CS32 Final Project: MeetUp

Group Members

- George Rusu
    - Strengths: Extensive experience working with frontend and backend frameworks like React and Django. Experience with deployment, databases, and user login/authentication.
    - Weaknesses: Django is python not Java. Not too comfortable with frontend development - more comfortable with setting up routing, endpoints, and testing for the frontend.
- Ermias Genet
    - Strengths: Front end development (familiar with React, Js, bootstrap, good experience with Html/css), connecting APIs
    - Weaknesses: little SQL experience, not strong with java
- Amin Hijaz
    - Strengths: have some experience working with React(a long time ago), know some Js, html, and Css. Love Learning new stuff
    - Weaknesses: new to Java, databases(I used google firebase with React before but not SQL).
- Hamzah Shah
    - Strengths: Databases (SQL), working with large datasets, functionality testing, Python, communication
    - Weaknesses: Front end development

Overall Project Idea

With everyone's busy schedules, coordinating spontaneous meet-ups with friends on campus can be difficult. Our idea is to create a service to automate the process: it will match friends together considering their preferences (e.g. what activity to meet up for, distance willing to travel, time willing to spend, etc) as well as their location. The core algorithmic complexity comes from using these preferences and location constraints to form meet-up groups via a graph-based matching algorithm. For the sake of this project, we will limit our scope to Brown University students and the Brown University campus.

Features Specification

Feature	Rationale	Brief Description	Anticipated Challenge
Friends list	Ensures that users meet up with people they know (guards against security issues)	Each user has a friends list (friend = mutual connection between two people)	How many friends or mutual friends must be present at an event for the algorithm to let you join? Should this be a preference the user gives or a developer's decision?
Preferences/constraints specification	The service must consider a user's preferences for a gathering as well as practical limitations	Users specify their preferences and constraints for a gathering (e.g. type of gathering, star/end time, current location, location where they next have to be, etc.)	The preferences and constraints are the inputs to our algorithm. How many inputs should our algorithm take in? Which preferences/constraints should we give more weight to? Should that be up to the user?
Matching Algorithm based on Graph Representation	Provides the algorithmic complexity for this project - matching users to a gathering on a map in an intuitive way.	Model users as nodes in graph, edges have costs representing a weighted sum of meeting preferences and location constraints	Do we want to have separate graphs for separate friend circles? And if so, how/when do we reconcile multiple graphs, to open up a gathering to more people?
Databases and User authentication/ login	We must have a database on the cloud containing all info.	Have a users table. Keep track of gatherings happening	How to use cloud databases?
Send confirmation emails to the	1-Don't have people not affiliated with campus showing up for events.	Each person, who signs up must have a valid *.edu email that we need to verify.	How to send unique links or codes and make them expire after a certain amount

university email (confirm users)	2-Safety issues.		of time?

More Specifics on Matching Algorithm

1. Build an undirected graph with people planning to do the same activity
2. Calculate edge values using user preferences and location and time constraints
    a. An edge value is computed as a weighted sum of preferences/constraints with each preference/constraint having a different weight
3. Matching nodes (i.e. users) depending on a certain cutoff for edge values
4. Deciding what to do with people excluded from the matching process
    a. Should we form a second graph for such people and re-match?

User Perspective:

1- People thought that this was a nice way of strengthening friendships.
2- Some Muslims found this to be helpful in organizing communal prayer because emphasized for Muslims to pray their daily prayers in congregation.
3- Someone noted this can be used to organize a study group, a previously unconsidered use case.

Additional Requirements:
1 - A messaging feature, which allows users more flexibility in discussing their potential gathering.
2 - Fleshing out the scope of the meetings/having public meetings (this goes beyond the initial scope of our project).
3 - An interactive map GUI, with pins at the location of the event that you click on for additional information.
4- A mobile app in addition to the website, which is a good idea to work on after CS32.