Group 5 Specialisation Project

CFG Autumn 2023

Smart Pantry

Group Members

- Amy Dangerfield
- Anna Ketre
- Dorothy McEvoy
- Lauren Aldridge
- Lauren Southall
- Vanessa Latchem-Smith
- Karen Lacey

Introduction:

Project Title: Smart Pantry

We developed a smart inventory system for the kitchen, designed to streamline the process of managing groceries. Our goal is to eliminate the hassle of manually assessing fridge, freezer, and pantry contents while also providing instant recipe suggestions based on available ingredients. Additionally, our system generates shopping lists on demand, saving users time and trouble.

Objective:

We have developed a smart system for kitchen management. Users can easily store and manage their inventory, receive recipe suggestions based on available ingredients, and have stock levels automatically updated. The system prompts users to add items to a shopping list when supplies are low, and it generates a shopping list on demand.

Background:

In the home kitchens of busy households, people often struggle with keeping track of their kitchen stock leading to issues such as last minute shopping trips for missing ingredients, or unfortunately, wasted food when ingredients are left unused. Our Smart Pantry Project addresses and aims to solve these problems for our users by creating a solution to assist people in effortlessly managing their kitchen inventory.

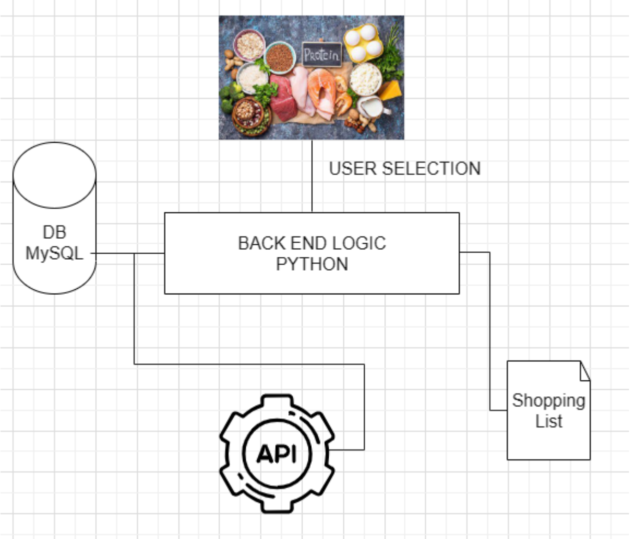
The Smart Pantry provides key features such as stock level updates, personalised recipe suggestions based on currently available ingredients and automated shopping list generation. This system eliminates the need for our users to manually track their kitchen stock, or attempt to remember what ingredients they have. It removes the need to write shopping lists and helps our users to make decisions around meal planning by providing recipes. The Smart Pantry aims to streamline and automate the process of managing the kitchen inventory leading to a more enjoyable and efficient cooking and shopping routine.

Steps:

Specifications and Design

The system will have a user interface that allows the user to interact with the system. The user interface will provide the following key features:

- Inventory management: The user can view, add, edit, and delete items in their kitchen inventory. The system will store the name, quantity, unit, and expiration date of each item in a database.
- Recipe generation: The user can request recipes based on the current available ingredients in their inventory. The system will use a recipe API to fetch recipes that match the user's preferences and dietary restrictions. The system will also show the user the required ingredients, steps, and nutritional information for each recipe.
- Stock level update: The system will automatically update the stock levels of the items in the inventory when the user uses or adds items. The system will also send notifications to the user when the stock level of an item is low or expired.
- Shopping list generation: The user can request a shopping list based on the items that are low or out of stock in their inventory. The system will generate a shopping list that contains the items that the user needs to buy. The user can also add or remove items from the shopping list manually.
- In a real world scenario, the system will have a cloud component that consists of a server and a storage. In this version, the user will host the webpage on localhost and hold their database in their own SQL.



Implementation & Execution

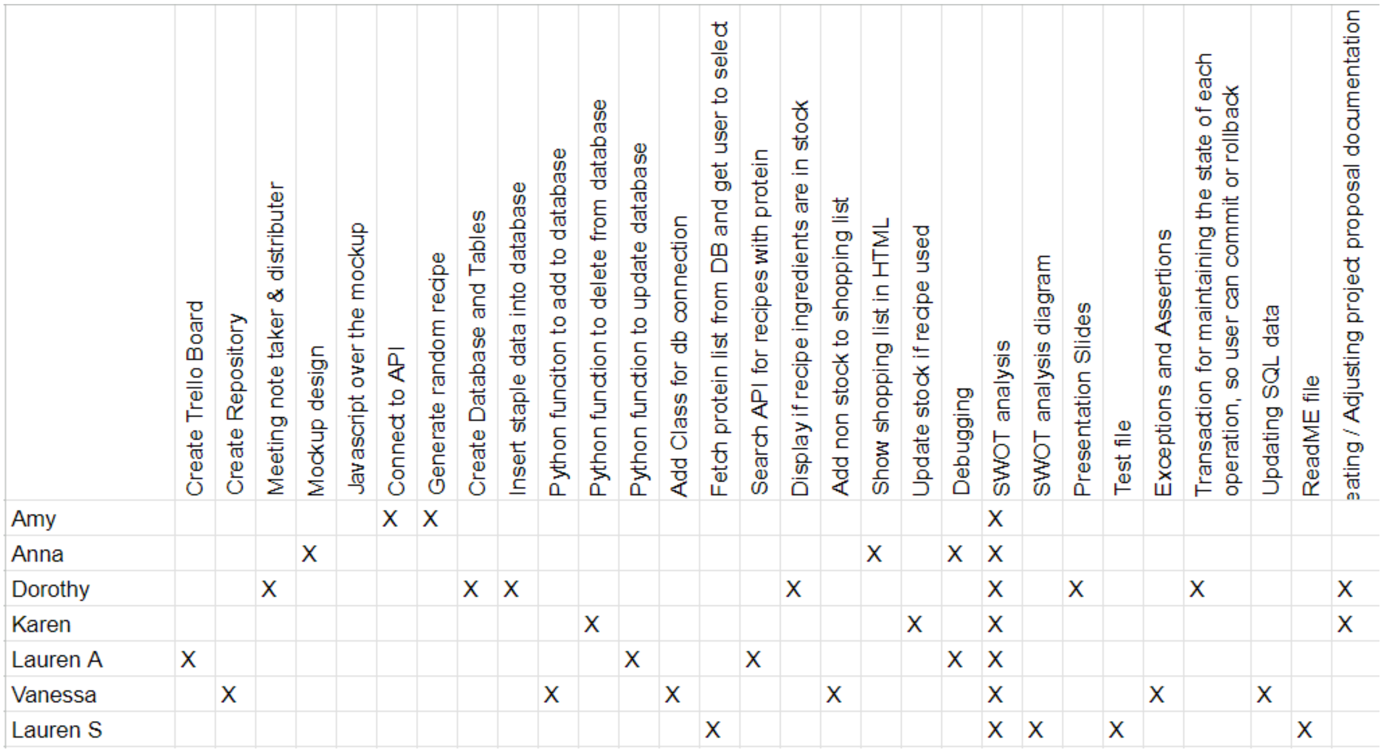
We chose to work with Agile methodology due to the following reasons:

- Allows for greater adaptability during the project
- It is an iterative & incremental development process which improves time frames
- Provides the space for continuous feedback
- Collaborative team environment
- Better transparency
- We are able to manage risks better

We used:

- Javascript, HML and CSS for the front end.
- Python for functionality, utilising Flask.
- mySQL for the database.

Below is a tracker showing what roles and tasks each member conducted during the project:



	Create Trello Board	Create Repository	Meeting note taker & distributer	Mockup design	Javascript over the mockup	Connect to API	Generate random recipe	Create Database and Tables	Insert staple data into database	Python funciton to add to database	Python function to delete from database	Python function to update database	Add Class for db connection	Fetch protein list from DB and get user to select	Search API for recipes with protein	Display if recipe ingredients are in stock	Add non stock to shopping list	Show shopping list in HTML	Update stock if recipe used	Debugging	SWOT analysis	SWOT analysis diagram	Presentation Slides	Test file	Exceptions and Assertions	Transaction for maintaining the state of each operation, so user can commit or rollback	Updating SQL data	ReadME file	eating / Adjusting project proposal documentation
Amy						X	X														X								
Anna				X														X		X	X								
Dorothy			X					X	X							X					X		X			X			X
Karen											X								X		X								X
Lauren A	X											X			X					X	X								
Vanessa		X								X			X				X				X				X		X		
Lauren S														X							X	X		X				X	

Database Setup (SQL):

Designed a relational database schema in SQL for inventory management, featuring three stock tables (Fridge, Freezer, and Pantry) with consistent columns:

- ➔ Ingredient Name
- ➔ Type of Ingredient
- ➔ Quantity
- ➔ Unit of Measurement
- ➔ Minimum Quantity Needed
- ➔ Sell By Date

Explored the idea of a shopping list table triggered by low stock levels.

Evaluated the necessity of establishing relationships between tables.

Implemented comprehensive CRUD functionality.

Developed views to display diverse food groups.

API Integration (Edamam):

Obtain API key from Edamam. Amy generated the initial code for this and documentation on how to implement it.

Wrote Python code to make API calls for recipe retrieval - initially based on a user selecting a protein from available stock. Executed join queries on tables as needed.

App Development:

Use Python for development, utilising Flask.

Implemented functionality to add, update, and delete items in the inventory.

Integrated recipe generation based on user inventory.

Automated inventory updates after selecting a recipe.

Automated inventory updates after user adds shopping

Prompt users when ingredients are low and add items to the shopping list

Source Code

We ensured our python components were modular.

Included clear comments where necessary for documentation.

Used git/gitHub for version control.

Included a Utills file.

Test files (unit tests for all functions).

Main file with a run function to execute the program.

Documentation/Readme

Wrote a comprehensive project specification document using Readme that provides clear instructions on how to run the project, what libraries are required and how to install them etc.

Packages to install for the project:

- pprintpp
- requests
- Flask
- json

Testing & Evaluation

Implemented thorough testing for each Python function to ensure functionality.

Conducted thorough unit testing.

Collaboration

Tools:

Regular Zoom meetings for updates/ discussions (weekly on a Sunday).

Slack for real-time messaging/communication.

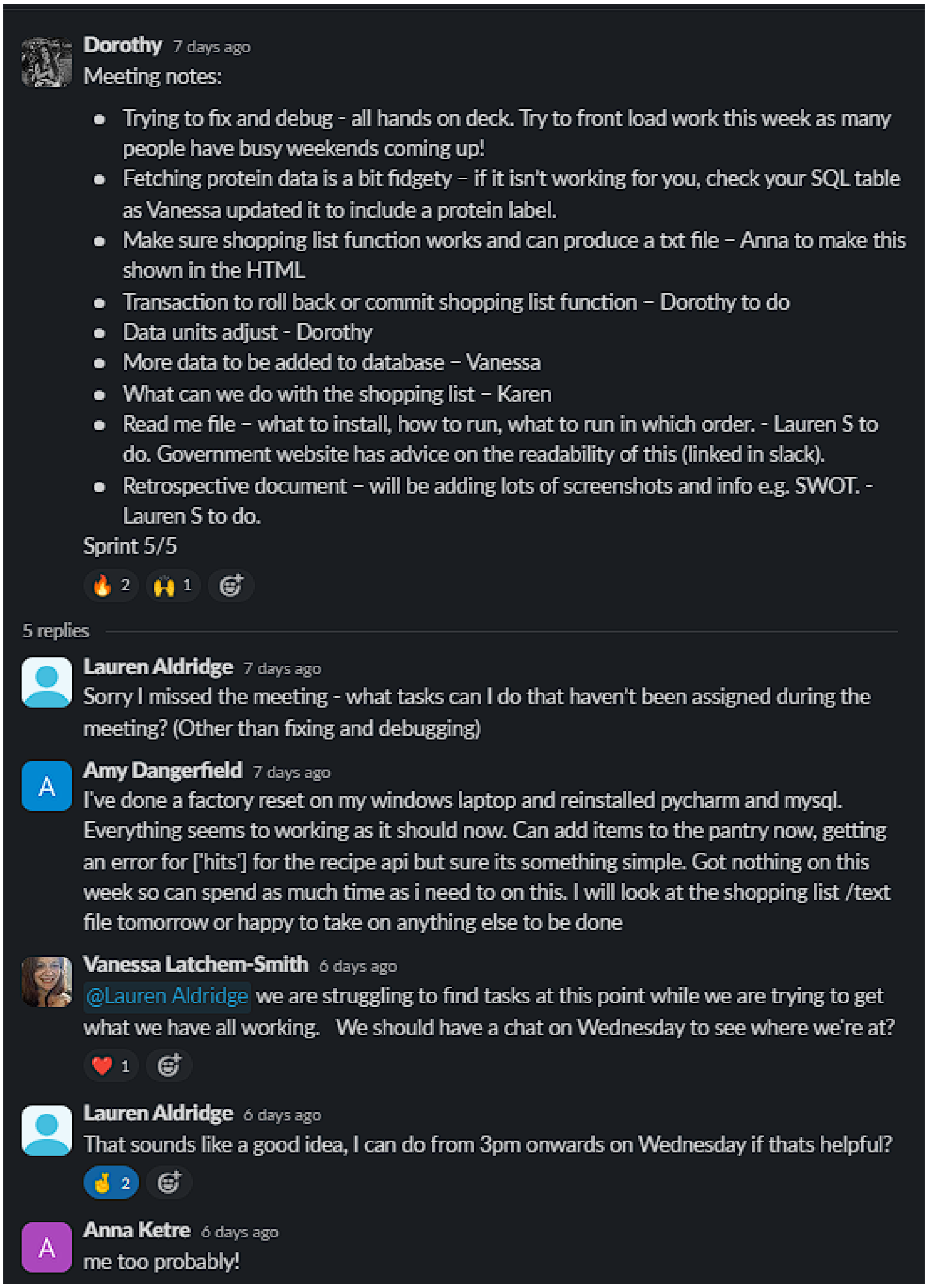
GitHub for source control.

Trello for task/project management.

Activity log for transparency.

Google Docs for collaboration on Project Document.

Below is a screenshot of our slack chat showcasing meeting notes distributed to the team, team members collaborating and acknowledging each other's alternative commitments empathetically.



Dorothy 7 days ago

Meeting notes:

- Trying to fix and debug - all hands on deck. Try to front load work this week as many people have busy weekends coming up!
- Fetching protein data is a bit fidgety – if it isn't working for you, check your SQL table as Vanessa updated it to include a protein label.
- Make sure shopping list function works and can produce a txt file – Anna to make this shown in the HTML
- Transaction to roll back or commit shopping list function – Dorothy to do
- Data units adjust - Dorothy
- More data to be added to database – Vanessa
- What can we do with the shopping list – Karen
- Read me file – what to install, how to run, what to run in which order. - Lauren S to do. Government website has advice on the readability of this (linked in slack).
- Retrospective document – will be adding lots of screenshots and info e.g. SWOT. - Lauren S to do.

Sprint 5/5

🔥 2 🙌 1

5 replies

Lauren Aldridge 7 days ago

Sorry I missed the meeting - what tasks can I do that haven't been assigned during the meeting? (Other than fixing and debugging)

Amy Dangerfield 7 days ago

I've done a factory reset on my windows laptop and reinstalled pycharm and mysql. Everything seems to working as it should now. Can add items to the pantry now, getting an error for ['hits'] for the recipe api but sure its something simple. Got nothing on this week so can spend as much time as i need to on this. I will look at the shopping list /text file tomorrow or happy to take on anything else to be done

Vanessa Latchem-Smith 6 days ago

@Lauren Aldridge we are struggling to find tasks at this point while we are trying to get what we have all working. We should have a chat on Wednesday to see where we're at?

❤️ 1

Lauren Aldridge 6 days ago

That sounds like a good idea, I can do from 3pm onwards on Wednesday if thats helpful?

🤞 2

Anna Ketre 6 days ago

me too probably!

Team Approach to Project Work:

Task Assignment

Tasks were assigned on a weekly basis during regular group Zoom meetings. The group discussed and agreed to follow an Agile method, breaking our project down into small manageable tasks at each sprint. Our zoom meetings were structured to begin by discussing progress made on the previous sprint, then deciding our next steps/tasks for the next sprint. Tasks were allocated by skillset or by team members volunteering to take on specific tasks, with consideration given to each team member's availability for the upcoming sprint. This method ensured we were making efficient progress and the workload was distributed fairly. Our team values adaptability and flexibility in our approach and we aimed to consider feedback from all team members and remain open to adjusting strategies or functionality as issues emerge or if improvements are required in the project.

Task Tracking

Aside from our regular zoom meetings, the group kept in regular contact via Slack to update each other with progress on our individual tasks for each sprint.

Moreover, Lauren A set up a project Trello board accessible to all group members. This board served as a centralised platform for assignment tasks, tracking progress, and updating tasks statuses upon completion.

Each team member is also keeping a record of their own tasks by recording time spent working on the project on the group Activity Log.

Version Control

We managed our code changes using Git/Github. Each team member worked on their own branch. We used the following naming structure for our branches which was suggested by Dorothy to keep code changes and branches organised and easy to understand: 'SprintNumber - TeamMember -Task/Function' . For example, Dorothy's first task was added to the repository with branch name: 'Sprint1-Dorothy-Database'

All code was added to the remote repository using team member's individual branches and was merged to the main branch by creating pull requests. When we created pull requests we made sure to request that our code was reviewed and merged by other team members which ensured conflicts were managed, code was functioning and it allowed us to provide feedback to each other for any improvements.

Testing Strategy

Our team tested our system by ensuring all functionality was working. The code was tested by writing unit tests for our functions as we progressed through the project. We made sure to discuss testing of code regularly at our Zoom meetings and ensured the tests were written and conducted throughout the project development lifecycle.

Finally:

Once our project is complete we will:

- ➔ Create our presentation
- ➔ Submit project
- ➔ Present as a team to our fellow CFG students and others
- ➔ Celebrate! 🥳