CoFridge : Project Full Design (MVP)

Changes

A section on what changes you made to address the review / suggested changes from your mentors reviewed it and suggested changes, plus any other changes you made.

1. Change utilities: users are only allowed to post item for free or for swap, no longer needed option.
2. Adding status for Item to enable tracking of each Item throughout their life process on Cofridge(posted 0, claimed 1, confirmed 2, finished 3, cancelled 4).
3. Changed parts of the visual UI of the app.
4. Removed the functionality of holding events, but just focused on developing the core functionalities.

Nielsen's Usability Heuristics

An explanation for how your UI fulfills each of Nielsen's Usability Heuristics or why the heuristic is not applicable to your app

1. Visibility of system status
In order to lessen the difficulty of navigating between different pages, we designed the navbar on the top of the screen. Then, users have a clear idea of where they should quickly point.

2. Match between system and the real world
Navigation bars are placed at the top of every page like how most sites are set up. The frontpage has some nice figures that match our theme

3. User control and freedom
Users can input a portion of self-input contents when creating an item; they can also edit their user profiles. Once they have claimed an item, they can then cancel the claimed item if they cannot make it to pick up.

4. Consistency and standards
Both item and fridge have a consistent format so users will not get confused with them. We also have tried to avoid using any words with duplicate meanings to avoid confusion.

5. Error prevention
While creating an Item, we only allow a subset of status and information to be entered by users; the rest of them are predefined so users are not allowed to type them in, instead they have to select them from a presetting.

6. Recognition rather than recall.
We have red star markers to remind users which fields are required, have placeholders in input fields so users know what to put, display the user's selected image for preview and store user-related information and actions in their profile so that they can directly access and remind themselves of the information (i.e. total karma points, items claimed, etc).

7. Flexibility and efficiency of use
Navigation bar is designed for easier access to different pages. Users can jump between the fridges dashboard and user profile. Users can decide whether to upload images for items.

8. Aesthetic and minimalist design
A landing page with a thorough introduction is designed for users to better understand our principles and solutions for problems.
A general color theme is applied to the whole web app for consistency.
We do not allow users to upload irrelevant items beyond the food grocery domain as we wanted to ensure that the platform is only for grocery food sharing.

9. Help users recognize, diagnose and recover from errors
We prompt notifications listening to events when users perform some actions to help them diagnose the current situation.

10. Help and documentation
We have red star markers to remind users which fields are required. We make the label for each input area easy to understand.

CoFridge : Project Full Design (Kickstarting)

Overview

Have you ever found unfinished groceries in your fridge at the end of your grocery cycle? Have you ever felt unmotivated to finish something you bought because you are ready to try something new? Have you ever thrown away groceries when you move? …

We are building a web app that seeks to minimize household food waste and alleviate food insecurity by facilitating grocery sharing among members of the local community. The system will act as a space for virtual shared fridges among groups of users to maximize the efficiency of grocery usage by leveraging people's different yet complementary grocery stock. We hope that this project could minimize situations of household food waste, increase food access for food-insecure people, foster tighter communities and raise awareness on the importance of saving food.

There are three key features of the app: transact, track and tribe.

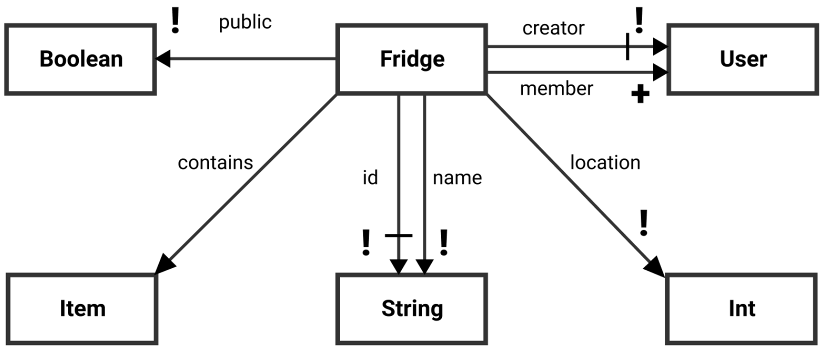
Transact | The purpose of this feature is to facilitate various types of grocery transactions.
In this feature, users can give, take, borrow and exchange items in one or multiple shared groups called "fridges". They can also create new fridges and invite users to the fridge. There is also a public fridge where all verified users of the app could access and allocate items to. When adding a new item, the user can determine if it is entirely free, up for borrow or exchange, or worth a certain number of points to get.

Track | The purpose of this feature is to motivate individuals and encourage them to be active, trustful users of the app.
In this feature, the users can see a data visualization of themselves' or their friends' contributions/participation overtime, karma points accumulated, and "magnets" collected along the way. Different amounts of karma points will be given to users based on their activities, from most to least points: giveaway, give, lend, exchange, buy (with points), borrow, take. Users will not be getting points if they ask for points in their added item. Points will be taken off if a user is reported to not have been abiding by community rules of the fridge(s) they are in.

Tribe | The purpose of this feature is to motivate and support communities in the process of sharing groceries.
In this feature, users can add friends and customize the appearance of fridges they are in (or the public fridge) using their magnets.

Concept Design

<Fridge>
Concept Fridge
Purpose To allow users to transact items in smaller groups
State



Actions

```
create (n: String, l: Int, p: Boolean, u: User): Fridge
        result := fresh Fridge
        result.name := n
        result.location := l
        result.public := p
        result.creator := u
delete (f: Fridge)
        remove f from set Fridge
modify (n': String, l': Int, p': Boolean, c': creator): Fridge
        result.name := n'
        result.location := l'
        result.public := p'
        result.creator := c'
search (n: String or l: Int): Fridge
        return f in Fridge, f.name == n or f.location == l
join (n: String, u: User, f: Fridge)
        if u not in f.member
        add u to f.member
exit (u: User, f: Fridge)
        if u in f.member
        remove u from f.member
```

Operational Principles

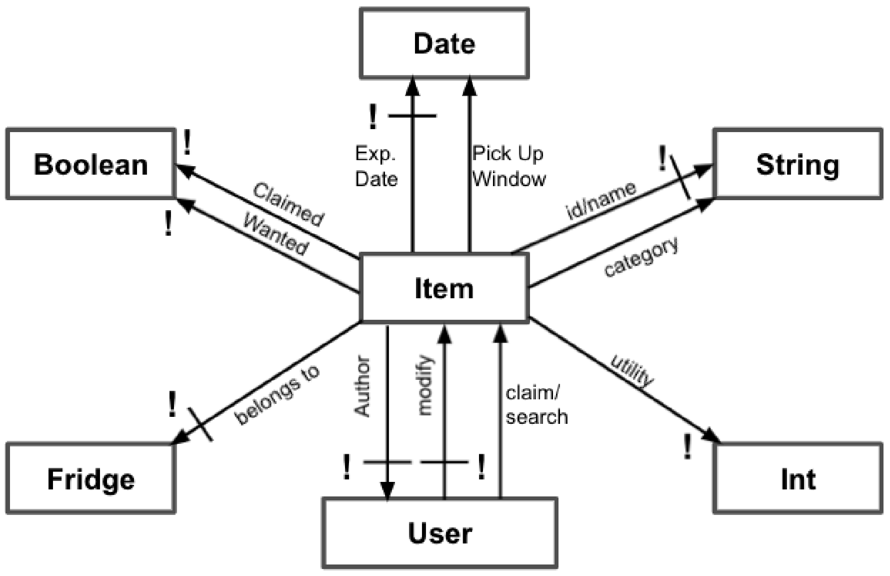
A user creates a fridge specifying its unique name, location(in zipcode) and publicness, and the user becomes the creator of the fridge (create(n,l,p,u,f)). The creator can delete the fridge, and modify the name, location, publicness and creator of the fridge (modify(n',l',p',c',f)). By modifying the creator of the fridge, the old creator transfers the modify & delete actions exclusive to the creator to a new creator. A user can search for existing fridges by name or location (search(n or l)), join a fridge after being approved by a creator of the fridge - in which case the user becomes a member of the fridge (join(n,u,f)), and exit any fridge that he/she is a member of (exit(u,f)).

<Item>

Concept Item

Purpose To allow users to post an item in the Fridge

State



Actions

Create (n: String, d: String, w: Boolean , u: Int , c: Int, p: Photo(default: None), f: Fridge, e: Date, a: String, t: String): Item

```
    result:= one Item
    result.id := Int
    result.name:= n
    result.description:= d
    result.wanted:= w
    result.utility:= u (free 0/swap 1)
    result.category:= c
    result.photo:=p
    result.fridge:=f
    result.expiration:= e
    result.author:= a
    result.pickUpWindow:= t
    Result.claimed = cl
```

Modify (property: String, p’: String):
    I.property := p’
Delete (I: Item):
    remove one Item I from one Fridge
Claim (I: Item, pickUpTime: String, u: Int, note:String): Item
    I.claimed := true | I.utility == u, pickUpTime in I.pickUpWindow
Search(n:String, w: Boolean, u: Int, c:Int, t: String): Item
    I in Item | I.name==n and I.wanted==w and I.utility==u and I.category==c and I.pickUpWindow==t

Operational Principles

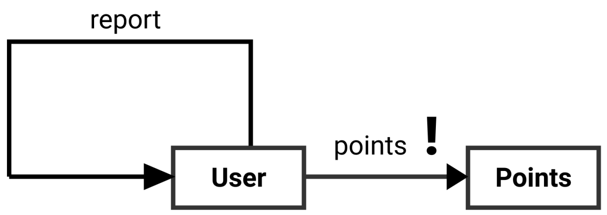
A user creates (create(**args))an Item specifying its name, grocery category, wanted or offered and utility type(free 0/swap 1). User also specifies its expiration date as well as a pick up window. Users can choose to upload a photo and add a description. Then the user becomes the author of the Item and a unique Id is generated for an item. The creator can delete(delete(**args)) the Item, and modify the property of an Item(modify(**args)). A user can use (search(**args)) to filter existing Items by name, wanted or needed, utility type, category, and pickupWindow. This allows users to specify the requirements for the food that they want. The result will be a joint set of all the conditions. Users can claim an Item by specifying the pick up time, utility type, and other information in notes, such as the swapping item and pick up instructions. Only a confirmed claim will be acknowledged and affect the state of an Item.

<Points>

Concept Points

Purpose To allow users to have a measurable reputation

State



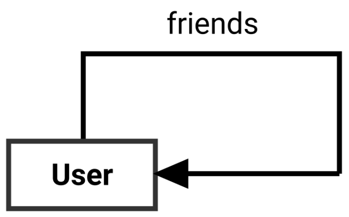
Actions

View (u: User): Points
Report (u: User, u’: User, r: String): u report u’ for some reasons r which will affect u’’s points
Earn(u: User, n:#Points): u.points += n
Loss(u: User,n:#Points): u.points -= n

Operational Principles

Users’ points are displayed in their Profile page which can be viewed(View(u:User)) by all the users. One is allowed to report(Report (u: User, u’: User, r: String)) other users if they are not following the rules. Administrators can change the points based on misbehavior. Users can earn(Earn(u: User, n:#Points)) points with each transaction. The number of points earned is based on the transaction type. They can also lose(Loss(u: User,n:#Points)) points when they deem points to make transactions or when they are penalized by administrators.

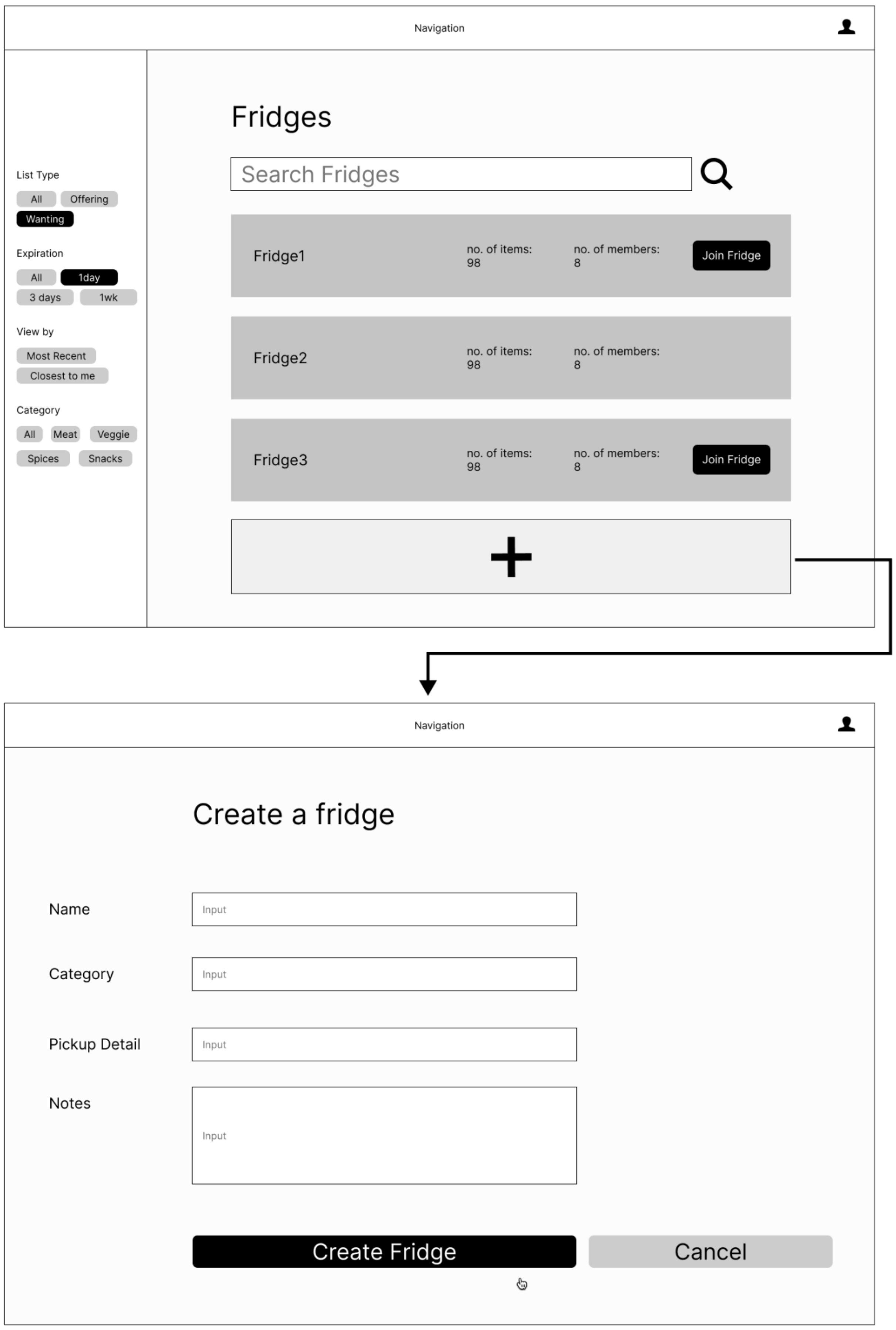
<Friend>
Concept Friend
Purpose For users to keep a close friend relationship with other users, allowing them to easily find their Fridge and Item
State

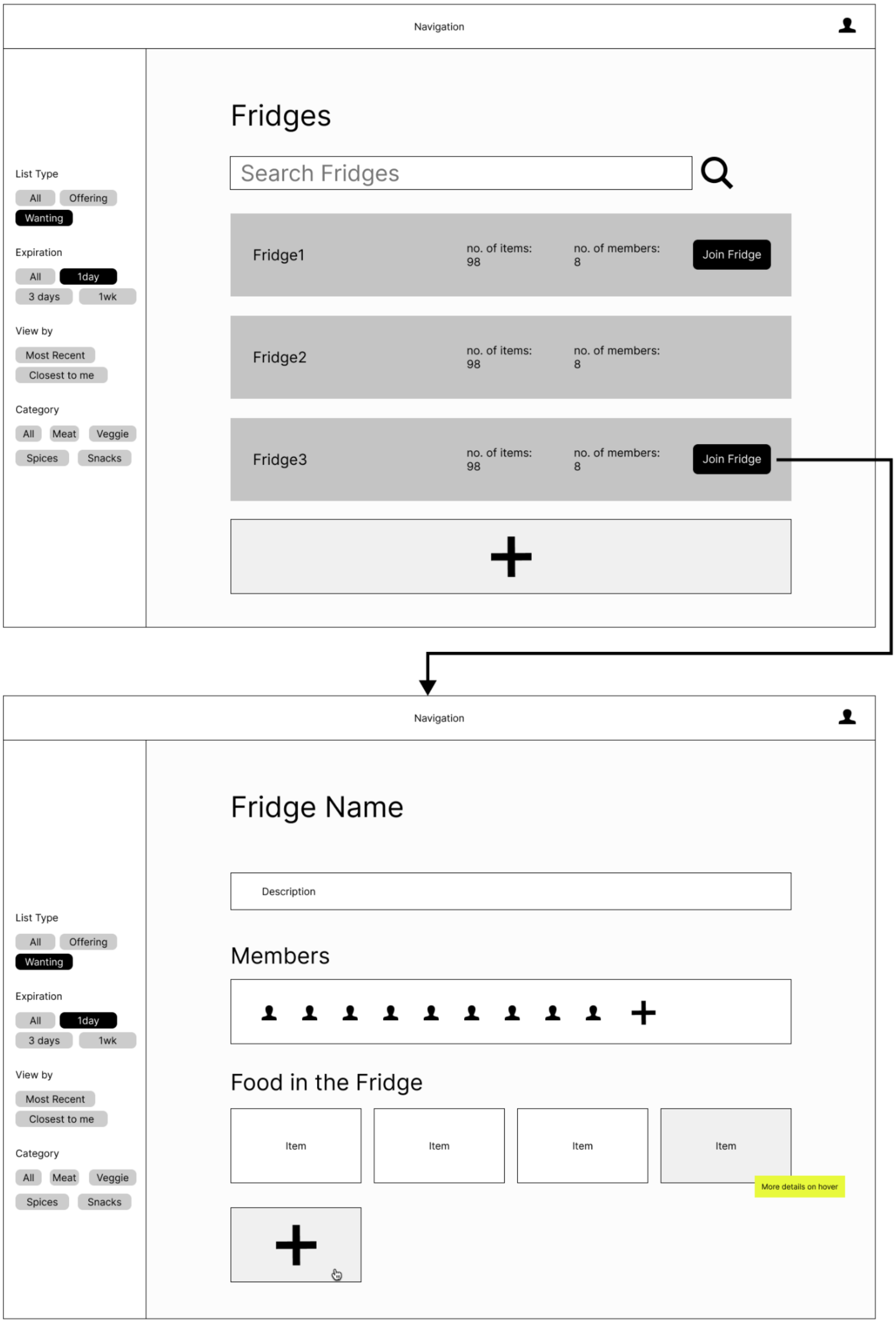


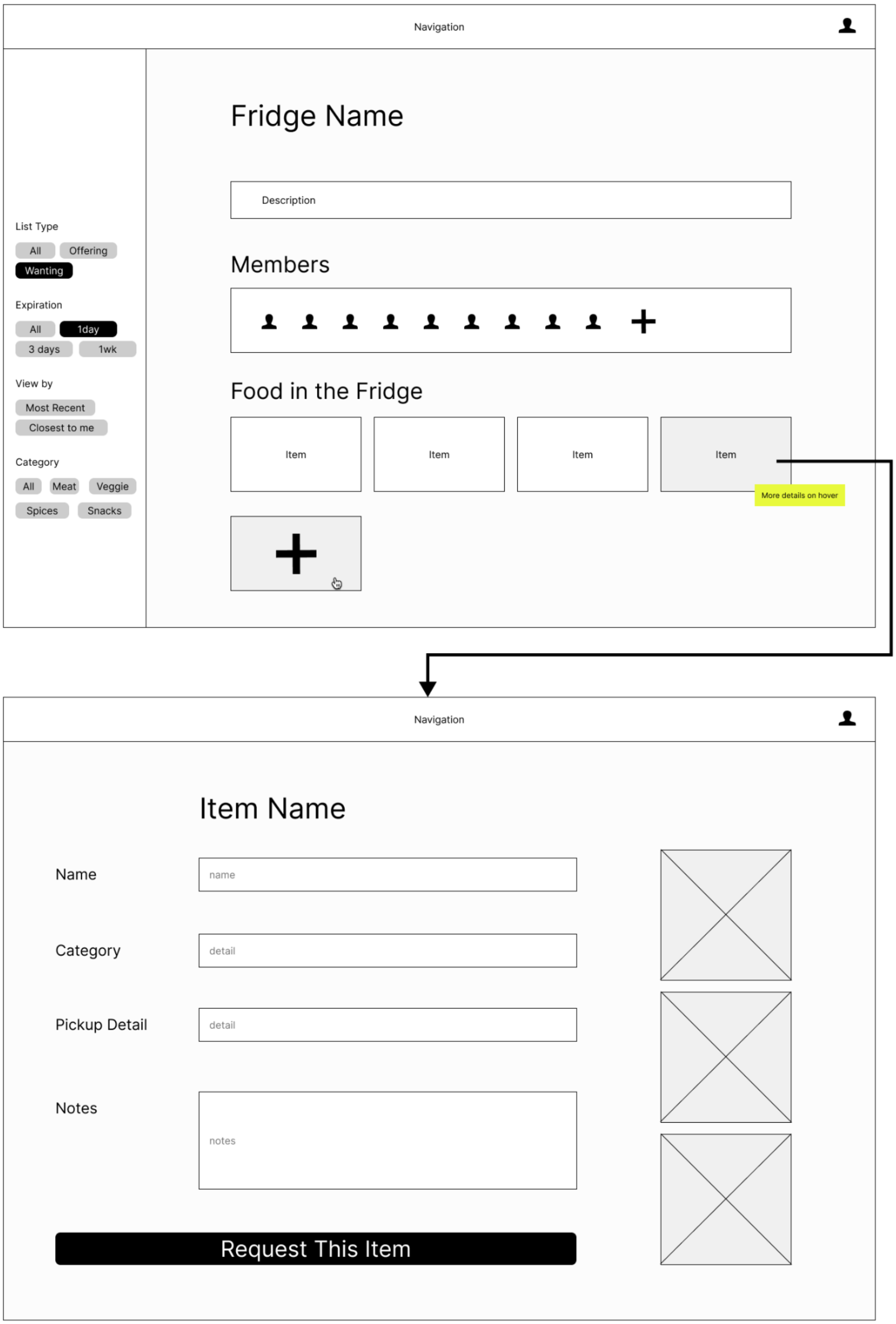
Actions
addFriend(u: User, u': User): u.friends = u.friends + u'
unFriend(u: User, u': User): u.friends = u.friends - u'
Operational Principles
A user can add other users as their friend (addFriend(u, u')). They can also unfriend users from their friend list (unFriend(u, u')). Users can invite friends to share private fridges and items their friends post on the platform. If two users are not friends with each other, one user can only see items by the other user if the item was created in a public fridge.

Sketches/ Wireframes (Charles)

- **List of Fridges**
- **Create a Fridge**
- **Fridge Dashboard**
- **List an Item**
- **Request an Item**
- **Add Members to a Fridge**


Navigation
Fridges
Search Fridges
List Type
All
Offering
Wanting
Expiration
All
1day
3 days
1wk
View by
Most Recent
Closest to me
Category
All
Meat
Veggie
Spices
Snacks
Fridge1
no. of items: 98
no. of members: 8
Join Fridge
Fridge2
no. of items: 98
no. of members: 8
Fridge3
no. of items: 98
no. of members: 8
Join Fridge
+
Navigation
Create a fridge
Name
Input
Category
Input
Pickup Detail
Input
Notes
Input
Create Fridge
Cancel


Navigation
Fridges
Search Fridges
List Type
All
Offering
Wanting
Expiration
All
1day
3 days
1wk
View by
Most Recent
Closest to me
Category
All
Meat
Veggie
Spices
Snacks
Fridge1
no. of items: 98
no. of members: 8
Join Fridge
Fridge2
no. of items: 98
no. of members: 8
Fridge3
no. of items: 98
no. of members: 8
Join Fridge
Navigation
Fridge Name
Description
Members
Food in the Fridge
Item
Item
Item
Item
More details on hover



Navigation

List Type

All Offering [✓ Wanting]

Expiration

All [✓ 1day] 3 days 1wk

View by

Most Recent

Closest to me

Category

All Meat Veggie Spices Snacks

Fridge Name

Description

Members

Food in the Fridge

Item Item Item Item

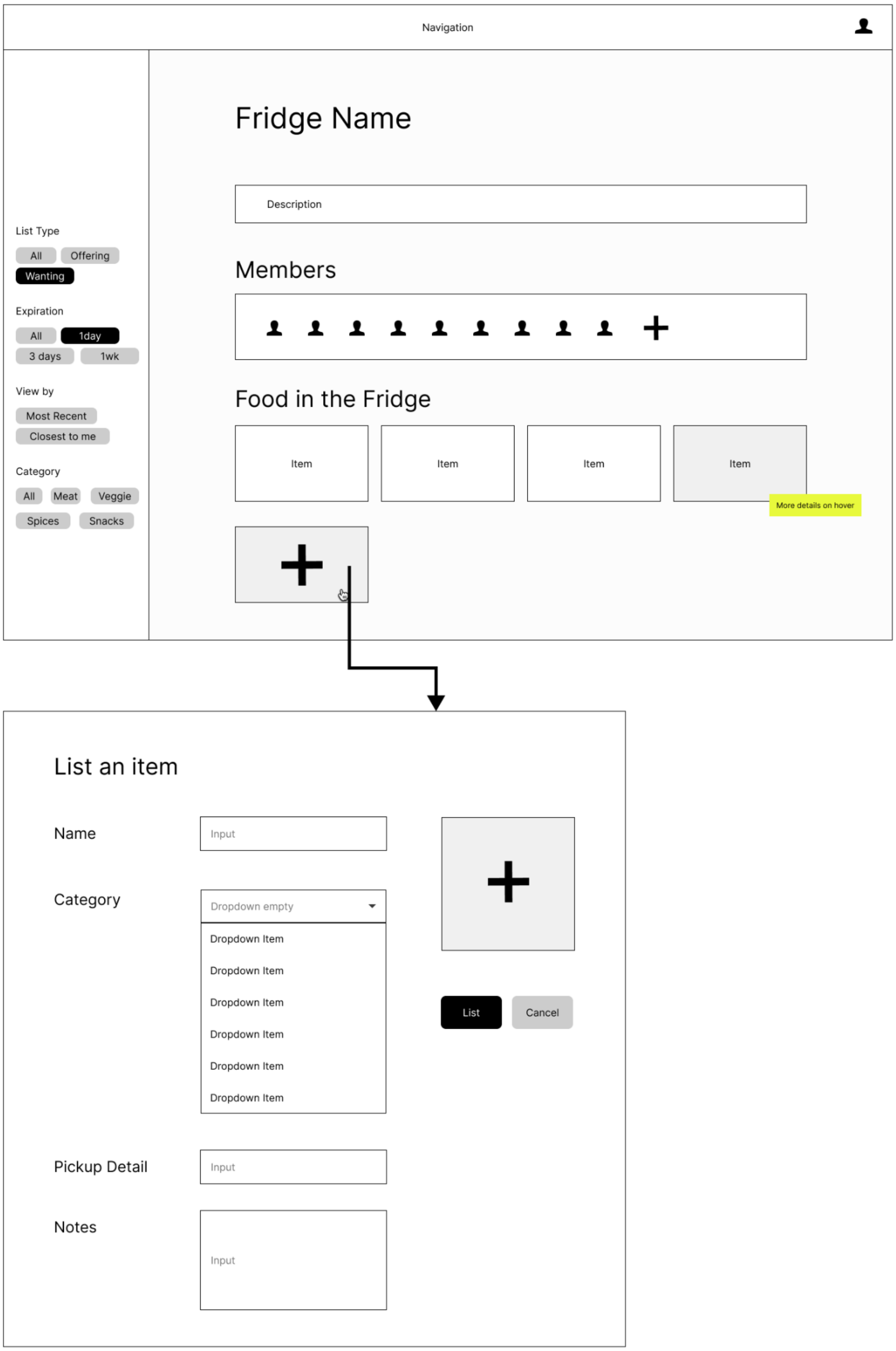
More details on hover

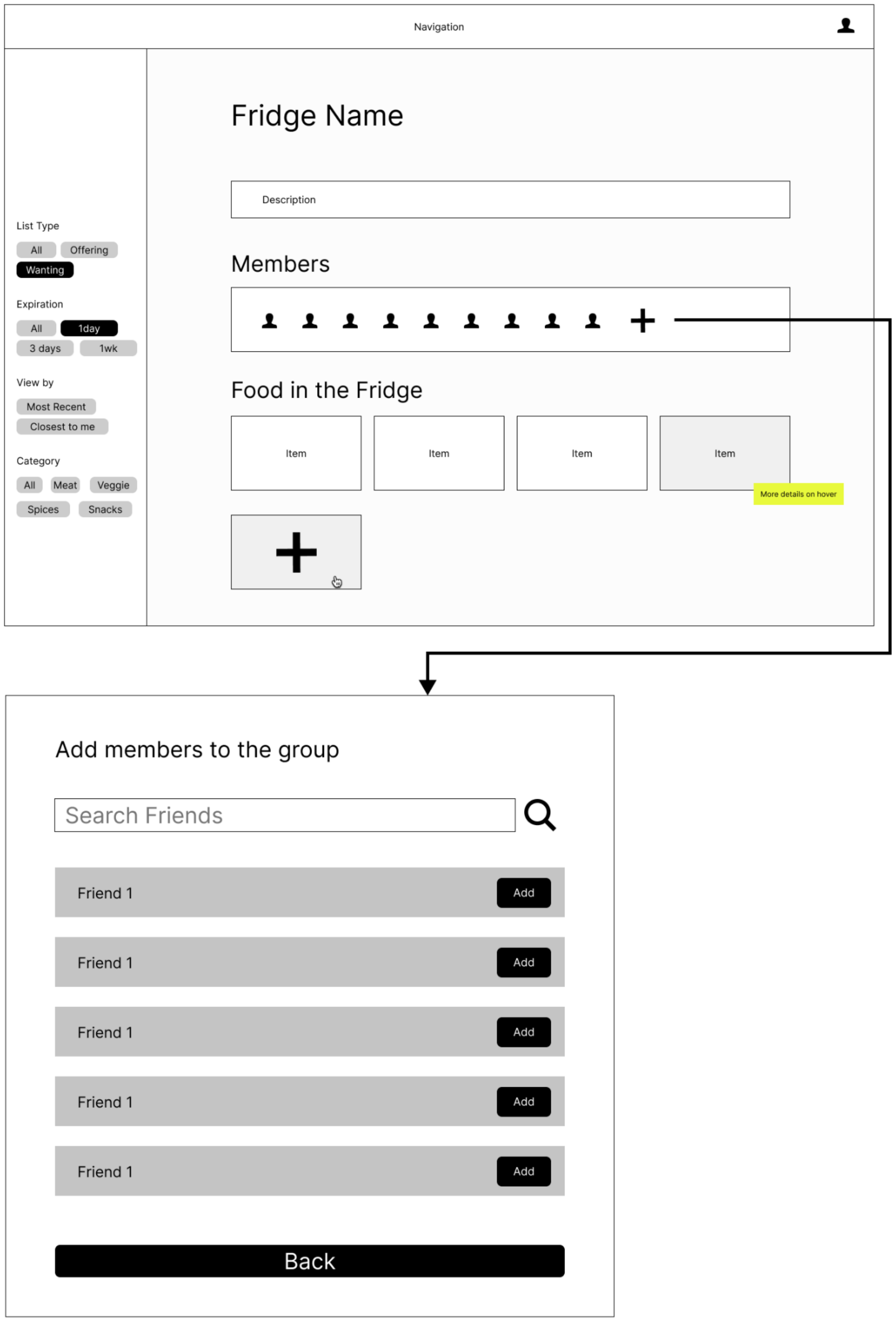
Navigation

Item Name

Name	name
Category	detail
Pickup Detail	detail
Notes	notes

Request This Item


Navigation
List Type
All
Offering
Wanting
Expiration
All
1day
3 days
1wk
View by
Most Recent
Closest to me
Category
All
Meat
Veggie
Spices
Snacks
Fridge Name
Description
Members
Food in the Fridge
Item
Item
Item
Item
More details on hover
List an item
Name
Input
Category
Dropdown empty
Dropdown Item
Dropdown Item
Dropdown Item
Dropdown Item
Dropdown Item
Dropdown Item
Pickup Detail
Input
Notes
Input
List
Cancel


Navigation
Fridge Name
Description
Members
Food in the Fridge
Item
Item
Item
Item
More details on hover
List Type
All
Offering
Wanting
Expiration
All
1day
3 days
1wk
View by
Most Recent
Closest to me
Category
All
Meat
Veggie
Spices
Snacks
Add members to the group
Search Friends
Friend 1
Add
Friend 1
Add
Friend 1
Add
Friend 1
Add
Friend 1
Add
Back

Design Commentary (All)

1. **To allowing users swapping food(Heather)**
   In our original plan, we had chosen to only allow users to share leftovers for free. We decided to also allow users to swap food or kitchen items. This will give users more flexibility to choose how to maximize the value of their items and make greater use of the sharing economy. Our original design only allowed users to share food for free, which could be demotivating, while swapping could bring more fun to the process.

2. **Including a points system to encourage users to be active and responsible (Tiange)**
   We decided to include a system where users can accumulate karma points by performing actions through the app to encourage users' active involvement and responsible behaviors. The metrics of giving points are based on different actions the users perform. For example, completing a grocery giveaway (i.e. giving out multiple free groceries in one take) are awarded the most karma points for this action, borrowing from others get a small amount of karma points for participation, taking items for free neither increases nor decreases the users' karma points and being reported for spam or other such behaviors results in negative amount of karma points. The amount of karma points a user has is viewable to other users so as a community the users could hold each other accountable. We are thinking of ways to reward users that hit a certain amount of points - for example, they can get different virtual fridge "magnets" and can use them to decorate the cover of the fridge(s) they are part of; they could also get some fresh groceries sourced by our platform delivered to them (in theory, because we couldn't actually do this in the scope of the project).

3. **Keeping small groups to enhance accessibility and ensure food safety (Heather)**
   In our original plan for scope, we had chosen to allow users to interact within the whole Cambridge area, where users can see an item that may be a bit far from them. This design can cause more inconvenience to users in their transactions and waste their time looking through items that they would not pick up due to time and distance constraints. In addition, think about the real-life scenarios that will most need this application, it is the school dormitory, the residents on the same floor of the building, a couple of neighbors in some community, and so on. Therefore, we decided to limit the effective scope to small groups by geographic location and common social groups. More specifically, Fridges are visible to the public but only within a close group of people limited by distance in order to guarantee accessibility and safety. We will require members of the group to somehow already know each other first and live close together. Thus, restricting the scope to a small number of people will be more conducive to accountability. The interchange process will also be more transparent and trustworthy.

4. **Allowing Fridges publicly searchable and joinable (Heather)**

As said before, we would allow Fridges visible to people within a certain distance. Other than that, people are allowed to search Fridge within the list of their visible Fridges, check the content, and join the Fridge. The meaning of having the concept of Fridge is to create a cozy and convenient interchangeable tribe. So users in a Fridge can choose to listen to new items in that Fridge. This way, a bunch of people within a Fridge are able to establish a stable connection for a long-term leftovers exchange. We initially decided that we would not make Fridge visible to the public, but only visible to a group of people invited by the creators from their acquaintances. However, we decided that this was not an effective way to encourage sharing and swapping of leftovers, as it is not ideal for individual users without many acquaintances on the app, or food insecure people whose social group also suffers from food insecurity.

Ethical Protocol Analysis (All)

Envision the Futures

Imagined Future: Co-Fridging is released. Within several months, many residents in the Cambridge community have created accounts and regularly use the application. Users have created and joined Fridges that are most relevant to them and started to share leftovers regularly. People get to know their neighbors and exchange groceries frequently. Co-Fridging helps to build a harmonious community and minimize the food waste to a large extent.

Identify Stakeholders

- People in food insecurity - many people struggle financially to pay for food, among other living expenses
- Residents & neighbors - we all have experience of wasting food & grocery because we buy too much and then forget about them, or missing that one item to make the meal we want to eat some night

Identify Values at Play

1. **Outcome**: In what ways does what you're making turn out better or worse for each stakeholder?
2. **Process**: How did the process treat stakeholders? Think: autonomy, consent, transparency, participation, etc.
3. **Structure / Justice**: How are the outcomes distributed among different stakeholders? What are the differences in how different stakeholders were treated by the process?

Identify Value-Laden Design Decisions

1. (Tiange) **How are sharing groups (fridges) formed?**

Possible Choice	Values promoted (and for whom?)	Values demoted (and for whom?)
Proximity only (less than a certain radius/within a certain zipcode)	**Process:** most convenience to users, less transparent within groups and not private **Structure:** Requires members of the group to be closely located **Outcome:** very convenient for transacting items but safety and privacy is the least promised for all users	**Process:** not private for users because they could expose their approximate home location **Outcome:** for anyone who is concerned about food safety and their personal security
Contacts/friends only	**Process:** most private to users and has a lot of transparency within groups **Structure:** Requires members of the group to somehow already know each other first and live close together **Outcome:** safety and security is stronger because users would share groceries among common social groups. Best for friends or acquaintance groups within the same community (i.e. school dorm).	**Process:** inconvenient for people who live far apart, which dis-incentivize participation. **Outcome:** Not ideal for individual users without many acquaintances on the app, or food insecure people whose social
Proximity + Contacts/friends	**Process:** better privacy and transparency **Structure:** empower users with choice to join fridges based on contacts	

	or proximity, in which case the fridges based on contacts are private and those based on proximity are public **Outcome:** safety and security for all users and convenience for those in greater need of grocery items	

2. (Heather) **How would we make an informative description of an (Item)?**

Possible Choice	Values promoted (and for whom?)	Values demoted (and for whom?)
User's raw inputs	**Process:** most convenience to the users who is listing items, varied quality based on users inputs **Structure:** Require users to enter all information, allowing 100% autonomy **Outcome:** Very flexible for users who list items	**Outcome:** no guarantee of completeness or validity of information for all users who view the list.
Automatic tags generation based on user input	**Process:** Allow large partial user autonomy when entering the description; **Structure:** require an text auto-completion model being embedded; **Outcome:** The generated tag is informative, uniform, and understandable by other users. Save the effort of users when they enter the description, the process	

	could be simple.	
User-selected predefined tags	**Process:** Allow small partial user autonomy when entering the description; only allow user to select predefined tags, such as: maturity: 20% 50% 70% 90% **Structure:** No need of user raw input **Outcome:** The user selected tag is uniform and understandable by other users. Save the effort of users when they enter the description, the process could be simple.	**Outcome:** Users may not find a suitable tag for an item; not very flexible, and may not be comprehensive due the lack of predefined tag.

3. (Charles)**How can we manage members in fridges?**

Possible Choice	Values promoted (and for whom?)	Values demoted (and for whom?)
Hosts could add/delete members	**Process:** The hosts who create the fridges have the right to add/delete members **Structure:** require middlewares to confirm if the users are hosts. **Outcome:** safety and security for all users and convenience for those in greater need of grocery items	

The system adds member automatically based on users' preferences	**Process:** The system adds member automatically based on users' preferences **Structure:** require algorithm to cluster users with the same preferences **Outcome:** Users with the same preferences being added to the	
Only users within certain distance could join certain fridges	**Process:** The system shows fridges that are nearby users **Structure:** require map API to detect users' location and request fridges nearby **Outcome:** safety and security for all users and convenience for neighbours to share food	

4. (NEW) How can users use their points?

Possible Choice	Values promoted (and for whom?)	Values demoted (and for whom?)
Users can use points to deem for virtual magnetic badges for their Fridge	**Process:** Users can use points to deem for virtual magnetic badges for their Fridge **Outcome:** Users who maintain a good reputation would gain decorations showing	**Outcome:** Not sure whether it is a good incentive mechanism for users to participate and be a good citizens

	their high records; attracting more users to join their Fridges and increase the variety of Items	
Users can use points to deem for real-life grocery coupons	**Process:** Users can user points to deem for real-life grocery coupons **Structure:** the platform negotiate with local grocery shops; a way to generate new leads to the grocery and a good way to motivate users to earn more points **Outcome:** Users get real-life benefits; local grocery shops participate in the process and gain	

Justify Value-Laden Design Decisions

1. How are sharing groups (fridges) formed?
    - Choice: Proximity + Contacts/friends
2. **UPDATE:** How would we make an informative description of an (Item)?
    - Choice: User-selected predefined tags + User Input
3. How can we manage members in fridges?
    - Choice: Only users within certain distance could join certain fridges + Hosts could add/delete members
4. **NEW:** How can users use their points?
    - Choice: Users can use points to deem for real-life grocery coupons