

ECS189E Homework 1

○ Overview

The app you are going to build for the homework is a wallet, which can be broken down into three parts: login, wallet, and transaction.

For the login part, your App will use the phone number authentication. To be more specific, your App will take a phone number from the user and send a verification code to this phone number by text message. Your App will then ask the user to enter the code and verify user's identity. You can assume that the user will use US phone number only. (Homework 1 and 2)

For the wallet part, your App will display the wallet, which has a table of accounts and the information of each account. For example, it might contain the name of the account, the available balance of each account and so on. (Homework 3)

For the transaction part, your App will allow the user to make transactions between the accounts in the wallet. It will also use card scan to "deposit" into your accounts. (Homework 4)

○ Expectation

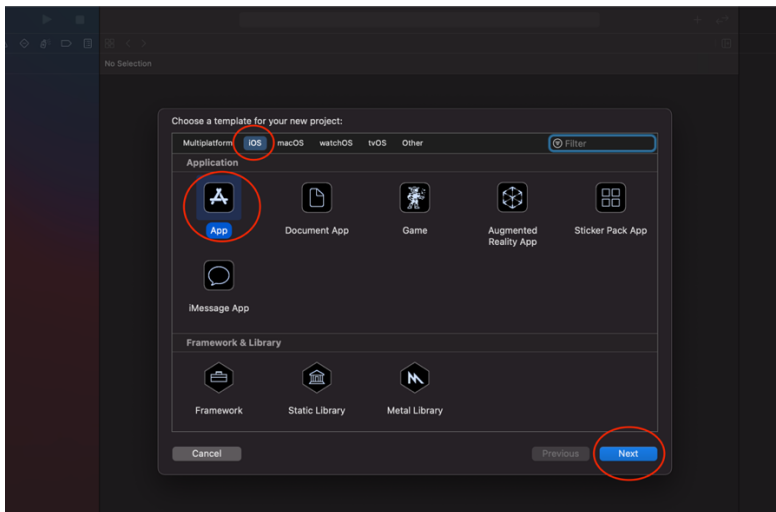
When building an App, there is not a single "correct" solution. As one of the purposes of this course, we do want you to be creative and put your unique thoughts and designs into your App. The homework instructions will list out the basic functions and objects that you have to include in this App. However, there are lots of things that you can build on top of these requirements.

○ Create and setup the Project

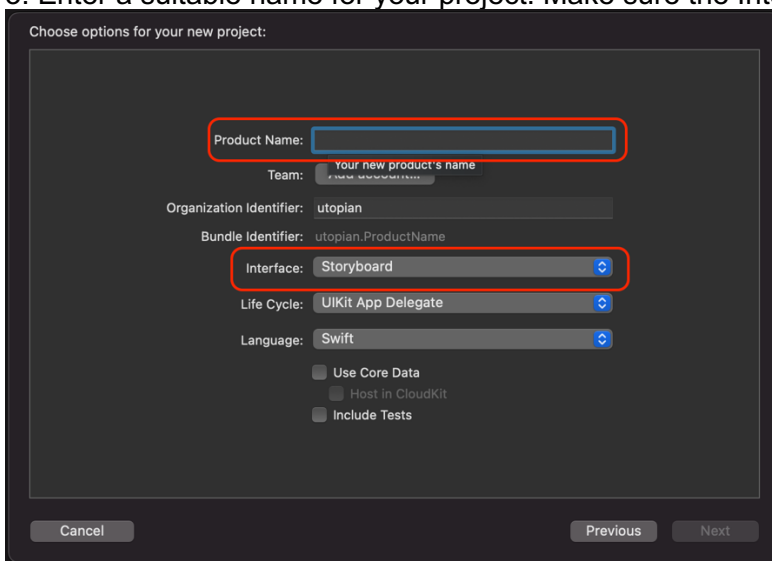
1. Open xcode and click on "Create new code project".



2. Create an iOS app.



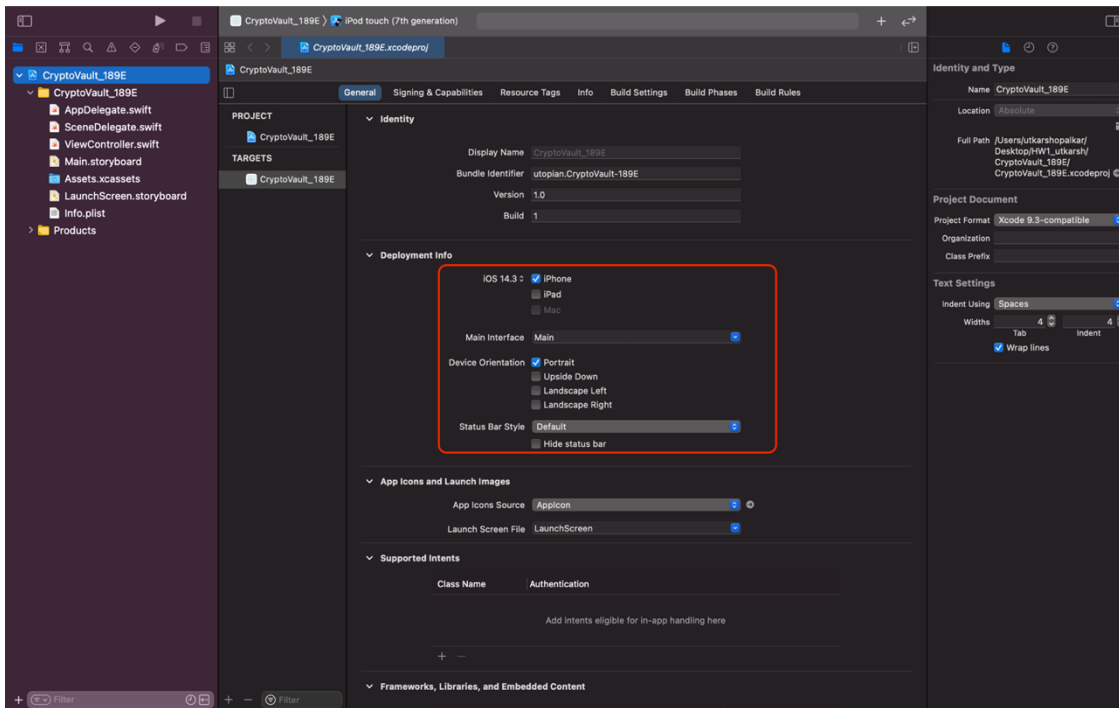
3. Enter a suitable name for your project. Make sure the Interface is selected to “Storyboard”.



4. Choose the appropriate location for the project. (Keep a separate folder for your HWs as we will be using what we built for next HWs).

5. Now a file should appear on your screen within XCode as <your_project_name>.xcodeproj

6. Make sure to your app is only for iPhone in Portrait mode.



7. Now open a Terminal window and navigate to your project folder. (Don't go inside the project folder, stay one folder out). Run the command “pod init” to initialize a pod file. Pods are like libraries for xcode. You should now be able to see a “Podfile” in the folder.

8. Open the pod file. Between the lines “target <your_project_name> do” and “end”, type in “pod ‘PhoneNumberKit’”. Your podfile should look something as the image below.

```

Podfile

# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'CryptoVault_189E' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # Pods for CryptoVault_189E
  pod 'PhoneNumberKit'
end

```

9. Run command “pod install” back at the terminal. This will install you pod for “PhoneNumberKit” which you are supposed to use for this HW1. If run successfully, it should look something as follows.

```

-bash

utkarshs-MacBook-Pro:CryptoVault_189E utkarshopalkar$ pod init
utkarshs-MacBook-Pro:CryptoVault_189E utkarshopalkar$ pod install
Analyzing dependencies
Downloading dependencies
Installing PhoneNumberKit (3.3.1)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `CryptoVault_189E.xcworkspace` for this project from now on.
Pod installation complete! There is 1 dependency from the Podfile and 1 total pod installed.

[!] Automatically assigning platform `ios` with version `14.3` on target `CryptoVault_189E` because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.
utkarshs-MacBook-Pro:CryptoVault_189E utkarshopalkar$

```

10. Your project is now set up. Go back to xcode. From the project navigator (mostly on left) and open the “LaunchScreen.storyboard”.

Launch Screen

11. You have now entered the land of unknown adventure; you are on your own! (Not really, you can always approach the instructors for help). We expect you to create a launch screen for your app by yourself which shall display at least, an image icon and text name for your cryptocurrency wallet.

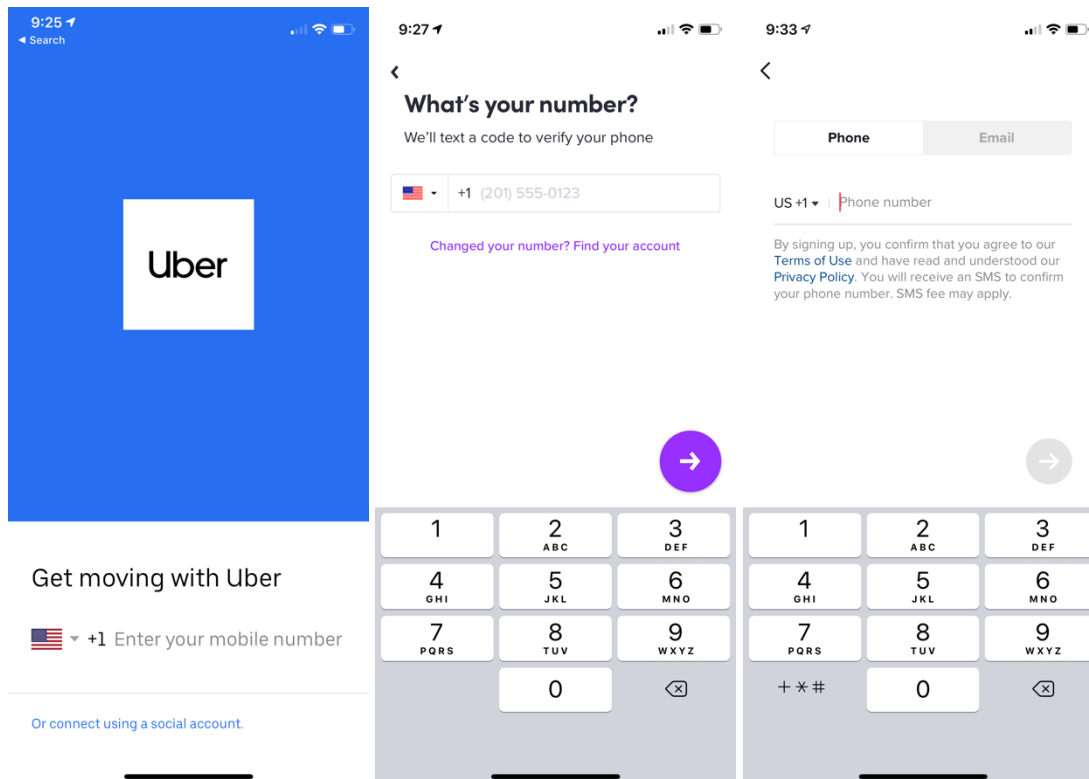
Login View

12. After you have designed your launch screen, open up the “Main.storyboard”. This is where you will develop a Login View which will be used to log in the users into their wallets.

13. Your code for the behavior of the UI components of your Main.storyboard should be written in “ViewController.swift”. Good luck!

NOTES:

- The instructions/screenshots above are from XCode Version 12.3 (12C33).
- SwiftUI is a new way of building UI introduced by Apple recently. We are not going to cover it in this class, but feel free to learn more about it.
- Follow this [link](#) to learn more about using CocoaPods. The only Pod that is required to use for this homework is called "PhoneNumberKit".
- This [link](#) can be a great source for you to pick icons. Always remember to check the copyright when you use any resource.
- Phone Number Authentication. There are a lot of Apps use the phone number authentication. Here are some examples:



- **Things to consider during the Planning and Development stages:**

- ◆ **UI Design**

On the first view of the login part, you should at least have:

- An instruction (or instructions) on the screen to tell users what to do.
- A text field showing the country code, i.e., +1 for US. You can assume the user will use US phone number only. User shouldn't be able to interact with the country code field for now.
- A text field for users to enter the phone number.
- A button to send the One Time Password/code via text.
- A label to display errors and information ONLY when needed.
- Use Auto Layout to make sure your app looks the same on different devices.

Besides these requirements, feel free to add anything or any function that you consider necessary.

- ◆ **Display the Input in Phone Number Format**

Instead of a list of digits, phone numbers are usually displayed in a certain format. Adjusting the format of the input while user is entering can be very helpful for the user. An example of formatted phone number would be: (012)-345-6789. For the case of this homework, you can use the "PhoneNumberKit" Pod that was installed.

One of the objectives of this HW is also to make you comfortable with going through official documentation of libraries (or Pods). You can find the documentation to for "PhoneNumberKit" pod, [here](#).

Hint: Check out the AsYouTypeFormatter

- ◆ **Dismiss keyboard**

Your app's keyboard should also be dismissed when the user tapped outside the keyboard or tapped the send button. To detect gestures, use UITapGestureRecognizer. You can learn more from [here](#).

Hint: Create a gesture recognizer object and add it to your view

- ◆ **Error Detection and Display**

When user taps the button, check if the input is a valid phone number and display the errors. Make sure you have at least two types of errors and an information for valid input. When the input from the user is valid, the label should display the data that will be forward to the next step.

In this homework, you need to store the phone number in E164 format. For example, if user entered 0123456789, it should be displayed in the text field as (012)-345-6789 (UITextField) but should be stored and displayed as +10123456789 in the information (UILabel). This string is the data that you will use in the next homework.

Hint: You should also be able to get the phone number in E164 format with the "PhoneNumberKit" Pod easily.

- ◆ **Coding Style**

It is always very important to make sure that your code is easy to read and understand. The following points will be considered when your coding style is graded:

- Reasonable names for ViewControllers, views and objects.

- Reasonable names for files, functions and variables.
- Necessary comments.
- Reasonable order of the functions.
- The use of optional (Question mark and exclamation mark).

◆ Documentation

Write a README.md to briefly state what functions does the App have and how did you implement them. Imagine yourself working in a big company, and this document is for the people who will take over your work or will become your partner and work on this App with you.

Important: Remember to include your name and student ID in the document.

Grading

- Launch Screen (5')
 - UI Design and Auto layout (15')
 - Correctly use the Pod (the format of phone number while typing) (10')
 - Correctly dismiss the keyboard (5')
 - Error detection and display (10')
 - Coding style and documentation (5')
-

○ Submission

Push your files to the repository.
