Updated Design Document

Nielsen’s Usability Heuristics

1. Visibility of System Status
    a. Our application fulfills this heuristic. All submission buttons lead to some kind of information given to user. For example, if the user submits a schedule with nothing filled out, we inform the user that they must fill out said form.
2. Match Between System and the Real World
    a. The calendar itself is a real world object people are familiar to use, making it obvious to click a date. We use an arrow to indicate to moving through the calendar, a convention used everywhere.
    b. Phrases with the Login/SignUp pages are clearly understandable and visible to users, no jargon here.
    c. Submission form also uses clear language (ie Choose Dining Halls)
3. User control and Freedom
    a. For the submission form, if users make a mistake and submit their form, they can simply edit the form to fix their mistake. Given that the form asks for confirmation, we felt this was sufficient.
4. Consistency and Standard
    a. Our GUI conforms to most standards. The Login Page is similar to other sites, like Google and Amazon. Furthermore, our calendar’s follows standard convention, with arrows to switch through months. Scheduling information through a When-2-Meet like interface, which most MIT students have been exposed to.
    b. In Addition, language within our application is always consistent. We always refer to meals as meals, and times as times.
5. Error Prevention
    a. We use confirmation boxes to ensure that the user is doing the correct action. This is especially true if the user cancels their match, or submits a form.
6. Recognition rather than Recall
    a. Our calendar is similar to most calendars, clicking on a date reveals a submission form with a clearly stated date and title. The timeslots suggests to the user to click on the timeslots, like when2meet. We argue its easily recognizable interface.
7. Flexibility and Efficiency of Use

   a. For experience users we have an accelerator to help fill schedules faster. To fill out multiple consecutive time slots, the user can drag select multiple slots.

8. Aesthetic and Efficiency of Use
   a. Calendar is simplistic in design, no irrelevant information. The size of the calendar is also large, to help focus on what's important.
   b. Similarly, the History Page displays the minimal amount of information for a user to meetup with others. That is only the date, person, and time are shown. It is easily readable and understandable by a first time user.
9. Help Users Recognize Diagnose and Recover from Errors
   a. All error codes within our application are in plain English. These error codes can be seen when signing in with incorrect login information, missing sign up information, submitting empty forms. Any error codes that are not in plain English are unintentional.
10. Help and Documentation
    a. Currently we have no documentation within our application. However, we may benefit from it. Though its use would be for explaining what swipeshare is, and how it works.

Design Decisions

Used Firebase as our Database

Firebase is a fast external service Database that supports JSON. Unlike SQL, which is typically vulnerable to SQL attacks/injections due to poorly written code, Firebase is more resistant to those types of attacks. Firebase additionally offers data validation.

Used OpenID

We originally chose to use MIT openID to make sure students who sign up are actually MIT students, and each user can only have one account (unless they make multiple kerberos - but that takes effort). We managed to connect to MIT openID; However, we are finding that the MIT openID api is sometimes slow (takes 10 minutes) and inconsistent (returns 500). Additionally we are experiencing technical difficulties in redirecting the session information from authentication to the user session. We may end up reverting to normal log-in if these issues persist.

Resigned Calendar UI

During implementing the calendar, we decided a Drag and Select system of time selection made more sense than our previous solutions. Its intuitive for MIT users most have been exposed to a When-2-Meet interface. We decided to color coat times based on dinner type as well, making it easier for users to understand what times they are selecting.

Additional Specificity in Selecting Locations

Calendar also includes selecting locations for individual time slots. These are done by selecting dorms a user prefers, then allow them to remove them from individual time slots for additional specificity. This is significantly better than in our old design that constrained the user to specify dorms for the entire meal (breakfast, lunch, etc).

Ethical/Social Issues

In addition to ethical reflection discussed in full design, some additional considerations are:

- The app lacks incentive for givers (other than, perhaps, friendship) - what if there are many receivers signed up but few givers?
- The admins have access to the database (unencrypted) which means we have the entire list of students who are givers and receivers. If enough students register, this could potentially give us data that can be used to observe things such as relationship between student demographic & giving/receiving patterns, which may be an unintended side-effect of the app.
  - If students register with their kerberos & password, this also means we as admins have access to other student's MIT log-in information
  - One way to fix this would be to encrypt all data in the db. Then the only thing the admins know is analytics-related information (number of users, match frequency, etc.)