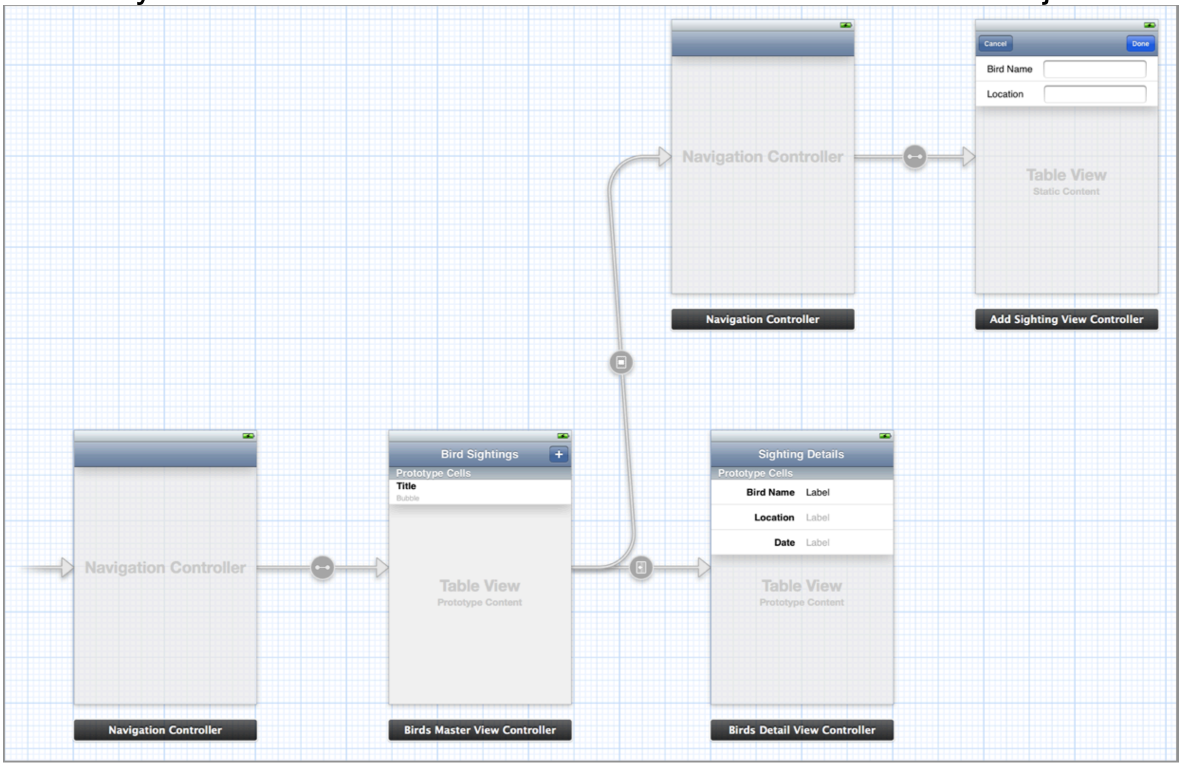
Using View Controllers in Your App

Working with View Controllers in Storyboards

1. A storyboard holds a set of view controllers and associated objects



2. A storyboard may designate one view controller to be the initial view controller.

3. You establish relationships from the initial view controller to other view controllers in the storyboard.
=> If the relationship is a segue, the destination view controller is instantiated when the segue is triggered.
=> If the relationship represents containment, the child view controller is instantiated when its parent is instantiated.
=> If the controller is not the destination or child of another controller, it is never instantiated automatically. You must instantiate it from the storyboard programmatically.

4. To identify a specific view controller or segue inside a storyboard, use interface builder to assign it an identifier string that uniquely identifies it.

> The Main Storyboard Initializes Your App’s User Interface
1. If a main storyboard is declared in information property list file, iOS performs the following steps:

=> It instantiates a window for you.
=> It loads the main storyboard and instantiates its initial view controller.
=> It assigns the new view controller to the window's rootViewController property and then makes the window visible on the screen.

> Segues Automatically Instantiate the Destination View Controller
1. A segue represents a triggered transition that brings a new view controller into your app's user interface.

2. Segues contains a lot of information about transition, including the following:
=> The object that caused the segue to be triggered, known as the sender
=> The source view controller that starts the segue
=> The destination view controller to be instantiated
=> The kind of transition that should be used to bring the destination view controller onscreen
=> An optional identifier string that identifies that specific segue in the storyboard

3. When a segue is triggered, iOS takes the following actions:
=> It instantiates the destination view controller using the attribute values you provided in the storyboard.
=> It gives the source view controller an opportunity to configure the new controller.
=> It performs the transition configured in the segue.

>> Triggering a Segue Programmatically
1. You programmatically trigger the segue by calling the source view controller's performSegueWithIdentifier:sender: method, passing in the identifier for the segue to be triggered.

2. Triggered a segue programmatically

```
- (void)orientationChanged:(NSNotification *)notification
{
    UIDeviceOrientation deviceOrientation = [UIDevice currentDevice].orientation;
    if (UIDeviceOrientationIsLandscape(deviceOrientation) &&
        !isShowingLandscapeView)
    {
        [self performSegueWithIdentifier:@"DisplayAlternateView"
sender:self];
        isShowingLandscapeView = YES;
    }
// Remainder of example omitted.
}
```

> Instantiating a Storyboard’s View Controller Programmatically

1. Instantiating another view controller inside the same storyboard

```
- (IBAction)presentSpecialViewController:(id)sender {
    UIStoryboard *storyboard = self.storyboard;
    SpecialViewController *svc = [storyboard
instantiateViewControllerWithIdentifier:@"SpecialViewController"];

    // Configure the new view controller here.

    [self presentViewController:svc animated:YES completion:nil];
}
```

2. Instantiating a view controller from a new storyboard

```
- (UIWindow*) windowFromStoryboard: (NSString*) storyboardName
                                    onScreen: (UIScreen*) screen
{
    UIWindow *window = [[UIWindow alloc] initWithFrame:[screen
bounds]];
    window.screen = screen;

    UIStoryboard *storyboard = [UIStoryboard
storyboardWithName:storyboardName bundle:nil];
    MainViewController *mainViewController = [storyboard
instantiateInitialViewController];
    window.rootViewController = mainViewController;

    // Configure the new view controller here.

    return window;
}
```

>> Transitioning to a New Storyboard Requires a Programmatic Approach

1. To display a view controller from another storyboard, you must explicitly load the storyboard file and instantiate a view controller inside

it.

> Containers Automatically Instantiate Their Children
1. When a container is instantiated, its children are automatically instantiated at the same time.

Instantiating a Non-Storyboard View Controller

Displaying a View Controller’s Contents Programmatically
1. There are several options for displaying a view controller’s contents:
=> Make the view controller the root view controller of a window.
=> Make it a child of a visible container view controller.
=> Present it from another visible view controller.
=> Present it using a popover (iPad only).

2. In all cases, you assign the view controller to another object—in this case, a window, a view controller, or a popover controller. This object resizes the view controller’s view and adds it to its own view hierarchy so that it can be displayed.

3. Installing the view controller as a window’s root view controller

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
   UIWindow *window = [[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]];
   levelViewController = [[LevelViewController alloc] init];
   window.rootViewController = levelViewController;
   [window makeKeyAndVisible];
}
```