

# **Fitness & Wellness Tracker Design Document**

**SDEV265-11P**

**Team 2**

## Table of Contents:

Table of Contents: .....	2
Introduction.....	3
System Architecture .....	4
Data Design.....	6
Interface Design .....	8
Test Cases .....	9
Summary .....	11

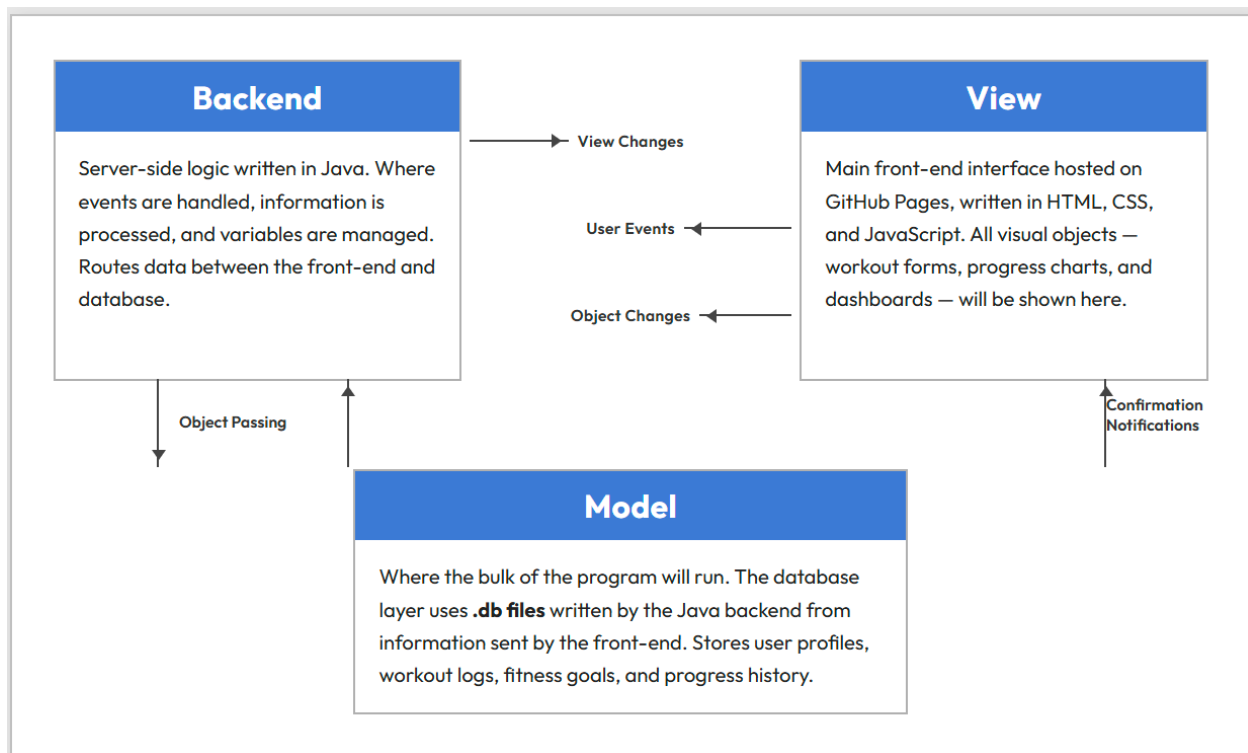
## Introduction

This project introduces a comprehensive fitness and wellness tracking application designed to meet the growing need for simple, meaningful, and user centered health management. Many individuals struggle to maintain consistent routines because their health information is scattered across multiple apps, each focusing on only one part of their lifestyle. This application brings everything together by integrating physical activity tracking, nutrition monitoring, and wellness insights into a single, accessible platform. The goal is to give users a clear understanding of how their daily choices influence their long-term well-being and to help them make informed decisions through organized, data driven feedback. By presenting information in a way that is easy to understand and personally relevant, the app aims to reduce confusion, increase awareness, and support healthier habits.

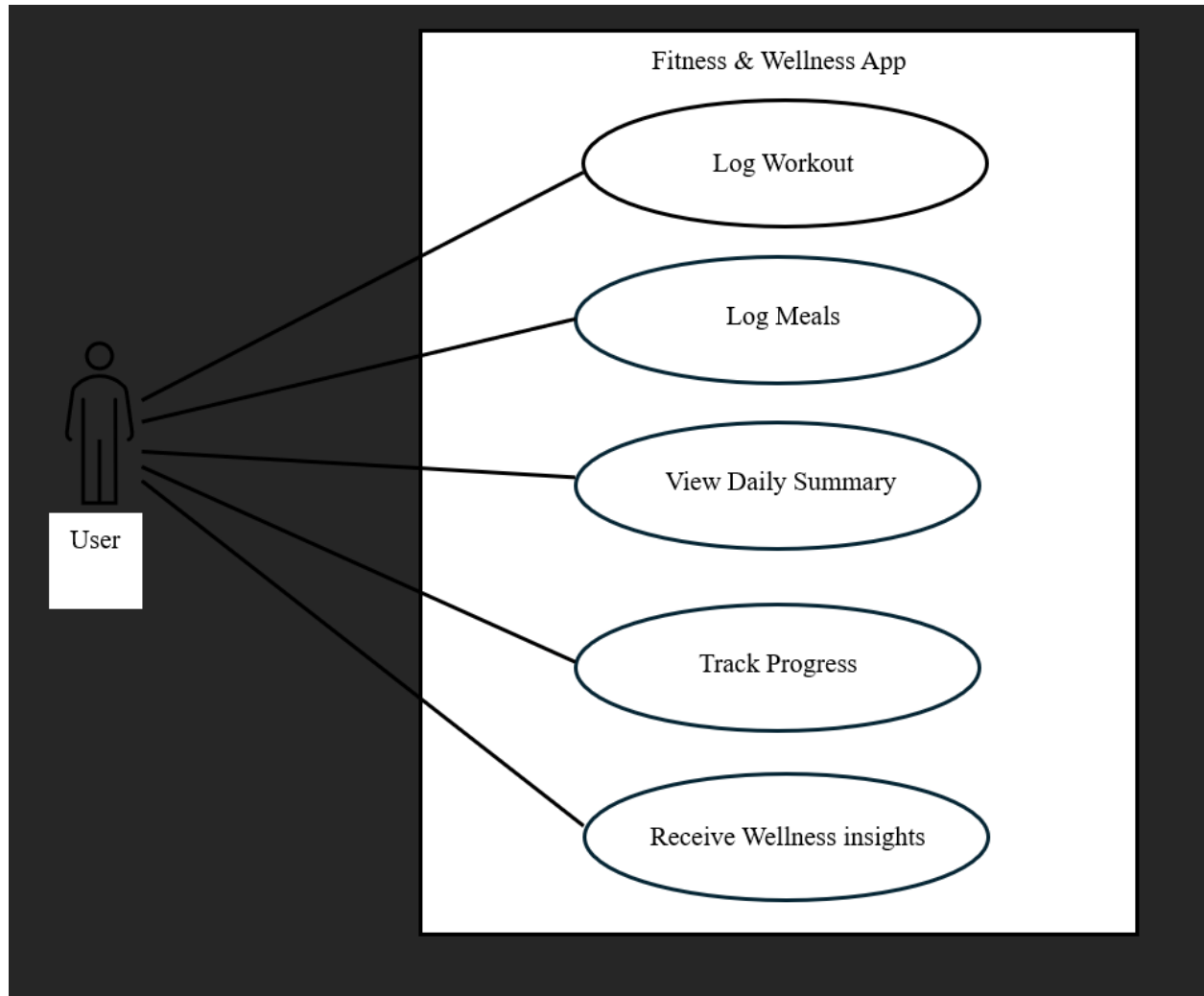
Beyond tracking numbers, this project focuses on motivation, accountability, and long-term engagement. Many people begin their health journey with excitement but lose momentum when progress becomes difficult to see or understand. This application addresses that challenge by offering personalized recommendations, visual progress indicators, and meaningful insights that encourage users to stay committed. Our team plans to make tracking your health successful by creating an experience that feels supportive, intuitive, and genuinely helpful. By combining fitness, nutrition, and wellness into one cohesive system, the project aims to provide a tool that not only records data but actively guides users toward a healthier and more balanced lifestyle. Many individuals struggle to maintain consistent routines because their health information is scattered across multiple apps, each focusing on only one part of their lifestyle. This application brings everything together by integrating physical activity tracking, nutrition monitoring, and wellness insights into a single, accessible platform. The goal is to give users a clear understanding of how their daily choices influence their long-term well-being and to help them make informed decisions through organized, data driven feedback. By presenting information in a way that is easy to understand and personally relevant, the app aims to reduce confusion, increase awareness, and support healthier habits.

## System Architecture

- System Requirements
  - Internet connection
  - Compatible internet browser
- Hardware/Software Environments
  - Chrome-based internet browser
  - Laptop, Desktop PC or mobile device
  - Intel i3 (4<sup>th</sup> gen or newer) CPU
  - 240-500GB SSD
  - 4GB RAM



## UML Diagrams



## Data Design

### 1. User Table

- user\_id (PRIMARY KEY, auto-increment)
- username (unique, not null)
- password\_hash (not null)
- email
- date\_created
- last\_login

### 2. Workout Types Table

- type\_id (PRIMARY KEY, auto-increment)
- type\_name (e.g., Cardio, Strength, Flexibility, HIIT)
- description

### 3. Workouts Table

- workout\_id (PRIMARY KEY, auto-increment)
- user\_id (FOREIGN KEY → User)
- type\_id (FOREIGN KEY → Workout Types)
- workout\_date (not null)
- duration\_minutes
- intensity\_level (e.g., Low, Medium, High)
- notes
- is\_scheduled (boolean — distinguishes logged vs. planned)

### 4. Meals Table

- meal\_id (PRIMARY KEY, auto-increment)
- user\_id (FOREIGN KEY → User)
- meal\_date (not null)
- meal\_type (e.g., Breakfast, Lunch, Dinner, Snack)

### 5. Meal Items Table

- item\_id (PRIMARY KEY, auto-increment)
- meal\_id (FOREIGN KEY → Meals)
- food\_name (not null)
- calories
- protein\_g
- carbs\_g
- fat\_g
- serving\_size

### 6. Fitness Goals Table

- goal\_id (PRIMARY KEY, auto-increment)

- user\_id (FOREIGN KEY → User)
- goal\_type (e.g., Weekly Workouts, Daily Calories, Weight Target)
- target\_value
- current\_value
- start\_date
- end\_date
- status (Active, Completed, Cancelled)

## **7. Relationships Summary**

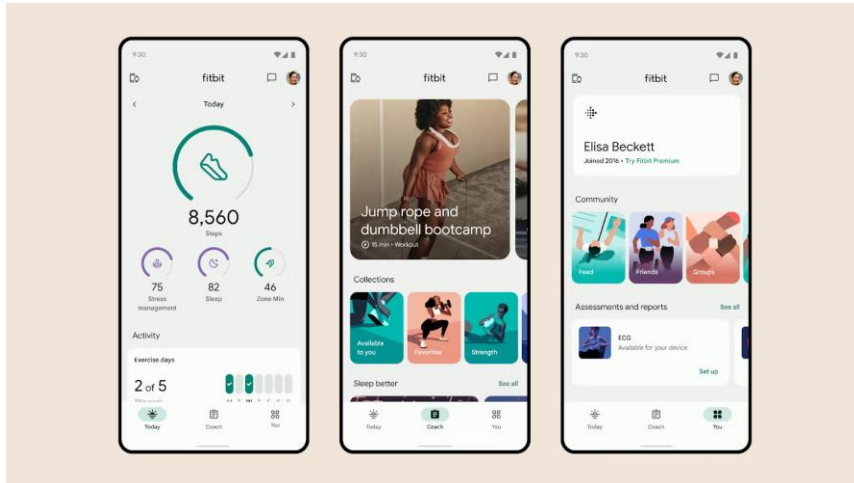
- One User → Many Workouts
- One User → Many Meals
- One User → Many Fitness Goals
- One Meal → Many Meal Items
- One Workout Type → Many Workouts

## **8. Storage Notes**

- All tables stored in local .db files managed by the Java backend
- Backend handles all read/write operations; front-end never accesses .db files directly
- Nutritional totals are calculated at query time from Meal Items, not stored redundantly
- Scheduled vs. completed workouts distinguished by the is\_scheduled flag in the Workouts table

## Interface Design

Generally to be considered early, rough draft concepts, and are subject to significant changes as the team moves forward with the design and implementation process.





## Test Cases

### 1. User Authentication & Profile

- TC-1.1: User can create a new account with valid credentials
- TC-1.2: User cannot create account with missing/invalid fields
- TC-1.3: User can log in with correct credentials
- TC-1.4: User is rejected with incorrect credentials
- TC-1.5: User profile data saves and persists across sessions

### 2. Workout Logging

- TC-2.1: User can log a new workout entry with date, type, and duration
- TC-2.2: User can select from predefined workout types (cardio, strength, flexibility, etc.)
- TC-2.3: Workout entry is rejected if required fields are empty
- TC-2.4: Logged workout appears on the calendar for the correct date
- TC-2.5: User can edit a previously logged workout
- TC-2.6: User can delete a workout entry

### 3. Workout Schedule

- TC-3.1: User can schedule a future workout
- TC-3.2: Scheduled workouts display on the calendar with correct dates
- TC-3.3: User can modify a scheduled workout
- TC-3.4: User can cancel a scheduled workout

### 4. Diet / Meal Tracking

- TC-4.1: User can log a meal entry with food items
- TC-4.2: Nutritional information (calories, protein, carbs, fat) is calculated and displayed
- TC-4.3: Meal entry is rejected if required fields are empty
- TC-4.4: User can edit a previously logged meal
- TC-4.5: User can delete a meal entry
- TC-4.6: Daily nutritional totals are calculated correctly from multiple meals

### 5. Calendar Display

- TC-5.1: Calendar loads and displays the current month by default
- TC-5.2: User can navigate to previous and future months
- TC-5.3: Dates with logged workouts are visually indicated
- TC-5.4: Dates with logged meals are visually indicated
- TC-5.5: Clicking a date displays all entries for that day
- TC-5.6: Calendar correctly displays entries from the database

## **6. Fitness Goals**

- TC-6.1: User can set a new fitness goal (e.g., weekly workout target, calorie limit)
- TC-6.2: Progress toward goals is tracked and displayed accurately
- TC-6.3: User can edit or remove existing goals

## **7. Progress History**

- TC-7.1: User can view a summary of past workouts over a selected time range
- TC-7.2: User can view a summary of past diet entries over a selected time range
- TC-7.3: Historical data matches what was originally entered

## **8. Database / Backend**

- TC-8.1: All user data writes correctly to .db files
- TC-8.2: Data persists after closing and reopening the application
- TC-8.3: Backend handles malformed or unexpected input without crashing
- TC-8.4: Concurrent read/write operations do not corrupt data

## **9. Front-End / UI**

- TC-9.1: All pages load without errors on GitHub Pages
- TC-9.2: Navigation between pages (workouts, diet, calendar, goals) functions correctly
- TC-9.3: Form validation provides clear error messages to the user
- TC-9.4: UI displays correctly across common screen sizes

## Summary

Fitness and wellness goals are very common for people to start making around the beginning of the year, and our app will be able to help with those goals. The system architecture gives us a good design to code off, the UML Diagrams create effective visuals to guide the development process, and our test cases should account for and eliminate most bugs or issues that may arise during development. These should make the development of our application a very manageable and smooth process.