**Objective: GroupMessage App**

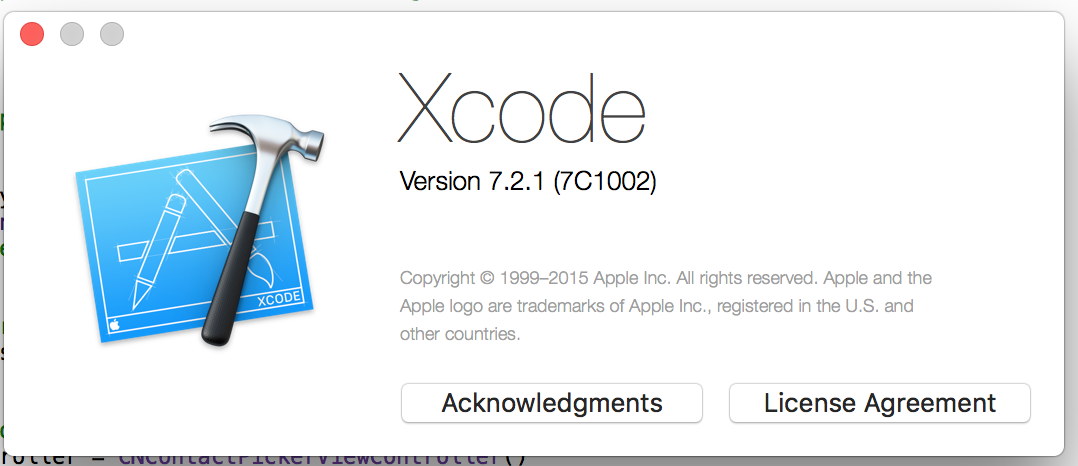
Create a smart app to choose a group of friends from the address book and send a short message. iOS messaging app displays all the users in the address book. **Lets create an app that is smart and simple to filter out users only with phone numbers and that too mobile phone numbers in their address book.**

**Getting Started**

1. Beginners get familiarized with Swift and fundamentals of App development. <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/DevelopiOSAppsSwift/> will provide a very good tutorial.
2. Get to know about predicate programming guide -<https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Predicates/AdditionalChapters/Introduction.html>
3. Familiarize with <https://developer.apple.com/library/prerelease/mac/documentation/Contacts/Reference/CNContact_Class/index.html>

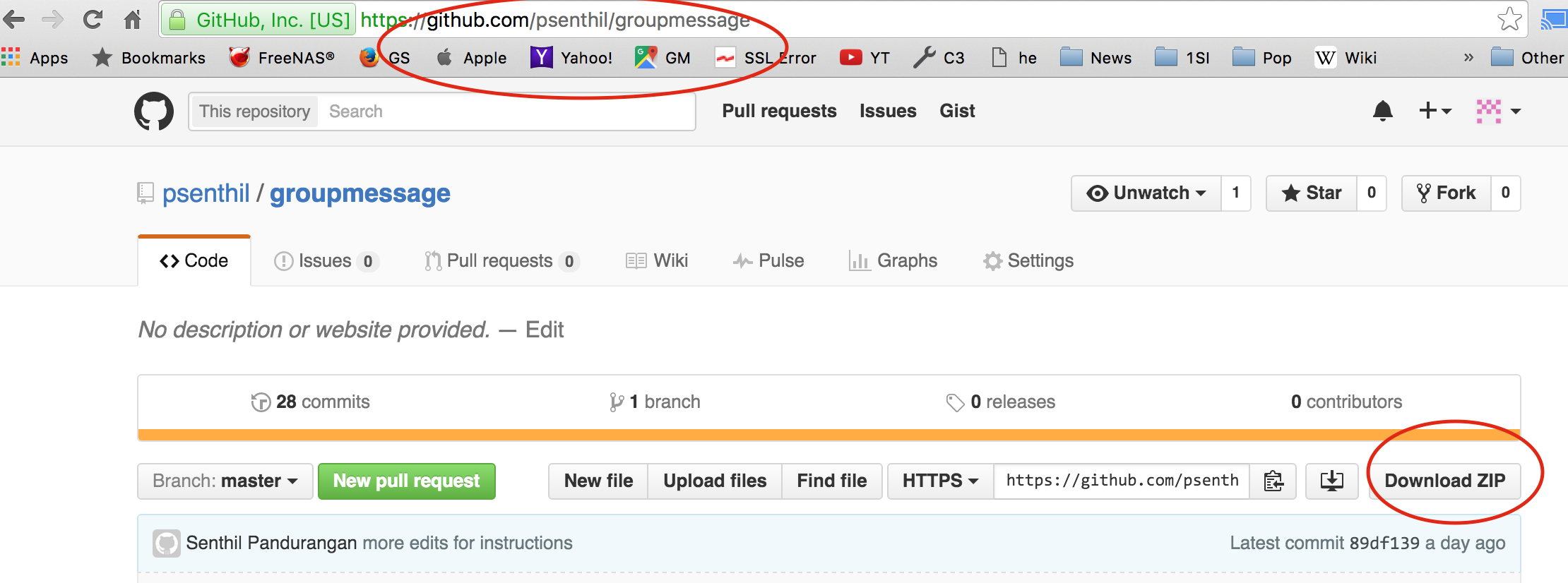
**Instructions**

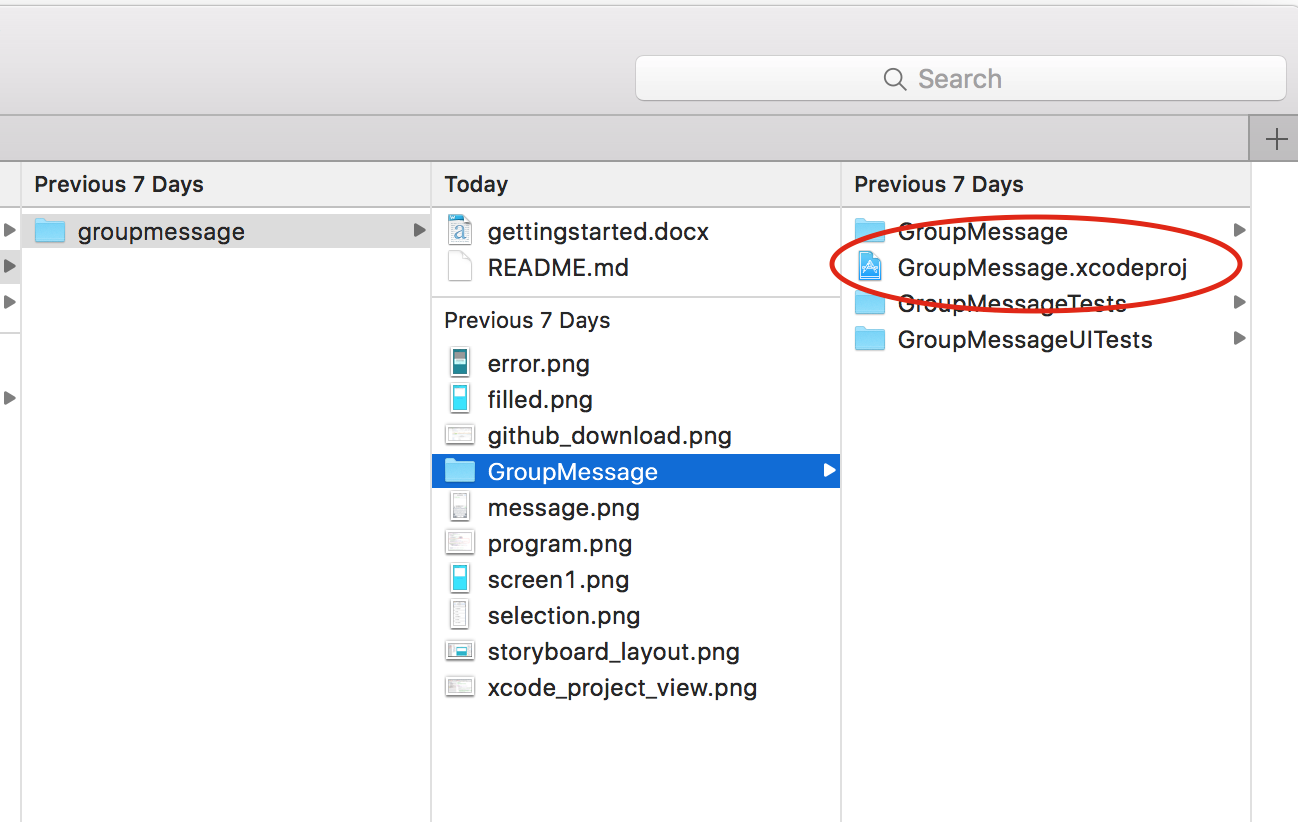
1. Mac OS with Yosemite or El Capitan is needed to build and run this app
2. Validate if you have the Xcode installed. Pressing Command space bar and type Xcode. If XCode shows up, make sure the version is above 7.2. To check the version, after opening up Xcode, XCode > About Xcode should show the following picture



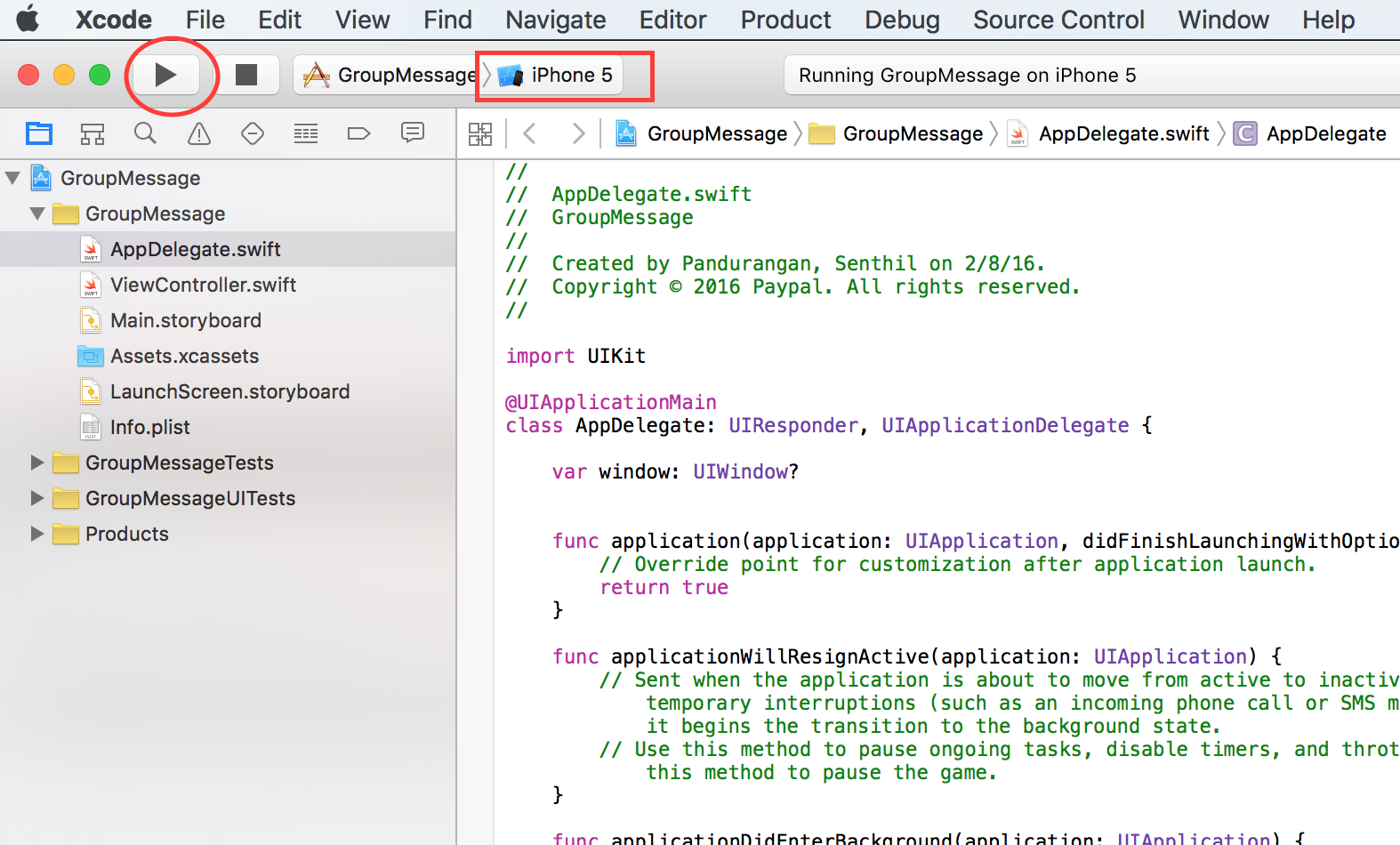
1. The sample code for this project was written using Xcode version 7.2.1. If Xcode is not installed, download and install it from Mac app store <https://itunes.apple.com/us/app/xcode/id497799835?ls=1&mt=12>



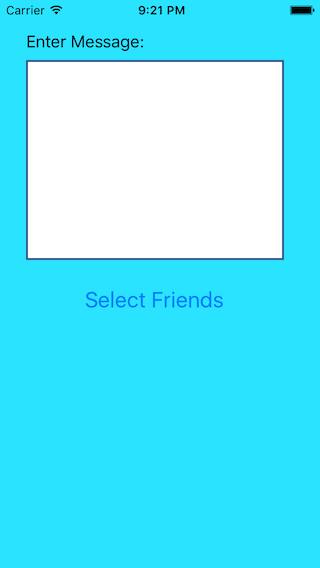
1. Download the project source code from <https://github.com/psenthil/groupmessage/archive/master.zip>  Alternatively you can download the code by opening up Terminal app and use git clone <https://github.com/psenthil/groupmessage.git> to download in your favorite folder.
2. Locate GroupMessage.xcodeproj from the downloaded folder location as shown in the following picture



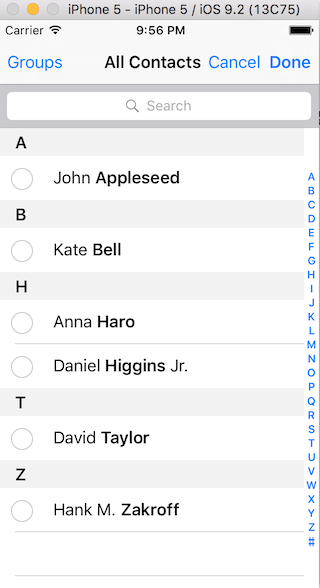
1. Double click on the GroupMessage.xcodeproj file and it should open up project in Xcode
2. Once opened, make sure to select a simulator device and click on Play button as shown following picture



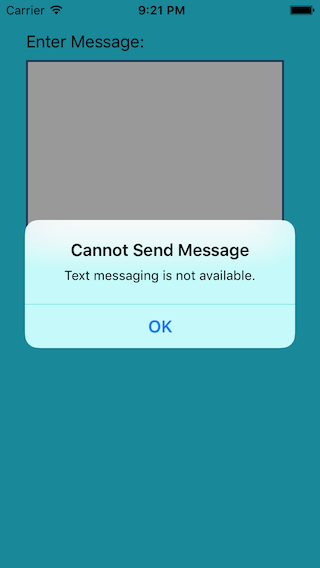
1. It should start the app after installing in the simulator. First run might take a bit longer and the subsequent runs will be faster. It should show following screen

[](https://github.com/psenthil/groupmessage/blob/master/screen1.png)

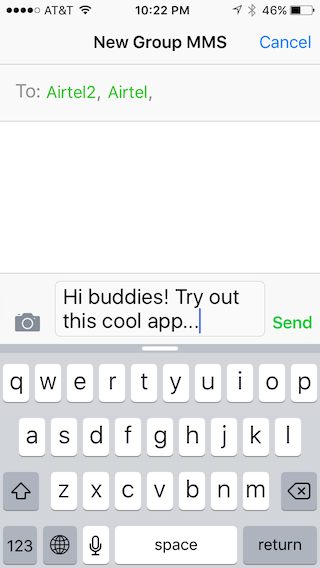
1. Type in the message you wanted to send to your group of friends in the message box and clicking on "Select Friends" button will show the following screen

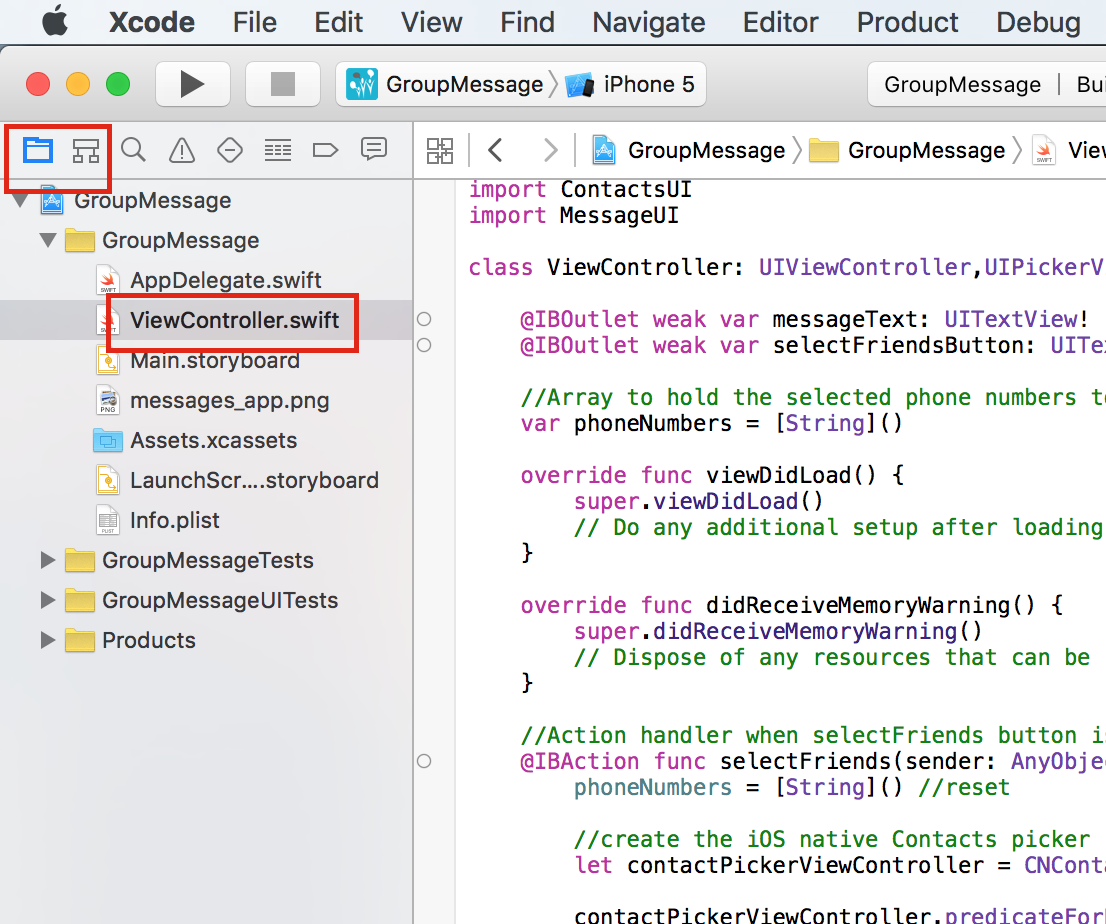


1. **Important Note**: iOS simulators cannot send messages. You will see the following screen



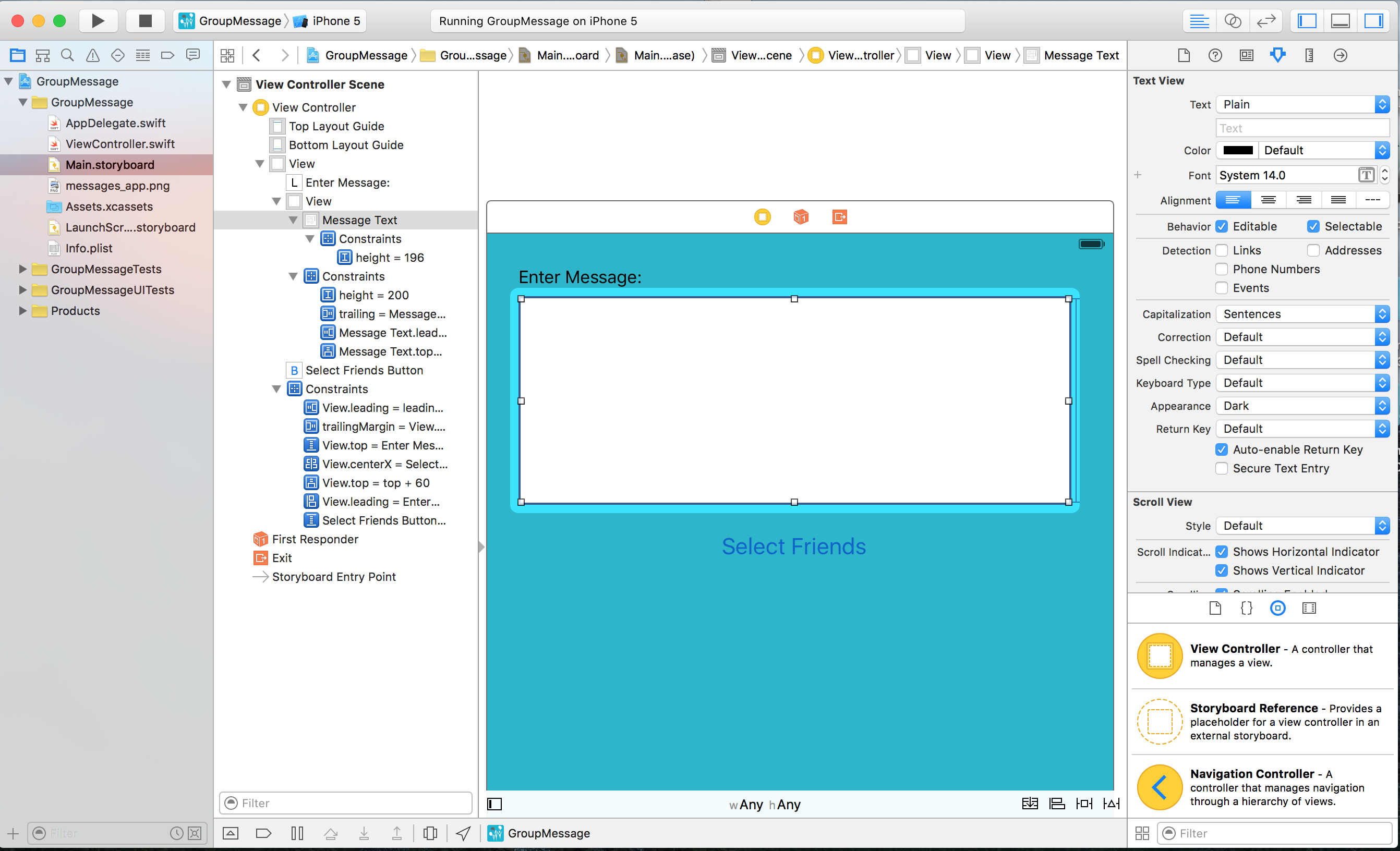
1. Connect your iPhone or iPad device and select the device to install app. This might also take a bit longer for the first time. Once installed follow above steps 8 to 10
2. Select few friends to send the message. Clicking on "Done" button will open up the message view with your selected friends and message populated as below



1. Going back to Xcode and clicking on the ViewController.swift will provide the glimpse of the program 

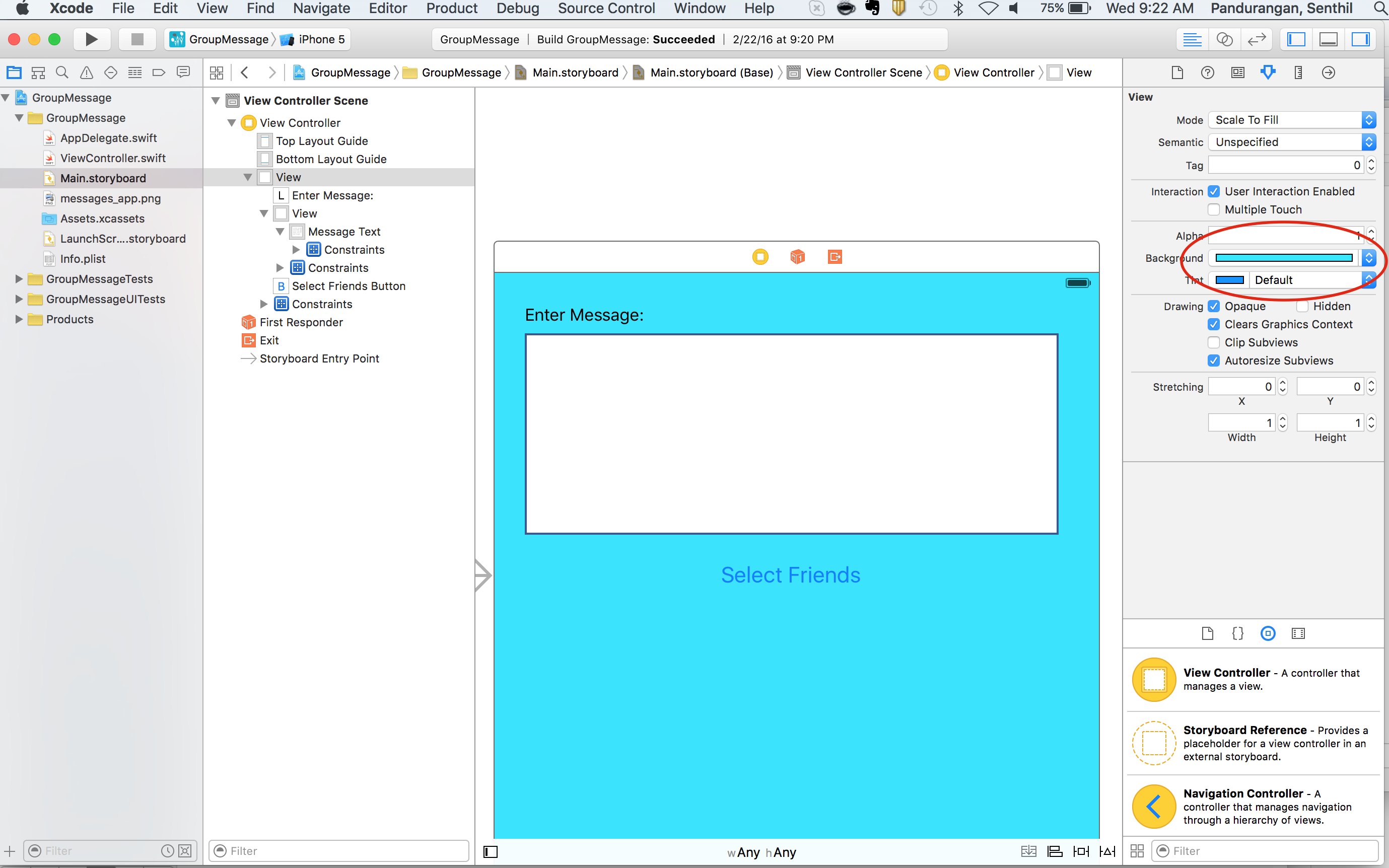


1. Also Main.storyboard will show how the UI components are laid out



1. Have fun and enjoy making changes and playing with it.

**Exercise**

1. Change the “Select Friends” to “Select Buddies” and run the app in Simulator. Double clicking on the “Select friends” will allow to change the label of the button.
2. Change the background color. 
3. Make changes to the app so that you can include the “iPhone” phone types along with “mobile” phone types as well. The changes will be in ViewController.swift

//Function to display list of contacts

//Additionally it filters the contacts with Mobile phone numbers

func contactPicker(picker: CNContactPickerViewController, didSelectContacts contacts: [CNContact]) {

for contact in contacts {

print (contact.givenName, contact.familyName)

for phoneNumber in contact.phoneNumbers {

if(phoneNumber.label.containsString("Mobile") || (phoneNumber.label.containsString("IPhone") ) {

phoneNumbers.append(phoneNumber.valueForKey("value")?.valueForKey("digits") as! String)

}

}

}

print(phoneNumbers)

self.dismissViewControllerAnimated(true, completion: nil) //dismiss the old view

sendSMSMessage("")

}

1. **Extra Credits/Homework:** Make changes to the above application to fall back in sending an email if the contact does not have any mobile phones. This exercise will need little more understanding on

**Additional Resources**

1. <https://developer.apple.com/swift/resources/>
2. <http://www.appcoda.com/ios-contacts-framework/>
3. <http://devstreaming.apple.com/videos/wwdc/2015/223rmo6dv9hxh/223/223_introducing_the_contacts_framework_for_ios_and_os_x.pdf>