Mobile Developers of Berkeley

iOS Final Assessment

Instructions: There are 3 parts to this assessment. You are given a starter project for each part. Please read the spec and implement what it is asking you in each part in the starter project given.

Part 1: Maps & Webs (Views)

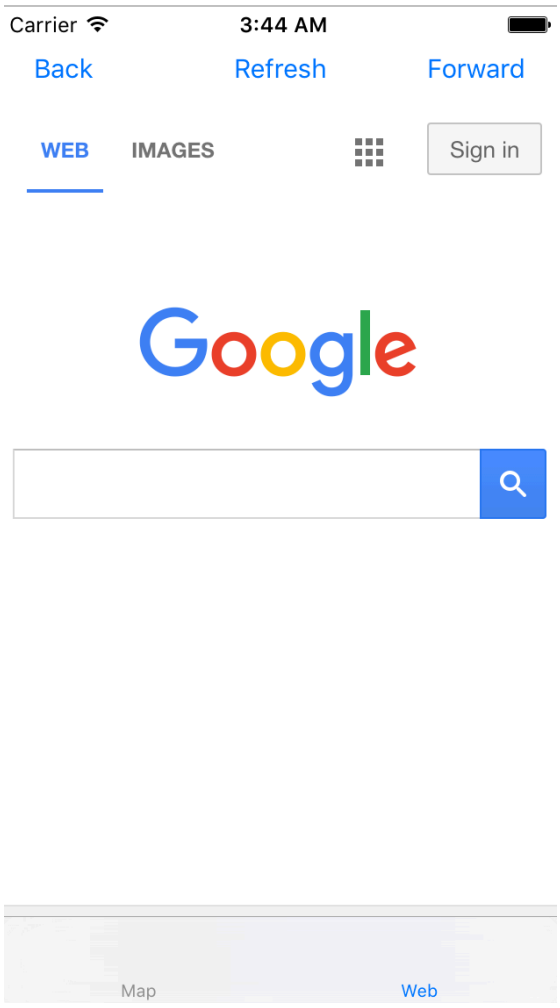
For this part you'll implement a basic tab bar view controller. You must implement one tab bar view controller with two distinct sub view controllers. One view controller must show a fully functioning map with the ability to switch between Default, Satellite, AND Hybird map formats.

The other must implement a fully functioning web browser whose homepage is your favorite search engine.

Your completed project should look somewhat like this:

Tab 1: MapView

Tab 2: WebView



Part 2: Members App (Parse)

You are given a starter project with all Parse libraries added. You are connected to a database with a table called “Members”. The table has all the in built fields, plus the following 3 custom fields: “firstName”, “lastName”, “email”. Open up the starter project. Look at the storyboard so you understand what’s going on. You are initially taken to the MembersViewController when you start the app. When you press the + button at the top, you are taken to the AddMemberViewController. Once a user enters his information into the AddMemberViewController and presses the add button, you should create a new object for the and save it in the “Members” table in your Parse DB. On the MembersViewController, when you click update, you should get the most recently uploaded member and display his/her information in the labels provided. You will write all your code in the updateData function in the MemberViewController and the addMember function in the AddMemberViewController. Here is an example of a Parse query, for your reference:

```
var query = PFQuery(className:"GameScore")
query.whereKey("playerName", equalTo:"Sean Plott")
query.findObjectsInBackgroundWithBlock {
    (objects: [PFObject]?, error: NSError?) -> Void in

    if error == nil {
        // The find succeeded.
        print("Successfully retrieved \(objects!.count) scores.")
        // Do something with the found objects
        if let objects = objects! as? [PFObject] {
            for object in objects {

            }
        }
    } else {
        // Log details of the failure
        print("Error")
    }
}
```

Mobile Developers of Berkeley

Part 3: Computer Parts App

Overview: Make an app from scratch that helps a user figure out which companies make certain computer parts.

Usage: When a user first opens the app, he/she should see a feed of computer parts (RAM, Hard Drive, etc). When a computer part is tapped on, the user should be taken to a new VC where the image of 1 main company that make the parts should animate into the screen.

Implementation Information: The app has 2 ViewControllers: PartsViewController and CompaniesViewController. PartsViewController should be the first VC that displays when the user opens the app. PartsViewController should be implemented using a UITableView inside a UIViewController (not a UITableViewController). Each cell in this feed should have a picture of the computer part and the name of the computer part. Once a cell is clicked, the user is taken to CompaniesViewController. As soon as this VC appears, the 1 relevant company image should animate into the screen (they should fly in and should originally be off the screen). When navigating from the PartsViewController to CompaniesViewController, CompaniesViewController should be presented modally. It should have a nav bar at the top with a Done button. When the done button is clicked, it should dismiss the current VC and show you the PartsViewController again.

Helpful Info: We have provided you with folders of images for both computer parts and companies inside the ComputerParts folder in the assessment folder. You must use at least 5 computer part images. The number of company images you use is up to you as long as you have a different company image displayed when each part is clicked. It doesn't matter if you pair a company that doesn't make the part with the part's cell as long as the pairing is clicking on each cell will display a different company.

Key Deliverables to Keep in Mind:

- TableView Implementation in PartsViewController, including Custom Cell (since you have to display an image too)
    - You should hard code data for at least 5 computer parts
- Passing Data Between PartsViewController and CompaniesViewController when a cell in the tableview in PartsViewController is tapped
- Animation of 1 company imagesflying into the screen when the CompaniesViewController Appears
- Making Sure the CompaniesViewController is presented modally, and has a NavBar at the top when displayed. There should be a done button that dismisses the current VC and takes the user back to the PartsViewController.

Mobile Developers of Berkeley

Submission Instructions: Please raise your hand when you are done. An instructor will come help you submit. You will zip your assessment and upload it. You may not leave until you have submitted. Please check that your upload has gone through. If you accidentally don't submit or you accidentally submit a blank project, you will get a 0 for that part.