CSULB Friend Finder
Release 2

Armando Zavala, **David** Garcia, **Gustavo** Yepes, **Kane** Keller, **Miguel** Navarro, **Nick** Colburn, **Tan** Tran

Features Implemented

Main interface:

- **Friends list**: friends added onto your Friend Finder account.
- **CSULB Map with friends around you**
- Click on a friend in friends to **view his/her profile.**
    - Profile has friend’s map pin and status.

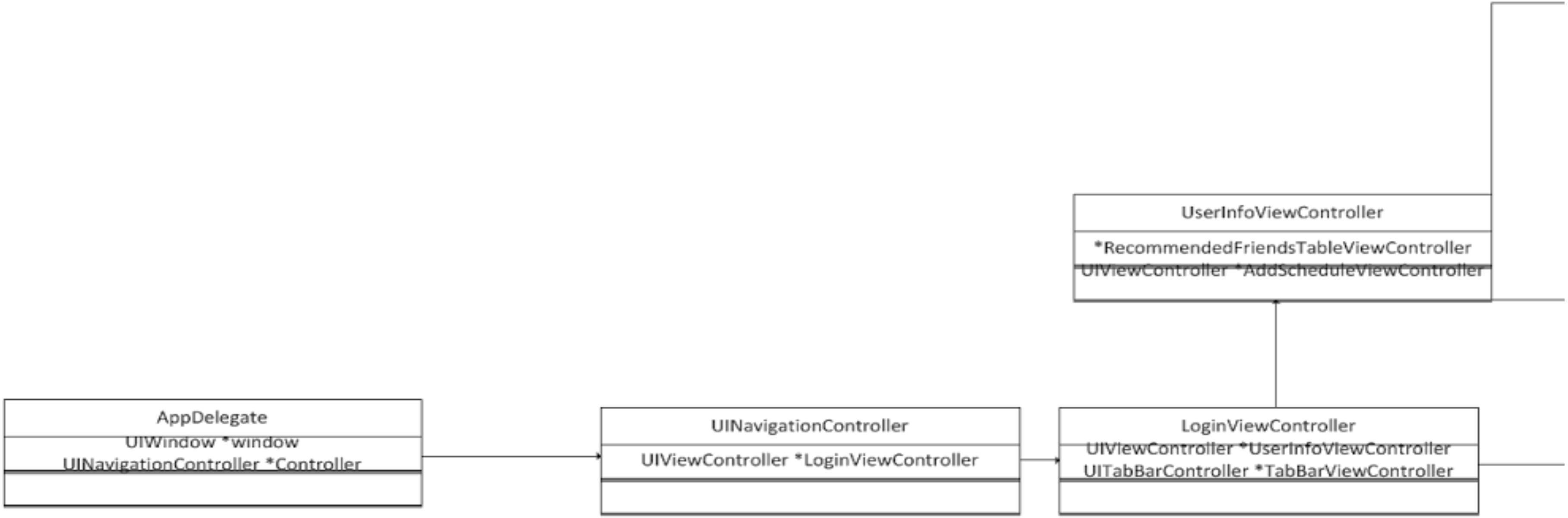
Plans for Next Release

- **Navigate** to a selected friend
  - Not turn-by-turn, but compass like Yelp
- **Update location** and status
- **Poke** friends to give them a notification

Plans for Final Release

- Update my location
- Update profile
  - Add FB profile photo, change email, etc
- Change privacy settings using slider

UML Class Diagram



UserInfoViewController

*RecommendedFriendsTableViewController
UIViewController *AddScheduleViewController

AppDelegate

UIWindow *window
UINavigationController *Controller

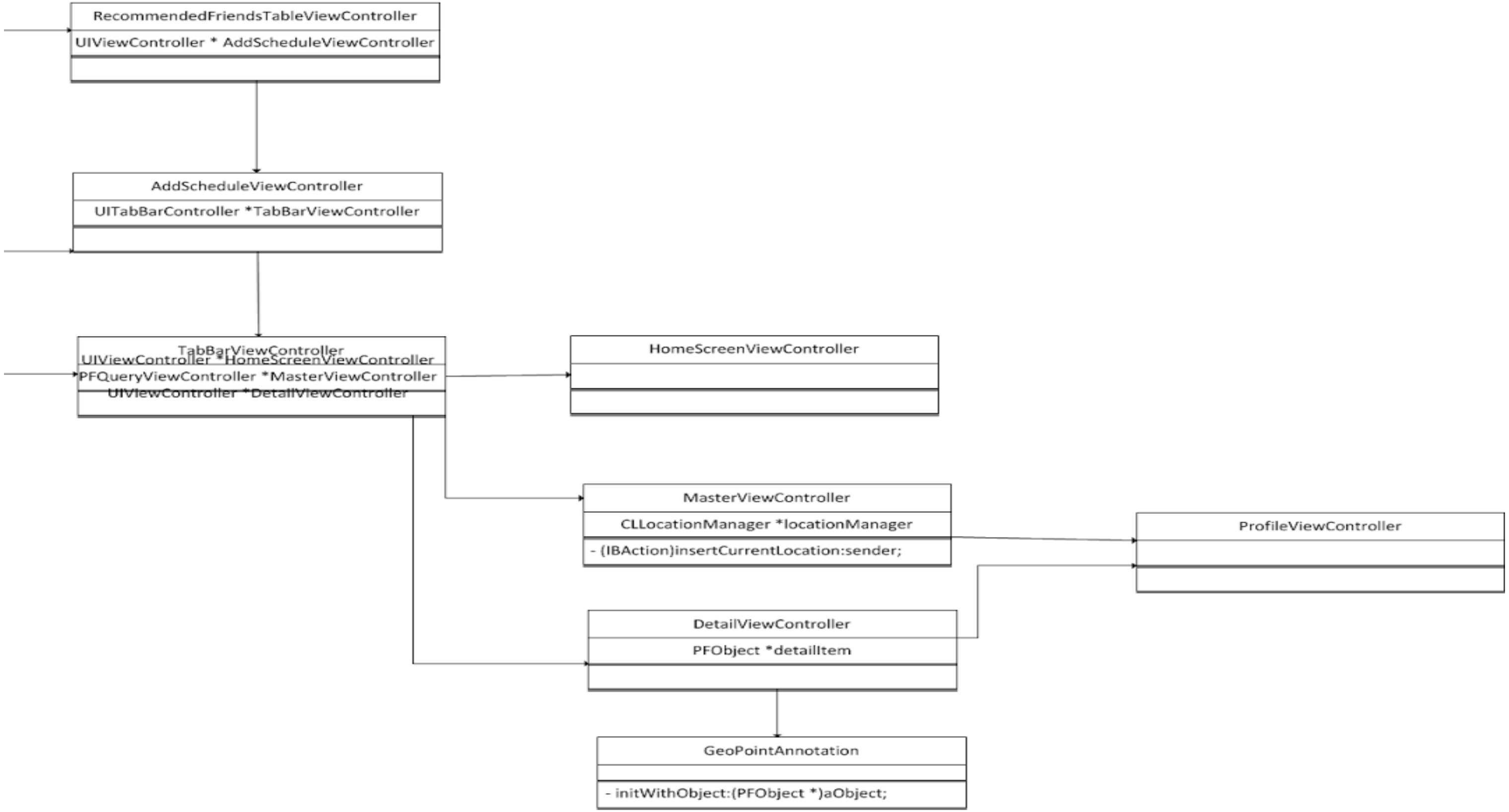
UINavigationController

UIViewController *LoginViewController

LoginViewController

UIViewController *UserInfoViewController
UITabBarController *TabBarViewController

UML Class Diagram



RecommendedFriendsTableViewController

UIViewController * AddScheduleViewController

AddScheduleViewController

UITabBarController *TabBarViewController

TabBarViewController

UIViewController *HomeScreenViewController

PFQueryViewController *MasterViewController

UIViewController *DetailViewController

HomeScreenViewController

MasterViewController

CLLocationManager *locationManager

- (IBAction)insertCurrentLocation:sender;

ProfileViewController

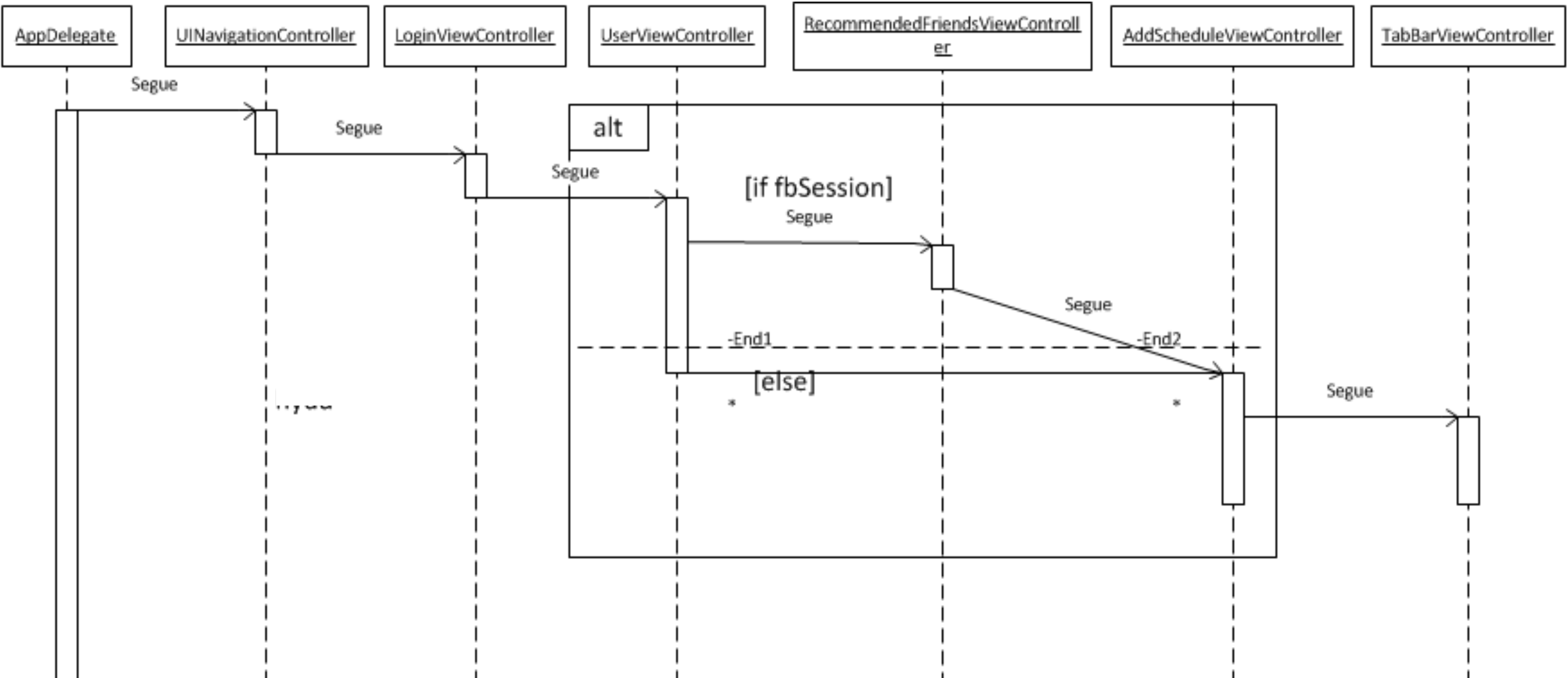
DetailViewController

PFObject *detailItem

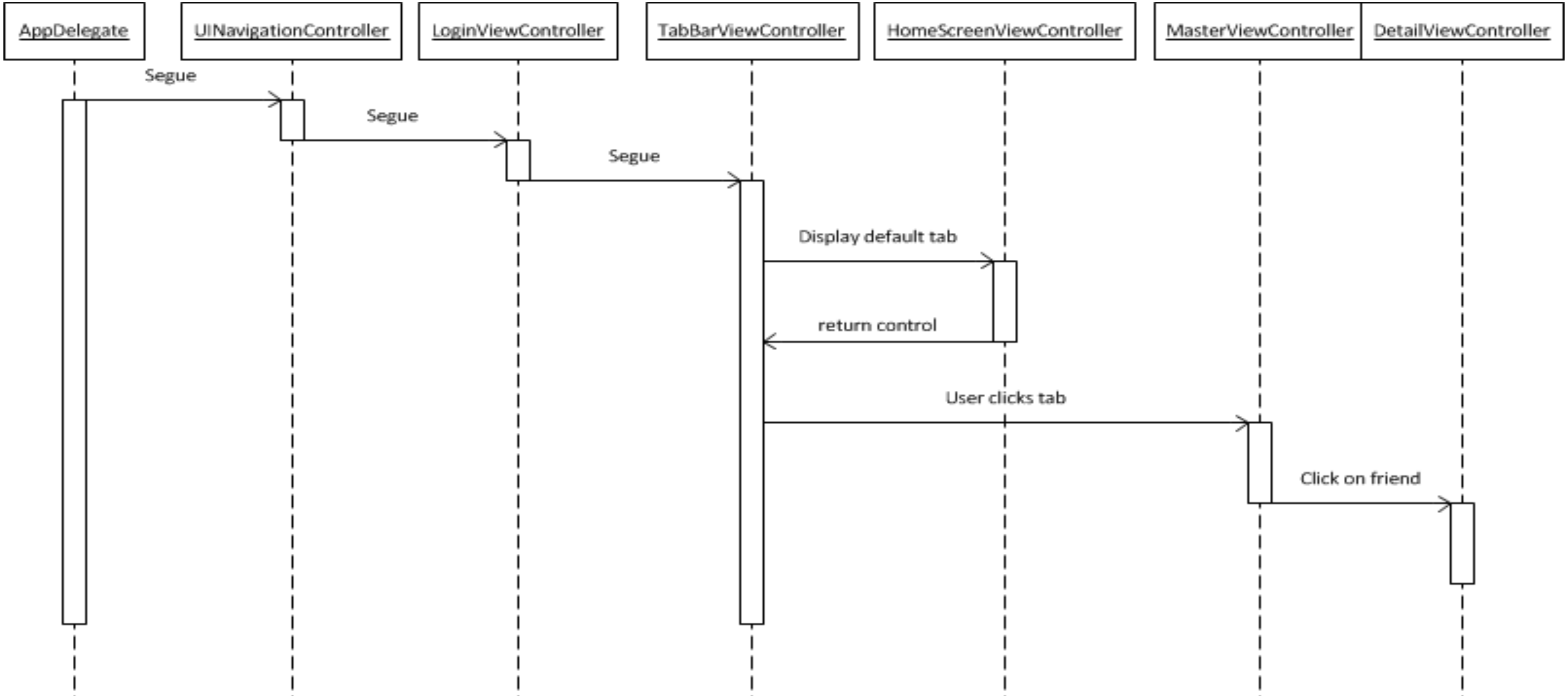
GeoPointAnnotation

- initWithObject:(PFObject *)aObject;

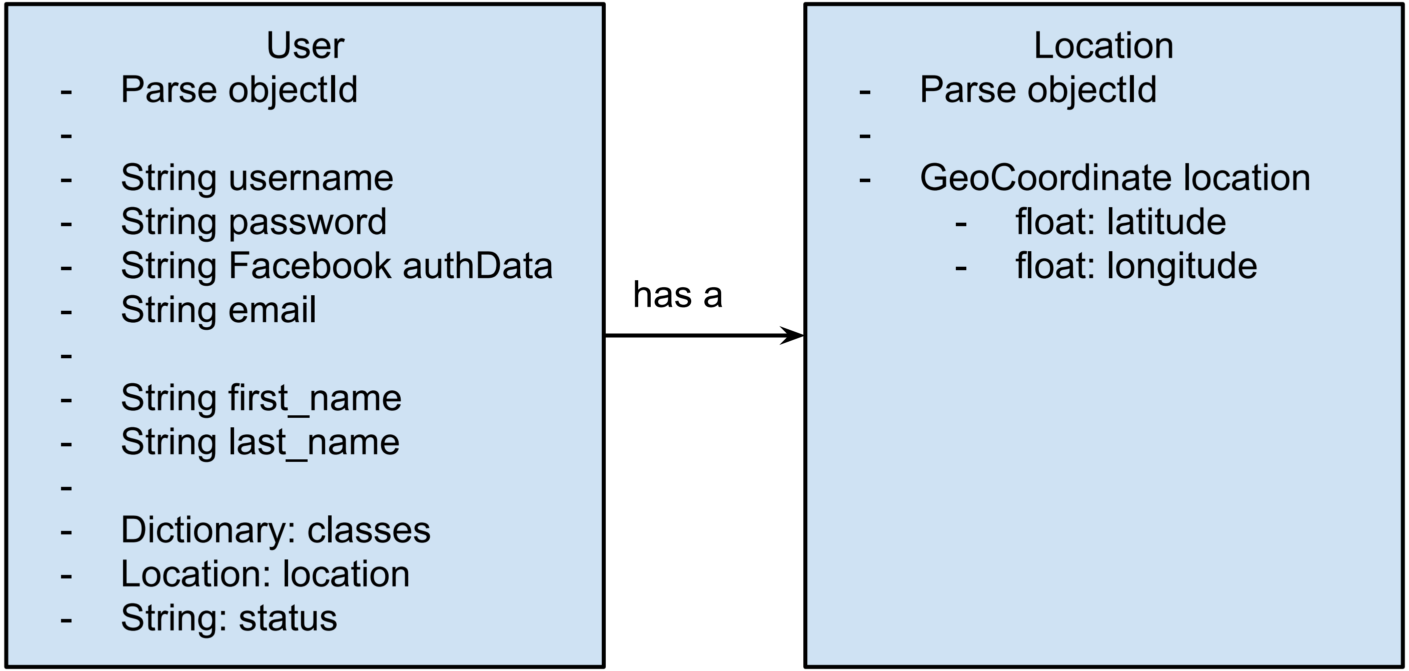
UML Sequence Diagram (Signup)



UML Sequence Diagram (View Friend)



UML Entity-Relationship Diagram



User

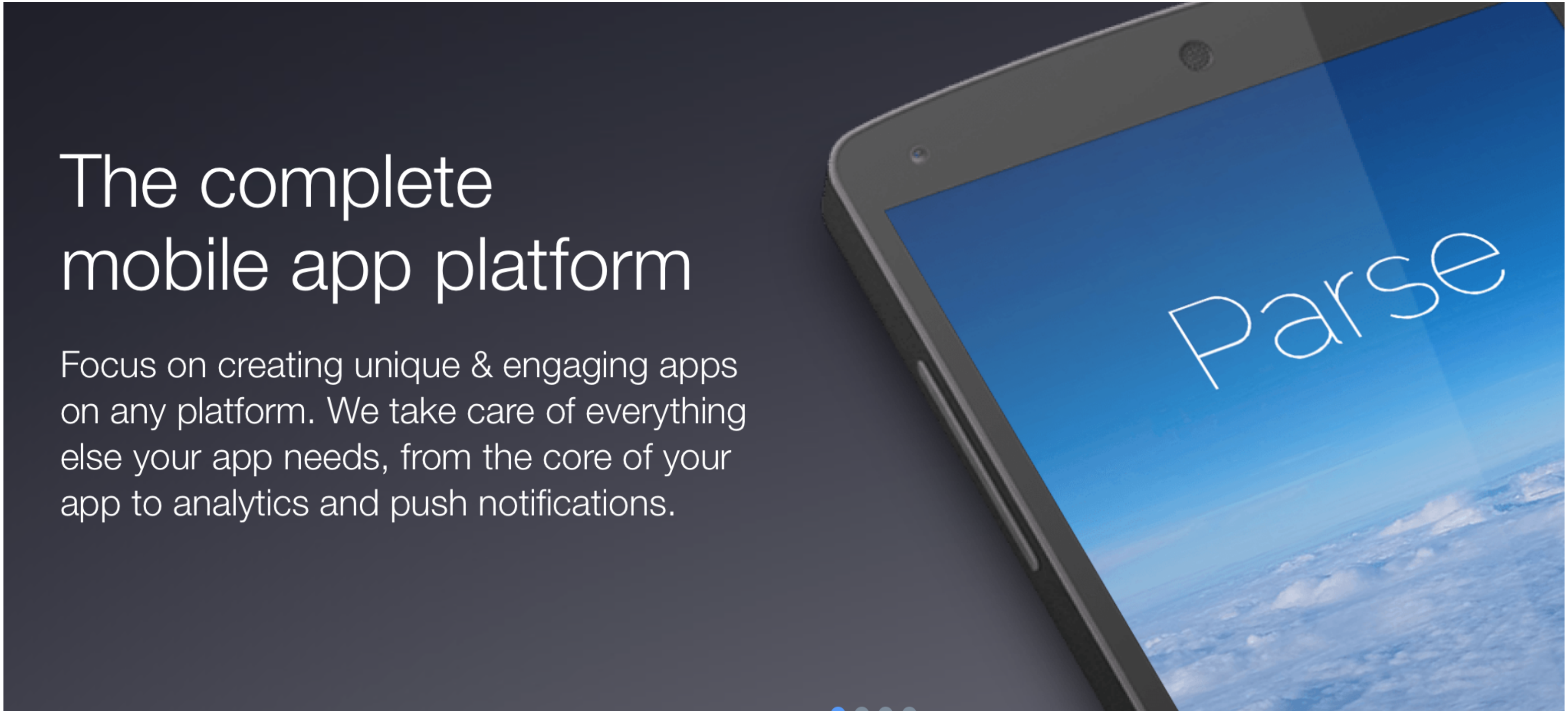
- Parse objectId
-
- String username
- String password
- String Facebook authData
- String email
-
- String first_name
- String last_name
-
- Dictionary: classes
- Location: location
- String: status

has a

Location

- Parse objectId
-
- GeoCoordinate location
    - float: latitude
    - float: longitude

Parse Framework Overview

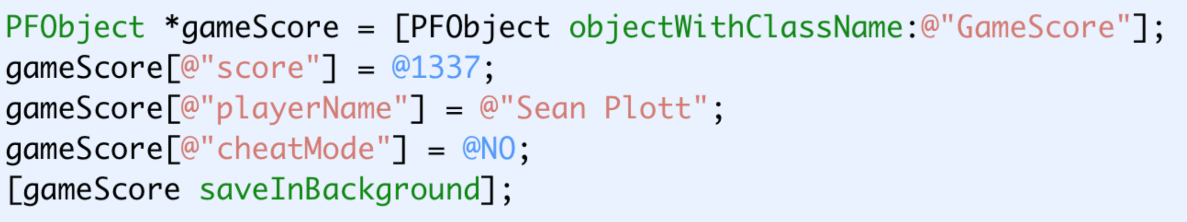


The complete mobile app platform

Focus on creating unique & engaging apps on any platform. We take care of everything else your app needs, from the core of your app to analytics and push notifications.

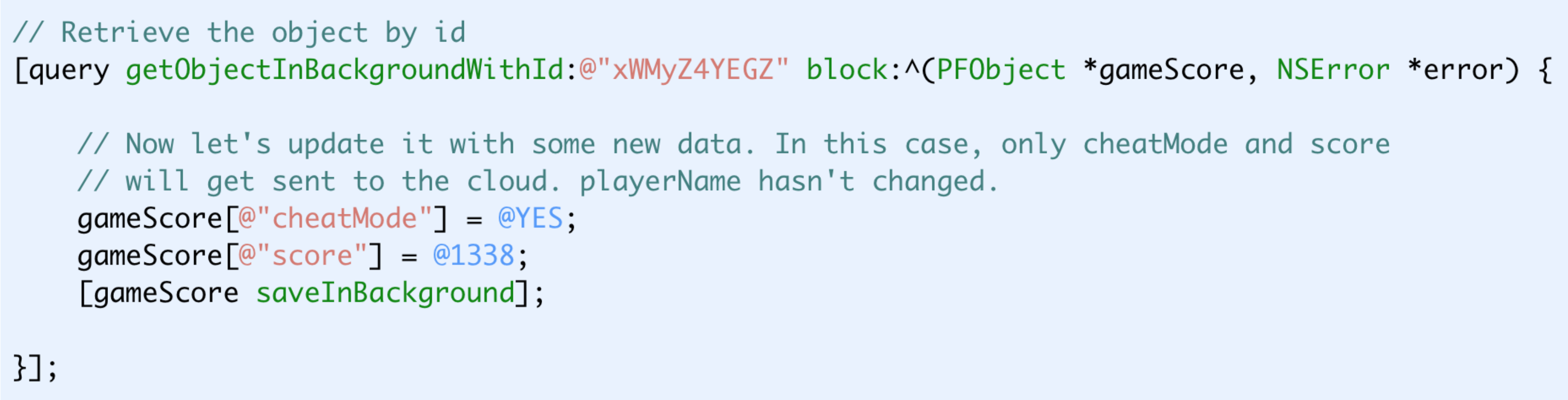
Parse Framework Overview

- Every Parse class subclasses PFObject, which stores JSON data.



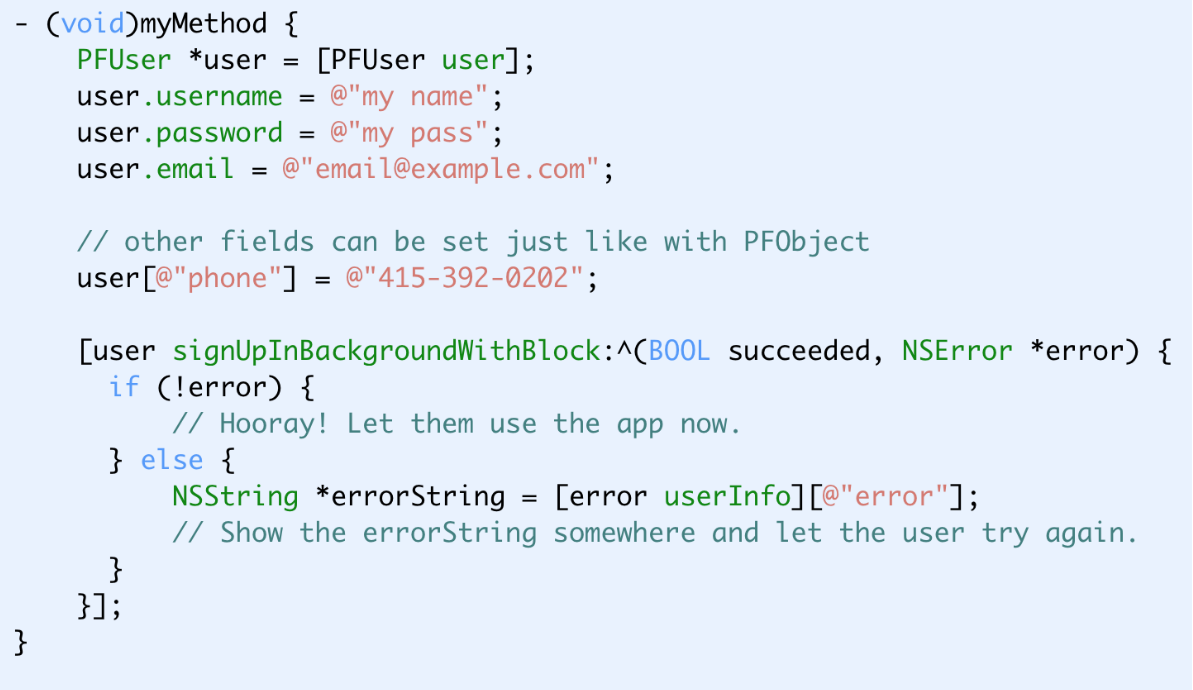
```
PFObject *gameScore = [PFObject objectWithClassName:@"GameScore"];
gameScore[@"score"] = @1337;
gameScore[@"playerName"] = @"Sean Plott";
gameScore[@"cheatMode"] = @NO;
[gameScore saveInBackground];
```

  - Retrieve objects by id or by query:
  -



```
// Retrieve the object by id
[query getObjectInBackgroundWithId:@"xWMyZ4YEGZ" block:^(PFObject *gameScore, NSError *error) {

    // Now let's update it with some new data. In this case, only cheatMode and score
    // will get sent to the cloud. playerName hasn't changed.
    gameScore[@"cheatMode"] = @YES;
    gameScore[@"score"] = @1338;
    [gameScore saveInBackground];

}];
```

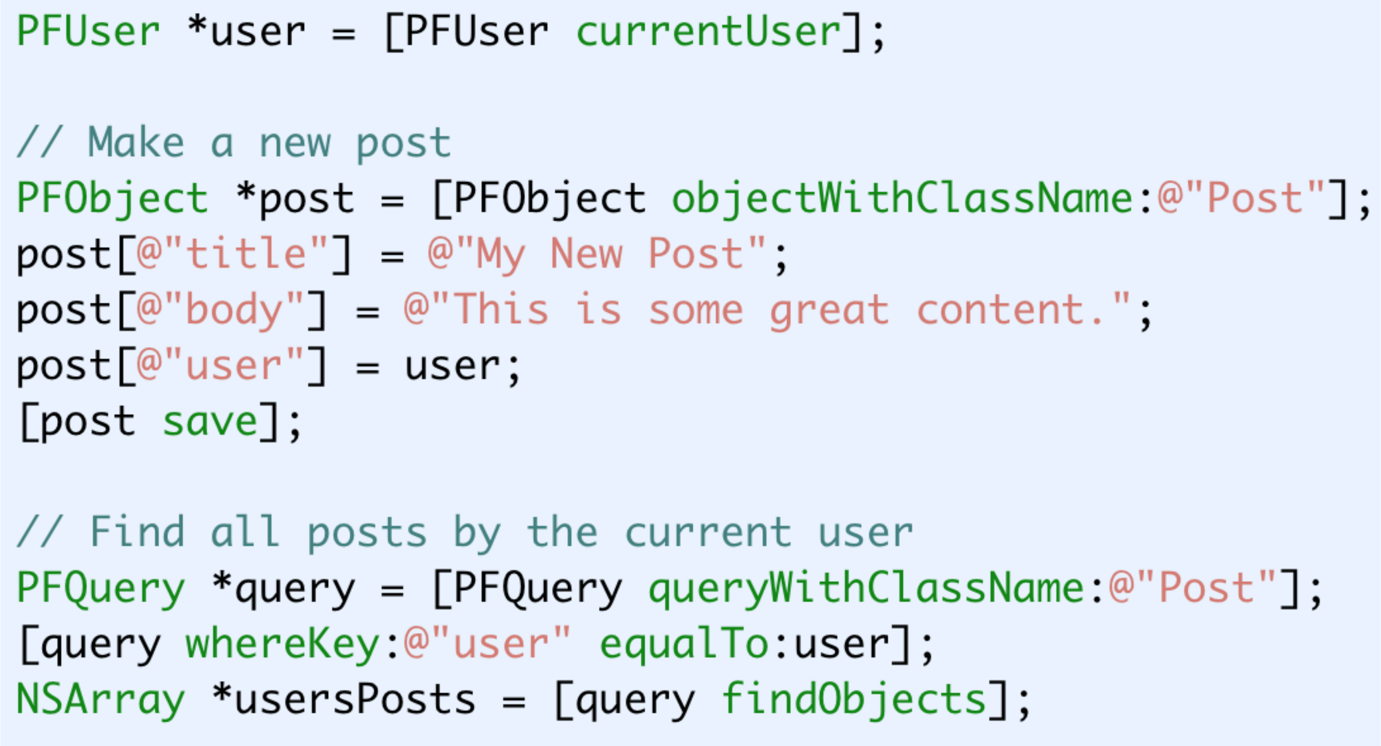
Parse Framework Overview: PFUser

- PFUser is a **convenient** PFObject with **identity provider integration** and **login/signup methods**:
- 

```
- (void)myMethod {
    PFUser *user = [PFUser user];
    user.username = @"my name";
    user.password = @"my pass";
    user.email = @"email@example.com";

    // other fields can be set just like with PFObject
    user[@"phone"] = @"415-392-0202";

    [user signUpInBackgroundWithBlock:^(BOOL succeeded, NSError *error) {
      if (!error) {
          // Hooray! Let them use the app now.
      } else {
          NSString *errorString = [error userInfo][@"error"];
          // Show the errorString somewhere and let the user try again.
      }
    }];
}
```

Parse Framework Overview: PFUser

Query PFUsers:

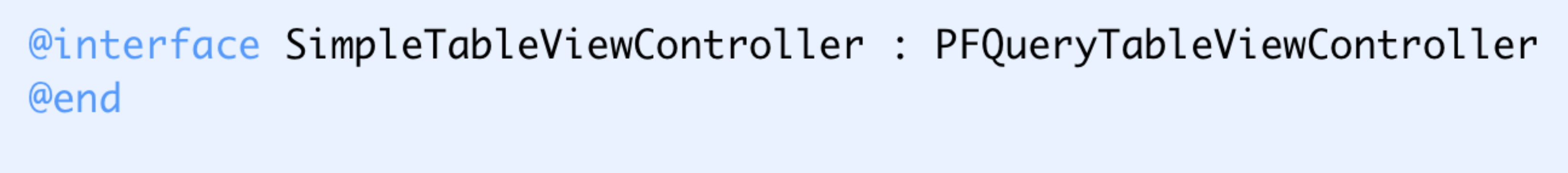


```
PFUser *user = [PFUser currentUser];

// Make a new post
PFObject *post = [PFObject objectWithClassName:@"Post"];
post[@"title"] = @"My New Post";
post[@"body"] = @"This is some great content.";
post[@"user"] = user;
[post save];

// Find all posts by the current user
PFQuery *query = [PFQuery queryWithClassName:@"Post"];
[query whereKey:@"user" equalTo:user];
NSArray *usersPosts = [query findObjects];
```

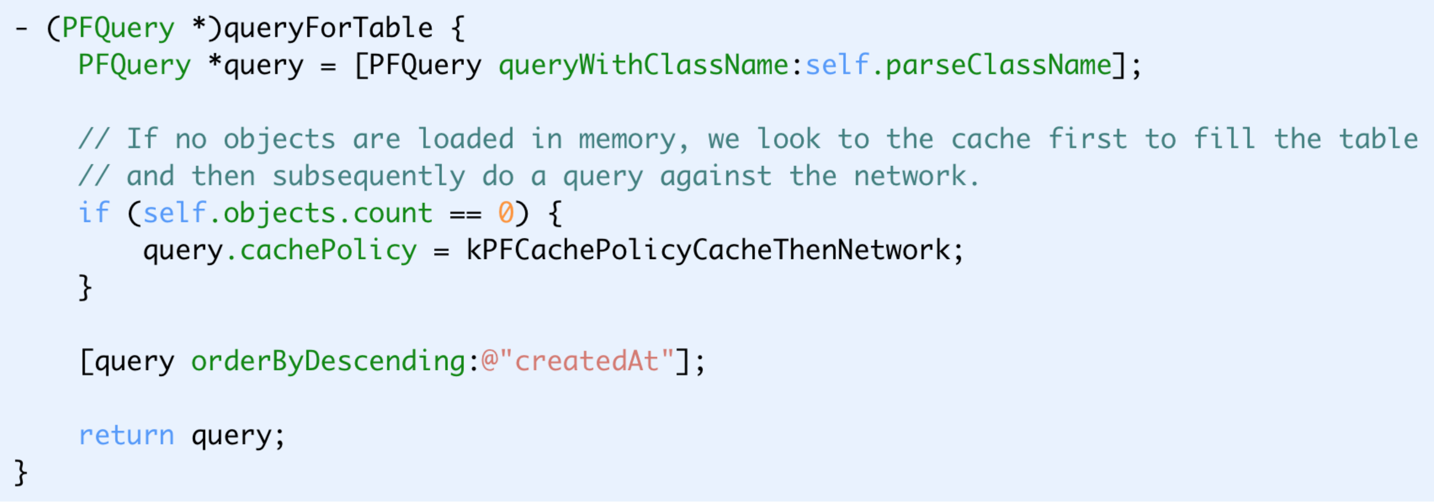
Parse Framework: Convenience Classes

- GeoPoint
- PFLoginViewController
- PFSignUpViewController
- PFQueryTableViewController
  - PFTableViewCell
- PFImageView

Parse Framework: Convenience Classes

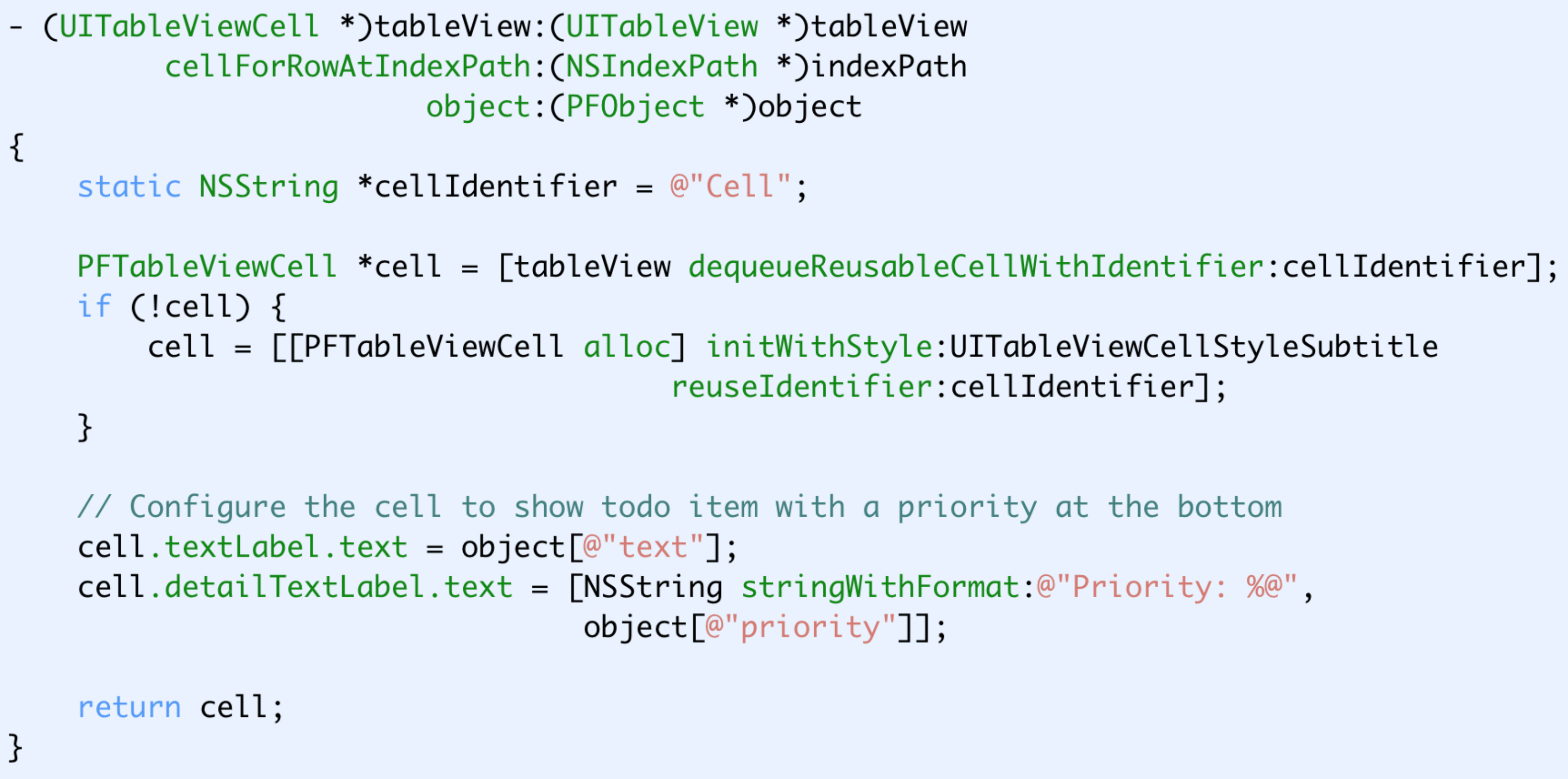


```
@interface SimpleTableViewController : PFQueryTableViewController
@end
```



```
- (PFQuery *)queryForTable {
    PFQuery *query = [PFQuery queryWithClassName:self.parseClassName];

    // If no objects are loaded in memory, we look to the cache first to fill the table
    // and then subsequently do a query against the network.
    if (self.objects.count == 0) {
        query.cachePolicy = kPFCachePolicyCacheThenNetwork;
    }

    [query orderByDescending:@"createdAt"];

    return query;
}
```

Parse Framework: Convenience Classes



```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
                        object:(PFObject *)object
{
    static NSString *cellIdentifier = @"Cell";

    PFTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];
    if (!cell) {
        cell = [[PFTableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
                                      reuseIdentifier:cellIdentifier];
    }

    // Configure the cell to show todo item with a priority at the bottom
    cell.textLabel.text = object[@"text"];
    cell.detailTextLabel.text = [NSString stringWithFormat:@"Priority: %@",
                                 object[@"priority"]];

    return cell;
}
```

Live Demo

Source Code Walkthrough

- Parse
  - PFQuery - Friends List
- Core Locations Framework
  - Location Manager for map features

Summary

Features Implemented:

- Friends List
- CSULB Map View
- User Profiles on Friend Finder

Future Plans:

- Poke Friends
- Navigate between friends
- Privacy Settings

Q&A