Draft Project - MunchEase

Team: Chandra’s Angels

<table>
<thead>
<tr>
<th>Member</th>
<th>Email</th>
<th>Github</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey Tellew - Lead</td>
<td><a href="mailto:jefftellew@gmail.com">jefftellew@gmail.com</a></td>
<td>jefftellew</td>
</tr>
<tr>
<td>Austin Quinn</td>
<td><a href="mailto:austinquinn@ucsb.edu">austinquinn@ucsb.edu</a></td>
<td>austinquinn</td>
</tr>
<tr>
<td>Anthony Hall</td>
<td><a href="mailto:anthonyhall.school@gmail.com">anthonyhall.school@gmail.com</a></td>
<td>ahal99</td>
</tr>
<tr>
<td>Suds Kannan</td>
<td><a href="mailto:sudarshan.kannan99@gmail.com">sudarshan.kannan99@gmail.com</a></td>
<td>sudarshankannan</td>
</tr>
<tr>
<td>Yagnya Patel</td>
<td><a href="mailto:yagnyapatel@gmail.com">yagnyapatel@gmail.com</a></td>
<td>yagnyapatel</td>
</tr>
</tbody>
</table>

Relevant Links

GitHub: [https://github.com/yagnyaPatel/Chandras-Angels](https://github.com/yagnyaPatel/Chandras-Angels)
Trello: [https://trello.com/b/AXdpdLP6/chandras-angels](https://trello.com/b/AXdpdLP6/chandras-angels)
Travis-CI: [https://travis-ci.org/yagnyaPatel/Chandras-Angels](https://travis-ci.org/yagnyaPatel/Chandras-Angels)
Slack: [https://cs48s19.slack.com](https://cs48s19.slack.com)
Burndown Chart

SPRINT 1: To Do, In Progress, Ready for Testing and Done

SPRINT 2: To Do, In Progress, Ready for Testing and Done
About our project

What problem are we solving?

Oftentimes when spending time with friends, we find ourselves at a standstill, unable to determine where exactly to eat. One friend might suggest eating Chinese food, another repeatedly suggests going to In-N-Out, and three other folks are probably churning out wildly differing alternatives. The point is, by the time you get everyone to agree to come together and methodically determine the group’s course of action, you’ve spent about 15 or more minutes arguing uselessly. Essentially, our app allows folks to receive recommendations on their phone based on search queries and location, and allows members of the group to then vote on all selected options. In addition, each user can store their preferences of food and favorite places that we can then use to generate suggestions. Thus, we streamline the process of idea suggestions and idea selections to improve efficiency, and save both time and a headache.

Why is the problem important?

When people gather together to spend time with each other, they don’t really want to spend time bickering over what to do. Our app aims to cut down on the amount of time spent deciding what to do, thus allowing people to spend more of their time enjoying each others’ company. It is easier to get people to vote on their phones, rather than trying to get everyone’s attention towards trying to raise their hands and vote, given that people always try to throw in one last addition or inevitably derail the process. In addition, we aim to suggest places or things to do that folks normally wouldn’t have thought of, provided they are related to the search parameters/location of the user.

How is the problem solved today?

Currently, Yelp actually does a decent job at suggesting restaurants to visit based of a query and geographic proximity; thus, we will be integrating the Yelp API into our application to obtain data and customer reviews of restaurants. However, there is currently no solution that allows a group of people to pool together suggestions and vote for their preferences.

Outcome

The project outcome will be an Android application that allows a user to create a group and add people to it. Users can search for events and places to go to through the app, and are able to view options that suit the user’s desires. Users can then push any desired activities to the rest of the group, and members of the group can vote for the things they want to do.
Creating Our Solution

Implementation Platform and Technologies

The Android application frontend will be built using Java and various associated libraries on Android studio. To obtain data (customer reviews, star ratings, price, etc.) about suggested establishments, we will use the Yelp fusion API. Firebase will be used as the backend to store group, voting, and idea suggestion information, as well as to relay information between devices.

Project Milestones

First Sprint (4/15 - 4/26)

**Milestone Goal:** Users in a party can search for and remotely add restaurants to a list
- Apply the Yelp Fusion API to the Android frontend to search for and select songs, create a list of preferences stored in the Firebase backend.

Second Sprint (4/29 - 5/10)

**Milestone Goal:** Users can vote for their preferred restaurants
- Create a voting interface that updates the tally on the host device

Third Sprint (5/13 - 5/24)

**Milestone Goal:** A user can host a new group and other users can join via add code.
- Allow the creation of multiple independent “party” instances in the backend.
- Test multiple instances of “parties” and member devices.
Architecture Overview

Yelp API

Restaurant queries

Restaurant information

Database

Current party information

Upvotes and downvotes
New restaurant suggestions
Joining and leaving party

User
### Detailed Design

#### UML Diagram

**Restaurant**

- alias: String
- Name: String
- url: String
- rating: String
- reviewCount: int
- displayAddress: String
- latitude: double
- longitude: double
- price: String
- schedule: RestaurantSchedule

- + accessors for all private members
- + parseFromJson()
- + convertToJson()
- + updateVoteCount()

**RestaurantSchedule**

- isOpen: boolean
- daySchedules: DaySchedule[7]

- + getIsOpen()
- + getOpeningTime() // of today
- + getClosingTime() // of today

**DaySchedule**

- day: string
- start: string
- end: string
- isOvernight: boolean

- + accessors for all private members
Party

- ID: String
- host: String
- members: String[]
- restaurants: Restaurant[]

+ addMember()
+ addRestaurant()
+ openMap()
Sequence Diagrams

p:Party

r:Restaurant

getName()

name

upvote()

User

MunchEase

Opens app

Shows home screen

Taps “Create”

Creates new party, shows “Share” screen

Taps “Add Restaurant”

Opens SearchActivity
Requirements

User Stories

1. **(1) As a User, I should be able to launch the app so that I can use it.**
2. **(2) As a Party Host, I can create a party so that people can join.**
3. **(3) As a User, I should be able to connect to a party so that I can be a party member.**
4. **(4) As a Party Member, I should be able to see the list of restaurants up for selection so that I can interact with them.**
5. **(5) As a Party Member, I should be able to search up restaurants to add in a search bar.**
6. **(6) As a Party Member, I can add restaurants to the list so that they can be voted on.**
7. **(7) As a Party Member, I should be able to vote on restaurants in the list so they are either more or less likely to be chosen.**
8. **(8) As a Party Host, I can specify a backup restaurant so that there will still be a default option if nobody suggests any other restaurants.**

Appendix

1. **App Interface**
   a. Java
   b. Android Studio
2. **Database**
   a. Firebase Firestore
   b. Firebase Realtime Database
3. **Restaurant Information**
   a. Yelp API
   b. Google Maps
4. **Development**
   a. GitHub
   b. Travis-CI