

fridgit 

## TEAM

Brian Hoang -- Participant Analyst

Colin Miller -- Development

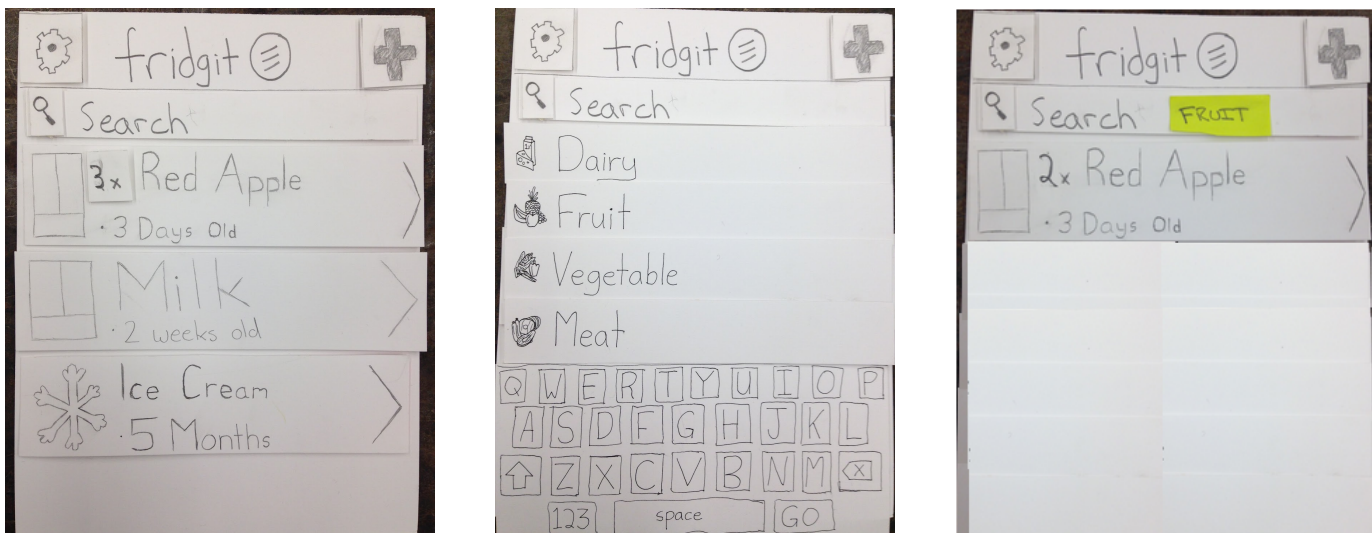
Francis Nguyen -- Product Manager

Nick Holden -- Ideation

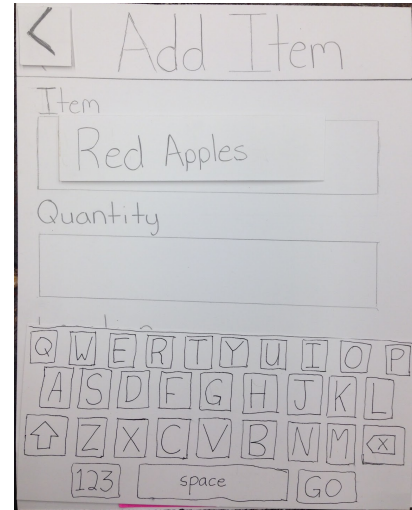
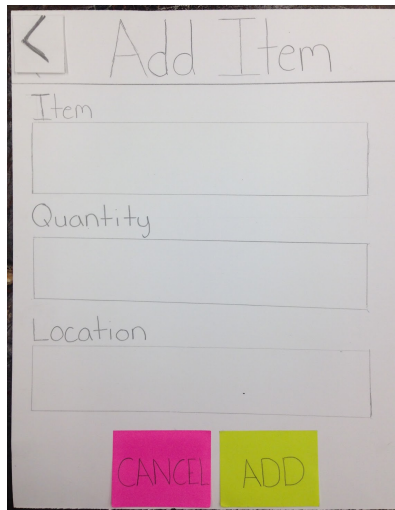
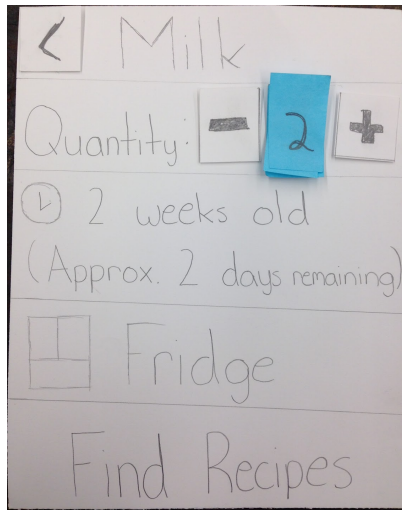
## PROBLEM & SOLUTION OVERVIEW

Our product is a smart refrigerator that keeps a user's food inventory for them. This refrigerator, the Fridgit, acts as a traditional fridge, but also houses an internal computer that can tell a user how long an item has been in it and if there are any items that the user should use soon. To make the process of adding items easy, the Fridgit's computer can track the movement of items in and out of it by using two internal cameras. An external display allows manual alteration of the food database.

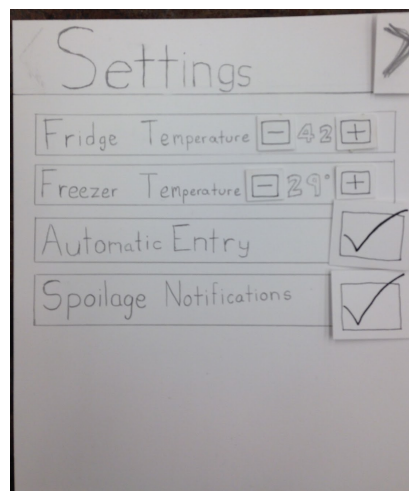
## INITIAL PAPER PROTOTYPE



The home screen of Fridgit is the food inventory, which addresses **Task 2: Checking the Food Inventory**. It instantly shows the user what items they have in their fridge and freezer and has refined searching methods for easy access. If the user wants to see just the fruit they have, they may choose it from the predefined food categories provided before the search menu. For a more sophisticated search, the user may also opt to type in a search term.



Fridgit makes it easy to **Add and Remove Items (Task 3)**. By default, Fridgit uses its internal cameras to track entry and removal of food items, but for items that aren't easily recognized, like leftovers or food in containers, a user can easily add and remove items manually. This gives users flexibility in adding food that might not be stored in the fridge. Fridgit's food inspector tells how old items are, where they're located, and allows the user to easily search the web for recipes with that item. Addition of new items, like red apples, is a cinch; Fridgit will then learn what this new item looks like when it enters the fridge so that it knows for next time.



It's easy to control Fridgit; simply go to the settings page. Users can adjust the internal temperatures of the fridge and freezer, toggle the automatic entry of food items via Fridgit's cameras, and toggle spoilage notifications.

# TESTING PROCESS

## USABILITY TEST #1

Our first usability test involved a participant who is a senior at the UW. He was selected because he is tech-savvy and thus able to provide good feedback on how to alter our user flows. The test was conducted in the user's house. For our test, we decided to adopt a talk-aloud protocol so we could better understand the initial reactions of users. Colin was the only team member for this test, so he played the role of observer and computer.

## USABILITY TEST #2

The second usability test took place with a current Seattle Central student in his home. As he is currently studying Web Design, he was likely to have relevant User Interface design feedback. We felt that performing the testing in his own home would encourage him to think about his own food habits and how they could affect the design. Nick performed this test, introducing the participant to the basic premise of the application and each page. The participant was asked to perform basic tasks, and asked to give feedback on each component of the design.

## USABILITY TEST #3

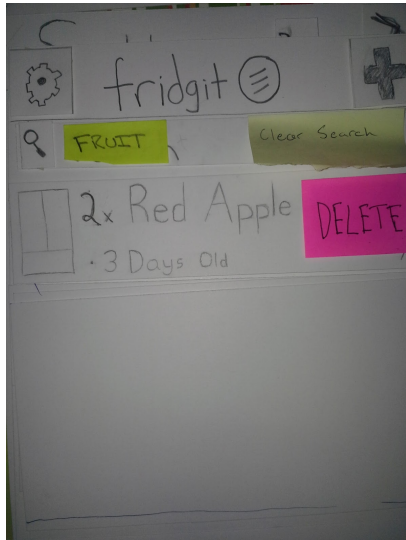
Our third usability test took place in the CSE labs. Our participant is a CSE major in her final year. We chose our participant because she had not taken CSE 440 yet and could provide a different perspective than a student versed in design. Our environment was a result of convenience; both our team and the user wanted someplace familiar and convenient. The testing protocol differed from the others; after a brief overview about the project, the user was told to play around with the interface instead of being given specific tasks. We chose to do this because we hoped it would reveal design issues that we hadn't previously noticed. Brian served as the computer while Colin observed and took notes.

## Retrospective on Design Refinement

Based on our tests, we found that all of our approaches were helpful in finding issues. The first two tests revealed a lot about our tasks and how we could make them easier to complete. Our test subjects pointed out several features that the add item screen lacked and gave great suggestions on how to improve the search and item deletion flows. Our third participant, due to its unstructured nature, pointed out a lot of inconsistencies and raised some great "what if..." scenarios.

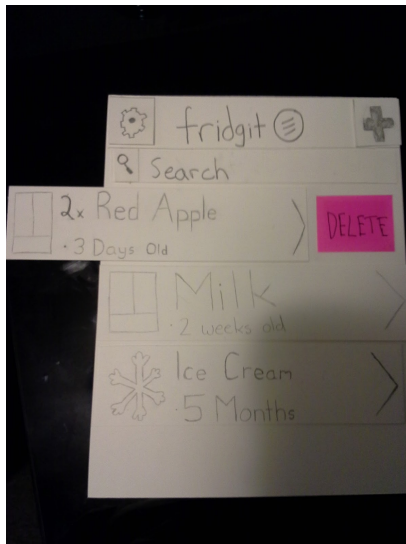
# TESTING RESULTS

## HEURISTIC EVALUATIONS



### BROKEN SEARCH

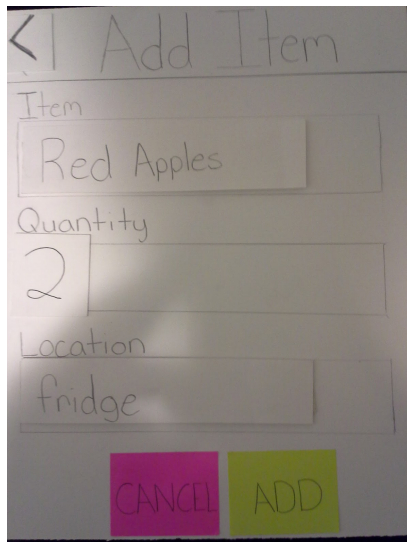
Our heuristic evaluations revealed a critical issue with our initial design: The user could not clear out a previous search performed on the main screen. Our team marked this as a major usability issue and fixed the problem by adding a “clear search” button to our next iteration of the design.



### DELETE ITEM FLOW

In our initial design, we intended that users could delete items from their main screen by swiping left on a food item. However, our heuristic evaluations revealed that this was difficult and unintuitive. The arrow sign on each item made participants swipe right or click. We marked this as a minor usability problem because there were other options for a user to remove an item. Our team addressed the issue in our next design by removing the swipe option and instead opting for a delete button on the item’s details page.

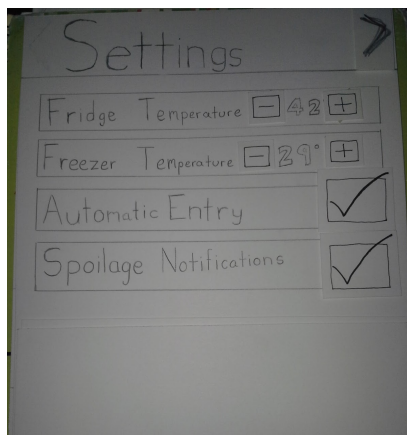
## USABILITY TESTS



### ADD ITEM SCREEN

From our usability testing, we learned that users expected a drop-down menu to appear when adding an item. Our team marked this issue as low severity because it did not impede the user from using our product. The initial design had a simple input box, but our next design iteration added a drop-down of previously entered foods.

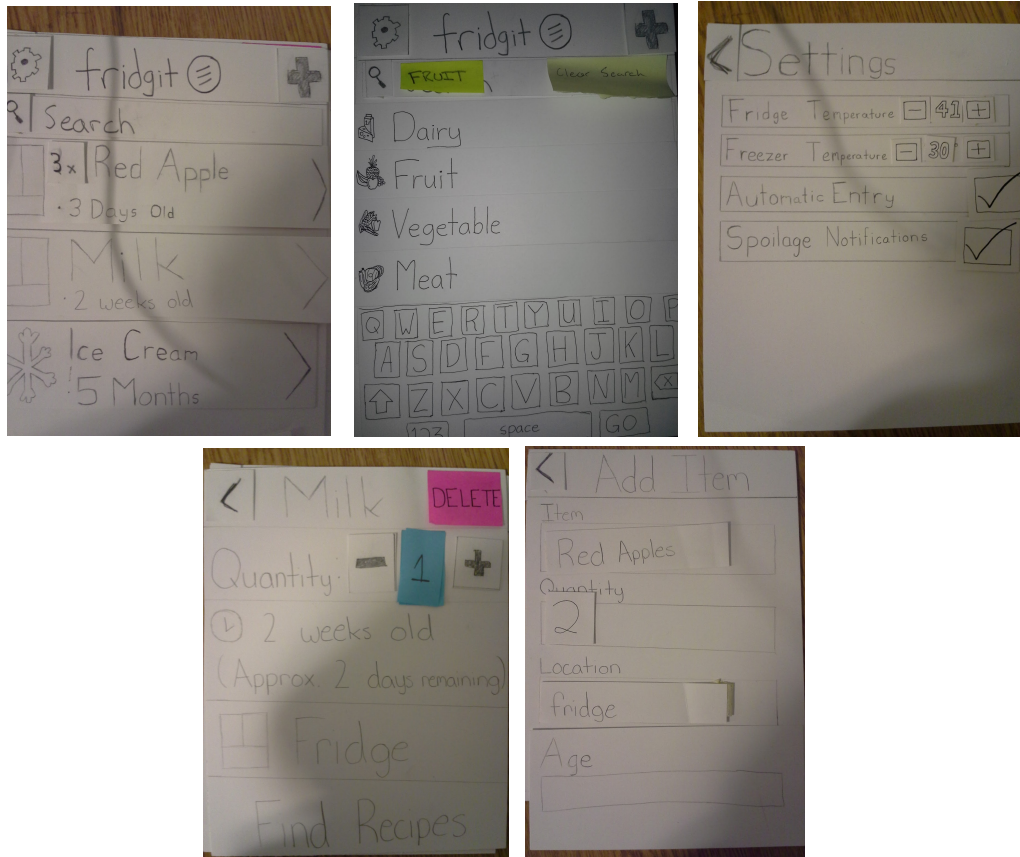
Another issue raised in our usability tests with our design was that users could not input an age other than brand new. Our team ranked this issue as a moderate usability problem because it prevented the user from adding pertinent information about an item. We addressed this by adding additional input fields to the Add Item menu in our final design.



### SETTINGS SCREEN INCONSISTENCIES

Our final usability tester noted that the Back button doesn't make sense in the top right corner, and should instead be in the top left. Our team marked this as extremely low severity because it causes no usability issues, but is still a minor inconvenience. We fixed it moving the button to the left side in our final design.

# FINAL PAPER PROTOTYPE

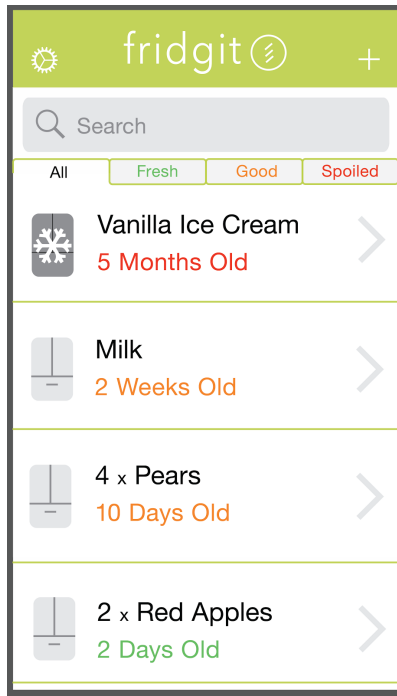


We made every alteration to the design by keeping our two primary tasks in mind: Quickly checking the food inventory and easily adding/removing items. Our design still retains its focus on the former by presenting the user with their items on the home screen and allowing more specific searches for food items. Our prototype did not take advantage of color, but our intention was that items on the home screen would be sorted by “freshness,” descending, so users would know what was about to go bad instantly.

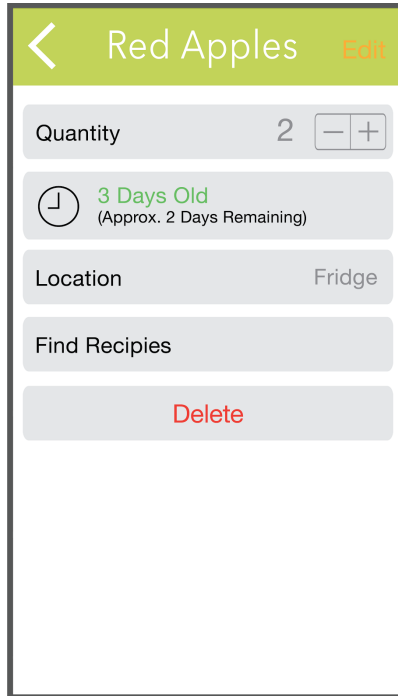
In regards to our second task, our design’s most significant changes occurred in the “Add Item” screen. Our fridge focuses on being automated, but we need this screen for unrecognizable or difficult items. The user is now presented with an ‘Age’ option, which defaults to 0 days, and can be used to set how old food is. The ‘Location’ option is a drop-down instead of a text input, since there are three main options. These changes add flexibility and make the interface easier to work with.

The other screens also have small changes of their own. One occurred on the main screen, with a “clear search” button appearing after a search has been performed. Another occurred on the item details page, with the delete button appearing on the top left of the item’s page. Finally, the settings’ page back button is now on the left side, to keep it consistent with all the other pages.

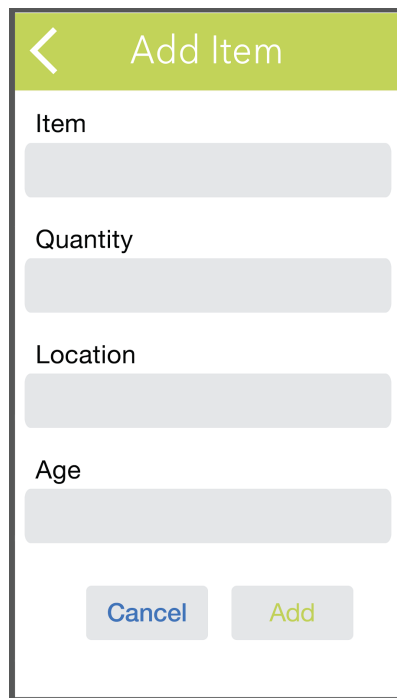
# DIGITAL MOCKUP



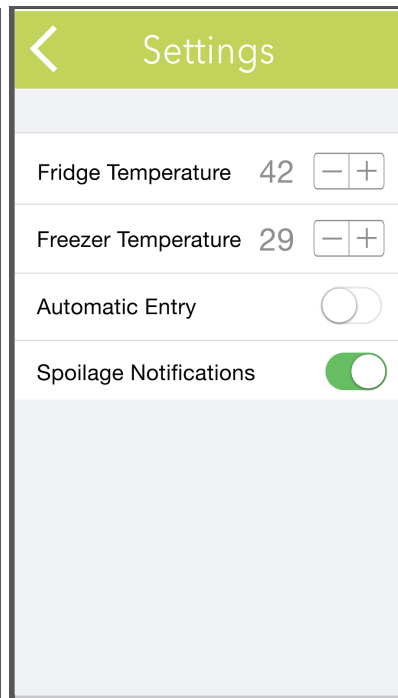
Home Page



Item Info Page



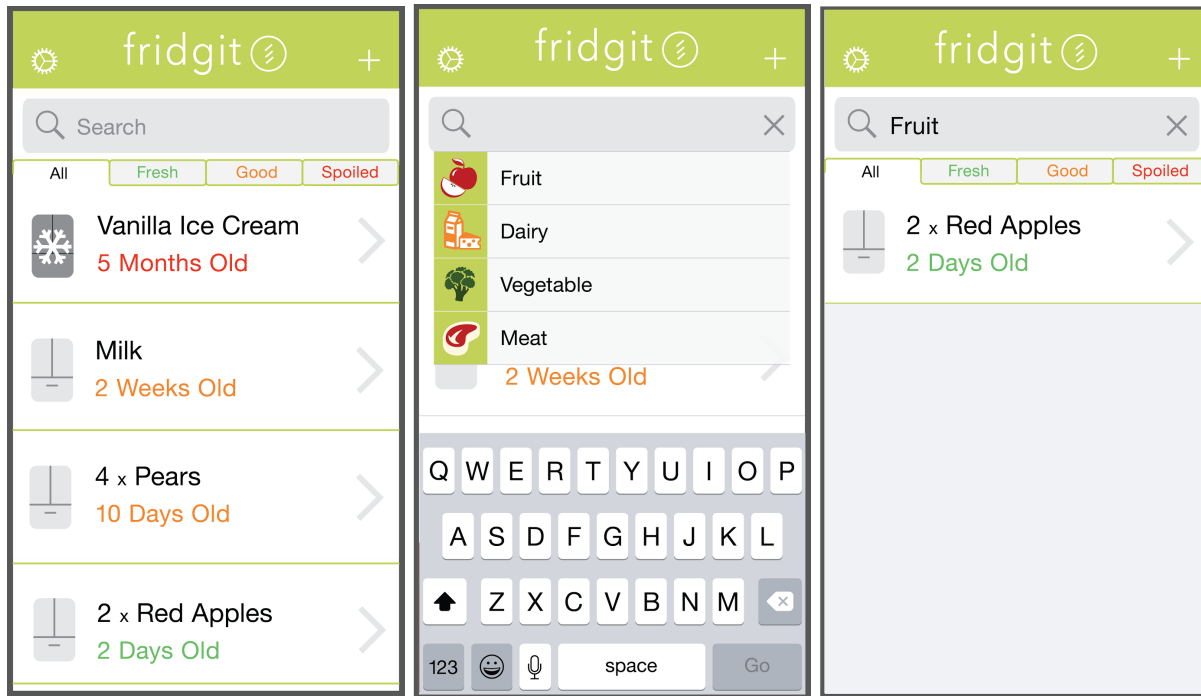
Add Item Page



Settings Page



## WALKTHROUGH 1: CHECKING INVENTORY

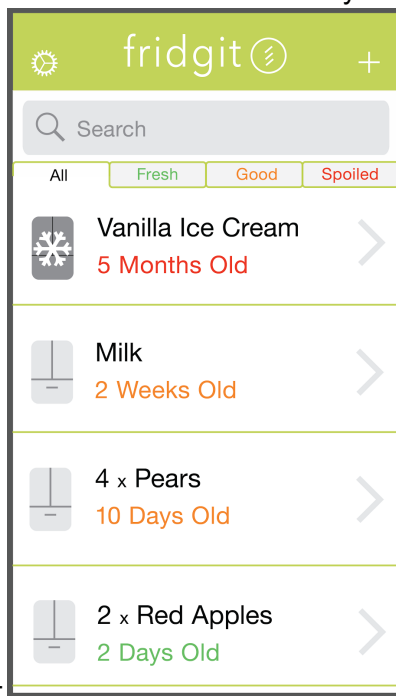


The main page of the app shows a list of items in the user's inventory, sorted by freshness (spoiled to fresh). Users can tap on an item to view its details. A search bar at the top allows users to search for items, or choose from a list of provided food categories. When a search is performed, the search results are listed. Users can tap "X" to clear the search and return to the inventory page.

## WALKTHROUGH 2: ADDING & REMOVING ITEMS

The image shows two side-by-side screenshots of a mobile application's 'Add Item' form. The left screenshot shows the form with empty input fields for 'Item', 'Quantity', 'Location', and 'Age'. The right screenshot shows the form filled out with 'Cheese', '3', 'Fridge', and '1 Day Old'. Both screenshots have 'Cancel' and 'Add' buttons at the bottom.

Users can easily manually add food items to their lists. They are brought to the add item page. They can take notes on the item's location and quantity. The fridge would, however, be responsible for empty boxes to enter information about the item's location and quantity. In order to manually add items, users can then cancel or confirm the addition of items by the buttons at the bottom of the page.



Users can click the "+" button

The largest changes during transitioning to digital mockups were adding the colored text to indicate freshness. We discussed it while making our paper prototype but never actually added the colored font until the digital mockup. Something important to note was color contrast between the colored font and the background. In a previous iteration of the digital mockup, a color palette was used that actually disallowed users from easily recognizing the freshness color of the font.

People also noticed this during Friday's critique section, where our peers and course staff brought to our attention several changes that would help our design. They noted to change the color contrast on good/spoiled items to make them more noticeable. Peers and TAs also noted the importance of buffing up the font by a couple pts in order to make it more visible, as well as adding potential grouping mechanisms in order to easily see which foods were spoiled and which were not. They stressed that the metaphors for our fridge/freezer icons didn't exactly make sense (which the fridge and the snowflake) and suggested changing it to add clarity.

## DISCUSSION

Through iterations, we have learned that most of design is not done all in one go. Since initial ideas and oversights that would not go well in the final product are forgiven, we don't have to limit ourselves too much in the early stages of the process. Additionally, subjecting our design to feedback after every stage of iteration ensures that we don't inadvertently make the design worse while trying to fix earlier feedback.

Performing usability tests showed us aspects of our initial design that were confusing or that limited functionality, such as the placement of the buttons and the limited number of fields available when adding a food item. Once we made these changes, in-class feedback prompted us to provide further sorting options so that users can more easily see items in their inventory.

Our tasks have remained the same since Assignment 2, and we have shaped our design around them. Fridgit still primarily addresses keeping an inventory of users' food, easy updating of this inventory, and letting users know the spoilage status of their food items.

Our design process could have been improved primarily by more iterations. Each iteration taught us more about the design process and improved our product, and with more time the design would be further refined to our target audience.



# APPENDIX

## INSTRUCTIONS & TASK DESCRIPTIONS FOR USABILITY TESTS

After a brief introduction to the tester of what Fridgit is:

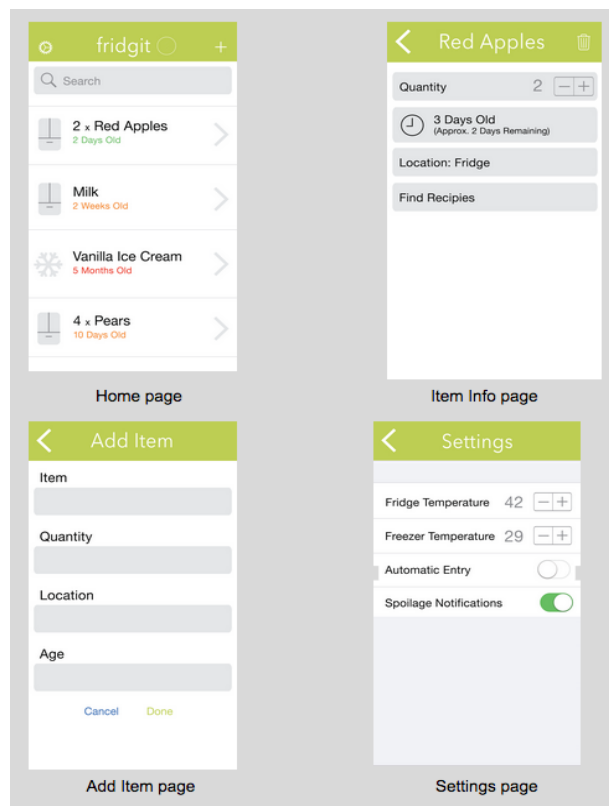
Fridgit has a focus on allowing users to easily access and alter their food inventory. For this usability test, we have six tasks to assess how users interact with Fridgit. We will now present the main screen of Fridgit.

1. Please add another gallon of milk to the inventory.
2. Next, please delete the red apples from the inventory.
3. Now, add the red apples back.
4. Narrow down the food inventory to show just “fruit”.
5. Remove the last filter so that Fridgit now shows everything again.
6. Change the temperature in the fridge to 41 degrees and turn off spoilage notifications.

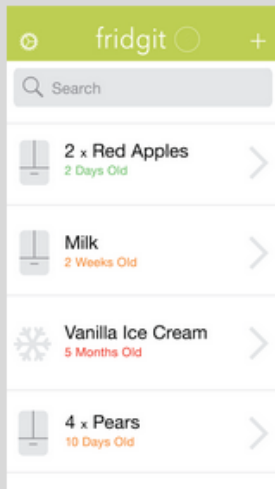
Thanks for taking part in our usability test. Do you have any feedback for us? Please be as brutally honest as possible--we won't be offended!

## ORIGINAL DIGITAL MOCKUP

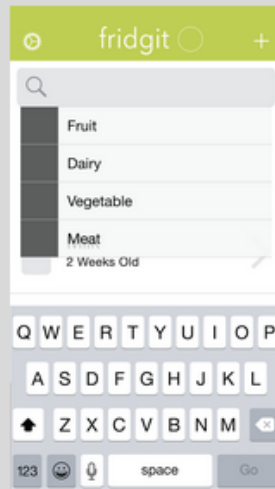
Our digital mockup experienced changes in response to critique from Friday's section. Included below is our original design:



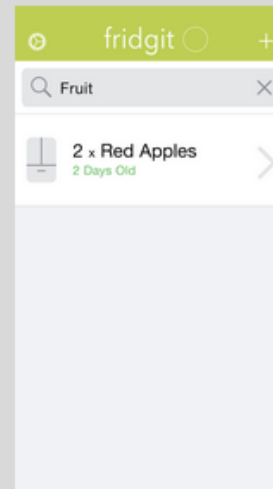
## Walkthrough 1: Checking Inventory



The main page of the app, showing the user their fridgit's inventory, sorted by spoilage date

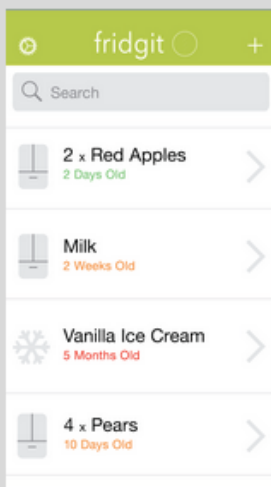


Users can search for items, or choose from a list of food item categories.



When a search is performed, the results are listed. Users can tap the X to return to the main inventory page.

## Walkthrough 2: Adding and Removing Items



Users can easily manually add food items to their inventory even though the refrigerator will mostly be doing all adding/removing food items. In order to do this task, users have to click the plus button.



By clicking the plus button, users can fill in all the details for the food they want to add in the inventory. After clicking the the backspace, the food item will now be added to the inventory.