Sam Alcosser
Dr. Juan Arias
Software Development 1
24 April 2019

35/35
Organize the document using the sections in the project description.

Milestone Report

The goal of this project was to create a functional workout tracker. This means having a way for multiple users to keep track of their past workouts as well as see their improvements over time.

The motivation behind this project came about when I was coming back from workouts over the long winter break and I would have notebooks full of past workouts. The problem occurred when I wanted to see my improvement over time, or even more, just be able to see all of the information in a clean, streamlined, and legible way. For this reason, I knew that most other workout tracker applications would either be too complex, were clunky, or required paid subscriptions or a constant barrage of ads in order to reap the full benefits of the application. For this reason, I decided to set out and make a clean, streamlined application in which I could track my workouts and use the software embedded in the app to show my growth over time.

In order to have a full understanding of the project, and for brevity in this report, please follow along at https://www.lucidchart.com/invitations/accept/003e77ad-e665-4b08-9b2a-bf8a555ae625. To start, the user starts off at the login class, which creates a window in java that can be interacted with. From here, the user can choose to either sign in, or make a new account. Making a new account brings the user to an instantiation of the SignUp class shown in the UML diagram. In this page the user can create a new user profile, and before trying to sign up, the user also has the option to check if their desired username has already been taken yet. This is done by querying the database table of the users for any occurrences of that username in the username

fields of all entities. This, along with many other functions in the program is done using Hibernate. It is possible to send SQL queries “directly” through the JDBC, but on large scale projects, this can be clunky. So, Hibernate is used as an in between Java code, and the JDBC.

Now, assuming that the user has either signed in normally, or has signed up and filled out all form fields correctly, they will be transferred to the main homepage along with the information about the signed in user to be used later. This is an instantiation of the Homepage class. Here the user has many options. The user can track a workout, or record an injury, or view past workouts and progress. When a user selects the track new workout button, at that instant a new workout object is created with that timestamp and the id of the user. This information is then sent to a new instance of the tracker class.

Here, a user can pick from a list of many exercises, select the completed sets and reps, along with the weight used. Once the user presses add exercise, the information filled out, along with the id of the workout and the id associated with the selected exercise is sent to a new instance of an exercise. That exercise is then temporarily stored in RAM, and will only be sent, or committed, to the database table once the save workout button is pressed. Also, the user has the option to report an injury, which is just selecting the injured area from a list and submitting it to the user table.

The problem that this program is addressing is creating a simple, purposefully barebones workout tracker that does exactly what the user wants it to do and nothing else. This means omitting unnecessary features like messaging systems or overly developed profile timeline or wall features, and instead getting a very easy to use system that tracks workouts, shows simple metrics and does nothing else. Also, if the app does achieve this, it costs money or has ads to keep it free.

It is easy to search and see that a workout tracker is not a new idea. Although many "good" workouts trackers that allow to track custom workouts cost money. For example, *HeavySet* for iOS devices claims to be free on the app store. Although once opened up, it is clear that unless the user is willing to pay the $10 price premium, the functionality of the application will be greatly reduced. Without paying, the user can only use 3 pre-defined workouts. Since the developers know that people who download these apps will use them for a long time, they know that tacking on a subscription is a great idea financially, but this is terrible for the user.

To use the app, first press the new user button on the front screen. Here fill out the desired forms and feel free to check your username before filling out the whole form. After this and signing up, you can use your username and password to log in. To log a workout, simply press the track workout button. For every exercise done during the workout, select from the dropdown list the exercise you completed, fill out your reps, sets, and weight, then press add exercise. Once all exercises are logged, press save workout. Also, if you get injured, simply go to the record injury button and log your injury by selecting it from the drop down and pressing the save button. To clear, simply press the clear button. To view past workouts, press the workouts button and view them from the list. Also, use the view metrics button to see your progress over time.

This system achieves a very simple goal. Making a free, easy to use, and simple workout tracker. With the purposefully minimal functionality, it makes interacting with the system simple and easy.

Baeldung. "Hibernate One to Many Annotation Tutorial." *Baeldung*, 24 Dec. 2018, www.baeldung.com/hibernate-one-to-many.

luv2code. "Hibernate Tutorial #1 - Hibernate Overview." *YouTube*, YouTube, 1 Apr. 2017, www.youtube.com/watch?v=0hm3QidFwYY.