House Hub

Barthelemy Martinon, Jacob Naranjo, Skyler Norgaard & Tanush Samson
https://github.com/jacobnaran/HouseHub

Introduction:

In college and early professional life, it is common to live in a shared space. This type of living situation brings a multitude of shared responsibilities and important living matters that the group must keep track of. From our personal experience, most houses use some form of group chat to manage such matters. However, many drawbacks exist. For example, group chats are normally used for both important household matters as well as everyday social interaction. This results in important messages regarding payments and house meetings being buried beneath several unimportant messages. To the best of our knowledge, no group chat has the functionality of group reminders. This means that someone from the group must act as a reminder, which creates more room for human error. Furthermore, payment information is often scattered among multiple locations. For example, the Charter website must be consulted to determine the cable bill, the city of Kalamazoo has a website for paying the utilities bill, etc. There is also a need for organizing and delegating the purchasing of food and cleaning supplies within a house. Once food has been purchased, it is easy for house members to forget about it, and for the food to spoil as a result. This is especially true with healthy food, since fruits and vegetables expire quickly.

To provide a solution to the problems discussed in the previous paragraph, we developed HouseHub. As the name would suggest, the purpose of our app is to provide a central location for all household matters. In particular, we feature public and private shopping lists, group reminders, notes, and organizable inventory lists. The user is also able to add an expiration date to each item in the inventory list, which would help to minimize the amount of food wasted by a household. All in all, we aim to improve the general health and headspace of our users by providing a better framework for managing the tedious details of domestic living. A more detailed description of the functionality of the app as well as the development process will be included in subsequent sections.

Although our app is primarily designed for college students and early professionals living in shared houses or apartments, the diverse functionality of the app makes it compatible for any household. The app even holds value for individuals living alone who are looking for a better solution to manage the food in their fridge.

High Level Description of Project Features and App Usage:

Each feature of the app as well as the details regarding the user's interaction with each feature are described in this section. The *Walkthrough of User Experience* section includes a visual representation of these descriptions as well as additional details.

Login Page:

Upon downloading the app, the user will be directed to a login screen. This screen features two text boxes for a user to enter their email address and password. Assuming that this information is valid, the user can click *"Log in"* to proceed. The app also gives a user the opportunity to log in as a guest. After clicking this option, the user is directed to the home screen of the app. In the case of new users, who do not want to log in as guests, we include a "*Register*" option as well. If this is selected, a new screen is displayed where the user must enter a valid email address, their name, and a password. The user is then given the option to either create a new household or join an existing household. If the user decides to create a new household, a textbox is displayed and the user is prompted to enter a household name (such as *"214 Douglas"*). Similarly, if the user decides to join an existing household, they are prompted for the name of that existing household. Screenshots of each process are included in the next section.

Home Page:

After entering their information or continuing as a guest, the user is directed to the home page. This is where house notes and reminders are displayed. The notes and reminders are displayed in descending order based on the time in which they were created. In other words, the newest notes and reminders are displayed first. Each user in a household has the ability to add elements to the home page, and these elements will be viewable by the other members of the house. Elements are added by clicking the plus icon in the bottom right corner of the page. Upon clicking this icon, an option to add either a note or reminder is displayed. When either of these elements is clicked by the user, a new screen is pulled up in which the user can add the necessary text and, in the case of a reminder, event time. A visual display of these screens are included in the next section.

Inventory Page:

The user is able to switch pages by clicking the different icons in the bar at the bottom of the app. This is also displayed in the next section. When the user switches to the inventory page, they are given the option to add items and tabs. Like the homepage, this is done by clicking the plus icon in the bottom right corner of the screen. If the user clicks "*Add Item*" a new screen will be displayed, in which the user will have an option to add an item name as well as the number of days until the product expires. Added items as well as the date they expire are displayed on the page. The user also has the option to add new tabs. This would be useful in cases where there are multiple storage units within the house. For example, food may be stored in both the freezer and the pantry. By allowing for the user to create tabs, we give them the opportunity to better organize the food within their house. This would also be advantageous in the case where different house members have different personal supplies that they would want to manage separately from the shared items. This is one of the few features that is yet to be fully implemented.

Shopping List:

The last of the three primary pages is the "*Shopping List*" page. Like before, the user can add new shopping items by clicking on the plus icon. When the plus icon is clicked, a similar screen is displayed in which the user can enter the name of the item. Upon clicking "*Done*", the item is added to the list. The shopping list page also features a drop down menu in the top left corner that allows the user to switch between a private and public shopping list. Similar to the tab functionality in the Inventory section, this allows for better organization and managing of shopping needs. For instance, there may be items such as trash bags or cleaning supplies or even communal food that are needed for all members of the house, but a particular member may want to buy a certain food item that they do not intend on sharing. By differentiating between *public* and *private* lists, we attempt make this as easy as possible.

Settings Page:

Regardless of the page that the user is currently on (Inventory, Home, or Shopping List), there is an icon in the top right corner of the app that will direct them to the settings page. The settings page contains valuable information for users of the app. The name of the user is listed under "*Active Account*", the name of the house is listed under "*Active House*", and the code needed for others to join is listed under "*Shareable Code*". There is also an "*About*" section, which pulls up a window displaying the version as well as a short description of the app. The settings page also gives the option to logout, which brings the user back to the login page of the app.

Notifications Feature:

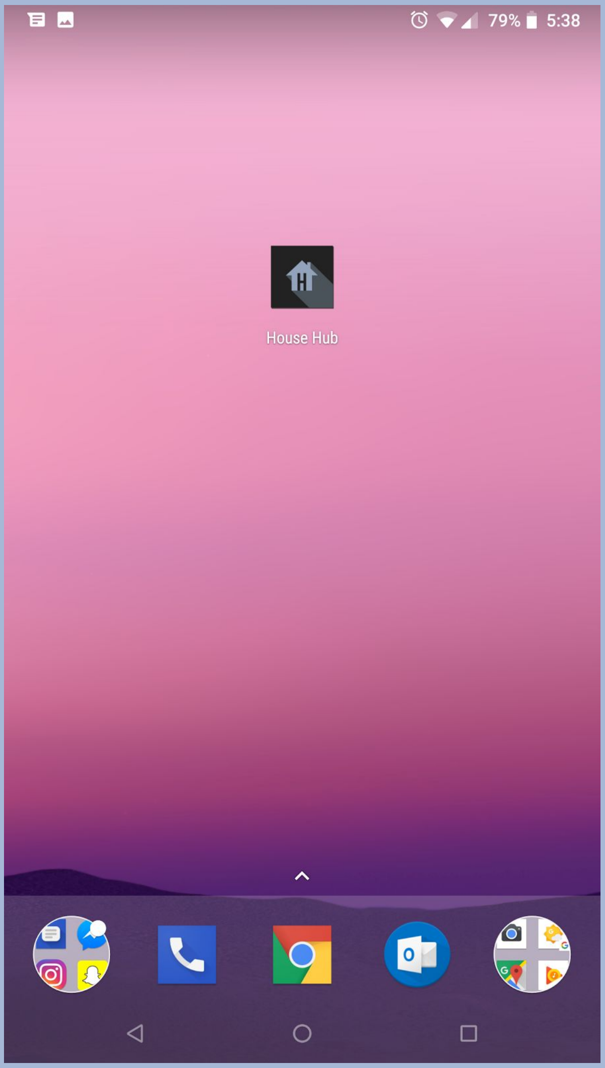
Our app also features notifications for expiring items and added reminders. This allows the user to be reminded of important deadlines without having to access the app in order to check them. An example of both types of reminders are shown in the next section as well.

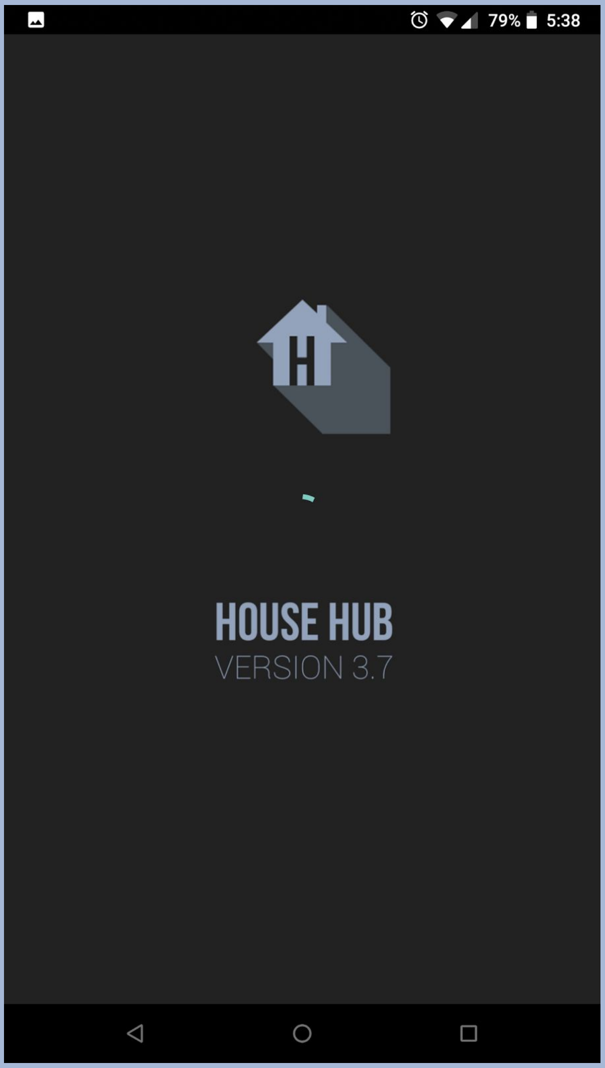
Walkthrough of User Experience:

In this section, we walk through a potential user experience, starting with their download of the app. This will help to supplement the descriptions in the previous section.

1.Opening the app from a phone:

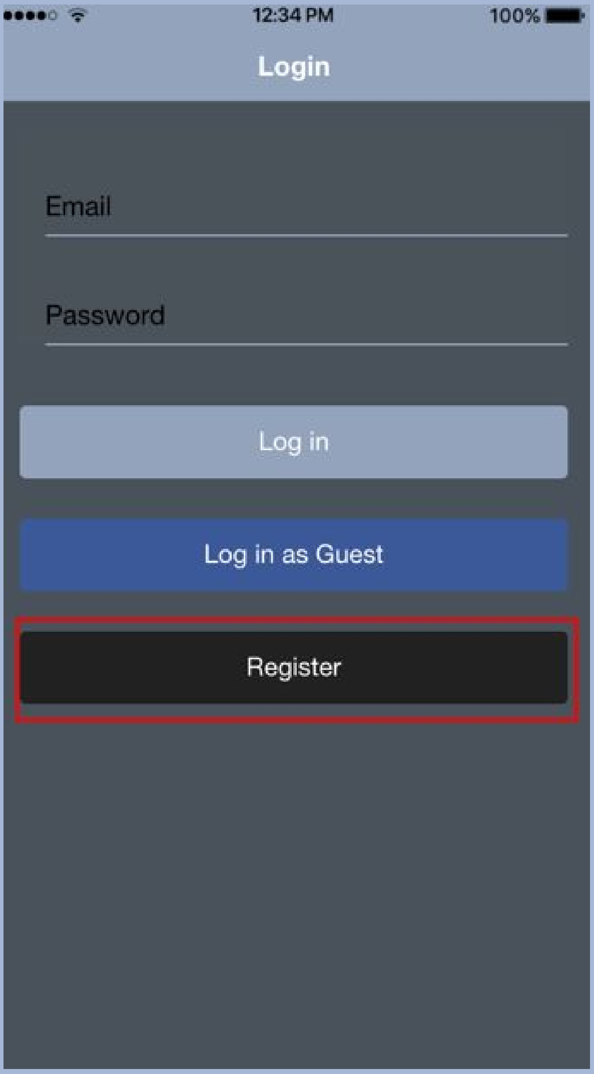
We first display the app icon that the user would see before entering the app as well as the splash screen that is displayed once the icon is clicked. It cannot be fully captured in a photo, but the splash screen features a spinning progress icon. This is a common feature of apps and was implemented to improve the user experience. In this first photo, we also see that the top status bar with information specific to the phone (battery life, wi-fi connection, etc.) changes colors to match the background of the app.

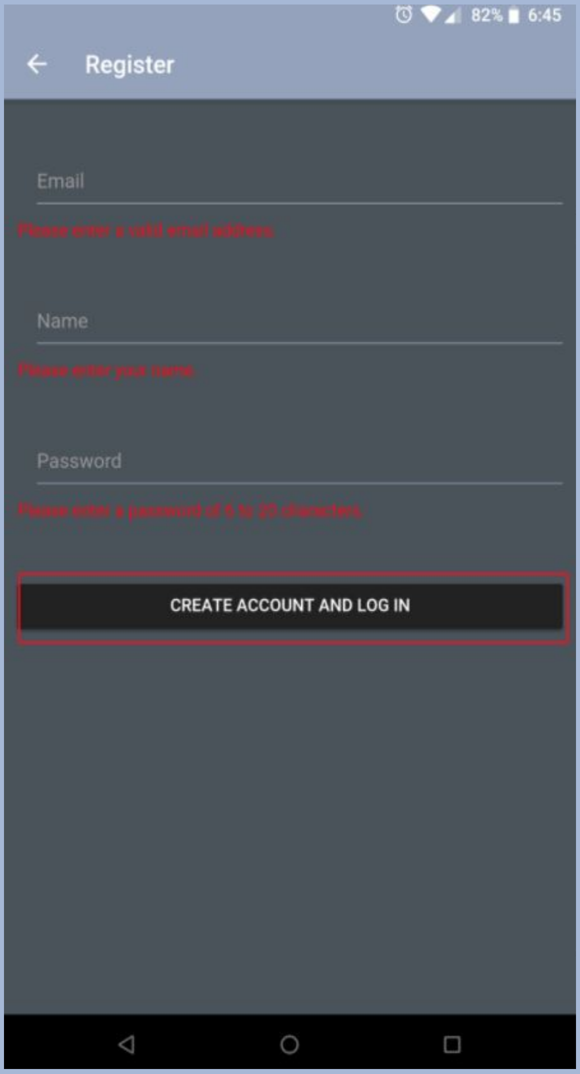




2. Registering for the App:

Since we are simulating the situation in which a user has just downloaded the app, we will show the case in which they select “*Register*.” If the user does not enter valid information, messages are shown displaying the requirements. This is similar to most registration services. After entering valid information, the user can select “*Create Account and Log In*”





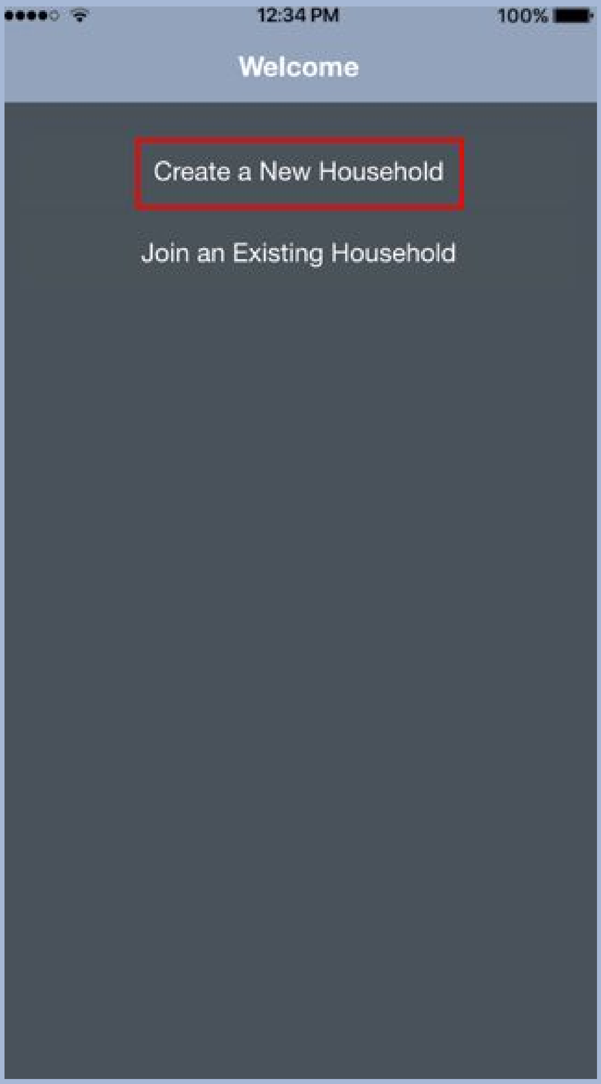
Please enter a valid email address.

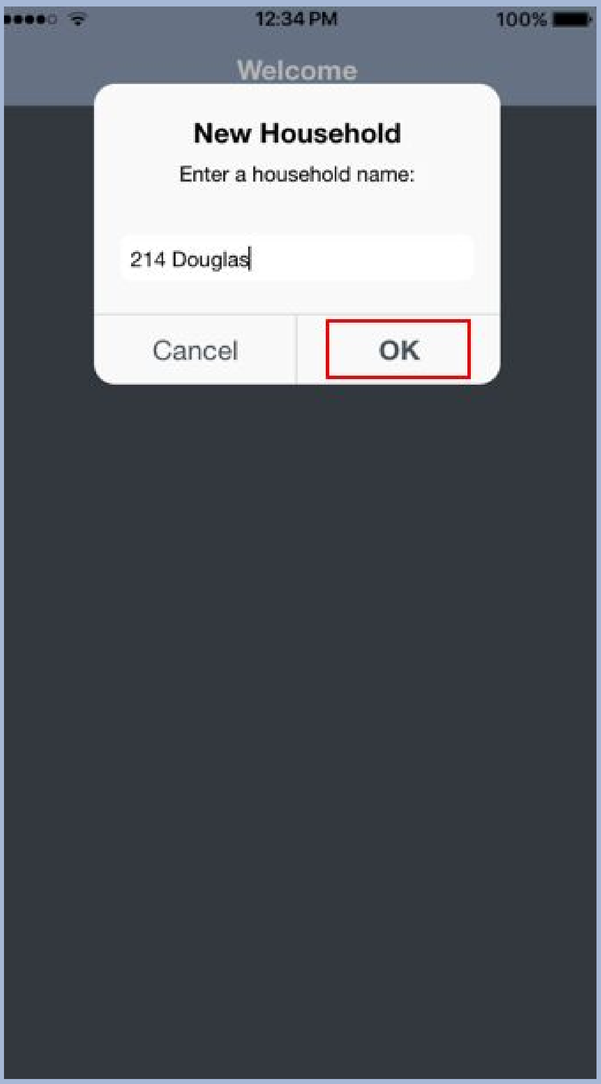
Please enter your name.

Please enter a password of 6 to 20 characters.

3. Creating a New Household:

The user will then be given the option to join an existing household or create a new one. We will show the case in which the user creates a new household. As we see here, the user is able to enter a simple name that is easy to remember and reflective of where they live and there are clear instructions as to what is required.

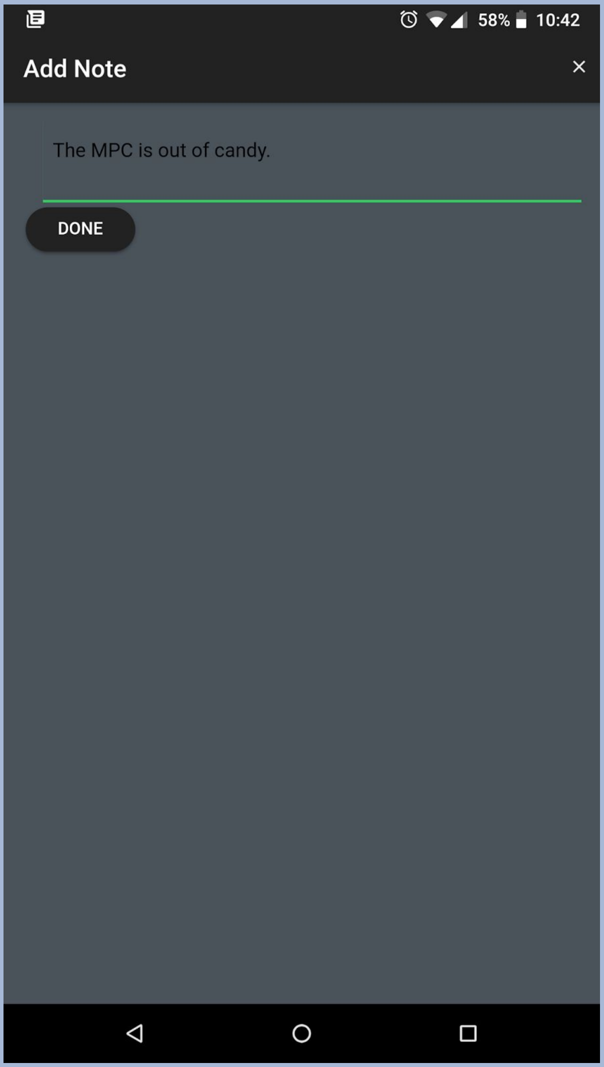




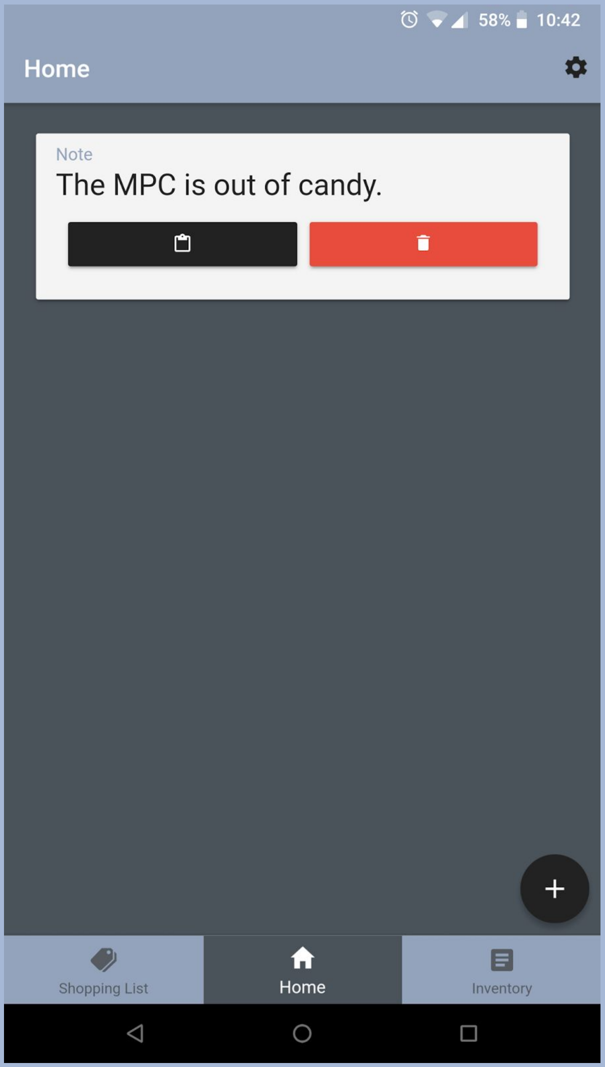
214 Douglas

4. Adding items on home page:

Here, we show the process of adding a note as well as a reminder to the home page. In the case of the reminder, we implement a sliding select menu for dates and times (shown below). We also show how each element is displayed in the home page. As is shown below, we make it clear whether the element is a note or a reminder, and we display the date and time of the reminder on the home page. As was described in the previous section, the plus icon must be selected in order to pull up the menu with the two options. We also show the way in which app reminder notifications are displayed from the lock screen as well as in an unlocked phone.



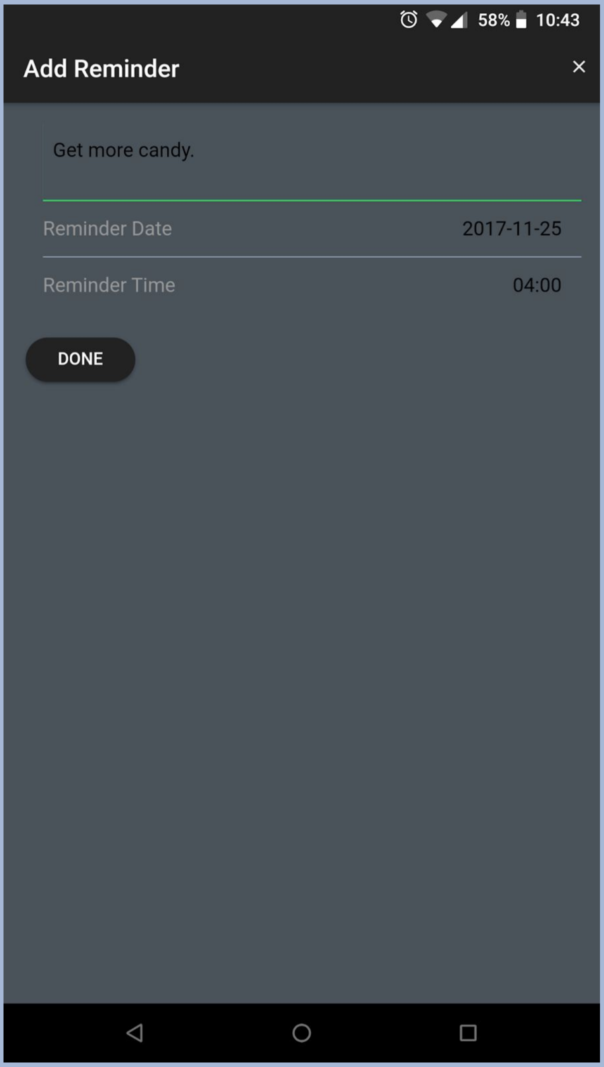
The MPC is out of candy.



Note

The MPC is out of candy.

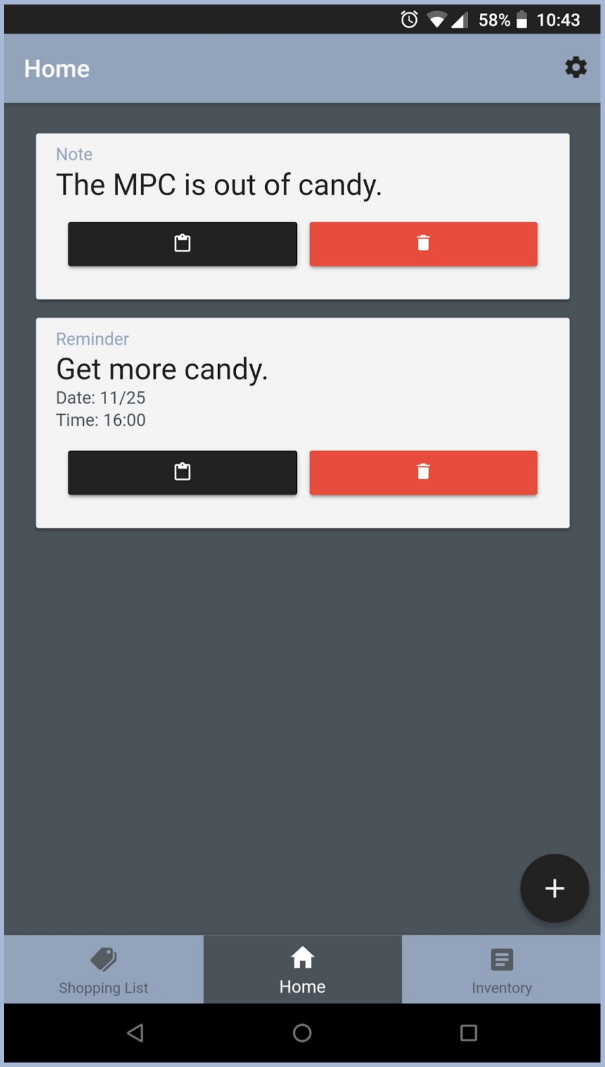
Adding items on home page (cont):



Get more candy.

Reminder Date 2017-11-25

Reminder Time 04:00



Note

The MPC is out of candy.

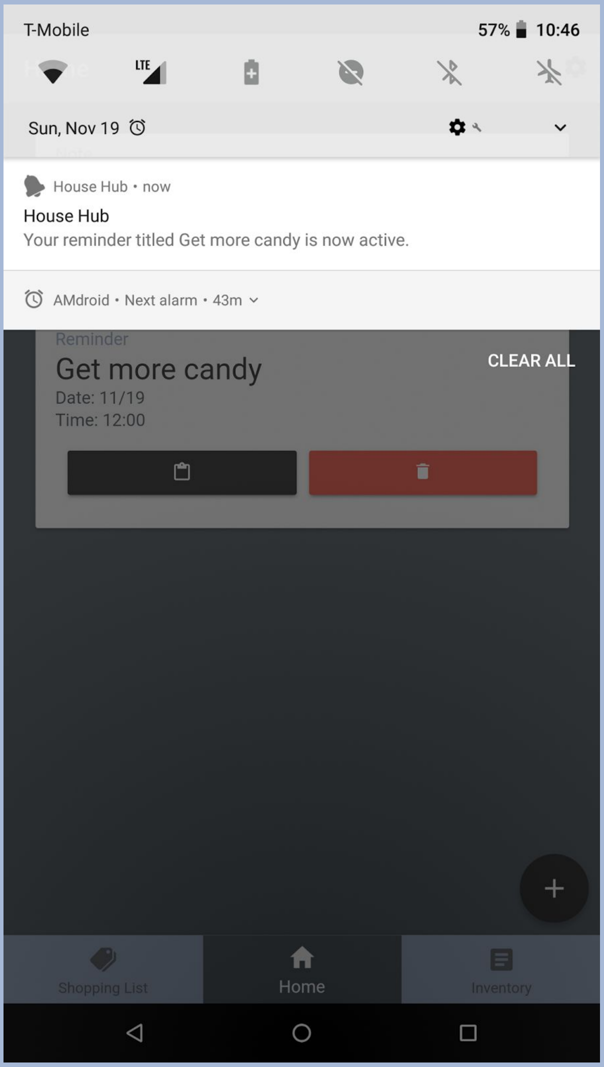
Reminder

Get more candy.

Date: 11/25

Time: 16:00

Adding items on home page (cont):


T-Mobile 57% 10:46

Sun, Nov 19

House Hub • now

House Hub

Your reminder titled Get more candy is now active.

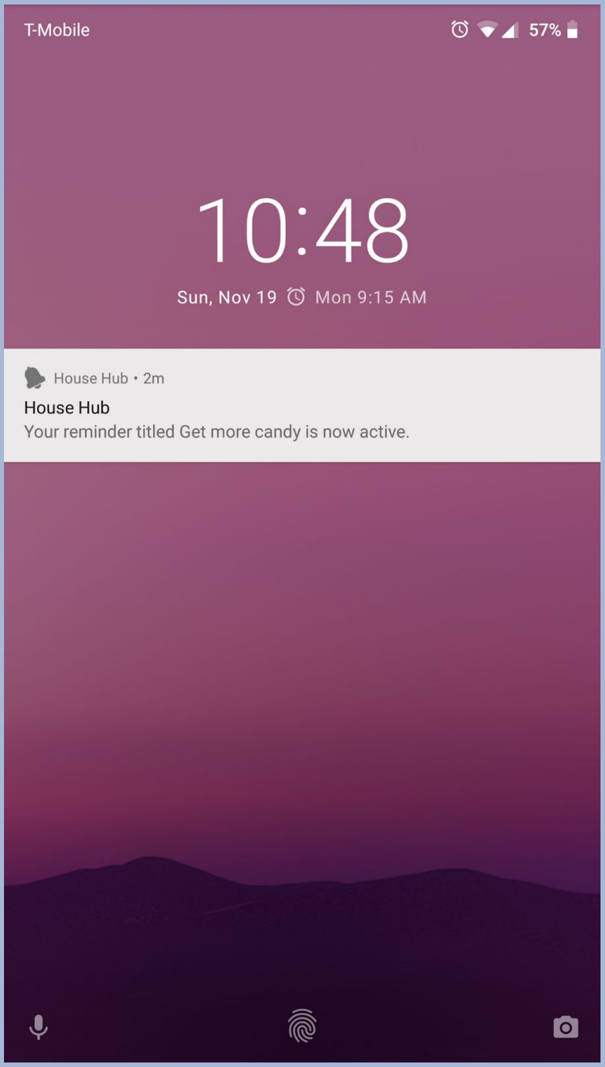
AMdroid • Next alarm • 43m

Reminder

Get more candy

Date: 11/19

Time: 12:00


T-Mobile 57%

10:48

Sun, Nov 19 Mon 9:15 AM

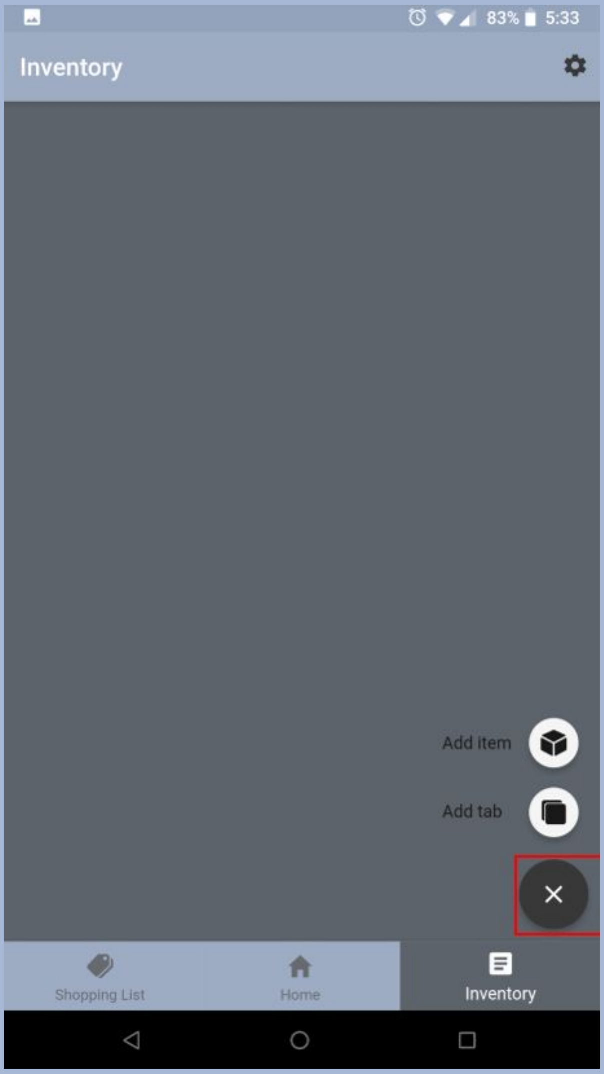
House Hub • 2m

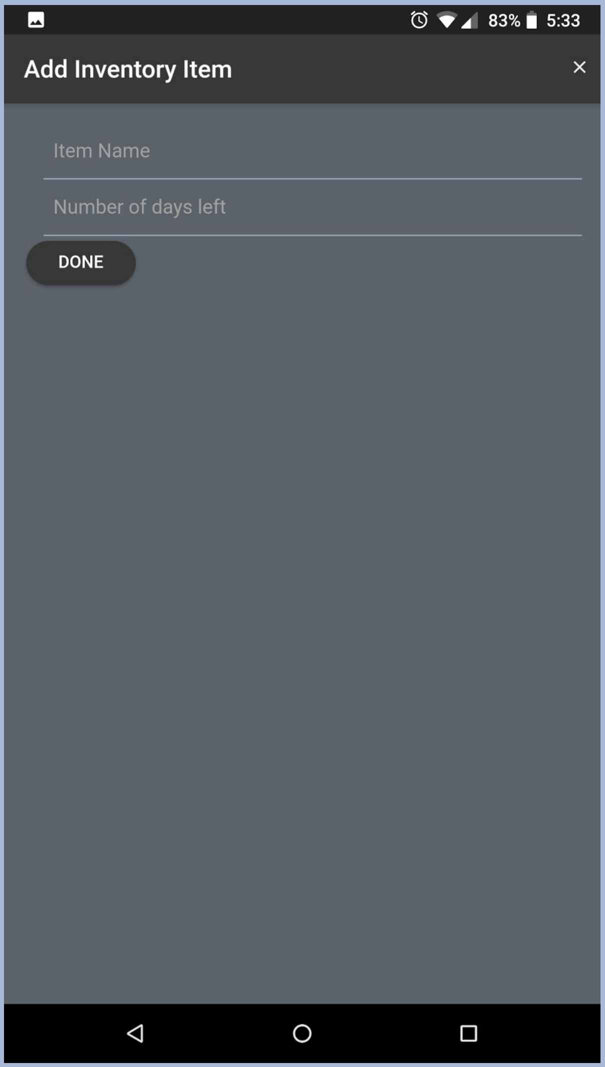
House Hub

Your reminder titled Get more candy is now active.

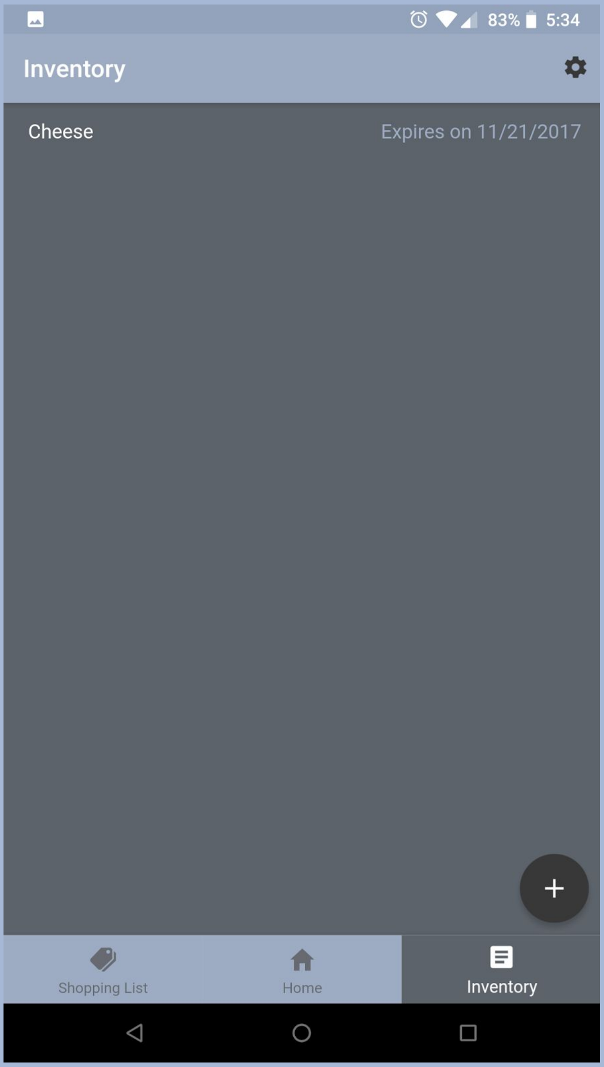
5. Adding Inventory items:

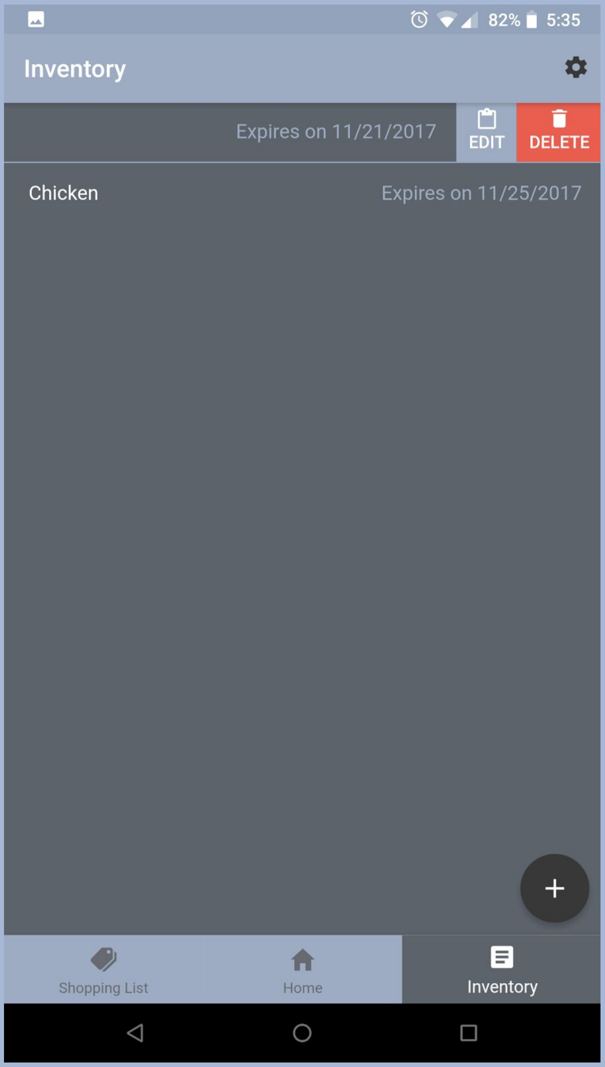
We then show the process of adding an inventory item as well as the result. To switch from the home page to the inventory page, the user would select the "*Inventory*" tab in the bottom bar. In this case, we add "*cheese*" and tell the app that it expires in 3 days. Upon adding an item, the app displays the date that the item expires. This is shown below. The item can then be edited or deleted by swiping right. This is also displayed.





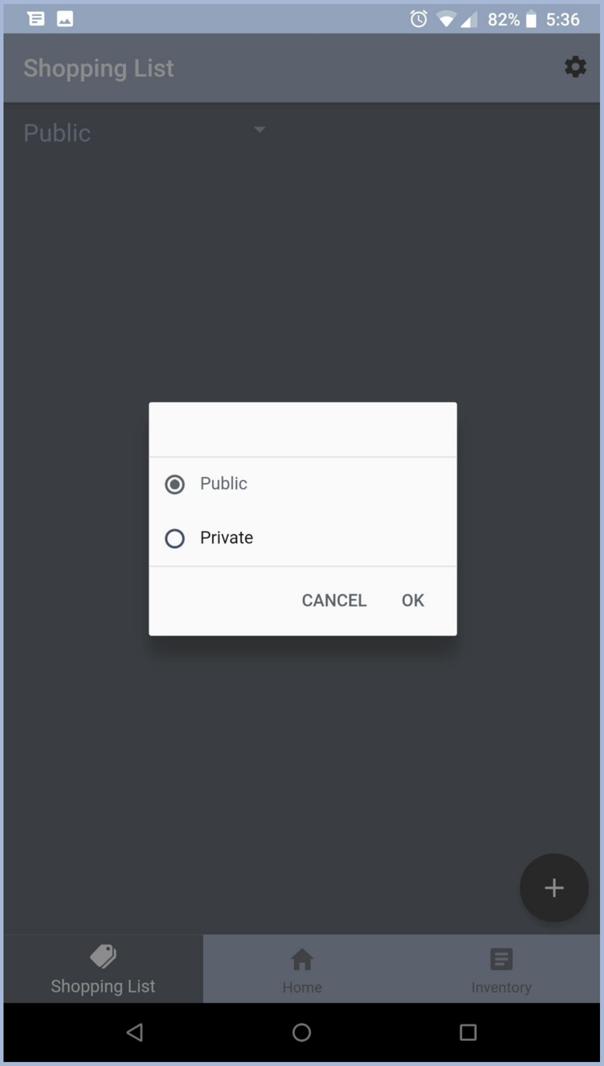
Adding Inventory items (cont.):

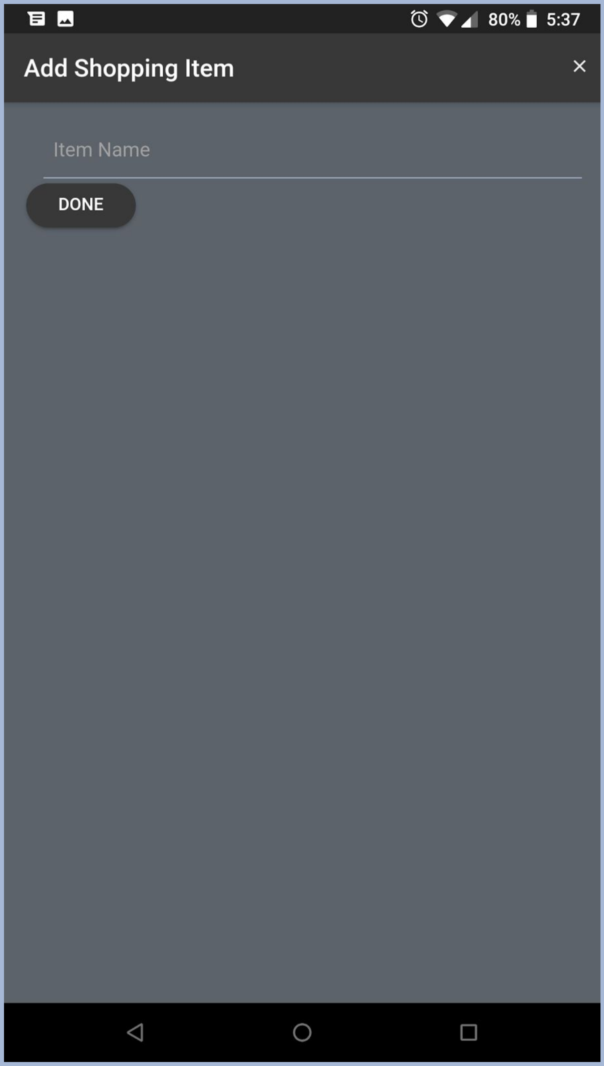




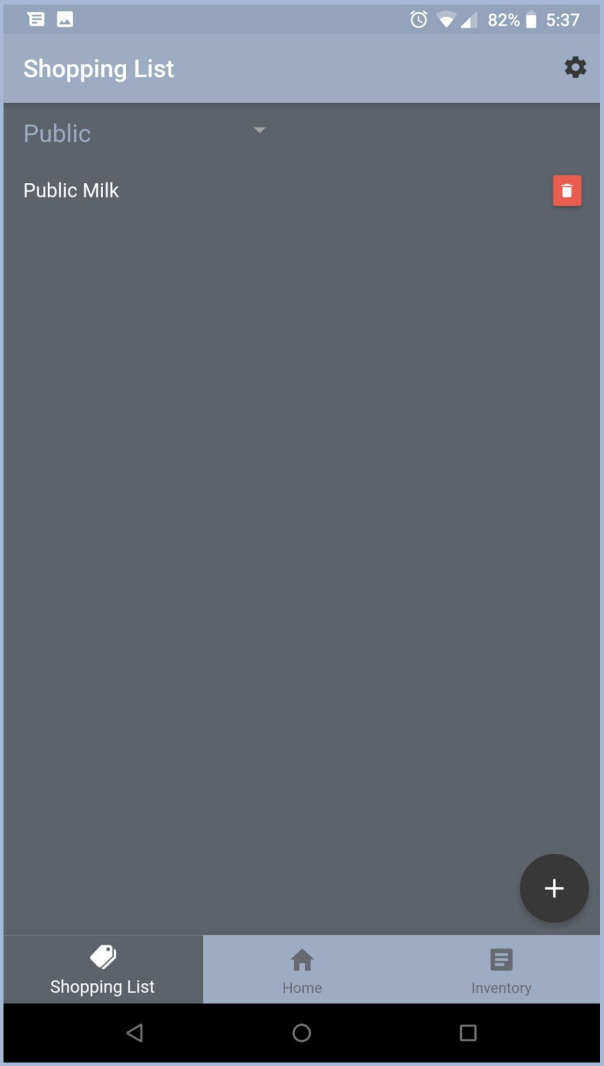
6. Adding items to the public and private shopping lists:

Here we show the functionality of our shopping list. Upon clicking the tab in the top right corner, the user is given the option to select either the private (only viewable by the user) or public (viewable by all house members) shopping list. Like before, items are added by clicking the plus icon. We display the screen for adding an item but do not display the process of clicking the plus icon since it has been shown previously.



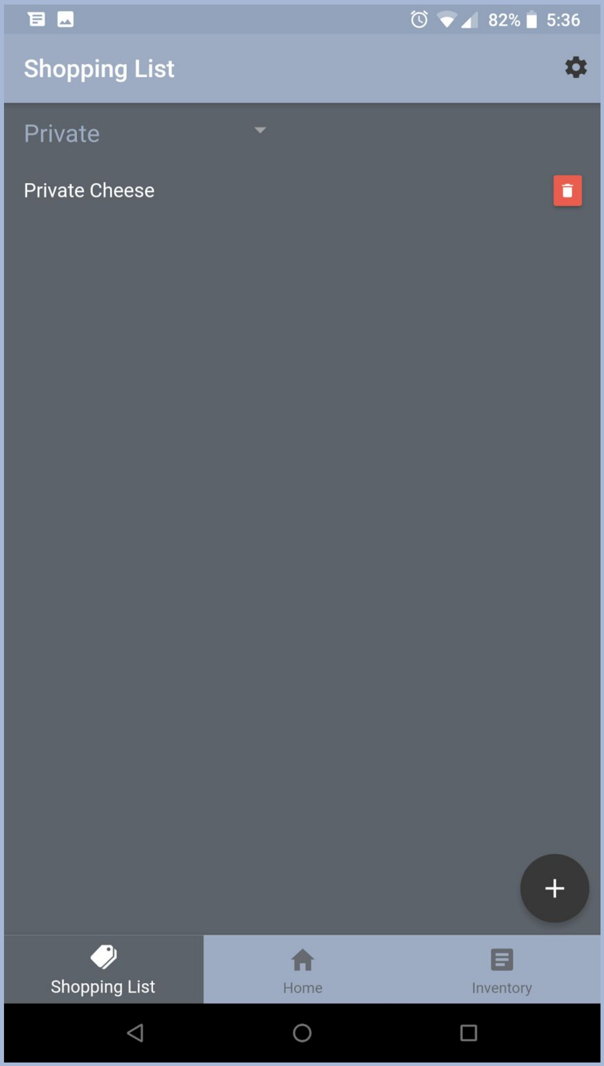


Adding items to the public and private shopping lists (cont.):



Public

Public Milk

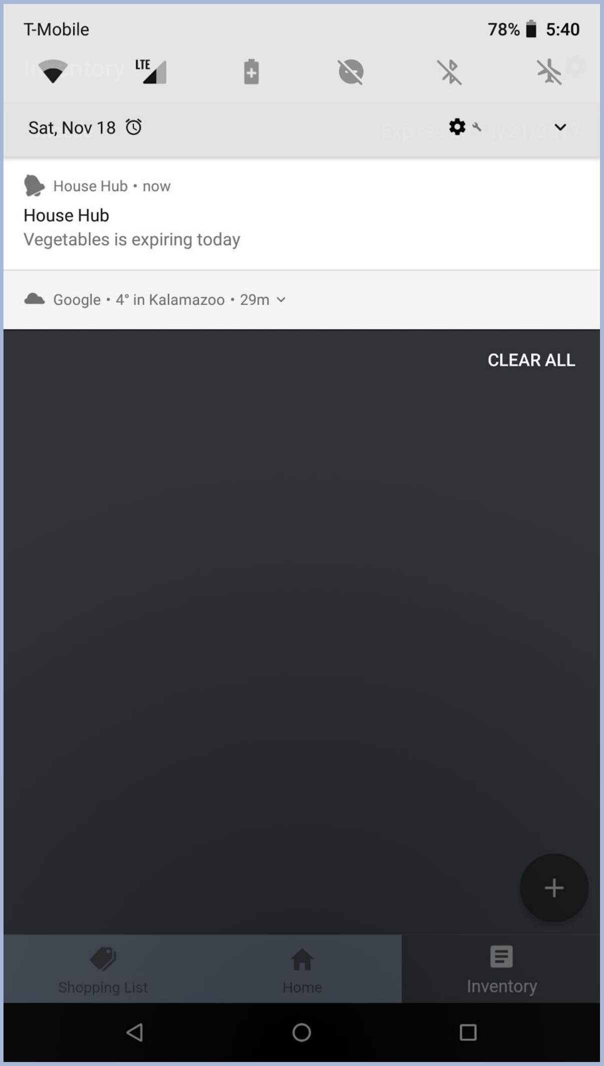


Private

Private Cheese

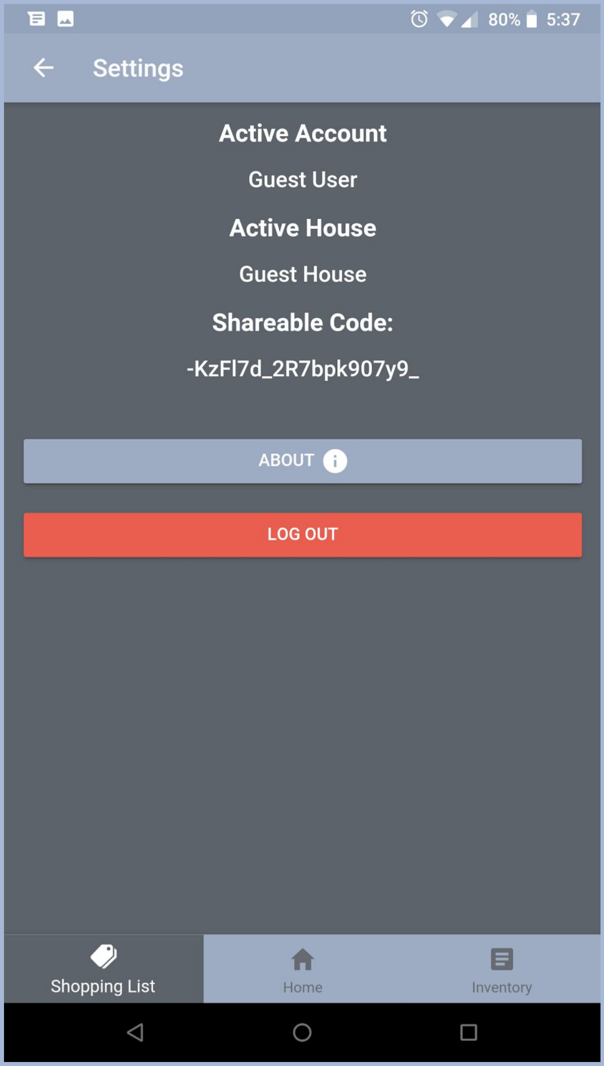
7. Inventory Notifications and Logging Out:

Finally, we show an example of the user receiving a notification for an item (in this case, "*Vegetables*") that will be expiring soon. "*Soon*" means that the item expires that same day. We also show the settings page, which allows the user to view the information related to their account and also gives them the option to logout.



House Hub • now
House Hub
Vegetables is expiring today

Google • 4° in Kalamazoo • 29m



Active Account
Guest User
Active House
Guest House
Shareable Code:
-KzFl7d_2R7bpk907y9_

Database Implementation

For back-end our app utilizes *Firebase*, a real-time NoSQL database,via the *AngularFire* plugin. Data is stored as JSON objects in trees. We separate data into five trees, each of which is described below.

1. *Users:* this tree stores user information. Each user is indexed by a unique key (their authorization ID generated by firebase) and contains four child nodes: *email*, *name*, *householdKey*, and *privateKey*. The variable *householdKey* stores the unique key for the household they are a part of, and also functions as the shareable key for a household. The variable *privateKey* is only used to index the user's private shopping list.

2. *Households:* this tree keeps a list of all active households, indexed by unique keys. The household key is generated at registration, at the Setup Page. Each household has two child nodes, *name* and *members*, the latter containing a list of all users that associated with that household. Currently, the *members* list only stores user names.

3. *Notes-lists*: this tree stores lists of notes and reminders (displayed on the home page) indexed by household key. Each list contains two types of child nodes: a *note* contains only text and a timestamp, while a *reminder* also contains a time and date. Notes and reminders are distinguished by an *id* variable which stores either "*Note*" or "*Reminder*".

4. *Inventory-lists:* this tree contains household inventories indexed by *householdKey*. Each inventory item is associated with two variables, a name and an expiration date.

5. *Shopping-lists:* this tree stores shopping lists, indexed by either *householdKey* (public household list) or *privateKey* (private user list). Each shopping item is associated only with a name, and is currently uneditable.

User Authentication

We make use of *Firebase* out-of-the-box user authentication (also via *AngularFire*). Upon registration with an email and password, a user is associated with a unique user id, which we use to index the user's profile in the *users* tree. To ensure that a user is registering with a proper email, name, and password, we use Ionic's *FormControl* to check that the email matches a regular expression and that the password is between six and twenty characters. This reduces the possibility of an error when we pass the information to firebase. Upon registration a user profile is created, and the user is then sent to the setup page where they can choose between creating a new household and joining an existing one.

A user logs in (on any number of devices simultaneously) with their email and password. Firebase stores a session even when the app is closed, so we are able to keep the user logged in until they explicitly log out. Unfortunately, it takes some time for the app to determine whether a user is currently logged in, so we show a loading screen for three seconds after the splash screen fades out.

Future Directions

In-App Messaging:

Although the app is very useful, it could be enhanced by in-app messaging. As of now, we have designed a framework for managing nearly all important household matters besides interpersonal communication. If this was implemented, users would be able to discuss household matters in a back-and-forth manner rather than solely relying on the different lists. This would also make the use of other group messaging apps unnecessary, further simplifying the user's experience.

Applications in Other Contexts:

Since the app demonstrates immense potential for managing inventory and group responsibilities, it could be extended to business contexts. For instance, warehouse managers could use a modification of the app to keep track of their inventory and to delegate purchasing tasks. Moreover, the group reminders could be effective for helping team members stay focused on their goals. The app could also be extended to create a better process for patient-doctor interaction. For instance, a doctor could set reminders for and view the reminders set by their patient. This would be especially useful for ensuring the safety of elderly users who are taking medication on a regular basis. Given its generalizability and functionality, the app has the potential to be extended in a variety of other contexts.

Documentation Log:

10/2/17

- Considered adding note functionality on Home tab for top-priority reminders.
- Column setup will follow example given during group meeting and have a two-column format through the Ionic grid system.
- Decided to follow through with note function on Home tab, added button for said function.
- Notes and Reminders are now said to be different in terms of functionality. Notes will remain on the Home tab at all times until changed or deleted, Reminders will be temporary and will use notification.

10/3/17

- Added Shopping List page for shopping list functionality, began implementation of note functionality and UI into Home page
- Considered using different symbols for different food types

10/4/17

- Began exploring Firebase database integration
- Began shaping Inventory page UI and further modified note functionality with addition of the "Add Item" option to create new notes and items on pages
- Fully implemented the ability to add inventory items on Shopping and Inventory pages, more tweaks to Inventory UI
- Adding items on Home and Inventory page brings up a slide-in (fab) menu with appropriate options to add items based on what page is currently displayed, and brings users to a separate page to specify what it being added.
- Added options and functionality to allow Inventory items to be edited and deleted

10/9/17

- Further tweaking of UI of overall app by changing a variety of variables
- Fully added note functionality into the Home page, explored UI options for notes

10/10/17

- Added ability to delete items from the Shopping List and Inventory pages
- Implemented sliding delete functionality
- UI color changes, cleaned up code and updated angularfire code to match docs.
- Began exploring Notification functionality

Angular 5 update hits, had to deal with the overall changes by relearning the language and also having to manage the update's repeated delays.

10/18/17

- Began exploring button functionality
- Begin implementation of Public and Private lists for the Shopping List page, still not fully functional
- UI changes to list colors, padding, line separators
- Fixed database issues with async
- Overhauled Inventory UI, simplified to avoid unnecessary complications
- Added functionality to adding Inventory items to show the amount of weeks left before expiration, rather than expiration date explicitly.

10/20/17

- Reviewed changes made on 10/18, cleaned up code a bit

10/22/17

- Began implementing Settings Page
- Added buttons to access Settings page from Home, Inventory and Shopping List pages
- Minor fix to Database issues, still running rampant

10/23/17

- Implemented a Login/Logout system with the use of email as a username and a password system
- Considering using Facebook as a API for the login procedure to be able to easily connect with those of the same households
- Considering having Households use a shareable ID code to differentiate them with each other and to avoid strangers from viewing said information. Private lists in Shopping List should supposedly still be visible to the user.
- Added Logout button, About Info (with placeholder text), Shareable Code and Active House name to Settings page
- Tweaked Logout code to improve functionality
- Imported settings from settings
- Implemented fab-close on tab switch so that the fab menus close if the user swaps between the Home, Inventory or Shopping List pages.

10/25/17

- Fully implemented separate public/private shopping list functionality mentioned before
- Added dropdown menu on Shopping List page
- Began working on having note ordering be determined by timestamp
- Modified Inventory items to show the amount of days left before expiration

10/30/17

- Added a Register page to allow users to be fully registered to an active household, retained ability to log into app as a guest user
- Added Active Account name to Settings Page
- Minor visual tweaks to Settings and Login pages, Home page UI changes now shows buttons on notes for editing and deleting them

10/31/17

- Global UI changes, mainly colors, text, visuals

11/1/17

- Began implementation and added Setup functionality and page to be able to allow users to set up different Households for different user groups.

11/2/17

- Tweaked Settings and overall app to consider updating info shown when switching between households.
- General UI and color tweaking, reworked Home page UI a bit more
- Pushed Setup addition and UI changes
- Revamped and reworked the ability to edit items in the Inventory tab.

11/3/17

- Tweaked and reverted tweaks made on the Home and Register pages regarding updating objects.
- Moved the code for signing in directly to the database provider.
- Hotfixed a bug related to registration in the database provider and the Household name display not properly functioning by adding a few checks via booleans.
- General code clean-up.

11/5/17

- Began implementation and added ability to add Reminders on the Home page, alongside adding the ability to set Date/Time for Reminders.
- Added the ability to be able to edit Notes on the Home page (finally).
- Modified the onclick function for adding Notes and Reminders on the Home page.
- Added date storing to the ability of adding Inventory items. Currently set to store dates in milliseconds, but should change to days soon.

11/6/17

- Added datepicker and datepicker UI for the add reminder functionality.
- Added ability to join/share households with other users using the Household ID. The ability to join/share was added to the Setup page.
- Inventory lists are now linked to households.

11/10/17

- Fixed an issue around the Login functionality, removed the “placeholder” email/password login code for an email/password system with a proper authentication method.
- Minor UI changes to placeholder Alert system.

11/12/17

- Added proper documentation in code, particularly in Home page.
- Added Status Bar code to all pages to change the color of certain UI aspects that are unique to iOS and Android to improve overall UI design.

11/13/17

- Further tweaking of Login functionality and around database, added and reverted async code to Login and the database.

11/15/17

- Added code to Settings page to display the proper Household name.
- Added more code commenting for organization.
- Can now join an existing household and be added to the list of users associated to household.

11/16/17

- Linked the Shopping List to the individual user for testing purposes, discovered that some user accounts can have public items that persist between accounts and households altogether
- Reworked the registration process by using alternative lines for the account creation and email/password input.
- Reverted earlier changes to Shopping List and reworked the public/private dynamic to ensure past problems don’t resurge by using alternate TS functions.
- Ran into an issue where users that had invalid email addresses would cause the database to glitch and delete all users on the app. This, alongside an improper unsubscribing problem, were dealt with reworking the code around the userRef object.

11/17/17

- General code clean-up session, removing a lot of scrap code documentation, testing lines, unused import statements, as well as junk code in the Register, Login, Setting and Shopping List pages on top of database code.

11/18/17

- Reworked the Register page to call for an error to ask the user to provide a valid username and email address to continue fixing bug from 11/16, as well as dealing with “broken” Households that had empty/null value/strings for Household Names and Shareable Codes. Also added code to call for errors to provide valid names, valid passwords and so on when logging onto an existing account.
- Completely reworked the code for Reminders so that they now perform the functionality that was expected of them: Reminders can be now created alongside Notes on the Home page and can be assigned to a time and date to send a notification to the user when that time and date is met.
- Added import statements and used the itemRef object to rework the code for editing inventory items and the related data assigned to them.
- Code for Inventory items has been changed so that when you add a new item to the list, rather than setting the amount of days or weeks that the user predicts that the item would expire, the code was changed to call for the user to pick the expiration date altogether in terms of days before the date. When set, the item displays the date based off of the amount of days chosen. When the date is met, the user should now receive a notification that will inform them that the item in question will expire on that day.
- Changed the Home page UI to shrink the buttons on Notes and Reminders for editing or deleting them. Also changed the About message in the Settings page to replace the placeholder text with a proper summarized description of the app’s purpose.
- Replaced empty image files with proper application splash art, app icons, and whatnot featuring HouseHub’s very own logo.

11/19/17

- Modified login code to allow the app to automatically log users in should the app be left but not shut down.
- Added a few files and code for a loading page.
- Reminder notifications discovered to be not functional. Fixed by tweaking code to convert date and time objects into strings.
- Discovered a bug where the user could not edit notes.
- Added a few lines for variables to the Home page to include text to show what tags are either Notes or Reminders, Reminders now show the date and time assigned to them on the Home page. Done so using variables to be assigned as ID variables to show the differences between the two object types.
- Modified Home page UI to enlarge the Notes and Reminders' assigned text to make them stand out a bit more.
- Tweaked database code and enabled notifications to be displayed for Reminders, previously not functional.
- Bug discovered where Reminder notifications pop for reminders on a specific day, regardless of which YEAR was specified. I.e. You would receive a notification on November 19, 2017 despite the fact you set the reminder for November 19, 2018.
- Fixed the edit note issue that would either delete or do nothing by copying and pasting the code for adding a new note, and tweaked it to replace all "add" code with "update" statements.
- Added the ability to edit reminders by having the code to edit notes actually edit both notes and reminders. A check would be made to see whether the object being called to edit is either a note or reminder and initiate the editing accordingly.
- Minor edits to icon pictures, for quality.