1.Introduction

1.1.Purpose

This document defines the functional, interface, and performance requirements for the KitchenSync system. KitchenSync is a kitchen aid that is designed to help home chefs from families to hobbyists to manage ingredients from different locations in their kitchens, plan recipes, estimate costs based on available ingredients and purchasing options for missing ingredients.

1.2.Scope

KitchenSync aims to make cooking a breeze by allowing for inventory tracking, recipe suggestions, and cost tracking. The system looks to benefit those needing help planning what to eat, where to get the ingredients, and how much it will cost them. The primary users are people who like to cook at home whether they are single or have a family.

2.Functional Requirements

2.1.Inventory tracking

Description

The System should allow users to create lists of ingredients, enter ingredients using barcodes, enter data using receipt scanning, delete ingredients, update ingredients. Each ingredient will have a name, quantity, location (fridge, freezer, pantry, or cabinet), expiration date, nutritional information. Ingredients should have tags such as minimum quantity, toss after x time, use soon.

Required Behaviors

1. Users must be able to enter ingredients manually or by scanning (barcodes or receipt scanning)
2. Users must be able to update ingredient information: quantity, location, expiration date, or tags
3. Users must be able create a list of ingredients
4. Users must be able to delete expired food, or used-up ingredients
5. System should track and auto adjust ingredients based on consumption habits of high use items like milk, eggs, bread and other commonly used items (these can be user dependent).

I/O

- Scanning
    - Expected (Receipt Scan)
        - Input: User scans a receipt with non food items
        - Output: The system parses the data from the receipt and tosses non food items, then adds the food items to default locations unless otherwise specified.
    - Incorrect (Receipt Scan)
        - Input: User scans a receipt with non food items
        - Output: Notification that non food items were not added to inventory
    - Expected (Barcode Scan)
        - Input: User scans a barcode of food item
        - Output: The system connects to the barcode lookup api and then extracts the name and quantity information. From here the food items are sent to default locations unless otherwise specified.
    - Incorrect (Barcode Scan)
        - Input: User scans a barcode of a non food item
        - Output: Error “Not Edible”
- Update Ingredient Information
    - Expected
        - Input: User clicks/taps onto an ingredient in their inventory to modify location such as chips from the pantry to a cabinet
        - Output: The ingredient is checked if it has any conflicts such as its a cold item and can’t be placed in the cabinet. If a valid location then the item is moved from location A to location B
    - Incorrect
        - Input: User clicks/taps onto an ingredient such as raw chicken to be placed from the freezer to the pantry.
        - Output: The system allows this move without prompting the user that this is not a safe food practice and can lead to foodborne illness

- Create a list of ingredients
    - Expected
        - Input: User clicks/taps onto the create list button
        - Output: A list is created and the user is prompted to add ingredients
    - Incorrect
        - Input: User clicks/taps into the create list button
        - Output: A list is not created or not modifiable
- Delete Ingredients from inventory
        - Expected
        - Input:

                - User clicks/taps onto an ingredient in their inventory to modify it
                - User clicks/taps delete button on the selected ingredient
                - System Prompts partial or full item deletion
                - System prompts users if they want to add items deleted to the shopping list.
            - Output:
                - System updates user inventory accordingly
                - If confirmed, System updates users shopping list with the deleted item
    - Incorrect
        - Input:
            - User clicks/taps onto an ingredient in their inventory to modify it
            - User clicks/taps delete button on the selected ingredient
        - Output:
            - System delete the ingredient but fails to prompt the user for the partial/full deletion or whether to add the item to the shopping list
            - The system updates the shopping list without user confirmation (e.g., adds ingredients that weren't meant to be added to the shopping list or fails to add the correct items).
- System Tracking and adjustment of highly consumed items
    - Expected
        - Input:
            - User or roommate wants a glass of milk and pours a glass.
            - Periodically User updates inventory when they go shopping or the system adjust the estimated amount of milk left based on purchase history and use trends
            - User buys more milk and adds it to the system
            - The system notices milk was added either through manuel input or by scanning it in
            - System assumes previous milk was either used or discarded and prompts the user to confirm new entry of milk
        - Output:
            - System updates quantity of commonly consumed items
            - If old item (milk) is marked as used or to be tossed, the system resets the consumption trend bas on the new quantity

        - If an old item (milk) is not discarded, the system adjusts the expiration date and quantity.
    - Incorrect
        - Input:
            - User or roommate consume milk, but fail to log it
            - User buys new milk and adds it to the system
            - The system fails to track the trend of depletion and notify the user they are out or almost out of an item.
            - The system doesn't detect the correct estimation of the item (milk) that is left. It now assumes both the old and new milk are in the system.
        - Output:
            - System inventory shows more of an item then there really is in the system.

2.2. Recipe Manager

Description

The System should allow for users to manage their recipes and those they find. Recipes shall be organized onto “cards” with tags, complexity, average ratings, feedback, ingredient substitutions, directions, price per serving, serving count, time to prepare and cook, equipment needed (with substitutions if possible), nutritional information and pictures of the meal. Users should be able add recipes to their account via manual entry or from the system. Users should be able to post their recipes to the system for others to try out and then rate.

Required Behaviors

1. System should organize recipes into cards that display key information, such as description of recipe, picture of meal, and name of the dish
2. Users should be able to filter recipes by assigned tags
3. Users should be able to sort recipes by attributes such as rating, popularity, and ingredients
4. Users should be able to Search for recipes
5. Users should be able to create recipes
6. Users should be able to post a recipe to the system
7. Users should be able to rate and give feedback on a recipe
8. Users should be able to add recipes from the system to their account
9. Users should be able to modify their recipes
10. Users should be able to delete their recipes

I/O

- Recipes Organizing
    - Expected
        - Input: User searches for a chicken dish and then navigates recipes with chicken
        - Output: Recipes are displayed as “Cards”, with key details. More details are available upon user clicking/tapping onto a recipe card.
    - Incorrect
        - Input: User searches for a chicken dish and then navigates recipes with chicken
        - Output: System displays recipe cards in a cluttered unorganized manner or with incorrect information such as beef instead of chicken
- Recipe Filtering
    - Expected
        - Input: User searches for chicken dish and then filters by under 30 mins cook time
        - Output: System returns chicken dishes with cook times under 30 mins
    - Incorrect
        - Input: User searches for chicken dish and then filters by under 30 mins cook time
        - Output: System returns dishes without chicken as an ingredient or cook time greater than 30 mins
- Recipe Sorting
    - Expected
        - Input: User searches for a beef dish and then sorts by preptime (least to greatest)
        - Output: System sorts the results by prep time from least to greatest
    - Incorrect
        - Input: User searches for a beef dish and then sorts by preptime (least to greatest)
        - Output: System returns results and sorts them in reverse order, recipes without beef, or recipes are random
- Recipe Creation
    - Expected
        - Input:
            - User clicks/taps onto the create recipe button
            - System prompts all the fields needed for a recipe card
            - User fills in the data
        - Output: System return the newly created recipe card

    - Incorrect
        - Input: Same as above
        - Output: System returns a blank recipe card
- Recipe Posting
    - Expected
        - Input: User clicks/taps on a recipe they made then clicks/taps on post to public
        - Output: System screens the recipe to check to make sure everything is valid then posts it to the public once it passes verification
    - Incorrect
        - Input: User clicks/taps on a recipe some else made then clicks/taps on post to public
        - Output: System should reject a recipe made by not the User, it posts the recipe even though someone else made it.
- Recipe Rating
    - Expected
        - Input: User selects a recipe that they have made in order to rate it
        - Output: System screens the feedback for invalid language or anything inappropriate before posting both feedback and the rating
    - Incorrect
        - Input: User selects a recipe they have not made and rates it
        - Output: System fails to screen it and check if a user has made it first and posts the rating.
- Recipe Feedback
    - Expected
        - Input: User selects a recipe that they have made in order to provide feedback
        - Output: System screens the feedback for invalid language or anything inappropriate before posting both feedback and the rating
    - Incorrect
        - Input: User selects a recipe that they have not made in order to provide feedback of “it sucks”
        - Output: System fails to screen it and check if a user has made it first and posts the feedback anyway.
- Adding Recipes to a User's account
    - Expected
        - Input: User searches for a recipe with chicken then clicks/taps on one that looks good to them and taps the add button
        - Output: The System adds the recipe card to the correct user account

    - Incorrect
        - Input: User searches for a recipe with chicken then clicks/taps on one that looks good to them and taps the add button
        - Output: System adds the wrong chicken recipe to the users account
- Modifying Recipes
    - Expected
        - Input:
            - User selects a recipe they have created from their recipe list
            - User clicks/taps on the edit button and modifies fields such as name, ingredients, or directions
            - User clicks/taps the save button
        - Output: System successfully saves the the changes and the updated recipe appears in the users list
    - Incorrect
        - Input: User selects a recipe and attempts its modification but fails to save
        - Output: The system saves incorrect changes or incomplete data
- Deleting Recipes
    - Expected
        - Input: User selects a recipe from their recipe list and clicks/taps on the delete button, the system prompts “are you sure?”
        - Output: System removes the recipe from the user
    - Incorrect
        - Input: User selects a recipe to delete but system does not display the warning of deletion
        - Output: System fails to delete or deletes without confirmation by user

2.3.Meal Planner

Description

The system shall allow for a meal planner which allows the user to add meals to a day which are to be made. Recipes should be able to be added, deleted, and modified to a meal plan. Each recipe can have different stages such as prep time and cook time that will be shown on the meal planner as to maximize efficiency. Ingredients that need to be thawed, proofed or other techniques will be listed under the prep time. Cook time will be an estimate of the time the meal takes to cook and put together. Meals that make a lot of servings can be counted as leftovers for a meal.

Required Behaviors

1. Users should be able to plan meals for up to two weeks at a time, i.e breakfast, lunch, and dinner for a total of 42 meals
2. Users should be able to share their meal plan or menu with family members/ other users in the home
3. The system should break meals down into their phases such as prep time and cook time and place them on the the two week meal plan with prep phrases for a meal before its cook phase
4. Users should be able to add, delete, or modify a meal.
5. Users should be able to move the prep and cook times depending on their schedule and the recipe's flexibility
6. The System should add ingredients based on the current planned meals to a list of ingredients that the user needs to purchase
7. The System should remove ingredients that are used by a meal when the meal has been checked off as made

I/O

- Adding Meals to Menu
    - Expected
        - Input
            - User selects a recipe from their recipe book or the community
            - User inputs date and time for meal to be planned
            - Number of servings is entered and recipe is scaled accordingly
        - Output
            - Meal is added to the meal planner on the date
            - Ingredients needed is updated based on user inventory with missing ingredients
    - Incorrect
        - Input
            - No recipe is selected, or user selects a recipe but inputs 0 for servings or a negative serving count
        - Output
            - Error messages indicating invalid input and what the input is
- Deleting Meals from Menu
    - Expected
        - Input
            - Selection of a meal from the meal planner
            - Confirmation of deletion "Are you sure?"
        - Output
            - Meal is removed from the menu

                - Shopping list is updated to reflect the removal of ingredients no longer needed
        - Incorrect
            - Input
                - No meal is selected or attempt to delete a meal that does not exist
            - Output
                - Error message of invalid selection
- Modifying a Meal
    - Expected
        - Input
            - Selection of a meal in the planner
            - Changes to recipe, date, time, or serving count
        - Output
            - Meal details are updated in the meal planner
            - Shopping list is updated
    - Incorrect
        - Input
            - Invalid changes such as date in the past or invalid serving amount
        - Output
            - Error indicating the changes are invalid
- Move Prep and Cook Phases
    - Expected
        - Input
            - Selection of meal from planner
            - User holds/drags time fro prep or cook phase to new time
            - Durations are adjusted for phase
        - Output
            - Updated meal timing in the planner
            - Updated reminders or notifications for meal prep and cooking phases
    - Incorrect
        - Input
            - Invalid time adjustment such as cook before prep
        - Output
            - Error message of invalid time
- Sharing Menus
    - Expected
        - Input
            - Selection of a menu or meal plan
            - Selection of Users to share with
            - Permission for shared users (View only or edit)
        - Output
            - Menu is shared with selected Users

                - Notifications the menu has been shared
        - Incorrect
            - Input
                - Invalid Menu or Empty Menu
                - Invalid User selection
            - Output
                - Error Messages indicates that invalid menu or user selection
- Ingredient List To be Purchased
    - Expected
        - Input
            - Needed Ingredient List created from meals on menu
        - Output
            - List of ingredients needed based on current inventory and menu needs
    - Incorrect
        - Input
            - No meals planned or invalid selection
        - Output
            - Error message indicating empty ingredient list
- Meal was made
    - Expected
        - Input: Meal was made by a user
        - Output: System removed the ingredients that were apart of that meal from users inventory
    - Incorrect
        - Input: Ingredients not in inventory, Meal calls for more of a ingredient then a user has
        - Output: Error inventory quantity mismatch

2.4.Shopping Aid

Description

The system shall allow for the creation of lists based on the ingredients a user has in their inventory and what they need based on the meal planned meals. From here the user can check major stores such as Walmart, Target, and warehouse stores like Costco and Sam's Club for a price per unit comparison. This will allow for an estimation of the serving cost for a meal and when a user scans in their receipt the price of an item is recorded as something might have a different price from what it was listed for online. A user shall be able to filter stores based on distance, membership requirements, and preferences such as not liking the meat at store A or the produce at store B. A user will be able to have shopping lists made based on their ingredients needed list and store inputs. A quick create option will work to create shopping lists with the fewest stops and

closest distance to the user. Shopping lists will be able to be shared with family members or other users in the home.

Required Behaviors

1. System shall allow for creation of shopping lists based on needed ingredient list made from the meal planner
2. System shall allow for the sharing of these shopping lists
3. System shall get prices of items from major retailers
4. System shall allow users to sign into retailers like Costco or Sams to get member only pricing
5. Users shall be able to filter stores based on membership, distance, and preference
6. System shall allow for the user to compare unit prices to find cheapest options
7. System shall update a meals estimated price per serving when a receipt is scanned and the price has a mismatch from the online price
8. System shall have a quick create option for making the shopping list

I/O

- Shopping List Creation
    - Expected
        - Input: Menu, User's Inventory, and Store Preferences
        - Output: Generated Shopping list with items categorized by store and section ie Meat, Produce, Dairy, and so on
    - Incorrect
        - Input: No planned meals or empty Needed Ingredient list
        - Output: Error of invalid menu or empty ingredient list
- Sharing Shopping Lists
    - Expected
        - Input: Shopping list is selected then shared to other users
        - Output: Shopping list is shared with the selected users
    - Incorrect
        - Input: Empty shopping list, invalid users
        - Output: Error of empty list or invalid users
- Prices of Items
    - Expected
        - Input: User selects a recipe to add to menu and a estimated price per serving is shown
        - Output: Displays the current cost per serving based on the available ingredient prices
    - Incorrect
        - Input: Invalid ingredients, Incomplete prices
        - Output: Error indicating price per serving is unavailable
- Retailer Sign in
    - Expected

        - Input: User inputs retailer login credentials
        - Output: Successful sign, access to member only pricing
    - Incorrect
        - Input: Incorrect username or password
        - Output: Error message indicating login failed
- Store Filter
    - Expected
        - Input: User imputed store preferences, filters like preferred stores, store distance, etc
        - Output: filtered list of stores based on user input
    - Incorrect
        - Input: Invalid store filter such as negative distance or incorrect store names
        - Output: Error indicating parameters are invalid
- Unit Price Comparison
    - Expected
        - Input: User selects a recipe and adds it to their menu
        - Output: System compares prices of stores to show a estimate of price per serving
    - Incorrect
        - Input: User selects a recipe with a ingredient with wrong unit or non traditional units
        - Output: Error message “Check ingredient sizes”
- Serving Estimated Price
    - Expected
        - Input: Menu of planned meals with current price data for the week
        - Output: Estimated price per serving based on current ingredient prices
    - Incorrect
        - Input: Missing price data for a ingredient or invalid meals
        - Output: Error for invalid meals or a Warning message about incomplete cost estimation
- Quick Create Option for Shopping Lists
    - Expected
        - Input: Menu of planned meals
        - Output: Automatically generated list of ingredients needed optimized for fewest stops and shortest distance while balancing cost
    - Incorrect
        - Input: Empty Menu, Invalid User Preferences
        - Output: Error indicating Invalid Menu, or Preferences

2.5.Admin Backend

Description

The system should allow for administrators to be able to view all current registered users, their feedback, and recipes posted to the system. Administrators should be able to review and remove fraudulent feedback invalid user names, and invalid recipes. Automated screening for common words and phrases that are inappropriate will be used to speed up the process of this screening. Administrators should also be able to seed the initial database with recipes for users to use and add to their own recipe lists.

Required Behaviors

1. System should allow Administrators to view current registered users names, feedback, and submitted recipes
2. System should allow Administrators to filter Names, feedback, and submitted recipes
3. Administrators should be able to remove invalid feedback and recipes
4. System should allow administrators to automatically screen feedback, recipes, and users for invalid names or words and inappropriate names or words
5. System should allow for a recipe approval
6. System should allow for screening management
7. System should allow for automated flagging of content to be reviewed by an administrator
8. System should allow administrators to seed the recipe database in bulk with a CSV or other formats
9. System should allow for a automated notification system to users
10. System should allow for Administrators to send users messages regarding content removal, account actions, or other community related issues
11. System should provide activity logs for all administrative actions such as content removal, account actions and bulk imports.
12. System should allow Users to report feedback, recipes, or other Users for invalid, inappropriate or “other” issues

I/O

- Administrator User Management
    - Expected
        - Input: Search criteria such as usernames, registration date, and account status
        - Output: Display list of users matching the search criteria
    - Incorrect

        - Input: invalid search criteria such as invalid username or a non existent date
        - Output: Error indicating the invalid criteria or date
- Filtering and Search of User content
    - Expected
        - Input: search criteria such as keywords, content types, or date range
            - Filters for the contents of a user such as all the recipes they posted
        - Output: Display list of content matching the search criteria and filters
    - Incorrect
        - Input: Invalid search criteria such as date range in a non standard format
        - Output: error indicating invalid dates
- Content Moderation
    - Expected
        - Input: selection of content for moderation
        - Output: confirmation of actions taken, updated on the content status, and log entry
    - Incorrect
        - Input: invalid selection of content such as nonexistent content
        - Output: error content selection is invalid
- Automated Screening
    - Expected
        - Input: Keywords to be be flagged are in a list
        - Output: Continent with flagged keywords are prevented from being posted until those words are removed by user
    - Incorrect
        - Input: Invalid words or special characters only being used in the list of flagged keywords
        - Output: Error indicating screening failure
- Bulk Recipe Seeding
    - Expected
        - Input: CSV or other structured file containing recipes with the required fields
        - Output: Confirmation to upload/import recipe list, display newly added recipes to the DB and log entry into admin logs
    - Incorrect
        - Input: wrong file type or structure
        - Output: Error file not readable “Please submit CSV or other accepted file types”
- Admin User Communication
    - Expected

        - Input: selection of user(s) to send a message to, message content
        - Output: confirmation of message being sent, display the message in the User(s) notifications
    - Incorrect
        - Input: Invalid user selection, missing message contents
        - Output: error message indicating message failed or can't send
- Reporting of content
    - Expected
        - Input: User selects content such as user, recipe, or a feedback message to report as inappropriate
        - Output: System flags and does a secondary screening of content
    - Incorrect
        - Input User reports a nonexistent user or a message that is deleted
        - Output: Error message that the content reported no longer exists
- Content and Admin Reports
    - Expected
        - Input: User activity logs, recipes posted, feedback
        - Output: Admin generates a report of recipes posted between a set of dates and have feedback from a certain user
    - Incorrect
        - Input: Inactive users with no feedback
        - Output: Warning message of no feedback for a user that matches report criteria

---

3. Interface (User Interfaces and System Interfaces)

3.1. System shall have a desktop application for Users
3.2.System shall have a desktop application for Admins
3.3.System shall have a Dark and Light Modes
3.4.System shall have a distinct look for Admins vs Users
3.5.System shall have the following UI/UX

- 3.5.1. User Dashboard
- 3.5.2. Admin Dashboard
- 3.5.3. User Profile
- 3.5.4. User settings
- 3.5.5. Community Dashboard
- 3.5.6. Recipe Cards
- 3.5.7. Meal Planner
- 3.5.8. User Inventory
- 3.5.9. Admin Reports

3.5.10. Admin Content Management

3.6. System shall use Tesseract for OCR

3.7. System shall use open food facts for nutritional information

---

4. Performance (Time, Space, Accuracy)

4.1. Mobile build size limit is 4GB

4.2. Prices should update weekly

4.3. Recipe Screening should take no more than five minutes

4.4. Inventory updates should take no more than 2-3 seconds

4.5. Recipes should be displayed within 5 secs of a user searching the database

4.6. Modifying meals or recipes should take no more than 5 seconds

4.7. Admin should be able perform moderation onto recipes, feedback, and users and have it applied within 2 seconds

4.8. Bulk recipe addinging should take no more then 10 mins for up to 1000 recipes

4.9. Scanned data from barcodes should be processed within 2 seconds

4.10. Receipts be scanned should have their data extracted and processed within 5 seconds

4.11. The System should support up to 100 concurrent users without degradation in recipe browsing

4.12. UI/UX should load and respond on all pages within 5 seconds of clicking/tapping on a button

4.13. Price preserving estimates should be 90% accurate when compared with the final price a user scans in from the receipt.

4.14. The receipt scanning should have a 90% accuracy rate