The Loop: Testing Strategy

For the testing strategy, we implemented both backend and frontend API tests to an extent. Given the amount of authenticating middleware that was included, we were able to only execute API tests to a certain degree and then decided to supplement those tests with a written strategy of steps we would execute ourselves.

For The Loop, there are four main models that were tested individually and in conjunction with each other. They are *users*, *events*, *pending groups* and *groups.* Each model's API interface was tested without authentication middleware during development and again with middleware once they were implemented.

For the backend tests, we tested the model Controllers that interacted with the database. To automate this, we used MochaJS and ShouldJS to test the functions and ensure that the correct results were returned.

For the frontend tests, we created a Javascript test script that requires a live user session to run (user must be logged in) and sequentially submit a series of API calls. For each call, the result is authenticated and the view is updated with the results. This can be reached at <domain>/tests.

Users

- Anyone is able to request an account by clicking on Login -> New User
    a. Go to email, click on the verification link to activate the requested account
    b. Before activation, login or any other user-required request will fail
    c. Cannot request an account under a non-MIT email, or under an email which is already associated with another user
- One can log in as the site admin using credentials hardcoded in routes/api/users.js
- User login session persists until logout or server restart

Events Testing

- Anyone can view/fetch events on the home and individual group pages
    a. Anyone can filter by start time, tags, and general locations
- Any logged-in user (except site admin) can RSVP to an event
- A group admin is able to edit any existing event hosted by their group
- A group admin is able to delete any event hosted by their group
- A group admin is able to create an event on their group's individual page
    a. All fields as defined in the route spec must be present for successful creation
- A site admin is able to perform any function a group admin is able to perform on any group
    a. Except for event creation and adding an image to an event

Groups Testing

- Any logged-in user (not site admin) can register a new group
    a. This user becomes the first admin of the group
- Any logged-in user can follow any group
- Anyone can fetch/view the list of existing groups
- Group admins (and site admins) can edit group information for the group which they administer
    a. Admins can be added or removed by current admins
    b. Admins can create and edit events
- Group admins (and site admins) can delete groups
    a. Deleting a group removes all events created under that group

Pending Groups Testing

- Non-admin users (including non-logged in) cannot access list on the page
- Logged in users register groups to add to pending groups list on groups list page
    a. Only users can use the API (not site admins)
- Site admin can approve and deny groups, with success response showing
    a. Approving makes groups
    b. Denying doesn’t make a group
    c. Both actions delete the pending group object in database
    d. Only site admins can use this API