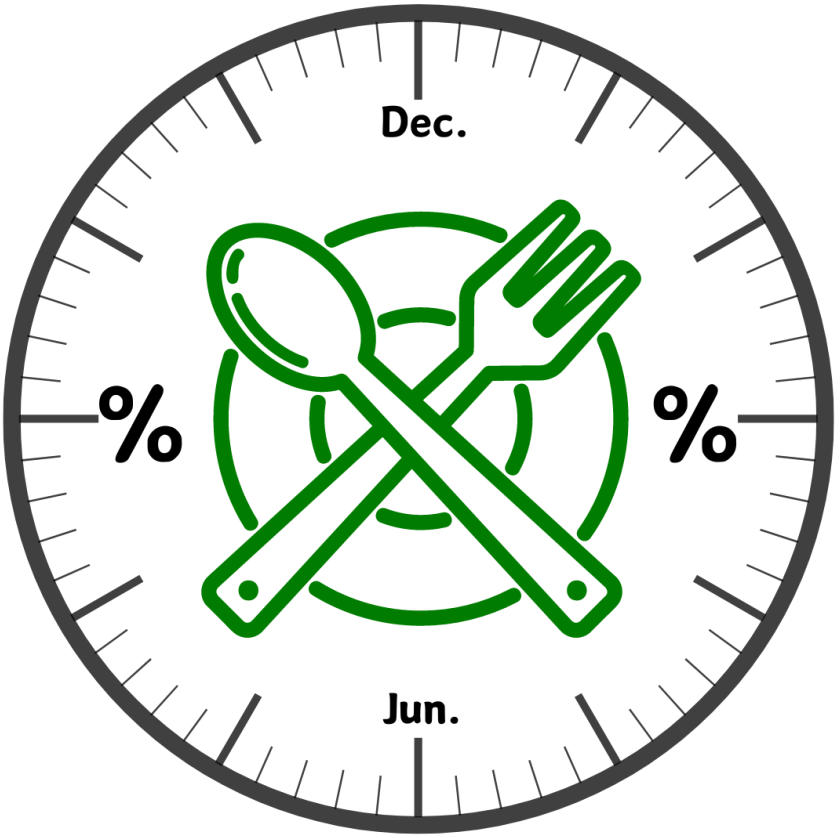

-FOO MY FOOD-

CS 30700
Sprint 2 Retrospective

Team 29
Xiuwen “Charlie” Fang, Xinyi Guan, Songyu “Hathaway” He, Xinyi Wu

What Went Well During Sprint 2?

In Sprint 2, we successfully completed a number of key functionality modules, all of which met pre-defined acceptance criteria for user stories, ensuring that core functionality was accurately implemented. This included the ability to delete inventory items, filter and sort based on category and shelf life, and sync expiration dates to the Apple calendar. By prioritizing these core features, we laid a solid foundation for enhancing the user experience, allowing additional functionality to gradually expand on a stable system foundation. We planned and prioritized our tasks so that we could efficiently deliver high-quality results within the timeframe we set. Despite some technical challenges in calendar synchronization and front-end/back-end data interaction, the team was able to solve the problems quickly. In addition, user interface optimizations, such as confirmation prompts for deletion operations and intuitive filtering and sorting options, significantly enhanced the user experience.

Completed Stories:

User story #1

As a user, I would like to be able to delete items from my current inventory shelf for recording the immediate consumption.

#	Description	Estimated Time	Owner
1	Implement delete functionality for items in the existing inventory	3hr	Songyu He
2	Create a UI element (e.g., delete button) for each inventory item	1hr	Songyu He
3	Add a confirmation prompt to prevent accidental deletions	2hr	Songyu He
4	Test the delete function to ensure all items can be removed correctly	2hr	Songyu He

Completed:The feature now enables users to delete items from their current inventory shelf, allowing for accurate recording of immediate consumption. Users can delete an item by clicking the designated delete button next to each inventory entry. A confirmation prompt appears to prevent accidental deletions, ensuring that users can confidently manage their inventory. All deletions are processed seamlessly, maintaining the integrity of the inventory records.

User story #2

As a user, I want to be able to filter ingredients based on different categories to make it easier to manage and view specific ingredient categories in my inventory.

#	Description	Estimated Time	Owner
1	Create filtering UI for ingredient categories	3hr	Songyu He
2	Create a data structure to categorize ingredients	2hr	Songyu He
3	Implement backend functionality to handle filtering requests	2hr	Songyu He
4	Connect the filtering UI with the backend to display results	2hr	Songyu He
5	Test the filtering functionality to ensure it works correctly	2hr	Songyu He

Completed:The feature now allows users to filter ingredients based on different categories, enhancing the management and viewing of specific ingredient types in their inventory. A user-friendly filtering UI has been implemented, enabling users to select from various ingredient categories easily.
A structured data model categorizes the ingredients effectively, and backend functionality has been developed to handle filtering requests seamlessly. The filtering UI is now fully integrated with the backend, displaying the filtered results accurately based on user selections. Comprehensive testing has been conducted to ensure that the filtering functionality works correctly and efficiently, providing users with a smoother inventory management experience.

User story #3

As a user, I would like to be able to filter and sort current inventory by preset or customized “expiration times”.

#	Description	Estimated Time	Owner
1	Create the filtering and sorting UI for expiration times	3hr	Songyu He
2	Implement functionality to support filtering by expiration dates	2hr	Songyu He
3	Implement functionality to support sorting by expiration dates	2hr	Songyu He
4	Connect the filtering and sorting UI with the backend to display results	2hr	Songyu He
5	Test the filtering and sorting functionality to ensure it works correctly	2hr	Songyu He

Completed:The feature now enables users to filter and sort their current inventory by preset or customized “expiration times,” improving inventory management and reducing waste.
A comprehensive UI for filtering and sorting by expiration times has been developed, allowing users to easily access and manipulate their inventory based on freshness.
Functionality to support filtering by expiration dates has been implemented, alongside sorting capabilities to arrange items by their expiration times.
The filtering and sorting UI is fully integrated with the backend, ensuring that results are displayed accurately and promptly based on user selections. Rigorous testing has confirmed that both the filtering and sorting functionalities work correctly, providing users with an effective tool for managing their inventory more efficiently.

User story #4

I would like to synchronize ingredient expiration information with my Apple Calendar so that I can receive notification on my device.

#	Description	Estimated Time	Owner
1	Set up permissions for Apple Calendar and configure EventKit. Enable permissions in Xcode and add required configurations	2hr	Xiuwen Fang
2	Request calendar permissions and handle user authorization, write permission request code and handle different permission scenarios (grant/deny).	3hr	Xiuwen Fang
3	Implement communication with iOS's EventKit and synchronize the food and expiration time data returned by the backend to Apple Calendar.	4hr	Xiuwen fang
4	The UI that asks the user whether they want to sync to the calendar when add food	1hr	Xiuwen Fang
5	Tests need to ensure Flutter communicates with the backend, and that platform channels are correctly integrated with iOS.	2hr	Xiuwen Fang

Completed: This feature now allows users to synchronize ingredient expiration information to the Apple Calendar to receive timely expiration reminders. We successfully configured the permissions for Apple Calendar, including configuring EventKit and handling multiple permission scenarios, to provide a smooth user experience. By integrating with EventKit for iOS, the expiration information provided by the backend was synchronized. When adding food, the user interface asks the user if they want to synchronize the expiration information to the calendar, which enhances the interactive experience. After thorough testing, it was confirmed that Flutter was able to communicate effectively with the backend and that the platform channel was correctly integrated with iOS to ensure reliable synchronization.

User story #5

I would like the system to suggest appropriate saving methods and saving times when adding or modifying ingredient information.

#	Description	Estimated Time	Owner
1	Create a database or configuration file containing the storage rules for common ingredients. Each ingredient corresponds to one or more storage methods (such as refrigeration, freezing) and the corresponding storage time.	1hr	Xinyi Wu
2	Create an API that returns matching food information based on the food name entered when the user searches.	2hr	Xiuwen Fang

3	a search button is provided to allow the user to search the database for matching foods.	1hr	Xiuwen fang
4	Automatically fill the food information returned by the backend into the input form, allowing users to see preset saving suggestions and information.	2hr	Xiuwen Fang
4	Logic for handling user input and calling the API to get suggestions	2hr	Xiuwen Fang
5	Test different search scenarios to ensure the system can correctly match food items, autocomplete, and handle user modifications. Test data saving and error handling.	2hr	Xiuwen Fang

Completed: This feature now allows users to get recommended optimal storage methods and preservation times when adding or modifying ingredient information, improving the accuracy of ingredient management. The database stores storage rules for common ingredients, each of which corresponds to one or more storage methods (e.g., refrigerated, frozen) and corresponding preservation times. Users can query the database using the search function, which automatically fills in relevant storage information into the input form. the API obtains and displays storage suggestions, logically processes user inputs and provides personalized recommendations. After testing, the system is able to accurately match foods, automatically complete inputs and effectively manage user changes, improving user experience and ensuring data processing stability.

User story #6

I would like to have an in-app list function to track what I am going to buy soon.

#	Description	Estimated Time	Owner
1	Create a table(database), such as shopping_list, with fields such as user ID, product name, quantity, date added, etc.	1hr	Xinyi Wu
2	Create a “Shopping List” page or section that includes an input box for adding products, a product list display area, and buttons for editing and deleting functions. And add a “Shopping List” button to the navigation bar.	3hr	Xiuwen Fang
3	Create APIs for users' shopping lists to support adding, updating, deleting, and querying items.	2hr	Xiuwen fang

4	link frontend and backend	2hr	Xiuwen Fang
5	Fully test and debug the shopping list functionality to ensure that the add, modify, delete and synchronize functions are working properly.	2hr	Xiuwen Fang

Completed: The shopping list feature provides a convenient in-app solution to help users keep track of planned purchases. A dedicated shopping list table was created in the database to record each user's list items, containing fields such as product name, quantity, date added, etc. The user interface provides a “Shopping List” section that allows users to add, edit and delete items. The user interface provides a “Shopping List” section that allows users to add, edit and delete items, and a “Shopping List” button in the navigation bar for access. The back-end API supports full add, delete and retrieve operations for shopping lists, ensuring that users can easily manage their lists. Full testing verified the reliability of the add, modify, delete and synchronize functions, ensuring excellent performance and a smooth user experience.

User story #7

I would like to receive an in-app notice of the expiration information.

#	Description	Estimated Time	Owner
1	Create a notice UI	2hr	Xinyi Guan
2	Write an in-app notification API to remind users that ingredients are about to expire.	3hr	Xinyi Guan
3	Connect frontend and backend	2hr	Xinyi Guan
4	Test/debug the expiration information notification function to ensure that it works	2hr	Xinyi Guan

Completed: Users can see notification in our application not only for ingredients that are about to expire(<=3 days including today), but also can see it for ingredients that has already expired. If user do nothing with the notification (i.e., click “Okay, I know now”, or just log out and re-log), the notification will show up again, and if the user click “Cleaned up already”, the notification will not show up again and these “cleaned up” ingredients will not in the ingredients page anymore.

User story #8

I want to be able to create my recipes via text, image or video links and keep accurate records.

#	Description	Estimated Time	Owner
1	Create a create recipe UI	2hr	Xinyi Guan

2	Create all recipes display UI	2hr	Xinyi Guan
3	Create API to allow users upload their recipes by text, image or video links and keep them.	3hr	Xinyi Guan
4	Connect frontend and backend	2hr	Xinyi Guan
5	Creating needed table and data structure changes in the database	1hr	Xinyi Wu
6	Test/debug to make sure this function is working	2hr	Xinyi Guan

Completed: The feature now allows users to create and upload their recipes via text, image, or video links, which are stored and displayed on the "My Recipe" page. Users can add their recipes by clicking the plus sign on the page as long as they fill in the Recipe name and ingredients fields(image, video link and other fields are optional), ensuring that their records are accurately saved.

User story #9

As a user, I would like to be able to edit and delete recipes that I have previously created.

#	Description	Estimated Time	Owner
1	Create a edit/delete recipes UI	2hr	Xinyi Guan
2	Create API to support users to edit/delete created recipe contents.	4hr	Xinyi Guan
3	Connect frontend and backend	2hr	Xinyi Guan
4	Store information into the database.	1hr	Xinyi Guan
5	Test/debug to make sure this function is working	2hr	Xinyi Guan

Completed: Users are now able to edit and delete their previously created recipes directly from the "My Recipe" page. All modifications are successfully updated in the database, and testing confirms that the edit and delete functionalities work as expected.

User story #10

As a user, I want to filter the current recipes with labels like ‘Breakfast’, ‘Vegetarian’, etc.

#	Description	Estimated Time	Owner

1	UI: Create adding labels function when add and editing recipes	1hr	Xinyi Wu
2	UI: Recipe filtering designs	2hrs	Xinyi Wu
3	Backend: Database add and modification with updates	1hr	Xinyi Wu
4	Backend: Implement filtering logic	1hr	Xinyi Wu
5	Connection: Front-end to display the filtered recipes.	1hr	Xinyi Wu
6	Test the filtering functionality to ensure it works correctly	1hr	Xinyi Wu

Completed: Users can now efficiently filter recipes by various labels, such as 'Breakfast' and 'Vegetarian', making it easier to find recipes suited to specific needs or preferences. The front-end UI allows users to add and edit these labels while creating or modifying recipes. A filtering interface was designed to display recipes according to selected labels. The backend was enhanced to support label additions and modifications and implemented the filtering logic. Finally, tests confirmed smooth interaction between the front-end and backend, with all label-based filtering functionality working as expected.

User story #11

As a user, I would like to see a calculated nutritional report after selecting specific dish recipes.

#	Description	Estimated Time	Owner
1	UI: Select and unselect recipes, calculation button, Select Status Alert	2hr	Xinyi Wu
2	UI: Illegal handling (e.g. user clicks on “Calculate” without selecting any recipes)	1hr	Xinyi Wu
3	UI: Pop-up nutritional report page	1hr	Xinyi Wu
4	Backend: Extract the information of the selected recipes from the database and Implementing Recipe Calculation Logic	2hr	Xinyi Wu
5	Connection: frontend and backend	1hr	Xinyi Wu
6	Test the report functionality to ensure it works and handles illegal situations correctly	1hr	Xinyi Wu

Completed: After selecting specific recipes, users can view a nutritional report that provides an analysis of their chosen dishes. The UI allows users to select multiple recipes, and a “Calculate” button initiates the report. Handling for cases where no recipe is selected ensures a smooth experience. Once triggered, a nutritional report is displayed in a pop-up.

The backend retrieves relevant data for selected recipes, performing calculations to generate the report. Front-end and backend connections were verified to ensure accuracy, and testing validated both the calculation logic and proper error handling.

User story #12

As a user, I would like to add new items into my inventory by selecting preset ingredient options for quick updates.

#	Description	Estimated Time	Owner
1	Change the UI for the "Add Inventory" feature, including a search box for the user to input keywords related to ingredients.	2hrs	Xinyi Wu
2	Develop the backend logic to query the database for the most relevant ingredients based on the user's input and display the results in a dropdown list.	3hrs	Xinyi Wu
3	Create or modify a table in the database containing preset ingredients, categories, and other related information to be used for quick adding.	2hrs	Xinyi Wu
4	Develop the feature that automatically fills in the ingredient's preset information when the user selects an item from the dropdown list.	2hrs	Xinyi Wu
5	Ensure that the preset data is editable by the user once the item is loaded into the food addition page, allowing further customization before saving.	1hr	Xinyi Wu
6	Provide feedback messages after the item is successfully added as a preset ingredient, confirming the addition to the inventory.	1hr	Xinyi Wu
7	Test and debug the entire workflow, ensuring the search, preset data loading, and item addition processes work smoothly.	2hrs	Xinyi Wu

Completed: Users can now quickly add items to their inventory by selecting from a preset list of ingredients, streamlining the inventory management process. The updated “Add Inventory” UI includes a search box for finding ingredients. Backend functionality returns relevant ingredients based on input, and selecting an item fills in preset details automatically. The database was updated with preset ingredients and categories for fast selection. Users can edit details before saving, with feedback confirming successful addition. Comprehensive testing ensured the smooth functioning of search, preset loading, and inventory addition workflows.

What Did Not Go Well?

We successfully completed all the user stories outlined in our Sprint 2 document; however, we encountered a number of challenges along the way that could be addressed and improved in future sprints.

1. **Insufficient Communication on User Story Redundancy and Cohesion**
   Throughout the sprint, there was a lack of in-depth communication about the overlap and cohesion among user stories. This resulted in some redundancy, with similar functionalities being addressed in multiple user stories without clear distinctions. This made it challenging to streamline the implementation and could have affected efficiency by leading to duplicate work.

2. **Code Confusion Due to Git Upload Conflicts**
   Since team members were not fully familiar with each other's modules, there were several instances of overlapping Git uploads. This led to repeated code uploads and occasional conflicts in the functionality. These issues introduced additional time in resolving merge conflicts and restoring intended functionalities, impacting our development pace.

3. **Usability and Functionality**
   In planning the application, we aimed to create a more usable and realistic interface, incorporating advanced UI features. However, we underestimated the challenges of implementing these complex UIs within the constraints of our platform (Flutter) and our current UI development skills. This miscalculation resulted in delays, as we spent considerable time modifying and simplifying our designs to fit within our technical capabilities and platform limitations. This led to unanticipated rework and impacted our timeline for coding and testing.

How Should We Improve?

1. **Enhance Communication on User Story Overlap and Cohesion**
   To avoid redundancy, schedule regular discussions at the beginning of each sprint to review user stories in depth. During these sessions, identify any overlap between tasks and clarify distinctions to ensure each team member has a clear understanding of their responsibilities. A shared document mapping out dependencies and overlaps between user stories could also help keep the team aligned.

2. **Implement Better Git and Code Review Practices**
   Establish clear guidelines for Git usage, such as regularly pulling updates from the main branch before committing and pushing changes. Encourage smaller, more frequent commits with descriptive messages to help others understand recent changes. Additionally, introduce code reviews where team members check each other's work to ensure integration issues are caught early and everyone remains familiar with each other's modules.

3. **Pre-search and learning on UI**
   To better manage complex UI requirements, initiate realistic UI planning sessions early in each sprint to assess the feasibility of proposed designs given the team's skill level and platform constraints. Break down advanced UI features into smaller, achievable tasks, and prioritize essential functionality over aesthetics when necessary. Encourage collaboration with designers or more experienced team members to find alternatives that achieve a similar user experience with simpler implementation first than listing the hard tasks. If time is not available, always think of functionality higher than the usability and aesthetics.