



实验 4

创建发送电子邮件、短信和拨电话的应用

实验结构

- » 创建一个应用，用于发送电子邮件、短信和拨电话

实验目标

本实验结束后，你将能够：

- » 创建一个发送电子邮件的应用
- » 创建一个发送短信的应用
- » 创建一个拨电话号码的应用

导论

iPhone 带有一系列内建应用，例如邮件、短信和电话。你可以从应用中使用 iPhone 软件开发工具包（SDK）提供的各种应用编程接口（API），在程序上调用这些应用。

实验：创建发送电子邮件、短信和拨电话的应用

背景

你要创建一个叫作 **emailer** 的应用，让用户能在该应用内发送电子邮件、短信和拨打电话号码。

要创建这一应用，你需要执行下面这些任务：

- 在 Xcode 中使用 UIObjects 创建应用
- 从应用内调用邮件功能
- 从应用内调用短信功能
- 从应用内调用电话功能

实验准备

要执行这些任务，你需要有：

- 带有 Xcode 的 Mac
- 苹果开发者帐户

实验推荐解决方案

1. 打开 Xcode 并选择 **Create a new Xcode project** 选项，如图 1 所示：

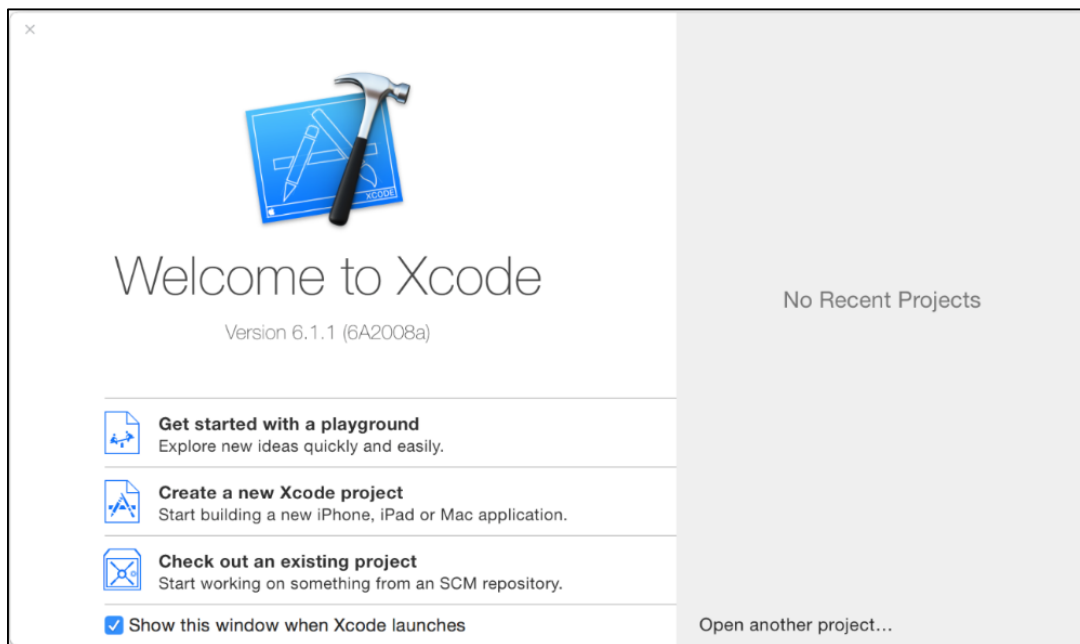


图 1: 选择创建新 Xcode 项目

2. 选择 **Single View Application** 模板，并点 **Next**，如图 2 所示：

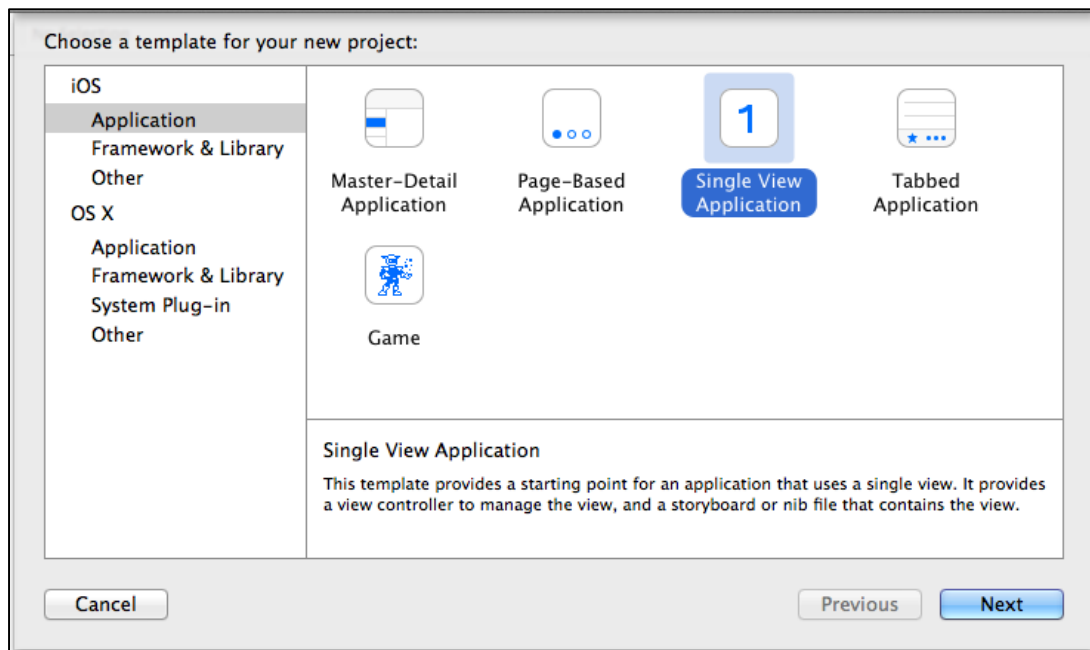


图 2: 选择 Single View Application 模板

3. 下一个界面上, 将项目命名为 **emailer**, 选择 **Objective-C** 作为编程语言, 然后点 **Next**, 如图 3 所示:

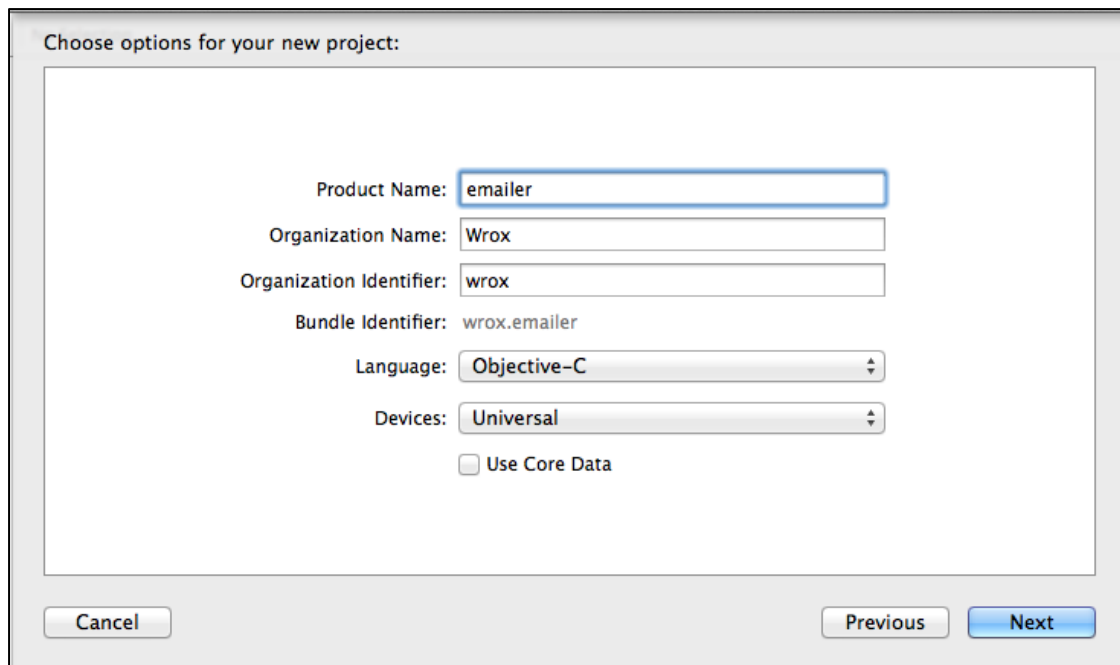


图 3: 将应用命名为 emailer

4. 选择项目保存位置, 然后点 **Create**, 如图 4 所示:

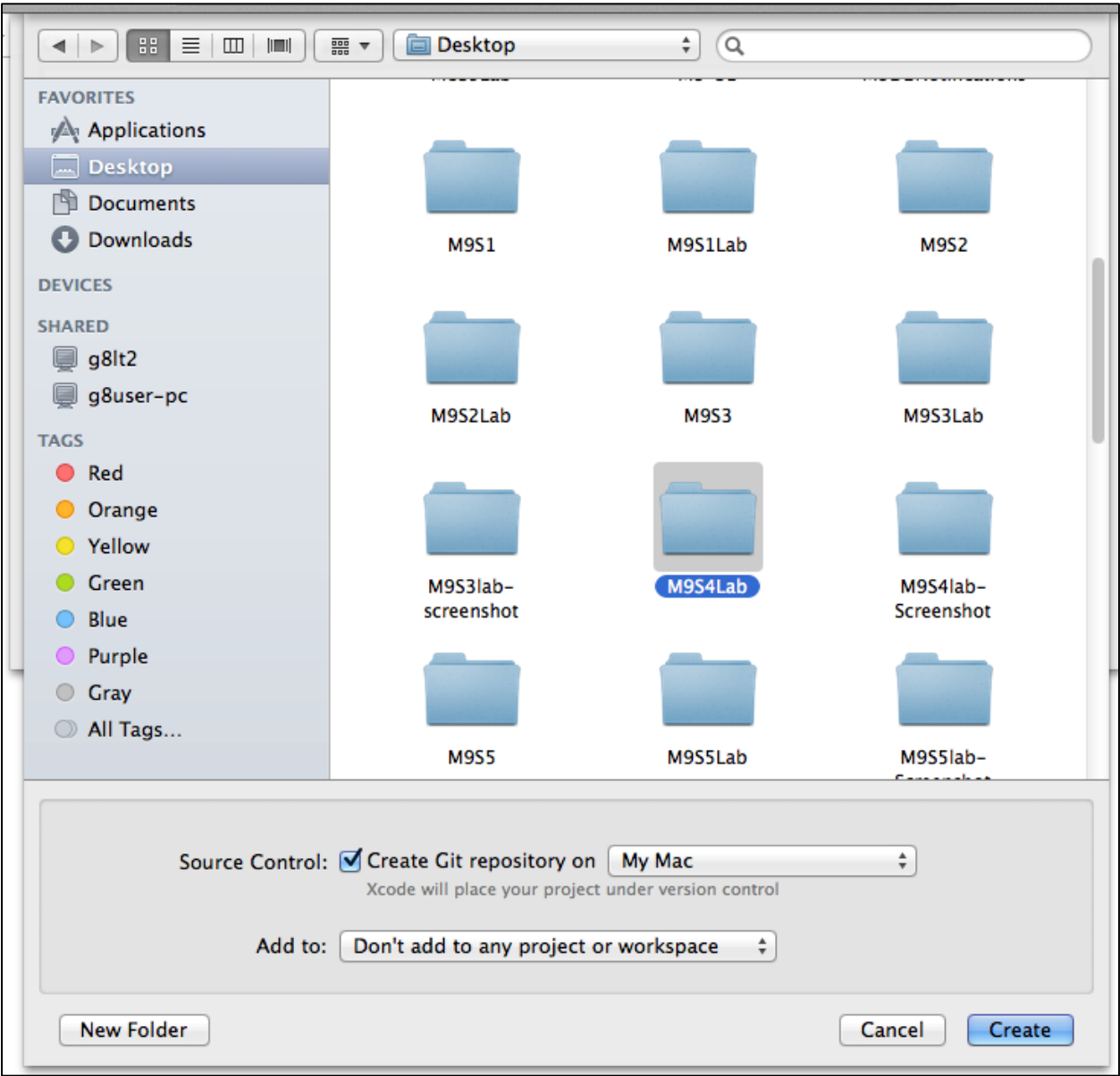


图 4: 保存 emailer 项目

5. 在 Xcode 中打开 emailer 项目。Xcode 5 中，你会得到三个核心框架：UIKit、Foundation 和 CoreGraphics，但 Xcode 6 中则不是如此，你需要手动添加这些库。你还需要在 ViewController.h 文件中添加 MFMailComposeViewController 到 emailer 项目，用于发送电子邮件和短信。
6. 要添加新库，选择 **General** 选项卡，并点 **Linked Frameworks and Libraries** 部分的+图标，如图 5 所示。然后选择并添加需要的库，如图 6 所示：

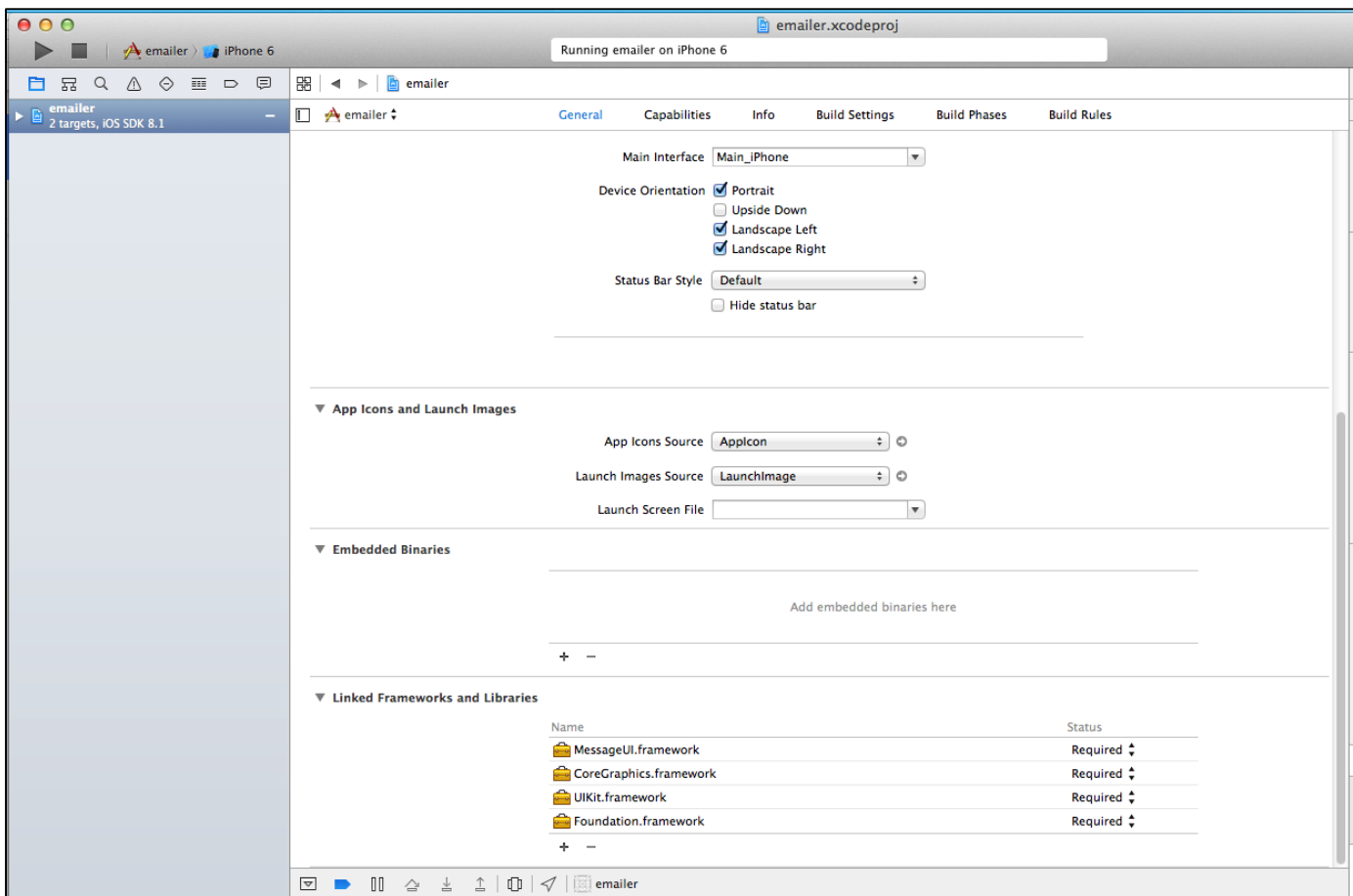


图 5: 点 Linked Frameworks and Libraries 中的+图标

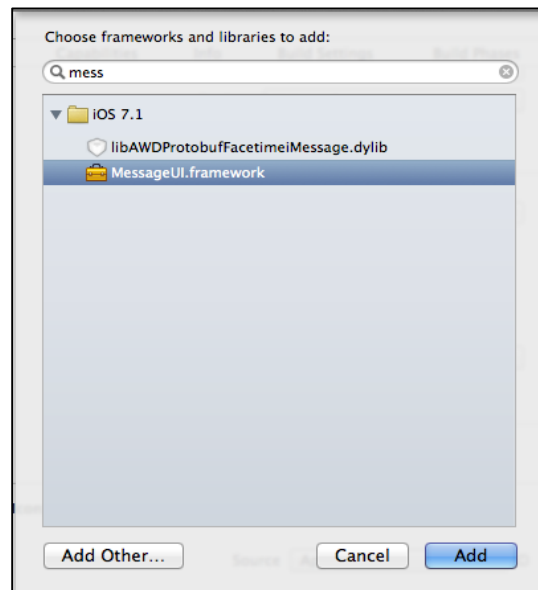


图 6: 选择用于电子邮件和短信的库

7. 选择 Main.storyboard 文件并设计应用的用户界面，过程中需要用到实用工具面板中的 UILabel、按钮等对象，如图 7 所示：

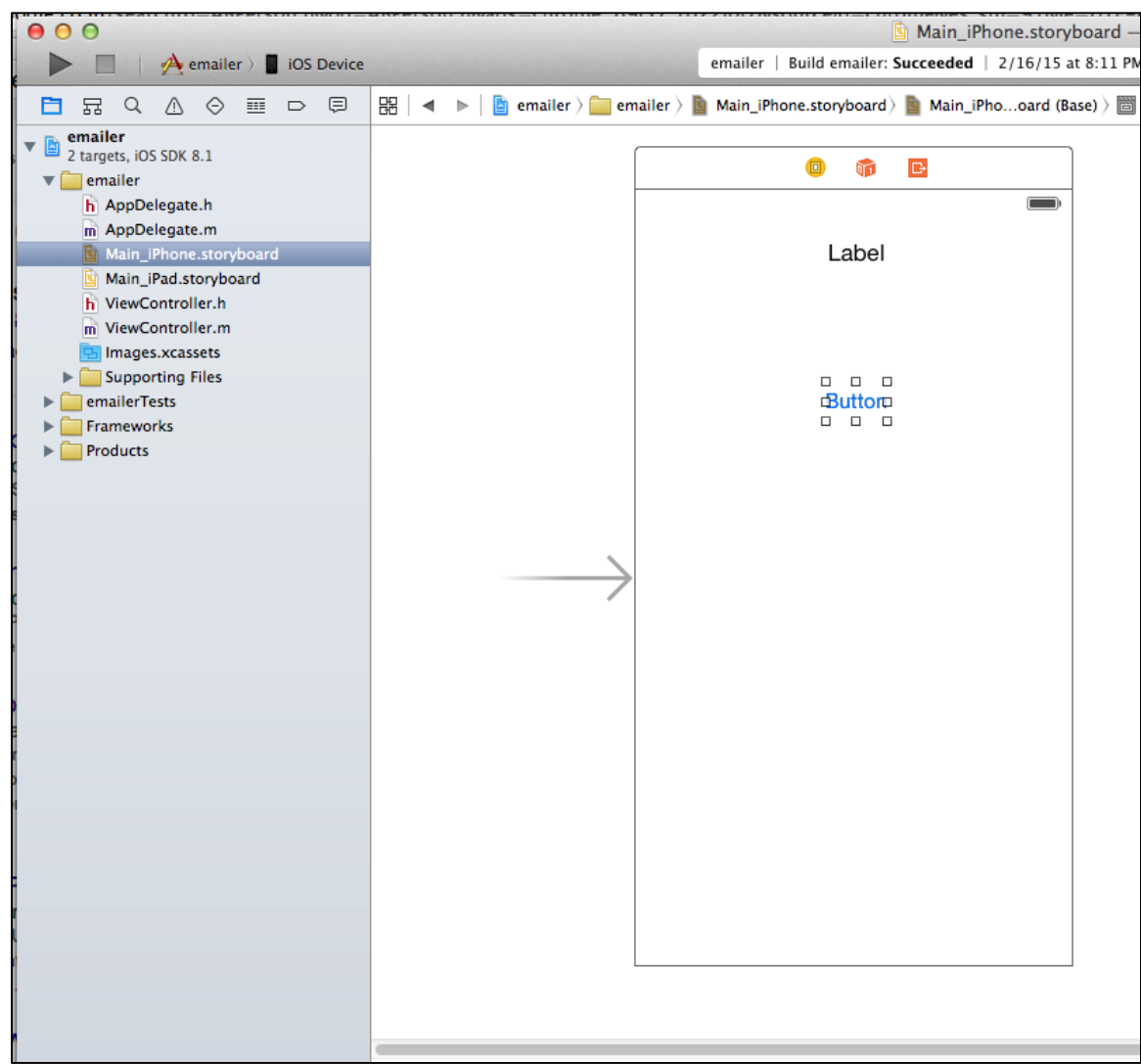


图 7: 应用的用户界面

8. 添加 **Send Mail** 按钮，需要右键按住这个按钮，并拖放到 ViewController.m，如图 8 所示。将按钮命名为 **SendMail**，然后点 **Connect**。

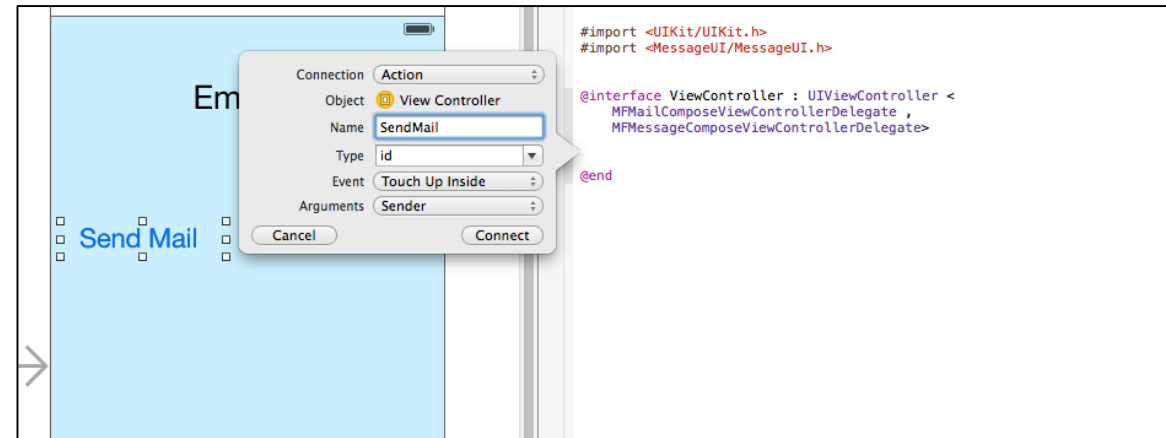


图 8: 将 UI Button 命名为 SendMail

9. 类似为应用添加 **Send SMS** 和 **Dial Phone No.**按钮, 如图 9 所示:

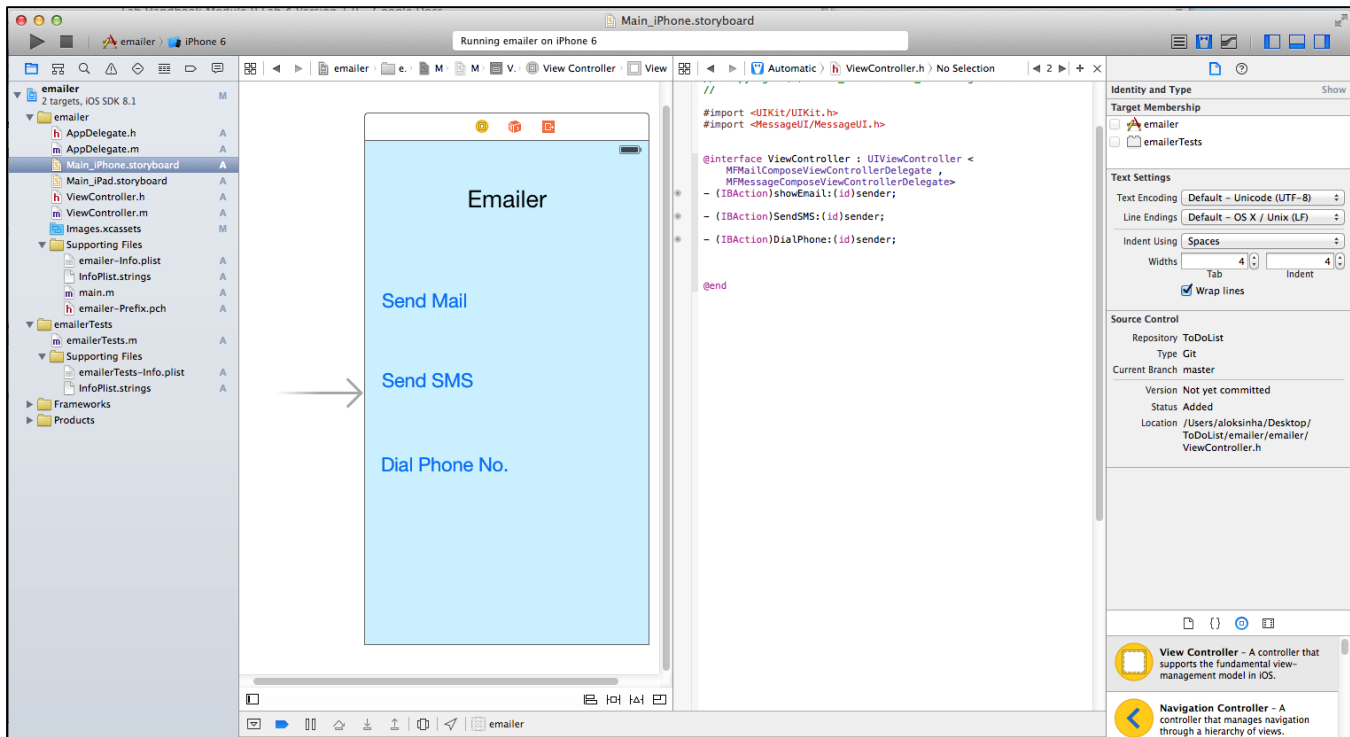


图 9: 添加其它 UI Button

10. 要发送电子邮件, 你需要在 ViewController.h 文件中添加 `#import <MessageUI/MessageUI.h>`, 具体代码如下所示:

```
#import <UIKit/UIKit.h>
#import <MessageUI/MessageUI.h>
@interface ViewController : UIViewController <MFMailComposeViewControllerDelegate >
- (IBAction)showEmail:(id)sender;
- (IBAction)SendSMS:(id)sender;
- (IBAction)DialPhone:(id)sender;
```

11. 然后, 编辑 ViewController.m 文件, 如下面代码所示。这里使用了 MFMailComposeViewController 方法, 它不能在 Xcode6 模拟器上工作, 因此你需要在实际设备上运行这段代码。

```
- (IBAction)showEmail:(id)sender {

    // Email Subject
    NSString *eSub = @"Test Email";
    // Email Body
    NSString *mBody = @"M9d3 EMailer app";
    // Recipient address
    NSArray *recipients = [NSArray arrayWithObject:@"john@gmail.com"];

    MFMailComposeViewController *Ecomp = [[MFMailComposeViewController alloc] init];
    Ecomp.mailComposeDelegate = self;
    [Ecomp setSubject:eSub];
```

```
[Ecomp setMessageBody:mBody isHTML:YES];
[Ecomp setToRecipients:recipients];

// Present mail view controller on screen
[self presentViewController:Ecomp animated:YES completion:NULL];

}

- (void) mailComposeController:(MFMailComposeViewController *)controller
didFinishWithResult:(MFMailComposeResult)result error:(NSError *)error
{
    switch (result)
    {
        case MFMailComposeResultCancelled:
            NSLog(@"Mail cancelled");
            break;
        case MFMailComposeResultSaved:
            NSLog(@"Mail saved");
            break;
        case MFMailComposeResultSent:
            NSLog(@"Mail sent");
            break;
        case MFMailComposeResultFailed:
            NSLog(@"Mail sent failure: %@", [error localizedDescription]);
            break;
        default:
            break;
    }

    // Close the Mail Interface
    [self dismissViewControllerAnimated:YES completion:NULL];
}

@end
```

- 最后，选择 **Project** 菜单中的 **Run** 选项，或是使用 **Command+Run** 选项来运行模拟器。你会得到如图 10 所示的输出。点 **Send Mail** 选项，**Test Email** 视图会打开，如图 11 所示。**Test Email** 中还有一个 **Cancel** 选项，会询问你是保存还是删除草稿。这是 iOS SDK 的一项内建属性。

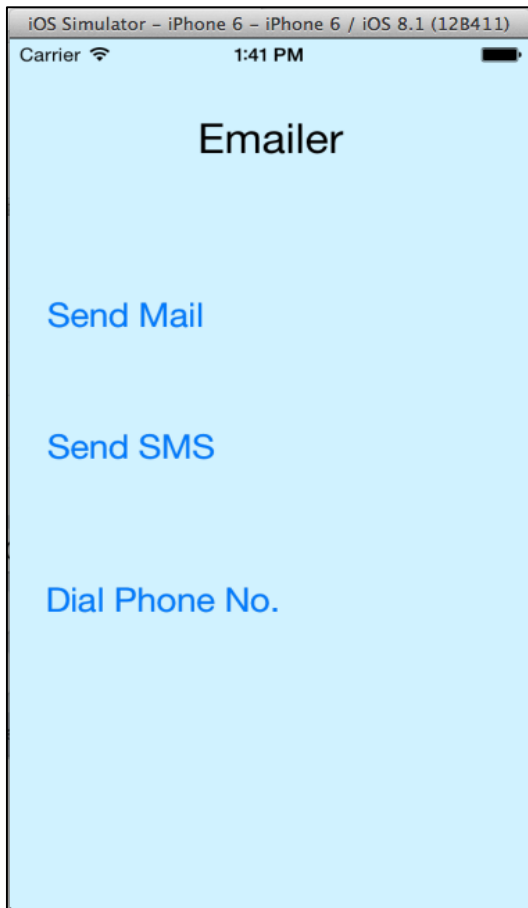


图 10: 运行 Email 应用

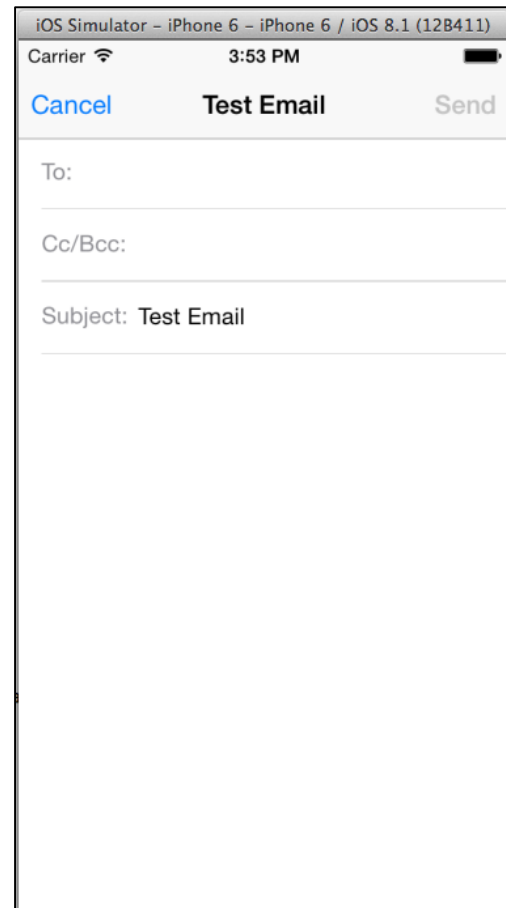


图 11: 点 Send Mail 后的结果

13. 再在 ViewController.m 文件中添加 SendSMS 按钮的代码。这里添加了 MFMessageComposeViewController 类，为发送短信提供标准系统用户界面。该类会调用 canSendText 类方法来确保用户设备配置好了短信发送功能。

```

- (IBAction)SendSMS:(id)sender;
{
    MFMessageComposeViewController *controller = [[[MFMessageComposeViewController
alloc] initWithSender:self];
    if([MFMessageComposeViewController canSendText])
    {
        controller.body = @"This is Test SMS from Emailer app.";
        controller.recipients = [NSArray arrayWithObjects:@"12345678", nil];
        controller.messageComposeDelegate = self;

        [self presentViewController:controller animated:YES completion:nil];
    }
}

- (void)messageComposeViewController:(MFMessageComposeViewController *)controller
    didFinishWithResult:(MessageComposeResult)result {
    switch (result) {
        case MessageComposeResultCancelled:
    
```

```
        NSLog(@"Cancelled");  
        break;  
  
        case MessageComposeResultSent:  
            NSLog(@"Message sent");  
            break;  
  
        default:  
            break;  
    }  
  
    [self dismissViewControllerAnimated:NO completion:nil];  
}  
  
@end
```

14. SendSMS 按钮会打开 iOS 实际设备的短信功能。由于这段代码是在 Xcode 模拟器上运行的，短信服务是不能用的，你会得到如图 12 所示的警告消息：

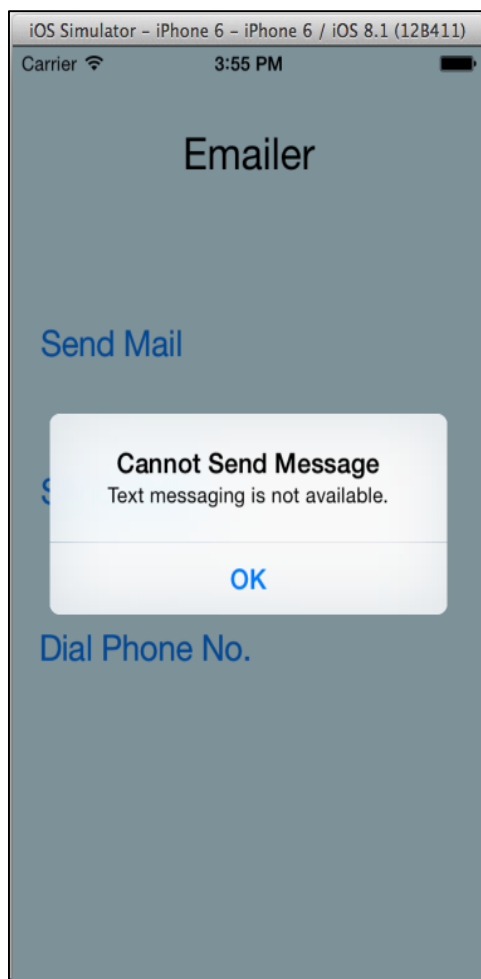


图 12: 点 Send SMS 的结果

15. 再在 ViewController.m 文件中添加 DialPhone 按钮的代码, 这段代码会在 Xcode 中自动呼叫电话号码:

```
- (IBAction)DialPhone:(id)sender;
{
    [[UIApplication sharedApplication]openURL:[NSURL URLWithString:@"tel:+9112345678"]];
}
@end
```

源码

1. ViewController.h 文件

```
#import <UIKit/UIKit.h>
#import <MessageUI/MessageUI.h>

@interface ViewController : UIViewController <MFMailComposeViewControllerDelegate ,
MFMessageComposeViewControllerDelegate>

- (IBAction)showEmail:(id)sender;

- (IBAction)SendSMS:(id)sender;

- (IBAction)DialPhone:(id)sender;

@end
```

2. ViewController.m 文件

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
```

```
- (IBAction)showEmail:(id)sender {

    // Email Subject
    NSString *eSub = @"Test Email";
    // Email Body
    NSString *mBody = @"M9d3 EMailer app";
    // Recipient address
    NSArray *recipients = [NSArray arrayWithObject:@"john@gmail.com"];

    MFMailComposeViewController *Ecomp = [[MFMailComposeViewController alloc] init];
    Ecomp.mailComposeDelegate = self;
    [Ecomp setSubject:eSub];
    [Ecomp setMessageBody:mBody isHTML:YES];
    [Ecomp setToRecipients:recipients];

    // Present mail view controller on screen
    [self presentViewController:Ecomp animated:YES completion:NULL];

}

- (void) mailComposeController:(MFMailComposeViewController *)controller
didFinishWithResult:(MFMailComposeResult)result error:(NSError *)error
{
    switch (result)
    {
        case MFMailComposeResultCancelled:
            NSLog(@"Mail cancelled");
            break;
        case MFMailComposeResultSaved:
            NSLog(@"Mail saved");
            break;
        case MFMailComposeResultSent:
            NSLog(@"Mail sent");
            break;
        case MFMailComposeResultFailed:
            NSLog(@"Mail sent failure: %@", [error localizedDescription]);
            break;
        default:
            break;
    }

    // To close the Mail Interface
    [self dismissViewControllerAnimated:YES completion:NULL];
}

- (IBAction)SendSMS:(id)sender;
{
```

```
MFMessageComposeViewController *controller = [[MFMessageComposeViewController
alloc] init] self];
if([MFMessageComposeViewController canSendText])
{
    controller.body = @"This is Test SMS from EMailer app.";
    controller.recipients = [NSArray arrayWithObjects:@"12345678", nil];
    controller.messageComposeDelegate = self;

    [self presentViewController:controller animated:YES completion:nil];

}
}

-(void)messageComposeViewController:(MFMessageComposeViewController*)controller
didFinishWithResult:(MessageComposeResult)result
{
    switch (result) {
        case MessageComposeResultCancelled:
            NSLog(@"Cancelled");
            break;

        case MessageComposeResultSent:
            NSLog(@"Message sent");
            break;

        default:
            break;
    }

    [self dismissViewControllerAnimated:NO completion:nil];
}

- (IBAction)DialPhone:(id)sender;
{
    [[UIApplication sharedApplication]openURL:[NSURL URLWithString:@"tel:+9112345678"]];
}

@end
```