Stakeholders

- This application is intended to be used strictly by MIT students who wish to meet new people in small groups (<= 4) and explore less frequented MIT dining halls. Our app will help these students achieve their goals.

Tasks

Task List:

Carlos:

a. Request Form HTML
b. Routing Request
c. Integrate work with team members' and deploy MVP
d. Security: Salt/limit and sanitize
e. Revision and integration

Sail:

a. Status HTML
b. Routing Status
c. Integrate work with team members' and deploy MVP
d. Conversation HTML, logic, and schema
e. Revision and integration

Jonatan:

a. Homepage HTML
b. Functions dealing with request matching
c. Integrate work with team members' and deploy MVP
d. Networks HTML, logic, and schema
e. Revision and integration

John:

a. Schemas for User and Request, and Login/Registration HTML
b. Routing Login/Homepage and functions dealing with request status
c. Integrate work with team members' and deploy MVP
d. Recommendation Function for request; Mobile App deployment
e. Revision and integration

Calendar:

- All (a) will be due Sunday November 15th
- All (b) will be due Wednesday November 18th
- All (c) will be due Monday November 23th (MVP due date)
- All (d) will be due Wednesday November 29th

- All (e) will be due Sunday December 6th (final code due)

Risks

Implementation Risks:

- Implementing our web app in a mobile platform might prove to be too difficult.
    - Mitigation: We'll evaluate this idea again a week before the final code is due and decide whether to keep it or not then.
- This project lends itself to some useful features for users (e.g., a feature that recommends times and/or dining halls), as well as there being different ways to implement the server/database logic. As such, when seeking to improve the useful (but not necessary) features or cutting down performance time, our team might spend too much time on these potential complexities.
    - Mitigation: We'll evaluate the feasibility of implementation details as the project progresses and decide whether or not to further pursue them

User Experience Risks:

- Someone might get matched with our service, but they don't actually show up to dinner.
    - Mitigation: We'll implement a system where users who consistently fail to show up to meetings receive lower priority when it comes to matching.
- No people to match with the user's input time and dining hall.
    - Mitigation: We'll implement a recommended time and dining hall for each user to see the popular place and time to eat with peer MIT students.
- Not many people will use the app in MIT community
    - Mitigation: After the app is launched, we will work with MIT dining to get discounts

Minimum Viable Product

Our minimum viable product will include only the core features necessary to make the app useful. From a high level view, this means our MVP should be such that:

- Users should be able to register and login
- Users should be able to submit a request, and be matched someone with compatible preferences and availability

In order to achieve that, the following list outlines the components for our app we deemed to necessities.

- Concepts
    - *Request* - A user is able to fill out a request specifying which times they are available, and which MIT dining halls they'd be willing to eat at.
    - *Status* - A user is able to see the activity of their current request. If no request sent, it will read "No Request Sent." If a request has been matched the user will get an email notification, and Status will read "Matched" and allow the user to see the other user they've been matched with. Otherwise, it will read "Pending."

Issues postponed until final version

- Other Request functionalities
    - A user is able to edit or cancel a current request before a match has been made
- Other Status functionalities
    - A user is able to confirm or cancel a match that has been made.
- Network
    - A user is able to see all the other users they've been matched with and have added connected with. From here, the user should be able to send messages to the users they're connected to.
- Security
    - Limit the number of password attempts a user gets in a certain timeframe
    - Keep hash of password with salt in credential database
    - Sanitize input in components that take in user input
- Recommendation
    - Upon an unmatched request, user will be provided with a list of the more frequent times that other users are available at, and which Dining halls are most visited