NUS Orbital 2024
Milestone 3

Team Moltaise

Tong Kian Kiat

Rayson Tay

Table of Contents

Posters

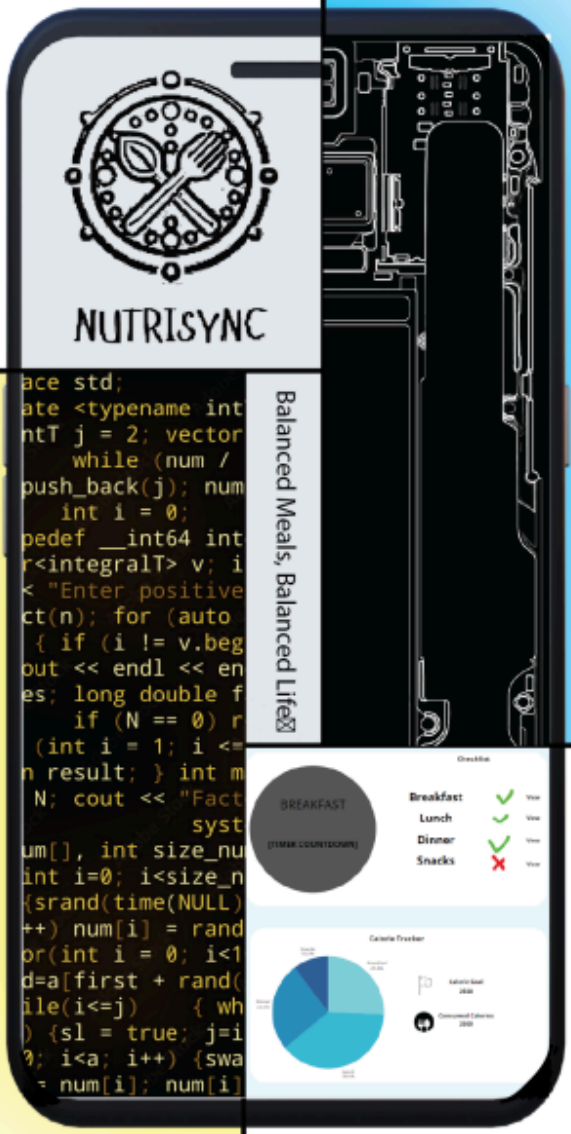

ABOUT
Nutrisync aids users in helping them stay more committed to a dietary plan, and also guide them towards achieving their goals by making it as easy as possible, even if they have no prior experience.
TECH STACK
FEATURES
• Nutrition Tracker
• Fitness Tracker
• Diet Mentor
• Fitness Mentor
• Meal Planner
• Friends List
• Mood Booster
• Tracking Gym Crowdedness
• Rewards System
SWE PRACTICES
Each component of our application is dockerized to ensure consistency throughout development and testing. Branching is implemented for version control, and Pull Requests are being performed to ensure that changes are being reviewed properly before merging into the main branch.
NUTRISYNC
Balanced Meals, Balanced Life
01 NUTRITION TRACKER
Easily log your meals and track daily caloric intake with our user-friendly interface, utilizing a comprehensive database of food items with preset calorie values, while giving users an option to input their own calorie values.
02 FITNESS MENTOR
Receive customized exercise suggestions tailored to your unique biometric data and daily caloric intake. These recommendations help ensure that your physical activity levels are optimized to meet your dietary goals, whether you are in a caloric deficit or surplus.
03 FITNESS TRACKER
Set and customize your dietary goals based on your specific needs, whether you aim to lose fat, gain muscle, or maintain your current weight, ensuring a tailored diet plan.
04 FRIENDS LIST
Link up with friends within the app to share diet plans and track each other's dietary progress. This feature allows for a supportive community environment that motivates and inspires everyone involved.
05 MEAL PLANNER
Never miss a meal log with our convenient reminders. Set alerts for meal times to help you maintain consistent caloric intake and keep your diet on track throughout the day.
06 MOOD BOOSTER
Benefit from targeted, informative tips related to nutrition that pop up in the app. These tips are designed to enhance your understanding of healthy eating and assist you in making informed dietary choices in order to achieve your desired dietary goals.
07 DIET MENTOR
Explore a variety of popular diet plans that fit your lifestyle and nutritional needs, complete with detailed macronutrient ratios to guide your food choices. Users may also integrate parts of these diets into their own.

Introduction

Nutrisync aids users in helping them stay more committed to a dietary plan, and also guide them towards achieving their goals by making it as easy as possible, even if they have no prior experience.

Design

TECH STACK



1 NUTRITION TRACKER

Easily log your meals and track daily caloric intake with our user-friendly interface, utilizing a comprehensive database of food items with preset calorie values, while giving users an option to input their own calorie values.

3 FITNESS MENTOR

Receive customized exercise suggestions tailored to your unique biometric data and daily caloric intake. These recommendations help ensure that your physical activity levels are optimized to meet your dietary goals, whether you are in a caloric deficit or surplus.

3 FITNESS TRACKER

Set and customize your dietary goals based on your specific needs, whether you aim to lose fat, gain muscle, or maintain your current weight, ensuring a tailored diet plan.

4 FRIEND LIST

Link up with friends within the app to share diet plans and track each other's dietary progress. This feature allows for a supportive community environment that motivates and inspires everyone involved.

5 MEAL PLANNER

Never miss a meal log with our convenient reminders. Set alerts for meal times to help you maintain consistent caloric intake and keep your diet on track throughout the day.

6 MOOD BOOSTER

Benefit from targeted, informative tips related to nutrition that pop up in the app. These tips are designed to enhance your understanding of healthy eating and assist you in making informed dietary choices in order to achieve your desired dietary goals.

7 DIET MENTOR

Explore a variety of popular diet plans that fit your lifestyle and nutritional needs, complete with detailed macronutrient ratios to guide your food choices. Users may also integrate parts of these diets into their own.

Evaluation

SWE PRACTICES

Each component of our application is dockerized to ensure consistency throughout development and testing. Branching is implemented for version control, and Pull Requests are being performed to

Conclusion

Summary

NutriSync is an intuitive and user-friendly meal planner and nutrition tracker designed to help individuals, particularly NUS students, stay committed to their fitness and dietary goals. By providing a comprehensive platform that integrates dietary tracking, personalized fitness plans, and smart recommendations, NutriSync simplifies the process of managing health and wellness. The app addresses common challenges such as lack of motivation and complexity found in existing solutions by offering an accessible interface suitable for beginners and experienced users alike. NutriSync fosters a supportive community, encouraging users to stay motivated and achieve their desired fitness outcomes.

Future Plans

NutriSync aims to enhance its capabilities by integrating advanced machine learning for dynamic workout and dietary recommendations, expanding its food and exercise database, and adding gamification features to boost engagement. We plan to integrate with wearable devices for real-time tracking, offer holistic health tracking including mental wellness, and establish corporate and educational partnerships. Additionally, NutriSync will localize for global accessibility and introduce personalized coaching services, ensuring it remains a comprehensive, adaptive, and supportive health and fitness platform for all users.

Nutrisync

Introduction

Nutrisync aids users in helping them stay more committed to a dietary plan, and also guide them towards achieving

Evaluation

VERSION CONTROL

BRANCHING

We use Git for version control to manage our codebase efficiently. The remote master branch is always kept as a stable, working set of code. For new features, we pull from the master branch and create a feature branch following the convention `feature-developer`. This method ensures isolated development, preventing code cross-contamination and simplifying debugging. Upon completing a feature, the feature branch is committed, and a pull request is initiated to merge the changes back into the master branch, ensuring smooth and controlled inte-

PULL REQUESTS

Pull requests are used to update our main branch. For new features, we create a branch off the main branch. After verifying the feature works properly in the separate branch, a pull request is created to integrate it into the main branch, ensuring continued development. Merge requests are accepted only after confirming no significant issues in the branch, preventing the accumulation of bugs during development.

GITHUB ISSUES

When a bug is detected in the development branches, a new issue is created to describe the bug. Feedback from other teams after each milestone is also documented as issues. These issues are referenced in our Git commits when fixes are implemented, allowing for easy reference and resolution in similar future situations.

Design

TECH STACK

1 USER AUTHENTICATION

Users must log in or register to use the application, providing a username, email, and password. Authentication is managed via Supabase. We use the default auth.users table and a custom public.users table to store sensitive information securely.

2 DIET PLAN GENERATION

For our diet plan generation, we made use of an open source model in order to help us with the prediction. The model uses the K-Nearest-Neighbor (KNN) algorithm. Its parameters include an array of nutritional values: Calories, Fat, Saturated Fat, Cholesterol, Sodium, Carbohydrates, Fiber, Sugar, and Protein.

3 GYM CROWDNESS TRACKER

After logging in, users are directed to the main application. The homepage displays the crowdedness of KR Gym, USC Gym, and U Town Gym. This data is queried from the NUS Pool and Gym Traffic Website using an Axios request and displayed on the homepage.

4 CALORIE BASED TRACKING

To track users' daily meals, we use FatSecretAPI to retrieve food items with their nutritional values. The API also filters food based on allergies, ensuring safe consumption. Allergy data is retrieved from the Supabase public.users table and sent as a parameter in our API request.

5 EXERCISE DATABASE

Our exercise tab lets users search for exercises from a database stored in a separate Supabase table. This tab performs RESTful API calls to retrieve and display the data.

6 INVITE FRIENDS

The friends tab allows users to send friend requests and view profiles. We use a public.social table for real-time updates, enabling notifications for new friend requests and accepted requests. Any changes are stored in our database

7 REWARDS SYSTEM

The rewards tab lets users redeem vouchers and gifts with coins. Currently, the rewards are intangible as a proof of concept. When the app is deployed, new users will receive enough coins to test this feature. For additional coins, contact us at nutrisync24@gmail.com.

Conclusion

NutriSync is an intuitive and user-friendly meal planner and nutrition tracker designed to help individuals, particularly NUS students, stay committed to their fitness and dietary goals. By providing a comprehensive platform that integrates dietary tracking, personalized fitness plans, and smart recommendations, NutriSync simplifies the process of managing health and wellness. The app addresses common challenges such as lack of motivation and complexity found in existing solutions by offering an accessible interface suitable for beginners and experienced users alike. NutriSync fosters a supportive community, encouraging users to stay motivated and achieve their desired fitness outcomes.

NutriSync aims to enhance its capabilities by integrating advanced machine learning for dynamic workout and dietary recommendations, expanding its food and exercise database, and adding gamification features to boost engagement. We plan to integrate with wearable devices for real-time tracking, offer holistic health tracking including mental wellness, and establish corporate and educational partnerships. Additionally, NutriSync will localize for global accessibility and introduce personalized coaching services, ensuring it remains a comprehensive, adaptive, and supportive health and fitness platform for all users.

Team Name

Moltaise

Proposed Level of Achievement

Gemini

Motivation

Staying committed to a workout plan is not easy. Oftentimes, we would find ourselves lacking the motivation to hit the gym in order to achieve our fitness goals. As such, we want to build an easy to use meal planner and nutrition tracker application for everyone to use. Most meal planners and nutrition trackers out there are too complicated and user-unfriendly. We also wish to make the app beginner-friendly for users who have no experience in diet planning, and ultimately aid different users in their journey to achieve their desired fitness goals.

Aim

We hope that the application helps NUS students to stay more committed to a dietary plan, and also guide them towards achieving their goals by making it as easy as possible, even if they have no prior experience.

Vision

NutriSync aspires to be the ultimate fitness and nutrition companion, seamlessly integrating dietary tracking, personalized fitness plans, and smart recommendations to support users in achieving their health goals. By leveraging user-specific data and advanced machine learning, NutriSync aims to provide tailored, adaptive plans that evolve with users' progress, ensuring they stay motivated and informed. Our vision is to create an engaging and supportive community where users can track their meals, set fitness goals, receive timely reminders, discover alternative exercises, and benefit from real-time gym crowdedness updates. NutriSync is dedicated to making fitness and nutrition management accessible, personalized, and rewarding for everyone.

User Stories

- As a beginner who wants to track their own calories, I want to be able to do so easily.
- As a consumer who wants to gain muscle, I want to be able to track my calories to ensure that I am on a high protein diet.
- As a consumer who wishes to lose weight, I want to be able to track my calories in order to maintain a caloric deficit.

Project Scope

NUtriSync is an intuitive meal planner and nutrition tracker app with features tailored to help NUS students stay committed to their fitness goals, including personalized diet plans, calorie tracking, gym crowdedness monitoring, and motivational rewards.

NutriSync addresses common challenges such as lack of motivation and complexity found in existing solutions by providing a user-friendly interface suitable for beginners. Key features include the ability to input and track meals with calorie information, set personalized fitness goals (e.g., fat loss, muscle gain), receive tailored diet plan recommendations, and get reminders to log calories. Additionally, the app will offer a gym crowdedness tracker for NUS gyms, alternative exercise recommendations, and a rewards system to incentivize gym attendance. Users can also invite friends for joint workouts and receive random informative tips. Advanced functionalities will include personalized workout plans based on user biometrics and caloric intake, with potential extension to integrate a machine learning model for dynamic workout adjustments.

Finalised Features

Note: We are having difficulties getting the model backend to run on our proxy server, so our diet generation in the APK is unable to function.

User Authentication

Users are required to log in or register for a new account when using the application. Firstly, we ask users for their username and email, as well as a password for their account. Our authentication is done based on Supabase authentication. Apart from the in-built auth.users table Supabase creates whenever there is a new user, we triggered Supabase to create our own public.users table, so that we can store sensitive information from each user here.

id uuid	email text	username text	goals text	height int4	weight int4	age int4
6198f63d-a286-4b83-a7be-92a5b031e8b	kktong500@gmail.com	test2	Gain Weight	176	56	22
13c24166-e791-42a7-9222-0b6f067cbade	tong.kiankiat@gmail.com	kiankiat	Gain Weight	176	56	22
69890757-8110-4c8c-8860-194626e988f	tototrangcvp@gmail.com	thanhchu	Lose Weight	155	48	22
bc310112-e016-43d5-874d-a1884198f139	huangqiyuan.1@gmail.com	Repkter	Lose Weight	180	75	21
b2ea9a46-93d1-48bb-9529-cb9f9afb16b	kktong200@gmail.com	kk	Gain Weight	176	56	22
897d0945-bddc-49b1-8318-d822a18b95b	kktong300@gmail.com	test	Gain Weight	176	56	22
0aa0ca65-18e2-4ac0-a2d4-a78b7c3331e	raysontay@gmail.com	raysontkm	Lose Weight	173	69	22

Supabase generates a unique uuid for each user, which we used as a primary key for our public.users table. We have also adjusted the table's policy such that only authenticated users have read access to this table.

Afterwards, we prompt users for more information:

- Gender
- Age
- Height
- Weight

These 4 pieces of information are used to calculate the user's Base Metabolic Rate (BMR), and Body Mass Index (BMI), in order to come up with the maintenance calories of our user. BMR is calculated using the Mifflin-St Jeor Equation:

Mifflin-St Jeor Equation:

For men:

$$BMR = 10W + 6.25H - 5A + 5$$

For women:

$$BMR = 10W + 6.25H - 5A - 161$$

while BMI is calculated using:

SI, Metric Units:

$$\text{BMI} = \frac{\text{mass (kg)}}{\text{height}^2\text{ (m)}} = \frac{72.57}{1.78^2} = 22.90\frac{\text{kg}}{\text{m}^2}$$

Next, we ask users for their fitness goals, and active level. Fitness goals are split into 3 categories: Lose Weight, Maintain Weight, Gain Weight. Taking the goal of maintaining weight as the baseline, we account for a 20% change in calorie intake if users wish to gain or lose weight. Active levels are also split into 3 categories: Sedentary, Lightly Active, Moderately Active. The multiplier for the active levels are, respectively, [1.2, 1.375, 1.55]. Combining this with the earlier information, we calculate our user's final caloric goal to be:

Calorie Goal = BMR * [Fitness Goal Multipler] * [Active Level Multiplier]

Next, we ask users for any allergies. The main purpose of this information is to filter through our nutritional API, provided by FatSecretAPI, so that any food items that our user is allergic to would not be provided to our user.

Finally, we ask users for their preferred meal times. This is to let us know when we can send reminders to our users to log their meals, if they have not done so.

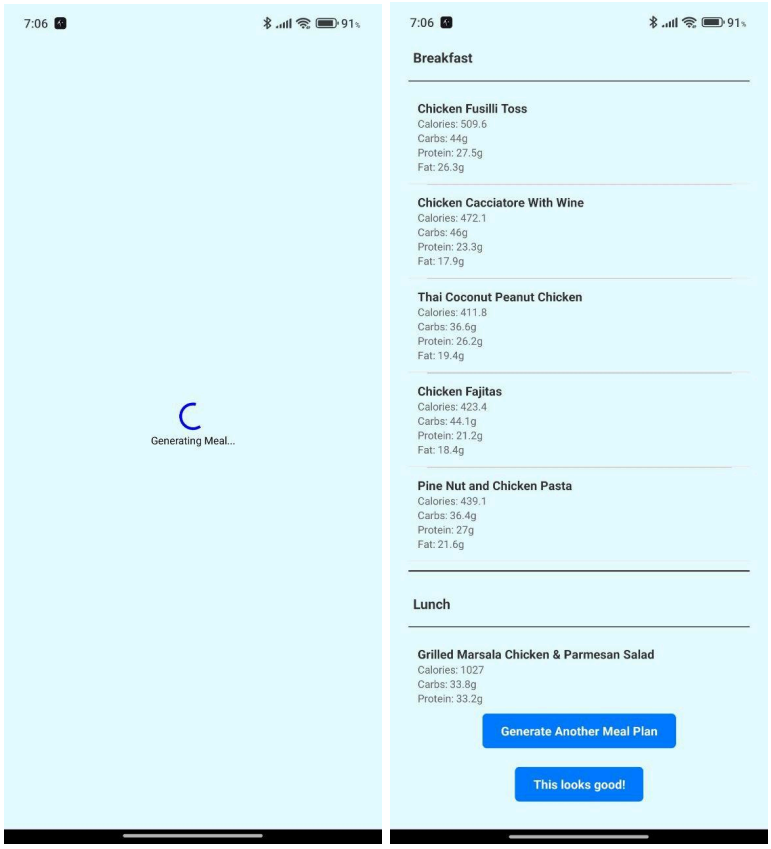
Diet Plan Generation

For our diet plan generation, we made use of an open source model in order to help us with the prediction.

How the model works (simply put):
The algorithm used by the model is the K-Nearest-Neighbour (KNN) algorithm. The model's parameters are an array of nutritional values, namely: [Calories, Fat, Saturated Fat, Cholesterol, Sodium, Carbohydrates, Fiber, Sugar, Protein]. Based on these parameters, the model standardises them and uses KNN to locate recipes that have similar nutritional values. The 'cosine' metric is used as we want the model to predict based on the relative proportions of each nutritional value.

The dataset used by the model is provided by Kaggle, and currently does not support any type of filtering system. As such, we integrated our FatSecretAPI, which has an inbuilt allergy filter, in order to provide recipes that would not be harmful towards our user. Upon retrieving the 5 food names for each meal, we run the food names through our FatSecretAPI, which already has the allergies array that we asked from the user previously. If we receive any relevant response from FatSecretAPI, we then accept the food as non-allergic for our user, and then display it to the user. However, due to the fact that they both use different datasets, this method is not guaranteed to be 100% foolproof. As such, we still urge our users to double check in the event there are any food items that they are unable to safely consume.

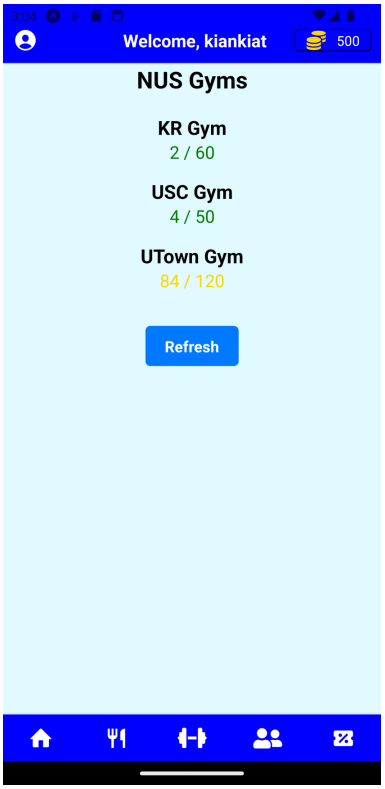
We separate the diet recommendations into 4 meals; Breakfast, Lunch, Dinner and Snacks. For each meal, we recommend 5 food items for the user. This allows users to have a different food option for every meal to prevent repetition. After users accept the diet plan generated, they are stored in our Supabase public.recommended_diets table, where we store the diet plan of all our users and refer back to them for logging purposes. This table stores the food name, the meal time (e.g. Breakfast), as well as the macros.



Diet Plan Generation

Gym Crowdedness Tracker for NUS gyms

After logging in, users are directed into the main application stack. In our home page, we display the crowdedness of NUS Gyms, namely KR Gym, USC Gym, and U Town Gym. We query this data from the [NUS Pool and Gym Traffic Website] using an Axios Request, and display the capacity of the gyms on the home page.

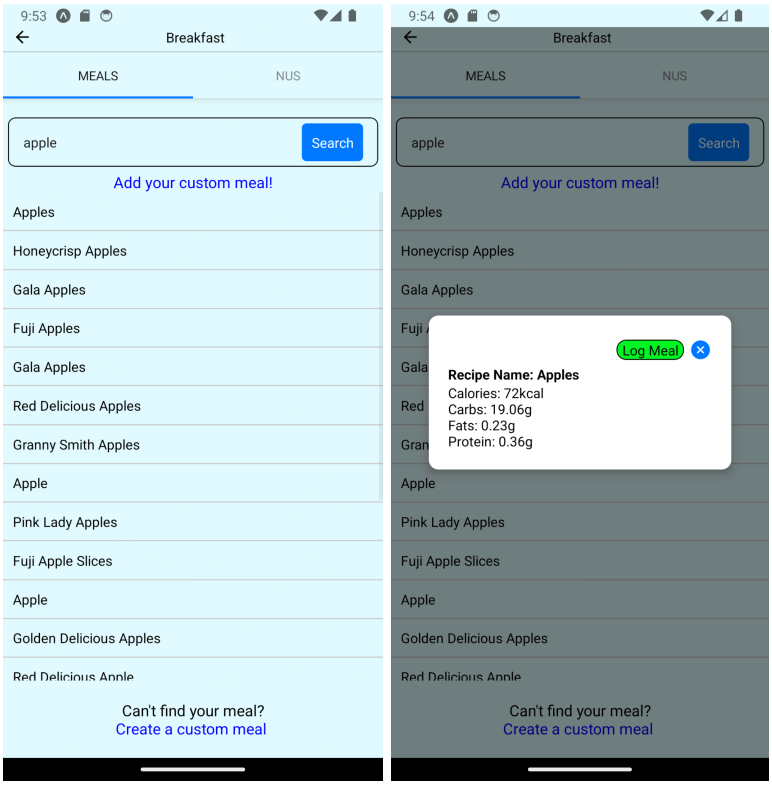


Gym Crowdedness Tracker

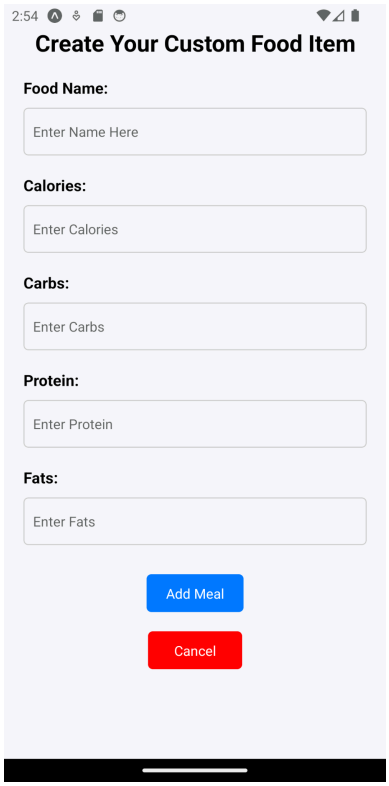
Calorie Tracker Based on Meal Logging

In order to track user's daily meals, we use our FatSecretAPI to retrieve food items with their nutritional values. FatSecretAPI offers filtering based on food allergies as well, which allows us to display food items that the user is able to safely consume. The allergies array is retrieved from the Supabase public.users table, and we send it as a parameter in our request to the FatSecretAPI.

Users can log their meal in 2 ways. By either searching it via our FatSecretAPI, or logging their own custom meal with custom macros.

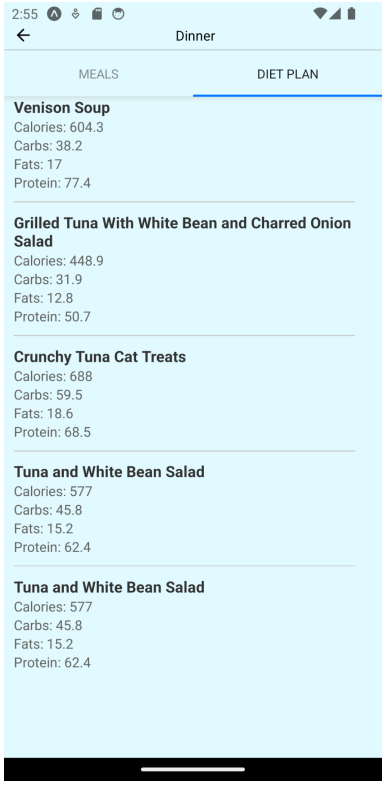


FatSecretAPI



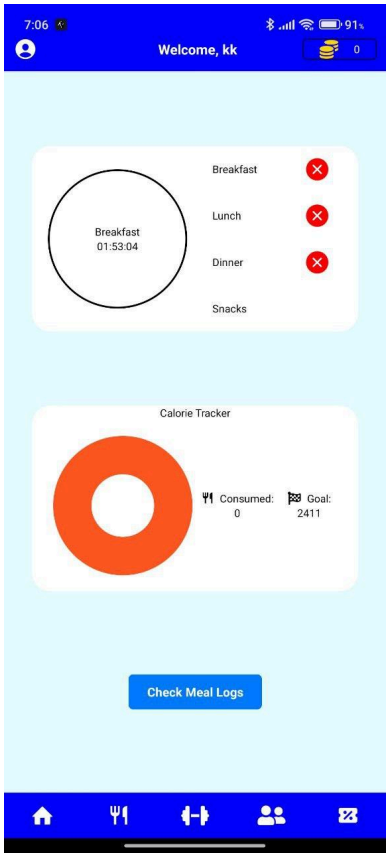
Custom Meal Logging

There is also a section for users to directly log the meal items that was generated for them by the model.



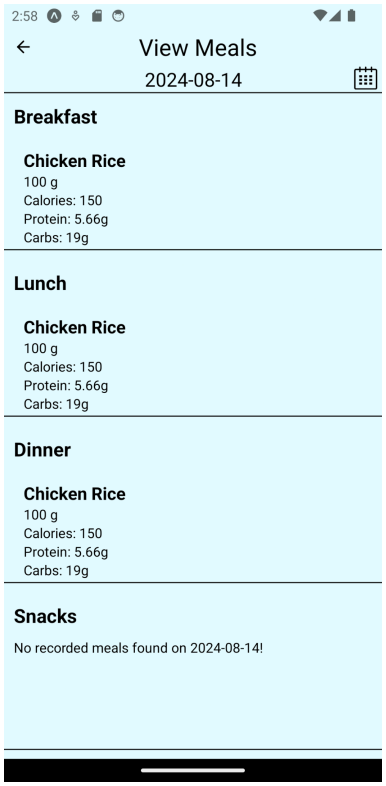
Recommended Meal Items

We also added a dynamic pie chart, that shows users how close or how far they are to their caloric goal for that day. The data for the pie chart also comes from the FatSecretAPI query, and we calculate the macros directly from the response.



Pie Chart

There is also a section for users to view the logged meals for previous days. Just by clicking on any of the meal times, users will be brought into a separate screen, where they can see what they ate on previous days by filtering using the date button.

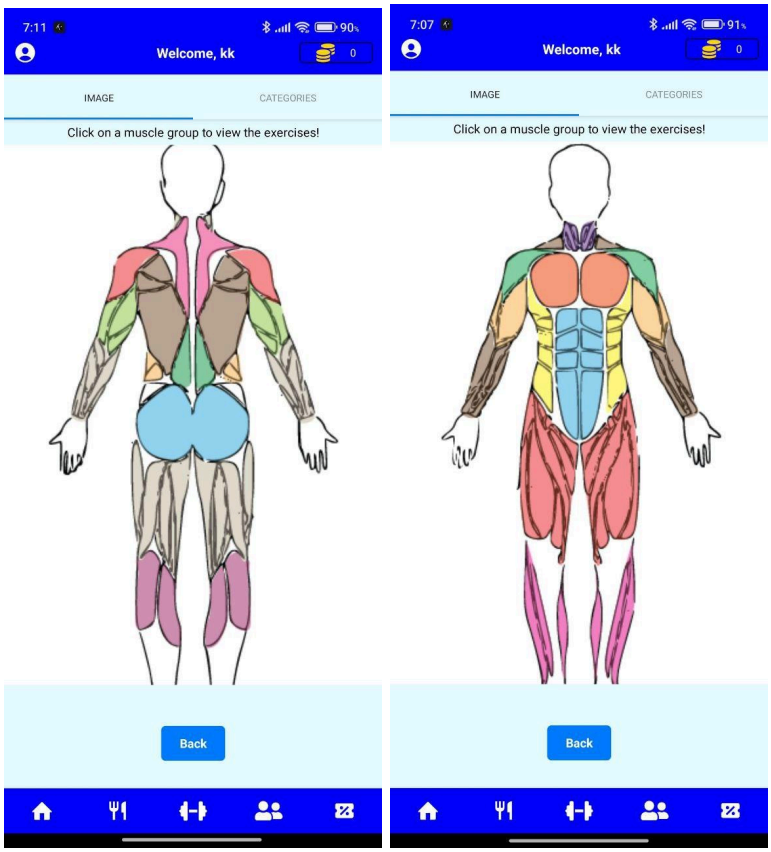


Check Meal Logs

Exercise Database

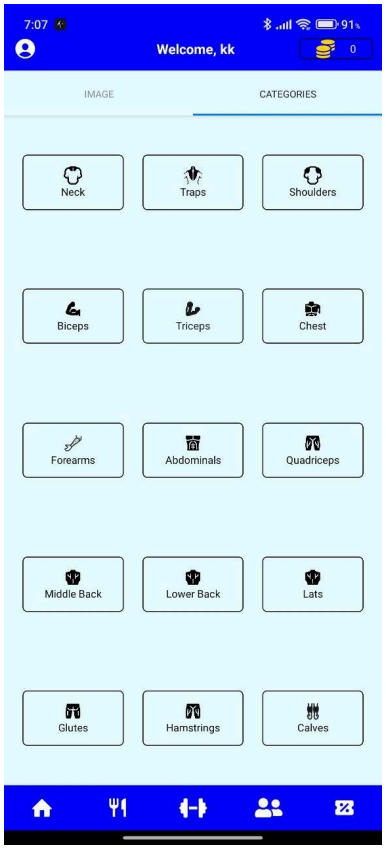
Our exercise tab allows users to search for any exercises based on this database. We have imported and parsed the data into a separate Supabase table, and this tab will perform RESTful API calls to this table and return the data to the application.

On this screen, we have 2 main ways for users to search for exercises. The first tab displays the human muscle anatomy, and users can simply click on each muscle group (highlighted by colour) in order to view the exercises for that muscle group. We feel that this feature also educates users on where the prominent muscles are located on the body.



Muscle Anatomy

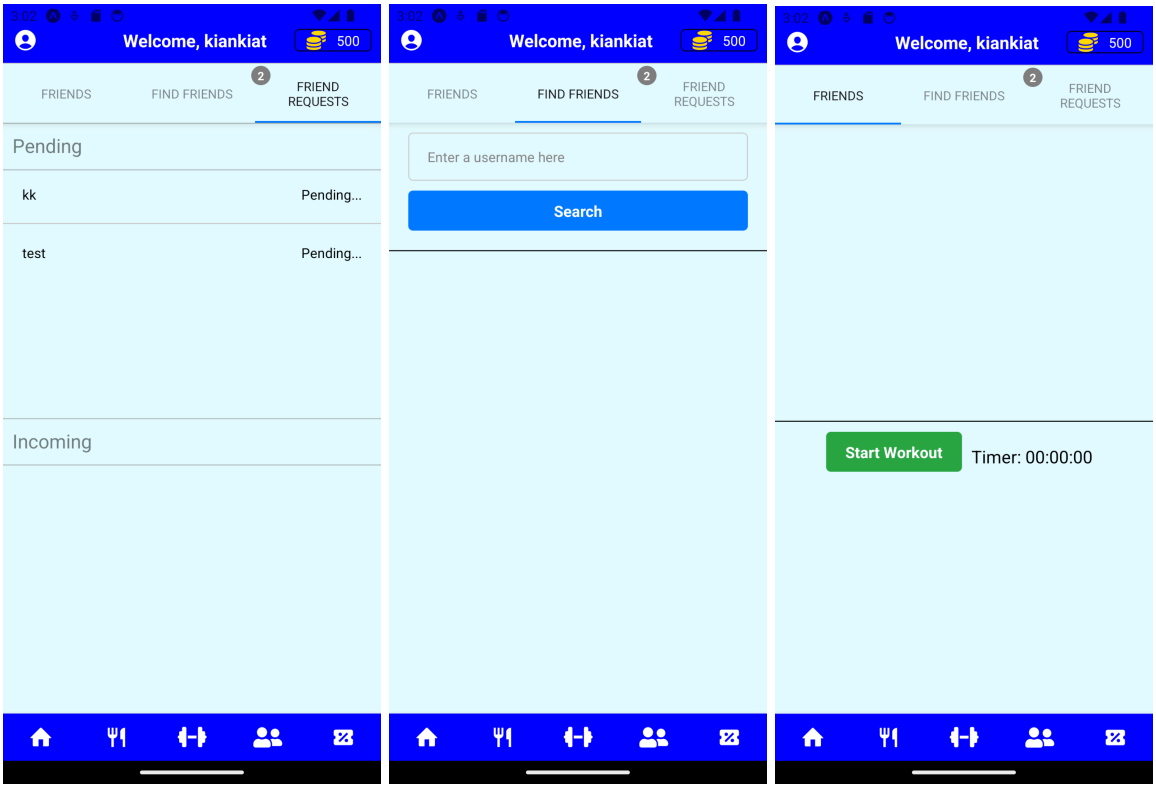
Alternatively, users can head on to the second tab, where they can just click directly on whichever muscle group they want to view the exercises for.



Muscle Groups

Friend System

The friends tab allows users to send friend requests to one another.. For this feature, we implemented a public.social table that we get realtime updates from, as documented here. The app updates in real time whenever there is a new friend request or new pending request.



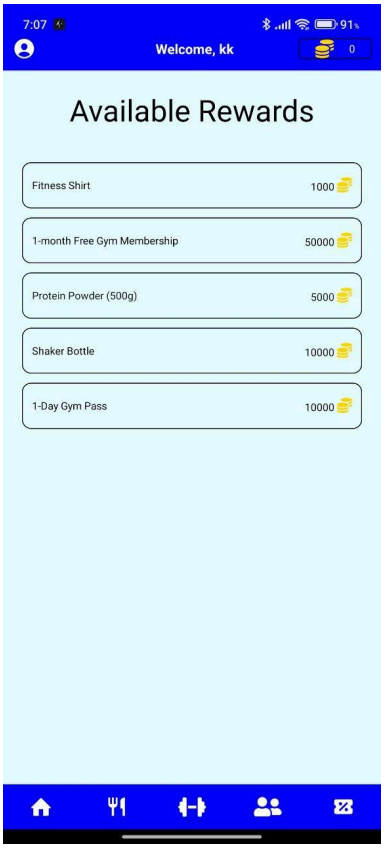
Friends Tab

The friends list and requests list are updated as long as there are any changes being made to the friends and requests array stored in our database.

There is also a section where friends can see if one another is exercising or not. When a user clicks on Start Workout, the user's friends will be notified through that section.

Rewards System

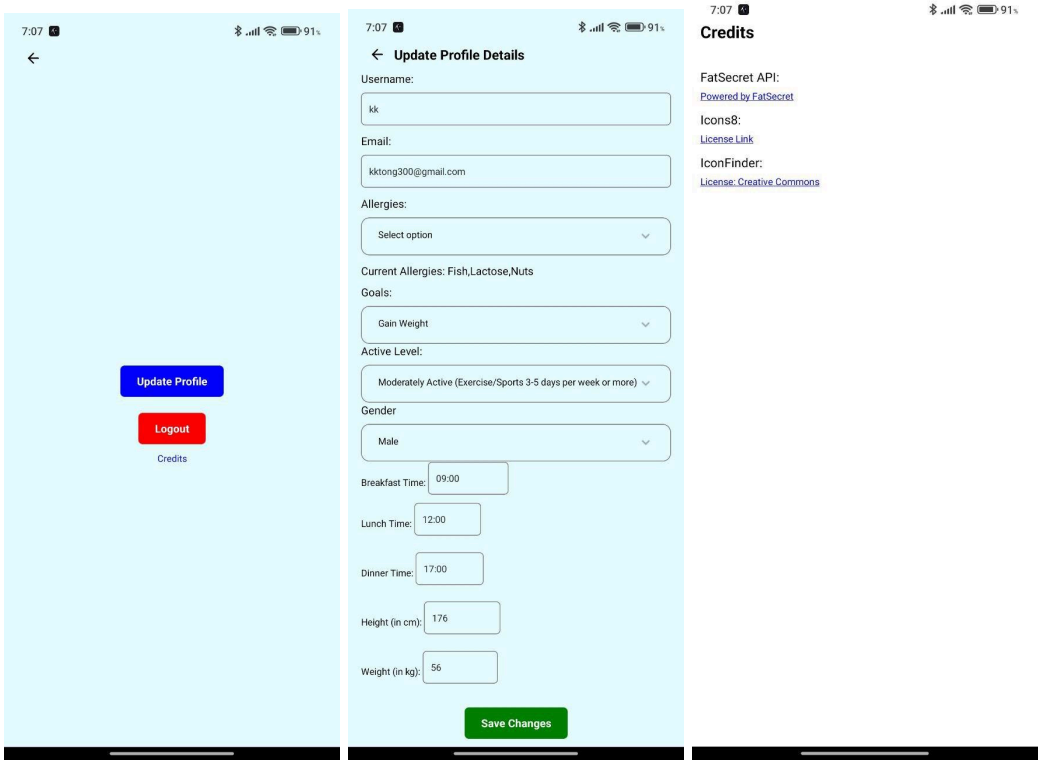
Our rewards tab allows users to redeem vouchers and gifts based on how many coins they have. As of now, the rewards we have put in the application are intangible, and they only serve as a proof of concept. When we deploy our app as an apk, we will ensure that new users have sufficient coins, so that this feature can be tested properly.



Rewards Page

Update User Details

Finally, we also allow users to update their details, and also credit the resources we used. This screen is accessible from the top left of the main stack screen, by clicking on the profile icon.

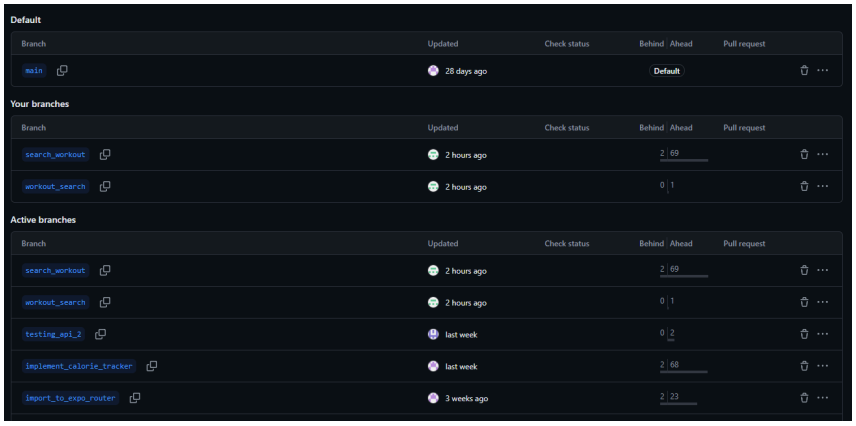


Update Profile, Logout, and display credits.

Software Engineering Practices

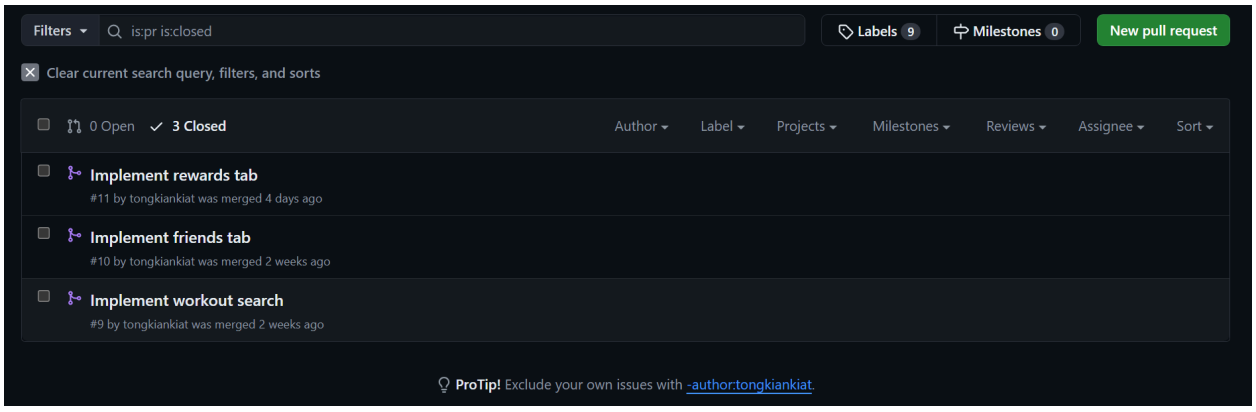
Version Control

Branching



We utilize Git for version control to effectively manage our codebase. The remote master branch is maintained as a stable, working set of code at all times. When developing new features, we pull from the master branch and create a feature branch named using the convention feature-developer. This approach ensures that development occurs in isolation, preventing code cross-contamination and simplifying the debugging process. Once a feature is completed, the feature branch is committed and a pull request is initiated to merge the new changes back into the master branch, ensuring a smooth and controlled integration of new functionalities into the app.

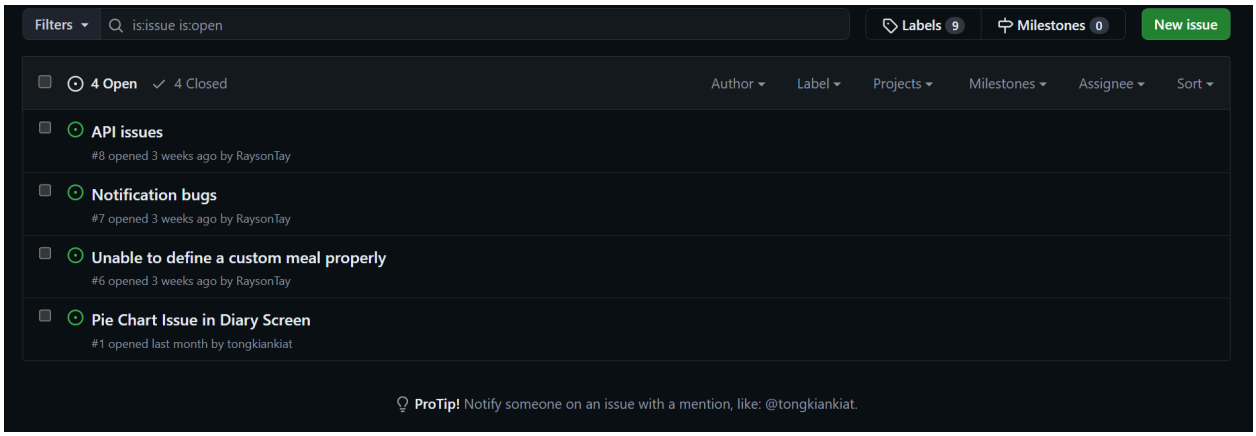
Pull Requests



Pull requests are used to update our main branch. Whenever we add a new feature, we would create a new branch off the main branch to work on the new feature. After ensuring that our new feature works properly in the separate branch, a pull request is created to integrate it into the

main branch, so that we can continue our development. These are also very useful for bug fixing. If we deem any bugs on the current working branch, we can simply create a new branch from it, and fix the issue before merging back into the working branch. Merge requests are accepted only when we deem that there are no glaring issues on that branch. This also prevents bugs from snowballing as we continue developing.

GitHub Issues



Whenever any of us detects a bug in the development branches, we would open a new issue to explain and depict the bug. We also used issues to include feedback that we got from other teams after each milestone. These issues are also quoted in our git pushes whenever we implement any fix for the issue, so that we can easily refer to how we resolved certain bugs in the event a similar situation occurs again next time.

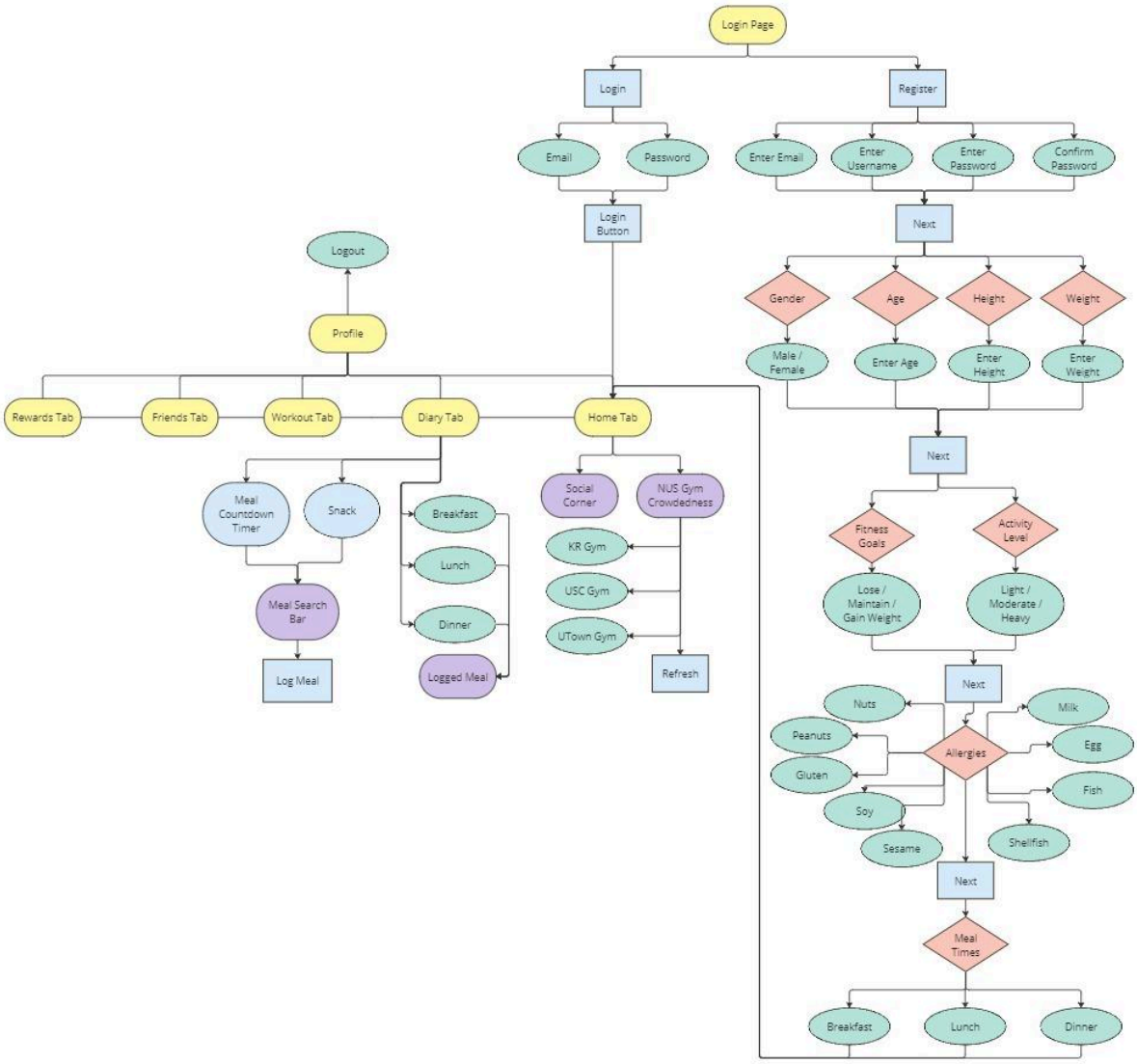
Quality Control

Integration Testing

Integration Test					Results	
Test ID	User Story	Testing Objective	Steps Taken	Expected Results	Pass / Fail	Date Tested
1	As a new user, I want to create a new account.	Test the ability to create a new user account.	1. Launch the app 2. Click on the Register button 3. Key in all relevant onboarding details 4. Click on Register button	Users are able to create a new account.	Pass	28/07/2024
2	As a user with an existing account, I want to log in to the app.	Test the ability to log into the app.	1. Launch the app 2. Click on the Login button 3. Key in tester account user email and password 4. Click on the login button	Users are able to log into the app.	Pass	28/07/2024
3	As a user, I want to log my workout.	Test the ability to log workouts.	1. Launch the app and log in with tester account 2. Click on the Start Workout button on the home page 3. Successfully log workout	Users are able to log their workouts.	Pass	28/07/2024
4	As a user, I want to be able to log my meals.	Test the ability to log meals.	1. Launch the app and log in with tester account 2. Navigate to Meals tab 3. Click on + button for selected meal 4. Search for specific Meal 5. Click on Log meal	Users are able to log their meals.	Pass	28/07/2024

			button			
5	As a user, I want to be able to look for exercises.	Test the ability to search up exercises for specific muscle groups.	1. Launch the app and log in with tester account 2. Click on specific muscle group 3. Click on Exercise 4. View description	Users are able to look for exercises.	Pass	28/07/2024
6	As a user, I want to be able to update my profile.	Test the ability to update profiles and preferences within the profile tab.	1. Launch the app and log in with tester account 2. Click on Profile button 3. Update any particulars 4. Click on Save Changes button 5. Database is updated with the respective changes	Users are able to update their profile.	Pass	28/07/2024
7	As a user, I want to be able to interact with my friends.	Test the usability of the friends tab.	1. Launch the app and log in with tester account 2. View friends in friends tab 3. Search for other friends in find friends tab 4. View incoming and outgoing friend requests in friend request tab	Users are able to interact with their friends in the friends tab.	Pass	28/07/2024
8	As a user, I want to be able to interact with my rewards.	Test the ability to view and claim rewards.	1. Launch the app and log in with tester account 2. Redeem rewards by clicking on them	Users are able to interact with their rewards in the rewards tab.	Pass	28/07/2024

Overall Workflow Navigation



Technologies

React Native (TypeScript)

React Native was the main language used to develop our application.

Expo-Go / Android Emulator

Expo-Go was the main platform we wrote our react native code on, and their mobile application also provided us a platform to perform development. Android Emulator was also used for development and testing purposes.

Flask, FastAPI (Python) and Express (JS)

For the model we implemented to recommend diet plans, the prediction model runs on FastAPI, and we created a Flask application to post and get requests from it. Our proxy server hosted on Heroku uses the express API framework.

Docker / Docker-Compose

In total, we had 3 APIs to host. Thus, to simplify the process, we created a docker container with 3 separate docker images for each server. The docker is then built on Heroku, and we made use of docker-compose in order to run all the servers concurrently. For development, we went with docker-compose to easily get all 3 APIs up and running. Since Heroku does not support docker-compose, we uploaded the docker images of the 3 separate applications to Heroku, and ran them from there.

SupaBase (Authentication and SQL Database)

We chose SupaBase as it was easy to use and also gave us flexibility in writing our own SQL code to perform more complex tasks or permissions.

FatSecretAPI

Nutritional API used to retrieve data regarding food items from.

NUS Gyms Data

NUS Gym capacity is taken from the website:
https://reboks.nus.edu.sg/nus_public_web/public/index.php/facilities/capacity

ChatGPT

ChatGPT was mainly used for debugging purposes, and was helpful whenever we encountered an error that we could not find anywhere else.

Timeline and Development Plan

Milestone	Task	Description	Done By	Duration
1	Drafting of UI Design	Drawing up of UI for the various screens of the application	Kian Kiat Rayson	20 May - 3 June (ongoing for the other features)
		Logo Design	Rayson	20 May
	Getting familiar with using Expo-go and React Native	Studying of React Native and Typescript in order to perform frontend tasks for the application	Kian Kiat Rayson	20 May - 24 May
	Designing the UI for the Login and Register Screens	Login Screen	Kian Kiat	25 May - 26 May
		Register Screen	Rayson	25 May - 26 May
	User Authentication	Implementing the Firebase Authentication and FireStore Database to store username and email	Kian Kiat Rayson	27 May - 29 May
	Input of fitness goals by users	Designing of UI for inquiring user's fitness goals and biometrics after taking registration	Rayson	30 May
		Storing of user's information (fitness goals, biometrics), and outputting a diet and workout plan based on the fitness goals (Work in Progress)	Kian Kiat	30 May - 3 Jun
2	Caloric Tracker Feature	Create database to with various preset food items and their corresponding calorie counter	Kian Kiat	Week 3 (3 Jun - 9 Jun)

		Integrate the caloric tracker into the application, based on the UI created	Kian Kiat	Week 3 - 4 (3 Jun - 16 Jun)
	Notification System	Integrate Notification System into Application that reminds user to input their calories	Kian Kiat	Week 4 (10 Jun - 16 Jun)
	Gym Crowdedness Tracker	Integrate API in application to get information regarding gym crowdedness	Kian Kiat	Week 5 (17 Jun - 23 Jun)
	Alternative Exercise Recommendations	Implement backend logic to refer to database and fetch these exercises for the user, in the event certain exercises are unable to be performed	Rayson	Week 5 (17 Jun - 23 Jun)
		For each exercise, include a video or photo that teaches users how to perform the exercise.	Rayson	Week 6 (24 Jun - 30 Jun)
3	Social System	Implement a friends list in the application	Rayson	Week 7 - 8 (1 July - 14 July)
		Enable friends to invite each other for exercise sessions (via messaging or prompt feature)	Kian Kiat	Week 7 - 8 (1 July - 14 July)
	Machine Learning Model	Studying on integrating Machine Learning algorithms into diet and workout	Rayson	Week 9 (15 July - 21 July)

		recommendations		
	Implement Dockerizing	Dockerise the various components of the application	Kian Kiat	Week 10 (22 July - 29 July)
	Informative Tips	Create a database for the tips, and based on the user's fitness goals, provide tips that can motivate or aid them on their journey	Rayson	Week 10 (22 July - 29 July)
		Implement the tips in the UI of the application in various screens	Kian Kiat	Week 10 (22 July - 29 July)
4	Testing	Testing and Debugging	Kian Kiat Rayson	Week 11 - 14 (29 July - 28 Aug)
		User Feedback		
	Features	Buffer period for polishing or improvement of features		

Deployment

Refer to the attached APK file: apk file
We foresee some issues with registration (especially diet generation), as heroku does sometimes run out of memory with our model application. Please use our test account if there are issues.

Test Account:
Email: tong.kiankiat@gmail.com
Password: 123456

Project Log

Refer to the attached log: Project Log