11/13 Meeting Notes

- security
    - okay to leave security stuff for after MVP
    - build things like users and authentication in with MVP
    - password encryption should go in early, kind of annoying to do later
        - concern with openshift - use different library or create own mini encryption using hashing, or use heroku
            - can maybe just use passport, which has encryption, or node hashing library
    - forgotten passwords help not necessary, up to us
- lead author per file not necessary, just mark who did what
- public lists
    - should be available to everyone, but how do we let users see them? search through all public lists?
        - don't need to worry about how long it will take to look through all public lists
- shared lists
    - should you be able to remove them?
        - interesting, some people would like it, they think we will have the resources
        - would have to consider being able to view again
        - could just mark hidden in database
        - consider how users actually behave
- single page app list sharing problem
    - route that can send link to, perform action where it adds that list to user and redirects to that view
    - separate URL for public link
- grinch problem
    - need to avoid being too restrictive, as this is kind of just a people problem
    - limiting number still doesn't hold people accountable
    - okay to ignore for private lists
    - banning users too complex
    - monetary donations for public lists…?
        - only want one thing, and infinite
        - pledges to donate
    - maybe being able to turn off claims is the best of our ideas
    - MVP will tell us how much to worry about this kind of stuff
    - this is not an easy problem and would require a good bit of thought
- data model => data design
    - name all relationships in data model, even if it seems obvious
    - maybe don't need name and email? important, but doesn't relate too much; doesn't really matter

- unclear on private vs public and both being able to be shared
- “contains” on gift-claim relation is weird
- stick to state, so not user “makes” claim, but some noun
- represent split claim functionality in data model? separate thing for split claim?
    - percentage doesn’t make it clear that there’s a notion of a split claim