11/15/15
TA Meeting Agenda

· 	Got the concept of branching under our belts – that's how we'll mainly be doing our independent work for this project.

· 	Made templates/ handlebars for the greet page, sign in page, registration page, status page, and homepage (post logged in)

· 	Merged all of our branches successfully.

· 	Discussed agenda for tomorrow's TA meeting.

- 	discuss initial design makes sense
- 	is our MVP enough or too minimum?
- 	does team plan look good?
- 	any place we can add API? like email notification and scheduling?

· 	Rescheduled time for Wednesday's meeting.

11/16/15 TA Meeting Minutes

Feedback from the TAs are good. Our initial design is clear and straightforward, and we need to address security issues as pointed out by the TAs. Some key questions answered in the meeting:

Q: What are the important features to be included in the MVP?
MVP: adding friendship is possible, but conversation can be included in the later stage

Q:What are the good API for sending emails?
API: use mandrill for Email

Q: What are the necessary updates in the revised design?
Revised Design: feedback for initial design will be sent out tonight and make change accordingly. Provide alternative designs pro and cons

Q: What is the threat model we need to address for this project?
Security: how to validate mit students

11/23/15 TA Meeting Minutes

Feedback from the TAs are good as we are on the right track to finish our project. Some key questions answered in the meeting:

Q: Is Karma System required?
karma system might not be necessary as penalize users might not be a good idea

Q: what is Restful design?
restful design enable URLs to carry meaning

Q: Is there any good API to verify a valid mit kerberos?
We can use mandrill API for email delivery. As for kerberos, we could confirm with @mit.edu, and send email to confirm it

Q: Do we need to deploy MVP?
prefer to deploy MVP!

Q: is circular dependencies allowed?
shouldn't have circular dependencies in the schema

11/30/15 TA Meeting Minutes

After updating our progress to the TF, we received positive feedback after positive feedback for our project. Some suggestions as from TF for our final implementations:

- Cleaner UI
  - Think iMessage style for conversations/networks
- Ranking
  - Cancel/Edit requests
  - Confirmations/Reminders
  - Make UI more crisp
  - Work on Conversations
- Security Features
  - Salt Passwords -- BCrypt
- Requests
  - Consider attaching date to request so users can make request ahead of time
  - Consider letting users have more than one request
- Scheduling Request Matching
  - Try NodeChron