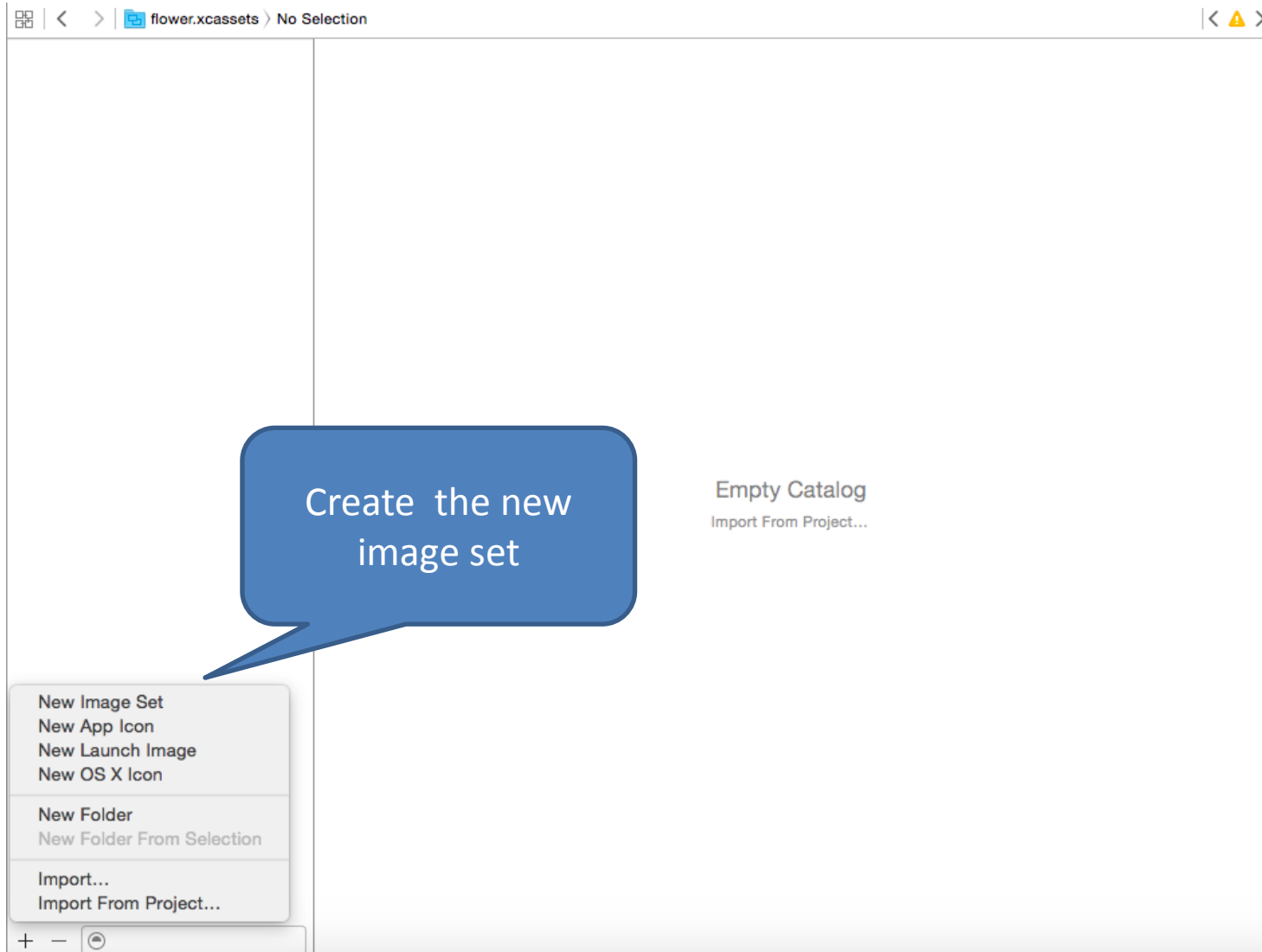


Image Assets and Icons

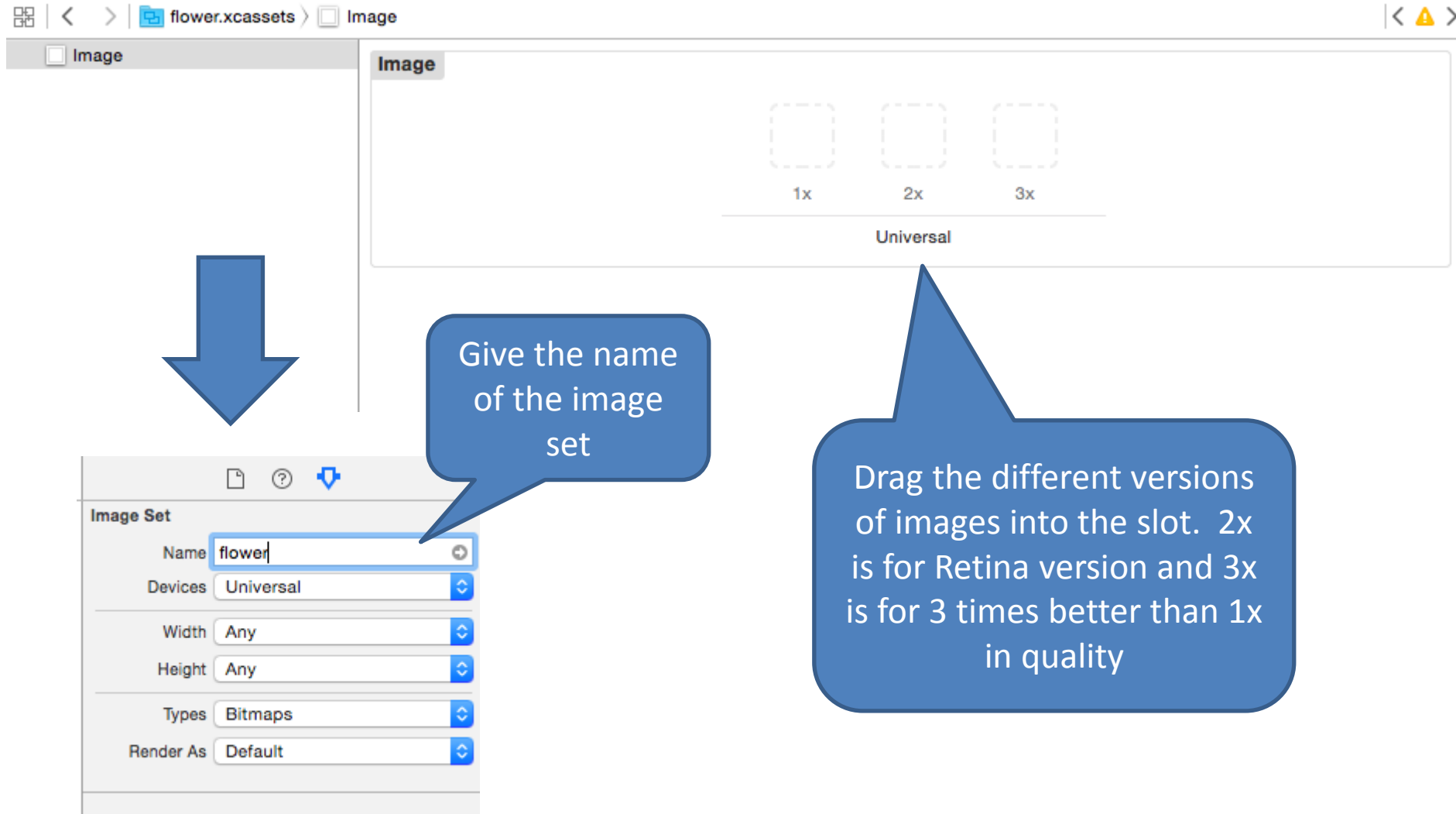
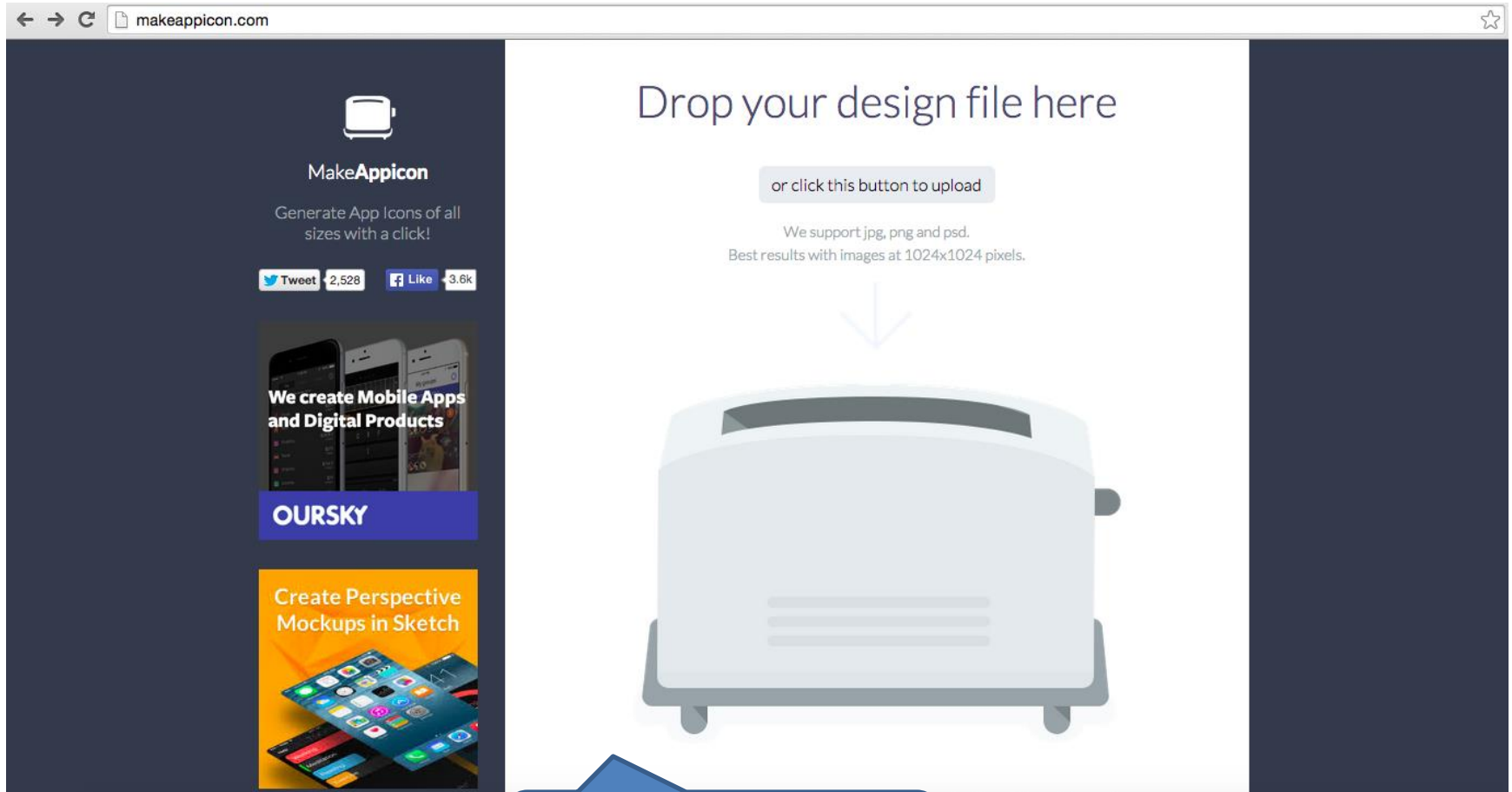


Image Assets and Icons



Use this web site to
generate the icon sets

Image Assets and Icons

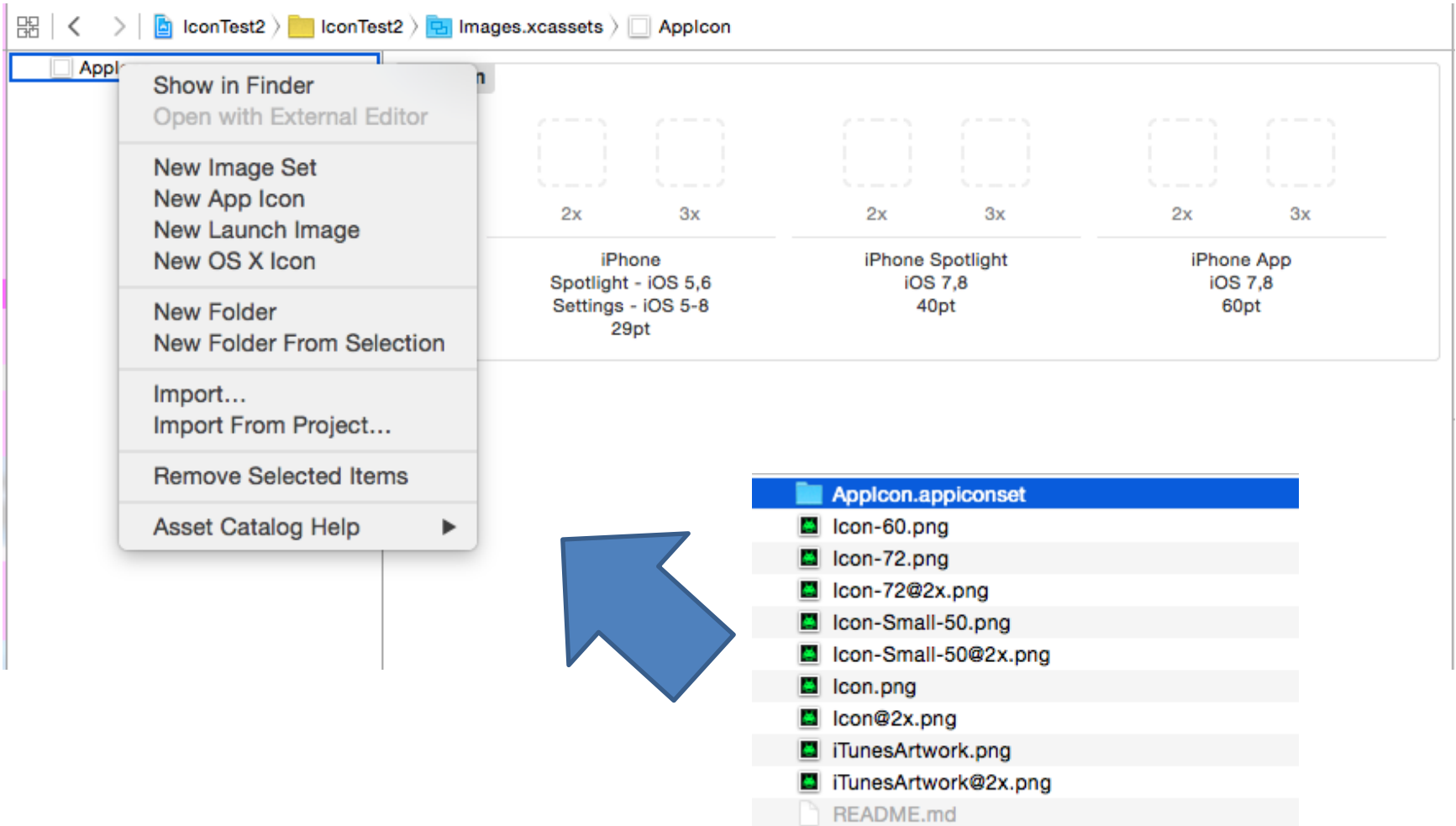


Image Assets and Icons

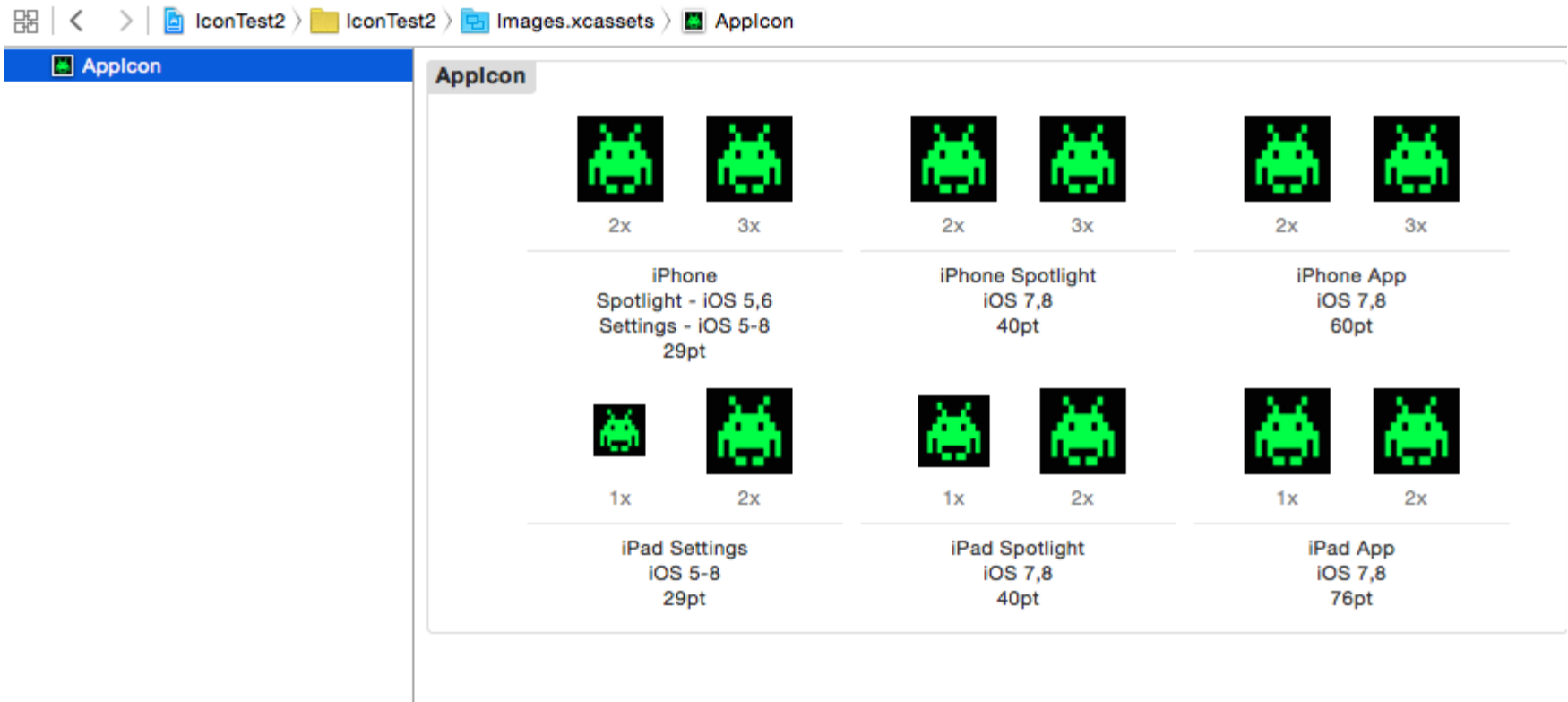
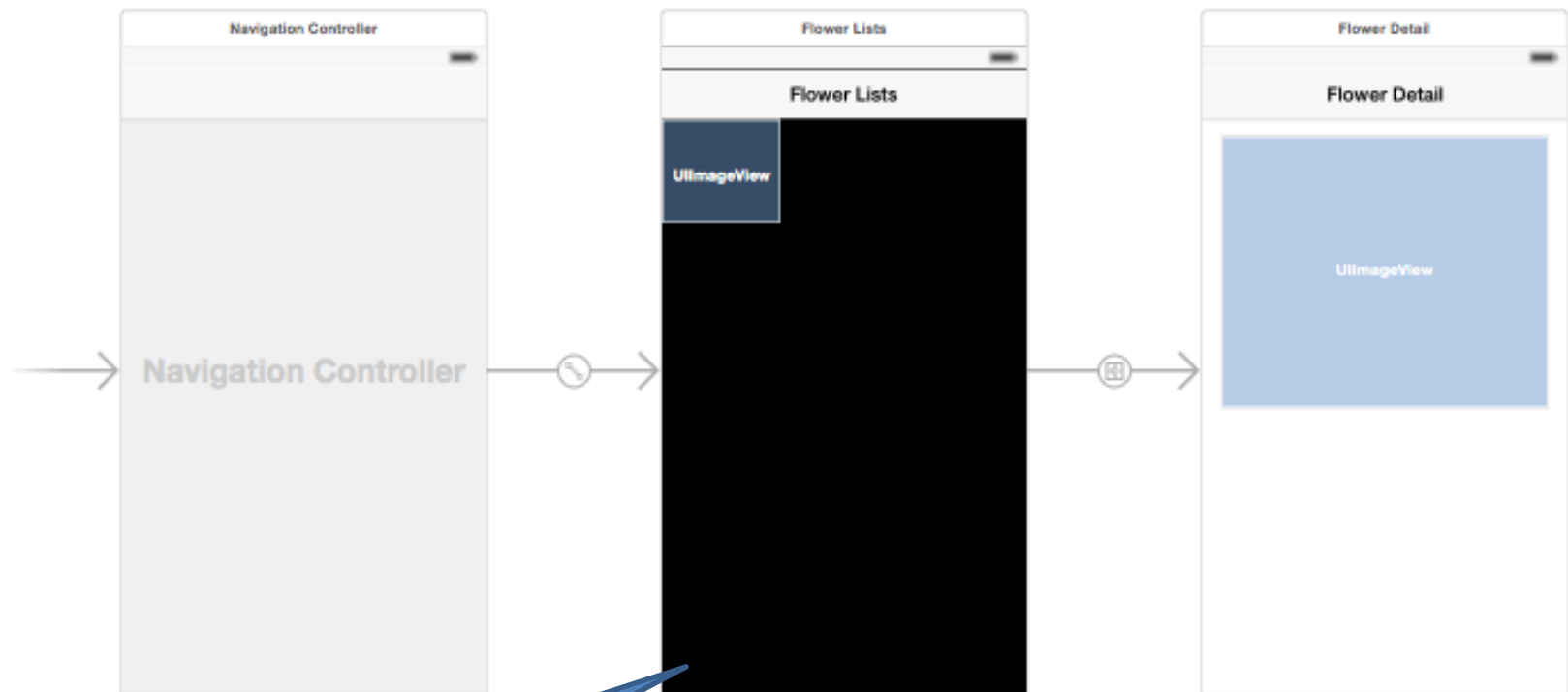


Image Assets and Icons

Icon sizes
required

Name	Size (px)	Usage
Icon-Small.png	29x29	iPad Settings
Icon-Small@2x.png	58x58	iPhone Settings, iPad Settings for Retina display
Icon-Small@3x.png	87x87	iPhone Settings for Retina display
Icon-40.png	40x40	iPad Spotlight results
Icon-40@2x.png	80x80	iPhone Spotlight results, iPad Spotlight results for retina display
Icon-40@3x.png	120x120	iPhone Spotlight results for retina display
Icon-60@2x.png	120x120	iPhone App Icon
Icon-60@3x.png	180x180	iPhone App Icon for Retina display
Icon-76.png	76x76	iPad App Icon
Icon-76@2x.png	152x152	iPad App Icon for Retina display
Icon-120.png	120x120	CarPlay App Icon
iTunesArtwork.png	512x512	App Submission
iTunesArtwork@2x.png	1024x1024	App Submission

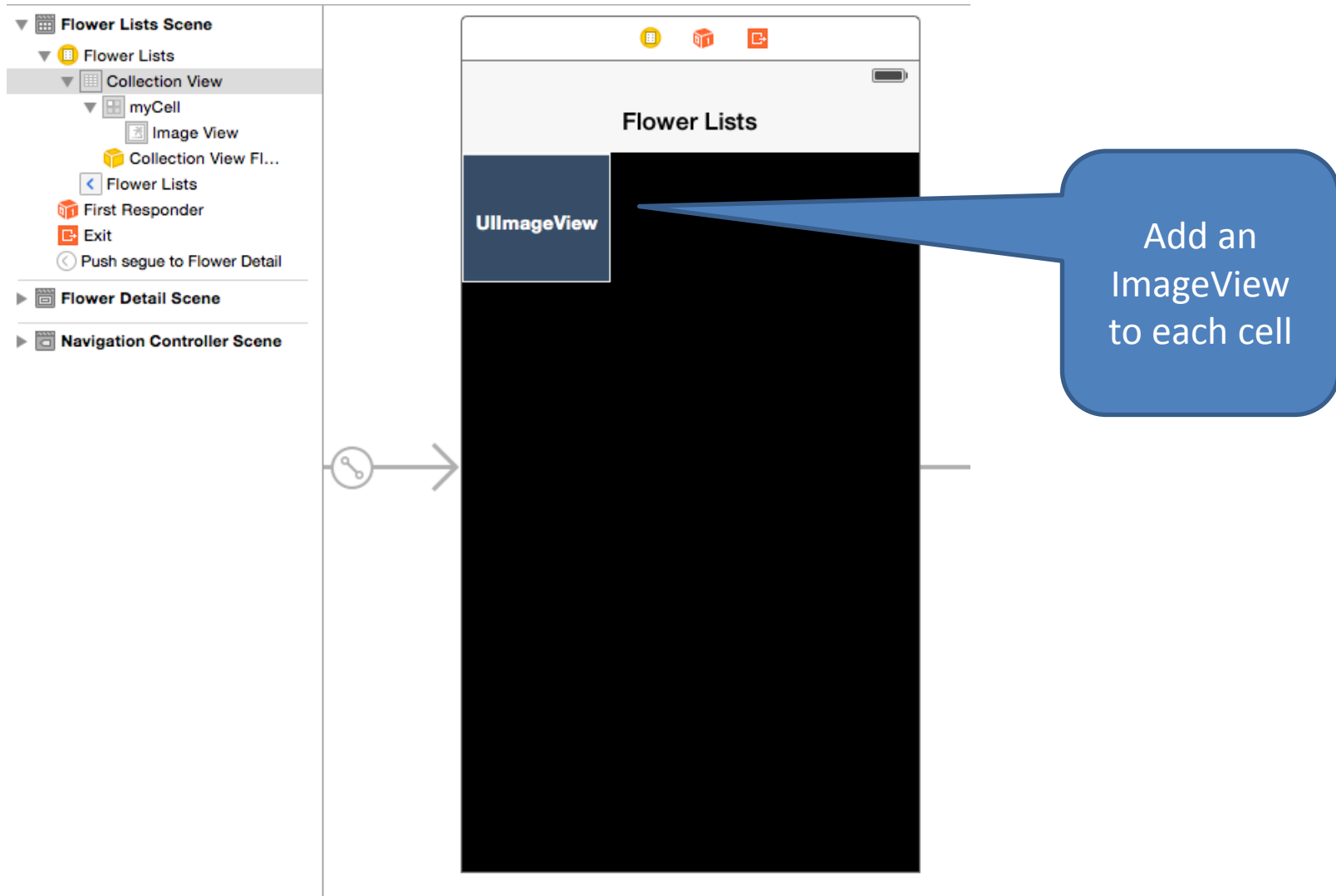
Collection View



This is the
CollectionView
Controller

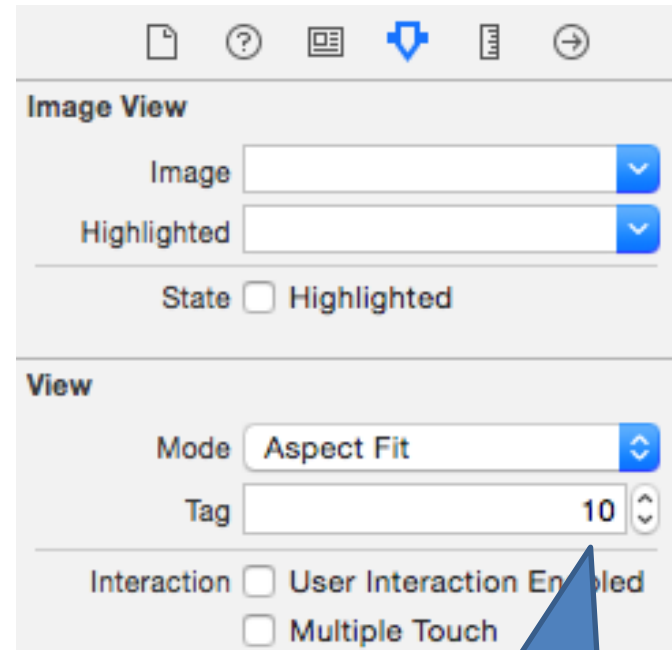
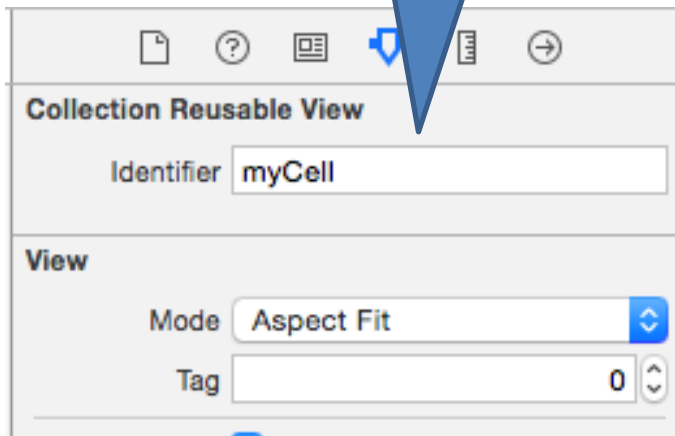
Design the
Storyboard like this

Collection View



Collection View

Give the name for each cell as "myCell"



Give a tag number to identify the UIImageView

Collection View

```
#import "MyCollectionViewController.h"
#import "DetailViewController.h"

@interface MyCollectionViewController ()
@property(n nonatomic) NSArray *myarray;
@end

@implementation MyCollectionViewController

static NSString * const reuseIdentifier = @"Cell";

- (void)viewDidLoad {
    [super viewDidLoad];

    _myarray = @[@"flower1.jpeg",@"flower2.jpeg",@"flower3.jpeg",@"flower4.jpeg",@"flower5.jpeg"];

    // Uncomment the following line to preserve selection between presentations
    // self.clearsSelectionOnViewWillAppear = NO;

    // Register cell classes
    [self.collectionView registerClass:[UICollectionViewCell class] forCellWithReuseIdentifier:
        reuseIdentifier];

    // Do any additional setup after loading the view.
}
```

Create an array for
storing the image
names

CollectionView

```
#pragma mark <UICollectionViewDataSource>
```

```
- (NSInteger)numberOfSectionsInCollectionView:(UICollectionView *)collectionView {
#warning Incomplete method implementation -- Return the number of sections
    return 1;
}
```

Return the
number of
sections

```
- (NSInteger)collectionView:(UICollectionView *)collectionView numberOfItemsInSection:(NSInteger)
    section {
#warning Incomplete method implementation -- Return the number of items in the section
    return [_myarray count];
}
```

Return
the of
number
elements

```
- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:
    (NSIndexPath *)indexPath {
    UICollectionViewCell *cell = [collectionView dequeueReusableCellWithReuseIdentifier:@"mycell"
        forIndexPath:indexPath];
```

```
// Configure the cell
```

```
UIImageView *cellimage = (UIImageView *)[cell viewWithTag:10];
```

```
cell.backgroundView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"photoframe"]];
```

```
cellimage.image = [UIImage imageNamed:[_myarray objectAtIndex:indexPath.row]];
```

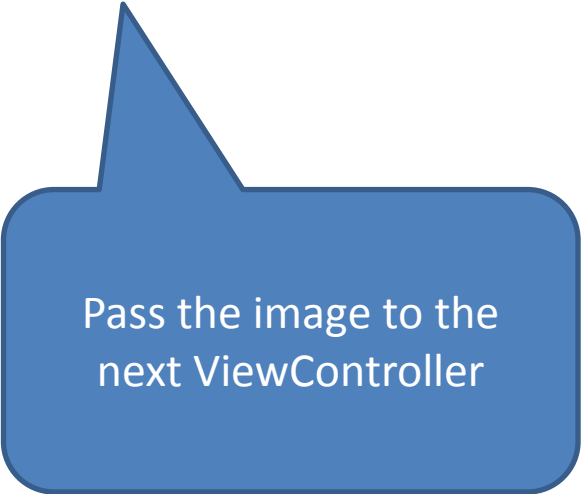
```
return cell;
```

```
}
```

Identify the UIImageView in each cell and add
the photo and background images.

Collection View

```
-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {  
    DetailViewController *dest = (DetailViewController *)[segue destinationViewController];  
    NSArray *indexPaths = [self.collectionView indexPathsForSelectedItems];  
    NSIndexPath *indexPath = [indexPaths objectAtIndex:0];  
    int cellnum = (int)[indexPath row];  
    UIImage *image = [UIImage imageNamed:[_myarray objectAtIndex:cellnum]];  
    dest.myImage = image;  
}
```



Pass the image to the
next ViewController

CollectionView

```
#import <UIKit/UIKit.h>

@interface DetailViewController : UIViewController

@property UIImage *myImage;

@end
```

Add an Image property to receive the image sent from the CollectionView

```
#import "DetailViewController.h"

@interface DetailViewController ()

@property (weak, nonatomic) IBOutlet UIImageView *myImageView;

@end

@implementation DetailViewController

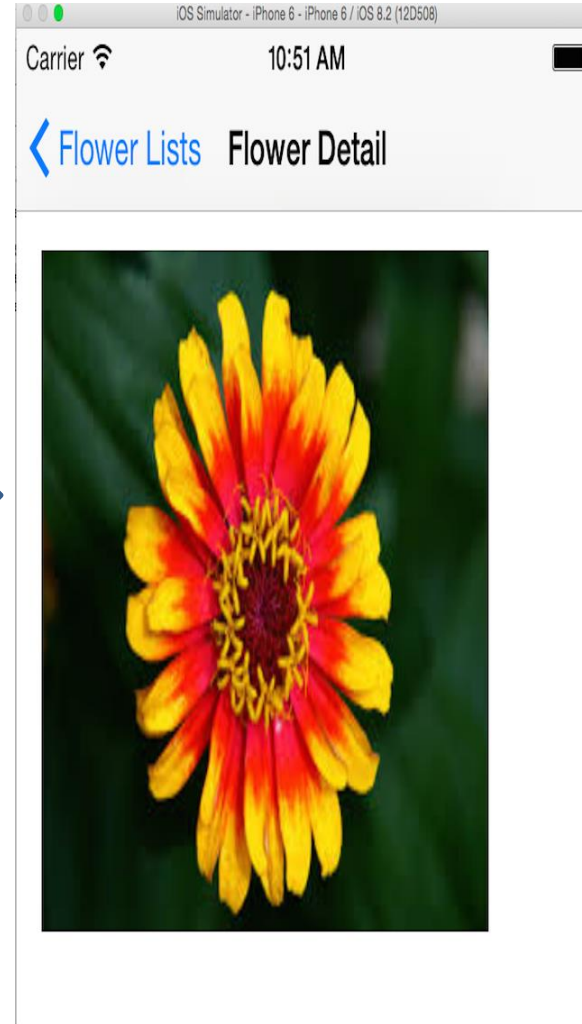
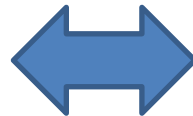
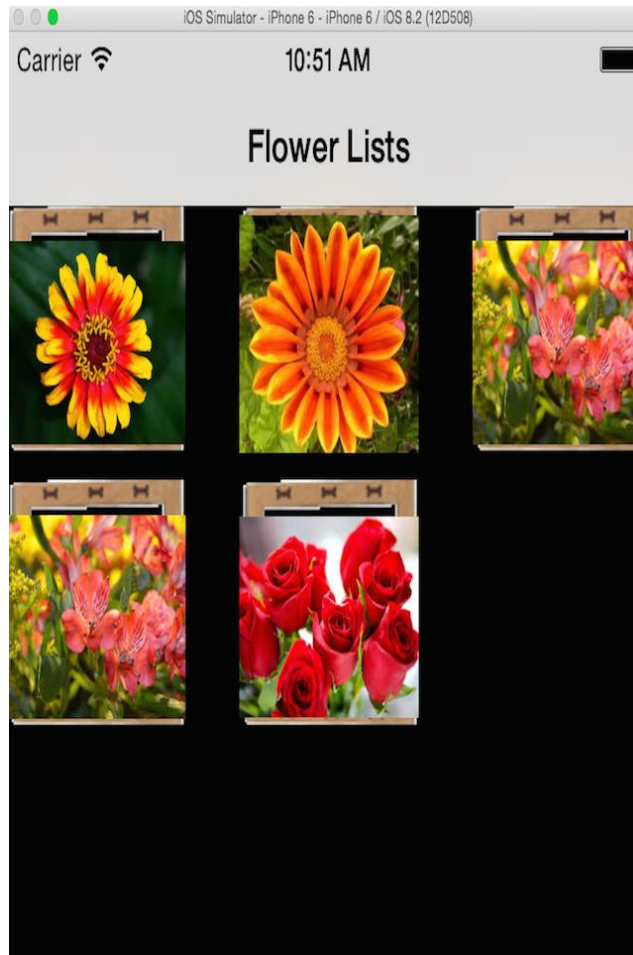
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    _myImageView.image = _myImage;
}

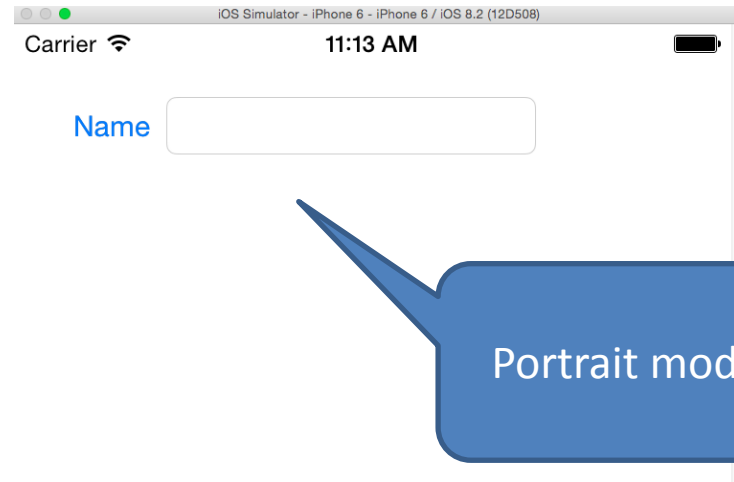
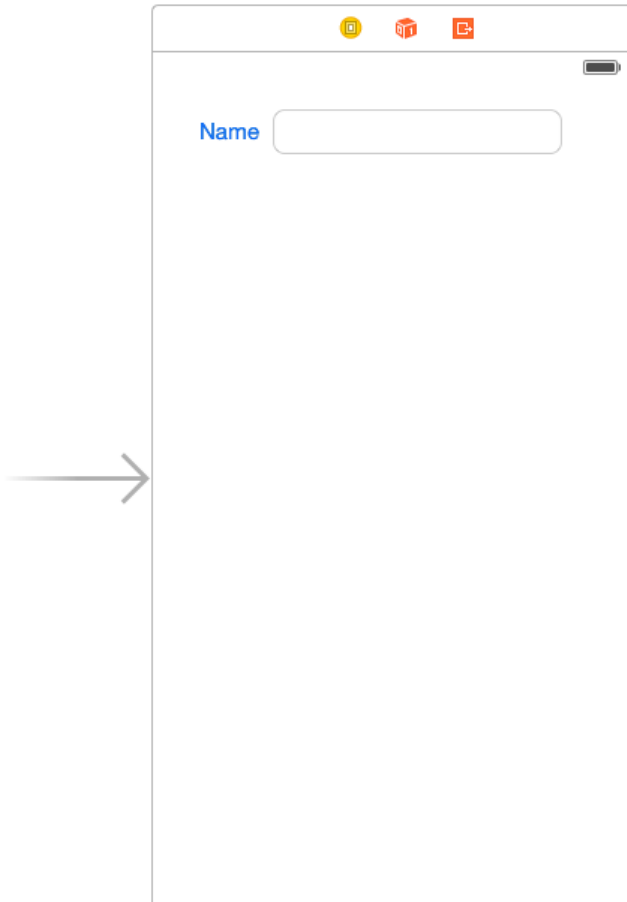
}
```

Load the Image

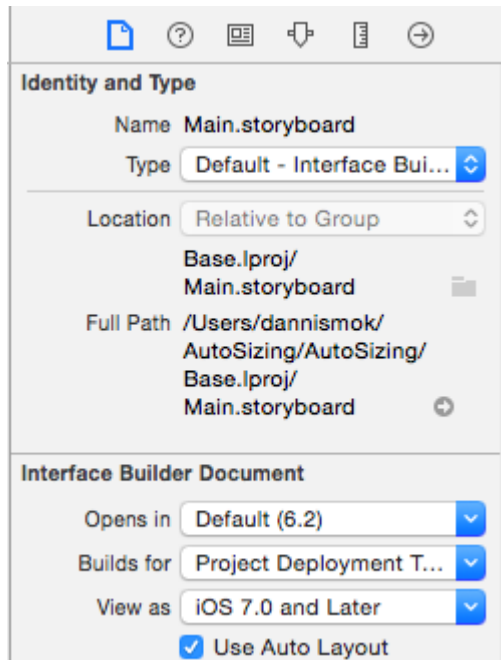
Collection View



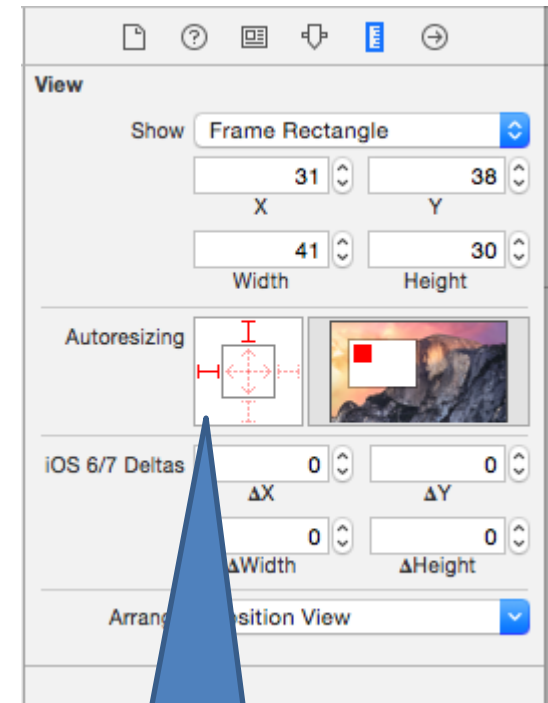
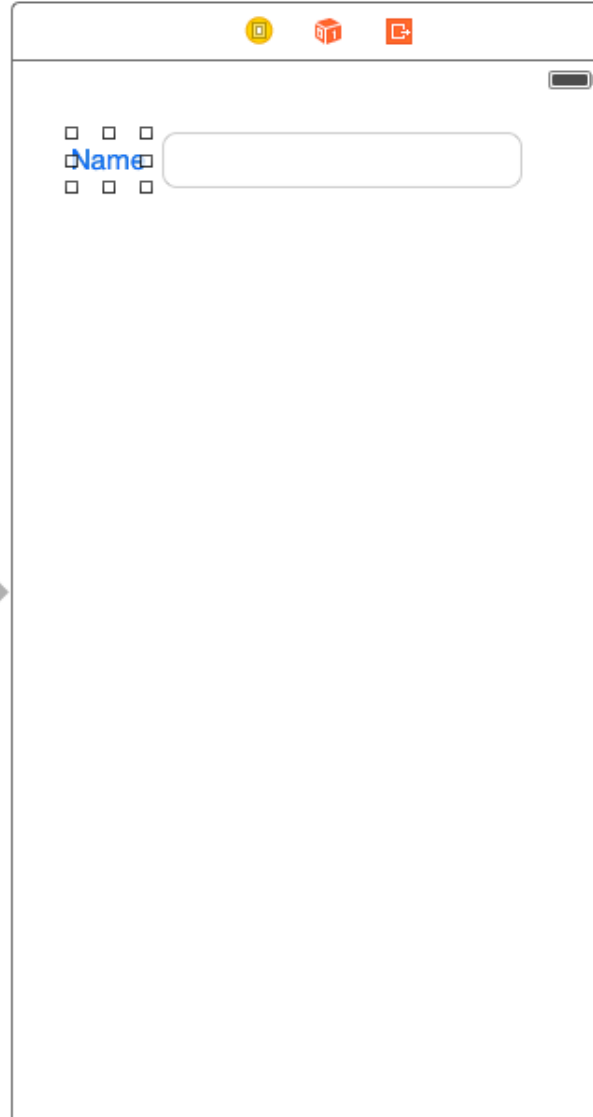
Auto Sizing



Auto Sizing

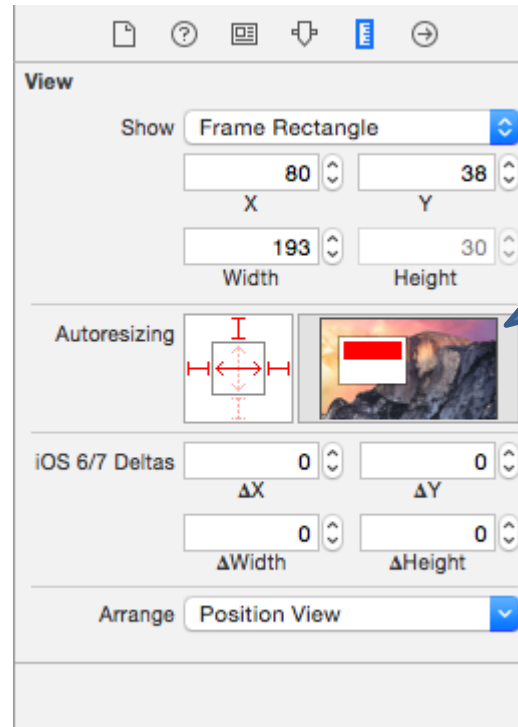
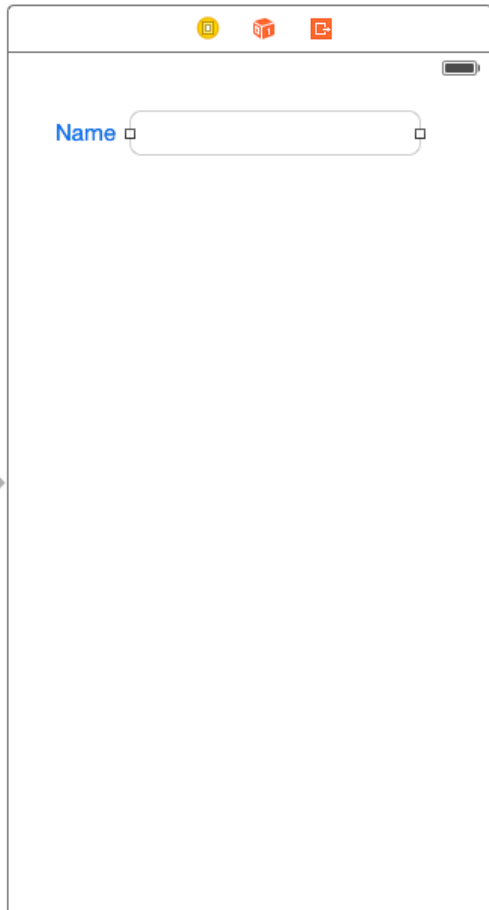


Uncheck Auto Layout



Use this red bars to fix the upper and left distance of the Name label

Auto Sizing



Fix the distances around and let the box expands to fill up the space

The box can be expanded in landscape mode



Production Digital Certificate

Developer Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles

Dannis Mok ▾

iOS Apps ▾

Certificates

- All
- Pending
- Development
- Production**

Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

iOS Certificates (Production)

1 Certificates Total

Name	Type	Expires
Dannis Mok	iOS Distribution	Apr 10, 2016

Select this to create production certificate for upload to Apple Store

Select this to add a new one

Production Digital Certificate

Apple Developer Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles

Dannis Mok ▾

iOS Apps ▾

Certificates

- All
- Pending
- Development
- Production**

Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development

Add iOS Certificate

+ 🔍

Select Type Request Generate Download

 **What type of certificate do you need?**

Development

- ☐ **iOS App Development**
Sign development versions of your iOS app.
- ☐ **Apple Push Notification service SSL (Sandbox)**
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment. A separate certificate is required for each app you develop.

Production

- ☒ **App Store and Ad Hoc**
Sign your iOS app for submission to the App Store or for Ad Hoc distribution.

Select this

Submit a CSR

The screenshot shows the Apple Developer portal interface for adding an iOS certificate. The left sidebar contains navigation links for Certificates, Identifiers, Devices, and Provisioning Profiles. The main content area is titled 'Add iOS Certificate' and shows a progress bar with four steps: Select Type, Request (active), Generate, and Approval. Below the progress bar, there is a section titled 'About Creating a Certificate Signing Request (CSR)' with a 'Certificate' icon. The text explains that to manually generate a CSR, a CSR file is needed from a Mac, and provides instructions on how to create one using Keychain Access. A list of steps is provided for creating the CSR file. At the bottom, there are 'Cancel', 'Back', and 'Continue' buttons. A blue speech bubble points to the 'Continue' button with the text 'Select file to upload'.

Developer Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles Dannis Mok

iOS Apps

Certificates

- All
- Pending
- Development
- Production**

Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development
- Distribution

Add iOS Certificate + 🔍

Select Type Request Generate Approval

About Creating a Certificate Signing Request (CSR)

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

Create a CSR file.

In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

- In the Certificate Information window, enter the following information:
 - In the User Email Address field, enter your email address.
 - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
 - The CA Email Address field should be left empty.
 - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generation.

Cancel Back Continue

Select file to upload

Download the Signed Certificate



The screenshot shows the Apple Developer portal interface. The top navigation bar includes links for Technologies, Resources, Programs, Support, and Member Center, along with a search bar. The main heading is 'Certificates, Identifiers & Profiles'. On the left sidebar, under 'iOS Apps', the 'Certificates' section is expanded, showing options for All, Pending, Development, and Production. The 'Production' option is selected. The main content area displays 'iOS Certificates (Production)' with a table showing one certificate. The certificate details are as follows:

Name	Type	Expires
Dannis Mok	iOS Distribution	Apr 10, 2016

Below the table, the certificate details are shown: Name: Dannis Mok, Type: iOS Distribution, Expires: Apr 10, 2016. There are 'Revoke' and 'Download' buttons at the bottom.

Download if
required

Apply an Explicit App ID

The screenshot shows the Apple Developer portal interface. The top navigation bar includes links for Technologies, Resources, Programs, Support, and Member Center, along with a search bar. The main header is 'Certificates, Identifiers & Profiles' with a user profile 'Dannis Mok'. The left sidebar contains a navigation menu with sections: 'iOS Apps', 'Certificates' (with sub-items: All, Pending, Development, Production), 'Identifiers' (with sub-items: App IDs, Pass Type IDs, Website Push IDs, iCloud Containers, App Groups, Merchant IDs), 'Devices' (with sub-item: All), and 'Provisioning Profiles' (with sub-items: All, Development, Distribution). The 'App IDs' sub-item is selected. The main content area is titled 'Register iOS App ID' and features a large 'ID' icon. Below the icon, the text reads: 'The App ID string contains two parts separated by a period (.)—an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. [Learn More](#)'. The 'App ID Description' section contains a 'Name' field with the value 'Platformer'. A blue callout bubble points to this field with the text 'App ID name'. Below the name field, a note states: 'You cannot use special characters such as @, &, *, ', "'. The 'App ID Prefix' section shows a 'Value' of '44Q4UMF5XL (Team ID)'.

Apple Developer

Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles Dannis Mok

iOS Apps

Certificates

- All
- Pending
- Development
- Production

Identifiers

- App IDs**
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development
- Distribution

Register iOS App ID

ID Registering an App ID

The App ID string contains two parts separated by a period (.)—an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. [Learn More](#)

App ID Description

Name: Platformer

You cannot use special characters such as @, &, *, ', "

App ID Prefix

Value: 44Q4UMF5XL (Team ID)

App ID name

Apply an Explicit App ID

App ID Suffix

● **Explicit App ID**

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID:

We recommend using a reverse-domain name style string (i.e. com.domainname.appname). It cannot contain an asterisk (*)

Add a explicit bundle ID using reverse domain name

● **Wildcard App ID**

This allows you to use a single App ID to match multiple apps. To create a wildcard App ID, enter an asterisk (*) as the last digit in the Bundle ID field.

Bundle ID:

Example: com.domainname.*

Create Distribution Provisioning Profile

The screenshot shows the Apple Developer portal interface for creating an iOS provisioning profile. The left sidebar contains a navigation menu with sections: Certificates, Identifiers, Devices, and Provisioning Profiles. The 'Provisioning Profiles' section is expanded, showing 'All', 'Development', and 'Distribution' (which is highlighted). The main content area is titled 'Add iOS Provisioning Profile' and features a progress bar with steps: Select Type, Configure, Generate, and Download. The 'Select Type' step is active, displaying a question: 'What type of provisioning profile do you need?'. Below this, there are two main categories: 'Development' and 'Distribution'. Under 'Development', there is one option: 'iOS App Development'. Under 'Distribution', there are two options: 'App Store' and 'Ad Hoc'. A blue callout bubble with the text 'Select this' points to the 'App Store' option. At the bottom of the page, there are 'Cancel' and 'Continue' buttons.

Developer Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles Dannis Mok

iOS Apps

Certificates

- All
- Pending
- Development
- Production

Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development
- Distribution**

Add iOS Provisioning Profile + [icon] [icon]

Select Type Configure Generate Download

What type of provisioning profile do you need?

Development

- ☐ **iOS App Development**
Create a provisioning profile to install development apps on test devices.

Distribution

- ☒ **App Store**
Create a distribution provisioning profile to submit your app to the App Store.
- ☐ **Ad Hoc**
Create a distribution provisioning profile to install your app on a limited number of registered devices.

Select this

Cancel Continue

Create Distribution Provisioning Profile

The screenshot shows the Apple Developer portal interface for creating a provisioning profile. The top navigation bar includes links for Technologies, Resources, Programs, Support, and Member Center, along with a search bar. The main header is 'Certificates, Identifiers & Profiles' with a user dropdown 'Dannis Mok'. The left sidebar contains a tree view with sections: Certificates (All, Pending, Development, Production), Identifiers (App IDs, Pass Type IDs, Website Push IDs, iCloud Containers, App Groups, Merchant IDs), Devices (All), and Provisioning Profiles (All, Development, Distribution). The 'Distribution' option is highlighted. The main content area is titled 'Add iOS Provisioning Profile' and shows a progress bar with steps: Select Type, Configure, Generate, and Download. The 'Select App ID' step is active, displaying a gear icon with 'PROV' and the title 'Select App ID.'. Below the title is explanatory text about App IDs and a dropdown menu for 'App ID:' showing 'Platformer (44Q4UMF5XL.com.uec.platformer)'. A blue callout bubble points to the dropdown with the text 'Select the App ID'. At the bottom are 'Cancel', 'Back', and 'Continue' buttons.

Apple Developer

Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles Dannis Mok

iOS Apps

Certificates

- All
- Pending
- Development
- Production

Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development
- Distribution**

Add iOS Provisioning Profile

Select Type Configure Generate Download

Select App ID.

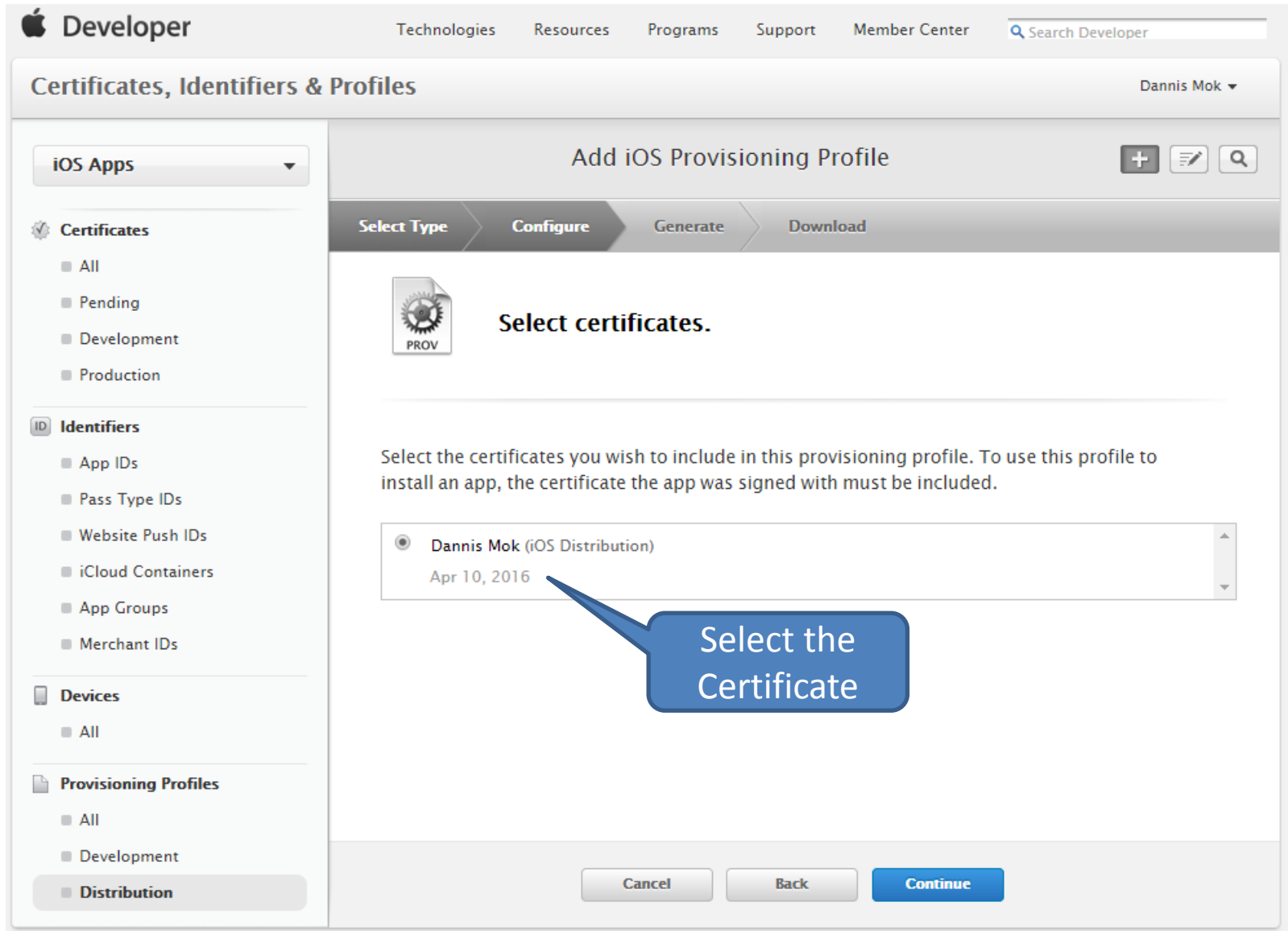
If you plan to use services such as Game Center, In-App Purchase, and Push Notifications, or want a Bundle ID unique to a single app, use an explicit App ID. If you want to create one provisioning profile for multiple apps or don't need a specific Bundle ID, select a wildcard App ID. Wildcard App IDs use an asterisk (*) as the last digit in the Bundle ID field. Please note that iOS App IDs and Mac App IDs cannot be used interchangeably.

App ID: Platformer (44Q4UMF5XL.com.uec.platformer)

Select the App ID

Cancel Back Continue

Create Distribution Provisioning Profile



Create Distribution Provisioning Profile

Developer Technologies Resources Programs Support Member Center Search Developer

Certificates, Identifiers & Profiles

 Dannis Mok ▾

iOS Apps ▾

- Certificates**
 - All
 - Pending
 - Development
 - Production
- Identifiers**
 - App IDs
 - Pass Type IDs
 - Website Push IDs
 - iCloud Containers
 - App Groups
 - Merchant IDs
- Devices**
 - All

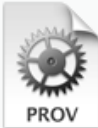
Add iOS Provisioning Profile

 + ✎ 🔍

Select Type Configure Generate **Download**

 **Your provisioning profile is ready.**

Download and Install
Download and double click the following file to install your Provisioning Profile.

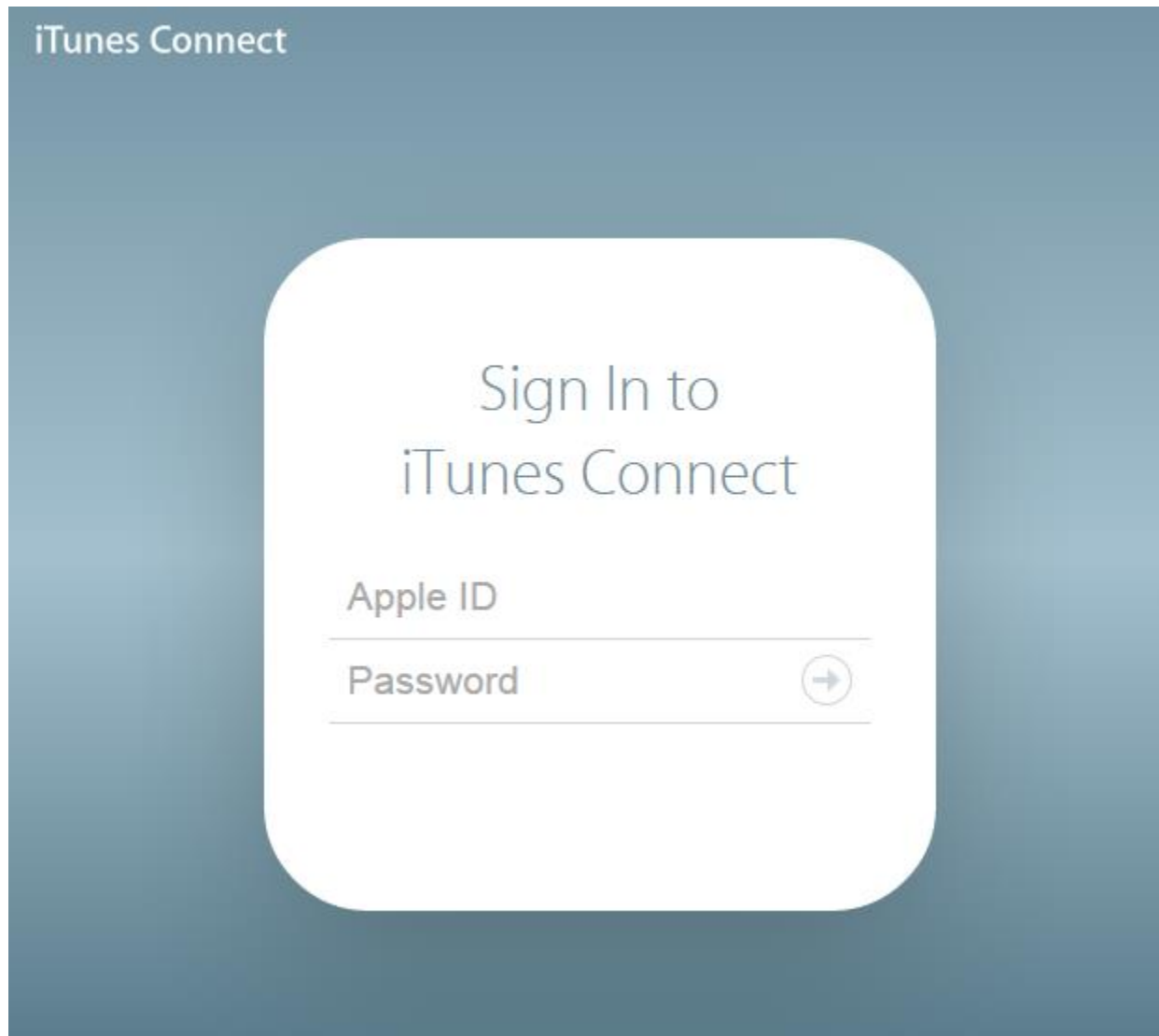


Name: PlatformerProfile
Type: iOS Distribution
App ID: 44Q4UMF5XL.com.uec.platformer
Expires: Apr 10, 2016

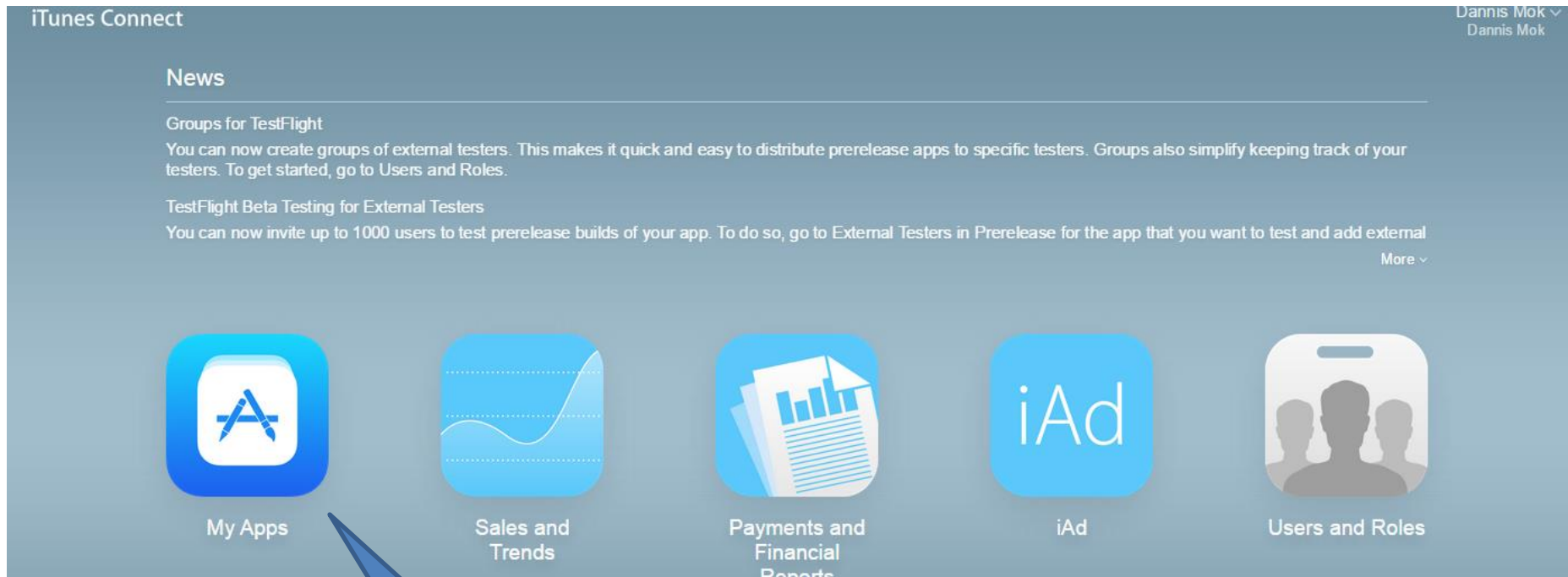
Download

Download if required

Using iTunes Connect upload Apps



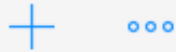
Using iTunes Connect upload Apps



Using iTunes Connect upload Apps

Click here to add a new App

iTunes Connect [My Apps](#) ▾

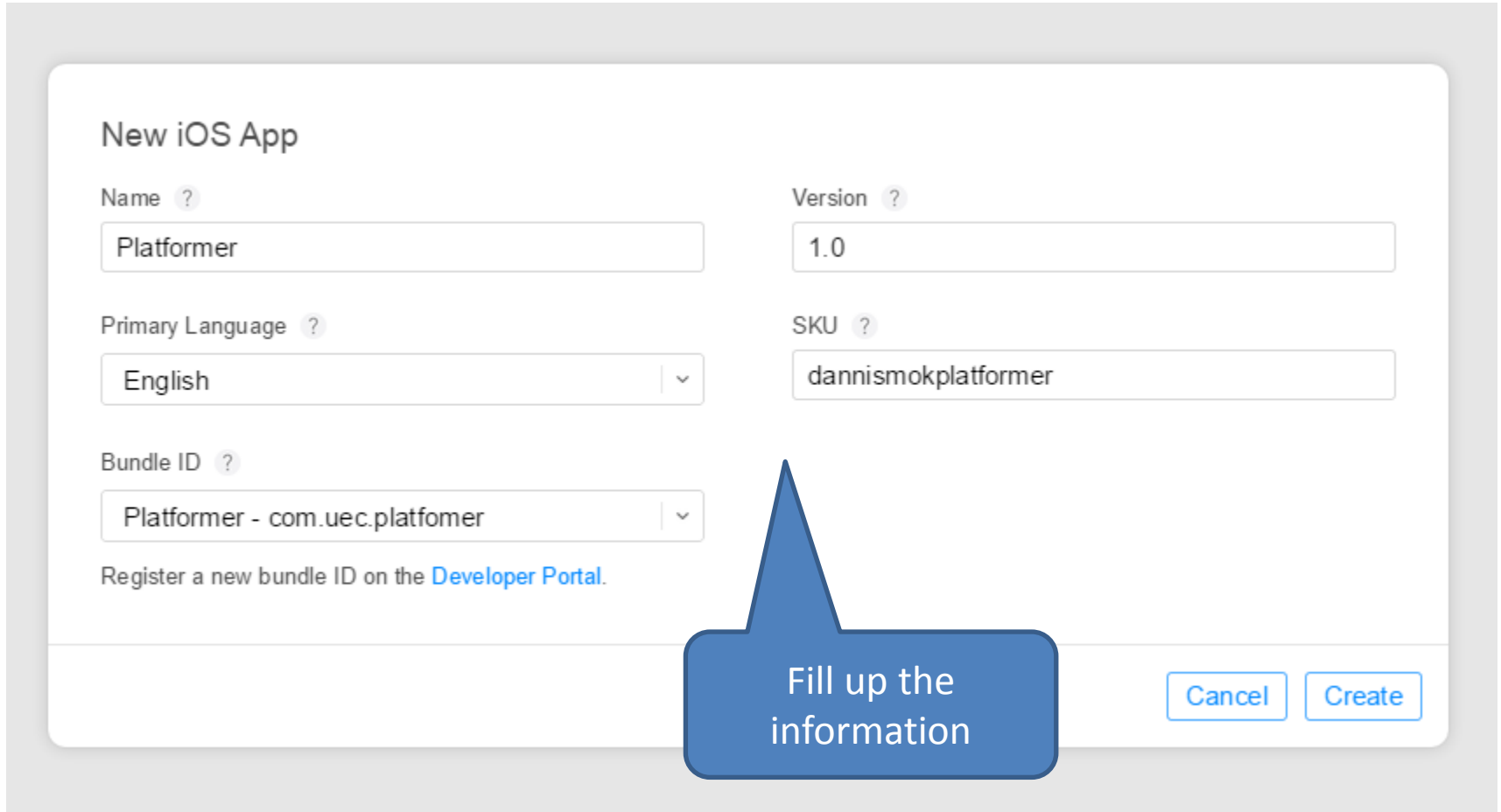


Space
Invasion...

iOS

● 1.0 Waiting For Review

Using iTunes Connect upload Apps



The screenshot shows the 'New iOS App' form in iTunes Connect. The form contains the following fields:

- Name**: Platformer
- Version**: 1.0
- Primary Language**: English
- SKU**: dannismokplatformer
- Bundle ID**: Platformer - com.uec.platfomer

Below the Bundle ID field, there is a link: Register a new bundle ID on the [Developer Portal](#).

A blue callout bubble with the text 'Fill up the information' points to the form fields.

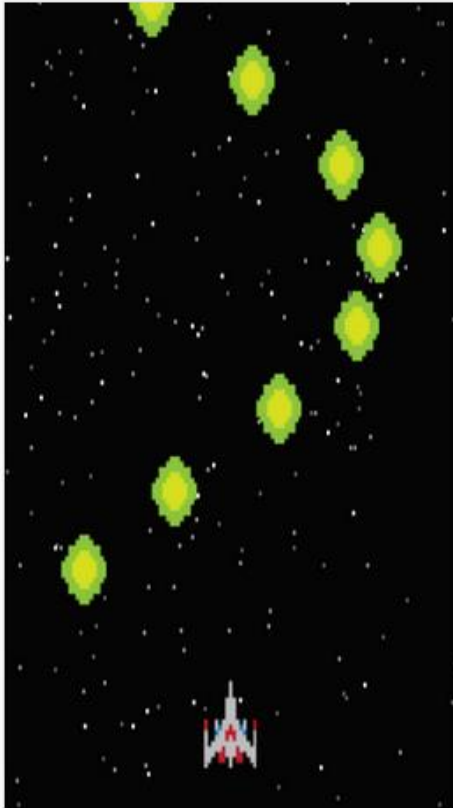
At the bottom right, there are two buttons: **Cancel** and **Create**.

Using iTunes Connect upload Apps

Version Information

App Video Preview and Screenshots ?

4.7-Inch 5.5-Inch 4-Inch 3.5-Inch iPad



Can upload at most 5
screen captures

Using iTunes Connect upload Apps

Name ?
Space Invasion Dannis Mok

Description ?
Space Invasion for everyone

Keywords ?
space invasion 86

Support URL ?
http://www.uec.edu.hk

Marketing URL ?
http://example.com (optional)

Privacy Policy URL ?
http://example.com (optional)

3973

Fill in the App Name,
Description, Keywords,
Support URL

Using iTunes Connect upload Apps

General App Information

App Icon ?



Add an icon here.
The icon must be
1024 x 1024

Apple ID ?

984856135

Version ?

1.0

Fill in the version and
select the category

Category ?

Games

Action

Subcategory (optional)

Entertainment

Rating [Edit](#)

Ages 4+

Edit the Rating

[Additional Ratings](#)

Using iTunes Connect upload Apps

iTunes Connect [My Apps](#) CollectionView Dannis Mok Dannis Mok ?

[App Store](#) [Features](#) [TestFlight](#) [Activity](#)

APP STORE INFORMATION

[App Information](#)

[Pricing and Availability](#)

iOS APP

1.0 Prepare for Submissi...

[+ VERSION OR PLATFORM](#)

Pricing and Availability

[Save](#)

Price Schedule

[All Prices and Currencies](#)

Price ?	Start Date ?	End Date ?
HKD 0 (Free) Other Currencies	October 7, 2015	No End Date

Availability

Available in all territories [Edit](#)

Volume Purchase Program

☐ Available with a volume discount for educational institutions ?

☒ Available with no discount

Make the price as free

Using iTunes Connect upload Apps

Edit Rating

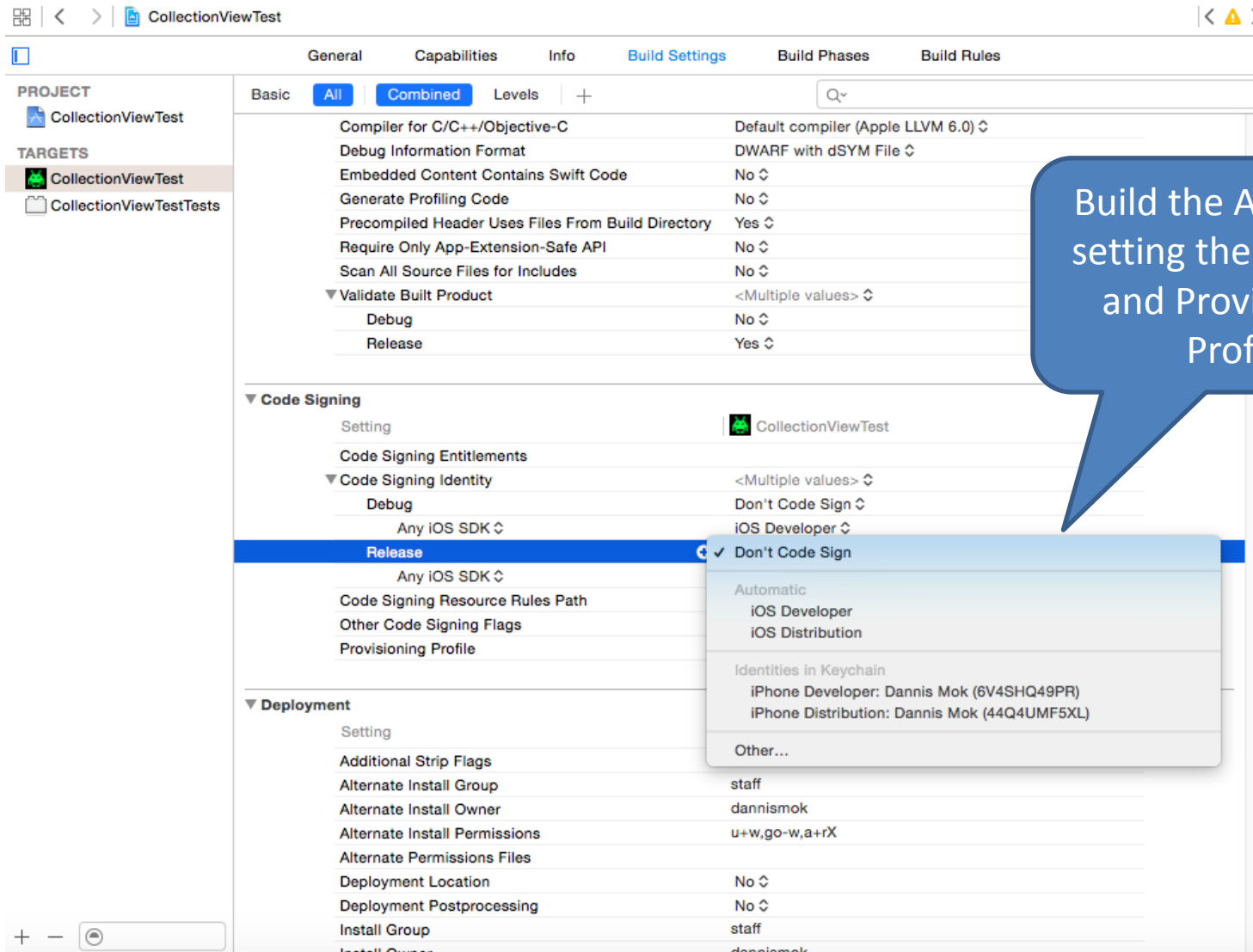
For each content description, select the level of frequency that best describes your app. The app rating that will display on the App Store is the same across all of your platforms. It is based on the app's platform with the highest rating. [Learn More](#)

Apps must not contain any obscene, pornographic, offensive, or defamatory or materials of any kind (text, graphics, images, photographs, and so on), or other content or materials that in Apple's reasonable judgement may be found objectionable.

Apple Content Description	None	Infrequent/Mild	Frequent/Intense
Cartoon or Fantasy Violence	☒	☐	☐
Realistic Violence	☒	☐	☐
Prolonged Graphic or Sadistic Realistic Violence	☒	☐	☐
Profanity or Crude Humor	☒	☐	☐
Mature/Suggestive Themes	☒	☐	☐
Horror/Fear Themes	☒	☐	☐
Medical/Treatment Information	☒	☐	☐
Alcohol, Tobacco, or Drug Use or References	☒	☐	☐
Simulated Gambling	☒	☐	☐

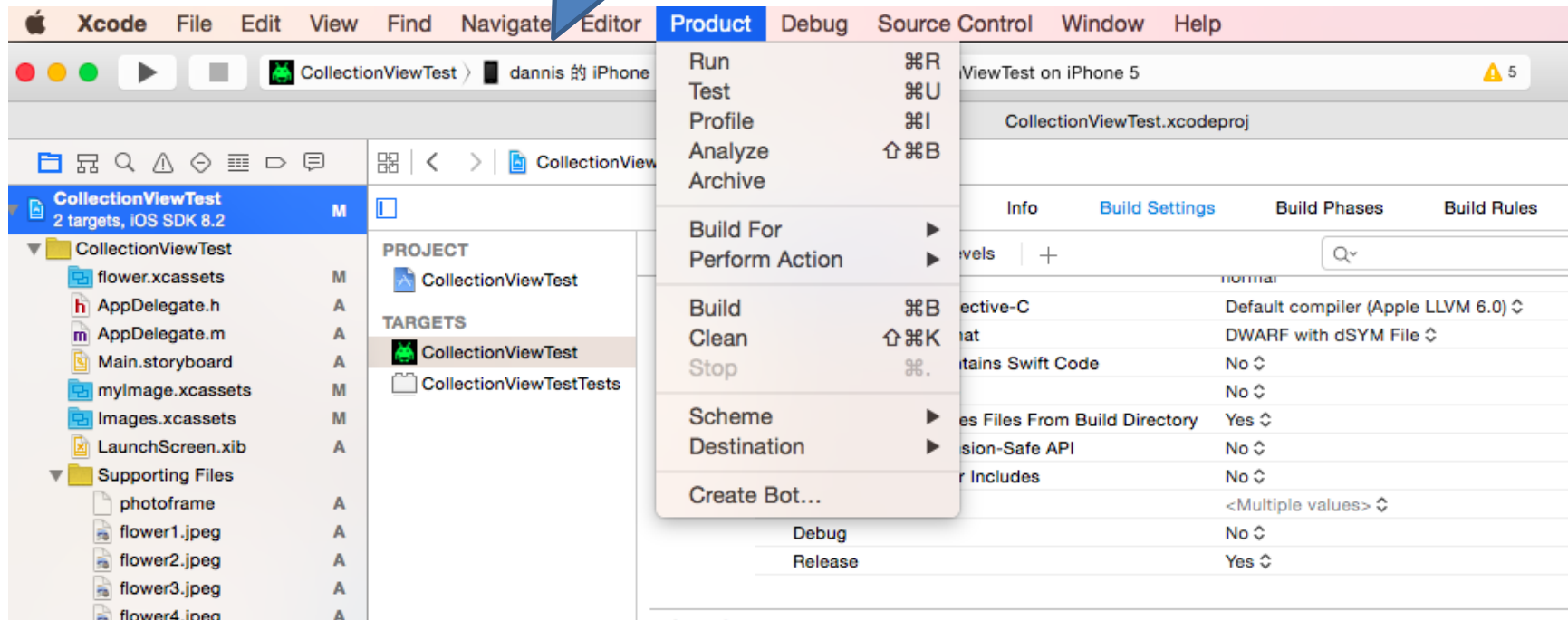
Edit the Rating

Using iTunes Connect upload Apps



Using iTunes Connect upload Apps

Select the device and
then run Archive



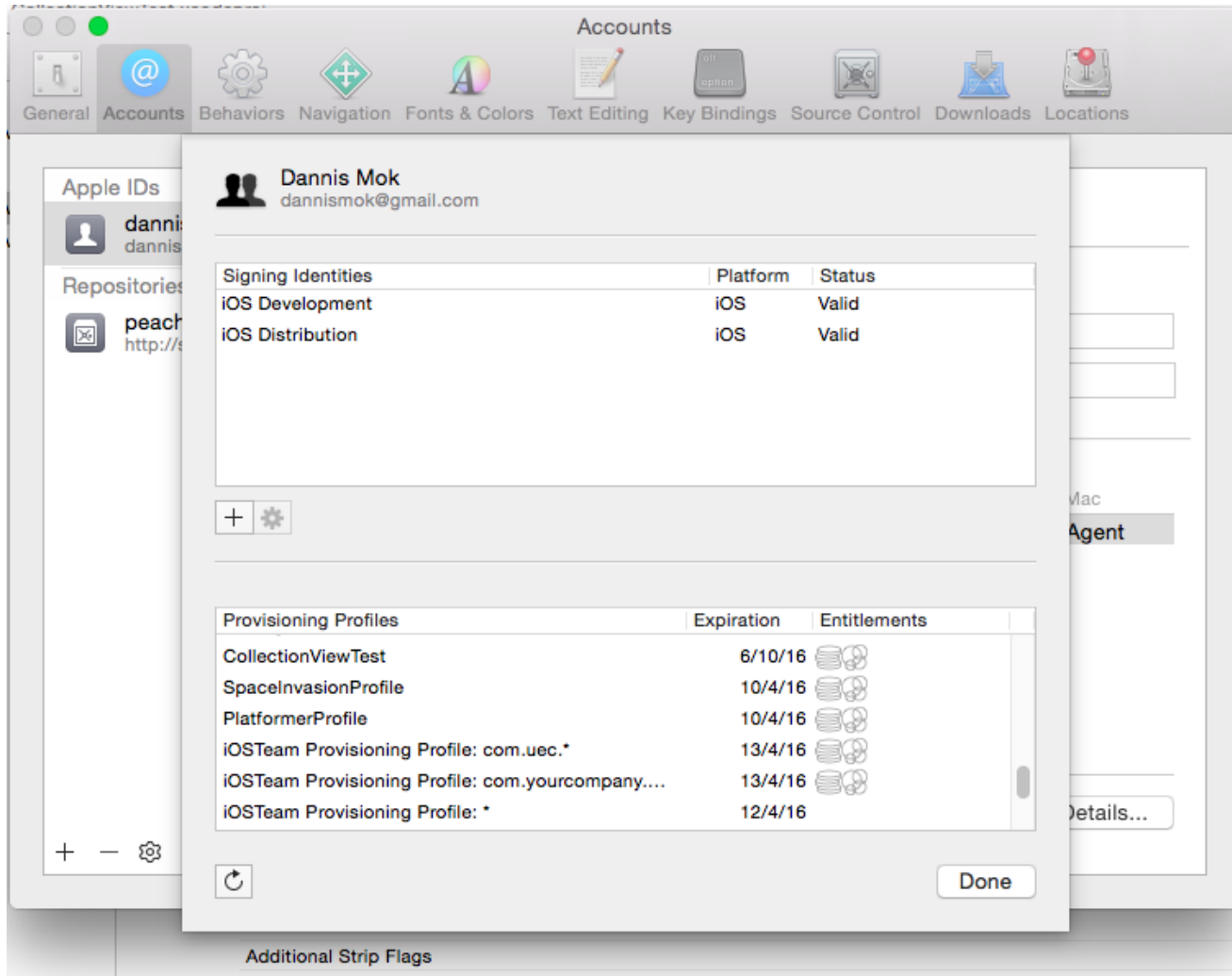
Using iTunes Connect upload Apps

Debug	No ↕
Release	Yes ↕

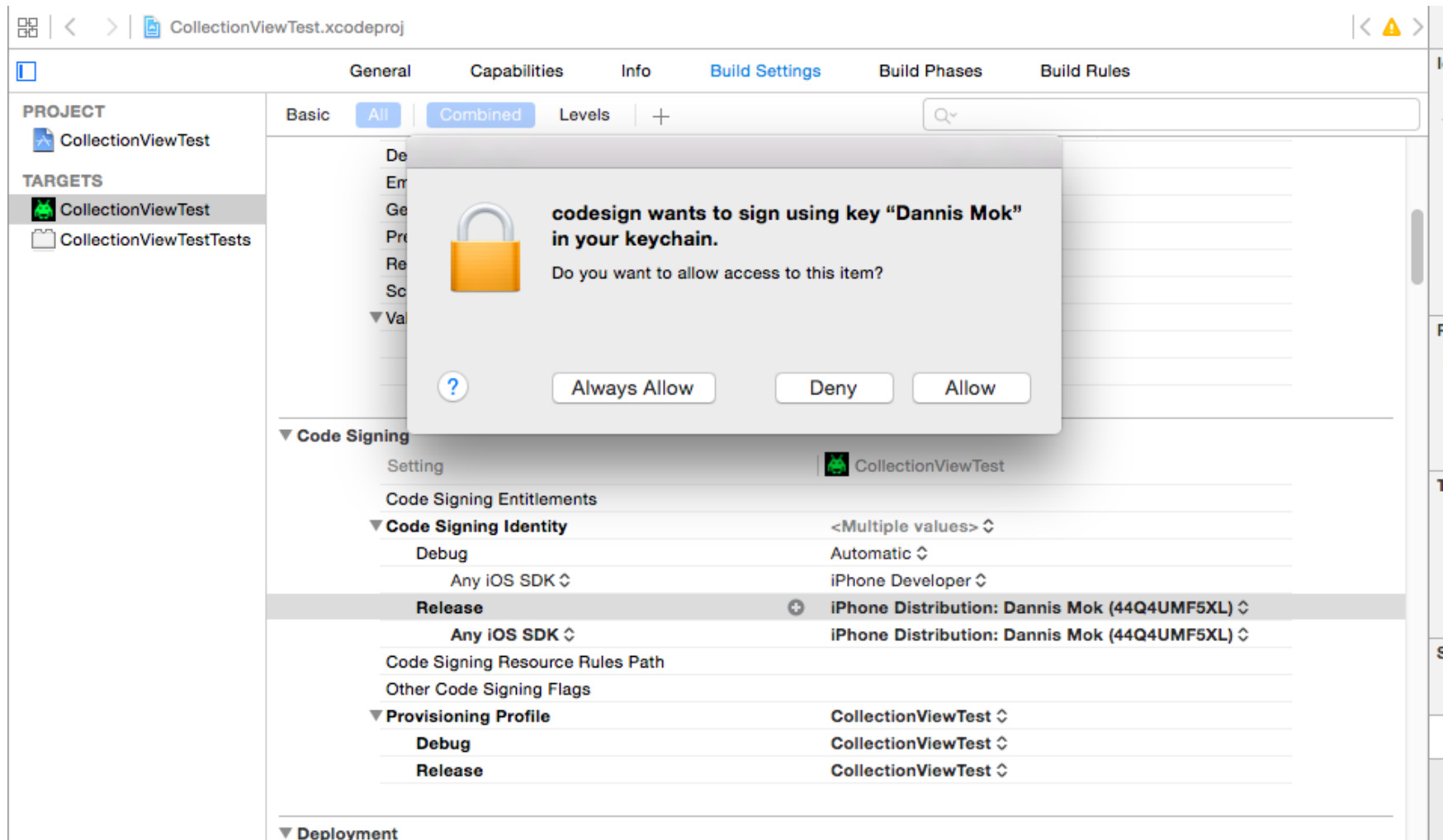
▼ **Code Signing**

Setting	 collectionViewTest
Code Signing Entitlements	
▼ Code Signing Identity	<Multiple values> ↕
Debug	Don't Code Sign ↕
Any iOS SDK ↕	iOS Developer ↕
Release	iPhone Distribution ↕
Any iOS SDK ↕	Automatic ↕
Code Signing Resource Rules Path	
Other Code Signing Flags	
▼ Provisioning Profile	<Multiple values> ↕
Debug	Automatic ↕
Release	collectionViewTest ↕

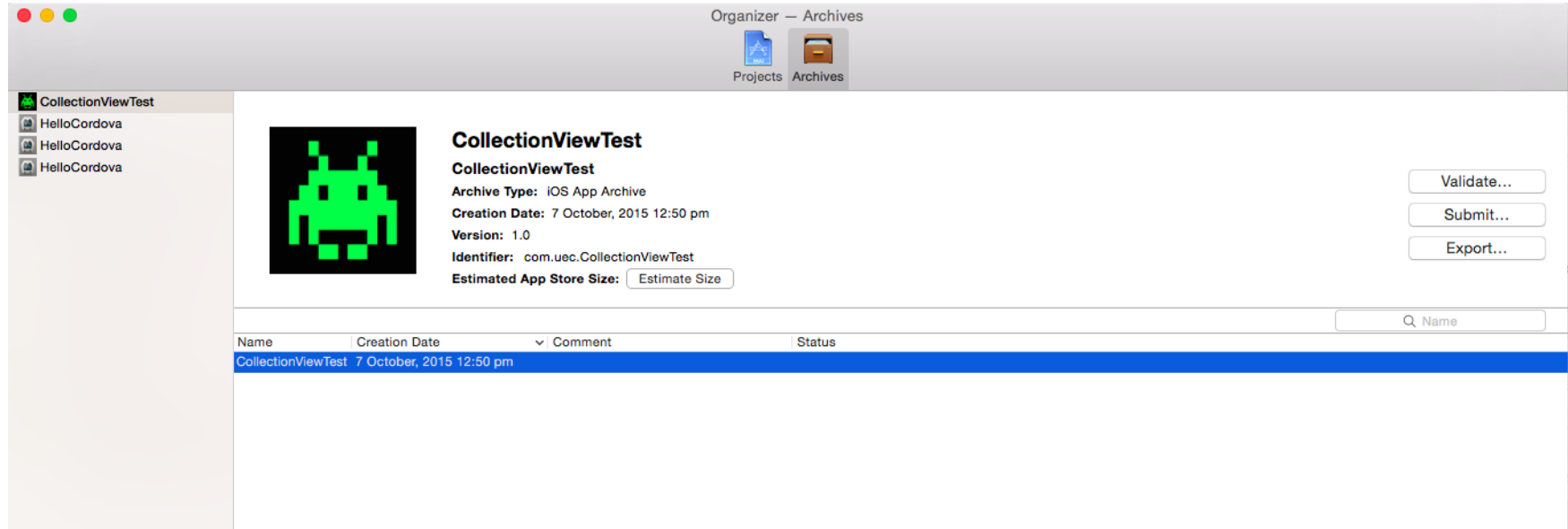
Using iTunes Connect upload Apps



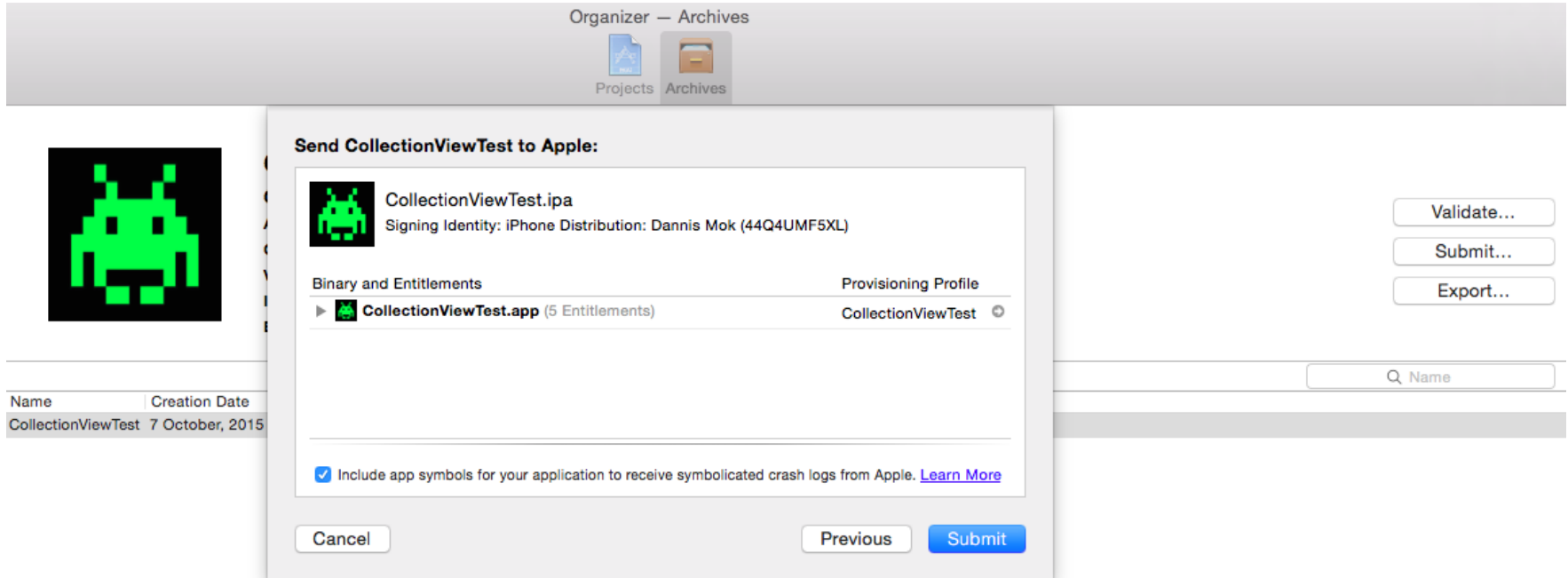
Using iTunes Connect upload Apps



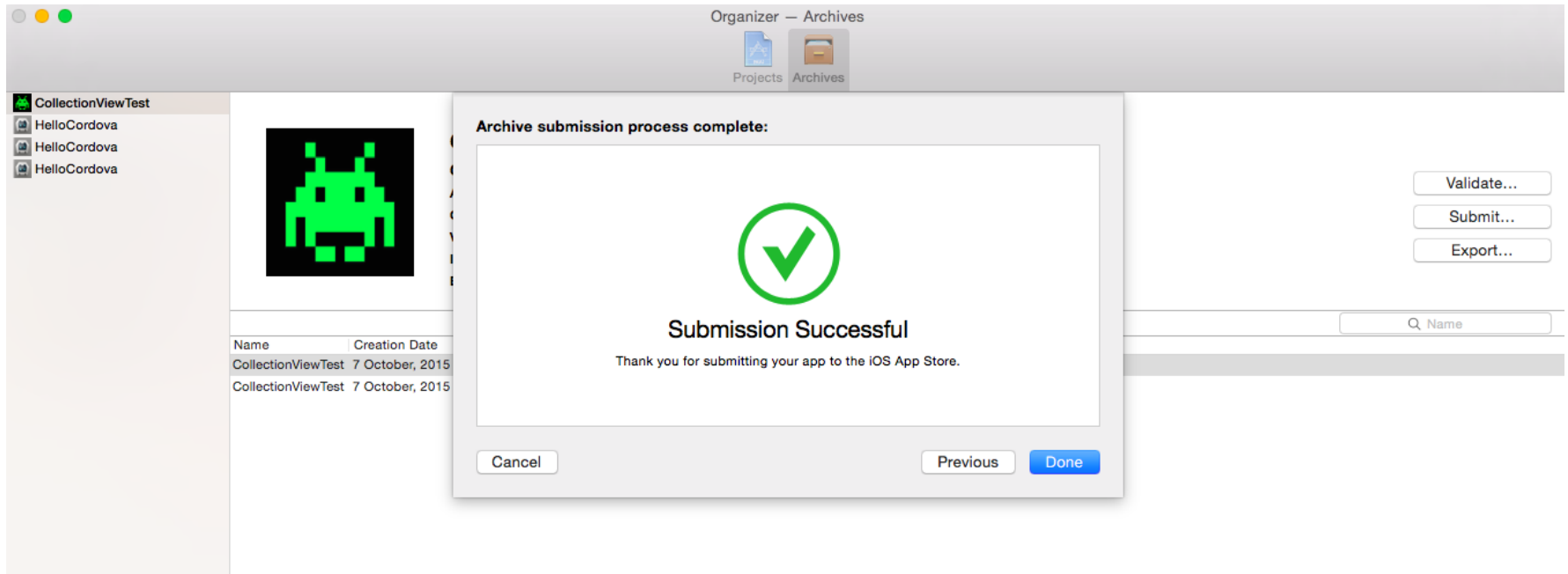
Using iTunes Connect upload Apps



Using iTunes Connect upload Apps



Using iTunes Connect upload Apps



Using iTunes Connect upload Apps

Screenshot Properties

Device	Specifications
3.5-Inch Retina Display Screenshots <i>(required)</i>	One screenshot is required. Up to four additional screenshots can be uploaded. You can rearrange the screenshots when you create or edit the iTunes Connect record. Don't include the device status bar in your screenshots. Screenshot requirements are: ▪ 72 dpi, RGB, flattened, no transparency ▪ High-quality JPEG or PNG image file format ▪ Any of the following sizes: ▫ 640 x 920 pixels for hi-res portrait (without status bar) minimum ▫ 640 x 960 pixels for hi-res portrait (full screen) maximum ▫ 960 x 600 pixels for hi-res landscape (without status bar) minimum ▫ 960 x 640 pixels for hi-res landscape (full screen) maximum

Using iTunes Connect upload Apps

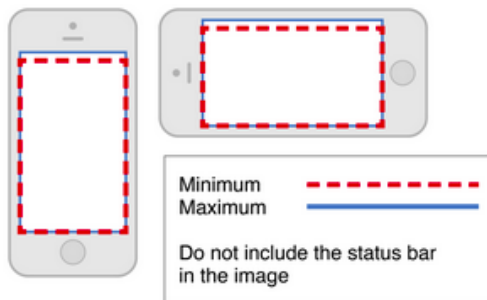
4-Inch Retina Display
Screenshots (Required)

You're required to upload at least one of these screenshots.

Up to four additional optional screenshots can be uploaded. You can rearrange the screenshots when you create or edit the iTunes Connect record.

Don't include the device status bar in your screenshots. Screenshot requirements are:

- 72 dpi, RGB, flattened, no transparency
- High-quality JPEG or PNG image file format
- Any of the following sizes:
 - 640 x 1096 pixels for portrait (without status bar) minimum
 - 640 x 1136 pixels for portrait (full screen) maximum
 - 1136 x 600 pixels for landscape (without status bar) minimum
 - 1136 x 640 pixels for landscape (full screen) minimum



Using iTunes Connect upload Apps

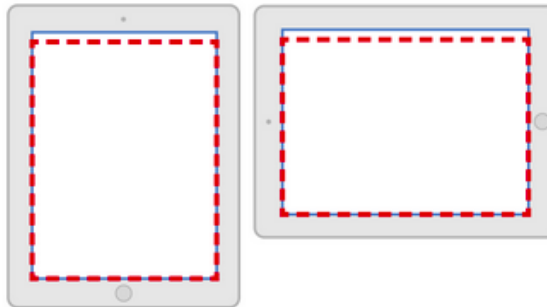
If your build indicates that your app runs on iPad, you're required to upload at least one iPad screenshot.

Up to four additional screenshots can be uploaded. You can rearrange the screenshots when you create or edit the iTunes Connect record.

Don't include the device status bar in your screenshots. Screenshots requirements are:

- 72 dpi, RGB, flattened, no transparency
- High-quality JPEG or PNG image file format
- Any of the following sizes:
 - 1024 x 748 pixels for landscape (without status bar) minimum
 - 1024 x 768 pixels for landscape (full screen) maximum
 - 2048 x 1496 pixels for hi-res (without status bar) minimum
 - 2048 x 1536 pixels for hi-res landscape (full screen) maximum
 - 768 x 1004 pixels for portrait (without status bar) minimum
 - 768 x 1024 pixels for portrait (full screen) maximum
 - 1536 x 2008 pixels for hi-res portrait (without status bar) minimum
 - 1536 x 2048 pixels for hi-res portrait (full screen) maximum

iPad Screenshots
(required if app runs on
iPad)



Using iTunes Connect upload Apps

4.7-inch Retina screenshot	If your app indicates that it is optimized for the iPhone 6, at least one 4.7-inch screenshot is required. Up to four additional screenshots can be uploaded. You can rearrange the screenshots when you create or edit the iTunes Connect record. Screenshot requirements are: ▪ 72 dpi, RGB, flattened, no transparency ▪ High-quality JPEG or PNG image file format ▪ 750 x 1334 pixels for hi-res portrait ▪ 1334 x 750 pixels for hi-res landscape
5.5-inch Retina screenshot	If your app indicates that it is optimized for the iPhone 6 Plus, at least one 5.5-inch screenshot is required. Up to four additional screenshots can be uploaded. You can rearrange the screenshots when you create or edit the iTunes Connect record. Screenshot requirements are: ▪ 72 dpi, RGB, flattened, no transparency ▪ High-quality JPEG or PNG image file format ▪ 1242 x 2208 pixels for hi-res portrait ▪ 2208 x 1242 pixels for hi-res landscape