ECS189E Homework 2

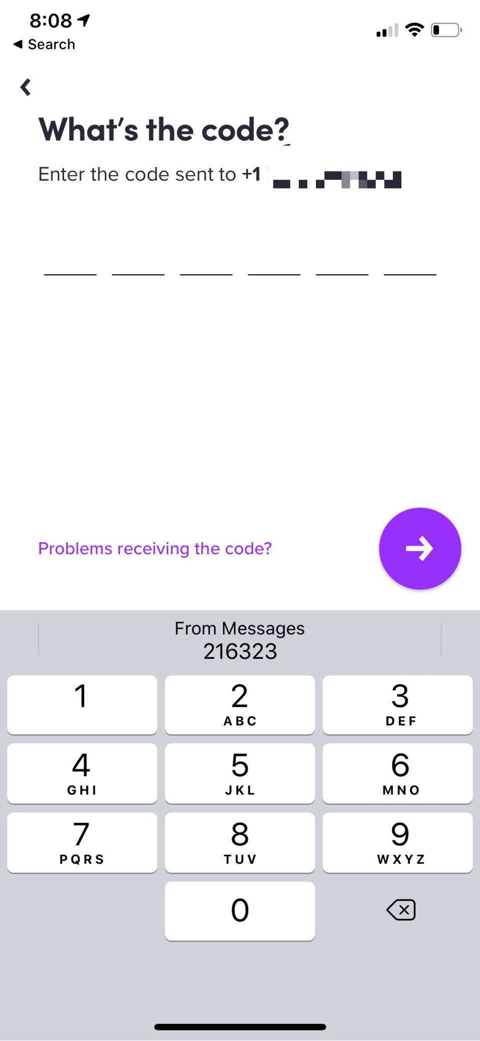
Overview

In the homework 2, you will complete the login part. Following the homework 1, your App will send out the verification code to the phone number that user entered. After that, you need to instruct users to enter the verification code that they received; verify if the code is correct.

Usage of Provide Code

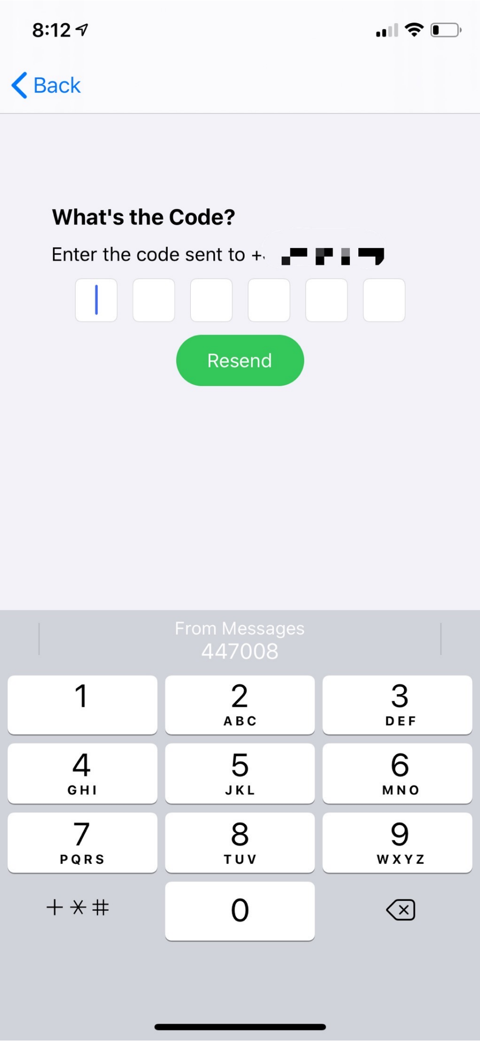
Clone the homework 2 repository to your local device. Move your code of homework 1 to this directory. Add the Api.swift file to your project folder (If your project is called "wallet", put it in the "wallet" folder.) and also use the Xcode to add the file to your project. My version of the homework will be provided as a start kit one day after the due. You can choose to work on my code or your own code for the next homework.

View of Verifying Code



What's the code?

Enter the code sent to +1 [redacted]



What's the Code?

Enter the code sent to + [redacted]

The first image shows how the Lyft instruct users to enter the verification code. And the second image shows mine example app.

For this homework, **You have to use a separated view for verification.** The purpose of this requirement is for you to have some experience of using navigation controller. For this verification view, just like the first view, you are required to include:

1. a instruction,
2. a OTP style text field,
3. a resend button that will resend the verification code,
4. and a label to present verification error.

Make sure you use a separate swift file for the new ViewController and give it a reasonable name.

More about Verification

1. **It is required to** make the text field in the OTP style. In other words, six separate boxes or positions each containing one digit. **You'll do this by using six different text fields**. I know there are online tutorials that tells you how to do it with only one text field and several labels. Don't follow that tutorial because you'll need to know protocols and delegates for that method.

- While the user is entering the code, the cursor should automatically go to next text field. While the user is deleting the code, the cursor should automatically go to the previous text field.
  - You might find that hitting backspace won't work if you are in the middle of typing in code, i.e., when you are entering code in the second box. Don't worry about this, you'll fix this bug in your next homework. **Just make sure when the user finished entering the code, the deletion will work correctly**
- The user shouldn't be able to interact with text fields besides the one he/she is editing. For example, if the user is entering the 2nd digit, he/she shoudn't be able to tap on all the 1,3,4,5,6 text fields and modify them.

2. **Your app is also required to** auto verify the code once the user has typed in all the code. In other words, there shouldn't be any next button needed.
   - Present the **Home View** if the verification code is correct. You don't need to have anything in the Home View. An empty view controller is fine.
   - Show the error message if the verification code is incorrect.
3. The resend button should function as you expected: send the verification code again to the phone number you entered in your Login View
4. Lastly, **your app should** only present the phone pad and **should have** the autofill functionality.
   - Hint: Check out the content type of the text field.

Segues

Since we are dealing with multiple ViewControllers from now on, you need to use the navigation controller and "present" for your App.

Navigation Controller

You'll use navigation controller to segue from the Login View to the Verification View.

Present

You'll use present to segue from the Verification View to the the Home View. In other words, there shouldn't be a back button on the top left so that the user can go back to the Verification View. **Hiding the navigation bar or remove the back button on the navigatino bar is not the same as "present".**

Send Verification Code and Verify Code

For this section, you will use the provided Api to send the verification code to the entered phone number, and verify the entered code. There is also a short note provided as instructions at the beginning of the file.

Usage

```
// Send Verification Code
Api.sendVerificationCode(phoneNumber: YourPhoneNumber) { response, error in
    // Handle the response and error here
}

// Verify Code
Api.verifyCode(phoneNumber: YourPhoneNumber, code: YourCode) { response, error in
  // Handle the response and error here
}
```

Possible Callbacks

In the case of success (verification code sent or code verified), the response should not be nil while the error should be nil. For the opposite (cannot sent the verification code cannot be verified), the response should be nil while the error should not be.

For this homework, you do not need to handle the response.

Error

There are two class variables that you can use: `error.code` and `error.message`.

code	message
"invalid_phone_number"	"Your phone number is invalid"
"incorrect_code"	"Incorrect verification code"
"code_expired"	"Your code expired"

You do not have to use the message provided. You can also test for code and customized your message.

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:

1. Reasonable names for ViewControllers, views and objects.
2. Reasonable names for files, functions and variables.
3. Necessary comments.
4. Reasonable order of the functions.
5. 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.

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

Grading

1. UI Design and Auto Layout (5')
2. Segues (15')
3. OTP Style Text Field and Resend Button Functionality (15')
4. Send and Verify One Time Code (10')
5. Coding Style and Documentation (5')

Submission

Push your files to the repository.