

iOS Development @ Cohab

Brent Nycum
brentnycum@gmail.com

Developer Survey

- <http://bit.ly/pHykzE>

Developer Survey Results

- As of Sept 15
- <http://i.imgur.com/twxwn.png>

Developer Survey Times

- Times / Lengths for more classes
- <http://j.mp/nDKNjG>

- Alerts
- Sheets
- Deployment
 - Certificate Management

Alerts

Alerts

- UIAlertView
- Must implement the UIAlertViewDelegate

Alerts

- Temporary views that appear when something requires a user's attention
- Or additional choices or functionality need to be offered

Alerts

- Keep them to a minimum
- Annoying to the user
- Can't interact with the application while shown
- Probably only show on error
- Initial selection on first load

Alerts

- Unnecessary if:
 - Merely increase view of some information
 - Update that tasks are progressing normally
 - Ask for confirmation, but not in case of deletion
 - Inform of errors they can do nothing about

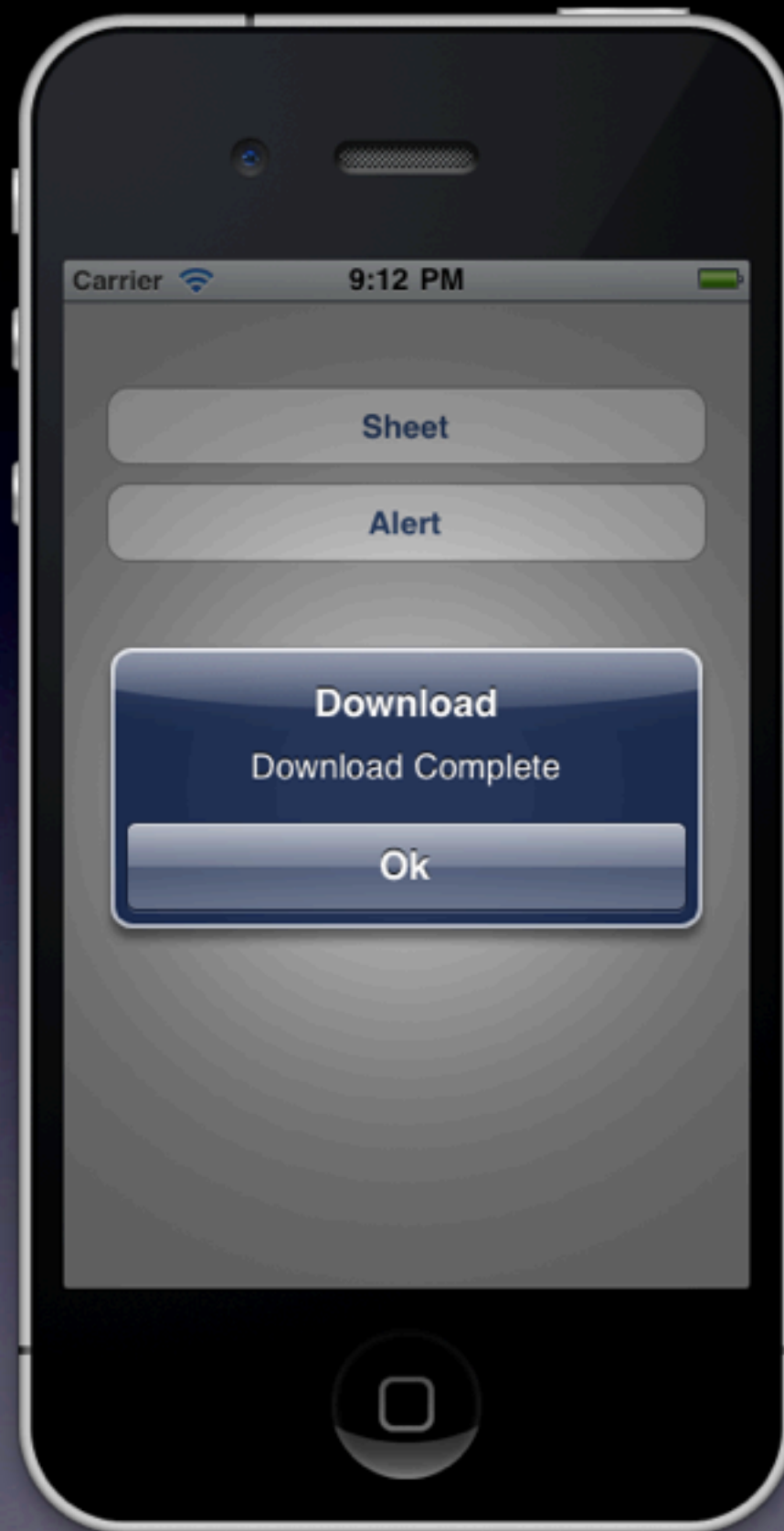
Alert

```
UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Alert!"  
                                                    message:@"This is an alert!"  
                                                    delegate:self  
                                                    cancelButtonTitle:@"Cancel"  
                                                    otherButtonTitles:@"Button 1", @"Button 2", @"Button 3", nil];
```



Alert

[illegible]



Alerts

- To display your alert
 - `[alert show];`

Alert Delegate

```
-(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
```



Index

1

2

3

0

Action Sheets

Action Sheets

- UIAlertController
- Displays a set of functions related to the task the user initiates

Action Sheets



Action Sheets

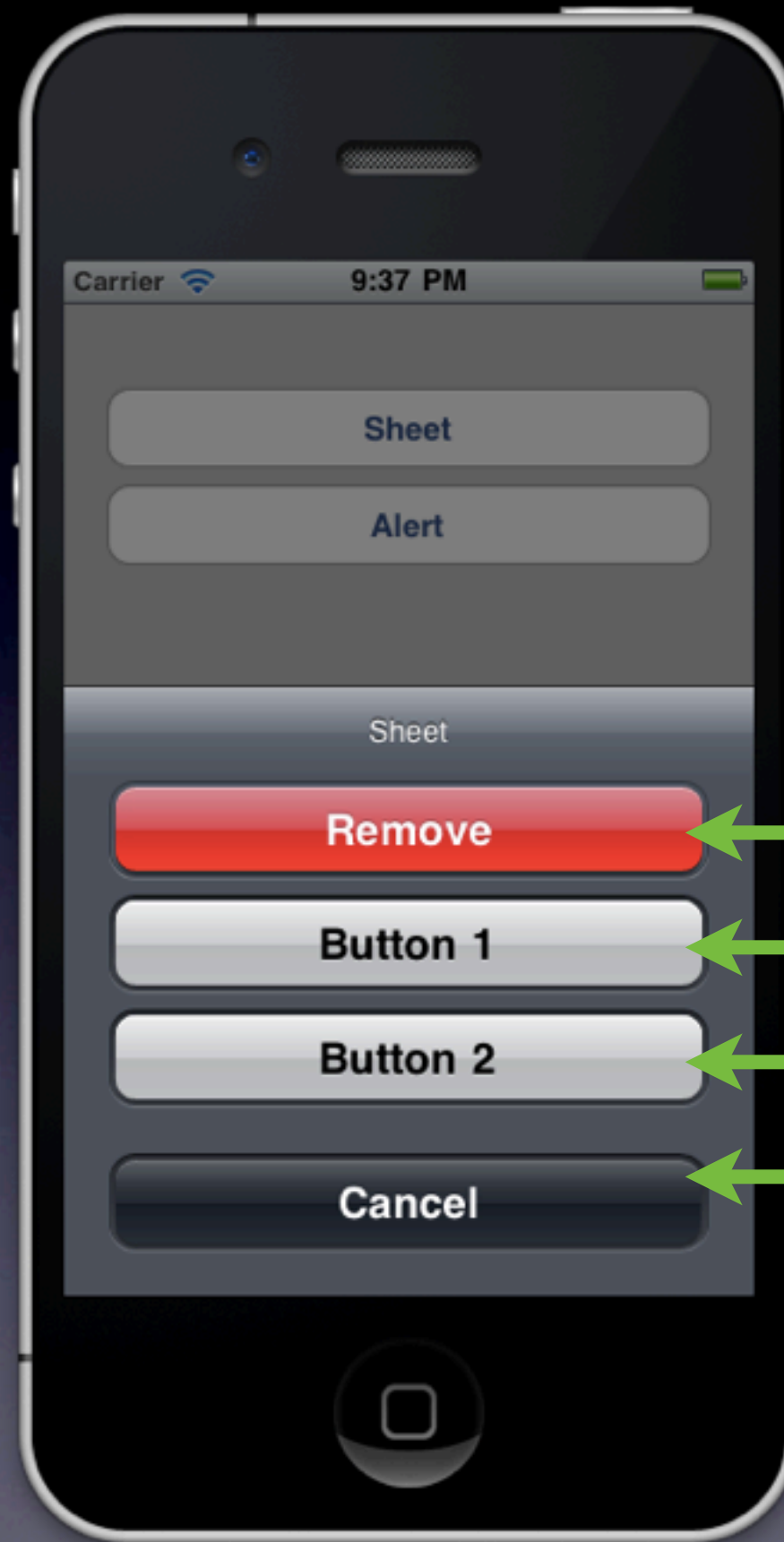
- On iPhone, an action sheet always emerges from the bottom of the screen and hovers over the application's views
- On iPad, an action sheet is always displayed within a popover; it never has full-screen width

Action Sheet

```
UIAlertSheet *sheet = [[UIAlertSheet alloc] initWithTitle:@"Sheet"  
                    delegate:self  
                    cancelButtonTitle:@"Cancel"  
                    destructiveButtonTitle:@"Remove"  
                    otherButtonTitles:@"Button 1", @"Button 2", nil];
```

Action Sheet Delegate

```
-(void)actionSheet:(UIActionSheet *)actionSheet didDismissWithButtonIndex:(NSInteger)buttonIndex
```



Index

0

1

2

3

Universal Apps

Universal Apps

- Apps that work on both iPhone and iPad
- All iPhone apps work for iPod Touches automatically
 - Just watch out for no Internet

Universal Apps

- Checking device type
 - `UI_USER_INTERFACE_IDIOM()`
 - `UIUserInterfaceIdiomPad`
 - `UIUserInterfaceIdiomPhone`

Universal Apps

- Must create new Interface Builder files
- If you have an iPhone app who's main Interface Builder file is MainWindow.xib
 - Create MainWindow~iPad.xib

Deployment

Deployment

- Requires \$99 Developer License
 - Either Personal or Company

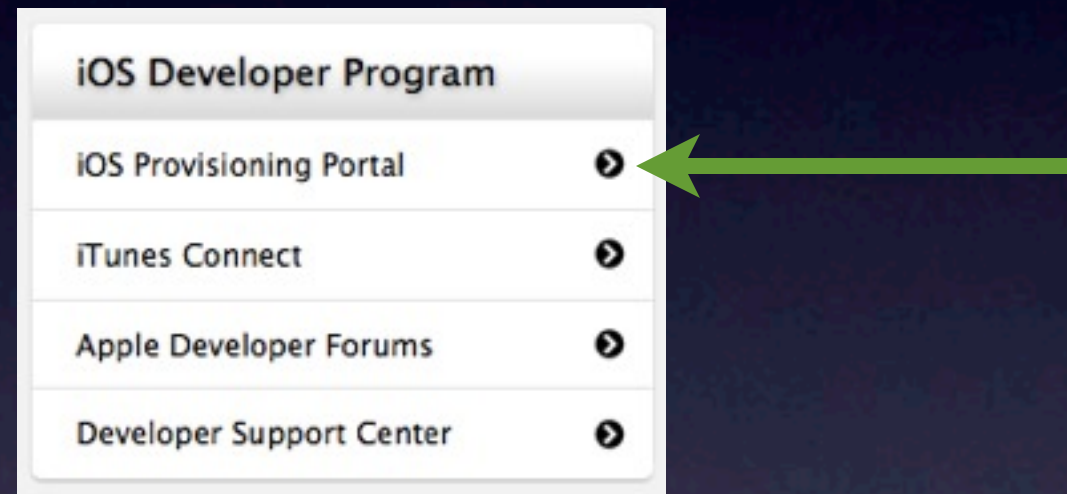
Deployment

- If company account only the team agent can deploy apps

Deployment

- First Step: Certificates
 - iOS Provisioning Portal
- Best advice: Watch videos and follow on-screen help

Deployment



developer.apple.com

Deployment

Portal Resources

 [Program User Guide](#)

 [Program User Guide \(JP\)](#)

How-To's

 [Obtaining your Certificate](#)
 [View video](#)

 [Assigning Devices](#)
 [View video](#)

 [Creating your App IDs](#)
 [View video](#)

 [Creating Provisioning Profiles](#)
 [View video](#)

Deployment

- Certificates
 - All iOS apps must be signed before they can be run on an Apple device
 - Even for testing

Deployment

- Certificate Process
 - Generate a Certificate Signing Request
 - Submit a Certificate Signing Request for approval
 - Download and Install Development Certificates
 - Backup your Private Key

Deployment

Provisioning Portal :

Go to iOS Dev Center

Home

Certificates

Devices

App IDs

Provisioning

Distribution

Development

Distribution

History

How To

Current Distribution Certificate

Name	Expiration Date	Provisioning Profiles	Status	Actions
 	Jun 27, 2012		Issued	Download Revoke

*If you do not have the WWDR intermediate certificate installed, [click here to download now.](#)

Deployment

Provisioning Portal : Go to iOS Dev Center

Home
Certificates
Devices
App IDs
Provisioning
Distribution

Development Distribution History **How To**

Obtaining your iOS Development Certificate

Test Application



```
graph LR; A[Device ID# & App ID#] --> B[Configure Profile]; B --> C[PROV]; C --> D[Download & Install]; D --> E[Build & Go]; E --> F[iPhone]
```

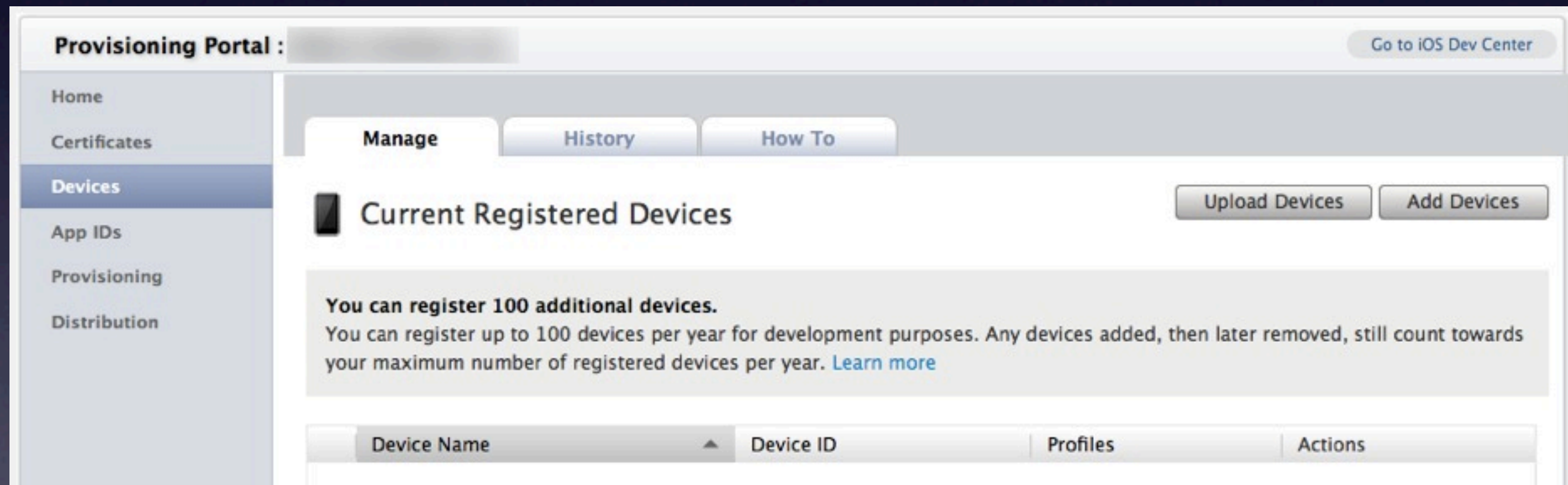
✓ Development Certificate
Device ID
App ID

The diagram illustrates the process of obtaining an iOS Development Certificate. It begins with a device and app ID, followed by configuring a profile, downloading and installing the provisioning profile (PROV), and finally building and going to the app on a physical device.

Deployment

- Add Devices
 - All devices have a UDID (Unique Device Identifier)
 - Only needed for testing or ad-hoc releasing

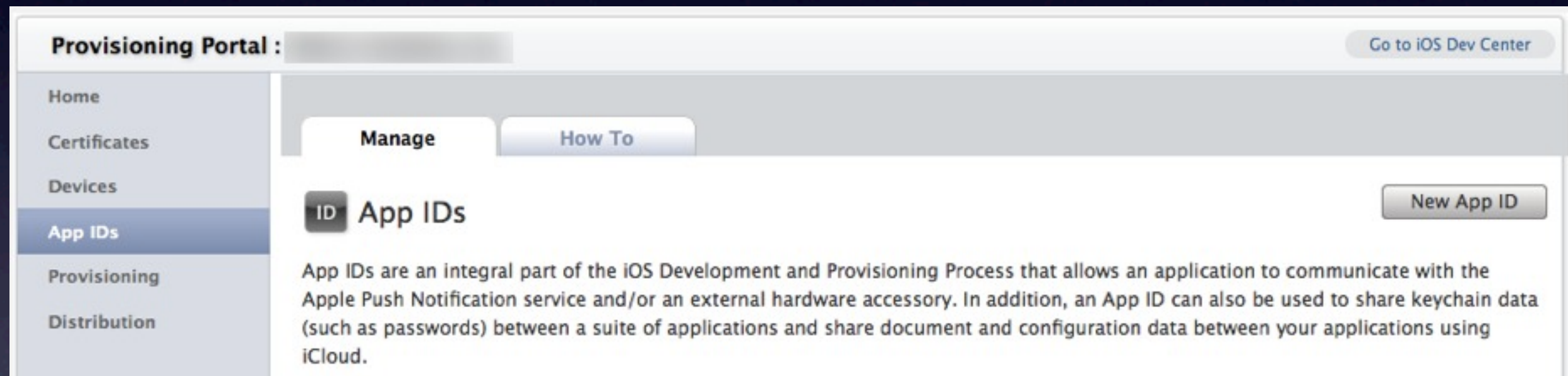
Deployment



Deployment

- Add your app
 - Name
 - Bundle Identifier
 - com.domainname.appname
 - com.business.app

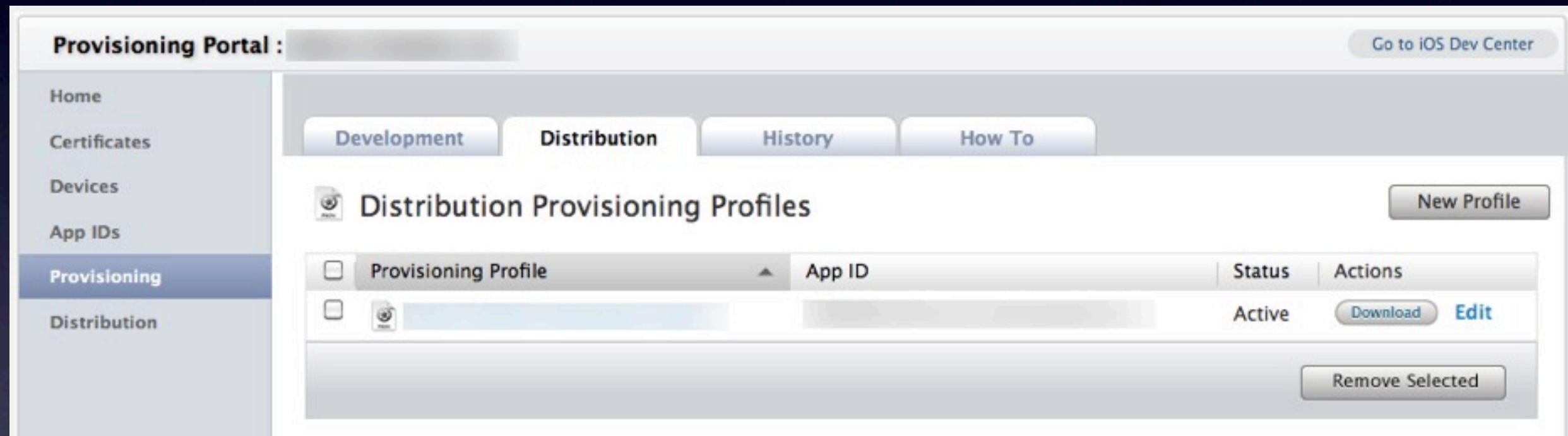
Deployment



Deployment

- Generate Provisioning Profile
 - Used for testing on devices
 - Need a provisioning profile for each app
 - Limited to devices we previously setup
 - A device can contain multiple provisioning profiles

Deployment



Deployment

- itunesconnect.apple.com

Deployment



Sales and Trends

Preview or download your daily and weekly sales information here.



Contracts, Tax, and Banking

Manage your contracts, tax, and banking information.



Payments and Financial Reports

View and download your monthly financial reports and payments.



Manage Users

Set access permissions and email settings for users of your iTunes Connect account.



Manage Your Applications

Add, view, and manage your applications in the iTunes Store.



Contact Us

Having a problem uploading your application? Can't find a Finance Report? Use our Contact Us system to find an answer to your question or to generate a question to an iTunes Rep



Download the Developer Guide.



FAQs Review our answers to common inquiries.

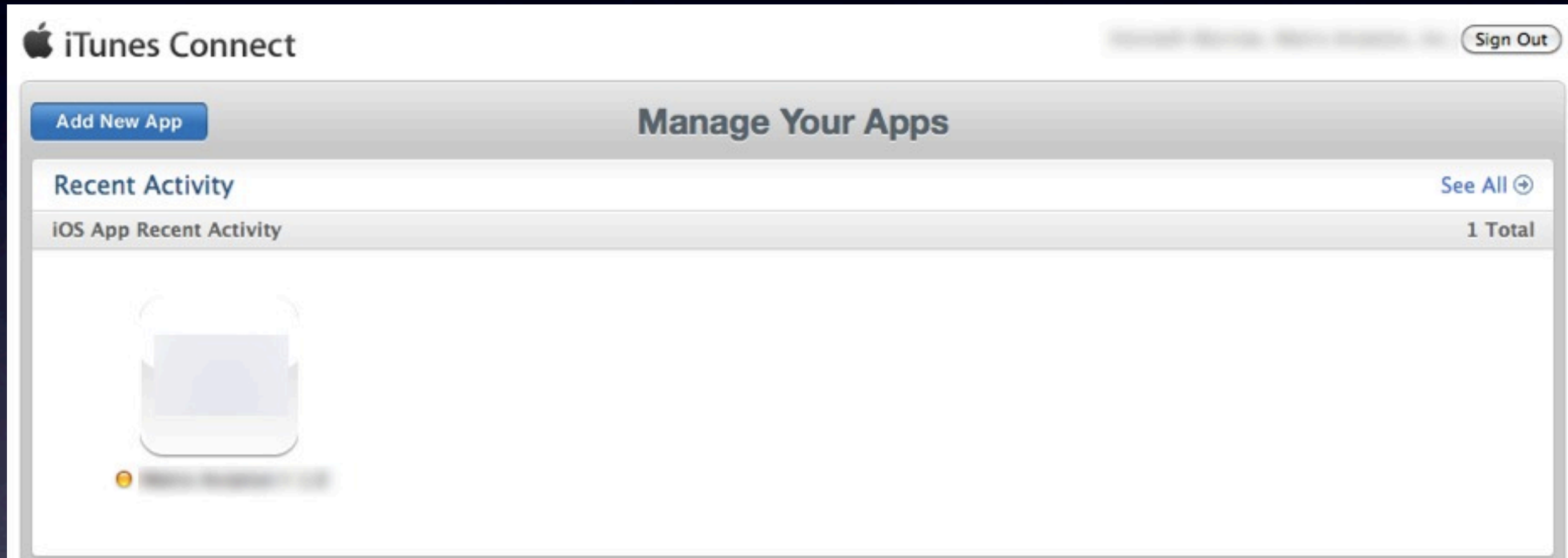
Deployment



Manage Your Applications

Add, view, and manage your applications in the iTunes Store.

Deployment



Deployment

App Information

Identifiers SKU 01 Bundle ID  Apple ID  Type iOS App	Links View in App Store	Rights and Pricing Manage Game Center Set Up iAd Network
---	---	--

Versions

Current Version  View Details	Version 1.0 Status  Waiting For Review Date Created 28 June 2011
---	--

[Done](#)

Deployment

- Uploading your app
 - Set Xcode to build for “Device” in “Release” mode
 - Build your application
 - Look in the Build menu

Deployment

- After building your app
 - Compress it
 - Right click 'Compress ...'
- Your application is technically a folder

Deployment

- Run “Application Loader” app
 - Located in /Developer/Applications/Utilities
 - Login with team agent credentials
 - Should be given an option to select which app to upload

Deployment

- After selecting which app you will be asked to upload your .zip file you created earlier
- Thats it!

Deployment

- Sit back and wait for approval

Deployment

- Initial approval takes longer than update approvals
- To specify you have an update, just login to iTunes Connect and mark app for update
- Re-upload app, no need to generate certificates again

Pros of Approval System

- “Second chance testing”
 - Might catch additional bugs
- Keeps your devices safe
 - No malware

Cons of Approval System

- Takes time
 - Sometimes multiple weeks
- Don't know if your app will be rejected till after all your work
 - Best bet is to check the guidelines

Human Interface Guidelines

Human Interface Guidelines

- Take them to heart
- <http://developer.apple.com/library/IOS/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>

Aesthetic Integrity

- Aesthetic integrity is not a measure of how beautiful an application is
- It's a measure of how well the appearance of the app integrates with its function

Aesthetic Integrity

- An app that enables a productive task generally keeps decorative elements subtle and in the background, while giving prominence to the task by providing standard controls and behaviors

Aesthetic Integrity

- An app that encourages an immersive task, such as a game, users expect a beautiful appearance that promises fun and encourages discovery

Consistency

- Is the application consistent with iOS standards?
- Does it use system-provided controls, views, and icons correctly?
- Does it incorporate device features in a reliable way?

Consistency

- Is the application consistent within itself?
- Does text use uniform terminology and style?
- Do the same icons always mean the same thing?

Direct Manipulation

- When people directly manipulate onscreen objects instead of using separate controls to manipulate them, they're more engaged with the task and they more readily understand the results of their actions

Direct Manipulation

- iOS users enjoy a heightened sense of direct manipulation because of the Multi-Touch interface
- Using gestures gives people a greater affinity for, and sense of control over, the objects they see onscreen, because they're able to touch them without using an intermediary, such as a mouse

Feedback

- Feedback acknowledges people's actions and assures them that processing is occurring
- People expect immediate feedback when they operate a control, and they appreciate status updates during lengthy operations

Metaphors

- When virtual objects and actions in an application are metaphors for objects and actions in the real world, users quickly grasp how to use the app
- The classic example of a software metaphor is the folder

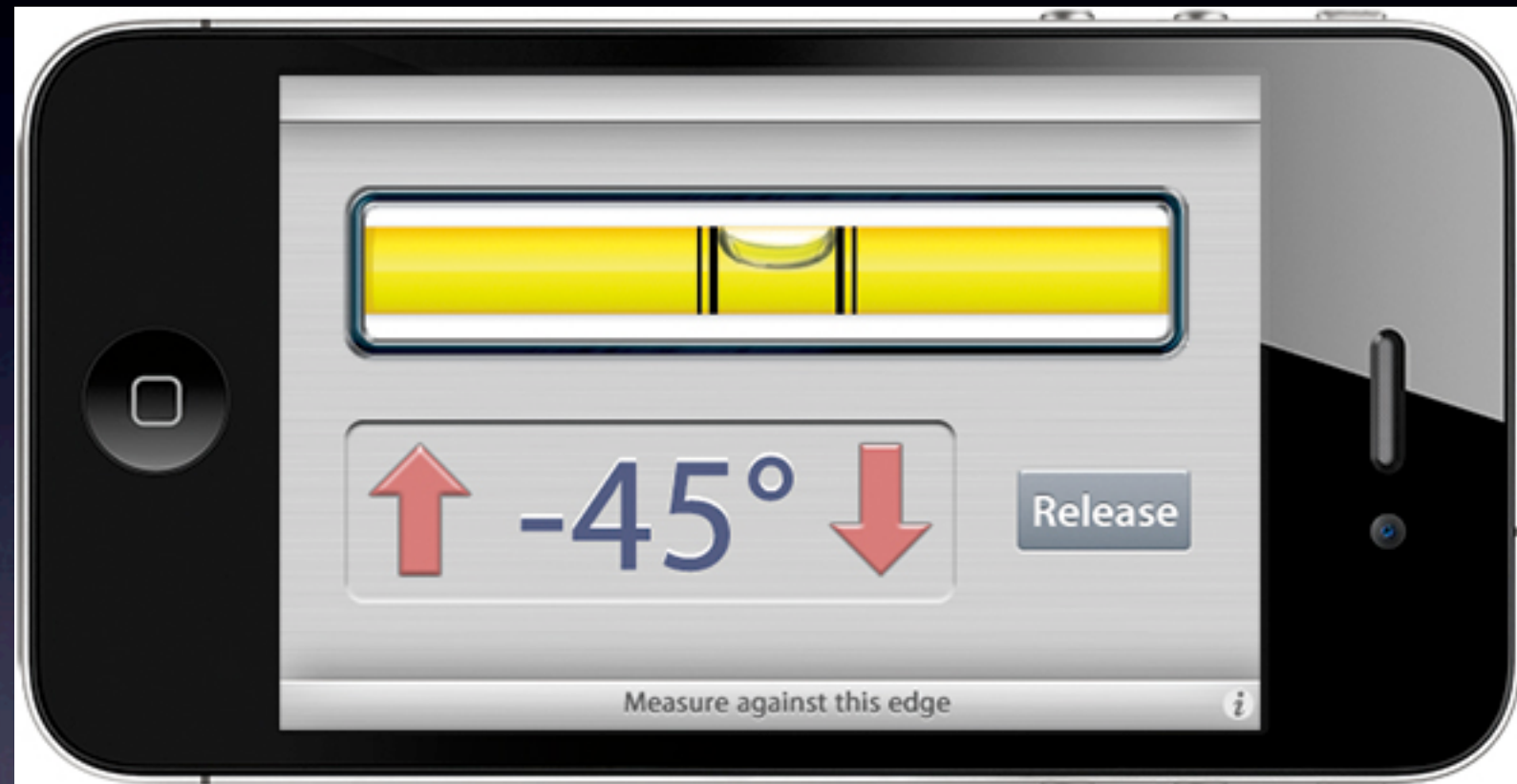
User Control

- People, not applications, should initiate and control actions
- Although an application can suggest a course of action or warn about dangerous consequences, it's usually a mistake for the app to take decision-making away from the user

User Control

- Users feel more in control of an app when behaviors and controls are familiar and predictable
- And, when actions are simple and straightforward, users can easily understand and remember them





Focus on the Primary Task

- When an iOS app establishes and maintains focus on its primary task, it is satisfying and enjoyable to use
- To maintain that focus, you need to determine what's most important in each context or screen

Elevate the Content People Care About

- Minimize the number and prominence of controls to decrease their weight in the UI
- Consider subtly customizing controls so that they integrate with your app's graphical style
- Consider fading controls after people have stopped interacting with them for a little while

Think Top Down

- The top of the screen is most visible to people
- People tend to interact with the device by holding the device in the following ways
 - In their nondominant hand (or laying it on a surface), and gesturing with a finger of the dominant hand
 - In one hand, and gesturing with the thumb of the same hand
 - Between their hands, and gesturing with both thumbs

Think Top Down

- Put the most frequently used (usually higher level) information near the top

Give People a Logical Path to Follow

- Make the path through the information you present logical and easy for users to predict
- In most cases, give users only one path to a screen

Make Usage Easy and Obvious

- Strive to make your application instantly understandable to people, because you can't assume that they have the time (or can spare the attention) to figure out how it works

00:28.3

Stop

Lap

lap 3	00:27.7
lap 2	00:29.2
lap 1	00:28.9

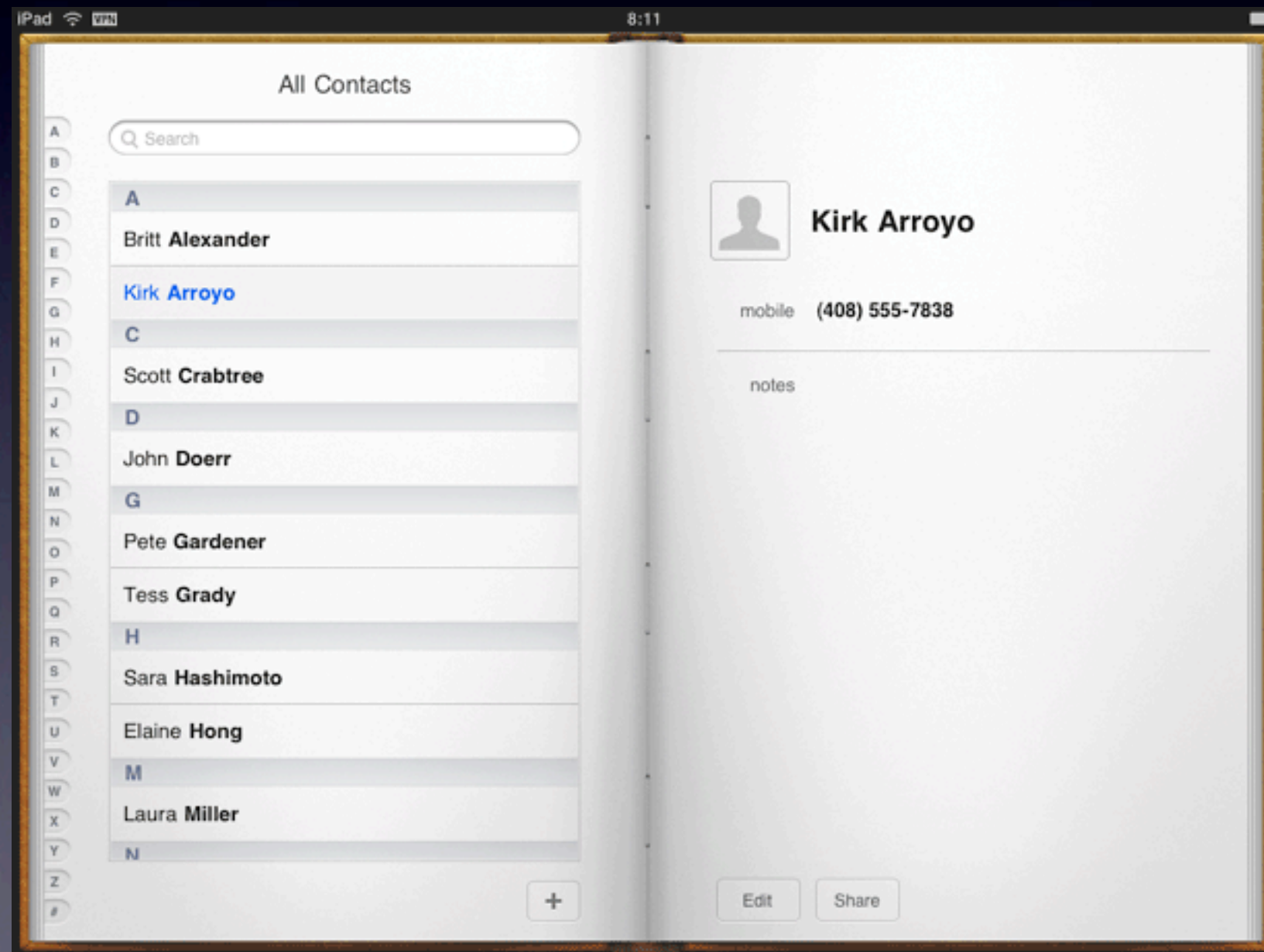
Minimize the Effort Required for User Input

- Inputting information takes time and attention, whether people tap controls or use the keyboard
- If your application requires a lot of user input before anything useful happens, that input slows people down and can discourage them from using your app

Downplay File-Handling Operations

- Although iOS apps can allow people to create and manipulate files, this does not mean that people should have an awareness of a file system on an iOS-based device
- There is no iOS application analogous to the Mac OS X Finder, and people should not be asked to interact with files as they do on a computer

Consider Adding Physicality and Realism



Make Targets Fingertip-Size

- The screen size of iOS-based devices might vary, but the average size of a fingertip does not
- Give tappable elements in your application a target area of about 44 x 44 points

iPod



12:38 PM



235

mc

m+

m-

mr

C

$\pm/\text{--}$

$\div$

$\times$

7

8

9

$-$

4

5

6

$+$

1

2

3

0

.

$=$

Ask People to Save Only When Necessary

- People should have confidence that their work is always preserved unless they explicitly cancel or delete it

Start Instantly

- iOS applications should start as quickly as possible so that people can begin using them without delay

Always Be Prepared to Stop

- iOS applications stop when people press the Home button to open a different application or use a device feature, such as the phone
- People don't tap an application close button or select Quit from a menu

Icons

Icons

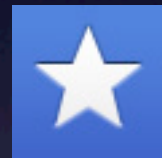
- Application Icon (required)
 - 57x57 / 114 x 114 (high-res)
 - 72 x 72 (iPad)
- App Store Icon (required)
 - 512 x 512

Icons

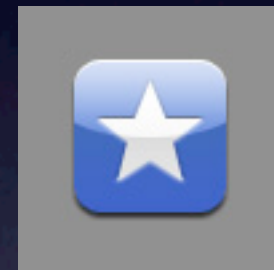
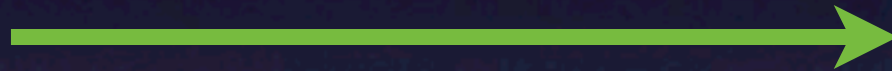
- Spotlight & Settings Icons
 - 29 x 29 / 58 x 58
- iPad Spotlight
 - 50 x 50
- iPad Settings
 - 29 x 29

Icons

- When iOS displays your application icon on the Home screen of a device, it automatically adds the following visual effects
 - Rounded Corners
 - Drop Shadow
 - Reflective Shape



Actual Icon



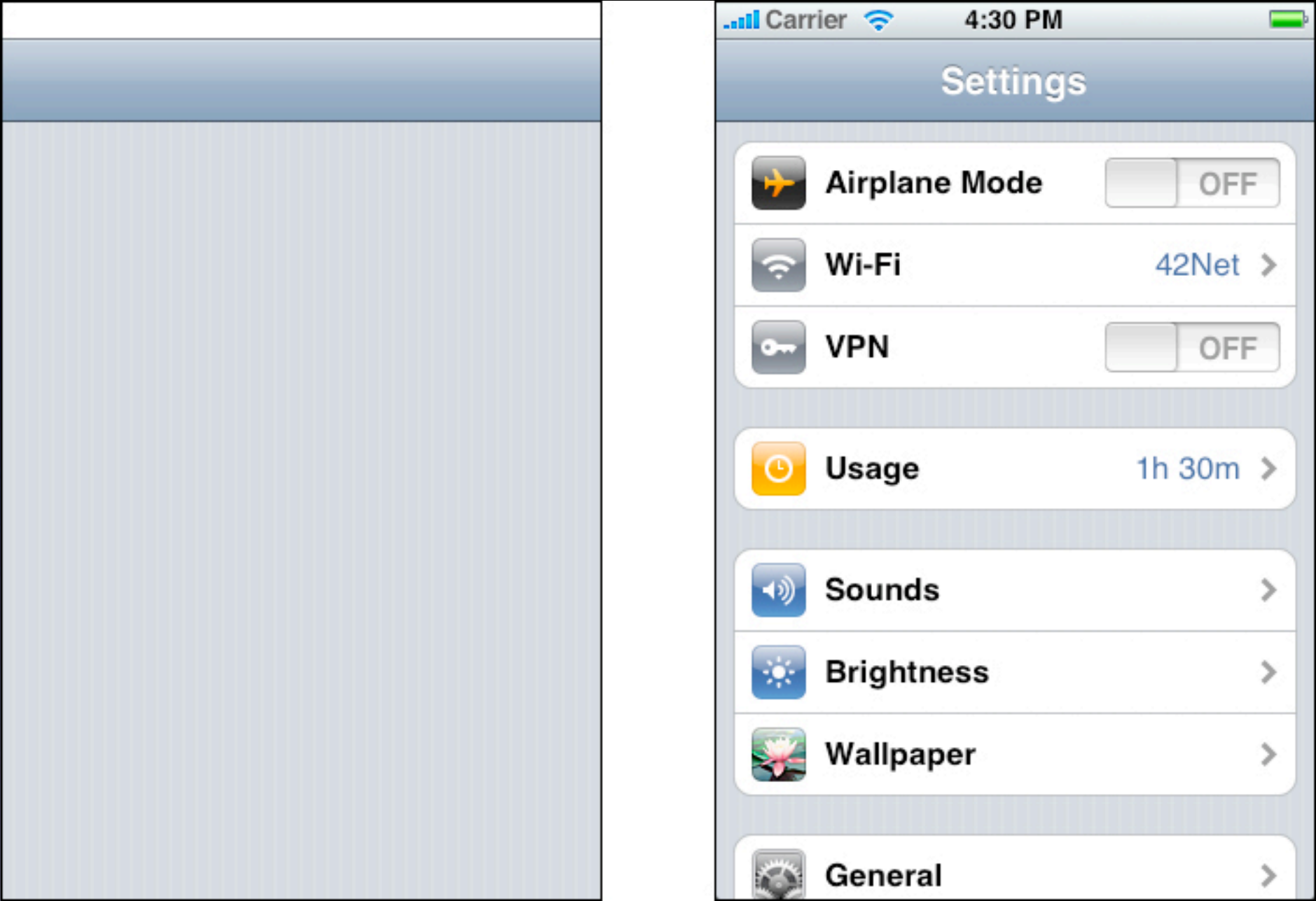
On Device

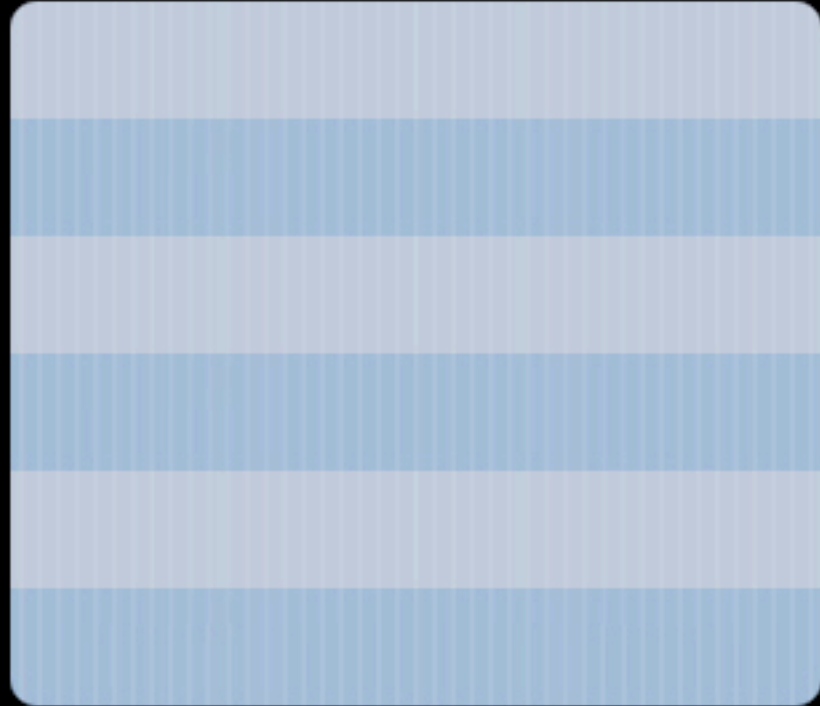
Launch Images

- To enhance the user's experience at application launch, you must provide at least one launch image
- A launch image looks very similar to the first screen your application displays

Launch Images

- iOS displays this image instantly when the user starts your application and until the app is fully ready to use
- As soon as your app is ready for use, your app displays its first screen, replacing the launch placeholder image





Icons & Launch Images

- Easily set these in Xcode 4
- Click on the Xcode project in the groups pane

Further Reading

Three20

- Developed out of Facebook iPhone App
- Open Source (on github)
- Contains collections of iOS UI classes not in standard SDK
- <https://github.com/facebook/three20>

FMDB

- Objective-C Wrapper around SQLite
 - Standard SQLite functions are in C
- <https://github.com/ccgus/fmdb>

ASI HTTP Request

- Objective-C Wrapper around CFNetwork API
- Suitable for interacting with REST-based services (GET / POST / PUT / DELETE)
- <https://github.com/pokeb/asi-http-request>

Cocoa Controls

- <http://cocoacontrols.com>
- UI Component Database for iOS and Mac

iOS Frameworks

- <http://iosframeworks.com>
- Catalog of iOS frameworks and classes

Questions

brentnycum@gmail.com