

Lab – “Pickers”

OVERVIEW

Date Due: Thursday, March 25th (after break) by 11:59 p.m. EST. ✧ **Value:** 15 points of extra credit

Features

This project focuses on services provided by UIViewControllers that you can drop into your applications. You will be writing an application that uses the people picker, image picker and mail compose controller. Once an email address and image are selected, users will be able to bring up a compose window prepopulated with a receiving address, subject and body containing the selected image. All features can be done in the simulator, but require you to do some setup - to populate the image and address book database.

Email Selection

- Allow selection of one email address
- Display the selected user’s composite name and email address on the main screen

Image Selection

- Allow selection of a single, croppable image
- Display the selected image on the main screen after being selected

Email Compose

- Prepopulate the subject with any text you want...
- Prepopulate the receiving email address with the email the user selected
- Prepopulate the message body with the selected image as an attachment
- Although you can’t send email from the simulator, you can still bring up the email compose view.
This is the perfect opportunity to try building and installing your first app on a device! If you want to get started with deploying to a device, start with these device setup instructions, here: <http://www.cs.umd.edu/class/spring2010/cmssc498i/files/DeviceSetup.pdf>

NOTES

- “init” – when creating each of the view controllers (UIImagePickerController, MFMailComposeViewController, and ABPeoplePickerNavigationController), do not use “initWithRootViewController:”, and instead use just “init”. This makes sense once you realize that these controllers are managing the hierarchy, not you...
- Frameworks – This is the first project requiring you to add additional frameworks to the set you link against. By default, projects link against Foundation, UIKit, and CoreGraphics. To use the people picker, and compose controller, you need to add: AddressBookUI.framework, AddressBook.framework, and MessageUI.framework. The helper code you are being given relies on MobileCoreServices.framework, so you will need to link against that too. An explanation of how to add frameworks is at the end of this handout.
- Starter resources – You are being provided with PickerLabHelper.[hm] and 3 jpgs. Drag each into your project. Call `seedPhotosAndAddressBook` when your app finishes launching. It will add the 3 images to your photo library and also make sure you have at least one person in your Address Book. The other methods help you parse AB picker results and help you attach a UIImage to a mail compose controller. Check out the code to learn what it is doing... The resources can be found here: http://www.cs.umd.edu/class/spring2010/cmssc498i/files/lab_Pickers_resources.zip



Figure 1

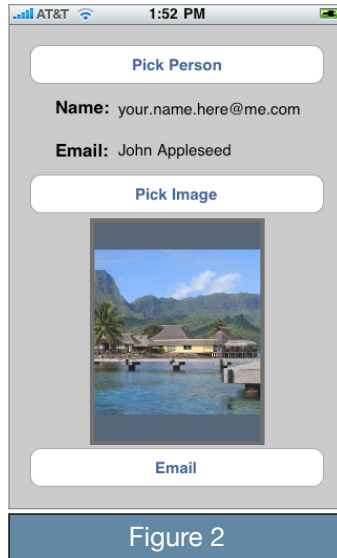


Figure 2

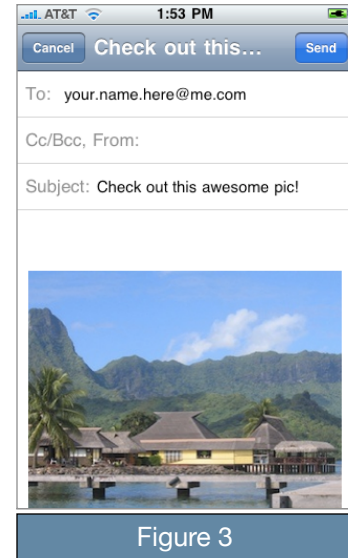


Figure 3

Figure 1 shows the application at launch. The “Pick Person” button brings up the people picker. When selected, the person’s name and email address is shown as in Figure 2. The “Pick Image” button brings up the image picker which should be configured to allow cropping a picked photo. When selected, the image will be shown as in Figure 2. Notice the selected image is scaled down and shows just the portion the user chose to crop to. The blue background and border can be easily implemented using background color properties and plain UIViews instead of doing custom drawing. Figure 3 shows the results of clicking on the “Email” button in Figure 2. Each picker should slide up and be presented modally.

To add a new framework to the link set, select the application’s main target in the project window (Figure 4). Choose Project→Edit Active Target, and the “Target Info” window will be shown (Figure 5 – without the attached sheet). Make sure the “General” tab is selected, then click the “+” button and a sheet listing the available frameworks will show up.

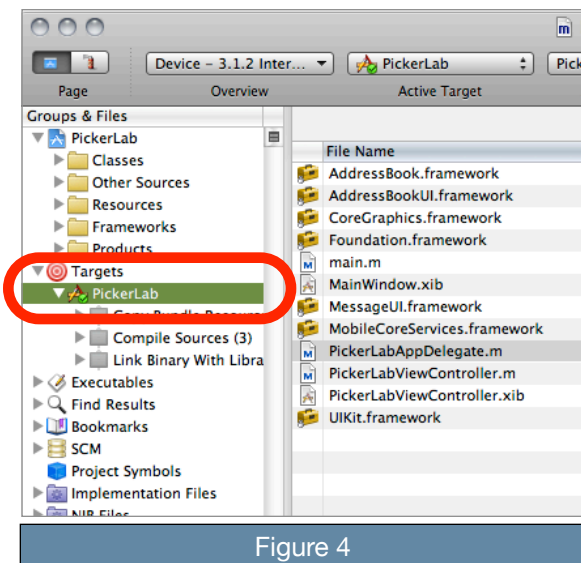


Figure 4

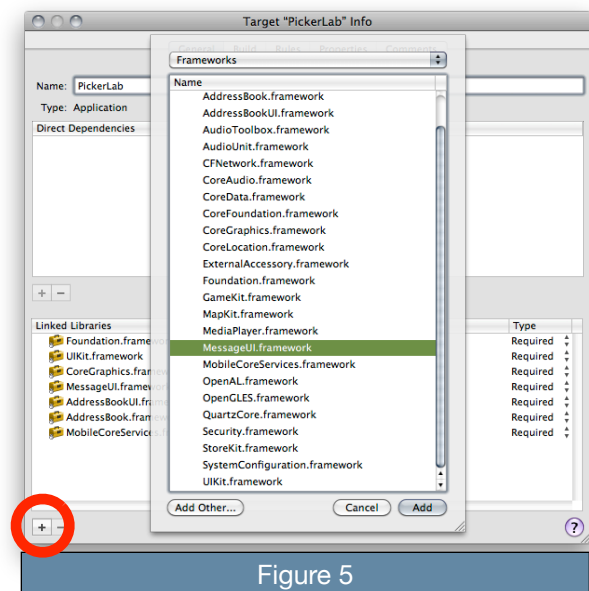


Figure 5