COS 333
Spring 2015

Partner 1 : William Bertran
Partner 2 : Gregory Leeper
Partner 3 : Nicholas Pai

FLOCK

Find your flock

Requirements

Operating System: iOS 8

Developing Environment: Xcode 6.3.2 (Requires Mac OS 10.10)

Language: Swift

Compiler: Swift Compiler (through Xcode IDE)

Frameworks: Parse SDK and Facebook SDK

Database: Parse

Website Implementation: HTML, CSS, Javascript, Twitter Bootstrap

Backend

We initially decided on having four different classes, but eventually eliminated two of them. The descriptions of each class and the reasons for keeping or removing them are listed below.

User Class: A user object would consist of their username, facebook ID, full name (on facebook), email, and their location. Users create their account through a Facebook login button on the first page, and all of the necessary data is scraped off of their facebook account to create their Parse user object.

Activity Class: Each activity object was basically a connection between a user and a friend. When a user object was created an activity object was created for each friend on Facebook who also had a Flock account. This allowed to iterate through Parse to find all existing friendships, but was giving us problems when creating new users among other things. So, instead of using this to track friendships on Flock, we simply compare the Facebook IDs of the users friends to the Facebook IDs of all the users in our Parse database.

Club Class: Our initial design involved a club object which would keep track of the attendance of each club; every time a user left a club, the attendance would be decremented by one, and every time they entered a

new club, the attendance would be incremented by one. However, this often caused problems when people would enter a club at the same time, so we decided to eliminate this class and find the attendance by iterating through the user objects.

Image Class: contains an image file and the club name of where it was taken. This allows us to query for images taken in a club which we use in the Clubs Page.

Implementation

Home Page (ViewController.swift): This is simply the homepage which controls updating location through the use of the locationManager instance variable. Again, since this is always on the stack when a user is signed in, the locationManager will constantly be updating. We also implemented the locationManager such that it continues to update location even when the app is running in the background.

Friends Page (ViewFriendsViewController.swift): This file contains a UITableView which is provided by Apples UI-Kit and allows the device to display data in the table which we fill up with friends names and locations. To obtain a list of friends, we iterate through each Facebook friend and compare their IDs to those stored on our Parse database, where we also store their location. We then take all of the matches and add those names and locations to the table. We also added a pull down refreshing function which re-queries for friends locations.

Map Page (MapPageViewController.swift): This file contains an MKMapView provided by Apples Map-Kit. We create custom objects called custom point annotations for each club. Then, we iterate through each Parse user and update the friends number and the total population number accordingly. Once we have all of the friends and population numbers, we add each to the appropriate custom point annotations which are then printed on the map at the specific coordinates for each club (defined in constants.swift).

Club Page (1/2) (ViewClubsViewController.swift): This file contains a collection view (similar to a table view). We query for the friends and population numbers (as we did on the map page) and then add these numbers to the appropriate collection view cell, all of which are then displayed as a list of images with titles. This also has a pull to refresh function similar to that of the Friends Page (see above). Clicking on a club leads to the following document:

Club Page (2/2) (ClubDetailViewController.swift): The number of friends and total population at a club are passed to here from the previous page and then displayed. Then we iterate through the images stored

on Parse (see image class) and the pictures associated with that club are then displayed at the bottom of the page.

Side Panel (1/2) (SidePanelViewController.sift): This file has a table view which is filled with NavItem objects (see below) which represent a button to each page along with a logout button. The logout button is provided for by the Facebook SDK.

Side Panel (1/2) (NavItem.swift): This simply contains the title and an optional image which would appear next to each button. We did not include any images.

Constants (Constants.swift): This page is where we keep track of all of the existing clubs. This page makes it extraordinarily convenient to add or remove locations where we want to keep track of attendance. This file contains the coordinates of each club and an array of all of the names. To add a club, simply increment the NUM_CLUBS variables and then add the coordinates of the new club to the list and then add the name to the array.

Miscellaneous (1/3) (AppDelegate.swift): This file is a file that is standard for every application. It initializes all of the frameworks required by our app and runs functions on start-up and shut-down.

Miscellaneous (2/3) (Main.storyboard): This is the what allows us to use MVC (Model View Controller) architecture. This allows us to easily layout views for each page.

Miscellaneous (3/3) (Images.xcassets): This file contains all of the images used by our app such as our logo. The only images that are not stored here are the ones stored on Parse for each Club Page.