Chosen Workout

Name of Team: Team Motres

Team Members: Nick Heavener, Rose McCormack, Jake Prince

Date 9/20/24

Abstract

This project was created to provide a free, easy to use, open source, and readily modifiable program to generate unique fitness plans customized to the individual user. Using various information gathered from the user, the software can programmatically select the best exercises for them based on a variety of factors. This includes what fitness equipment and exercise machines are available to them, what kind of results they are trying to achieve, level of exercise experience, and what areas of the body should be focused on. The app even defines details of the workouts, such as how much weight to use, how many reps to do, and how many days of the week to work out. This is done by utilizing specific information about the user like height, weight, caloric intake, and gender to compute the ideal level of rigor. Despite the versatility of our product and the wide range of scenarios it can serve, there are always users and requirements that can not be fulfilled. The only way to provide a truly one size fits all solution is to allow the user to modify the product as they see fit in order to suit their specific needs. In order to facilitate this, our software is distributed under the MIT license and is designed to be as readable and easy to jump into as possible.

Keywords: These keywords will allow the user to look up this project with better context.

- Fitness
- Workout
- Exercise
- Fitness Plan
- Planner
- Gym
- Weightlifting
- Cardio
- Weight Loss
- Weight Gain
- Health

Table of Contents

Introduction and Background

Product Vision

This application is for people needing more experience with workouts and who are interested in losing weight/gaining muscle. The app *Chosen Workout* is a fitness application that gives recommended workouts depending on your BMI, height, age, etc., giving users who may not be familiar with certain exercises instructions on how to perform them. Unlike Fitbit/Apple Watch, our product allows users to input data points, and a workout plan will be chosen from a repository of exercises to continue their physical development.

Personas

Persona 1: Peter, age 19, is a full time college student at Appalachian State University and has a software engineering internship where he sits at a desk all day. He majors in Computer Science and is currently in his second year. He is originally from Wilkesboro and commutes to school every weekday and He has a firm understanding of how to use technology. He is interested in the Chosen Workout because he does not have much knowledge of exercising and wants to exercise more due to long periods where he has to sit in the car.

Persona 2: Carson, age 30, is a plumber at Greensboro. He grew up in the area and has been working at the same place for 10 years after going to trade school out of high school. He is active throughout the day at his job, but feels he should be doing more workouts to target specific areas, as he mostly uses the same muscles at work every day. He has a firm understanding of exercising, as he took weight lifting classes in high school, but is technologically challenged. He is interested in the chosen workout to target the muscles he has not been able to hit while at work.

Scenarios

- Scenario 1: Carson wants to create a personalized workout plan.
    - Carson will have already logged into the app and has it installed on their system. By entering in his username, password, age, height, weight, the number of days wanting to workout, a bodyweight or weightlifting focus, whether he wants to gain, lose or maintain

body weight, his gender, whether he wants a push, pull, legs focus, and his activity level , Carson will retrieve a sample workout plan, along with detailed instructions. When he retrieves the plan, he is able to look at the exercises along with the sets and reps for each exercise that fits around he will be able to use around his busy work schedule.

- Scenario 2: Peter wants to start doing body weight exercises in their workouts.
    - Peter will already have an account with *Chosen Workout* and likely wants to incorporate different exercises into their plan. The user will be able to choose from a selection of recommended exercises based on their current workout preferences.
- Scenario 3: Carson wants to see his plan again
    - Carson will already have an account to *Chosen Workout* and access to the application. Carson wants to see his information again without having to enter in the information twice. By just entering his username and password at the start of the program, he is able to retrieve the workout plan and maintenance calories.

Background

People have a great many reasons for why they exercise. It could be because of health issues, because they want to look better, because they want to gain muscle mass or lose body weight, or for a variety of other reasons. Each of these motives requires a vastly different approach to fitness in order to achieve the desired results. For example, someone preparing for a strongman competition needs plenty of weight lifting and a diet in a caloric surplus, while someone trying to build their endurance should likely incorporate more cardio and run a caloric deficit. These differences compound and become significant, especially with the millions of differences that exist between each person. Due to this, it is very difficult to know what workouts are best for you. Even someone experienced with fitness will struggle with selecting the most appropriate exercises, such as when changing their goals (like going on a “cut”). This difficulty is compounded when you take into consideration all of the other intricacies of a fitness program, such as how much weight to lift, how often and when to work out, and how much to eat. Our product seeks to take some of the headache and guesswork out of this process and provide a solid basis for which to build your fitness routine.

Product Scope

As explained above, there are millions of factors that can go into which workouts are appropriate and how to do them. We attempted to consider the most impactful traits in our selection of exercise, but there are plenty of others that would improve the competency of the product if they could be included. For example, knowing more specific medical information about a user would allow us to better predict how many calories are necessary and what workouts may be potentially dangerous. Having more granularity in the goals the user would like to achieve would enable us to better match the workouts with the person. There is nearly infinite information about the user and their requirements that we could gather that would hone the fitness plans generated further. At some point more data simply requires too much of the user's time and effort to input to justify the diminishing improvements.

Product Features

- User Profile Creation
    - The user would be allowed to create a profile where their information can be stored in a database. This will allow the user to stand out on the app and keep track of their data.
- Custom Workout Plans
    - The user would be allowed to bring exercises in and out of their plan. For instance, a knee injury might eliminate leg workouts for about 3–4 weeks and the user can only work out the upper body. They can change their exercises manually.
- Exercise Instructions
    - Sometimes, certain exercises are obscure or difficult to comprehend intuitively. The user will have an option to display workout instructions for increased accessibility.
- Customization of Workout Focus
    - Some people enjoy body weight workouts more or cannot access a gym, and some people prefer the gym over home workouts. A user can initially start with at-home/body weight workouts, but change them to gym-based workouts if the situation improves.

User Stories

List your user stories here, in user story format. Indicate which feature, or features, each is related to. This is from the personas, scenarios, and features assignment.

- User Profile Creation
    - As a high-profile fitness influencer, I need to have a record of my weekly workout to make sure I am hitting all the exercises.
- Custom Workout Plans
    - As a fitness buff, I am constantly changing my workout routines and fitness goals. As a result, I need a tool to manage all of my different plans and easily switch between them.
- Exercise Instructions
    - As a fitness instructor, I need a tool that can teach me tons of new workouts so that I can teach them to my different students.
- Customization of Workout Focus
    - As someone newly focused on fitness, I haven't decided what to focus on, what equipment to buy, and whether to purchase a gym membership, so I need a tool that can easily switch between workouts based on different levels of equipment.
- Custom Workout Plans, Customization of Workout Focus
    - As a crossfit enthusiast, I need a tool that can provide me workout plans for wherever I am and whatever level of fitness I am at, so that I can push myself to the absolute limit anywhere I might be.
- Custom Workout Plans
    - As a student, I need a tool that can provide me exercise plans that can get me back into working out after taking a break, so that I can temporarily focus on major projects and exams.
- User Profile Creation, Exercise Instructions
    - As a physical therapist, I need a tool that allows me to track my patients fitness and can remind patients of how to do an exercise I assigned if I am not there.

Design and Architecture

High-Level Architecture

Describe the high-level architecture of your system here. One or more diagrams should be added to help illustrate this.

Architectural Styles

We will discuss architectural styles in class. Describe any architectural styles used in your architecture here. Make sure you map these to your system – you should explain how they apply, don't just say "this system uses a repository architecture".

Logical Components

How is your system broken up into logical components that work together? Describe each of these components here.

Class Design

We do not have a specific class design we are following, but we are more likely following the interface class design. Even without an actual interface, we have classes like ClientHelper.java and others that allow programmers to manipulate and organize the Client object or Client objects to increase efficiency of the program and its operations. This situation allows us to hold more freedom whenever we test code and also when we implement the program.

Design Patterns

We are mostly using the Singleton pattern in our project. This is because all operations revolve around the Client object in Client.java and store each client object into an ArrayList structure. For instance, we have helper classes like ClientHelper.java, but do not specifically represent other objects. It is important to simplify code as it allows the programmer to have an easier time refactoring code and it allows the user to understand the overall program better.

Non-Obvious Design Decisions

The main design decision not obvious from our code is that we wanted to create a program with a simple object implementation. This is why we only have a terminal input for user input along with a Client object that stores the information for each user in an ArrayList structure. Sometimes, all you need is simplicity.

Development and Test

Configuration Management

1) Code should be committed once the code you have been working on is working as it is planned to. No matter how small the change is, if it is working as intended, commit it.
2) We are using separate branches to work on features. Once we have made substantial changes to the feature we merge to the main branch.
3) For commit messages, we expect there to be a summary of everything you changed (just a sentence or two).We do not enforce this, but we all try to write down a simple update on what exactly the commit did.
4) If someone commits code that breaks the system, we first catch it on the feature branch and see what we can fix there. We use code reviews to see if the new code functions as it should, and this prevents code that breaks the system on the main branch.

Technology

1. We currently only use libraries that are standard to Java. We use the java.io package to read and write from files in order to manage different workouts/exercises that are chosen by the user. We use Maven to build our project.
2. We enforce a basic readable code style with no specific requirements except for use of the Javadoc comment style to document our code (A brief description of the method/class functionality, an author tag, and a version tag). One member of the team is typically assigned to review style and documentation at each team meeting to ensure consistency.
3. We opted for an open source license, specifically the MIT License, because it allows users to modify and use the free software as much as they would like, without allowing other developers to release closed-off versions based on our project. We believe that fitness should be accessible to everyone, and we also realize that many people have different needs that may not be supported by our product.

Testing

We are testing our code based on the functionality of the code. We are using a terminal-based input which means that we have to test user input along with our file parsing method. The tests do not determine if we can commit or not, but we generally do not commit if our tests are still failing. This is due to Maven not being able to build properly if our tests do not work, even just one test not working allows the build to fail. We will know if we have tested the features of our product once we know the tests run well and also once we have our expected input and output.

Final Results

We managed to implement the user account creation, customized workout plans, and Customization of Workout Focus. We were not able to finish Exercise Instructions. Over the course of the semester, we decided to add the ability to give the user the maintenance calories they must consume, and we were working on switching the exercise instructions from just plain text to a video format. Organizing the project we decided to split the work where Nick and Jake worked on features of the project, while Rose worked on unit testing.

Contributions include:
Nick - File parsing to get workouts, customizing workout plans considering focus and number of days the user is able to workout, and getting the calories from the user.
Rose - Created the test suite, looked into potential database connectivity and media packages/frameworks, created the Singleton for the project, managed feature-driven workflow
Jake - Created license, added workout instruction video links and posted videos, worked on presentation and report