Da Silva Rémi
Valentin BAILLEUL

Only Gains

Calories Counter and Meal Planner Application Documentation

Da Silva Rémi
Valentin BAILLEUL

Only Gains

Objective

The objective of this application is to provide users with a tool to set their health goals, track their daily caloric intake, search for foods in a nutrition database, and plan their meals accordingly. The application aims to promote healthier eating habits and assist users in achieving their desired weight and fitness goals.

Application's Architecture

Here is the folder structure of the application:

1. components: Contains all reusable components of the application, such as modals, forms, and UI elements.
2. screens: This folder contains the components responsible for displaying information to the user. Each screen of the application has its own file in this folder.
3. assets: It is used to store all images and other static resources used in the application.
4. services: Include a file for connecting to the food database API and files for creating different React contexts used to manage data state across the application.
5. other files: The application also includes the base files of React Native, such as App.js which serves as the main entry point of the application.

Succeeded Features

1. Health Goal: Users can input their personal information and health goals, and the application calculates their suggested daily calorie intake based on the provided information. The data of health goals are stored in the local storage.
2. Food Database: Users can search for foods in a nutrition database by entering the food name and add it to the meal plan with a modal. The application retrieves nutritional information for the searched foods from the food database API https://developer.edamam.com/food-database-api.
3. Meal Planning: Users can view for each day and each meal the food added with Food Database, it can add foods from the food database and assign them to specific meals (breakfast, lunch, dinner, snack) for each day of the week. Also the users can delete

foods and see the daily calories. The data of meal planning are stored in the local storage.

Unsucceeded Features

Nothing

Roadmap

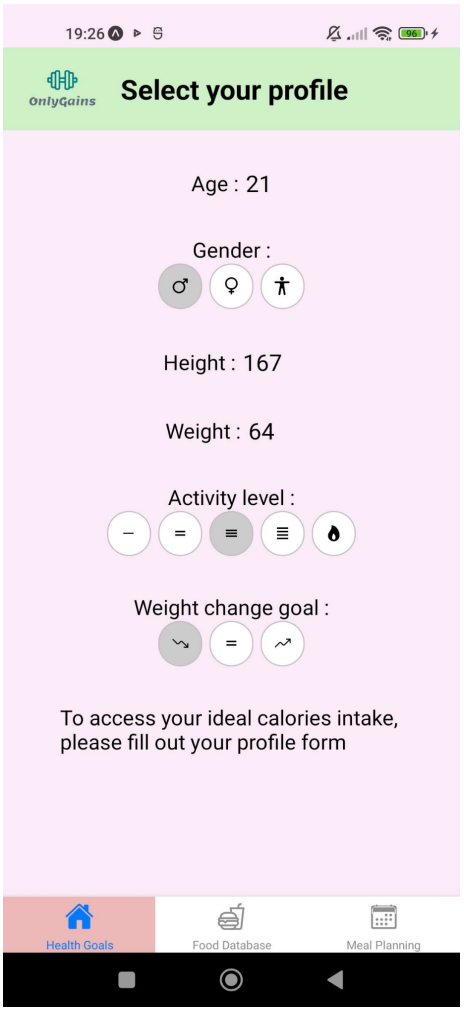
First Week (29/05):

1. Creation of Health Goal
2. Creation of Food Database

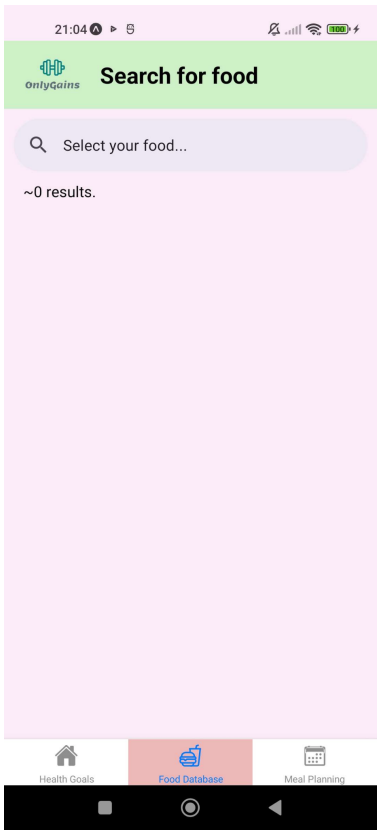
Second Week (05/06):

1. Creation of Meal Planning.
2. Adding missing features in previous pages
3. User interface and styling improvements
4. Bug Fixes

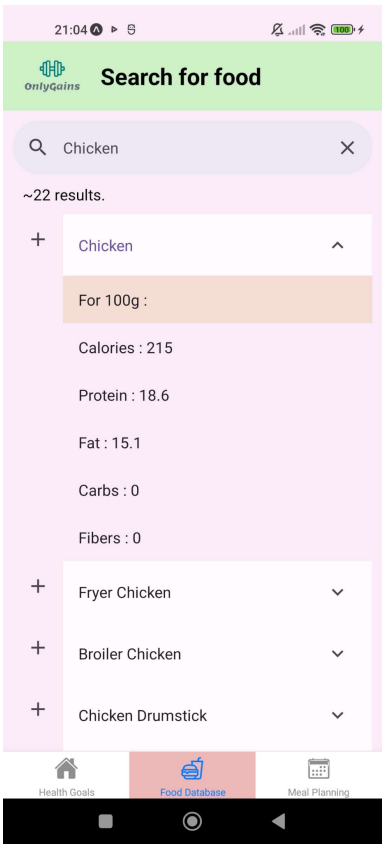
Screenshots



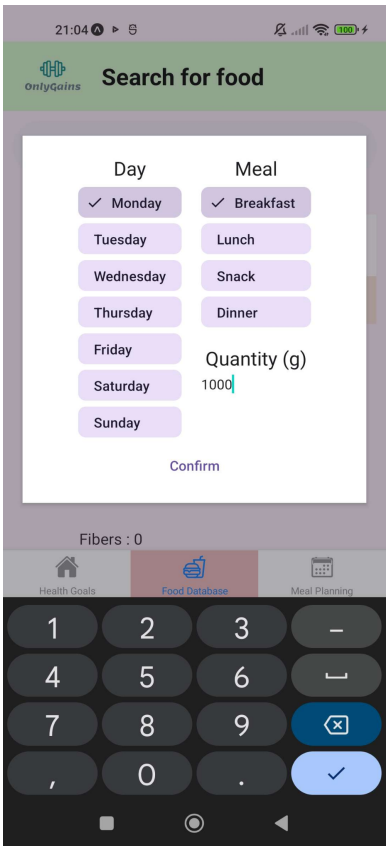
Picture 1 : Health Goals screen for profile customisation (Home page)



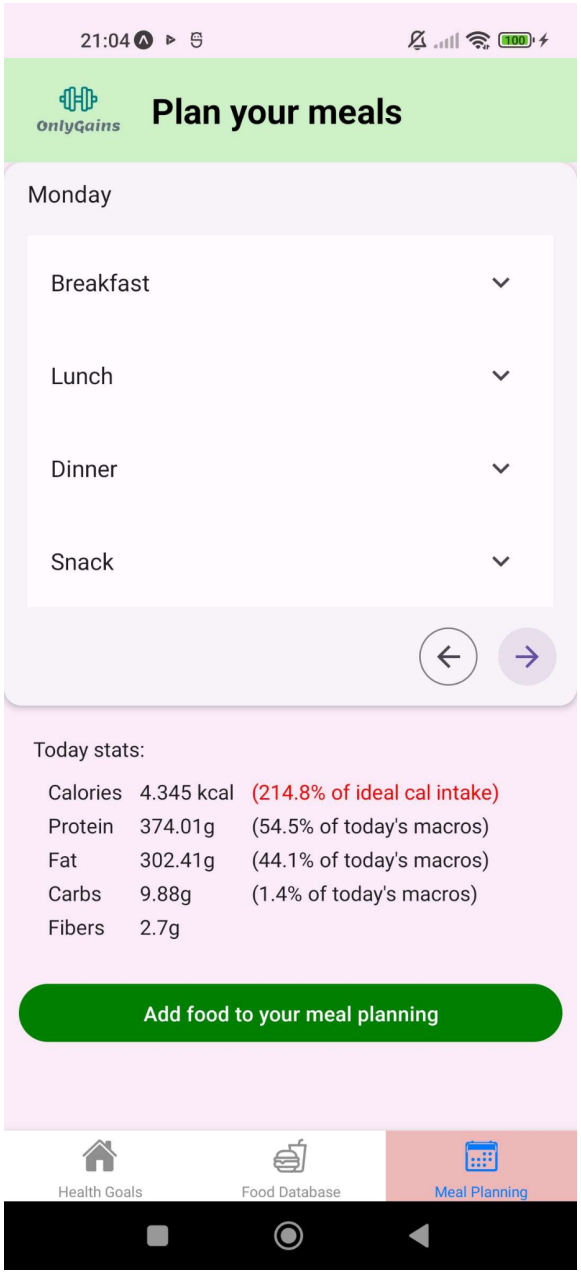
Picture 2 : Food database screen with no result



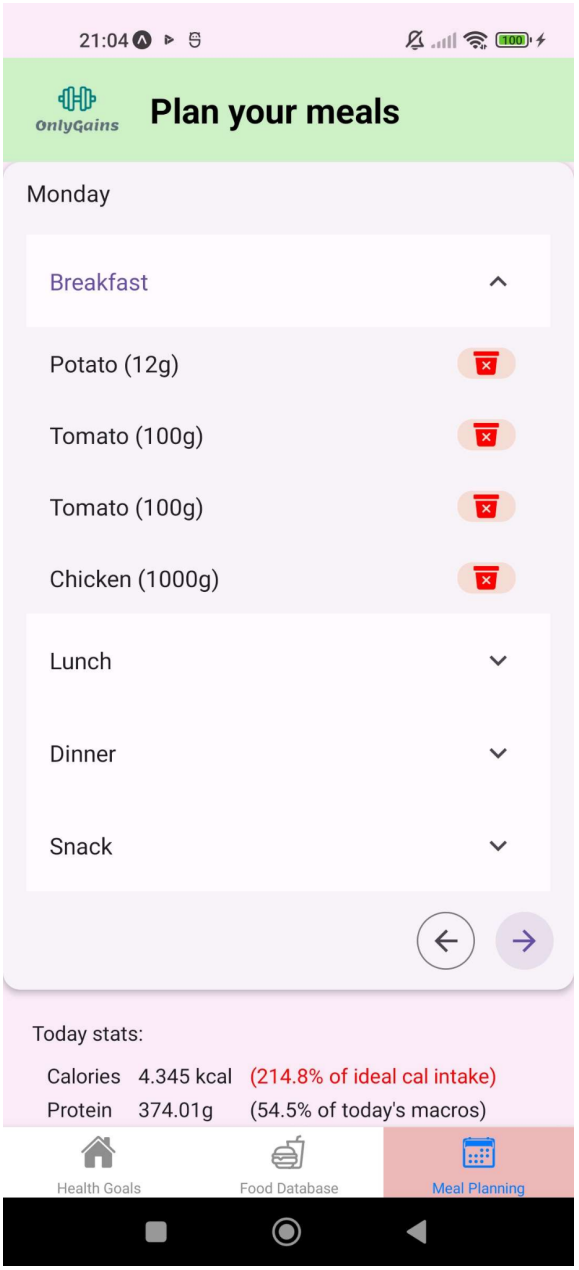
Picture 3 : Food database screen with results with one expanded



Picture 4 : Food database screen modal for adding food the meal plan according to the day, meal and quantity



Picture 5 : Meal Planning screen to review each day with daily stats



Picture 6 : Meal Planning screen with one meal expanded

Challenges Faced

Understanding the Food Nutrition API

We faced some difficulties in understanding the structure of the data returned by the nutrition API. The information provided by the API was nested and organized in a different format. However, through examination of the API documentation, we were able to understand the structure of the results and extract the necessary information accurately.

Handling Asynchronous Data Fetching

Another challenge we encountered was dealing with asynchronous data fetching from the API and ensuring that the retrieved data was displayed correctly in the application. As the API requests and data retrieval were asynchronous processes, we had to carefully manage and update the application's state using the appropriate React hooks. By correctly utilizing hooks such as useState and passing asynchronous updated state directly to children.