



## iOS Training Program: Week 1 Mini-Project

### Match the Members!

#### PROJECT DESCRIPTION

Welcome to MDB! This semester, we'll have over 50 members, including a new recruit class of 14 of the brightest minds on campus. What better way to get to know everyone in the club than to build an app about them? We'll be building a simple, multiple-choice trivia game to see how fast you can learn everyone's names.

You will be working alone for this project.

#### SPECS

You will be provided a text file with the names of every member in the club as well as a folder with everyone's picture.

Create a new project in Swift and build an application with the following features:

#### Basic functionality of the game:

##### Start Screen

- A screen with a button that says start. Upon pressing the button, the user is taken to the main screen.

##### Main Screen

- *Member Picture* - Randomly display a picture of a member on the screen
- *Name Options* - Display four options below the picture as buttons, each with different names that may correspond to the picture (one of the names should actually be correct, but the others can be random – be sure you don't randomly also select the correct answer)
- *Score* - Display a score count for the user – when the user correctly answers, increase the score by 1. Otherwise, just move on to the next question.
- *Answer Display* - You can go from one question to another just by changing the image and button values, but there should be a one second delay in between questions where the

button the user pressed flashes either red or green (depending on whether they're correct or not). You'll have to look up how to implement timers.

- *Timer* - Add an additional timer that limits the user to 5 seconds to answer the question. If they take up the full 5 seconds, the question is counted as incorrect. The user should be able to start or stop the game. When the app is first opened, display a "Start" button (at this point, the user should not be able to see any questions on the screen). This button should toggle to "Stop" when they click on it. The game should automatically stop if the user goes to a different screen.

### **Statistics Screen**

- *Icon* - Add a statistics icon to the navigation bar of the game. If the user clicks on this, the game should stop (if currently playing) and the screen should segue to a new window that displays the following statistics:
  - Longest streak of correct answers
  - Results of last three questions (or less if not answered yet)
- If the user segues to the statistics page and then returns back to the game, the game should continue as normal with the same score – don't worry about resetting the score.

For this project, you are again required you to handle the layout programmatically (only using Storyboard to set the navigation between ViewControllers but not actually populating each ViewController with subviews). If you are unsure about what this means, ask the instructor!

All screens must be designed well. This is a very easy project, so we will pay extra attention to design.

Note that the design of this application is entirely up to you, but make it look as clean as possible. See examples of trivia apps on the App Store for further inspiration.

### **SUBMISSION**

You should create a Git repository for your submission that includes the entire Xcode project and submit a link to the repo using the submission link on Slack. Add a README outlining the purpose of the app and which features are included. Also include screenshots of the 3 screens in your README. We'll have a pinned message in Slack regarding how to do this if you are not sure.