Coordinating Efforts Between View Controllers

By convention, smaller pieces of of content are managed by different view controller classes.

When Coordination Between View Controllers Occurs

1. The lifetime of a view controller has three stages during which it might coordinate with other objects:

=> View controller instantiation
=> During the view controller's lifetime
=> View controller destruction

With Storyboards, a View Controller is Configured When It Is Instantiated

> Configuring the Initial View Controller at Launch

1. The app delegate configures the controller

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    UINavigationController *navigationController =
(UINavigationController*) self.window.rootViewController;
    BirdsMasterViewController * firstViewController =
[[navigationController viewControllers] objectAtIndex:0];

    BirdSightingDataController *dataController =
[[BirdSightingDataController alloc] init];
    firstViewController.dataController = dataController;

    return YES;
}
```

2. Creating the window when a main storyboard is not being used

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]];
    UIStoryboard *storyboard = [UIStoryboard
storyboardWithName:@"MyStoryboard" bundle:nil];
    MainViewController *mainViewController = [storyboard
instantiateInitialViewController];
```

```
    self.window.rootViewController = mainViewController;

    // Code to configure the view controller goes here.

    [self.window makeKeyAndVisible];
    return YES;
}
```

Configuring the Destination Controller When a Segue is Triggered

1. iOS performs the following tasks when a segue is triggered:
=> It instantiates the destination view controller.
=> It instantiates a new segue object that holds all the information for the segue being triggered.
=> It calls the source view controller's prepareForSegue:sender: method, passing in the new segue object and the object that triggered the segue.
=> It calls the segue's perform method to bring the destination controller onto the screen. The actual behavior depends on the kind of segue being performed.
=> It releases the segue object and the segue is complete.

2. Listing below shows an implementation of the prepareForSegue:sender: method.

```
- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:
(id)sender
{
    if ([[segue identifier] isEqualToString:@"ShowSightingsDetails"])
    {
        DetailViewController *detailViewController = [segue
destinationViewController];
        detailViewController.sighting = [self.dataController
objectInListAtIndex:[self.tableView indexPathForSelectedRow].row];
    }

    if ([[segue identifier] isEqualToString:@"ShowAddSightingView"])
    {
        AddSightingViewController *addSightingViewController = [[[segue
destinationViewController] viewControllers] objectAtIndex:0];
        addSightingViewController.delegate = self;
    }
}
```

Using Delegation to Communicate with Other Controllers
1. In a delegate-based model, the view controller defines a protocol for its delegate to implement. The protocol defines methods that are called by the view controller in response to specific actions. The delegate is then responsible for implement these methods.

Guidelines for Managing View Controller Data