KitchenSync
A Pantry and Recipe Companion

•••

Milestone 2

Milestone 2 Objectives

Inventory Management System

Recipe Organizing System

Basic GUI

Cloud DB for Recipe Sharing

Inventory System

Key Highlights

1. User-Friendly Entry Methods:
    - Manual: Simple, direct input.
    - Barcode: Swift product lookup using UPC codes.
    - Receipt Scanning: Extracts items and prices directly from store receipts.
2. Challenges with Receipt Scanning:
    - Data Variability: Some stores, like Walmart and Target, include UPCs, which simplify data extraction. Others, like Publix, only display product names and prices, requiring additional processing.
    - Image Accuracy:
        - Optimal Conditions: Clean, black-and-white scanned receipts yield ~90% accuracy.
        - User-Uploaded Photos: Accuracy drops to ~60% due to factors like lighting, receipt age, and photo quality.

Recipe Organizing System

Current Progress

1. Frontend Development:
   - User Entry Interface: Allows users to add and organize personal recipes.
   - Intuitive Design: Adapted for consistency with Inventory Management, enhancing ease of use.
2. Local Database:
   - In Progress: Setting up a local database to store user recipes for easy access and management.

Basic GUI

Development Process

1. GUI Design:
   - Used Scene Builder to streamline GUI creation based on initial wireframes.
   - Simplified export of interface elements and rapid iteration on UI design.
2. Project Structuring:
   - Initial Challenge: Sharing Java packages across team members was difficult due to file structure constraints.
   - Solution: Adopted Maven for project management, enabling easier exporting and team collaboration.

Current GUI Functionality

- Module-Specific Screens: Interfaces developed for current milestones, with additional screens planned as features expand.

Scaling Issues

- Screen Scaling: Certain menu items became inaccessible due to scaling challenges. Ongoing adjustments aim to enhance responsiveness across screen sizes.

Cloud DB for Recipe Sharing

Database Design

Our solution leverages AWS DynamoDB and AWS S3 to optimize storage and performance for high user concurrency.

1. DynamoDB:
   - Stores dynamic data such as recipe details, user information, and private data like individual inventories and non-shared recipes.
2. S3:
   - Stores static images of recipes to handle large file sizes effectively.
   - Designed to support scalability with thousands of users uploading images for multiple recipes.

Progress Matrix of Milestone 2

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%	70%	10%	20%	● Recipe Card Creation ● Recipe Searching with filters ● Recipe Sharing ● Review System ● Nutrition Breakdown ● Connect to Local DB for storage

Progress Matrix of Milestone 2

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

Milestone 3

Task	Tyler Son	David Tran	Chris Nederhoed
1.Finish Implementing Recipe Organizer + Inventory Manager	Continue work on the recipe database. Allow Users to save recipes from the cloud	Finish connecting local DB to the frontend components	Finish widget creation for different menus
2. Meal Planner		Create a shopping list script based on the ingredients needed based on what a user has	Building the meal planner with visual comps
3. New GUI Elements			Any GUI elements needed
4.Admin Backend	Connect DynamoDB and S3 for each recipe	Connect the front-end and back-end functions.	