ENTITIES

User

Responsibilities:

- Store username, gender, food preference, exercise preference, risk factors, and personal data

Attributes:

- username
- gender
- foodPreference
- excercisePreference
- riskFactor
- personalData

Methods:

- getUsername()
- getGender()
- getFoodPreference()
- getExcercisePreference()
- getRiskFactor()
- getPersonalData()
- setFoodPreference()
- setExcercisePreference()
- setRiskFactor()
- setPersonalData()

Collaborators:

- UserAnalyzer

Food

Responsibilities:

- Store food information from database

Instance Attributes:

- calories
- garbohydrate (percentage of DV)
- protein (percentage of DV)
- fat (percentage of DV)
- sugar (percentage of DV)
- foodType
- vegetarianFriendly
- nutrientScore
- id

Methods:

- getCalories()
- getCarbohydrate()
- getProtein()
- getFat()
- getSugar()
- getFoodType()
- getVegetarianFriendly()
- getNutrientScore()
- getId()

Collaborators:

- MealPlanGenerator
- FoodAPI
- FoodManager

Exercise

Responsibilities:

- Stores every Workout moves from the database

Instance Attributes:

- type
- majorMuscleExercised (e.g. full body, leg, arm)
- minorMuscleExercised (e.g. glutes, hamstring, biceps)
- equipmentNeeded

Methods:

- getType()
- gerMajorMuscleExercised()
- getMinorMuscleExercised()
- getEquipmentNeeded()

Collaborators:

- WorkoutAnalyzer
- ExerciseManager
- ExerciseAPI

Disease

Responsibilities:

- Stores disease and its symptoms from the database

Instance Attributes:

- disease
- symptoms

Methods:

- getDisease()
- getSymptoms()

Collaborators:

- DiseaseAnalyzer
- DiseaseAPI

USE CASES

UserAnalyzer (Abstract parent Class)

Responsibilities:

- Parent class for all analyzers, which analyzes information from the User class

Methods:

- analyze()
  - Takes in inputs of the type User
  - Manipulates with different information of the user, and returns the calculated output

Collaborators:

- MealPlanGenerator
- User

Child Classes:

- DiseaseAnalyzer
- BMIAnalyzer
- EERAnalyzer

UserAnalyzer

BMIAnalyzer

Responsibilities:

- Analyze the BMI of the user to calculate their value
- Give the comparison of the BMI with the typical averages

Methods:

- analyze()
  - Takes-in inputs of the type User, specifically from User.personalData
  - Outputs the BMI value and compare the results with the health standard

Collaborators:

- User
- RunCommand

UserAnalyzer

EERAnalyzer

Responsibilities:

- Calculate EER (energy requirement per day) using information from UserPersonalData

Methods:

- analyze()
  - Takes in inputs of the type User
  - Outputs the EER value

Collaborators:

- User
- RunCommand

UserAnalyzer

DiseaseAnalyzer

Responsibilities:

- Takes in a list of symptoms that the user is experiencing and outputs potential diseases they could be at risk for based on our dataset

Methods:

- analyze()
  - Takes in inputs of the type User, specifically from User.riskFactor
  - Outputs a list of disease the User might potentially be at risk for

Collaborators:

- User
- RunCommand

UserAnalyzer

WorkoutAnalyzer

Responsibilities:

- Calculates a list of workout moves for users based on their preference for exercising and their energy requirement per day

Methods:

- analyze()
  - Takes in inputs of the type User, specifically from User.exercisePreference
  - Outputs a suggested list of workout moves along with the part of muscles trained and if any equipment is needed

Collaborators:

- User
- Exercise
- RunCommand

MealPlanGenerator

Responsibilities:

- Generate meal plans for user base on user input on FoodPreference, and based on the Energy requirement (measured in calories) of the User

Methods:

- generateMealPlan()
  - Take User and FoodManager as input
  - Output a list of combinations of food within the desired calorie range and fits the user's preferences

Collaborators:

- Food
- FoodManager

FoodManager

Responsibilities:

- A Collection of all food objects
- Passed into MealPlanGenerator to generate Meals

Instance Attributes:

- Food (a collection of all food objects)

Method:

- Vegetarian(): return a collection of all food that falls under the category of vegetarian
- healthyFat(): return a collection of all food that has percentage of fat under 20% of the recommended daily value of intake
- healthySugar(): return a collection of all food that has percentage of sugar under 20% of the recommended daily value of intake
- healthyCarbs(): return a collection of all food that has percentage of carbohydrate under 15% of the recommended daily value of intake
- nutrientScore(): return a collection of food with a nutrient score under a certain number

Collaborators:

- MealPlanGenerator
- Food

CreateUser

Responsibilities:

- Create a user object using the provided arguments: two arrays of String that contains name and gender of the user, and the height, weight, and age of the user

Method:

- createUser: return a user object based on the provided arguments

Collaborators:

- RunCommand
- User

CONTROLLER

RunCommand

Responsibilities:

- Takes in String input that is entered by the user and execute the command if valid

Method

- executeCommand
  - call the corresponding analyzer that runs the functionality the user want

Collaborators:

- CreateUser
- BMIAnalyzer
- EERAnalyzer
- DiseaseAnalyzer
- WorkoutAnalyzer

DRIVER

API

Console

Responsibilities:

- Interacts with the user by outputting String prompts to the terminal, and take in user's input and pass it to the controller, RunCommand

Methods:

- getBasicUserInfo()
- getPersonalUserInfo()
- main

Collaborators:

- RunCommand

API (Interface)

Responsibilities:

- Retrieve the necessary data based on the given dataset and return the corresponding object.

Collaborators:

-FoodAPI

-ExerciseAPI

-DiseaseAPI

API

FoodAPI

Responsibilities:

- Read in the Food and nutrition dataset and store each row in the file as a Food Object

Collaborators:

- Food
- FoodManager

API

ExerciseAPI

Responsibilities:

- Retrieves the exercise/workout moves data and stores each move as an Exercise Object

Collaborators:

- WorkoutAnalyzer
- ExerciseManager

API

DiseaseAPI

Responsibilities:

- Retrieves and read the disease data, storing them in a collection that maps the disease to a list of their corresponding symptoms

Collaborators:

- DiseaseAnalyzer