Agenda

- Revised design feedbacks
- Progress report
- Demo
- Reflection
  - Team or individual
  - When is it due?
  - Expectation for evaluation and lessons learned
- Project Presentation
  - How long? What are requirements? When to sign up?
- Other Questions
  - For protection against csrf, we use “Encrypted Token Pattern” instead of “Double submit cookies”. Is that acceptable?

Progress Report

- Kerberos validation and account verification

  When a user signs up for an account, we first check that the entered kerberos is valid by using MIT People API, next we ask the user to either sign in to their Stripe account or sign up for a new one so it gets connect to the account in our app. We then send a verification email to the user’s MIT email, only after the account verifies that we let the user starts using our app.
- After every click on action button, there will be a red/green message on the top of the page confirming whether the action was successful.
- Complete Flow:
  - Request stage:

    A user makes a request for an item specifying item details, stores, pickup location, deadline, tip and minimum shopper rating. Once he/she presses the “Make Request” button, the request will show up on the request feeds (Deliver tab) of other users whose shipping rating is at least the specified shopper minimum rating. The request also shows up under the requester’s “Your Request” section on the dashboard. As long as no shoppers has claimed the item, the requester can cancel the request by clicking on the “Cancel” button for the request.

- Claiming stage:

  Shoppers go to the request feed (Deliver tab) to browse through requests. They can see each item info along with full name and request rating of the requester. When a shopper can claim the request by clicking on the “Claim” button, the delivery shows up in the shopper’s “To Deliver” section. The requester will receive an email notification informing him/her that the shopper has bought the item and is ready to deliver, and asking him/her to contact the shopper to decide on a pickup (phone number is given). In addition, there will also be a notification for this on the requester’s dashboard, and the request under Request table is updated accordingly.
- Delivery stage:

  Once the shopper and requester agree on pickup time, the shopper goes to his/her dashboard and checks the item and press “Set Pickup Time”. He will be prompted to enter a pickup time and actual price of the item. Once he presses “Confirm”, the requester will be notified again via email and notification on the dashboard to go meet the shopper. After the item has been delivered, the requester needs to accept/pay or reject, and rate the item. The shopper then can close the delivery where he will be prompted to rate the requester. For every of this action, there will be a notification email sent to both the requester and the shopper.

- Functional stripe payment - users can connect Stripe account to the account in our app, pay for a delivery and get paid after delivering for others.
- Notifications
  - Requesters can reject a delivery if the shopper never replies to setup a pickup time.
  - Shoppers can close a delivery before the delivery is accept or reject if the requester never shows up.
- Shoppers can filter the requests on the request feed by dorm, location, minimum requester rating, and sort them by due date and tip
- A user will get suspended from delivering for a period of time if his/her shipping rating fails below 3.
- Measure against CSRF:
  - Use npm csurf (Encrypted Token Pattern)

        - How Encrypted Token Pattern works (taken from here):

          After successful authentication, the server generates a unique Token comprised of the user's ID, a timestamp value and a nonce, using a unique key available only on the server. This Token is returned to the client and embedded in a hidden field. Subsequent AJAX requests include this Token in the request-header, in a similar manner to the Double-Submit pattern. Non-AJAX form-based requests will implicitly persist the Token in its hidden field.

Next Steps

- Fix styling of modals and tables (Czarina):
- Format and fix a bug in notification email, reformat alert messages, and allow users to
- Move instance methods in Delivery model to static methods, and do the sorting of request items and to-deliver items (Joseph)
- Enhance the profile display and editing and write tests for edit profile (Vincent)