

# Pantry

CS 130: Software Engineering  
UCLA, Spring 2015  
Application Proposal

Jiexi Luan, 704047806  
Queenie Ma, 504002470  
Wesley Situ, 904050869  
Justin Yuan, 304044541

## I. Motivation

Oftentimes, people find food in their refrigerator or pantry only to discover that it is expired or past the "best by" or "sell by" date on the package. Food and money go to waste, and people have less choices for their next meal. Ingredients are re-bought only to sit in storage again until remembered to be used. In other cases, people find frozen items in their freezer or canned items in their pantry that have been stored there for a while and are unsure whether it is safe to eat.

Pantry will be a web application (that is also mobile-friendly) that will allow users to log grocery items into a digital pantry and remind them when ingredients are about to expire. This will reduce, and hopefully eliminate, waste of kitchen ingredients through neglect and forgetfulness. It will do this by notifying users when their food is about to expire and suggest recipes that uses the particular food item as an ingredient. For foods where the expiration date is not so clear (e.g., fresh produce), we will pull publicly available data to use as a best guess to determine the quality of food.

In addition to indicating when food items expire, Pantry can also set up shopping lists for the user's next trip to the store. Items that have been used or are perished will be added to a shopping list. When the list reaches a specified length, the app can notify the user to go shopping. Items can also be marked as *important* or *optional* to decide if they will be automatically added to the next shopping list. The user will also have the option of manually adding items to the shopping list. This feature will make trips to the market as efficient as possible by preventing users' forgetfulness of which items need to be bought. The app will also analyze the users' usage behavior and recommend the quantity of each ingredient when populating the shopping list.

Our vision is to create a practical everyday application that everyone can use. Users will be able to customize Pantry to their liking. The app will allow users to enter in their own food items and set push notifications. To provide a user-friendly experience, we will make the app visually appealing and easy to use.

## II. Feature Description and Requirements

The Pantry app is intended to be an everyday companion to its user. An example of how a user may interact with the app can be as follows.

- The user creates an account and logs in. The user configures Pantry by inputting some basic information such as room temperature, refrigerator temperature, options on how soon to be notified before an item's expiration, and the shopping list's desired length.
- After buying grocery items, the user adds them to Pantry by selecting them from a list of possible items, or creating a custom item (this will require inputting additional information such as expiration period).
- Pantry will then begin tracking the longevity of an item from its log-in date.

- A reminder will be pushed to the user when an item is about to expire. Recipes will also be suggested to the user based on what else is in the pantry.
- Used ingredients or expired ingredients will be deleted from the pantry and populated into the shopping list.
- When enough items have been added to the shopping list, a reminder will be sent to the user to go to the market.

The following use case diagram illustrates this example.

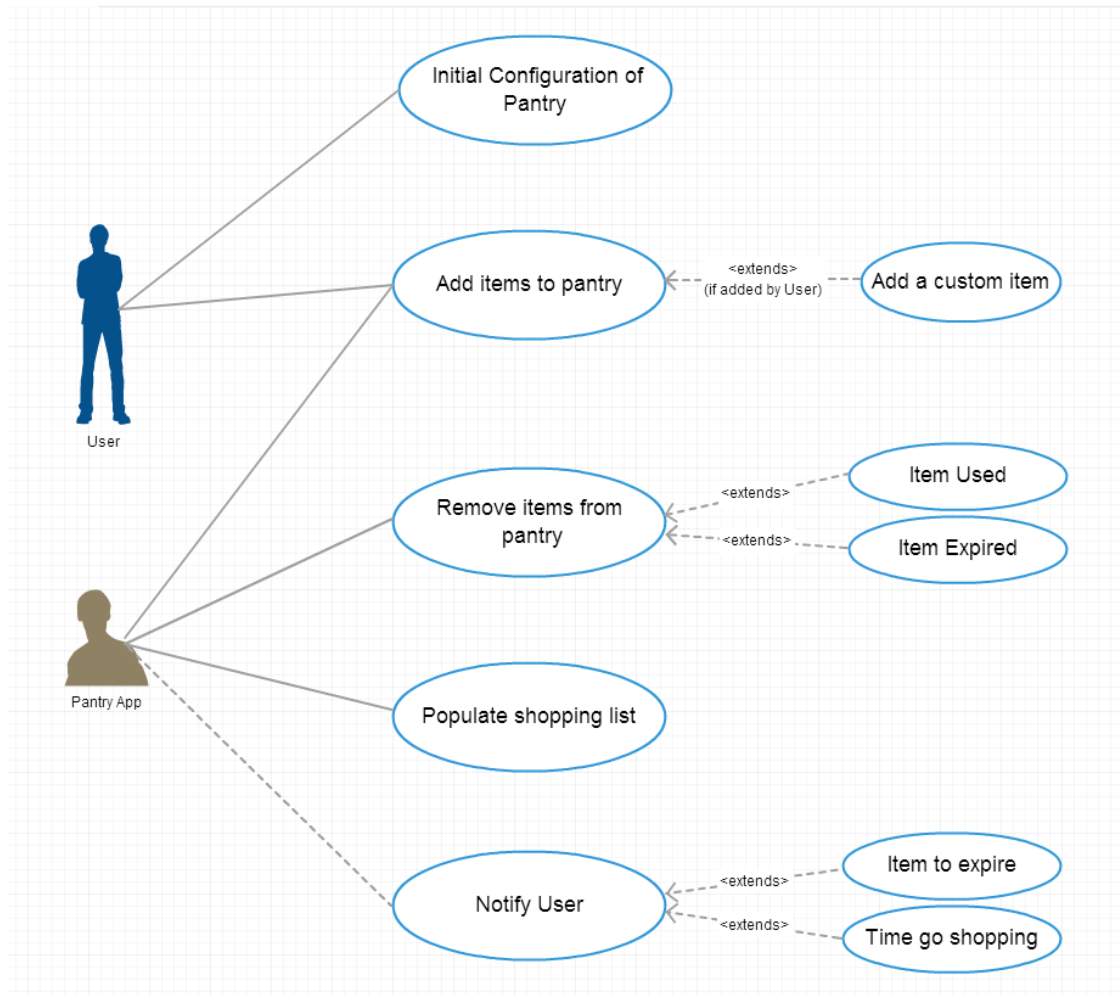


Figure 1: Use case diagram depicting user's interaction with Pantry

Another user story involves the shopping list. When the user decides to go on a shopping trip, the shopping list will have a suggested quantity for each item. The user will check off the items as they are added to the cart.

- The user goes to the grocery store and opens up the application to view the shopping list.
- As the user adds items to the cart, the user taps the item to cross it off the list.

- There will be a pop up that allows the user to input the quantity of the item that is placed into the cart.
- If there are any ingredients that the user does not want, the user can simply swipe the item off the list.
- For items that are not bought, the application will store that information so that the item is not suggested for the next shopping trip.

The use case diagram in Figure 2 illustrates this user story.

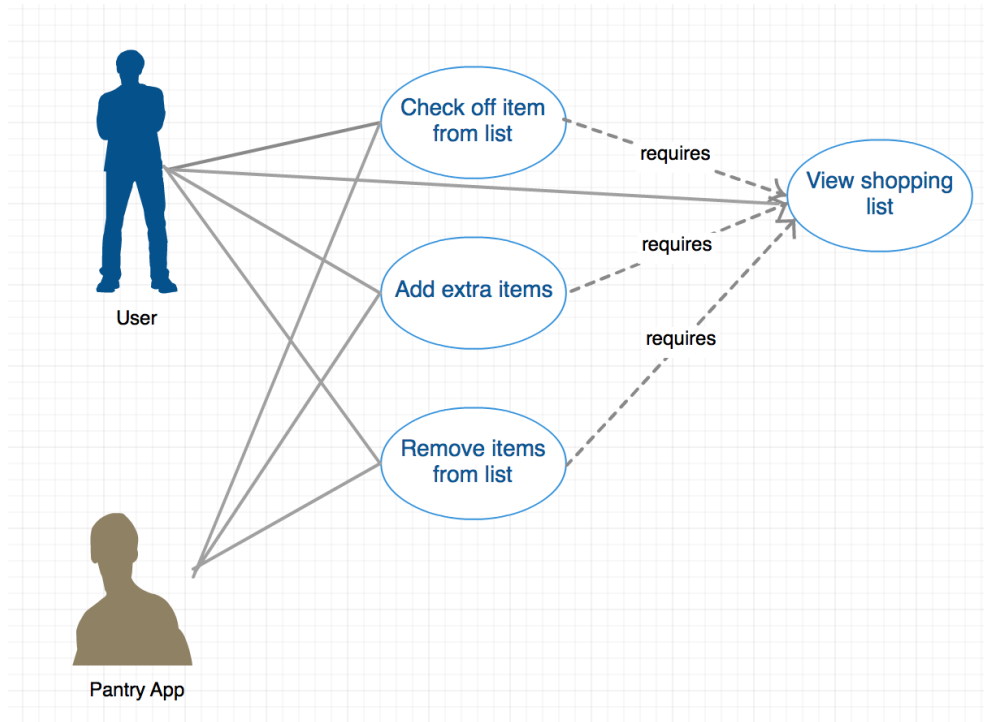
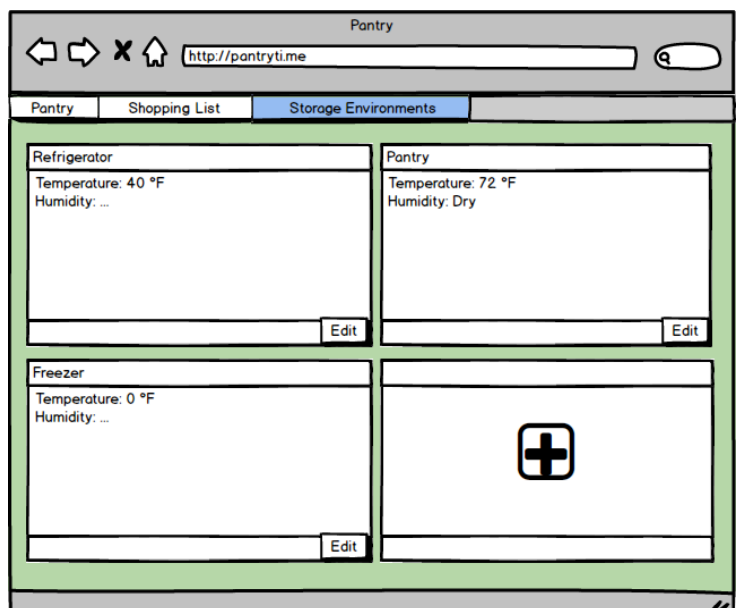
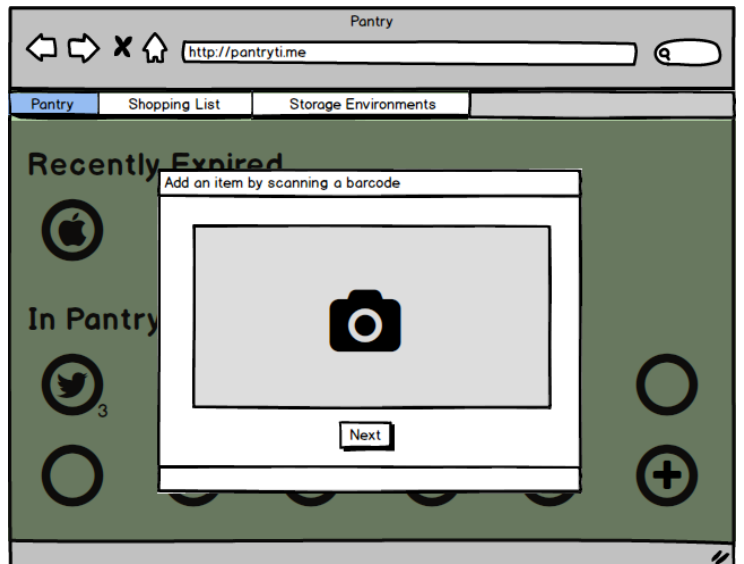
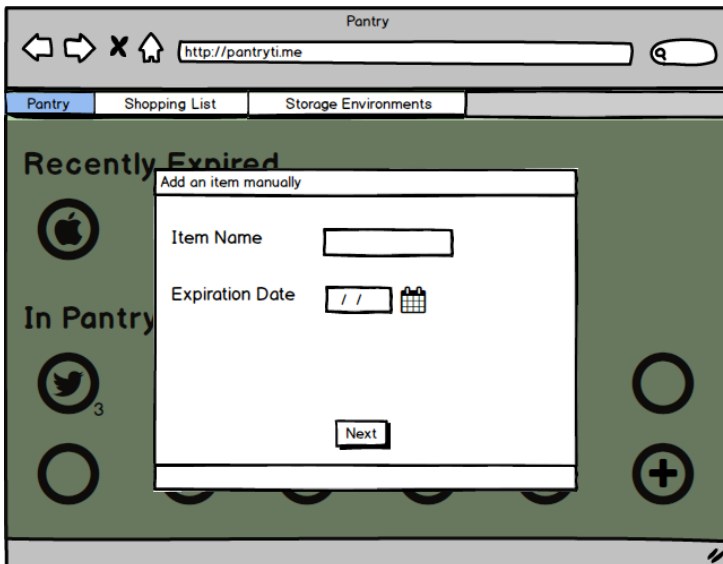
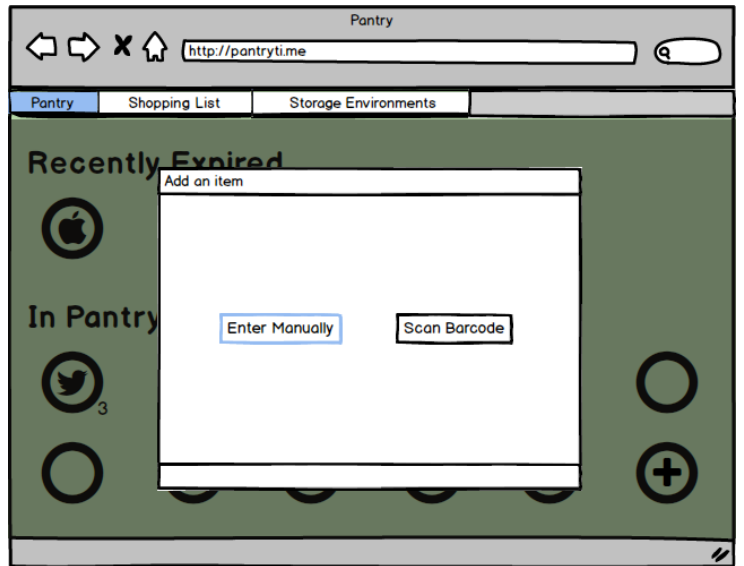
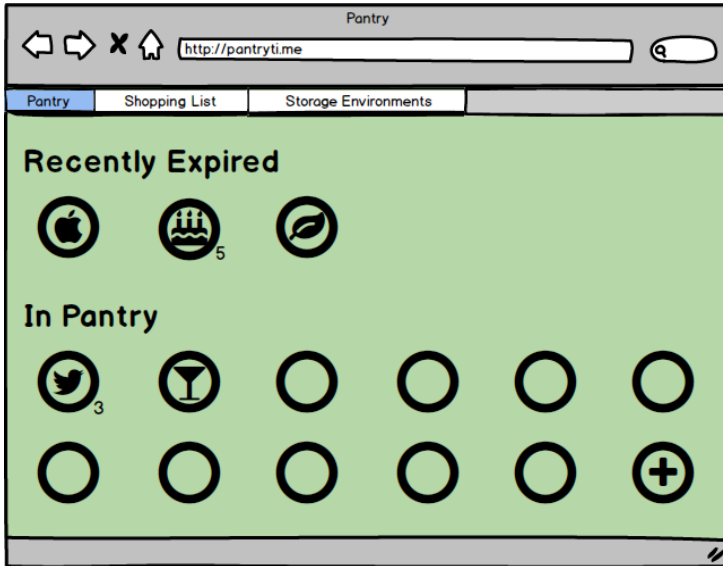


Figure 2: Use case diagram depicting user's interaction with the shopping list

The next page contains mockups of our interface. At the moment, we are planning to have 3 tabs in the navigation bar. The first is the Pantry which contains a grid of food items that the user owns and displays expiring items. Items can be added manually or through a barcode scan, and information for each item can be obtained by clicking on their icons. The second is the shopping list, whose function is explained by the second user story. The third tab lists the environments that the user can store the items in and displays their temperatures and humidity levels. Environments can be added and also edited.



### **III. Feasibility**

Pantry is most similar to a task/todo list because each item being tracked on behalf the user of must eventually be used and removed from the list. We will initially build Pantry as a task list with user account support. Once this is complete, we will add functionality such as quantity and expiration date as attributes to each item in the task list. Expiration notifications and shopping list recommendations will be much easier to implement once basic input functionality is completed. Additional features such as camera input and product recognition that require additional libraries or APIs will likely be completed last.

We plan to make Pantry a web based application with desktop and mobile versions with notifications being served by an email/sms based approach. Pantry will be written in PHP and will serve web requests via Apache or Litespeed http servlets. MySQL will be used for the storage layer of our application. These technologies have been time proven and are currently the standards used when developing any sort of website of web application. We don't expect any issues by using them for as the backbone for Pantry.

Our application also provides additional functionality that extends on task lists. Pantry must deal with food items expiring which will require the usage of a server side scheduler such as CRON. This CRON script will run a PHP script every few minutes to ensure the application can appropriately notify users of food that may become stale in the near future. CRON is the de-facto standard for running server scripts on any sort of schedule without user interaction. It is included in almost all \*nix distributions and should be a simple process to setup.

In order for users to submit pictures of their food items for cataloging, we will use Media Capture from HTML 5 to utilize the camera on our website's mobile version. This will rely on our users using up-to-date browsers on their phone, but allows us to avoid having to branch our code into a mobile application. We expect most users to be using HTML 5 compliant browsers such as Chrome or Firefox on their smartphones.

Interpretation of barcode images will be handled using the ZBarCode PHP library (<https://github.com/mkoppanen/php-zbarcode>). Barcode recognition is reliable given that they are designed for computer consumption. We will use a 3rd party database (UPCDatabase or Amazon) to convert UPC codes into actual product information. Image recognition for items that do not have barcodes will be handled using a 3rd party image recognition service such as CamFind (<http://camfindapp.com>). Because computer vision relies on complex and immature algorithms, successful recognition of products will require that these items already exist in the 3rd party database and that pictures taken depict the product clearly. These functionalities will be the most difficult to implement due to the complexity that may arise from using several 3rd party APIs will likely require the most time after the core functionalities of Pantry are completed.

### **IV. Capability**

Based on our goals for Pantry, we believe that our team is reasonably capable of implementing our proposed application. We feel that each member has had enough experience in web development and general programming to create Pantry. Additionally, majority of us have already worked in a team environment and are familiar with the processes of code reviews and working with multiple members in a large code base.

Queenie has some experience in creating Android applications as she was part of a team that built an app from the ground up for the CS M117 course at UCLA. She focused mainly on the user interface, but also contributed to back-end development. In addition, she has held internship positions at IBM as a Software QA Engineer, where she has gained testing skills (unit testing with JUnit and functional testing) and has applied them to a few IBM products to prevent defects and ensure high quality of code. To align with her experiences, Queenie will be focusing on the overall design and flow, as well as testing major components of the app.

Jiexi has been developing web applications for more than a decade now. He has experience in building and selling several websites, including licenses for his own file mirroring script. His recent projects include an automated fantasy sports betting bot, which uses NodeJS and PhantomJS to navigate and make bets on a popular fantasy sports website. Jiexi is fluent in PHP, MySQL, Javascript, Coffeescript, NodeJS, and Canvas. He is primarily a back-end programmer, but is comfortable using front-end frameworks such as Bootstrap. He has held an internship position as a Software Engineer at Originate, during which he worked on a multidisciplinary team to tackle code sprints and code reviews. Jiexi will work primarily on the back-end, but will shift to other aspects of development as needed.

Justin has experience in building form layouts, entity structures and customizing them using Javascript plugins. He has been exposed to understanding process flows, data layouts, and entity dependencies through working with Microsoft Dynamics CRM. Justin has worked with retrieving backend data and displaying on user-friendly forms and fields. He has also worked with testing flows to ensure entities are created, retired, and carryover data fields correctly. Justin will work on user interface layout and testing the app to make sure features are implemented correctly and unexpected functionalities are removed. He can also assist with creating data and objects for demonstration.

Wesley has experience with web development through multiple projects. He has done back-end development at CCLE as a student developer, where he utilizes PHP and JavaScript to maintain code and squash bugs. He also has worked with Bootstrap to create responsive landing pages. Furthermore, Wesley has experience with Android development through CS M117. He will be using these skills to contribute to both layout design and data logic.