Meeting Notes

- Look over wireframes
    - How does party information on home page get populated?
        - When user RSVPs to party, get prompted to sign up for an event
        - User’s parties appear on homepage whether RSVPed yet or not
            - Once RSVP no, party disappears
            - Once RSVP yes, prompted to choose item to bring
            - Can later update items bringing
        - Must decide how to handle RSVPs and sign up for supplies
- Look over collections
    - Make sure we know Users reference to Party is just a reference
    - How to deal with RSVPs
        - List of invited users and list of attending users
    - Once person RSVPs no, cannot change
    - Invited users stores emails
- Cost-sharing
    - How to integrate APIs
    - Splitwise: submit expenses and then it splits
        - Has an API
    - Each party is a group to which attendees submit expenses
- Pitch feedback
    - Not feasible to store payment information, we shouldn’t need to worry about that
    - Concepts
        - When does a party disappear
            - Once a host receives all payments
        - Invite APIs
            - Sendgrid for emails
            - Twillo for SMS
            - Notifications should direct user to page, not submit request
            - Look at Fritter React to see how to handle requests and responses
- Milestones for MVP
    - Discuss API
        - RESTful routes
            - URL describes thing to act on, action describes verb
            - Only include ID if relevant to request
    - Hold off on tests if run out of time
- Meeting Monday 3:30-4
- Things to do for Monday
    - Basic React components
    - Able to create party, invite users, add supplies to supply list

Summary: From our wireframes, it was unclear how a user RSVPed to a party and signed up for supplies. From our collections, how does a party distinguish between guests attending and people invited to the party? Look at Splitwise for cost-sharing API. Our pitch feedback included recommendation not to handle financial information and to be clearer about the difference between an item and a contribution. We discussed how we want to have our basic React app up for Monday.

Advice: Party should disappear from a user's homepage once they RSVP "no." Know that a reference to a collection within another collection is just a reference, not a full population. Separate invited users and users actually attending the party. Look at Splitwise for cost-sharing integration so each party becomes a group on Splitwise to which attendees submit costs. Don't store payment information, security needs to be through the roof, can integrate with other applications to do that. Text or email notifications of a party invite should direct a user to the homepage to RSVP, difficult to complete the action itself. Look at Fritter React to see how to handle requests and responses. Use RESTful routes! The action describes the verb and the URL describes the thing acting upon. Only include an ID in a request URL if it's relevant to the request.

New Decisions: Once a user RSVPs "no", cannot change decision since party is removed from user's homepage. Add list of invited users to Party collection, where invited users are identified by their emails. A party disappears from a user's homepage once they've reimbursed the host.

Changes to plan/milestones: Hold off on tests before MVP due date if running out of time. Meeting again on Monday and want to have a basic app up and running for then, so a user can sign up, login, and create a party with guests and supplies.