

Perfect Pantry

Team #7

Chiaghame Allen

Sarah Asghar

Nina Dang

Wan Ning Ma

Michel Nwoye-Vincent

Edison Tran

Perfect Pantry

Perfect Pantry is an app designed to help users reduce food waste in their homes by providing tools to better organise, track, and utilise the items in their kitchen pantries, fridges, and freezers. This digital assistant aims to bring efficiency and sustainability to everyday food management.

The program serves as a kitchen inventory manager. It allows users to add, remove, and monitor food items across different storage locations. It is equipped with features like a calendar view, virtual location view, colour-coded expiration tracking, and recipe recommendations based on available ingredients.

This tool is planned for a wide range of users, from busy families to students to individuals living alone, and is especially useful for environmentally conscious individuals looking to reduce food waste and optimise grocery usage. Because everybody's time and mind are usually preoccupied with their daily tasks, like with school and work. Most of the time, they are not thinking about shelf life of their products. As such, the application will help organize and keep track of it for them. It also helps save time with planning on what to use with food items, and especially expiring food items by presenting to the user with recipes that use those food items so that they do not need to do research and find the recipes themselves.

Perfect Pantry addresses the critical issue of food wastage, a global concern with significant environmental and economic implications. By providing an efficient means of tracking and utilising food items before they expire, Perfect Pantry helps reduce waste, save money, and promote sustainable living practices. Additionally, it aids in meal planning and grocery shopping, making the process more efficient and less time-consuming.

The app will be considered successful if surveys of users show they are throwing away less spoiled food after using Perfect Pantry. We aim for users to reduce their household food waste by 15-40%. Environmental impact will also be measured by tracking the reduction of organic material entering landfills from users' homes.

To achieve our goal, Perfect Pantry will be equipped with many systems to help users in tracking their products and which products will soon expire. We plan on implementing warnings on the app to notify the user of any products that are going bad. For example, there will be a calendar view to present to the user on which days specifically their products expect to go bad. As well, users will be able to sort their products in many ways, one of which being sorting by expiration date to see what items are going to expire the soonest.

Overall user stories (Both big and detailed ones):

Iteration 1:

Big Story Description: Initialise Fridge/Pantry With items

Goal of user: The user should start being able to managing their virtual fridge

Priority: High

Cost: 14 Days

User Stories:

Create Fridge

Priority: Med

Cost: 0.5 Days

1. I should be able to create a new fridge/pantry and give it a specific name. This will help me keep track of what pantry space I'm managing.

Add Items

Priority: High

Cost: 2 Days

2. I should be able to add whatever food item I want to my virtual pantry. This will help me know what I currently have available.

Remove Items From Pantry

Priority: High

Cost: 2 Days

3. I'd like to also be able to remove items from the pantry, if I no longer have it available. When foods are consumed or expired, I would need to be able to stop tracking them.

Tagging Food Groups

Priority: Med

Cost: 2 Days

4. I want to tag each food based on their food groups. I.e Fruits, Vegetables, Grains, Proteins and Dairy. This will be useful to know the composition of food items in the pantry.

Add User

Priority: Med

Cost: 2 Days

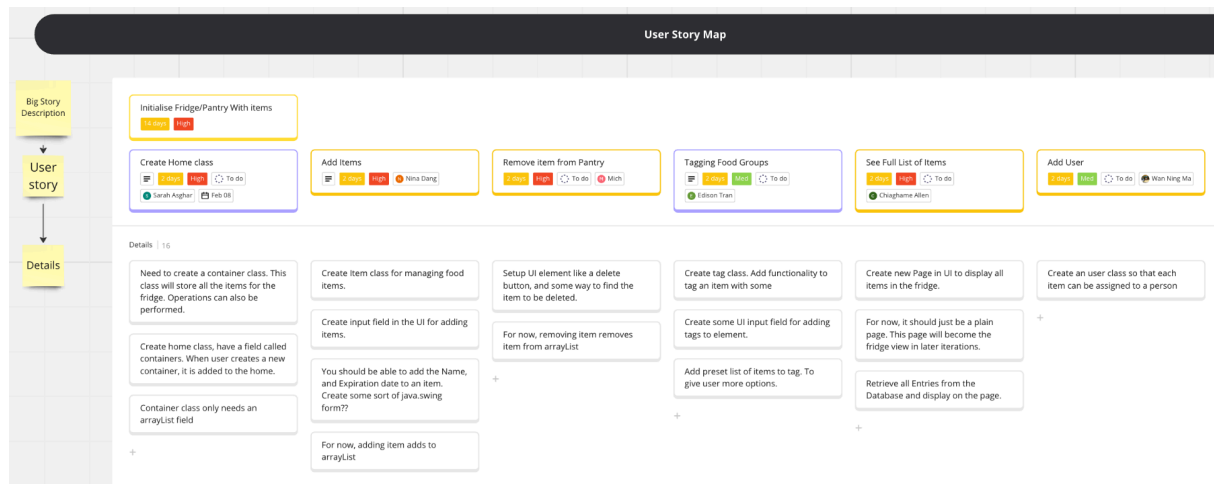
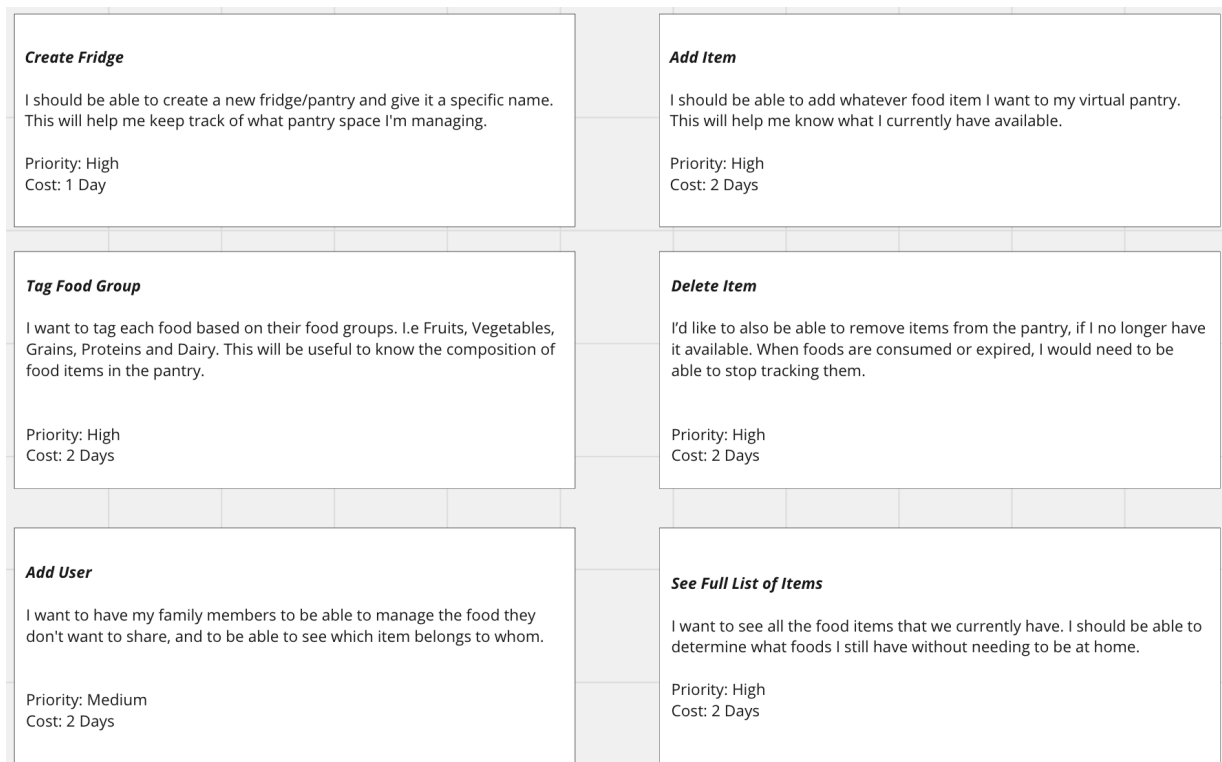
5. I want to have my family members to be able to manage the food they don't want to share, and to be able to see which item belongs to whom.

See Full List of Items

Priority: High

Cost: 2 Days

6. I want to see all the food items that are currently located in my pantry. I should be able to determine what foods I still have without needing to be at home.



Miro View Link: https://miro.com/app/board/uXjVNvII-FM=?share_link_id=150674584264
https://miro.com/app/board/uXjVNvII-FM=?share_link_id=150674584264

Development Tasks

1. Create Fridge (Sarah)

Dev Tasks..

- Create the container class. A container represents a pantry/fridge of items that users can add and delete from.
- Create Home and ContainerView, using java.swing to develop the UI.
- Setup the home to receive user input (name of container to be created).
- Create stub db class.

Estimated Cost: 2 Days

Actual Cost: 7 Days

- Setting up this user story took longer than estimated. It also served as the foundation for other user stories as there must be consistent logic for setting up the UI and data binding.

2. Add Items Class (Nina)

Dev Tasks..

- Create an item class to represent a food added to a user's pantry.
- Setup the AddItemPanel view to receive user input.
- Implement error handling on user input.
- Populate stub db with items.
- Test implementation of adding an item

Estimated Cost: 2 Days

Actual Cost: 3 Days

3. Remove Item from Pantry (Michel)

Dev Tasks...

- Create DeleteItemView to prompt users for the item they want to remove from their pantry.
- Implement error handling on user input to ensure that items actually exist in the database.
- Test implementation of deleting an item

Estimated Cost: 2 Days

Actual Cost: 3 Days

4. Tagging Food Groups (Edison)

Dev Tasks...

- Create a tag class and several "tag" classes.
- Implement tagging test cases

Estimated Cost: 2 Days

Actual Cost: 1 Days

5. See Full List of items (Allen)

Dev Tasks...

- Create ItemsList ListView to display list of items
- Display this view over top the containerView

Estimated Cost: 2 Days

Actual Cost: 1 Days

6. Add User (Ning)

Dev Tasks...

- Setup user class to implement user logins in future iterations

Estimated Cost: 2 Days

Actual Cost: 1 Day

Iteration 2:

Pantry Awareness (This big story aims to help the user make more informed decisions about what is in their pantry, by developing the virtual fridge)

Colour Coded Fridge Items

Priority: Low

Cost: 1 Days

1. Colour coded fridge items would be extremely beneficial. For example, food items that are going bad could be coloured red. This should give a sense of urgency for me to use them. The food groups could also be colour coded.

Calendar

Priority: High

Cost: 2 Days

2. I want to see all food expiry dates in a calendar format. This will help me have a better idea of when foods will go bad. This should help me make more informed decisions about what I cook.

Manage Grocery List

Priority: Med

Cost: 2 Days

3. I definitely would like to have a grocery list feature implemented. When I see that food is running low or has expired, it would be great to add them to a grocery list.

QUESTION: *Is it better to have one big grocery list or one per pantry/container?*

- One big list is preferred

Generate Food Storage Tips

Priority: Med

Cost: 1 Day

4. I'd like to be able to choose an item and get proper food storage tips. This will help me ensure that my food lasts longer.

Sorting

Priority: Low

Cost: 4 Days

5. It would be nice if I could sort foods based on food groups, and also based on the days that they will expire.

Notifications

Priority: Med

Cost: 2 Days

6. Whenever I open a pantry, I'd like to receive some instant notification on the app. This should mostly be for foods that are about to expire. This should help me be alert and aware of foods that are about to go bad.

Development Tasks

1. Colour Coded Fridge Items (Nina)

Dev Tasks:

- Colour coded item entries based on food freshness
- Auto assign Food Freshness Tag based on distance between current date and expiry date
- Make enums in database and enums in food group and food freshness consistent
- Allow Food Group tag dropbox to update items enum in database
- Display the item tags on ItemListView
- Refactor itemUtility methods such as verifyAddItem to return a boolean, making it consistent with verifyDeleteItem.
- Add ItemUtility JUnit test
- Add integration test for RecipeUtility

Estimated Cost: 1 Day

Actual Cost: 1 Day

2. Calendar (Sarah)

Dev Tasks:

- Create Calendar view
- Retrieve items and corresponding expiry date information from the database, and display it in the calendar view.
- Refactor SeeContainersView to use right click to edit or delete containers instead of a separate page

Estimated Cost: 2 Days

Actual Cost: 1 Days

3. Manage Grocery List (Ning)

Dev Tasks:

- Implement add, remove, and export functionalities.
- Design GUI components for interaction.
- Enable strikethrough font for crossed off items.
- Handle user interactions and events.
- Ensure data consistency and error handling.

Estimated Cost: 2 Days

Actual Cost: 2 Day

4. Generate Food Storage Tips (Michel)

Dev Tasks:

- Update database schema to include table for storing food tips
- Setup right click popup menu with generate storage tip option.
- Initialise storage tips database with some values for the user.
- Create a utility function to query the database with the name of the item that the user wants a tip for.

Estimated Cost: 1 Day

Actual Cost: 2 Days

5. Sorting (Edison)

Dev Tasks:

- Integrate database for storing a food item in a container
 - Initialise database on program startup to load each item to their respective container from the database
 - Create methods for accessing and editing values inside of the database
- Include test cases for database integration
- Implement sorting and filtering functionality to the JTable inside of a container.

Estimated Cost: 1 Days

Actual Cost: 1 Days

6. Notifications (Allen)

Dev Tasks:

- Creating notification that alerts the users of expired or soon to expire products.
 - Update database to allow notification read through the database for expired products.
 - Initialise the notifications to work with newly created containers and already existing containers.

Estimated Cost: 1 Days

Actual Cost: 1 Days

Iteration 3:

Research and Analytics (The point of this big story is to provide the user with information they need about their pantry, which foods tend to spoil quickly, and start offering some ways to utilise fridge items)

Support for Recipes (Based on available ingredients)

Priority: High

Cost: 3 Days

- In order to reduce food waste and create more food options, I'd like to view recipes that involve the ingredients I have, especially ones that are going to expire soon so that I don't have to throw them out.
- 'Star' recipes
- ~~— Get reminders if starred recipes ingredients are low~~

Editable System Settings (Based on available ingredients)

Priority: Med

Cost: 3 Days

- It would be very useful to have app customization options. For example I might not want to get reminders for certain things, or maybe i'd like to change font size of the UI.

Custom Tags and Custom Notes

Priority: Med

Cost: 3 Days

- Having more tagging options would be great especially if I'm sharing a pantry with others. I could tag things with my name so others know it's mine, or even tag special items. Having custom notes could help give me or others personal reminders.

Pantry Composition Graph

Priority: Low

Cost 2 Days

- I'd like to see a graphical overview of my pantries. It would be great to have a quick overview of what types of foods I usually buy and store, and other useful stats.

Show All Items Filter

Priority: High (Customer Request)

Cost: 3 Days

- It would be great to be able to see all of the items in every pantry instead of needing to click into each one.

Add Item from Homescreen

Cost: 2 Days

- I'd like to be able to add new items to whatever pantry I want, directly from the homescreen. This would save lots of time as opposed to opening a container first.

Development Tasks

Support for Recipes (Based on available ingredients) (Nina)

Dev Tasks:

- Set up API client for Spoonacular with Gson, external library for parse through JSON
- Create GUI for recipes recommendation and for starred recipes. Allow right click to remove starred recipes.
- Add recipe related tables to the database and make database queries to retrieve and delete those starred recipes. Also updated stub DB for recipe related operations.
- Create domain objects such as ingredient and recipes
- Make exceptions to API rate limited errors
- Resolve bugs from AddItemView and refactored ItemListView

Estimated Cost: 3 Days

Actual Cost: 6 Days

Editable System Settings (Sarah)

Dev Tasks:

- Create database table that will hold initial setting values as well as be updated when new values are used
- Create GUI component of settings to allow user to easily change settings such as notifications or font sizes
- Refactor GUI Views of previous iterations to be adaptable to different fontSizes and still be usable

Estimated Cost: 3 Days

Actual Cost: 4 days

Custom Tags(Ning)

Dev Tasks:

- Implement database table creation script for storing item tags
- Integrate database operations for custom tags into ItemsListView
- Develop a separate class responsible for handling custom tags
- Integrate custom tag functionality into the application flow

Estimated Cost: 3 Days

Actual Cost: 1 Day

Custom Notes (Ning)

Dev Tasks:

- Implement database table creation script for storing custom notes
- Develop methods in the DB class for adding, retrieving, and deleting custom notes
- Create CustomNoteView to utilize database operations for managing custom notes
- Implement functionality to add and delete custom notes from the GUI

Estimated Cost: 3 Days

Actual Cost: 2 Days

Pantry Composition Graph (Michel)

Dev Tasks:

- Create StatsView gui for displaying the pantry composition data
- Setup database queries to retrieve food group information of all items and items per container
- Extract StatsView into StatsRectangleView and StatsDescView to allow for reusability of code.

Estimated Cost: 2 Days

Actual Cost: 3 Days

Show All Items View (Edison)

Dev Tasks:

- Create AllItemsView to display a list of every item in every container
 - Clickable table that changes the view to the container associated with the row
- Resolve case sensitive storage tip bug

Estimated Cost: 3 Days

Actual Cost: 2 Days

Add Item from Homescreen (Allen)

Dev Tasks:

- Create a new add item button to the existing homeview gui.
- A small pop up shows where food items can be added using name,quantity and expiry date .
- These added items show up in the container of choice.

Estimated Cost: 2 Days

Actual Cost: 2 days

Major Changes

There are a few major changes we made from the start of our planning.

Deliverable 1

- The name of the application has changed from ExpireMeNot to Perfect Pantry. The name fits and sounds better. Additionally it was agreed upon by most of the members.
- We decided to stop working on the user logins for the first iteration of the project. It may be scrapped for later iterations of the project as well. Since this application is for personal use, there is little need for there to be multiple users with access to different containers. Additionally, it adds extra complexity to the project, which may not be necessary for us to consider at this moment.

Iteration 2

- Detailed Fridge View has been replaced in favour of Food Storage Tips. The UI for such a story is very complex, and seems to be beyond the scope of this particular project. Additionally, it isn't something that was highly requested by our user, so it is of lower importance.
- "Add to grocery list" was replaced with "manage grocery list". This user story should be a full implementation of the grocery list (UI, adding, deleting, etc) rather than just simply adding to a list.

Iteration 3

- Recipes were combined into one user story, as opposed to three separate ones. It made more sense to do so and allowed for more stories to be added.
 - Scraped "Get reminders if starred recipes ingredients are low" since the Spoonacular API function that determines missing or available ingredients is sometimes not correct. This is seen in bug reports of recipe ingredient mismatch:
<https://github.com/EECS2311/EECS2311/issues/119>
<https://github.com/EECS2311/EECS2311/issues/118>
- Login system was not implemented in this iteration, it was instead scrapped in favour of custom notes and tags. This way if there are multiple users with access to the same pantry, they can simply tag their own items.
- "Add Item from homescreen" and "Show all items view" were added as user stories in this iteration as requested by the customer. They were subsequently given higher priorities.