

# Hinder

*An easy way to find teammates*

**Discussion 1B**

**Team Name: TBD**

Kimberly Svatos, 604425426

Marshall Briggs, 304417630

Kyle Haacker, 904467146

George Archbold, 604407413

Daniel Berestov, 404441309

Apurva Panse, 504488023

**Repository:** <https://github.com/marshdevs/Hinder>





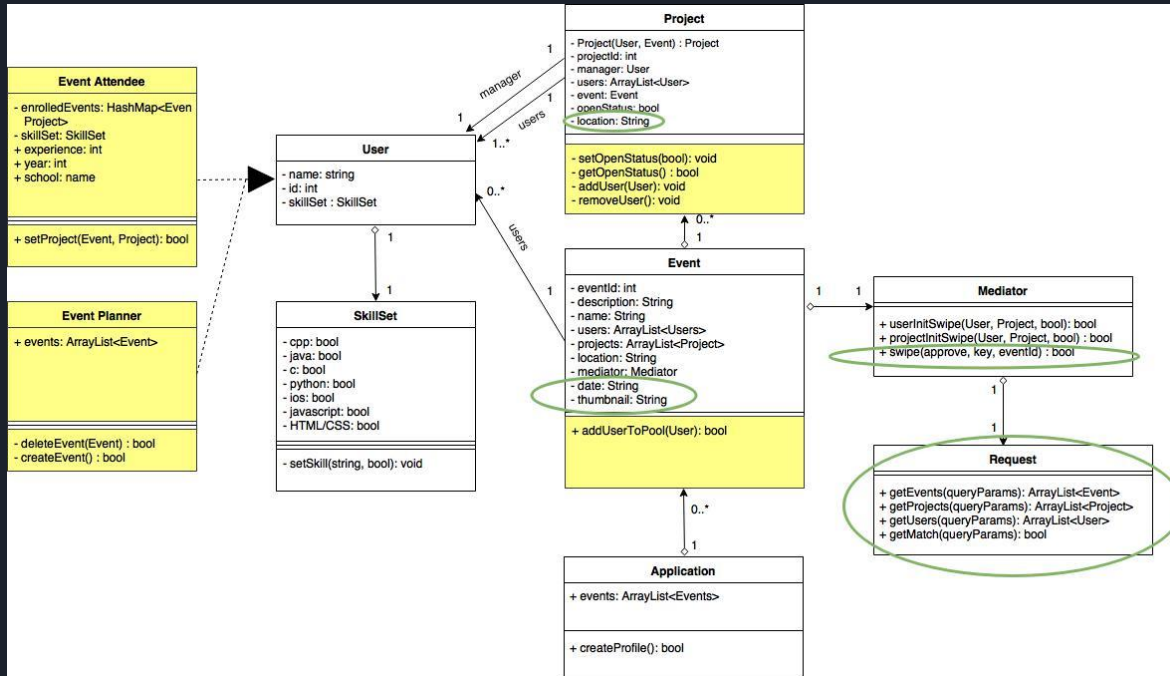
# Overview

Hinder is an application that allows users to easily find teammates based on skills for group projects and hackathons.

We want to help people build **better** groups, ***faster***

# UML - Overview of Changes

- Circled items - added since Part A
- Highlighted items - will be implemented for Part C





# Design

## iOS Components

- **Views**
  - **HinderNavigationBar** (UINavigationController): app's navigation bar, with settings/menu buttons
  - **FeedEventCell** (UICollectionViewCell): for displaying list of events on home screen
  - **HinderCreateProfile**: Facebook user authentication (get name, contact info, picture via Facebook API)
- **ViewControllers**
  - **HomeViewController**: logic for home page, settings + menu buttons, list of events close to user
  - **EventPageViewController**: logic and actions for when a user wants more information about a specific event
- **Models**
  - **Event**: class representing an event, parses json file on initialization to extract data fields
  - **User**: similar structure as Event
  - **Project**: also similar, but includes *addUserToProject* method for matching
  - **Mediator**: handles logic of users swiping on projects (and vice versa)
  - **Request**: enables app to communicate with server, issue queries to database



# Design

## Node Server

- **App.js:** server endpoint for all database actions (query, create, delete, update)
- **Request.json:** info for communicating with server
  - How to receive data from server (schemas for Event, Project, User, Match)
  - How to send data to server (json structure for all server calls)
- **Package.json:** all necessary information for powering node server
  - Project documentation
  - URL for homepage
  - Git repository
  - App dependencies

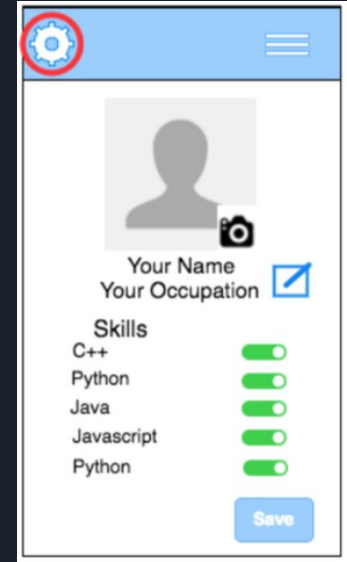
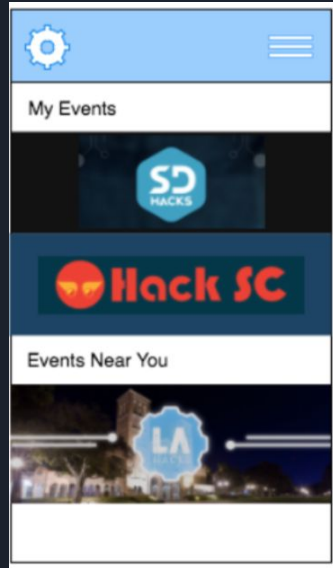
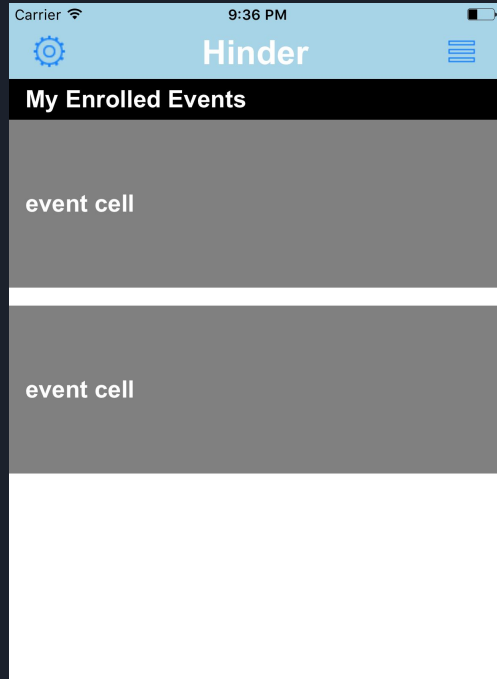


# Design

## Dynamo

- Fast and flexible NoSQL database
- **App.js** acts as sole gateway between application and database
  - Extra security measures, at the (slight) cost of performance
  - May use DynamoDB Accelerator in future to improve performance
- Four tables, values stored as JSON objects
  - **hinder-events**: store instances of Event model
  - **hinder-projects**: store instances of Project model
  - **hinder-users**: store instances of User model
  - **hinder-matches**: store swipe interactions between users and projects
- Interact with tables via GET, POST, PUT, DELETE

# User Interface



# API Overview

- Currently, APIs reside in **Mediator.swift** and **Request.swift**
  - **Mediator:** *userInitSwipe, projectInitSwipe, swipe*
  - **Request:** *getMatch, getEvents, getProjects, getUsers*
- Use Markup, Quick Help in XCode to create and view API descriptions
  - Jazzy creates documentation in HTML (similar to Javadoc)

```
/**
 A swipe action.

 This function takes in swipe parameters
 and queries the database to see if a pair
 of matching User/Project IDs exists.

 - parameter approve: Left (false) or right (true) swipe
 - parameter key: The ID to search the database
 - parameter eventId: The corresponding ID of a previous swipe

 - returns: A true or false indicating existence of the match
 */
func swipe(approve: Bool, key: String, eventId: String) -> Bool {
    let direction = approve ? "/right/" : "/left/"
    return Request.getMatch(params: "match/" + eventId + direction + key)
}
```

## Quick Help

Declaration `func swipe(approve: Bool, key: String, eventId: String) -> Bool`

Description A swipe action.

This function takes in swipe parameters and queries the database to see if a pair of matching User/Project IDs exists.

Parameters

- `approve` Left (false) or right (true) swipe
- `key` The ID to search the database
- `eventId` The corresponding ID of a previous swipe

Returns A true or false indicating existence of the match

Declared In `Mediator.swift`

## Mediator

```
class Mediator: NSObject
```

Matches users to projects based on swipes.

```
swipe(approve:key:eventId:)
```

A swipe action.

This function takes in swipe parameters and queries the database to see if a pair of matching User/Project IDs exists.

## Declaration

```
SWIFT
```

```
func swipe(approve: Bool, key: String, eventId: String) -> Bool
```

## Parameters

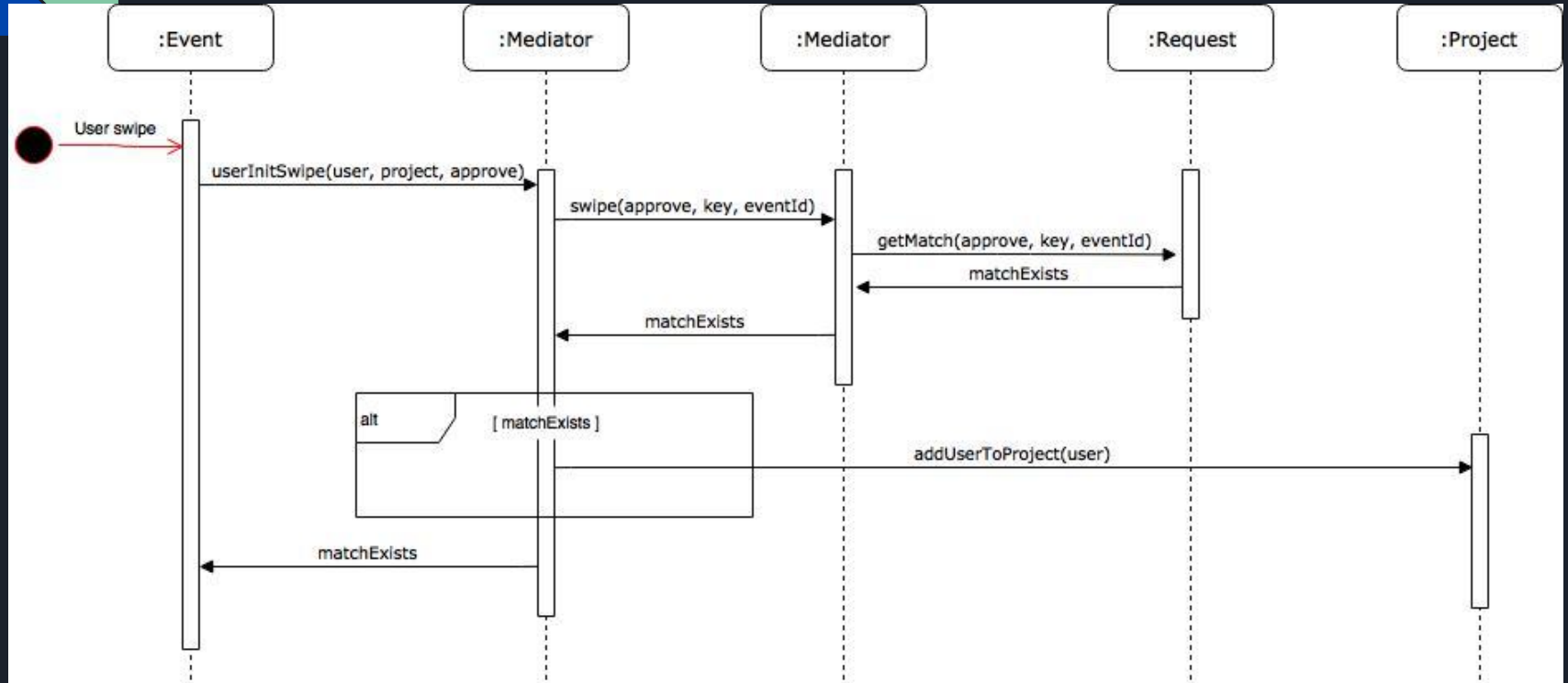
<code>approve</code>	Left (false) or right (true) swipe
<code>key</code>	The ID to search the database
<code>eventId</code>	The corresponding ID of a previous swipe

## Return Value

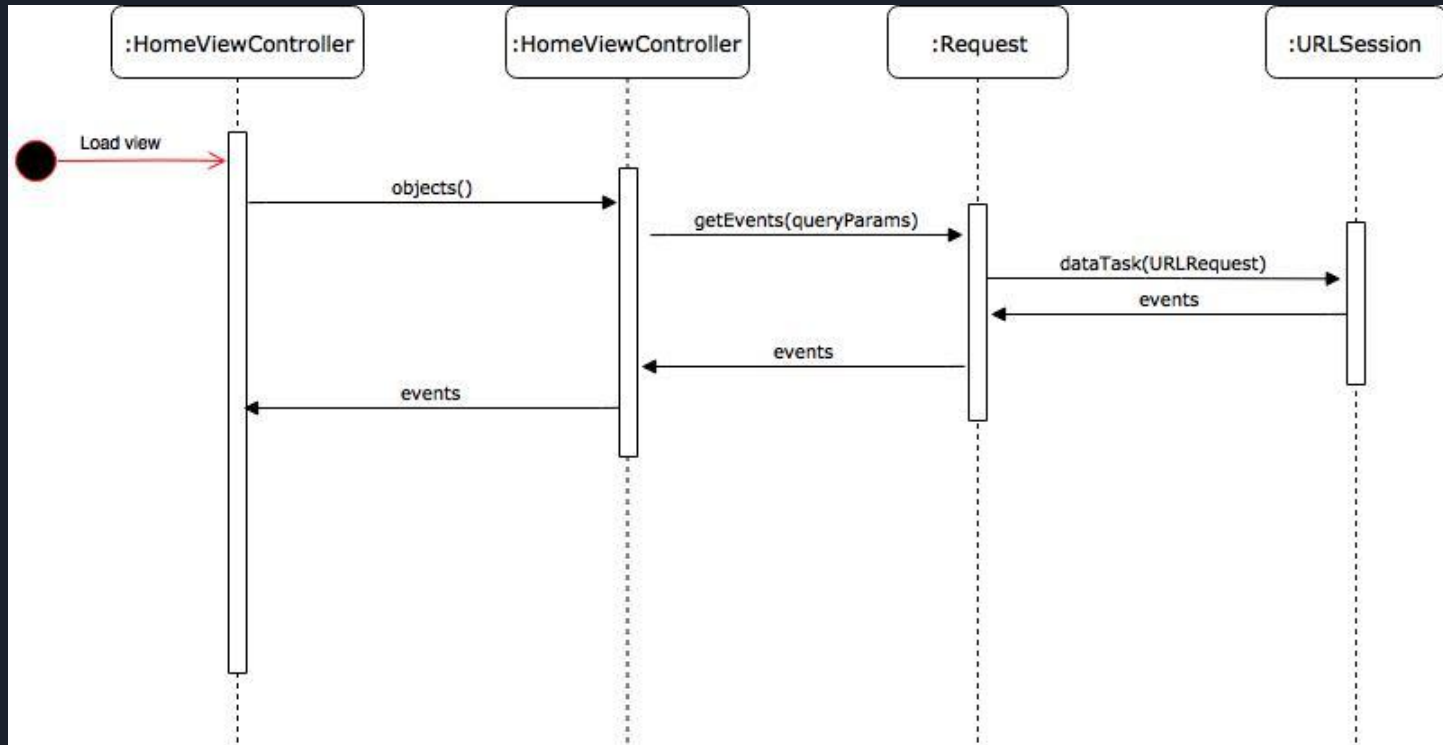
A true or false indicating existence of the match



# Sequence Diagram - Mediator



# Sequence Diagram - Request



# IH Principle - Possible Evolution Scenarios

	Application View	Mediator	Request
<i>Filter potential swipe choices based on location or time</i>		<b>X</b>	
<i>Add additional user profile information (e.g., school)</i>	<b>X</b>		<b>X</b>
<i>Add additional event information (e.g., see event coordinator, companies attending event)</i>	<b>X</b>		<b>X</b>
<i>Add smooth and intuitive swiping gestures</i>	<b>X</b>		
<i>Limit the number of times users and projects can re-swipe on one another</i>		<b>X</b>	
<i>Allow users to upload resumes and/or transcripts</i>	<b>X</b>		<b>X</b>



# Testing

- Test cases written in ViewControllerTests.swift
- Currently, not many conditional statements / paths to test
  - Tests focus on whether objects are created and updated correctly in application
  - Example: test whether HomeController inserts a cell *with* a header at the top of the event list, and that all remaining cells do *not* have headers
- Future test cases:
  - Verify correct behavior for successful / unsuccessful user login via Facebook
  - Verify Facebook picture, profile information matches what is stored in AWS
  - Mediator: check correctness of paths for when there is a match / when there isn't a match
  - Request: check correctness of data returned for getEvents(), getProjects(), getUsers()

```
if(controller.collectionView.cellForItem(at: IndexPath(row: 0, section: 0)) != nil) {  
    XCTAssertTrue(((controller.collectionView.cellForItem(at: IndexPath(row: 0, section: 0)) as?  
        FeedEventCellWithHeader) != nil))  
}
```