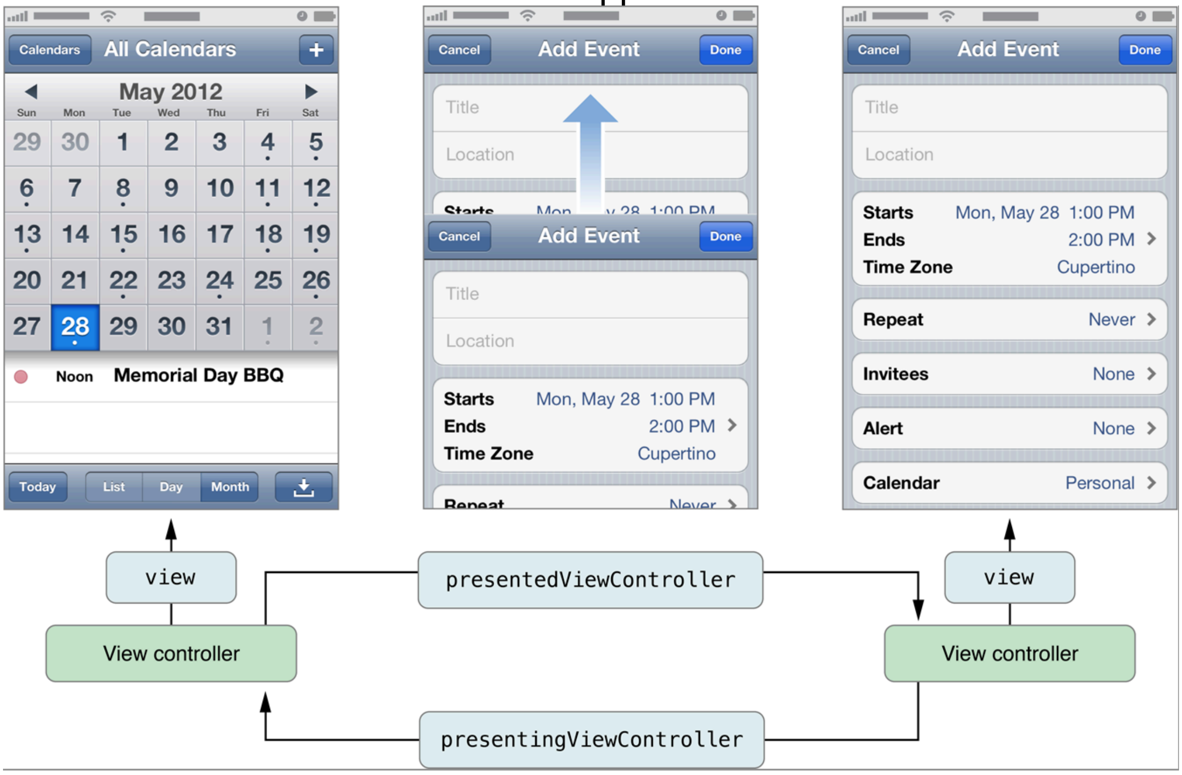
Presenting View Controllers from Other View Controllers

1. The ability to present view controllers is a tool that you have at your disposal for interrupting the current workflow and displaying a new set of views.

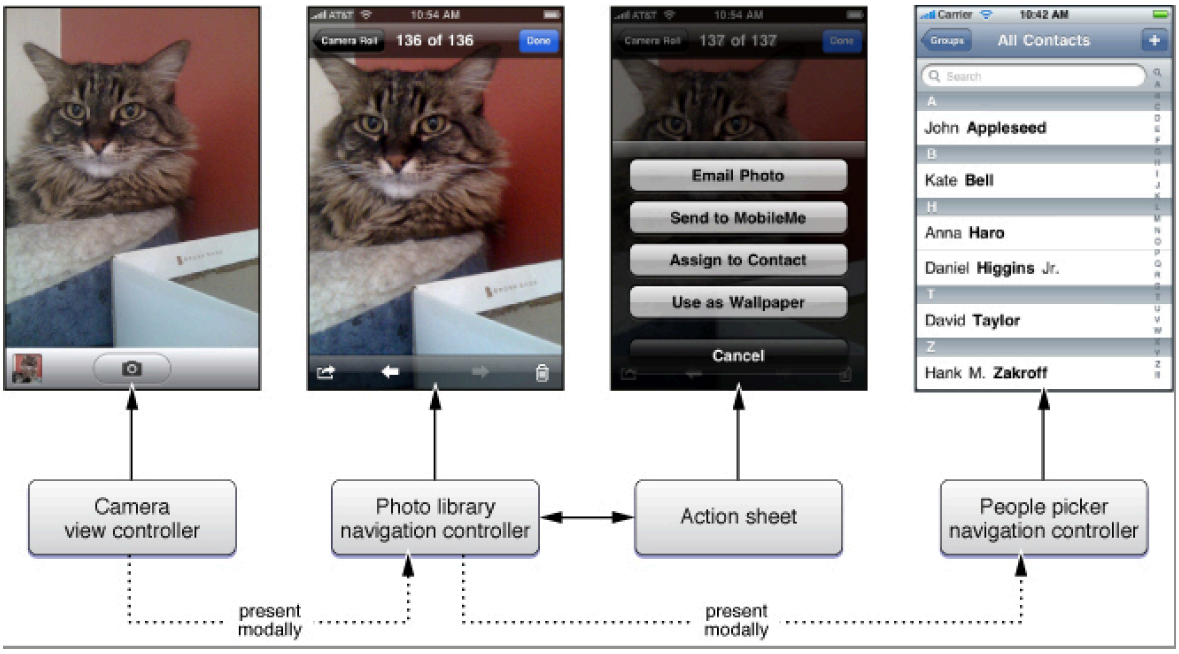
How View Controllers Present Other View Controllers
1. A presented view controller is not a specific subclass of UIViewController. Instead, any view controller can be presented by your app.

2. Specifically, the view controller that did the presenting updates its presentedViewController property to point to its presented view controller. Similarly, the presented view controller updates its presentingViewController property to point back to the view controller that presented it.

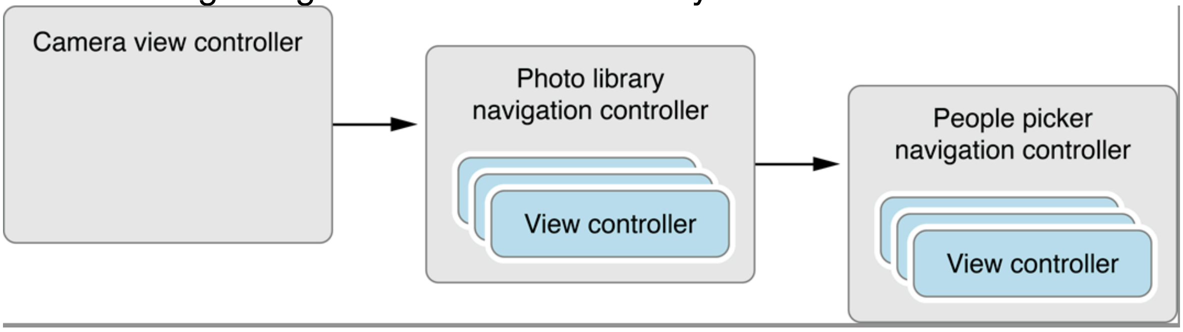
3. Presented views in the Calendar app.



4. Creating a chain of modal view controllers

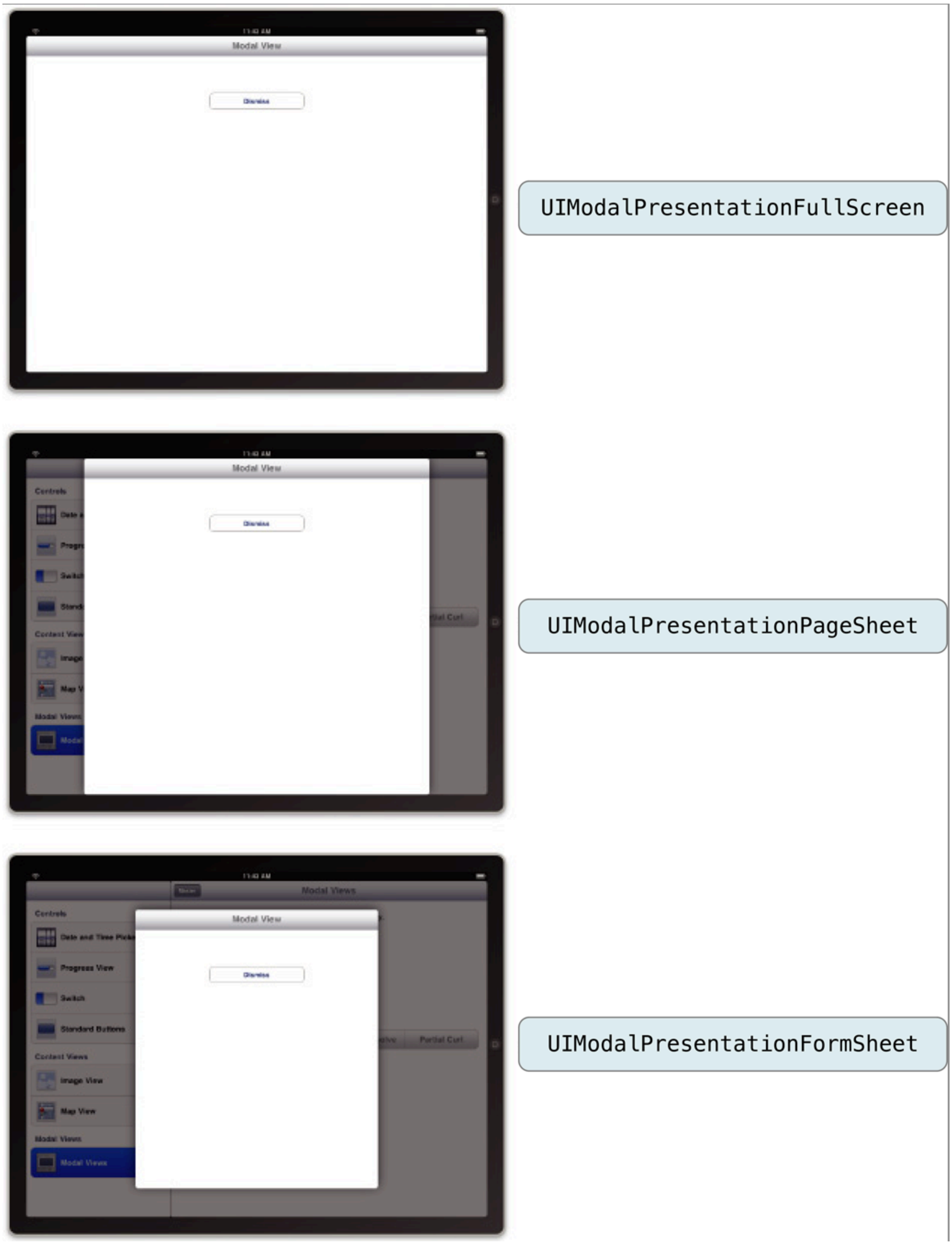


5. Presenting navigation controllers modally


Camera view controller
Photo library navigation controller
View controller
People picker navigation controller
View controller

Presentation Styles for Modal Views

1. iPad presentation styles


Modal View
Dismiss
UIModalPresentationFullScreen
Modal View
Dismiss
UIModalPresentationPageSheet
Modal Views
Modal View
Dismiss
UIModalPresentationFormSheet

Presenting a View Controller and Choosing a Transition Style
1. When a view controller is presented and using a storyboard segue, it is automatically instantiated and presented.

2. If you need to present a view controller programmatically, you need to do the following:

=> Create the view controller you want to present.
=> Set the modalTransitionStyle property of the view controller to the desired value.
=> Assign a delegate object to the view controller. Typically, the delegate is the presenting view controller.
=> Call the presentViewController:animated:completion: method of the current view controller, passing in the view controller you want to present.

3. Transition styles for modal view controllers
=> UIModalTransitionStyleCoverVertical
=> UIModalTransitionStyleFlipHorizontal
=> UIModalTransitionStyleCrossDissolve

4. Presenting a view controller programmatically

```
- (void)add:(id)sender {
   // Create the root view controller for the navigation controller
   // The new view controller configures a Cancel and Done button for the
   // navigation bar.
   RecipeAddViewController *addController = [[RecipeAddViewController
alloc]
                       init];

   // Configure the RecipeAddViewController. In this case, it reports any
   // changes to a custom delegate object.
   addController.delegate = self;

   // Create the navigation controller and present it.
   UINavigationController *navigationController = [[UINavigationController
alloc]
                             initWithRootViewController:addController];
   [self presentViewController:navigationController animated:YES
completion: nil];
}
```

Presentation Contexts Provide the Area Covered by the Presented View Controller
1. The area of the screen used to define the presentation area is determined by the presentation text.

2. When a view controller is presented, iOS searches for a presentation

context.

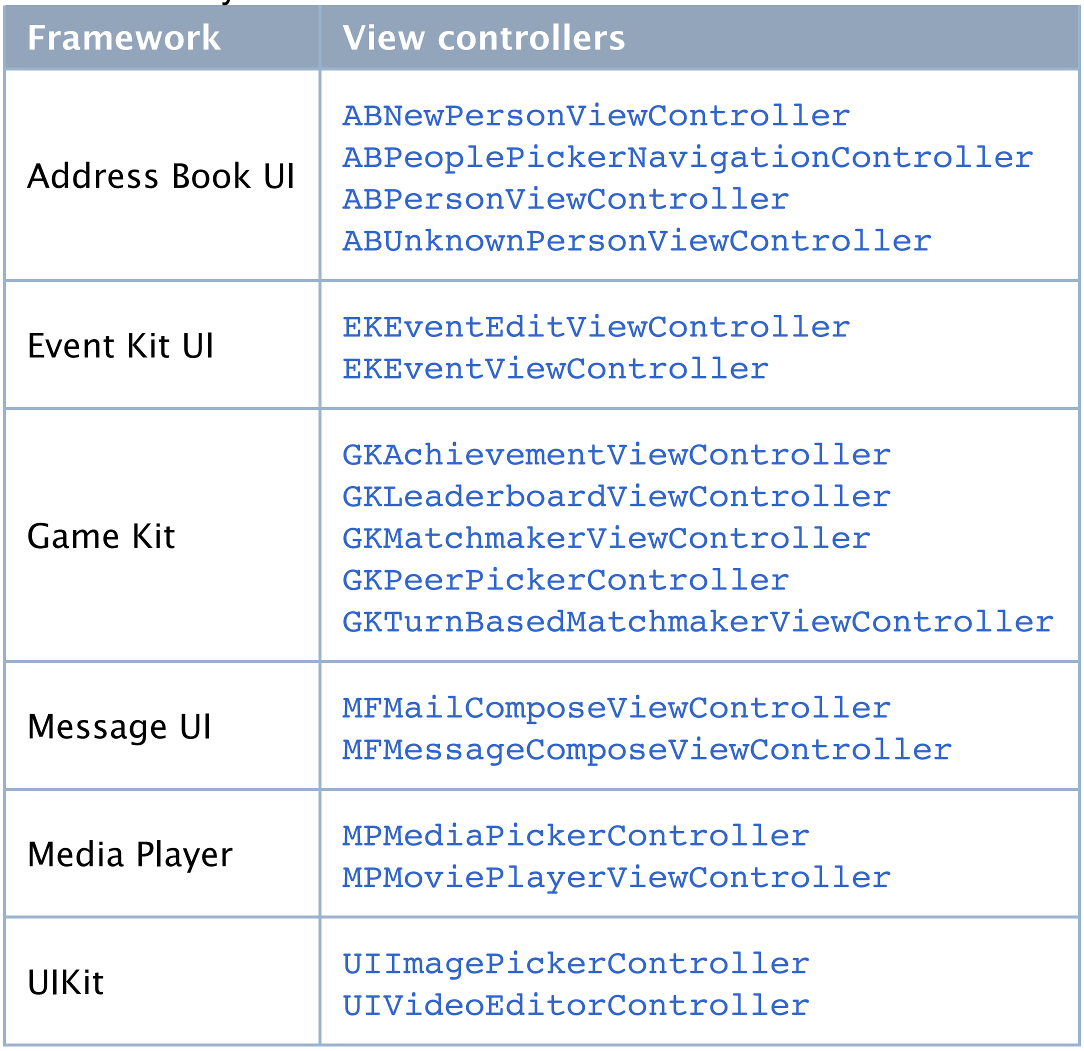
Dismissing a Presented View Controller

Although there are several techniques for notifying the presenting view controller that its presented view controller should be dismissed, the preferred technique is delegation.

Presenting Standard System View Controllers

1. Each system view controller defines a corresponding protocol, whose methods you implement in your delegate object.

2. Standard system view controllers



Framework	View controllers
Address Book UI	ABNewPersonViewController ABPeoplePickerNavigationController ABPersonViewController ABUnknownPersonViewController
Event Kit UI	EKEventEditViewController EKEventViewController
Game Kit	GKAchievementViewController GKLeaderboardViewController GKMatchmakerViewController GKPeerPickerController GKTurnBasedMatchmakerViewController
Message UI	MFMailComposeViewController MFMessageComposeViewController
Media Player	MPMediaPickerController MPMoviePlayerViewController
UIKit	UIImagePickerController UIVideoEditorController