

1. KitchenSync: A Pantry and Recipe Companion
2. Team Members
 - a. David Tran
 - i. dtran2021@my.fit.edu
 - b. Chris Nederhoed
 - i. cnederhoed2020@my.fit.edu
 - c. Tyler Son
 - i. tson2022@my.fit.edu
3. Advisor - Fitzroy Nembhard
 - a. fnembhard@fit.edu
4. Client - Fitzroy Nembhard
 - a. Affiliation - College professor in the Department of computer science
5. Progress of current milestone

Task	Completion %	Tyler Son	Chris Nederhoed	David Tran	To do
Inventory Management System	90%	20%	70%	30%	Improve Accuracy Of Receipt Scanner
Recipe Organizing System	80%	20%	50%	20%	- Recipe Searching with filters - Review System - Nutrition Breakdown
Basic GUI	90%	0%	100%	0%	- Notification Center - User Profile - User Settings
Cloud DB	95%	100%	0%	0%	- Seeding initial recipes - Connect To Community Recipe Page
<i>Meal Planner</i>	60%	0%	100%	0%	- Add Meal Form - Nutritional Break Down
Admin Backend	20%	0%	0%	100%	- User Accounts - Backend Manager

6. Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone:
 - a. Inventory Management System
 - i. The inventory system is almost complete with a few components

still being worked on such as the new item upload pipeline. This pipeline is where a user is presented with three options to add an item into their inventory such as manual, barcode, or receipt. All three options will lead to the manual option in the end as a user will still need to review an item's details before adding them. In the barcode upload a user adds an image(s) of item barcodes the system then looks for a product's details based on the UPC found and then shows the form with prefilled information and the user then adds in the missing parts. It will then loop to the next item to be added until all items have been added. The receipt uploader is going to act very similar the only difference is a single image of a receipt is uploaded and then it is parsed for UPC's and then checked to see which are food, it should be noted that a item might not be parsed correctly in which case it will let the user know and then the user can add that item manually. We also have to finish connecting this to the local users DB for persistence.

b. Recipe Organizing System

- i. The recipe organizing system is nearly complete as well. The remaining parts here are getting the filter/sort by systems working. This should only take a day or so to add but due to time constraints it was delayed. We also need to add the nutrition breakdown script which will calculate the values for a recipe like how many calories, protein, carbs, and fats. We also need to connect image persistence to our recipe cards.

c. New GUI Elements (Basic GUI)

- i. This is at about 90% complete. We finished adding the meal planner and community pages to the UI. We also added item and recipe widgets and several pops throughout. Some of the things left are our Notification center, user profile, and user settings. Until we add the user accounts these are on the back burner. Part of our stretch plan is to add Tooltips / a on boarding tutorial to help users get familiar with the system and its tools.

d. Cloud DB

- i. The Cloud DB in AWS is set up and just needs to be seeded with recipes a user can save locally. The Cloud DB is connected to the community recipe page. The recipes can be saved onto the Cloud DB. This function now needs to be included for ingredients. In addition, user account creation needs to be able to form users that have their own keys to ensure security.

e. *Meal Planner*

- i. This module is one of the primary focuses of this milestone and is about halfway through completion. We set up the GUI components and the needed UI elements. The remaining components to be completed are adding time blocks that can be moved around similar to Google calendar. Here is another spot where we want to add the nutritional breakdown. We also need to add in a meal form which

will show the user meals they can make now based on their inventory and then all their other meals. From here meals that get planned will have their ingredients added to their shopping list if the user doesn't have that ingredient or enough of it.

- f. Admin Backend
 - i. This module saves ingredients and recipes that the user inputted into a JSON file, which should be used to store data into the database, so later data can be retrieved from the database to be stored in the JSON file and then placed into the app itself. Currently it only saves the ingredient inventory and not anything else.
- g. Shopping Companion
 - i. This module was not worked on this time around due to time constraints. Our plan for it is to have price data from websites for ingredients compared and then show users where they can get the best deals. Filters and other criteria would be used to make a shopping list that factors in distance, store preferences, among other things. This would then create a set of shopping lists to better inform a user where they can save money while cooking what they love.
- 7. Discussion (at least a few sentences, ie a paragraph) of the contribution of each team member to the current Milestone:
 - a. Tyler Son: I worked on the backend components we needed for this milestone. I also helped test the completed portions of the frontend components.
 - b. David Tran: I worked on saving the ingredients and recipes the user imputed as a JSON file, which will serve as the app's local storage, and allow for transferring the data to the database.
 - c. Chris Nederhoed: I primarily worked on more frontend components we needed for this milestone. I also worked on the meal planner, ingredient and recipe widgets, and necessary python scripts. I also worked on UI persistence along with other data persistence.

8. Plan for the next Milestone (task matrix)

Task	Tyler Son	David Tran	Chris Nederhoed
User Accounts		Create user account creation process, and link	

		user accounts to AWS DB	
User Sub Components (Inbox, Settings, Profiles)	Allows users to customize their settings such as "light mode" vs "dark mode" and users to include profile pictures.		
Frontend Facelift			Polish front end UI/UX making things look more professional while adhering to good principles. Implement window scaling.
Accepted Receipt Expansion			Use machine learning to parse hard to read user images and get the information from them
Shopping Companion	Create a list creation algorithm based on a user shopping list and preferences.	Get item prices from websites and/or api's.	
Feedback/Review System	Create attributes that store user feedback for recipes.		

9. Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone or "Lessons Learned" if this is for Milestone 6
 - a. User accounts will be handled through AWS so that way they can access the cloud DB. We also will be able to track users this way and manage their content once we are ready to allow user uploads to the cloud. The overall plan is to gather minimal data from users such as just their name and email address. If we find we need more data like a phone number then we can add this as a field we need and then will have to add a security plan to manage this data.
 - b. User Sub components include all the views and ways a user can further interact with the application in ways that are user specific. These are things such as an inbox for notifications, settings like light or dark mode, and then profiles to provide any stats a user might want to see or change. We will add more settings to let the user customize their experience as we think of them/provide support for them.
 - c. First semester left us with a fairly stable front end but it leaves a lot to be desired from it. Our goal this time around is to create a style sheet to make extending functions and adding new functionality have some standard to adhere to but also make the application more professional. We also want to improve the UI/UX of the application so that there are little quality of life features that help make the application better.
 - d. Shopping companion is going to gather information from websites that provide us with the cost for an ingredient and then allow a user to create a shopping list and see where they need to get them. We have two core parts to this: 1) get the prices, 2) create lists based on a users shopping list and their preferences. This will let users be able to take into account the user's store preferences and distance to the store making the lists more reasonable.
 - e. The feedback and review system is part of the community recipes page where users can give feedback and/or review a recipe another user has posted or that admins have posted. This is so that other users can see if the recipe is good as is or needs changes such as shorter cook time or less salt. This will be another table we store in AWS and link the recipes table using the recipe ID to reference the two table together through this foreign key.
10. Date(s) of meeting(s) with Client during the current milestone: 11/20/2024
11. Client feedback on the current milestone
 - a. ... (if Client and Faculty Advisor are the same, write "see Faculty Advisor Feedback below")

b. ...

12. Date(s) of meeting(s) with Faculty Advisor during the current milestone: 11/20/24

13. Faculty Advisor feedback on each task for the current Milestone

a. Task 1: Machine learning can be used to help with scanning the receipts from other stores.

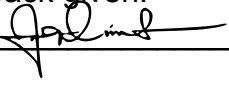
b. Task 2: No major feedback given.

c. Task 3: The coloring scheme for the application can be improved. For example, a different shade of gray should be used.

d. Task 4: It would be better if all users had their own public keys. A service account should be used that handles all the user accounts that only admins have access to.

e. Task 5: No major feedback given.

f. Task 6: No major feedback given.

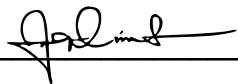
14. Faculty Advisor Signature:  Date: 11-25-2024

----- on a separate page -----

15. Evaluation by Faculty Advisor

- a. Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu
- b. Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Tyler Son	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
David Tran	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Chris Nederhoed	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

- Faculty Advisor Signature:  Date: 11-25-2024