# Exercise 2

In this exercise, we will build an app that allows you to create user accounts on a mock server and log in a user with their username and password. In the extra-credit addition to this exercise, you will be able to log out a user as well as update their password. Create a Server class

1. Create a class called `Server` with three properties:

    - `registeredUsers` initialized with an empty Dictionary of type `[String : String]`
    - `loggedInUser` of type `String?`
    - `requiredPasswordLength` of type `Int`

2. Create a method called `createNewUser`:

    - takes two `String?` parameters: `username` and `password`
    - returns a tuple of type `(Bool, String)` which indicates whether the user was created successfully with a description of why or why not
    - the method should add a key-value pair to `registeredUsers` in the form `username: password` as long as the following conditions are met:

        - `username` and `password` have values
        - the key `username` is not already present in `registeredUsers`
        - the length of `password` satisfies `requiredPasswordLength`
        - Note: you can find the length of a `String` as follows: `exampleString.characters.count`

3. Create a method called `logIn`:

    - takes two `String?` parameters: `username` and `password`
    - returns a tuple of type `(Bool, String)` which indicates whether the user was logged in successfully with a description of why or why not
    - the method should update the `loggedInUser` property with the `username` value if the following conditions are met:

        - `username` and `password` have values
        - `registeredUsers` has a value for the key `username`
        - `password` matches the stored password Create an App that interacts with your Server class

4. Create a new Xcode Project
    - Choose a Single View project

    - Create a name for your project
    - Make sure the Language is set to Swift

5. Create your UI using Storyboards
    - Add a textfield
        - Set the textfield's placeholder text to 'Username'

    - Add another textfield
        - Set the textfield's placeholder text to 'Password'

    - Add a button
        - Set the button's type to 'Custom'
        - Set the button's title to 'Login'
        - Choose a background colour for your button
        - Choose a text colour for your button

    - Add another button
        - Set the button's type to 'Custom'
        - Set the button's title to 'Change Password'
        - Choose a background colour for your button
        - Choose a text colour for your button

    - Add a label

6. Add Actions and Outlets to your ViewController code
    - Open the Assistant editor to view your Storyboard and ViewController code simultaneously
    - Create an outlet called called `usernameTextField`:

        - Control-drag from the appropriate textfield in the Storyboard to the body of your ViewController class and choose 'Outlet'

    - Create an outlet called `passwordTextField`:

        - Control-drag from the appropriate textfield in the Storyboard to the body of your ViewController class and choose 'Outlet'

    - Create an action called `loginButtonPressed`:

        - Control-drag from the appropriate button in the Storyboard to the body of your ViewController class and choose 'Action'

    - Create an action called `createAccountButtonPressed`:

        - Control-drag from the appropriate button in the Storyboard to the body of your ViewController class and choose 'Action'

    - Create an outlet called called `responseLabel`:

        - Control-drag from the label in the Storyboard to the body of your ViewController class and choose 'Outlet'

    - Test your connections by adding `print` statements to your IBAction methods and running your app in the Simulator

7. Configure your Application Logic to interact with your Server class
    - Create a new Swift file in your Project called Server.swift, and add the code you created for your `Server` class to this file
    - Create a property in your ViewController that is assigned to an instance of your `Server` class
    - When `createAccountButtonPressed` is fired, call the `createNewUser` method on your Server instance and update your `responseLabel` outlet to display the server's response message
    - When `loginButtonPressed` is fired, call the `logIn` method on your Server instance and update your `responseLabel` outlet to display the server's response message