Smart Grocery Management System

CISC. 4900

Contact Information

Group Members:

Sena Orucu (VC1C) orucusena21@gmail.com

Zarina Shevchenko (VC1A) zarinashevchenko@gmail.com

Supervisor (Working Professional):

Tugba Sezginsoy

tgbyldrmm96@gmail.com

Project Overview

Smart Grocery Management System is a mobile application designed to help users efficiently manage their grocery inventory by tracking expiration dates, suggesting recipes based on available ingredients, and alerting users of food recalls. The project aims to reduce food waste, save money, and enhance household food organization through an intuitive and automated system. The core functionalities include barcode scanning for easy item input, real-time expiration tracking, and integration with food safety APIs for recall alerts.

Our team, consisting of two members, divided responsibilities between authentication and inventory management. The goal is to create a seamless grocery tracking experience while learning full-stack mobile development.

Tools

Software & Frameworks:

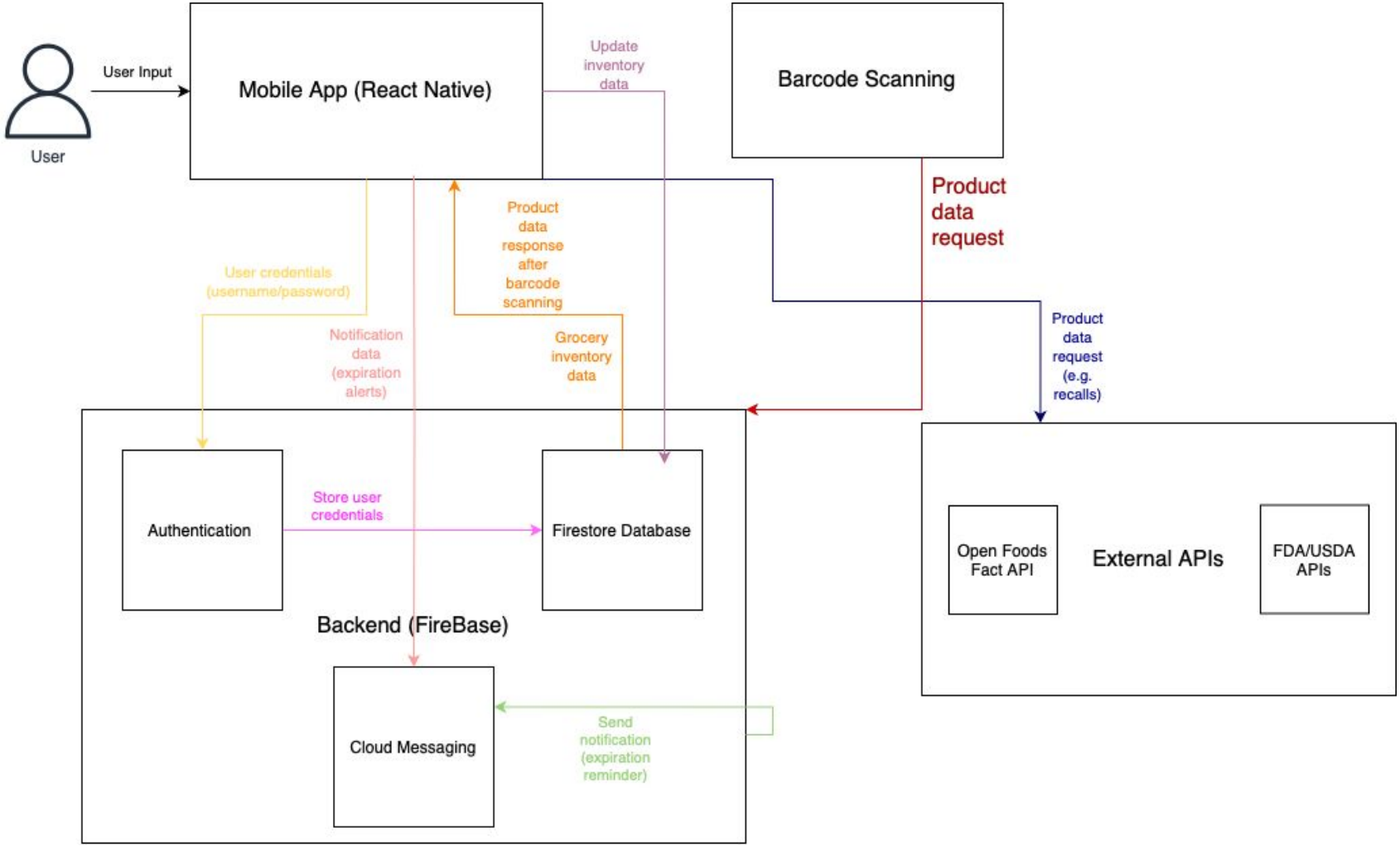
- React Native (Latest Version) for cross-platform mobile development
- Firebase Authentication & Firestore Database for backend services
- OpenAI API for potential NLP-related features
- FDA/USDA & Open Food Facts API for food recall data
- React Native Camera for barcode scanning

Hardware:

- Smartphones (iOS & Android) for testing
- Development Machines (Windows/Mac) for coding

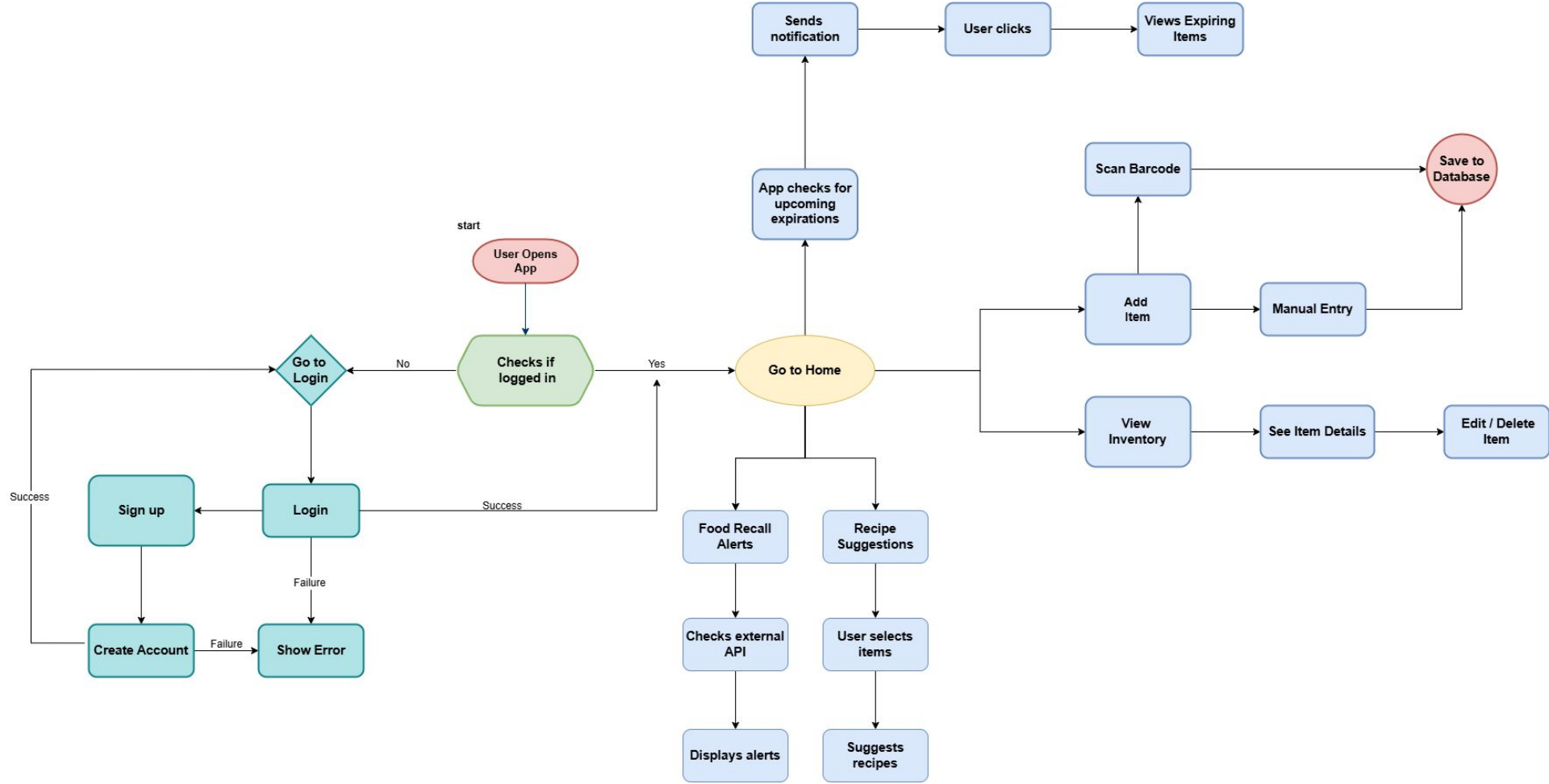
System Architecture Diagram

- Illustrates the interaction between the frontend (React Native app), backend (Firebase Firestore & Authentication), and external APIs.


User
User Input
Mobile App (React Native)
Update inventory data
Barcode Scanning
Product data request
Product data response after barcode scanning
User credentials (username/password)
Notification data (expiration alerts)
Grocery inventory data
Product data request (e.g. recalls)
Authentication
Store user credentials
Firestore Database
Backend (FireBase)
Cloud Messaging
Send notification (expiration reminder)
Open Foods Fact API
External APIs
FDA/USDA APIs

User Flow Diagram

- Visualizes the user's journey from login/signup, adding grocery items, setting expiration alerts, and receiving notifications.


Sends notification
User clicks
Views Expiring Items
App checks for upcoming expirations
Scan Barcode
Save to Database
start
User Opens App
Add Item
Manual Entry
Go to Login
No
Checks if logged in
Yes
Go to Home
View Inventory
See Item Details
Edit / Delete Item
Success
Sign up
Login
Success
Food Recall Alerts
Recipe Suggestions
Failure
Checks external API
User selects items
Create Account
Failure
Show Error
Displays alerts
Suggests recipes

Wireframes (Optional UI/UX Mockups)

- Showcases the app’s general layout and design. It includes the key screens and interactions, with some extra features planned for development if time permits.

Smart Grocery Management App

Your Pantry's Best Friend

Already have an account?

Login

First time here?

Sign up

Welcoming animation/gif

Sign Up animation/gif

Name

Enter email address

Last Name

Enter email address

Email

Enter email address

Password

Enter password

Sign Up

Already have an account log in!

Log In animation/gif

Email

Enter email address

Password

Enter password

Log In

Don't have an account sign up!

Profile Mascot

Welcome Back John!

Inventory (View / Add/ Edit)

Recipe Suggestions

View Recent Food Recalls

Settings (Notification settings/account settings etc.)

Saved Recipes

Grocery List

Log Out

Contact Us

View Inventory

Banana
Item: Banana
Purchase Date : 2/12/2025
Expiration Date: 2/22/2025
Labels: Fruit Room Temperature
Edit Delete

Tomatoes
Item: Tomatoes
Purchase Date : 2/12/2025
Expiration Date: 2/28/2025
Labels: Fruit Fridge
Edit Delete

Milk
Item: Milk
Purchase Date : 2/15/2025
Expiration Date: 2/20/2025
Labels: Dairy Fridge
Edit Delete

Chicken Breast
Item: Chicken Breast
Purchase Date : 2/12/2025
Expiration Date: 11/6/2025
Labels: Meat Frozen
Edit Delete

Egg
Item: Egg
Purchase Date : 2/12/2025
Expiration Date: 3/6/2025
Edit Delete

Go Back(Dashboard) | Add New Item Manually | Add New Item Scan Barcode

Add New Item

Item Name:

Purchase Date:

MM / DD / YYYY

Expiration Date:

MM / DD / YYYY

Category:

Select storage condition:

Back Add

Add New Item

Start Scanning

Back

Recipe Suggestions for Your Inventory....

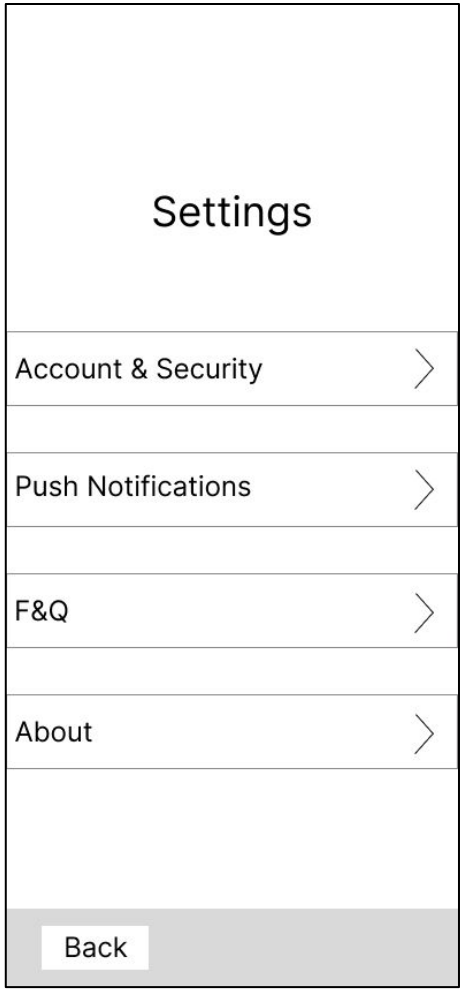
Banana Pancakes

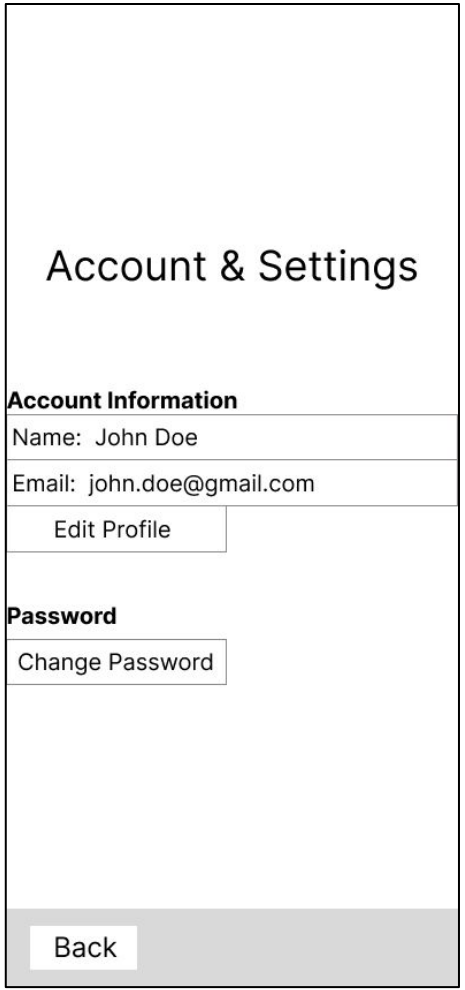
Creamy Garlic Butter Chicken

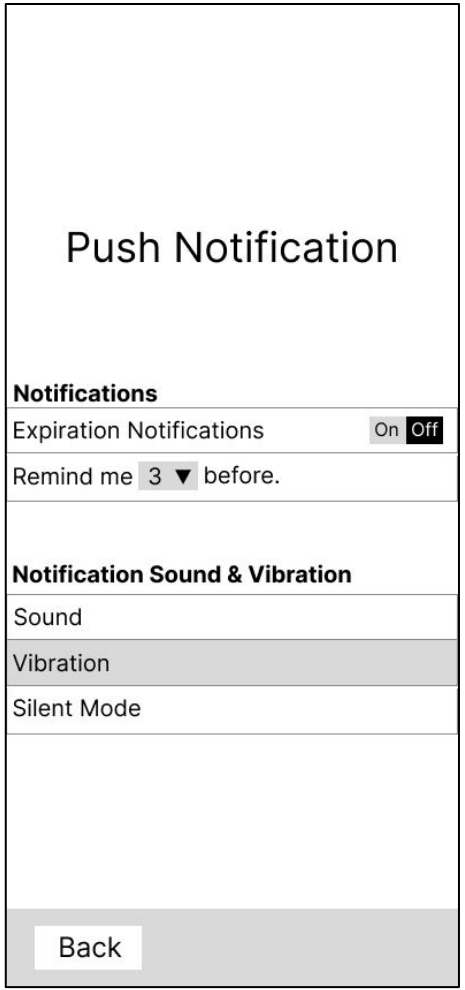
Double Chocolate Brownies

Back

Your Saved Recipes
Spicy Shrimp Tacos
Strawberry Shortcake
Chickpea & Spinach Curry
Avocado Toast
Grilled Veggie & Hummus Wrap
Fluffy Banana Pancakes
Classic New York Cheesecake
One-Pot Lemon Herb Pasta
Tropical Mango Smoothie
Back
Edit


Settings
Account & Security
Push Notifications
F&Q
About
Back


Account & Settings
Account Information
Name: John Doe
Email: john.doe@gmail.com
Edit Profile
Password
Change Password
Back


Push Notification
Notifications
Expiration Notifications
On
Off
Remind me 3 ▼ before.
Notification Sound & Vibration
Sound
Vibration
Silent Mode
Back

F&Q

Getting Started

How do I add items to my inventory? ▼

How does the barcode scanner work? ▼

Account & Security

How do I reset my password? ▼

How does the barcode scanner work? ▼

Troubleshooting

How do I reset my password? ▼

I'm not receiving notifications - how do i fix it? ▼

The app is not recognizing a barcode, what do I do? ▼

Back

About Smart Grocery Management App

About The App

Smart Grocery Management System is a mobile app that helps users track grocery inventory, manage expiration dates, get recipe suggestions, and receive food recall alerts. Designed to reduce food waste and save money, it features barcode scanning for easy input, automated reminders, and real-time food safety updates.

App Version	You're using version X.X.X
Developer Info	Made by Logi Team
Rate The App	Link to App Store/ Play Store

Back

Recent Food Recalls

Mauna Loa Milk Chocolate Covered Macadamias

Aleppo Tahini Sesame Paste

Fresh Direct Dark Chocolate Covered Pretzels

Shirakiku Snack foods-Corn Puffs

Paras Premium Golden Raisins

Pearl Milling Company Pancake and Waffle

Mauna Loa Milk Chocolate Covered Macadamias

Fresh Direct Dark Chocolate Covered Pretzels

Shirakiku Snack foods-Corn Puffs

Pearl Milling Company Pancake and

Back

Contact Us

Customer Support

smart.management.help@gmail.com

Social Media

Instagram

LinkedIn

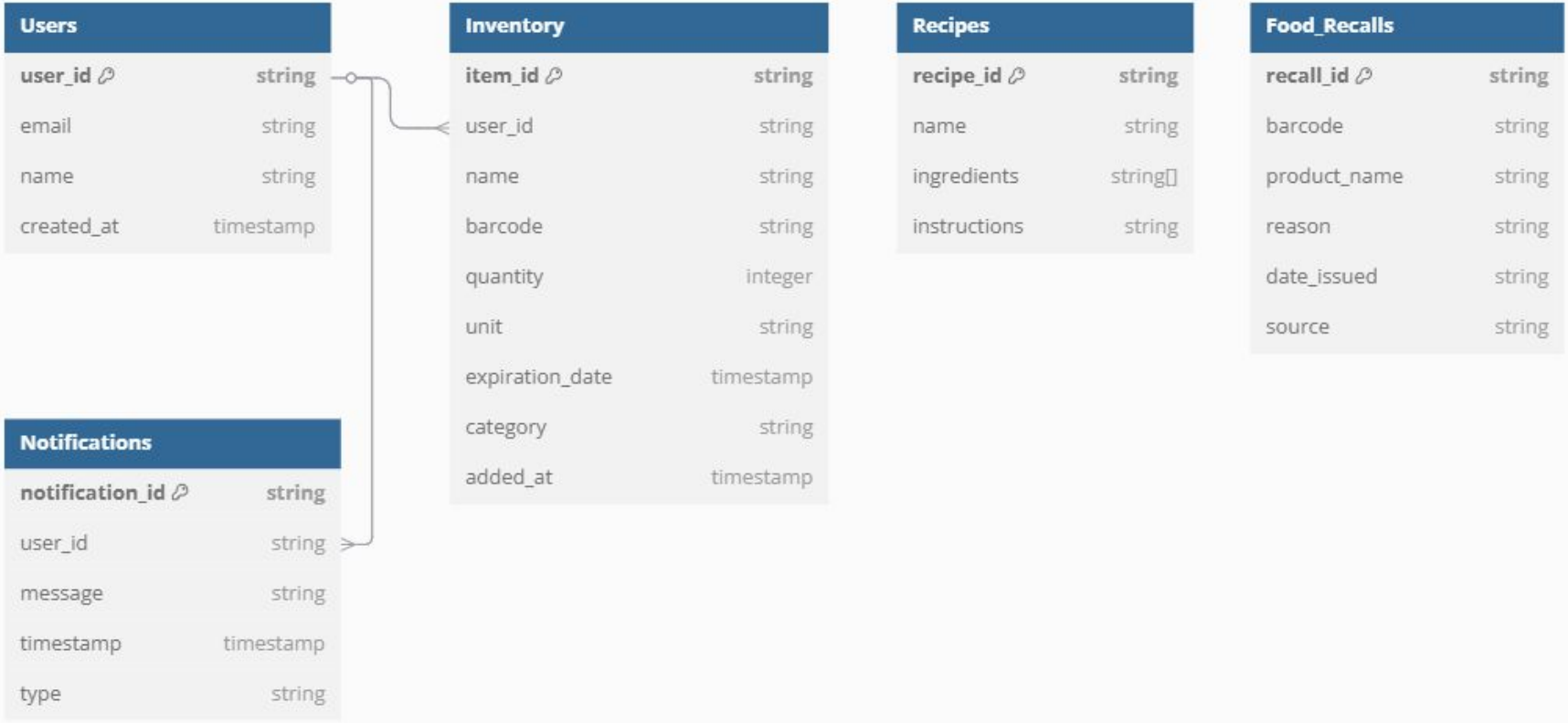
TikTok

Since 2025

Back

Database Schema Diagram

- Shows Firestore collections and document structure (e.g., Users, Inventory, Expiration Date, Recipes).


Users
user_id string
email string
name string
created_at timestamp
Inventory
item_id string
user_id string
name string
barcode string
quantity integer
unit string
expiration_date timestamp
category string
added_at timestamp
Recipes
recipe_id string
name string
ingredients string[]
instructions string
Food_Recalls
recall_id string
barcode string
product_name string
reason string
date_issued string
source string
Notifications
notification_id string
user_id string
message string
timestamp timestamp
type string

Tentative Schedule

WEEK	TASK	ESTIMATED TIME
Feb 10 - Feb 16	Setup Development Environment, Learn React Native & Firebase	1 week
Feb 17 - Feb 23	Finalize MVP Features, Setup Development Environment, Learn React Native & Firebase	1 week
Feb 24 - Mar 9	Core Features Development	2 week
Mar 10 - Mar 23	Core Features & Testing & Presentation Prep	2 week
Mar 24 - Apr 6	Feature Refinement & Demo Preparation	2 week
Apr 7 - Apr 20	Final Testing, Debugging	2 week
Apr 21 - May 5	Final Demo Recording & Submission	2 week
May 6 - May 12	Final Review & Submission	1 week

Data Sources

- **Inventory Data:** User-inputted grocery items with expiration dates.
- **Food Recall Data:** FDA/USDA APIs & Open Food Facts database.
- **Recipe Suggestions:** Basic rule-based system with OpenAI API for NLP-enhanced recommendations.

Use Cases

1. **User Authentication**

    - **Input:** Email/password signup or Google Authentication.
    - **Process:** Firebase Authentication verifies credentials.
    - **Output:** User gets access to the app dashboard.

2. **Inventory Management**

    - **Input:** User scans barcode or manually enters item details.
    - **Process:** System saves item data to Firestore.
    - **Output:** Item appears in inventory list with expiry tracking.

Use Cases

3. **Expiration Alerts**

- **Input:** System checks stored items for approaching expiration dates.
- **Process:** Sends push notification reminders to the user.
- **Output:** User receives alerts and can take necessary action.

4. **Edit/Delete Inventory Items**

- **Input:** User taps item from list
- **Process:** Pre-filled form opens → User edits or deletes
- **Output:** Firestore updates or removes the item

Team Roles & Workflow

We split the project between two collaborators:

- **Zarina:** Inventory features, Firestore CRUD, expiration alerts, UI/UX decisions
- **Sena:** Authentication flow, Firebase Auth, Google login, Barcode Scanning

We collaborated through GitHub Project Board, weekly check-ins, and code reviews to ensure cohesion and progress.

System Design

- We designed a modular system using React Native for the frontend and Firebase Firestore as the backend.
- The system integrates multiple external APIs for enhanced functionality:
  - FDA/USDA & Open Food Facts for recall alerts
  - OpenAI for future recipe suggestions
- We also created diagrams such as:
  - System Architecture Diagram to show the flow of data between app, database, and APIs
  - ER Diagram to illustrate how inventory and users are related
- Our division of labor made it easier to maintain a clean architecture with separated concerns.

Programming Skills

- Together, we wrote mobile app code using JavaScript and React Native.
- Our codebase includes:
  - **CRUD operations** for inventory data with Firestore
  - **Authentication logic** using Firebase Auth
  - **Date picker integration** and conditional rendering of alerts
- Debugging sessions and real device testing helped us polish cross-platform issues.
- Code was version-controlled through GitHub with regular commits and issue tracking.

Algorithm Efficiency

- We evaluated the efficiency of our expiration alert system:
  - Filtered upcoming expirations based on date differences
  - Minimized unnecessary reads by narrowing Firestore queries
- Future optimization plans include:
  - Indexing fields for faster Firestore queries
  - Debouncing user input for search/filter features
- Our priority was balancing responsiveness with low battery and data usage on mobile devices.

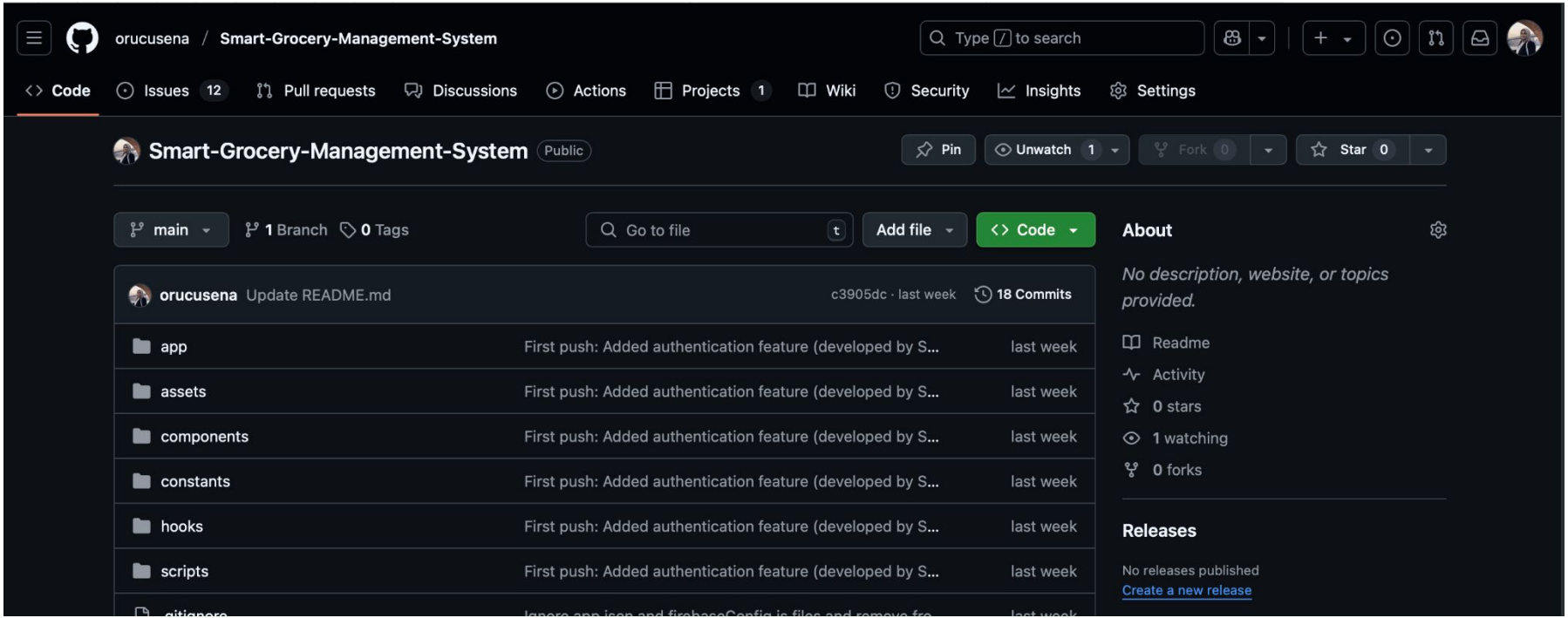
Integration of CS Concepts

- This project involved combining knowledge from multiple CS domains:
  - **Databases:** Firestore schema design and ER modeling
  - **Software Engineering:** Version control, modular structure, backlog/task management
  - **Mobile App Development:** Platform-specific issues and UI layout for small screens
  - **User-Centered Design:** Focused on making the app intuitive and efficient
- Working across these areas helped us deliver a cohesive, well-rounded product.

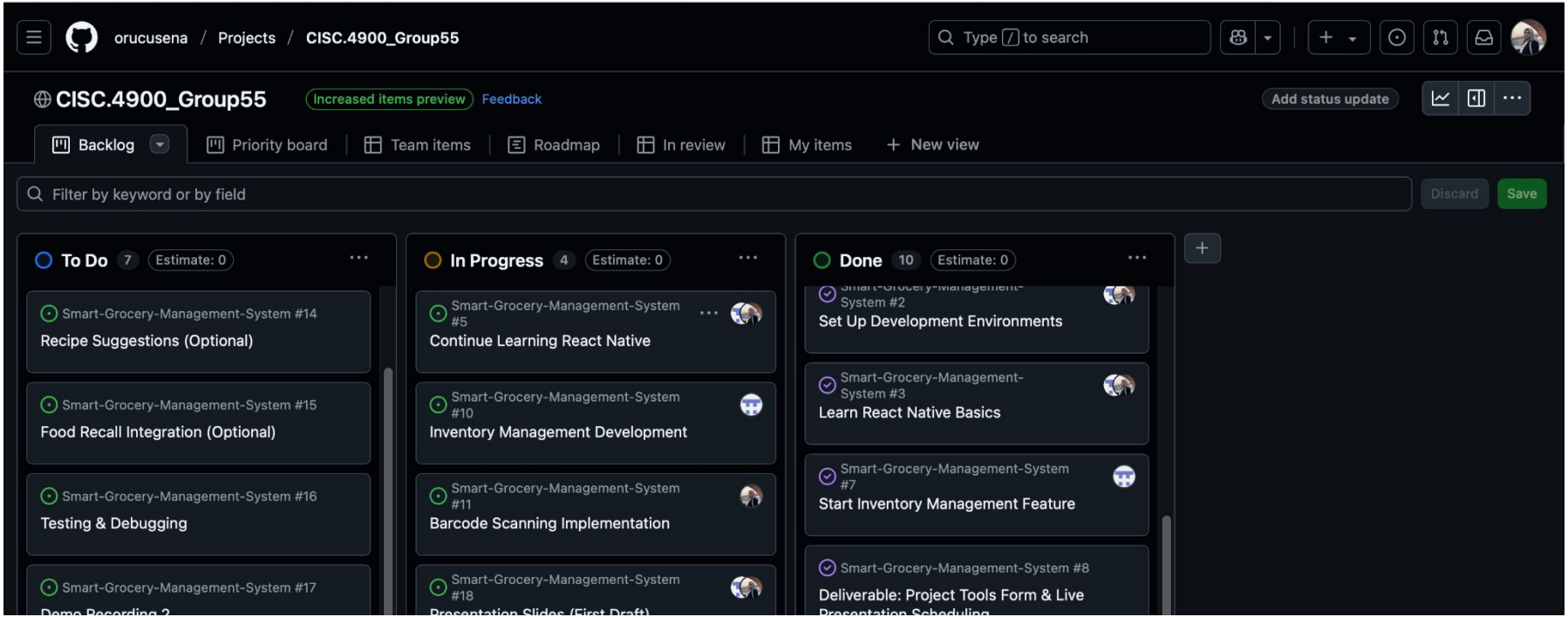
Professional Practice

- We practiced team communication, scope management, and planning:
  - Weekly team meetings to discuss blockers and progress
  - Used GitHub Projects to organize our sprints and features
- Documentation included time logs, code comments, and presentation prep
- Our workflow followed principles of **Agile** development in a simplified, academic setting

GitHub Repository



Project Management Board



GitHub

GitHub Repository:

https://github.com/orucusena/Smart-Grocery-Management-System

Project Management Board:

https://github.com/users/orucusena/projects/1