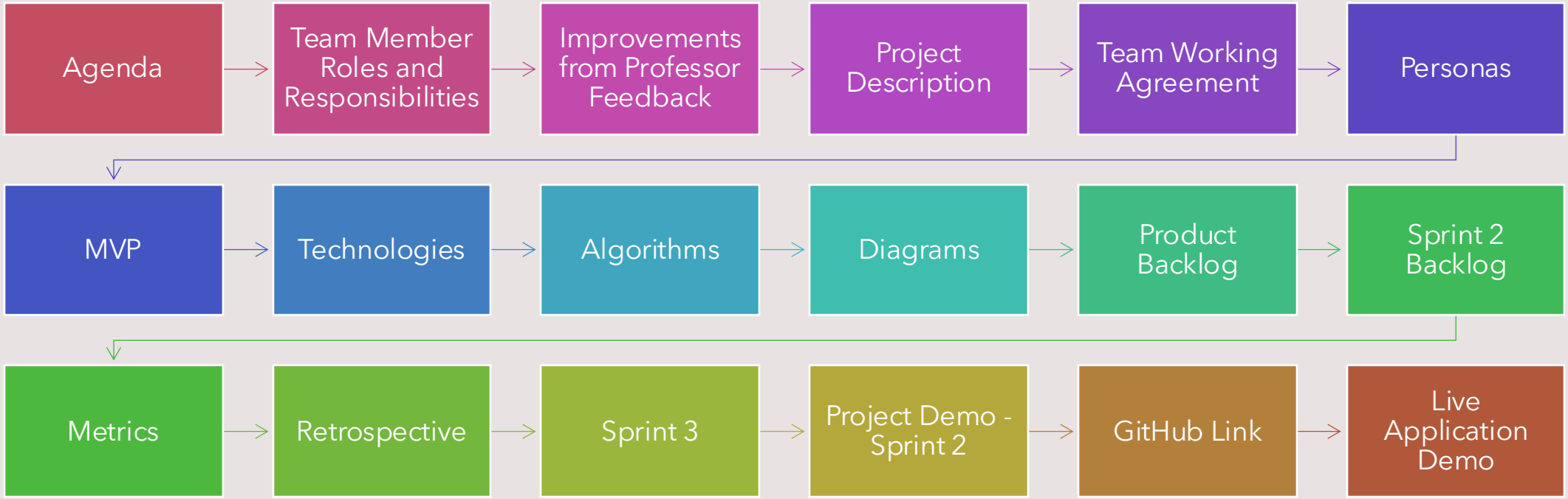
CalorieMate

By Team 3 - The Innovators

Agenda


Agenda
Team Member Roles and Responsibilities
Improvements from Professor Feedback
Project Description
Team Working Agreement
Personas
MVP
Technologies
Algorithms
Diagrams
Product Backlog
Sprint 2 Backlog
Metrics
Retrospective
Sprint 3
Project Demo - Sprint 2
GitHub Link
Live Application Demo

Team Member Roles and Responsibilities

Ranjitha Durgasi – Frontend Developer- Team Leader

Nikhitha Reddy Nallanagula – Frontend Developer

Harshitha Korapati Murali – Backend Developer

Team Member Roles and Responsibilities

Sai Bhargav Ram Koduru – Backend Developer

Saikumar Gone – Machine Learning Engineer- Scrum Master

Kariveda Vikranth Reddy – Machine Learning Engineer

Improvements

- Fixing Stories Completed

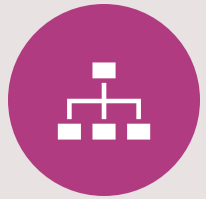
Project Description

Project Name:	*CalorieMate*
Team:	The Innovators
Project Description:	**For** health-conscious users **who** want to easily track their food intake and manage their diet**,** **the** CalorieMate app **is a** smart, AI-powered solution **that** automatically identifies food from photos, calculates nutrition, and offers personalized meal suggestions**.** **Unlike** traditional calorie-tracking apps that require manual data entry**,** **our application** provides a fast, hassle-free way to track calories and nutrition, saving time and improving accuracy**.**
Benefit Outcomes:	• **Time Savings:** Users can quickly track their meals through photo analysis without manual data entry. • **Improved Accuracy:** AI provides precise estimates for calories and nutrients, reducing guesswork. • **Health Goal Support:** Personalized diet plans help users stay on track with their weight loss, muscle gain, or overall wellness goals. • **Convenience:** Barcode scanning for packaged food makes nutritional tracking fast and effortless. • **Consistency:** Easy tracking and user-friendly features increase engagement, encouraging long-term healthy habits.
Github Link:	https://github.com/htmw/2025S-The-Innovators/wiki

Team Working Agreement

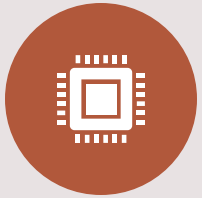
Communication: Lets make sure to share updates and have discussions on platforms like Slack. Make sure to have check ins, such, as weekly stand up meetings or sprint reviews to keep everyone on the same page and working together smoothly. Cooperation;

Team Meetings & Collaboration: To ensure effective meetings, the team should agree on key points, such as scheduling regular meetings twice a week at 6-8pm via virtual (zoom).Our main agenda is to make all team members work and make submissions on time. As a team we should help one another by sharing each other thoughts and make sure all perspectives are heard before deciding.



Roles and Responsibilities: Each team member has the task of finishing assignments based on their designated roles in frontend development, backend development, and machine learning. Coordinate with teams to seamlessly incorporate various functionalities.

Decision-Making: Decisions will be collaboratively reached in team gatherings. In situations where there are differences in opinions or views arise among us team members, we go for majority votes.



Deliverables: Each sprint should lead to producing the outcomes like presentations updates, to technical papers and wikis. Make sure to review and test all your work before submitting it for consideration.

Retrospectives: After each sprint is completed it's important to have a session to review the successes and areas for improvement as well as to outline specific steps, for the upcoming sprint.

Agreed by:

- Ranjitha Durgasi
- Nikhitha Reddy Nallanagula
- Harshitha Korapati Murali
- Sai Bhargav Ram Koduru
- Saikumar Gone
- Kariveda Vikranth Reddy

Persona

Sneha (Fitness Freak)

•**Age:** 28

•**Occupation:** Marketing Specialist

•**Goals:** To build muscle and maintain a healthy diet.

•**Challenges:** Finds it time-consuming to manually log every meal and calculate macros (proteins, carbs, fats).

•**How CalorieMate Helps:** Sneha can take photos of her meals to instantly track nutrients, saving time and keeping her fitness plan on track.

Persona

Surya cook (Busy Professional)

•**Age:** 35

•**Occupation:** Software Engineer

•**Goals:** To lose weight and adopt healthier eating habits.

•**Challenges:** Limited time to research or record calorie information due to his demanding work schedule.

•**How CalorieMate Helps:** The app's photo recognition and barcode scanner allow cooking to quickly track meals, making calorie management fit seamlessly into his day.

Persona

Kruti Patel (Health-Conscious Parent)

•**Age:** 30

•**Occupation:** Teacher

•**Goals:** To maintain a balanced diet and teach her family healthy eating habits.

•**Challenges:** Difficulties estimating nutritional content of home-cooked meals.

•**How CalorieMate Helps:** Kruti can easily track home-cooked meals using photo recognition and adjust portions to get accurate nutritional insights, helping her plan meals for the entire family.

MVP

- Food identification from photos using AI
- Automatic calorie and nutrition calculation
- Barcode scanning for packaged food
- Daily calorie and nutrient tracking dashboard
- Personalized meal suggestions based on user goals

Technologies

Frontend: Expo (React Native)

Used to build the mobile app for both Android and iOS platforms, ensuring a smooth and consistent user experience across devices.

Technologies

Backend: FastAPI, MongoDB, AWS

- **FastAPI:** Manages API communication between the mobile app, machine learning models, and database.
- **MongoDB:** Stores user data such as meals, nutritional logs, and preferences.
- **AWS:** Provides cloud infrastructure for hosting the app, including APIs, databases, and machine learning services, ensuring scalability and reliability.

Technologies

Hugging Face

Machine Learning: PyTorch, Hugging Face

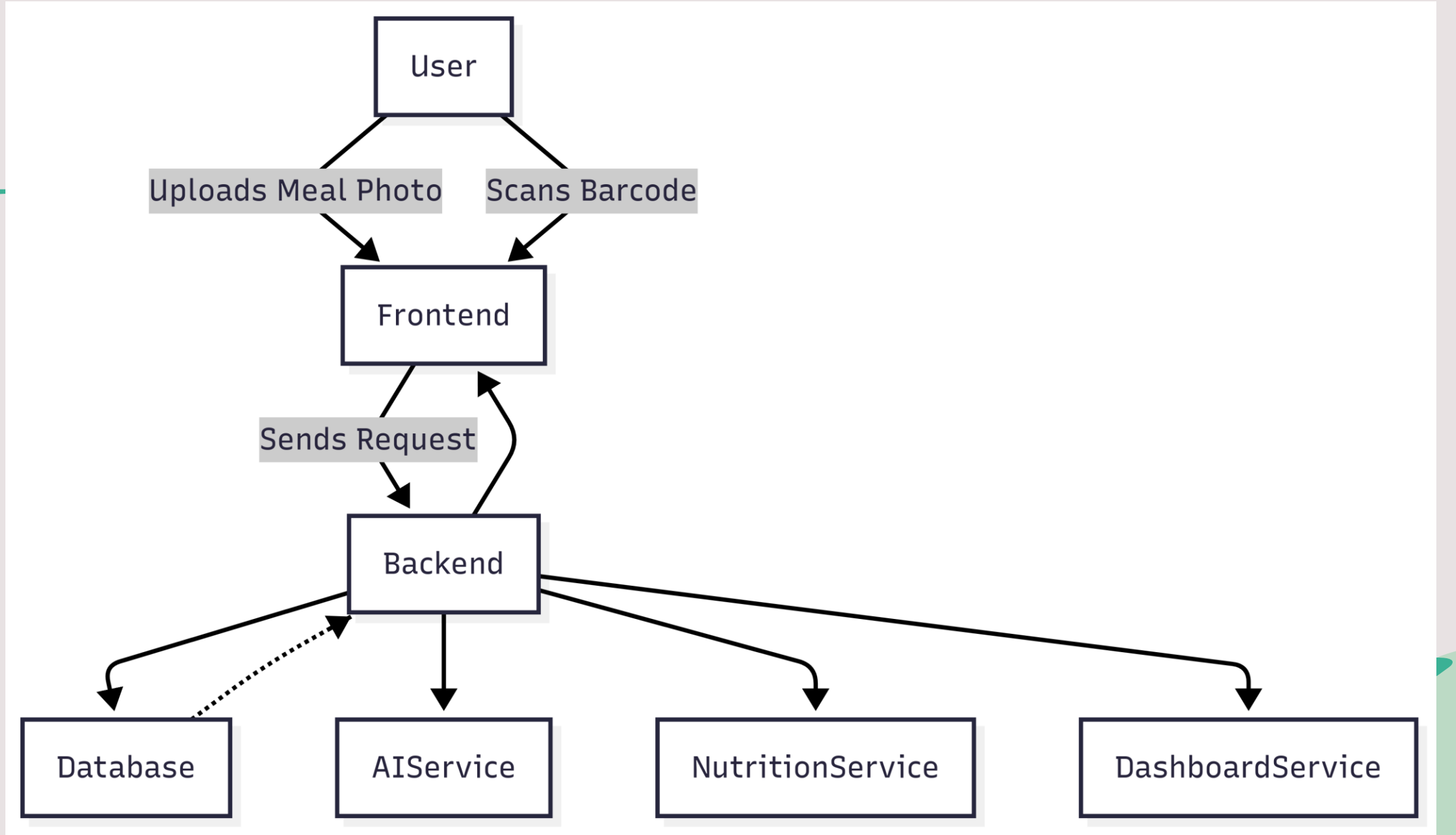
- **PyTorch:** Used to build and deploy AI models for food recognition and nutrition estimation.
- **Hugging Face:** Supplies pre-trained vision-language models (VLM) to improve image recognition accuracy, reducing development time.

PyTorch

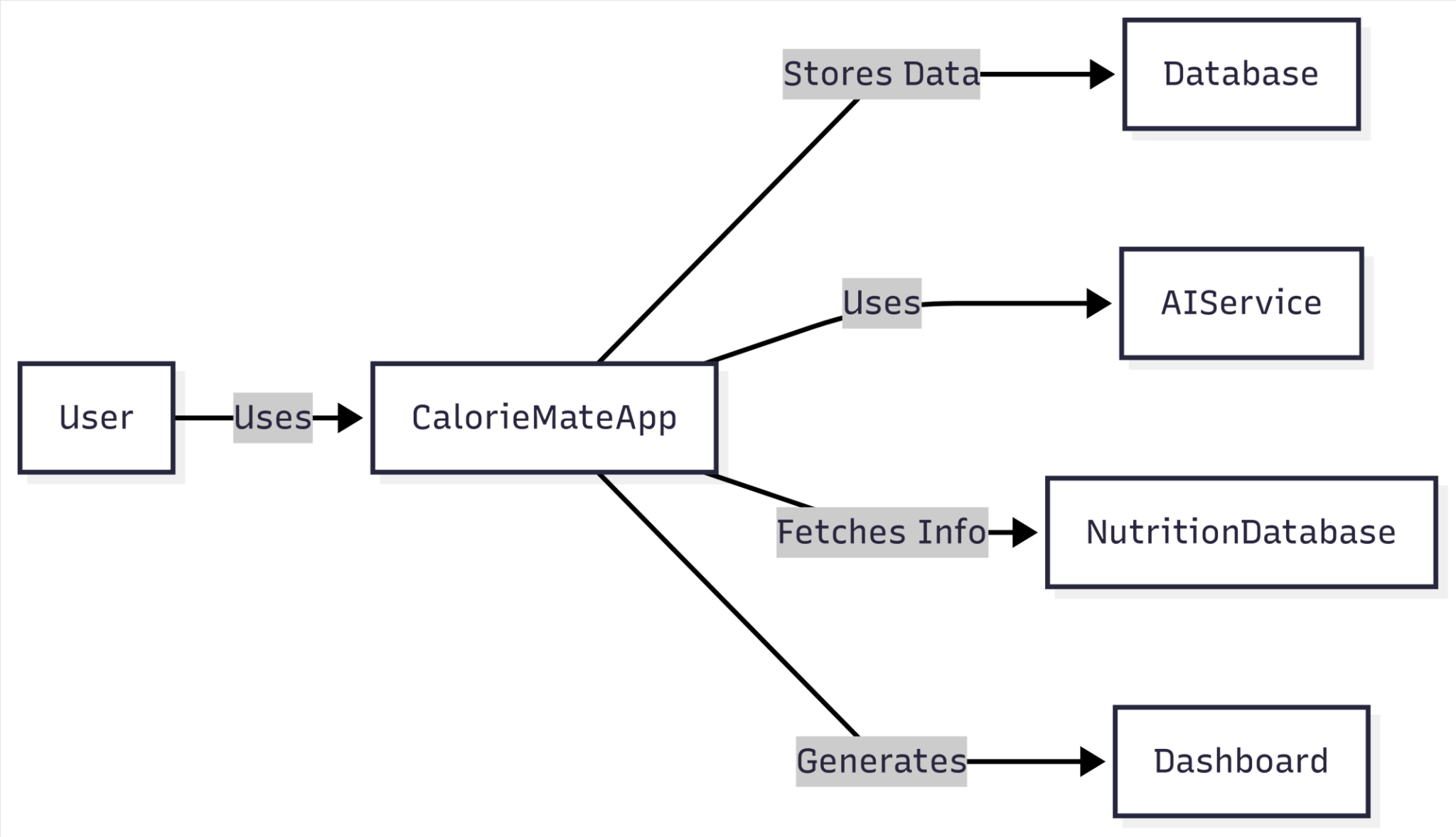
Algorithm

Our application utilizes some important algorithms to automate and enhance the process of calorie and nutrition tracking. The food image recognition algorithm employs the **InstructBLIP** model to process meal images, identify food items, and recognize them with high accuracy. After recognizing the foods, a nutritional estimation algorithm estimates the calories, proteins, carbohydrates, and fats based on nutritional databases and approximate portion sizes.

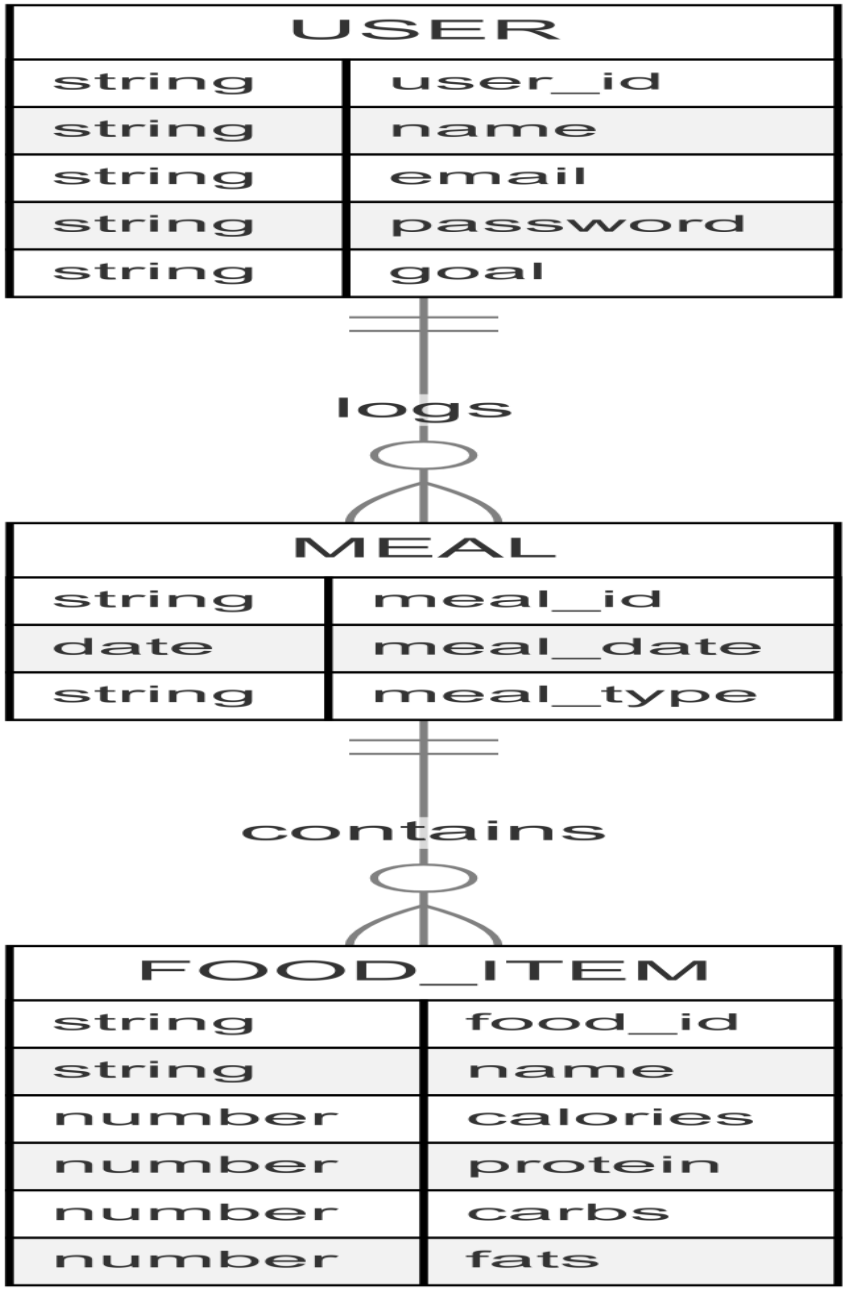
Architecture Diagram



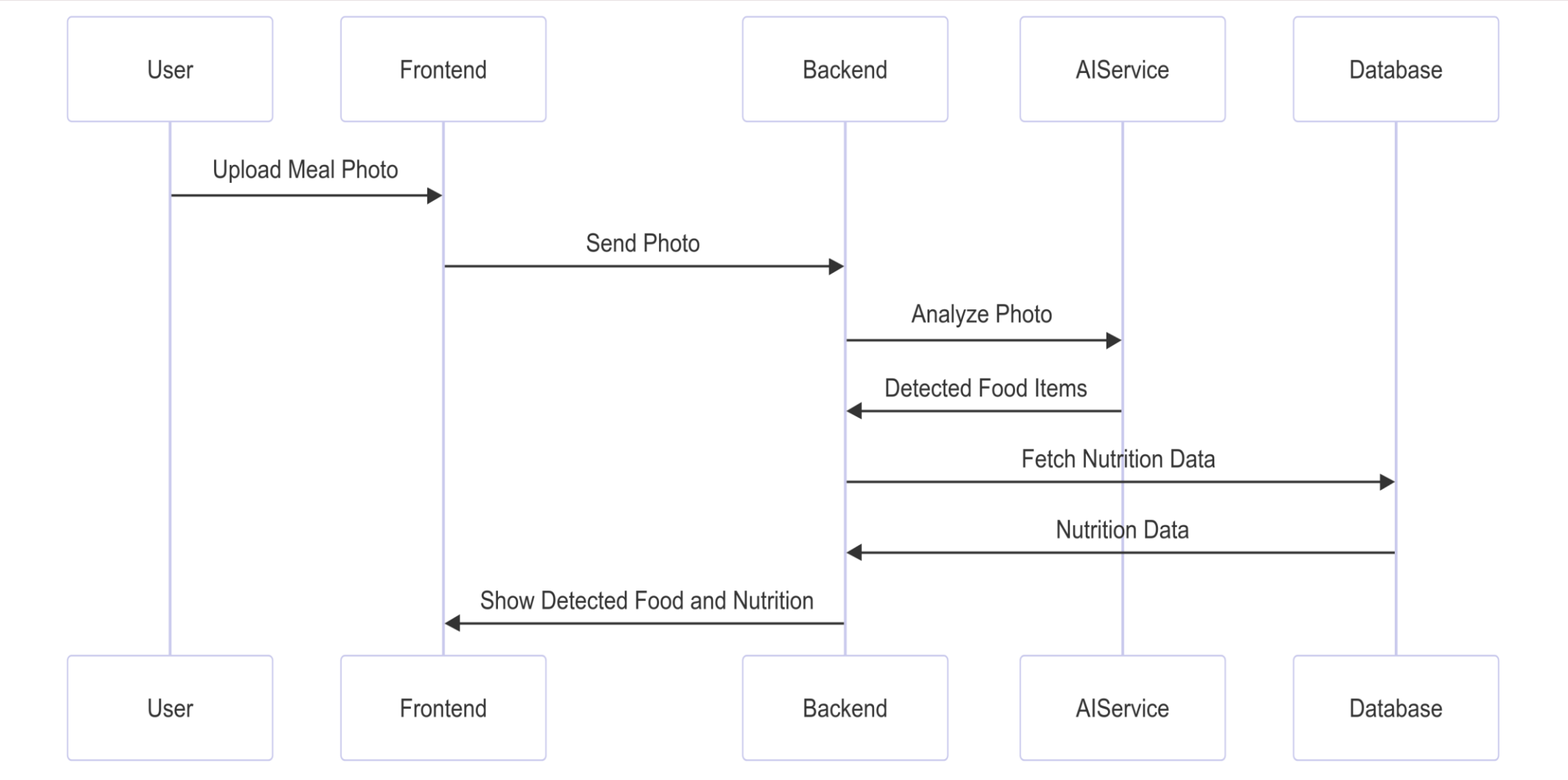
Context diagram



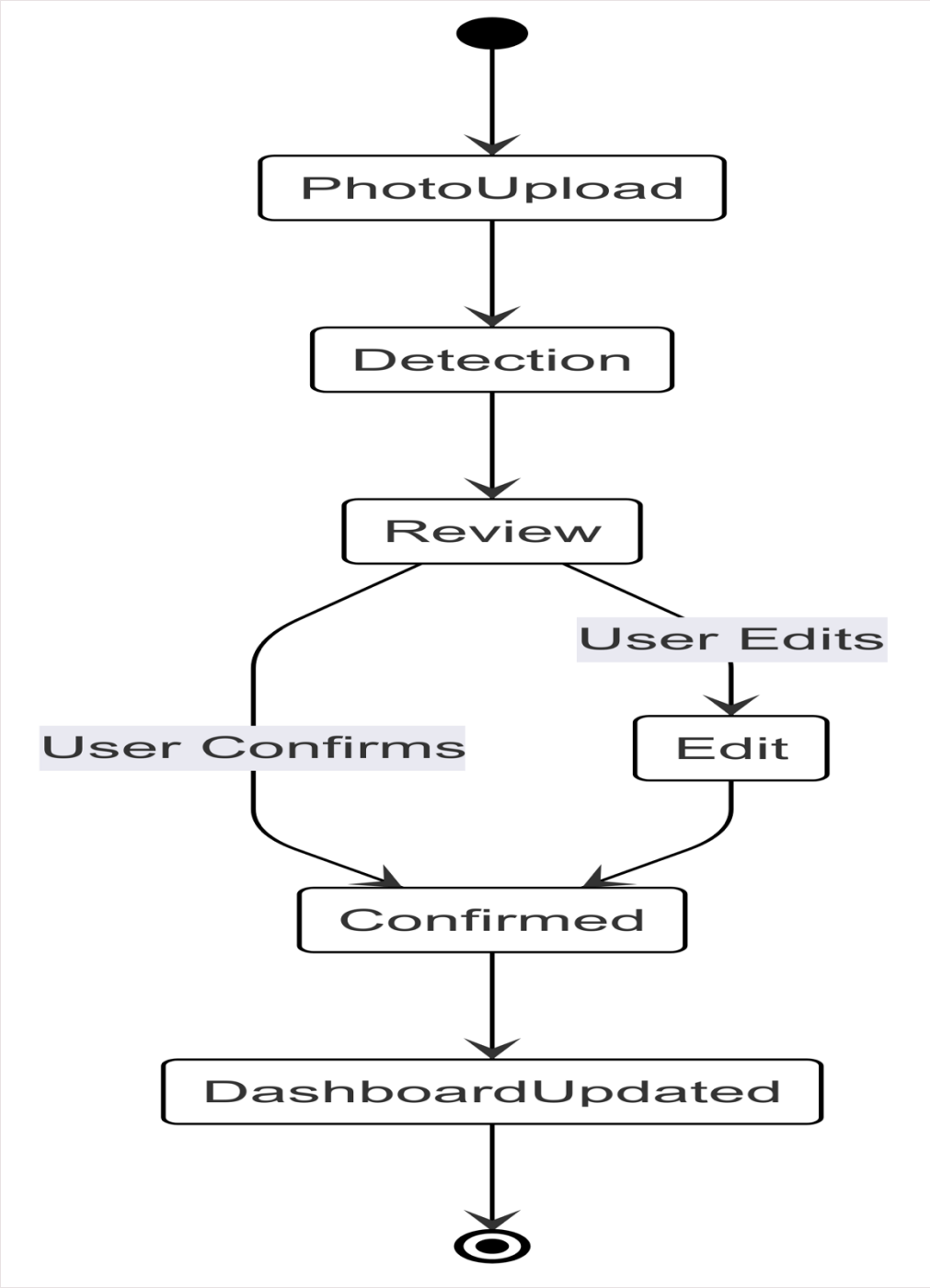
ER Diagram


USER
string user_id
string name
string email
string password
string goal
logs
MEAL
string meal_id
date meal_date
string meal_type
contains
FOOD_ITEM
string food_id
string name
number calories
number protein
number carbs
number fats

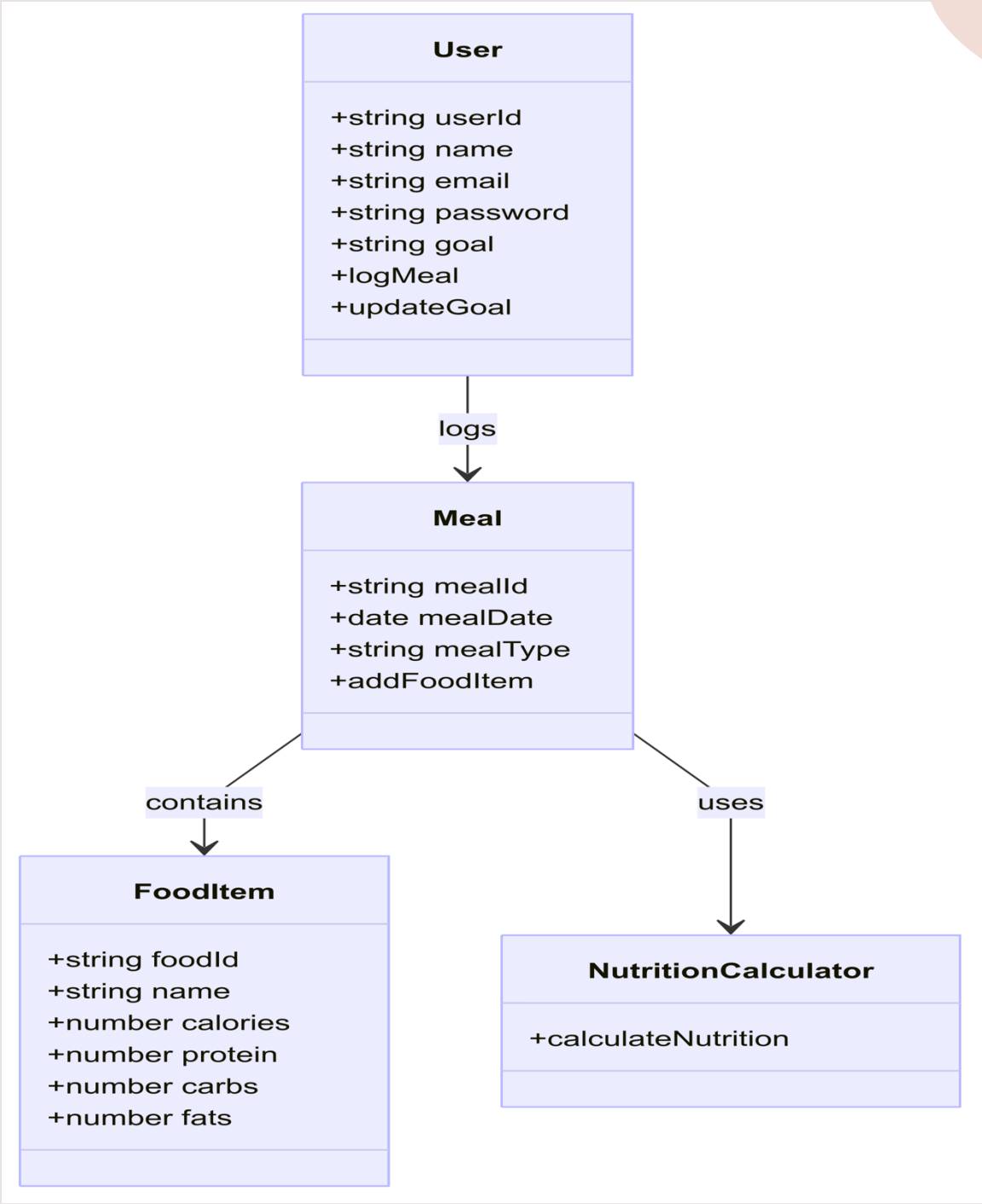
Sequence Diagram



State Diagram



Class Diagram



Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
US1	As a user, I want to create an account and log in so that my data is saved.	User can sign up and log in Error shown for wrong credentials Logout works	Authentication	3	1
US2	As a user, I want to upload a meal photo or take an image from the camera so that I can manually enter my food details.	Users can upload an image or take a photo. Users manually input food items. Error shown if no items are added.	Photo Upload	5	1
US3	As a user, I want to manually enter food items so that I can track my meals.	Users can enter food items manually. Users can input portion sizes. Data is saved to the meal log after confirmation.	Food Entry	3	1
US4	As a user, I want to edit food items so that my meal log is accurate.	Users can modify the food name and portion size. Changes are saved automatically to the meal log.	Manual Food Editing	3	1
US5	As a user, I want to see a daily dashboard so that I can track my calories and nutrients.	Dashboard shows calories consumed, remaining budget Macro breakdown shown	Dashboard	3	1

Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
TS1	Set up user authentication backend	Database created for users Signup, login, logout APIs ready Passwords securely stored	Authentication Backend	3	1
TS2	Set up photo upload UI	Users can upload images from the gallery or take a photo.	Photo Upload	5	1
TS3	Build UI for manual food entry	Users can enter food names and portion sizes manually.	Food Entry	3	1
TS4	Build UI for food editing	Users can edit food items and portion sizes.	Manual Food Editing	3	1
TS5	Build daily dashboard UI & backend	Dashboard reads meal log Calculates totals Shows simple charts	Dashboard Backend + UI	3	1

Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
US6	As a user, I want to scan barcodes to log packaged food so that I can log faster.	Barcode scanning works Finds product in database Adds to meal log	Barcode Scanning	5	2
US7	As a user, I want to set my health goal so that my calorie target is personalized.	Goal can be set Calorie target updates Goal can be changed later	Goal Setting	2	2
US8	As a user, I want to see my daily macro breakdown so that I know if my meals are balanced.	Dashboard shows grams & percentages for macros Updates after each meal log	Dashboard	2	2
US9	As a user, I want to edit or delete past meal entries so that I can fix mistakes.	Past meals listed Edit or delete works Dashboard recalculates after change	Meal Log Management	3	2
US10	As a user, I want to view my full meal history so that I can review my past eating habits.	All past meals listed by date Clicking shows full details	Meal History	2	2

Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
TS6	Build barcode scanning backend and food database	Food database includes packaged foods Barcode scanning works on mobile Adds directly to log	Barcode Scanner Backend	5	2
TS7	Set up goal setting feature	Goal stored in user profile Goal adjusts calorie target in dashboard	Goal Backend	2	2
TS8	Add macro breakdown logic to dashboard	Dashboard calculates macros from meals Updates automatically	Dashboard Logic	2	2
TS9	Build meal history with edit/delete	Meal logs stored by date Edit and delete APIs work Dashboard updates after changes	Meal Log Backend	3	2
TS10	Build full meal history page	Meals listed by date Clicking opens meal details	Meal History UI	2	2

Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
US11	As a user, I want to see calories and nutrients after entering my meal so that I know what I'm eating.	Users enter food items manually. Calories, protein, carbs, and fats are displayed. Data is saved to the meal log.	Calorie & Nutrition Display	5	3
US12	As a user, I want to see a weekly progress report so that I can track my long-term trends.	Weekly calories graph shown Weekly macro average shown	Progress Report	3	3
US13	As a user, I want a settings page to manage my preferences so that I can customize the app.	User can set units (metric/imperial) User can change goal Changes saved immediately	Settings	2	3
TS11	Build AI-based calorie & nutrition calculation	Users enter food items manually. AI calculates calories, protein, carbs, and fats. Data is saved to the meal log.	AI-Based Nutrition Calculation	5	3
TS12	Build weekly progress backend & UI	Backend calculates 7-day summary UI shows graph & macro summary	Progress Report	3	3

Product Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
TS13	Build settings page backend & UI	Preferences saved per user Changes reflected in dashboard Supports units and goals	Settings Backend + UI	2	3

Sprint 2 Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points
US6	As a user, I want to scan barcodes to log packaged food so that I can log faster.	Barcode scanning works Finds product in database Adds to meal log	Barcode Scanning	5
US7	As a user, I want to set my health goal so that my calorie target is personalized.	Goal can be set Calorie target updates Goal can be changed later	Goal Setting	2
US8	As a user, I want to see my daily macro breakdown so that I know if my meals are balanced.	Dashboard shows grams & percentages for macros Updates after each meal log	Dashboard	2
US9	As a user, I want to edit or delete past meal entries so that I can fix mistakes.	Past meals listed Edit or delete works Dashboard recalculates after change	Meal Log Management	3
US10	As a user, I want to view my full meal history so that I can review my past eating habits.	All past meals listed by date Clicking shows full details	Meal History	2

Sprint 2 Backlog

ID	User Stories	Acceptance Criteria	Feature	Story Points
TS6	Build barcode scanning backend and food database	Food database includes packaged foods Barcode scanning works on mobile Adds directly to log	Barcode Scanner Backend	5
TS7	Set up goal setting feature	Goal stored in user profile Goal adjusts calorie target in dashboard	Goal Backend	2
TS8	Add macro breakdown logic to dashboard	Dashboard calculates macros from meals Updates automatically	Dashboard Logic	2
TS9	Build meal history with edit/delete	Meal logs stored by date Edit and delete APIs work Dashboard updates after changes	Meal Log Backend	3
TS10	Build full meal history page	Meals listed by date Clicking opens meal details	Meal History UI	2

Test Cases

US ID	Test Case ID	Test Case Description	Steps to Execute	Expected Result	Result (Pass/Fail)
US6	TC6.1	Launch barcode scanner	1. Open the app 2. Go to meal log 3. Tap barcode icon	Scanner activates camera	Pass
US6	TC6.2	Find product by barcode	1. Open scanner 2. Scan known barcode 3. Wait for result	Product details are retrieved and displayed	Pass
US6	TC6.3	Add scanned product to log	1. Scan product 2. Tap "Add to meal" 3. Save entry	Product added to current meal log	Pass
US6	TC6.4	Handle invalid barcode	1. Open scanner 2. Scan unknown barcode 3. Wait for result	Message shown: “Product not found”	Pass

Test Cases

US ID	Test Case ID	Test Case Description	Steps to Execute	Expected Result	Result (Pass/Fail)
US7	TC7.1	Set health goal	1. Open profile 2. Select goal 3. Save changes	Goal is saved to profile	Pass
US7	TC7.2	Calorie target updates	1. Set goal 2. Go to dashboard 3. Check calorie target	Calorie target reflects selected goal	Pass
US7	TC7.3	Change goal later	1. Open profile 2. Edit goal 3. Save changes	Updated goal reflected in dashboard	Pass
US8	TC8.1	View macro breakdown	1. Open app 2. Go to dashboard 3. Locate macro section	Macros shown in grams and percentages	Pass

Test Cases

US ID	Test Case ID	Test Case Description	Steps to Execute	Expected Result	Result (Pass/Fail)
US8	TC8.2	Update macros after meal log	1. Log meal 2. Save entry 3. Check dashboard	Macros update with new meal	Pass
US9	TC9.1	View past meal entries	1. Open app 2. Go to meal history 3. View listed meals by date	Meals listed chronologically	Pass
US9	TC9.2	Edit meal entry	1. Tap meal 2. Edit details 3. Save changes	Meal updates with edited values	Pass
US9	TC9.3	Delete meal entry	1. Tap meal 2. Press delete 3. Confirm deletion	Meal is removed from log	Pass

Test Cases

US ID	Test Case ID	Test Case Description	Steps to Execute	Expected Result	Result (Pass/Fail)
US9	TC9.4	Recalculate after change	1. Edit/delete meal 2. Return to dashboard 3. Check totals	Dashboard recalculates updated values	Pass
US10	TC10.1	List full meal history	1. Open app 2. Go to meal history tab 3. Scroll through list	All past meals displayed by date	Pass
US10	TC10.2	View meal detail	1. Tap on a date 2. Select meal 3. View components	Full details of meal shown	Pass

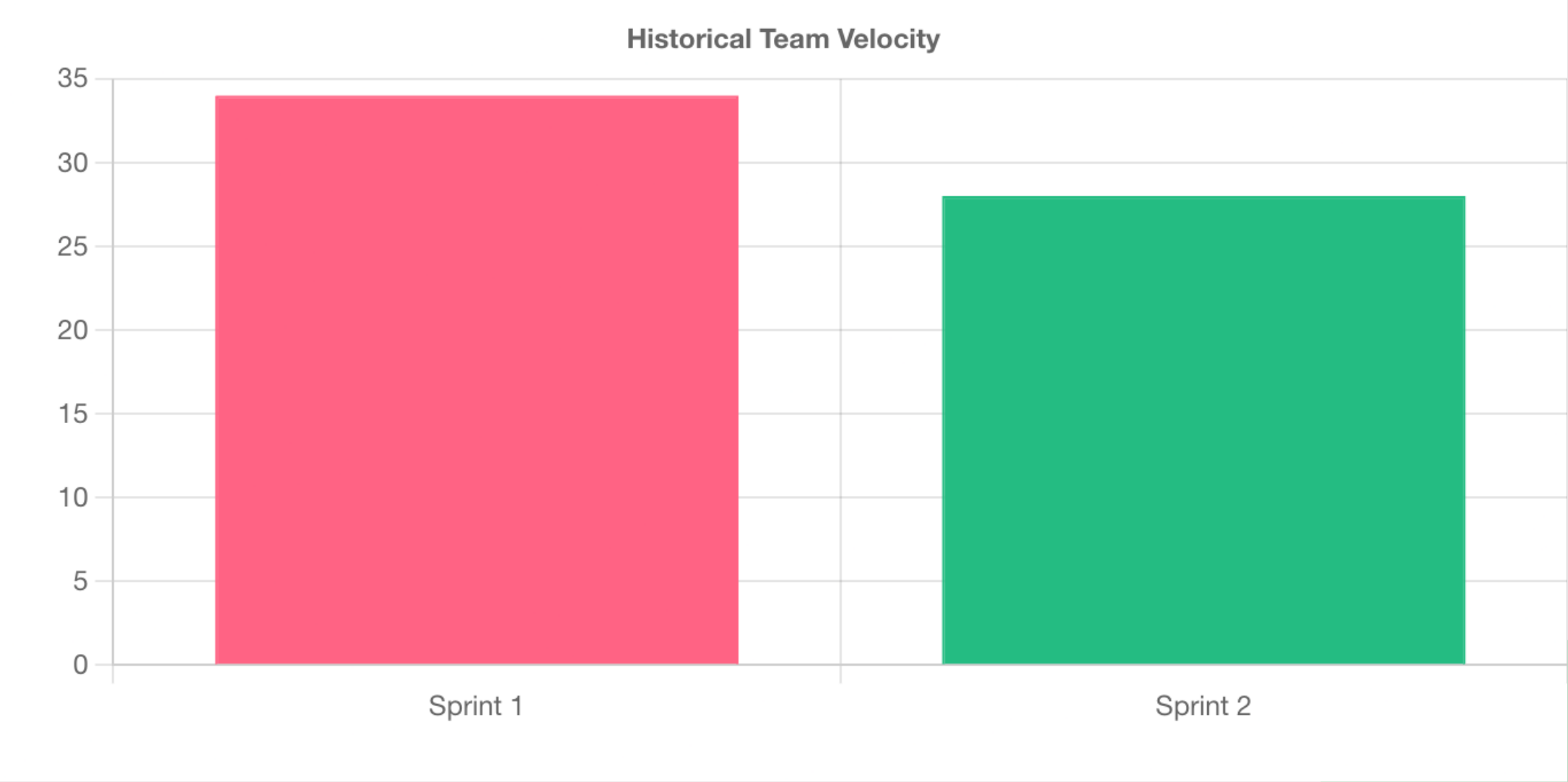
Completed Stories Sprint 2

ID	Story Description
US6	As a user, I want to scan barcodes to log packaged food so that I can log faster.
US7	As a user, I want to set my health goal so that my calorie target is personalized.
US8	As a user, I want to see my daily macro breakdown so that I know if my meals are balanced.
US9	As a user, I want to edit or delete past meal entries so that I can fix mistakes.
US10	As a user, I want to view my full meal history so that I can review my past eating habits.
TS6	Build barcode scanning backend and food database
TS7	Set up goal setting feature
TS8	Add macro breakdown logic to dashboard
TS9	Build meal history with edit/delete
TS10	Build full meal history page

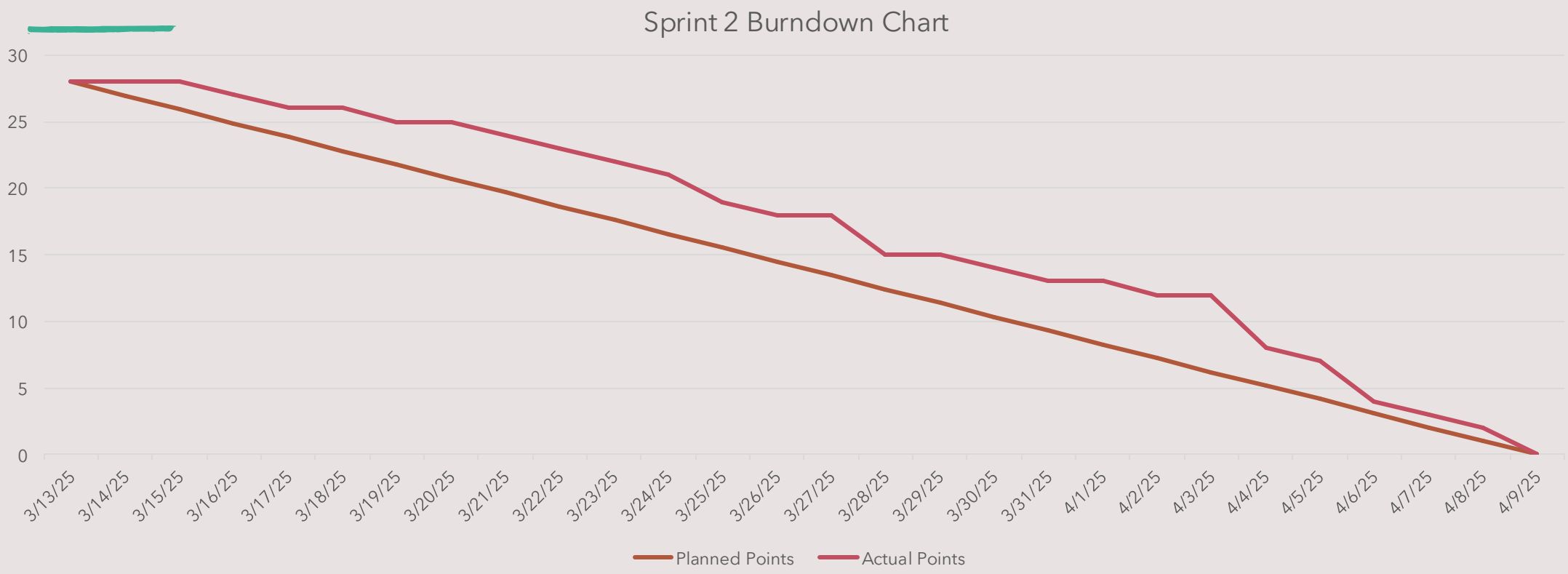
Team Velocity

- **Team Velocity:** 28 story points
- All committed stories (28 points) were completed successfully.

Historical Team Velocity



Burndown Chart



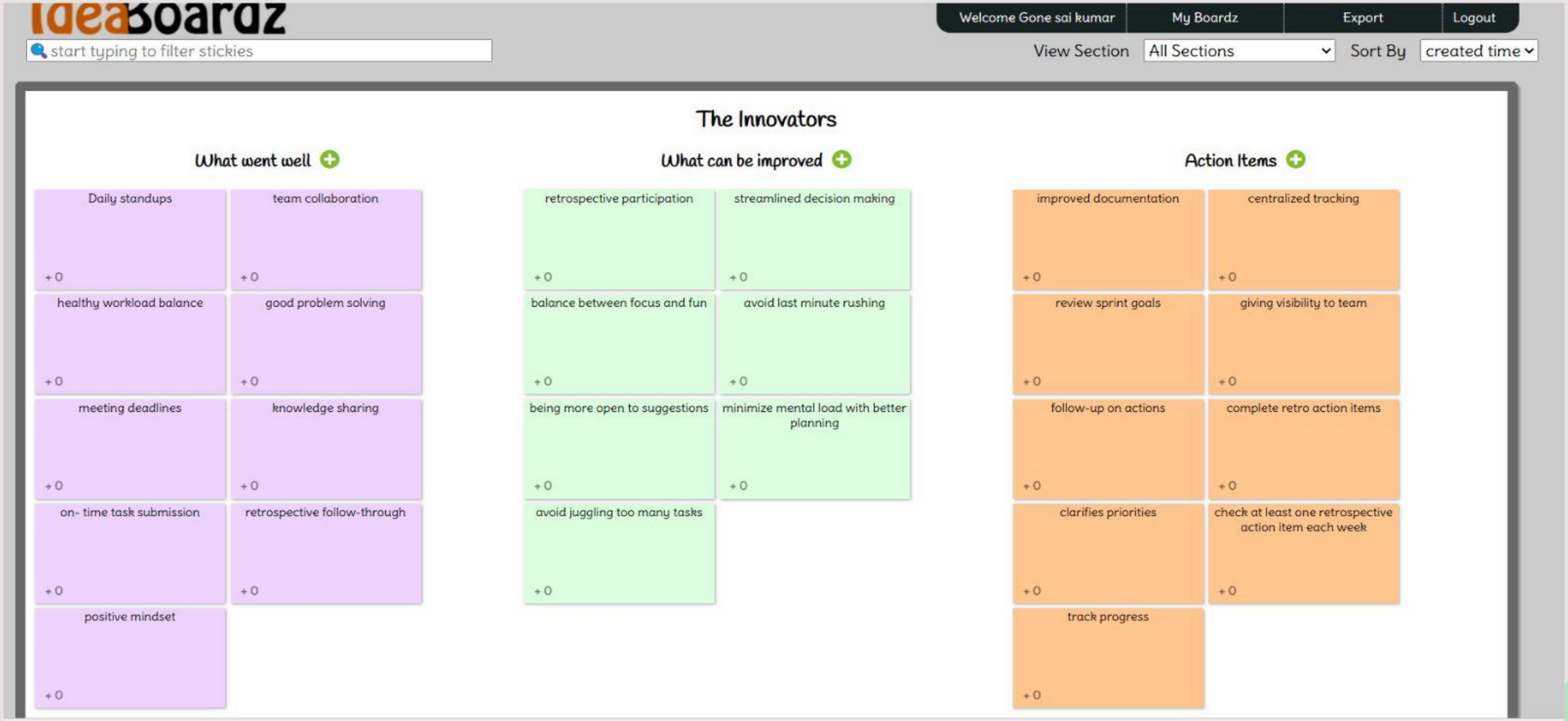
Completed/Committed Ratio

Sprint 1
Committed: 34 story points
Completed: 34 story points
Completed/Committed Ratio: 34 / 34 = 100%

Sprint 2
Committed: 28 story points
Completed: 28 story points
Completed/Committed Ratio: 28 / 28 = 100%

Average Completed/Committed Ratio (Sprint 1 & 2)
(100% + 100%) / 2 = 100%

Retrospective



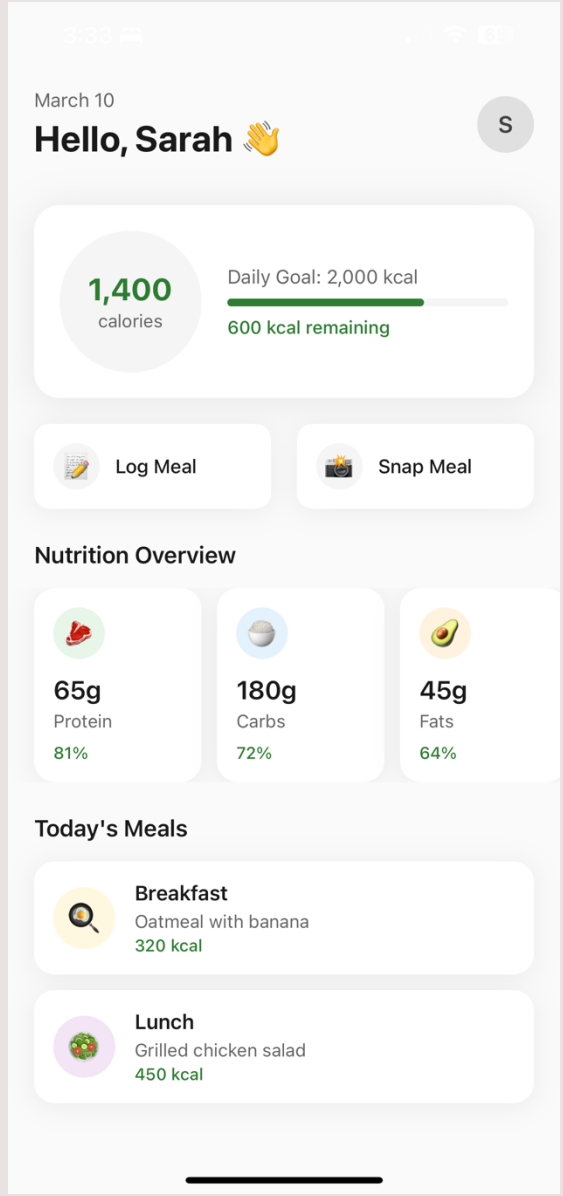
Sprint 3 Planning

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
US11	As a user, I want to see calories and nutrients after entering my meal so that I know what I'm eating.	Users enter food items manually. Calories, protein, carbs, and fats are displayed. Data is saved to the meal log.	Calorie & Nutrition Display	5	3
US12	As a user, I want to see a weekly progress report so that I can track my long-term trends.	Weekly calories graph shown Weekly macro average shown	Progress Report	3	3
US13	As a user, I want a settings page to manage my preferences so that I can customize the app.	User can set units (metric/imperial) User can change goal Changes saved immediately	Settings	2	3
TS11	Build AI-based calorie & nutrition calculation	Users enter food items manually. AI calculates calories, protein, carbs, and fats. Data is saved to the meal log.	AI-Based Nutrition Calculation	5	3
TS12	Build weekly progress backend & UI	Backend calculates 7-day summary UI shows graph & macro summary	Progress Report	3	3

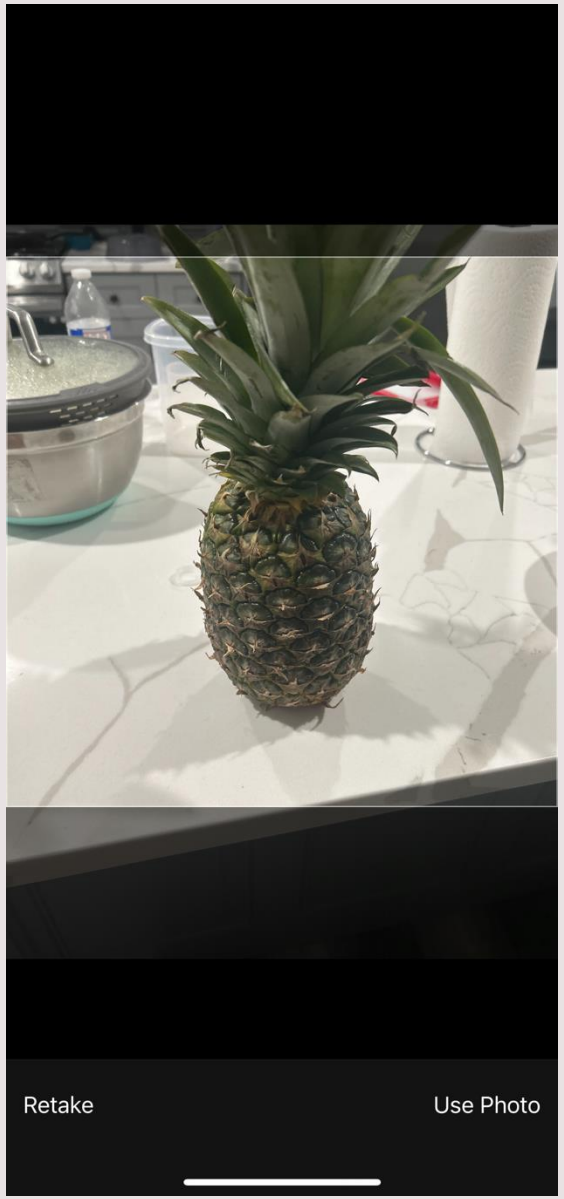
Sprint 3 Planning

ID	User Stories	Acceptance Criteria	Feature	Story Points	Sprint
TS13	Build settings page backend & UI	Preferences saved per user Changes reflected in dashboard Supports units and goals	Settings Backend + UI	2	3

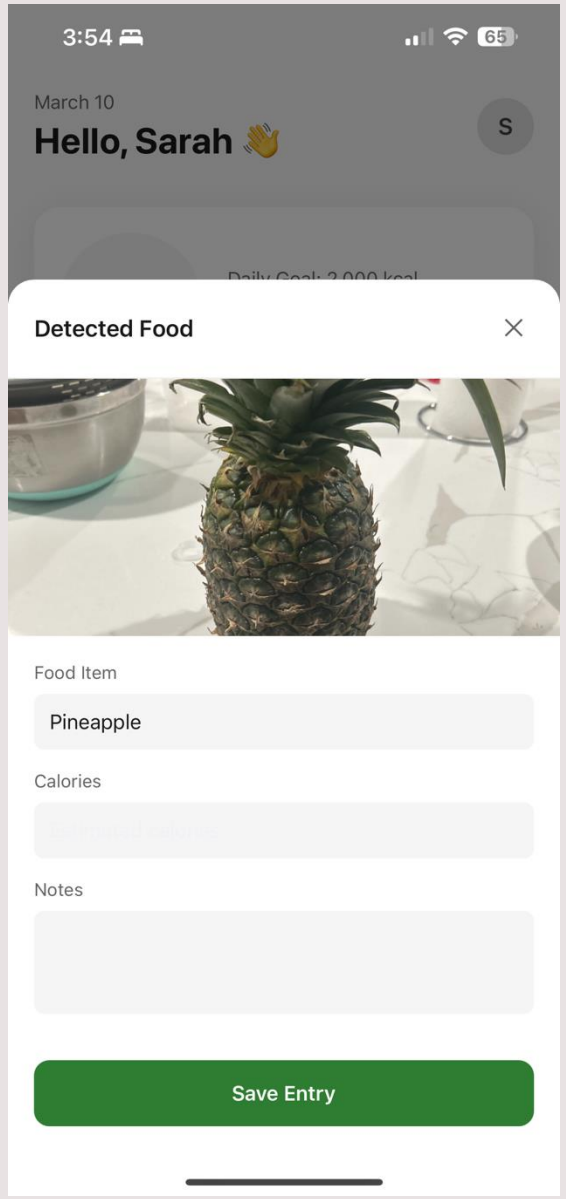
Application Screenshots



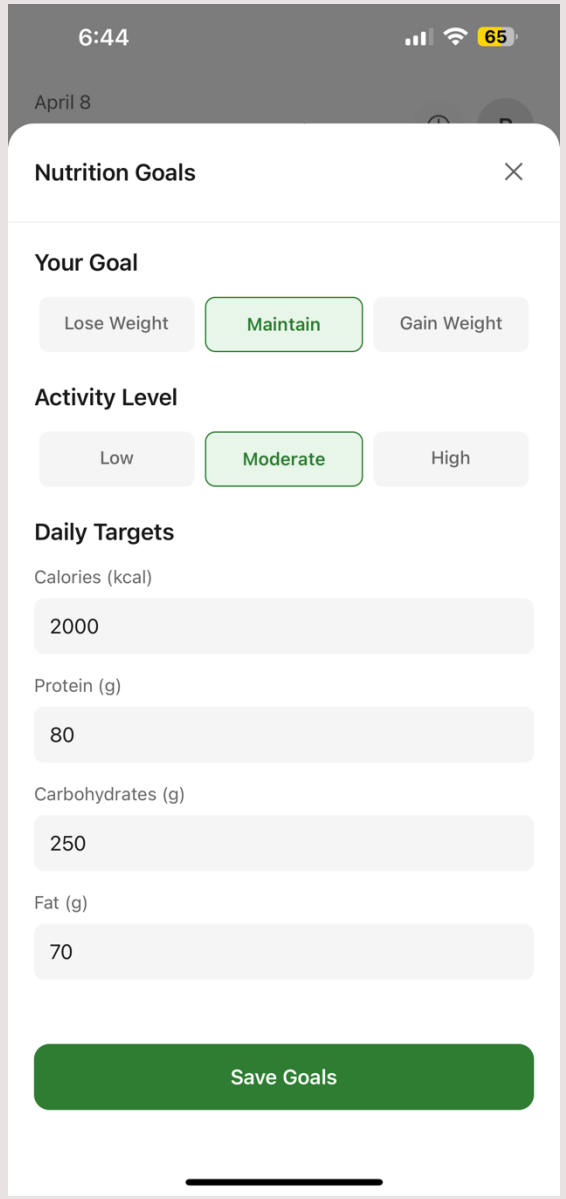
Application Screenshots



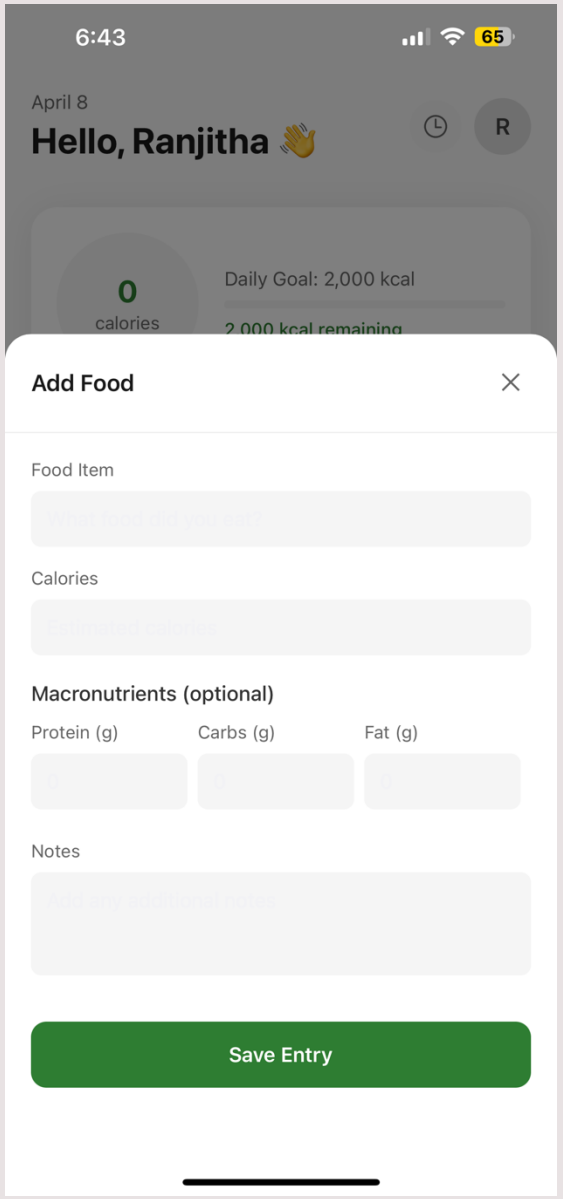
Application Screenshots



Application Screenshots



Application Screenshots



API

Get all foods

$ curl http://localhost:3000/api/foods

{ "success": true, "data": ["apple", "banana", "orange", "strawberry", "blueberry", "watermelon", "grape", "pineapple", "mango", "avocado", "carrot", "broccoli", "spinach", "potato", "tomato", "chicken breast", "salmon", "rice", "bread", "egg"] }

API

Get a specific food (apple)

$ curl http://localhost:3000/api/foods/apple

{ "success": true, "data": { "food": "apple", "calories": 52, "unit": "per 100g" } }

Wikipage Link

https://github.com/htmw/2025S-The-Innovators/wiki

Live Demo