iOS Training Program: Week 3 Mini-Project

MDB Socials

OVERVIEW

This week we'll be exploring backend data storage/retrieval via Firebase. One of the cool features of Firebase over most other BaaS tools is its real-time functionality (in other words, Firebase automatically notifies you when some value in the database is updated). We'll be building an app called MDBSocials which will allow members to post and RSVP to social events.

Note: This is the most difficult and most important of all of the MDB Training Program projects. Expect it to be difficult but don't get discouraged!

SETUP

Before you get started you'll need to make sure you have created a new project on Firebase and completed the necessary steps for setup (see https://firebase.google.com/ docs/ios/setup for detailed instructions). Note that setting up Firebase within your iOS app requires installing the library via Cocoapods. You can find a tutorial on Cocoapods here: https://www.youtube.com/ watch?v=SlY39ga19ws

SPECS

Your app should have the following screens and features:

- **LoginViewController**
  - Require the user to enter their username and password to log in. Also, allow the user to create a new account by signing up (so have a signup button that takes you to the SignupViewController).
  - If a user is already logged in and opens up the app, they should automatically advance to the feed view.
- **SignupViewController**
  - The user should be brought to this screen when they hit the sign up button in the SignupViewController. Require the user to input their full name, username, password, and email when signing up.
  - Once the user has filled out all the necessary info, they should tap the sign up button on this screen and be redirected to the FeedViewController.
  - Both login and signup should be done using the Firebase Auth microservice. See the tutorial video on user authentication for more details. We highly encourage you to also read the documentation: https://firebase.google.com/docs/auth/ios/ password-auth
- **FeedViewController**
  - Display a list of socials where each cell contains the following information:
    - name of member who posted
    - name of event
    - picture of event
    - number of people who RSVP'd "Interested"
  - Tapping a cell should lead the user to the detail screen for that particular social. Display a button somewhere on the screen that leads the user to the NewSocialViewController to

  create a new social.

  - Socials should be displayed in order of most recent to oldest.
  - Any time someone creates a new event, this screen should automatically update - you'll need to test this across more than one device.
- **DetailViewController**
  - This screen should display an expanded view of the same details from the FeedViewController in addition to a description of the social.
  - Additionally, display a button saying "Interested" that, when clicked on, does the following: the "Interested" button should show up as being checked, and the number of interested people on the screen should increase.
- **NewSocialViewController**
  - When the user creates a new social, they should be required to fill out the following:
    - Name of Event
    - Picture – allow them to open up the default camera to take a picture OR to upload from camera roll.
    - Description - user should be allowed to type a short description (up to 2 lines)
    - Date – start date of the event including day & time.
  - Once the user creates the event, they should return to the FeedViewController with their new event at the top.
  - Make sure your navigation flow is appropriate (hint: present this VC modally and dismiss it when the event is posted)

OTHER REQUIREMENTS

- Think carefully about navigation! Sometimes it might make more sense to present something modally than with the normal show segue. The user should always be able to navigate between screens freely.
- Although we've listed most of the features that you need on each screen, please think carefully because there will be some intuitive features that you should have that we haven't mentioned (like back buttons).
- As always, good quality design is an expectation. Feel free to look at existing apps for inspiration. We will especially be looking for great design on this project so once you finish, please spend time improving your design.
- The entire project should be done programmatically. You should have ViewControllers in your storyboard with the appropriate segues, but each VC should be blank in the storyboard
- Make sure your code is clean and easily readable. See https://github.com/raywenderlich/swift-style-guide for Swift style guide. We will be assessing you on your style.
- Last but not least, add a README outlining the purpose of the app and which features are included.
- **Note:** A lot of issues can be avoided if you plan your app out BEFORE beginning to developer. Draw out the app flow and all UI elements before you open Xcode.

SUBMISSION

Just like your previous submissions, submit your app through this google form: https://goo.gl/forms/Rf9FiYZXkLKFI7kI3.