

*BEST B4*

GROUP MEMBER: HEET SHAH, MAHIR PATEL, JEET  
PATEL

# INTRODUCTION

CS422 / 522

# TEAM ROLES

## MEMBERS

Heet Shah

Mahir Patel

Jeet Patel

## RESPONSIBILITIES

Developed authentication system, implemented Core Data model, managed navigation flow between screens.

Designed UI/UX in Figma, handled edge case logic, researched OCR integration strategies.

Researched APIs for expiration tracking, managed project scope and timeline, coordinated team planning.



# APP IDEA & TARGET USERS

## **Problems Identified**

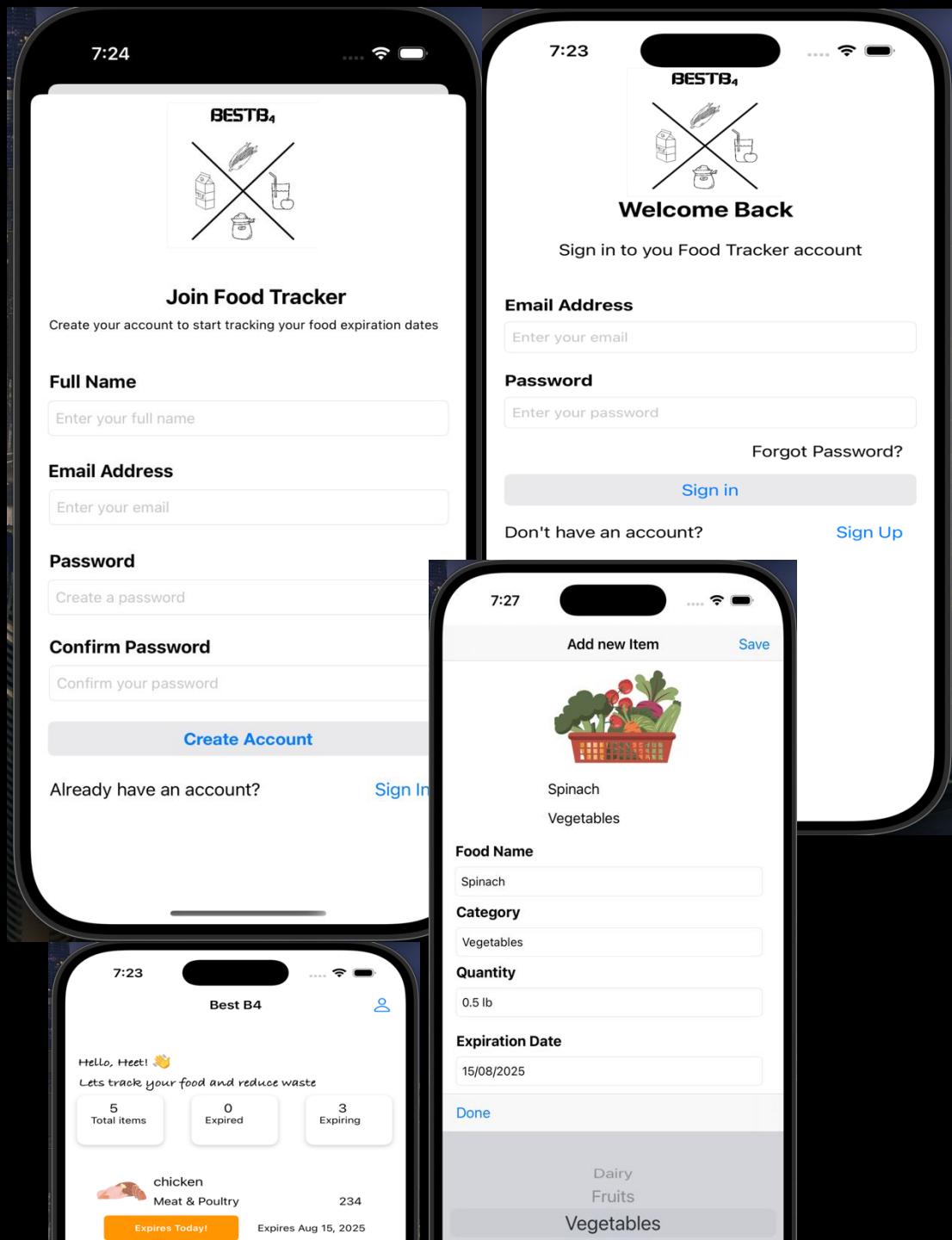
Many users forget expiration dates, leading to food waste and financial loss

## **Proposed Solution**

A mobile app that tracks food items, categorizes them, and sends alerts before expiration

## **Target Audience**

Students, families, and busy individuals who want to manage their pantry efficiently



# CORE FEATURES

- User Authentication: Secure login and sign-up with persistent sessions using Core Data and UserDefaults
- Product Entry: Users can input item name, category, quantity, and expiration date
- Dashboard Display: Scrollable list of items with color-coded status indicators based on expiration proximity
- Notification System: Scheduled alerts that notify users before items expire.

# DESIGN PROCESS

## ***FIGMA PROTOTYPING:***

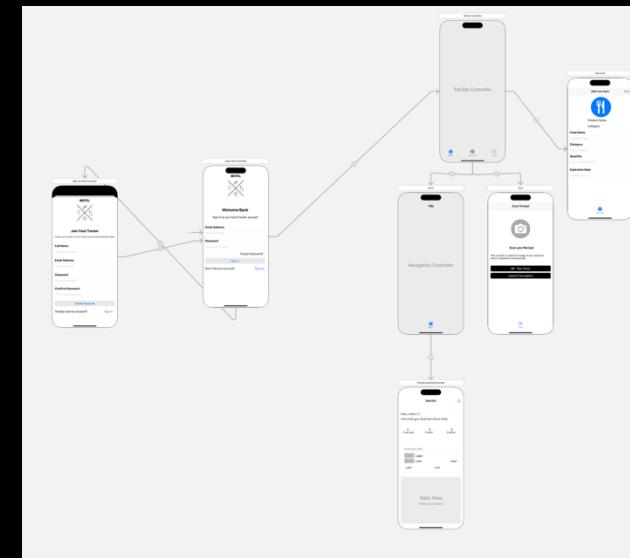
Initial UI layouts and flow diagrams created to visualize user experience

## ***Category & Date Logic:***

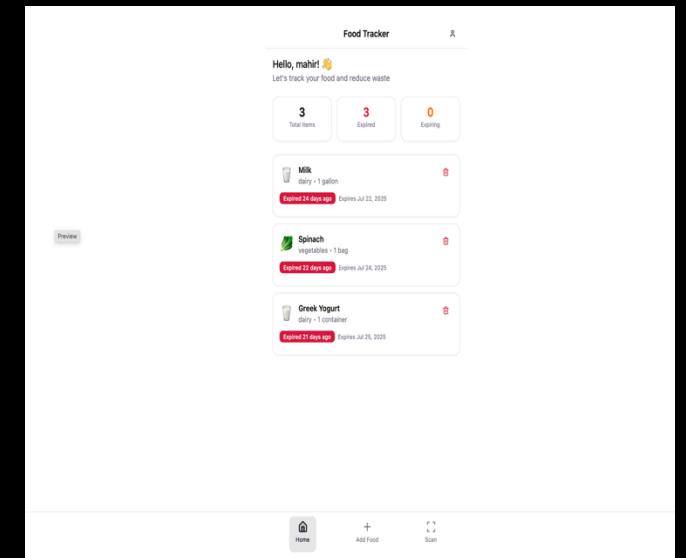
Implemented logic to prevent invalid date selection and dynamically update UI based on category

## ***Accessibility & Responsiveness:***

Ensured the app works across devices and screen sizes, with readable fonts and intuitive controls



UI flow diagram showing navigation between screens



Figma mockups of key screens

# TECHNICAL CHALLENGES

BEST B4

**Navigation Bugs:** Default modal transitions caused UI issues; required custom navigation controller.

**Core Data Integration:** Initial failures due to misconfigured entity relationships and code generation.

**OCR Complexity:** No single library met all needs; required multi-library research and planning.

**API Limitations:** Free APIs lacked reliability or had restrictive terms; forced alternative logic development

foodExpiryTracker

ScanViewController

func requestPhotoLibraryAccessAndOpenLibrary() {

switch photoLibraryAuthorizationStatus {

case .notDetermined:

PHPhotoLibrary.requestAuthorization { [weak self] in

DispatchQueue.main.async { // Ensure UI

updates on the main thread

if status == .authorized {

print("Photo Library permission granted (after request). Opening library.")

- self?.openImagePickerController(sourceType: .photoLibrary)

Cannot infer contextual base in reference to member 'photoLibrary'

Value of type 'ScanViewController' has no member 'openImagePickerController'

(for: "Photo Library")

}

}

case .authorized:

print("Photo Library permission already

Image picking canceled.

Take photo tapped! Checking camera permission

Failed to find a valid fallback video configuration.

[CAMCaptureEngine] Received a session runtime error notification : Error Domain=AVFoundationErrorDomain Code=-11800 "The operation could not be completed" UserInfo={NSLocalizedFailureReason=An unknown error occurred (-12782), NSLocalizedDescription=The operation could not be completed, NSUnderlyingError=0x60000cc3570 {Error Domain=NSOSStatusErrorDomain Code=-12782 "(null)"}}

[CAMCaptureEngine] Performing recovery from error: Error Domain=AVFoundationErrorDomain Code=-11800 "The operation could not be completed" UserInfo={NSLocalizedFailureReason=An unknown error occurred (-12782), NSLocalizedDescription=The operation could not be completed, NSUnderlyingError=0x60000cc3570 {Error Domain=NSOSStatusErrorDomain Code=-12782 "(null)"}}

[CAMCaptureEngine] Attempting to recover from a session runtime error by restarting the AVCaptureSession...

Image picking canceled.

Message from debugger: killed

# SOLUTIONS & FIXES

## **Custom Navigation Controller:**

Replaced default transitions for smoother screen flow.

## **Core Data Restructuring:**

Refactored entity relationships and ensured proper data persistence.

## **OCR Framework Skeleton:**

Created a modular plan for future OCR integration using VisionKit and Firebase ML.

## **Expiration Logic:**

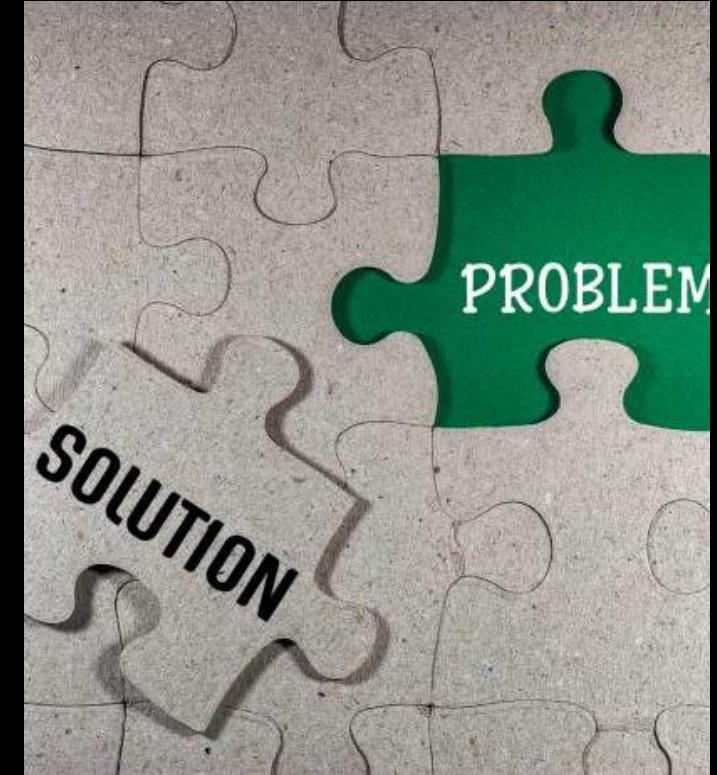
Built internal logic to calculate urgency without relying on external APIs

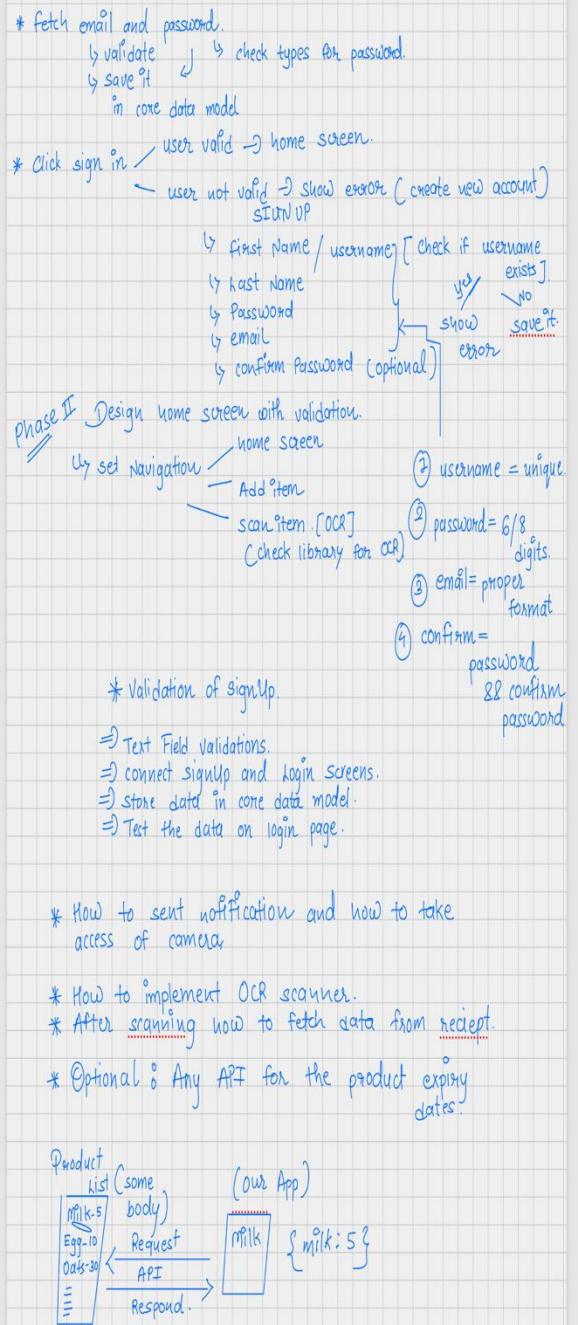
## **Edge Case Handling:**

Added checks for invalid dates, missing fields, and duplicate entries to ensure clean data and smooth user flow.

## **Notifications:**

Scheduled alerts using UNUserNotificationCenter, with unique IDs for easy cancellation and user permission handling.

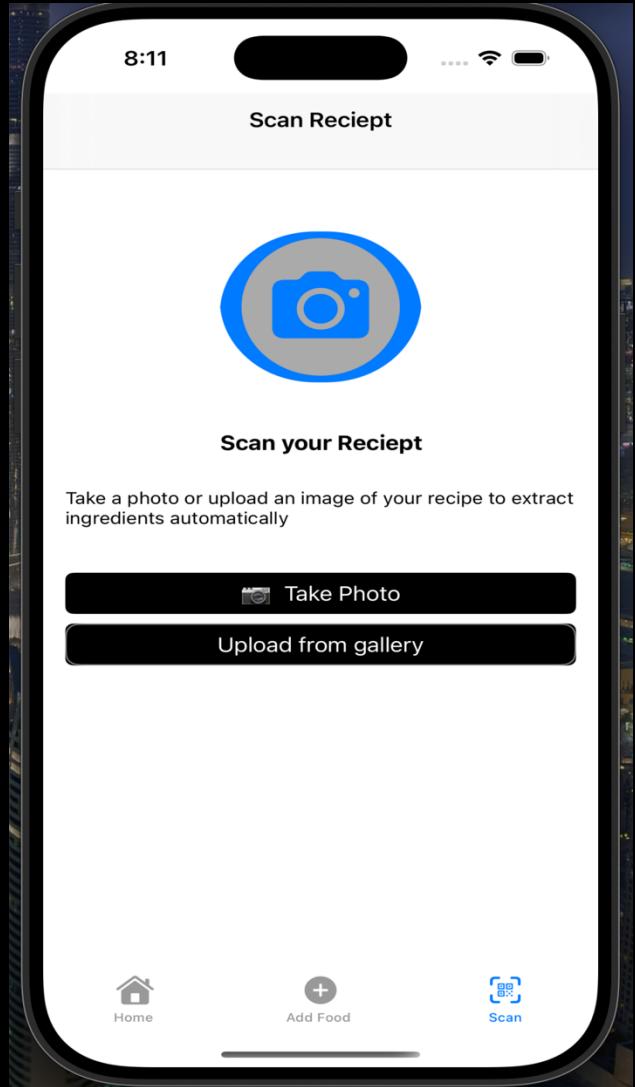




# PROJECT EVOLUTION

- **Initial Scope:** Included OCR, barcode scanning, and predictive analytics
- **Final MVP:** Focused on core tracking, notifications, and UI polish.
- **Team Coordination:** Adjusted scope based on team size and timeline; maintained progress through task board and weekly syncs

# FUTURE PLANS



## ***Barcode Scanning:***

Enable faster item entry using device camera

## ***OCR Input:***

Extract product details from receipts or packaging

## ***Predictive Analytics:***

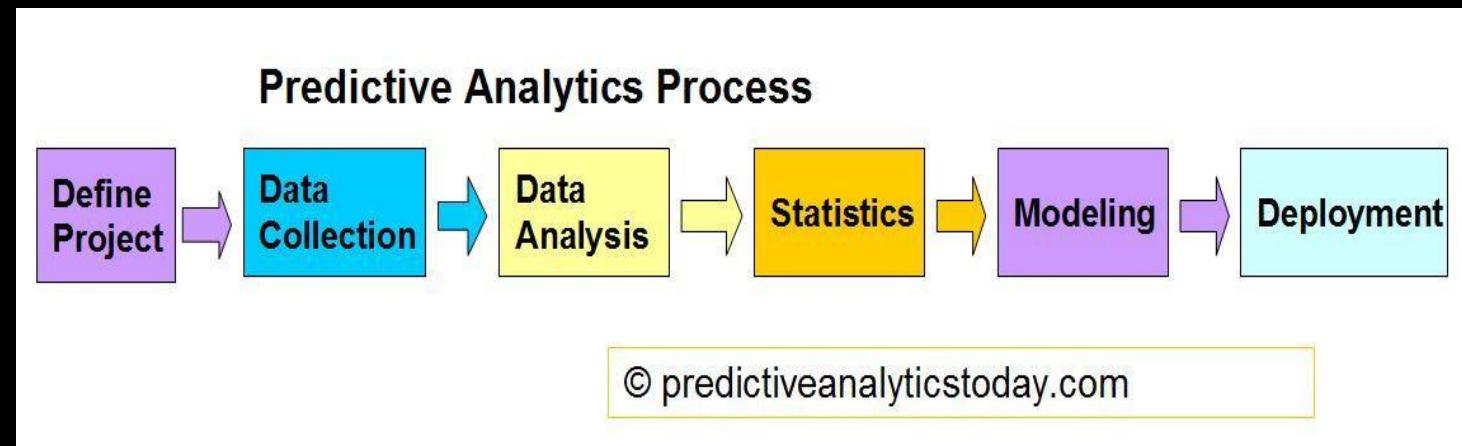
Suggest usage patterns and waste reduction tips

## ***Cloud Sync:***

Allow multi-device access and backup

## ***AI Integration:***

User will get AI based recommendation of the recipes.





# DEMO & CLOSING

## LIVE DEMO!

## SUMMARY:

Best B4 helps users track food expiry, reduce waste, and stay notified, all through a clean, intuitive app.

We tackled key challenges with smart fixes in navigation, data handling, and notification logic.

Our team collaborated smoothly and built a solid foundation for future features like OCR and cloud sync.

THANK YOU FOR YOUR TIME AND FEEDBACK. WE'RE PROUD OF WHAT WE'VE BUILT AND EXCITED TO KEEP IMPROVING BEST B4!

THANK YOU!

⋮

# QUESTIONS?

We'd love to hear your thoughts!

BEST B4

Got any questions, feedback, or suggestions?

We're happy to dive deeper into any part of the app, our process, or future plans.



o