FOODSTUFFS

INVENTORY MANAGEMENT AND RECIPE SYSTEM

Welcome

Username

Password

Enter

Sign Out

Welcome: User

FOODSTUFFS
INVENTORY MANAGEMENT AND RECIPE SYSTEM

Navigation Menu

Inventory Dashboard

Recipe Manager

Meal Planning

Scan Groceries

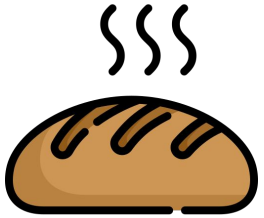
Shopping List

Metrics and Reports

Exit

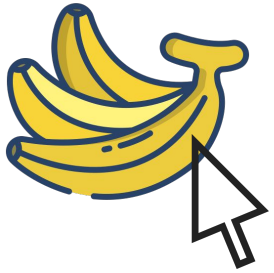
Item to Search for

Sort by:
Quantity
Type
Expiration



Sourdough Bread

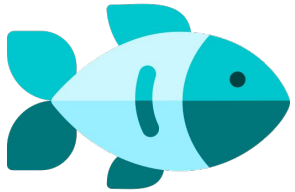
Remove item



Bananas

Remove item

Price
Quantity
Brand



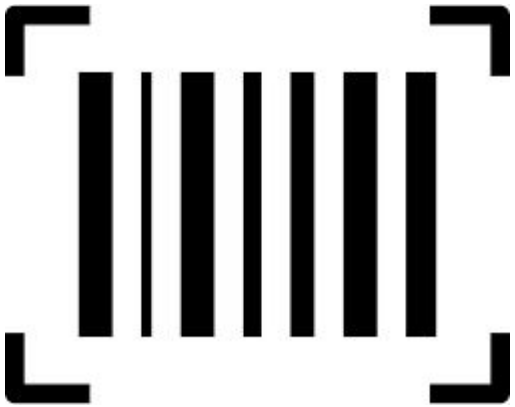
Salmon

Remove item

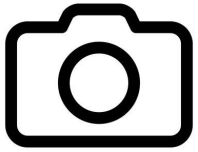
Expires in 2 days

Add item

Exit



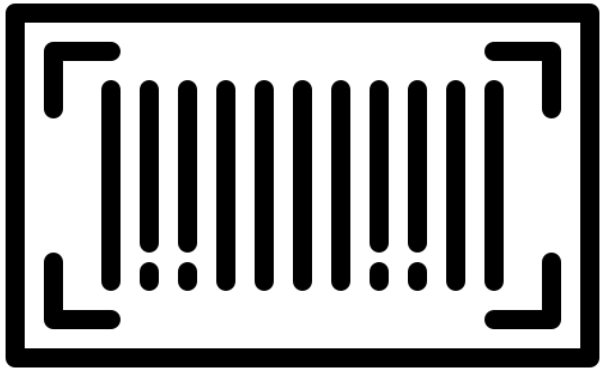
Select to scan Barcode (Camera Access Required)



Select to scan Manually type barcode

Exit

Align Barcode to Center



Input Quantity

Exit

Input Barcode

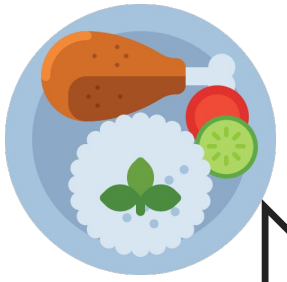
Input Quantity

Exit

Recipe to Search for

Sort by:
Time to Cook
Price
Difficulty
Healthy

Spaghetti and Meatballs



Chicken and Rice

Ingredients required

Burrito

Missing “cheese”

Add Custom Recipe

Meal Plan Queue

1. Chicken and Rice
2. Lasagna
3. Meatloaf
4. Chicken Parmesan
5. Sushi
6. Steak and Fries
7. Hamburger and Fries

Exit

Your Cart:

- | 2 | +

$13.14

Kroger

Cheapest Price

- | 1 | +

$0.69

Kroger

Cheapest Price

Added by Meal Plan Queue:

- | 1 | +

$2.69

Kroger

Alternative Options

$2.48 Walmart Replace

Recommended to restock:

- | 0 | +

$0.00

Kroger

Alternative Options

Order Summary

(4 items)

Subtotal: $16.52

Checkout

Frequently Bought Together

+

Add to cart

Exit

◀ Week of 11/17/24 ▶

11/17/24
- Burrito
- Chicken and r…
- Sushi

11/18/24
- Waffles
- Egg Fried Rice
- Tacos

11/19/24
- Eggs and Bac…
- Caesar Salad

11/20/24
- Ham Sandwich
- Meat Loaf
- Chicken Parm…

11/21/24
- Omelette -
- Lasagna -
- Spaghetti and… -

+

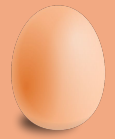
11/22/24
- Chicken and r… -

+

11/23/24
+

Missing Ingredients:

Omelette



Add to cart

Add to cart

Spaghetti and Meatballs

Add to cart

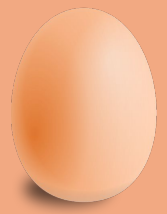
Exit

Weekly Report

Export as PDF

Item Usage

7 Recipes

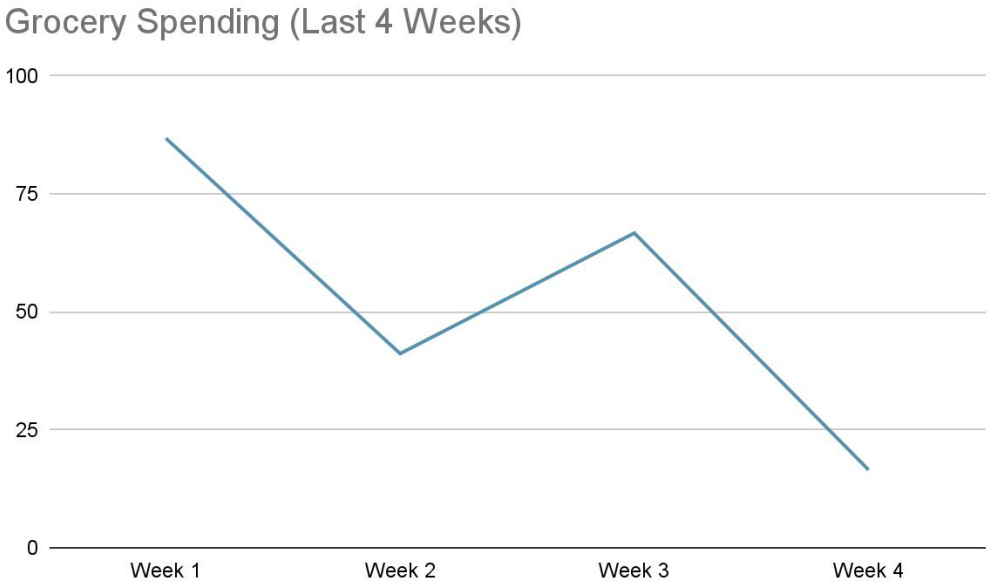


5 Recipes

4 Recipes

Inventory Tracker

- + 2 Chicken (11/21/24)
- + 1 Onion (11/21/24)
- + 1 Cheese (11/21/24)
- - 1 Chicken (11/20/24)
- - 1 Broccoli (11/20/24)
- - 1 Egg (11/20/24)
- - 1 Rice (11/20/24)
- - 1 Egg (11/19/24)



30% (-7%)

73% (-0%)

51% (-1%)

Pseudocode Use Cases for FoodStuffs Design Project

Andrew Prieto, Braedon Johnson, Bryan Rich, Evan Howe, Fardin Khan, Zifeng Li

November 22, 2024

Algorithm 1: Store User Recipe into Database

```
Input: r - Recipe to store, d - Recipe database
Output: Recipe is added to database
if r is invalid then
    Notify user: "Recipe is invalid"
    Return to main screen
else
    Add r to d
    Notify user: "Recipe added to recipe database."
```

Algorithm 2: Create Custom Recipe

```
Input: t - Template type (Online recipe or new recipe)
Output: New recipe is created
if User selects option t 'Online Recipe' then
    if Device is connected to internet then
        Connect user to external Recipe API to search for desired Recipe to save
        if User selects Save and Finish then
            Save recipe to database, exit recipe creator
        else
            Restart recipe creation process
    else
        Notify User: "Device is not connected to the internet"
        Prompt user to connect to internet.
User then has to select create new recipe
User creates custom recipe with name n
Notify User: "Recipe n created"
```

Algorithm 3: Add to Shopping List If Ingredients Are Missing

```
Input: r - Selected recipe, f - Fridge inventory, s - Shopping list
Output: Shopping list is updated with needed ingredients
User opens library of saved recipes
User selects recipe r
if Ingredients in r are missing from fridge f then
    Device prompts user: "Missing ingredients found. Add to shopping list?"
    if User chooses to add to shopping list then
        if User has no shopping list then
            Notify user to create new shopping list
        else
            Add ingredients to shopping list
            Notify user: "Shopping list s has been updated with ingredients"
    else
        Notify user: "No shopping list created."
else
    Notify user: "Fridge has all ingredients for recipe r."
```

Algorithm 4: Sort Recipes Based on Ingredients

```
Input: r - Library of saved recipes, i - Ingredient(s) selected by user
Output: Sorted list of recipes based on selected ingredients
User opens library of saved recipes
User selects option to sort recipes by ingredient(s)
if Device is connected to the internet then
    if i is valid then
        Device prompts user to input ingredient(s) they want to sort by
        User inputs the ingredient(s)
        foreach Recipe that contains i do
            Display recipe to user
        User selects a recipe from the filtered list
    else
        Notify user: "Invalid ingredient(s) entered."
        Prompt user to re-enter valid ingredient(s)
else
    Notify user: "Device is not connected to the internet"
    Filter and sort recipes based on locally available recipes
if No recipes found with the selected ingredients then
    Device suggests partial matches or substitutions
    User selects a suggested recipe or redoes the search with new ingredients
```

Algorithm 5: Manually Add New Item(s) to Inventory

```
Input: i - Item details (name, quantity, expiration date), q - Minimum quantity for alert system
Output: Item is added to the inventory database
User selects 'Add item to inventory'
Device prompts user to scan item via barcode or manual entry
if User selects manual entry then
    User inputs details for i (name, quantity, expiration date) and q
    User confirms entry
    if Item details are within reasonable limits and q greater than or equal to 0 then
        Add item i to inventory database
        Notify user: "Item added to inventory"
    else
        Notify user: "Incorrect item details, please edit"
        User selects 'edit' and starts over from step 2
else
    User scans barcode
    Barcode is read in using external API that delivers basic product information (Name)
    if Barcode reader produces an error while reading then
        Notify user: "Error reading barcode. Please manually input information."
        Prompt user to enter in details using manual entry mode
    else
        User confirms entry
        Add item i to inventory database
        Notify user: "Item added to inventory."
```

Algorithm 6: Alert Items Getting Low

```
Input: d - Inventory database, i Item in database, s - Shopping list
Output: Notification of low items and updated shopping list
User logs into the application
if i in d is below user specified amount then
    Notify user: "Item i is running low."
    Device prompts: "Would you like to add these items to your shopping list?"
    if User accepts then
        Add item i to shopping list s
        Notify user: "Shopping list updated with item i"
    else if User ignores request then
        Item i is marked as ignored in d, will not notify about low quantity upon rerun.
        Notify user: "Item i is ignored and not added to shopping list."
else
    Notify user: "All items are sufficiently stocked in the inventory."
```

Algorithm 7: Remove Item(s) from Inventory

```
Input: r - Items to Remove, d - Item Database
Output: Item(s) are removed from the database
if d is empty then
    Notify user: "Inventory is Empty"
    Return to main screen
else
    if r is empty then
        Notify user: "Removal Selection is Empty"
        Return to Main Screen
    else
        Prompt User: "Confirm Removal Selection"
        if r is correct then
            Remove items specified in r from d
            Notify User: "Removal Complete"
            Return to Main Screen
        else
            Return to Selection Screen
```

Algorithm 8: Scan Grocery Items into Inventory

```
Input: a - Scanned Item to Add, d - Item Database
Output: Item(s) are added to the database
if a has a valid UPC then
    Prompt User: "Enter Quantity"
    Prompt User: "Enter Expiration"
    if Quantity and Expiration are Valid then
        Enter a into d
        Notify User: "Item Added"
        Prompt User: "Add more items?"
    if Add More Items then
        Return to Scan Item Screen
    else
        Return to Main Screen
else
    Notify User: "Item is Invalid, Please Enter Manually"
    Enter Algorithm 5 Functionality
```

Algorithm 9: Track quantity of Partitioned Goods

```
Input: i - Partitioned Item to Modify, d - Item Database
Output: Item(s) quantity are updated in the database
if i is a partitioned item then
    Prompt User: "Update Quantity of a"
    Prompt User: "Confirm Update?"
    Submit changes in i to d
    if i is expended then
        Prompt User: "Remove Item?"
        if Remove Item then
            Remove i from d
            Notify User: "Item Removed"
            Return to Main Screen
        else
            Prompt User: "Enter Amount Remaining"
            Prompt User: "Confirm Update?"
            Submit changes in i to d
            Notify User: "Item Updated"
            Return to Main Screen
    else
        Notify User: "Item Updated"
        Return to Main Screen
else
    Prompt User: "Change Quantity Type?"
    if Change Quantity Type then
        Change Quantity type of i to be partition-able
        Prompt User: "Enter Amount Remaining"
        Prompt User: "Confirm Update?"
        Submit changes in i to d
        Notify User: "Item Updated"
        Return to Main Screen
```

Algorithm 10: Identify Untracked Item Usage

```
Input: s - Abnormal Sensor Inputs
Output: Item(s) are added to the database
if System Detects s then
    Prompt User: "Check for Untracked Items"
    if Untracked Items Exist then
        Prompt User: "Add Untracked Items, Scan or Enter Manually"
        if User enters Scan then
            Enter Algorithm 5 Functionality
        if User enters Scan then
            Enter Algorithm 8 Functionality
        else
            Return to Main Screen
    else
        Return to Main Screen
```

Algorithm 11: Modify Inventory Based on Completed Recipes

```
Input: r - Prospective Recipe, d - Item Database
Output: Item(s) are/are not removed/modified database
if All ingredients in r are present in d then
    if All ingredients in r are NOT on hold status then
        Place items listed in r on hold status in d
        Prompt User: "Confirm Recipe Completion"
        if Completion Confirmed then
            Remove and update quantities in d in accordance with r
            Return to Previous Page
        else
            Prompt User: "Please Update Inventory"
            Prompt User: Update Quantity of all items in r
            Return to Previous Page
    else
        Notify User: " Ingredients in use by In-Progress Recipe: X, Y,...,Z"
        Return to Recipe Page
else
    Notify User: "Insufficient Ingredients: X, Y,...,Z"
    Return to Recipe Page
```

Algorithm 12: Modify Inventory Based on Grocery Receipt

```
Input: r - Receipt, d - Item Database
Output: Item(s) are added to the database
if r is NOT supported (does not have a valid QR Code) then
    Notify User: "Receipt is unsupported, entry will use OCR"
else
    if r exists in Receipt History then
        Notify User: "Receipt has been previously entered."
        Prompt User: "Enter again?"
        if User Enters Again then
            Notify User: Scanned List of Items
            Prompt User: "Confirm Item List?"
            if User confirms Item List then
                Add items from r into d
                Notify User: "Items added to Database"
                Return to Previous Screen
            else
                Re-scan items in r, restarting Algorithm 12
        else
            Return to Previous Screen
```

Algorithm 13: Track and Alert User on Upcoming Expiration Dates

```
Input: i - Inventory database, e - Expiration date of item(s), l - Grocery list
Output: Notifies user of expiring items and updating grocery list
foreach food item f in i do
    if current date + 5 ≥ f's expiration date then
        Notify user of expiring item
        if User acknowledges notification then
            Prompt user: "Add this item to the grocery list?";
            if user confirms then
                Add item to grocery list
            else
                Mark item as ignored in the inventory list
```

Algorithm 14: Allow Shopping List Substitutions

```
Input: l - Shopping list, external shopping list API
Output: Update shopping list with substitutions
Output: Updated shopping list with substitutions or ignored items
foreach item in l do
    Query availability of item from store API
    if item is unavailable then
        Fetch substitution options from store API
        Notify user: "Item unavailable. Would you like to substitute?"
        if user accepts substitution then
            Replace item in l with substitution item
        else
            Mark item as ignored for this trip
```

Algorithm 15: Add Items to Shopping List

```
Input: l - Shopping list, i - Item to add to shopping list
Output: Updated shopping list with added item
User clicks 'Add Item' button on shopping list l; Prompt user to enter item name and quantity
if Item details are valid then
    Search for item in the online database;
    if item is found then
        Display search results to user
        if user selects an item then
            Add the selected item to the shopping list
    else
        Notify user: "Item not found. Re-enter or add a generic line item."
        if user reinputs details then
            Repeat search
        else
            Add a generic item with only a name and quantity
else
    Notify user of missing input and restart add item process
```

Algorithm 16: Remove Items from Shopping List

```
Input: l - Shopping List, i - Item(s) to remove
Output: Update shopping list with specified items removed
User selects items to be removed
Display confirmation dialog with selected items i
if user confirms removal of i from l then
    foreach i specified to be removed from l do
        Notify user: "Item i removed from shopping list l"
        Remove item i from l
else
    Cancel removal process;
    Notify user: "No changes made to shopping list."
    Return user to the main shopping list view
```

Algorithm 17: Display Shopping List Amount

```
Input: l - Shopping List
Output: Total item count and detailed shopping list displayed
User selects shopping list to view;
if l is not empty then
    Count the total number of items;
    foreach i item in l do
        Display item i with quantity, and cost (if avaliable)
else
    Notify user: "Shopping list is empty.";
    Return to the main menu;
```

Algorithm 18: Sort Shopping List by Department

```
Input: l - Shopping List, store API that contains item's departments
Output: Sorts shopping list by department
User selects shopping list l to sort
User clicks 'Sort by Department' button
foreach item in l do
    Query item department from database/API
    if department is found then
        Assign item to its department
    Mark item as 'Unsorted'
Sort shopping list by department, with unsorted items at the top
Display sorted shopping list to the user
```

Algorithm 19: Track Shopping Metrics

```
Input: L - Shopping list, R - Scanned receipts
Output: Shopping metrics presented to the user
if L or R is empty then
    Notify user: "Shopping data is incomplete"
    Return to main screen
else
    Calculate weekly spending W from R and L if Receipts R and Shopping Trips T do not align
      then
        Display weekly amount W and overall amount O
    else
        Display weekly spending and shopping metrics with graphs from W
```

Algorithm 20: Recommend Frequently Bought Items

Input: L - Shopping list, R - Scanned receipts
Output: Receive suggestions on what they frequently buy so they can add to shopping lis

```
if L or R is empty then
    Notify user: "Shopping data is incomplete"
    Return to main screen
else
    Get suggestion list S from items bought often in R
    Display S to user as suggestion list
if user accepts suggestion items from S then
    Add items to shopping list L
else
    Remove items from S
```

Algorithm 21: Setup Meal Plan

Input: L list of recipes the user wants to make T time period
Output: A 'meal plan' is created with queued recipes over a specified time period and notifications are scheduled

```
if R is empty then
    Notify user: "Recipe list is empty"
    Return to main screen
else
    system prompts user to input days D to spread out the recipes and reconfirms the users selection
while D ≤ 0 do
    the system throws errorE stating invalid time amount and re promts user for time D
if user rejects scheduled then
    prompt user to reinput days D
if user ignores notification then
    Notify user of missed meal plan
    if user wants to repeat recipe then
        extent time by one day and update R
    else
        keep original list R
```

Algorithm 22: Append to shopping list based on meal queue

Input: L list of recipes ,M meal plan, T start times
Output: User's shopping list is automatically updated with items needed for meal plan

```
if L is empty then
    Notify user: "Recipe list is empty"
    Return to main screen
if T start time is in the future then
    if are all ingredients in L then
        merge shopping list with ingredients list
    else
        Add required items to shopping list
else
    prompt user that time must be in the future
```

Algorithm 23: Sort Recipes Based On Criteria Tags

```
Input: T - Tags selected by the user, D - Recipe database
Output: User is presented with recipes that correspond with requested tags
if T is empty or D is empty then
    Notify user: "Invalid Inputs"
    Return to main screen
else
    recipes that match the presented tags saved inR
Display R recipe list
```

Algorithm 24: Allow Recipe Substitution on Meal Plans

```
Input: T items in inventory, M meal plan from user, I ingredients list from user,UI update
       inventory list with used items
Output: User's meal plan consistently has recipes prepared similar to user specification
while M is not empty do
    if item missing in I then
        generate suggestion SUG from I in T inventory
    prompt user with suggestion
    if user confirms substitution then
        update inventory with SUG Notify user: "Meal plan updated with substitution"
    else
        Skip recipe
```