11/20 Meeting Notes

- MVP Notes:
  - when you share a wishlist, not obvious whether emails have to be users of system (he thought it was; eventually it won't need to be)
    - might be nice to make distinction between people who have accounts and are actually able to view the list and people who just got an email that they could view it (once they got an account)
    - she thought people would be able to view a wishlist without being able to interact
      - counterargument is that it might go against purpose of avoiding duplicate gifts
  - error messages should show better
  - on the signup page, suggestions maybe a little confusing, like suggesting that you use them; maybe italicize, or maybe remove entirely
- Meeting expected scope:
  - our planned set of features seems like a good set; would be nice if we could think of more things maybe ?
  - some of the features could be done different ways and with levels of complexity
    - ex: public lists - how to find, how shared, security, etc.; if do really good job addressing them, good for scope
    - split claims raise interesting questions, if do good job handling that, another big step toward being good
- Sharing Lists via links:
  - something as a separate route, like lists/directlink/:id, with a render at the end
  - have a completely separate page for viewing it, where you can I guess interact if you're logged in?
  - could change main display to take additional parameters that cause extra javascript to run
    - ex: main view shows welcome message, after you click a wishlist it displays it, we want that from link; could put parameters in link to cause javascript to "click" the list
      - could also have this info as query param on login page, so it can be passed along
      - also could store info in the session
      - can't rely on link being followed by only the user it was sent to; should still check/enforce login (and mention in email that they must log in)
    - public lists are weird because they don't show in the sidebar
      - after first time, user can have a button to save it under shared
      - first time, though, the display might be kind of weird (how does it fit in to the main page, or is it a separate one?)
  - public list discovery:

        - page where you can view and search all public lists, then save them to your shared lists

- Design Revision:
    - Data Design with contours and stuff
    - implementation challenges
    - anything we’ve changed
- Atomicity of Requests:
    - simplest thing is to make request synchronous (not great for scalability)
    - locking per wishlist, gift, etc. - look up everything then lock it?
    - maybe address in design challenges and think of simplest possible way to make a few code changes that will help combat it in most cases; don’t have to solve it 100%
        - ex: delete wishlist first so it doesn’t show up for people, and then delete the stuff on it
        - node.js is single threaded, so can only do one thing at once; if order db calls in safest possible way, should avoid invalid state, but have good error messaging (so like do delete in correct order so that even if someone sees a thing and tries to do something to it, the db stops them)
- Security:
    - where taking inputs, make sure not evaluating them
    - injection, xss, routes that don’t require authentication being used for evil, sanitizing user inputs
    - xss handled by default in node/handlebars/ejs (group that used xss thingy seemed unnecessary), so don’t worry about it
- Concurrency Issues (like unique email account creation):
    - in most apps, concurrency grey area is between checking if taken and taking it
    - is a potential problem due to asynchronicity
    - enforce uniqueness using mongoose models (we did that! :D)