KitchenSync

A Pantry and Recipe Companion

•••

Milestone 1

Milestone 1 Objectives

- Explore and select technical tools
- Create hello world demos to test the tools
- Resolve technical issues
- Select collaboration tools
- Requirements Document
- Design Document
- Test Plan

Technical Tools

- Tesseract for OCR (Optical Character Recognition) - Receipt processing
- OpenFoodFacts - Data on Ingredients from product codes
- PIL and Pyzbar - For processing Barcodes into getting the product codes
- AWS's Dynamodb - This is for our cloud recipe storage

Technical Challenges

Resolved:

- Read Product Code from barcode
- Cloud storage for communal recipes

Unresolved:

- Information on receipts need to be filtered out
- How small can we make a recipe for storage
- Getting access to stores that require membership

Requirements

Functional Requirements

1. Scanning of barcodes and receipts for ingredient entry
2. Modify Ingredient Entries
3. System auto tracking of commonly used or purchased items
4. Recipes will be organised into cards for a standard structure
5. Communal Recipe sharing between users
6. Meal planning for 42 meals or two weeks
7. Sharing of the menu created
8. Shopping lists based on ingredients needed from the menu
9. Lists will be shareable
10. Quick Create Option for shopping lists
11. Admins will be able to see all users
12. Audit Logs for all activities
13. Bulk recipe Uploads

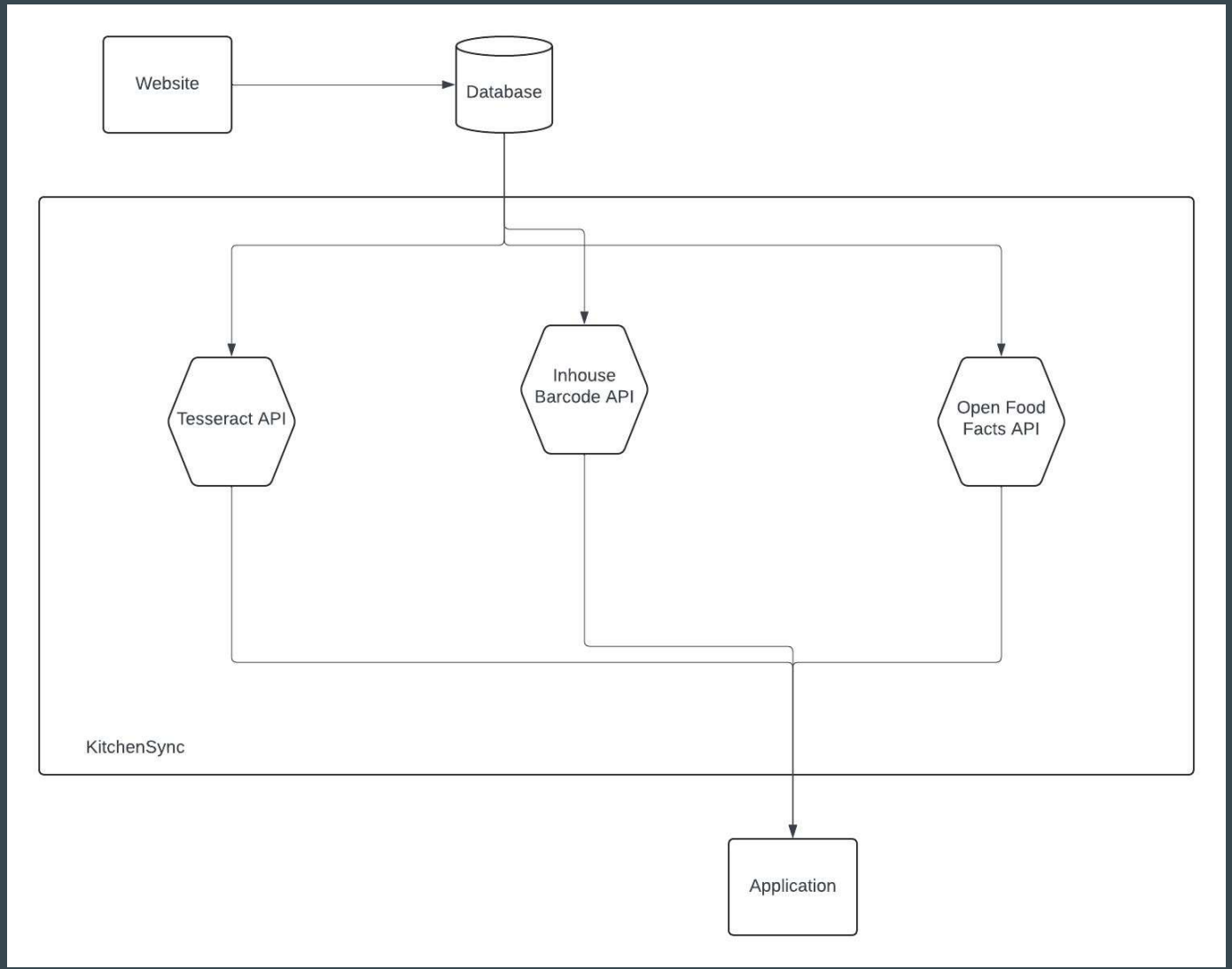
Interface Requirements

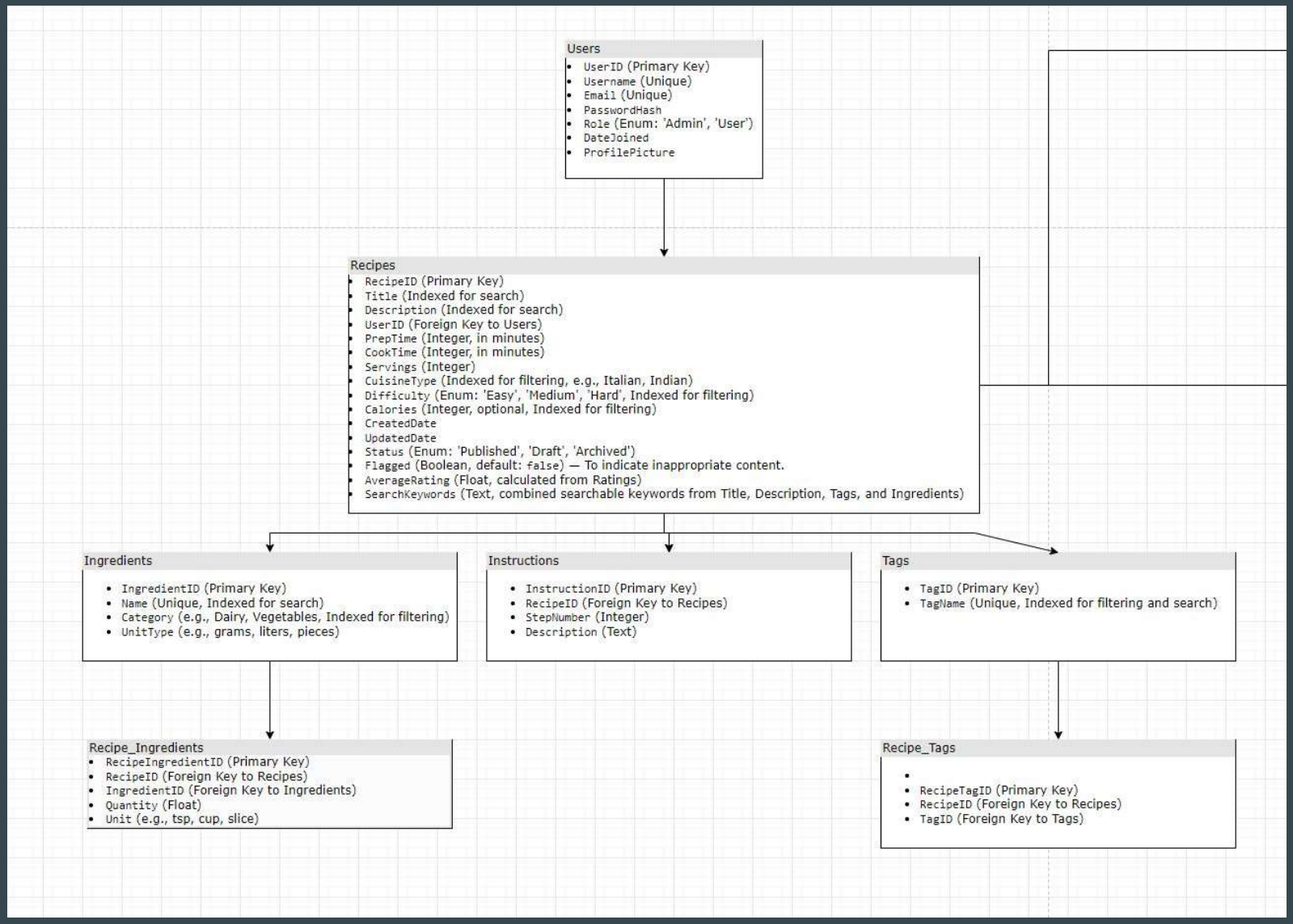
1.1. System shall have a desktop application for Users
1.2. System shall have a desktop application for Admins
1.3. System shall have a Dark and Light Modes
1.4. System shall have a distinct look for Admins vs Users
1.5. System shall have the following UI/UX
    1.5.1. User Dashboard
    1.5.2. Admin Dashboard
    1.5.3. User Profile
    1.5.4. User settings
    1.5.5. Community Dashboard
    1.5.6. Recipe Cards
    1.5.7. Meal Planner
    1.5.8. User Inventory
    1.5.9. Admin Reports
    1.5.10. Admin Content Management
1.6. System shall use Tesseract for OCR
1.7. System shall use open food facts for nutritional information

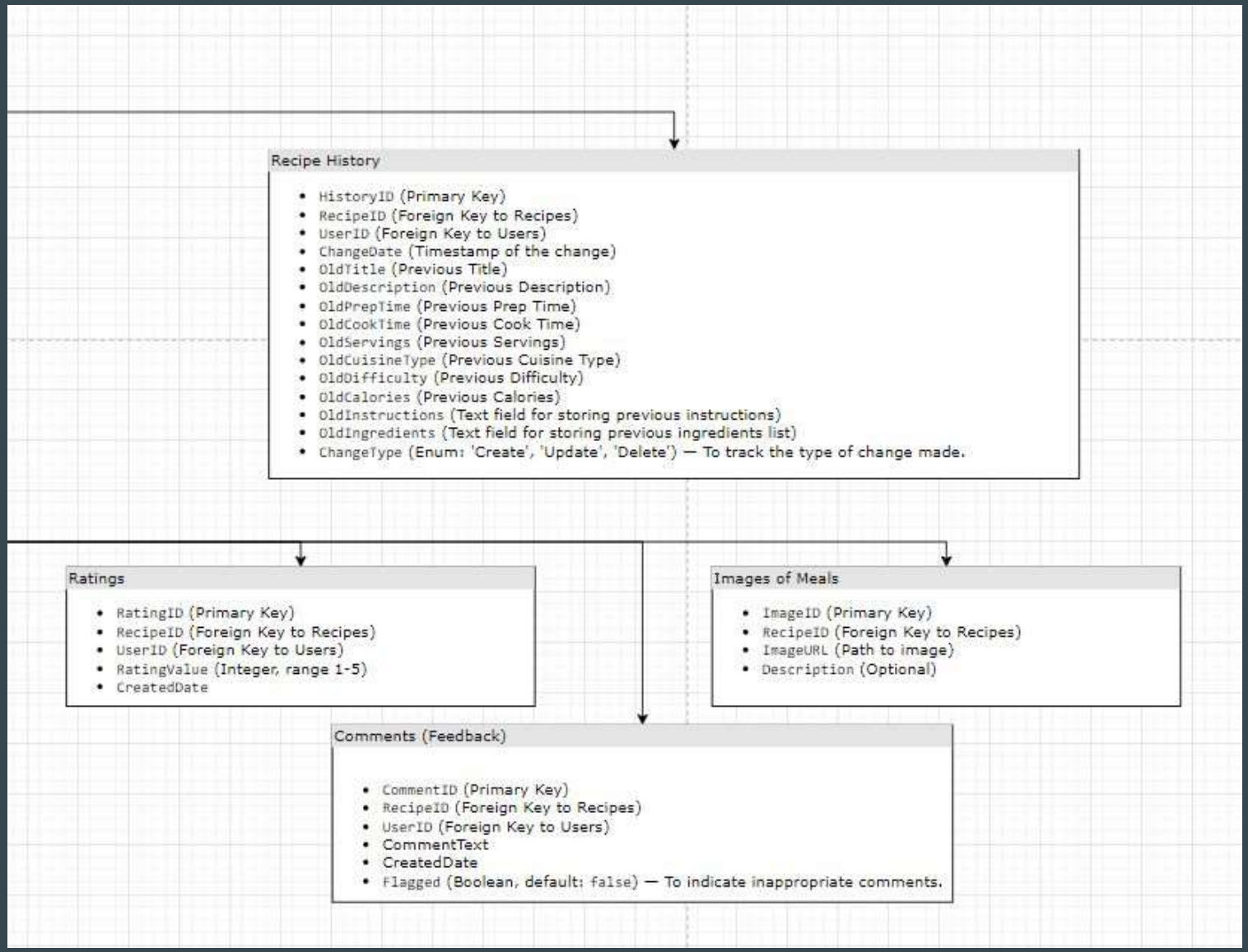
Performance Requirements

1.1. Mobile build size limit is 4GB
1.2. Prices should update weekly
1.3. Recipe Screening should take no more than five minutes
1.4. Inventory updates should take no more than 2-3 seconds
1.5. Recipes should be displayed within 5 secs of a user searching the database
1.6. Modifying meals or recipes should take no more than 5 seconds
1.7. Admin should be able perform moderation onto recipes, feedback, and users and have it applied within 2 seconds
1.8. Bulk recipe addinging should take no more then 10 mins for up to 1000 recipes
1.9. Scanned data from barcodes should be processed within 2 seconds
1.10. Receipts be scanned should have their data extracted and processed within 5 seconds
1.11. The System should support up to 100 concurrent users without degradation in recipe browsing
1.12. UI/UX should load and respond on all pages within 5 seconds of clicking/tapping on a button
1.13. Price preserving estimates should be 90% accurate when compared with the final price a user scans in from the receipt.
1.14. The receipt scanning should have a 90% accuracy rate

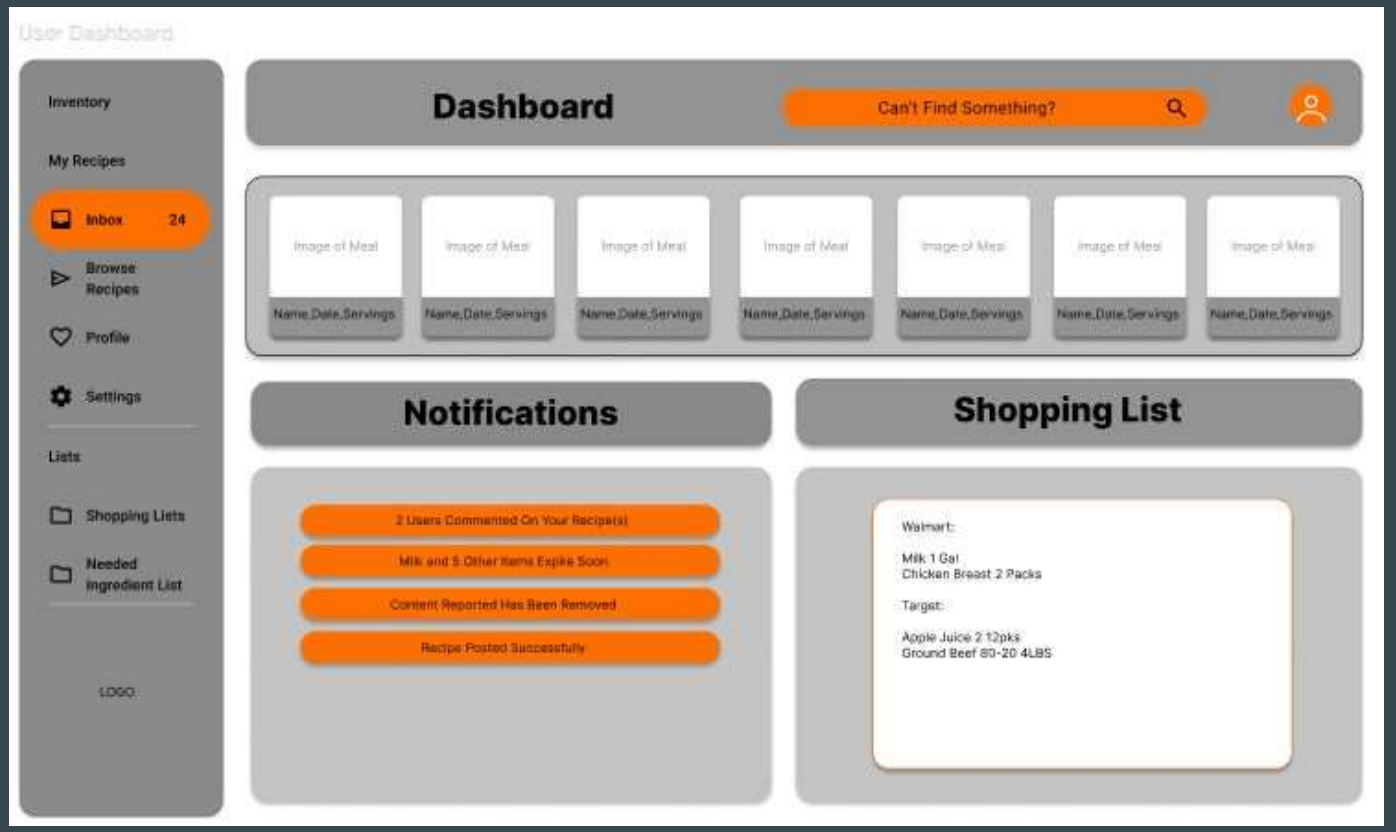
Design


Website
Database
Tesseract API
Inhouse
Barcode API
Open Food
Facts API
KitchenSync
Application


Users
UserID (Primary Key)
Username (Unique)
Email (Unique)
PasswordHash
Role (Enum: 'Admin', 'User')
DateJoined
ProfilePicture
Recipes
RecipeID (Primary Key)
Title (Indexed for search)
Description (Indexed for search)
UserID (Foreign Key to Users)
PrepTime (Integer, in minutes)
CookTime (Integer, in minutes)
Servings (Integer)
CuisineType (Indexed for filtering, e.g., Italian, Indian)
Difficulty (Enum: 'Easy', 'Medium', 'Hard', Indexed for filtering)
Calories (Integer, optional, Indexed for filtering)
CreatedDate
UpdatedDate
Status (Enum: 'Published', 'Draft', 'Archived')
Flagged (Boolean, default: false) — To indicate inappropriate content.
AverageRating (Float, calculated from Ratings)
SearchKeywords (Text, combined searchable keywords from Title, Description, Tags, and Ingredients)
Ingredients
IngredientID (Primary Key)
Name (Unique, Indexed for search)
Category (e.g., Dairy, Vegetables, Indexed for filtering)
UnitType (e.g., grams, liters, pieces)
Instructions
InstructionID (Primary Key)
RecipeID (Foreign Key to Recipes)
StepNumber (Integer)
Description (Text)
Tags
TagID (Primary Key)
TagName (Unique, Indexed for filtering and search)
Recipe_Ingredients
RecipeIngredientID (Primary Key)
RecipeID (Foreign Key to Recipes)
IngredientID (Foreign Key to Ingredients)
Quantity (Float)
Unit (e.g., tsp, cup, slice)
Recipe_Tags
RecipeTagID (Primary Key)
RecipeID (Foreign Key to Recipes)
TagID (Foreign Key to Tags)


Recipe History
HistoryID (Primary Key)
RecipeID (Foreign Key to Recipes)
UserID (Foreign Key to Users)
ChangeDate (Timestamp of the change)
OldTitle (Previous Title)
OldDescription (Previous Description)
OldPrepTime (Previous Prep Time)
OldCookTime (Previous Cook Time)
OldServings (Previous Servings)
OldCuisineType (Previous Cuisine Type)
OldDifficulty (Previous Difficulty)
OldCalories (Previous Calories)
OldInstructions (Text field for storing previous instructions)
OldIngredients (Text field for storing previous ingredients list)
ChangeType (Enum: 'Create', 'Update', 'Delete') — To track the type of change made.
Ratings
RatingID (Primary Key)
RecipeID (Foreign Key to Recipes)
UserID (Foreign Key to Users)
RatingValue (Integer, range 1-5)
CreatedDate
Images of Meals
ImageID (Primary Key)
RecipeID (Foreign Key to Recipes)
ImageURL (Path to image)
Description (Optional)
Comments (Feedback)
CommentID (Primary Key)
RecipeID (Foreign Key to Recipes)
UserID (Foreign Key to Users)
CommentText
CreatedDate
Flagged (Boolean, default: false) — To indicate inappropriate comments.

User Dashboard



Test Plan

Functional Test Example:

Task	**Adding Recipes to a User's account**
Methods	Ensure that the user can add recipes to their personal recipe list.
Inputs	User takes a recipe that they like and presses the "add" button.
Outputs	The system adds the recipe card to the correct user account.
Schedule	Initiate immediately after the start of milestone 2. Must be completed before the initiation of milestone 3.
Resources	N/A
Risk(s) and assumptions	Risk: the system does not add the recipe. Assumption: there will be recipes seeded into the system by default.
Roles and responsibilities	Primary Responsibilities: the "add" button serves the main role as it gives the user the ability to add more recipes to their personal database. The recipe database also serves a main role as that is the place the new recipe will be stored in. Secondary Responsibility: the initial seeding of recipes serves as a secondary role as it gives a default list of recipes for the user to work with.

Demo Hello World

Barcode Reading and Ingredient Data

```
from pyzbar.pyzbar import decode
from PIL import Image
import openfoodfacts

api = openfoodfacts.API()
api.product.get(code, fields=["code", "product_name"])

# Load an image with a barcode
image = Image.open('IMG_5274.png')  # replace .png with the image to open

# decodes image

decoded_objects = decode(image)

for obj in decoded_objects:
    print(f'Type: {obj.type}')
    print(f'Data: {obj.data.decode("utf-8")}')
```

Progress Matrix of Milestone 1

Task	Completion %	Tyler Son	Chris Nederhoed	David Tran	To do
1. Investigate tools	100%	33%	67%	0%	none
2. Hello World demos	75%	0%	75%	0%	Create a tesseract receipt scanner
3. Requirement Document	100%	5%	90%	5%	none
4. Design Document	100%	0%	0%	100%	none
5. Test Plan	100%	100%	0%	0%	none

Milestone 2

Task	Tyler Son	David Tran	Chris Nederhoed
1. Implement, test & demo *inventory management system*	*Demo Tagging items.*	*Demo List Creation and Sharing*	*Demo of Barcode and receipt scanning. Demo moving items to different storage locations.*
2. Implement, test & demo recipe organizing system and Sharing	Demo Recipe Card Creation from user imputed recipe. Demo Recipe Filters and Tags	Demo Recipe Sharing. Demo Review System	Demo Recipe Nutrition breakdown
3. Create a GUI			Create User Dashboard and supporting menus for the features being tested
4. Create the Database of recipes to be seeded by admins and later added to by users	50% - Implement the schema	50% - Seed the initial recipe DB	