

BMR/Calorie Tracker Project Proposal

Brayden Mavis, CS121

Purpose

This project is intended to demonstrate mastery of concepts learned in CS121, specifically Object-Oriented Programming (OOP), data structures, and algorithm design. The project will focus on tracking Basal Metabolic Rate (BMR) and caloric intake to help users manage their daily nutrition and fitness goals.

Overview

The BMR/Calorie Tracker will allow users to calculate their daily caloric needs based on their BMR and track their calorie consumption. The system will include functionality for users to enter their personal data (age, weight, height, activity level), log their daily food intake, and analyze their progress.

The application will have a command-line interface (CLI) for simplicity but may include a GUI if time permits. It will store user data persistently, allowing users to revisit and update their information over time.

Intended Users

- Individuals looking to track their caloric intake for weight management
 - Fitness enthusiasts aiming to monitor energy balance
 - Users seeking a simple and effective way to calculate BMR and track meals
-

Problem Statement

Many individuals struggle with maintaining a balanced diet due to a lack of understanding of their daily caloric needs. This project aims to provide a simple yet effective way for users to:

1. Calculate their daily calorie requirement based on BMR.
2. Log daily meals and track caloric intake.
3. Analyze trends in their consumption over time.

Technologies & Structures

- **Programming Language:** Java (for object-oriented structure and data management)
 - **Data Structures:** ArrayLists for meal tracking
 - **File Handling:** Text files or serialization for data persistence
 - **OOP Principles:** Encapsulation (user data stored privately), Inheritance (different user types)
-

Use Case Analysis

Use Case 1: Calculating BMR

Actors: User **Flow:**

1. User inputs age, gender, weight, height, and activity level.
2. System calculates BMR using standard formulas.
3. System displays the daily caloric needs.

Use Case 2: Logging Meals

Actors: User **Flow:**

1. User selects 'Log a Meal'.
2. System prompts for food name and calories.
3. System stores entry and updates daily total.
4. User can view total calories consumed.

Use Case 3: Viewing History & Analysis

Actors: User **Flow:**

1. User selects 'View History'.
 2. System displays past calorie logs.
 3. System analyzes trends and suggests adjustments if needed.
-

Data Design

- **User Class** (stores user details)
 - Attributes: name, age, weight, height, gender, activity level
 - Methods: calculateBMR(), updateInfo()

- **Meal Class** (stores meal information)
 - Attributes: foodName, calories, date
 - Methods: getCalories(), displayMealInfo()
 - **Tracker Class** (manages meals and user data)
 - Attributes: list of Meal objects
 - Methods: addMeal(), getTotalCalories(), displayHistory()
-

UI Design

Command-Line Interface (CLI)

- **Main Menu:**
 5. Enter User Information
 6. Calculate BMR
 7. Log a Meal
 8. View History
 9. Exit
 - **User Prompts:**
 - Input fields for user information
 - Selection for activity level
 - Meal logging input
 - **Output Display:**
 - BMR results
 - Daily and historical calorie logs
-

Algorithm

BMR Calculation Algorithm

1. Prompt user for gender, weight (kg), height (cm), and age.
2. Apply BMR formula:
 - Men: $BMR = 88.36 + (13.4 * weight) + (4.8 * height) - (5.7 * age)$
 - Women: $BMR = 447.6 + (9.2 * weight) + (3.1 * height) - (4.3 * age)$
3. Multiply BMR by activity level factor.
4. Display daily calorie needs.

Meal Logging Algorithm

1. Prompt user to enter meal name and calorie count.

2. Store meal in an ArrayList.
3. Update daily total calorie count.
4. Allow user to view calorie history.

History & Analysis Algorithm

1. Retrieve stored meal data.
2. Display calories consumed per day.
3. Provide feedback if consumption is above or below target.

Milestone Plan

Milestone	Task
Week 1	Finalize project scope, create UML diagram
Week 2	Implement User and Meal classes
Week 3	Implement BMR calculation and meal logging
Week 4	Implement calorie history tracking
Week 5	Add data persistence (file handling)
Week 6	Final testing and debugging

UML

