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	70%	20%	50%	30%	• Improve Accuracy Of Receipt Scanner • Widget Creation on item added • Connect to Local DB for storage
Recipe Organizing System	60%	30%	10%	20%	• Recipe Card Creation • Recipe Searching with filters • Recipe Sharing • Review System • Nutrition Breakdown • Connect to Local DB for storage
Basic GUI	80%	0%	100%	0%	• Meal Planner • Community Recipe Dashboard • Notification Center • User Profile • User Settings

					• Misc Popups
Cloud DB	60%	100%	0%	0%	• Seeding initial recipes

6. Discussion (at least a few sentences, ie a paragraph) of each accomplished task (and obstacles) for the current Milestone:
    a. Inventory System
        i. Setting up an inventory management system was fairly easy however as part of this milestone we also had different ways a user can add things to this inventory. Those ways were manual, barcode, or receipt. Manual and barcode entry were not too complicated but the receipt scanner had its challenges which prevented it from being usable in this milestone. Image accuracy and the data on a store’s receipt have proved to be the two biggest challenges. Stores like Walmart and Target both have an item's upc code on the receipt making it very easy to get the product details but stores like Publix only have product name and price making it harder to get the products information. As for image accuracy under a perfectly black and white image that is scanned we can get about 90% of the data from a receipt but with a user uploaded image that falls to around 60% and lower. This is primarily due to the lighting, age of the receipt, background, and quality of the image. To mitigate this we can add post processing to an image and try to prompt the user to take the best image they can in order for Tesseract to extract more of the data and get our accuracy up to 80% for user uploaded images.
    b. Recipe Organizer
        i. Similar to the inventory management system we were able to modify a few of our GUI elements to work for user entry of their own recipes. We currently have the frontend mostly done for this component but still have to finish setting up the local db for the user to be able to store their recipes. Because this is currently a desktop application we have to figure out a way to share a recipe with both another user and possibly themselves like sending it to their phone.
    c. Basic GUI
        i. Building the frontend proved to be a bit of a challenge. We chose Javafx as it seems to be a good build once run anywhere, although this will be put to the test later if we choose to migrate the platform

to mobile in semester 2. Getting the frontend build was fairly simple as we were able to use a scene builder tool to quickly create and export GUI elements created from our wireframes. However once put everything together in a java package it was not straightforward for sharing with teammates. We had to go back and rebuild the project file structure using Maven to be able to export it and share it with the team. In addition there were issues with scaling, which made selecting certain options in the menu impossible. The current screens in the GUI are based on the modules we were implementing this milestone and as we go forward more screens will be added.

d. Cloud DB

   i. This module is for the user to be able to share their recipes with others so that we can foster a community of users. This DB is broken down into two parts: AWS Dynamodb and AWS S3. We choose these so that we can store our dynamic data like the recipes themselves, and later all the user’s data like their inventory and recipes not shared with others, and then the static images of the recipes that users add. We had to do this approach due to image file sizes so that we could support about 1000 concurrent users with several dozen recipes each uploaded plus the baseline seeding we will do.

7. Discussion (at least a few sentences, ie a paragraph) of contribution of each team member to the current Milestone:

   a. Tyler Son: I focused on the backend components, specifically setting up AWS. I also set up some of the java scripts for adding a recipe into the database.

   b. David Tran: I worked on implementing the functions for adding ingredients to the shopping list menu. I also worked on adding lists within the controllers to store the inputted data.

   c. Chris Nederhoed: I primarily worked on the frontend components the user will interact with. I also constructed the python scripts for our barcode and receipt scanners along with their integration with the frontend.

8. Plan for the next Milestone (task matrix)

Task	Tyler Son	David Tran	Chris Nederhoed
Finish Implementing Recipe Organizer	Continue work on the recipe database.		
Meal Planner			Building the meal planner with visual comps
New GUI Elements			Any GUI elements needed
Admin Backend		Connect the front end and back end functions.	
Shopping Companion	Find the prices of a product from various stores.	Compare all the prices against each other.	Build recommendation system based on prices.

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. Finish Implementing Recipe Organizer
      i. Our goal for this is to allow the user to create recipes, share recipes, organize them based on tags, see the nutritional breakdown of a recipe per serving. The biggest challenge here as mentioned above is the recipe sharing. Between users should be fine as we will have the cloud db for this. Getting all the components in place and working will mainly be a learning curve.
   b. *Meal Planner*
      i. *Our goal on this one is to let a user have the ability to have a calendar that they can add meals to and see when and what they are eating. This is to help them plan the prep phases and shopping*

        that needs to happen. We also need to make sure that ingredients get removed as the user creates different recipes. We also need to have shopping lists created based on the recipes planned for the selected window with a max of two weeks.

    c. New GUI Elements

        i. This one is just adding new GUI elements to the system as we add features that need them. The modules above will each need some screens made but also smaller popups and widgets that we will make as we need them as we might have not thought of a button while making the wireframes initially.

    d. Admin Backend

        i. The admin backend is really just a way for (us the admins) to query the db for things and filter anything a user might post to the cloud like recipes, feedback, or a review for invalid criteria. We also want admins to be able to have a few tools at their disposal such as recipe seeding for the db, audit logs, and speedy review process for user flagged content.

    e. Shopping Companion

        i. This module is likely going to be the last to be worked on as it is not necessarily quintessential in our project. If we have time this will be at least partly, but hopefully fully, added. This module's goal is to better help a user in understanding how much they are spending on their meals by providing an estimate on the serving cost. We will also try to have the prices on a receipt a user uploads reflect the estimate of price per serving.

10. Date(s) of meeting(s) with Client during the current milestone: 10/28/2024
11. Client feedback on the current milestone

    a. ... (if Client and Faculty Advisor are the same, write "see Faculty Advisor Feedback below")

    b. ...

    c. ...

12. Date(s) of meeting(s) with Faculty Advisor during the current milestone: 10/28/24
13. Faculty Advisor feedback on each task for the current Milestone

    a. Task 1: No major feedback given.

    b. Task 2: No major feedback given.

    c. Task 3: It would be better to have only one search bar per menu. An advanced search would work better as a search bar that covers the entire application.

    d. Task 4: No major feedback given.

14. Faculty Advisor Signature: ________________________________ Date: ________

------ 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: ___________