Fitness Tracker Enhanced

Patrick Quagge

CST-451 Final Architectural Plan

Grand Canyon University

Instructor: Professor Michael Landreth

Revision: 3

Date: 3/14/2024

Abstract:

This document contains the final architectural plan for the Fitness Tracker Enhanced Application. This will go over the direction that the development team has decided to go. As well going over the final concepts for the application

Technical Requirements:

The technical requirements for the development of the is application are as follows:

Development Planning:

The developmental planning of this application is a requirement to achieve a technical understanding of how this application will function. As well as to achieve the main goal for the application. Through said planning the core functionality of the application can be met.

Technical Architecture:

The technical Architecture of the application are as follows:

* Home
* Registration
* Login
* Workout list
  + Read Workout
  + Update Workout
  + Delete Workout
* Goal List(optional based on time)
  + Read Goal
  + Update Goal

Software Testing and Deployment:

Software testing and deployment of the application will be done locally. Once the testing has been completed the final version of the application will be deployed locally to showcase features of the application. This versions source code can always be transmitted to the customer upon final milestone achievement and customer acceptance.

The Technical requirement software tools are as follows:

Operating system: Windows

* This application will be written in and for a windows operating system. As it will be developed to be reactive it can be supported on multiple screens to include mobile devices on their web browser.

IDE: Visual Studio Code

* The source code for this application will be developed in Microsoft Visual Studio code. As well from this IDE the application can be tested and deployed locally.

Frameworks: Express (Backend) React (Front End)

* The frameworks that will be supporting this application will be Express for the back-end development and React for the front-end. These two frameworks are for the JavaScript Language as the code for this application will be written in that selected language.

Database: MySQL

* The database that will be used for this application will be MySQL.
  + As stated in prior submitted documents, if the database connection is an issue with this application a standalone version will be developed to work around this issue.

Design Planning Summary:

This specific development project was conceptualized for the want to improve people’s lives. This comes from the need that has been observed in everyday life. There are many people that do not realize how their overall fitness affects their health. It is believed that one does not go along with another. This cannot be farther from the truth, as your fitness reflects your health. Through this observation it has been ascertained that with perhaps with the right interactive visual aid that people can be helped to achieve a higher level of fitness that can lead to longer more fulfilling lives. Though fitness is not an easy road the idea is to help people see how much they can achieve with a little bit of effort. This application will show the user just how much they can achieve.

Overview of Design Concepts:

The overall design solution for this application focuses on the simple but elegant design philosophy of this project. The Idea is to enhance the user’s fitness experience and allow them to continue to strive for their goals with this fitness tracker. The process flow for the user interaction with the application is as follows:

A diagram of a program

Description automatically generated

As seen the process flow is very straight forward as it is intended to be. The user will first open the application on the web browser on their device. Then if they are a current user, they will login. If not, they will then register. Once registered they will be able to login and view their workout list. From there they can work out following their prior gym day or enter new workouts. They then can update their current or new workout durations and calories burned. If they would like to set goals, they are able to do so. Also, they can update their current goals as they progress on their fitness journey.

Solution Architecture:

The proposed design fits into the overall project structure in a few ways. Firstly, it follows the design principles put forth by following a simple process flow. Home>Register>Login>Read Workouts>Create Workout>User Exercises>Updates Workout. If the user wants to set goals for themselves, they can do that with Home>Login>Read Goals>Create Goal>Update Goal. From the process flow the main goal of the solution can be derived. All of this fits the overall project structure. The Front end of the application will be easy to use and keep the user engaged. Nothing overcomplicated, just a simple fitness tracker to help people reach fitness goals.

Object definitions:

UML Section 1:

A black screen with white text

Description automatically generated

As seen above in this first section of the UML it focuses on the Home, Registration, and Login objects of the application. To register the user will be directed to the registration screen where they will enter their UserName, name, email, and Password. Once that data has been captured, they will be able to login. By entering their login credentials, they will then be directed to the workout list page. See following wire frames for this section:

Home:

User is directed to login or Register.

A screenshot of a computer

Description automatically generated

Register:

If the user is not registered, they can do so from this screen.

A screenshot of a computer

Description automatically generated

Longin:

If they are a current user or they have completed registration they can login

A screenshot of a computer

Description automatically generated

UML Section 2:

A screenshot of a computer screen

Description automatically generated

As Seen above this section of the UML focuses on the Workout and Goals Controllers. These backend aspects of the application will control how the workouts and goals function upon user input of data. The main focus of these is to direct said data to the app.ts component that will then control the routes of the data to the application. The main function is to define the workouts and goals as the users input them. See screens of the wireframes below:

Create Workout:

Users can enter their workout and it will be saved in their current workout list

A screenshot of a computer

Description automatically generated

Workout List:

Once a workout has been entered it will be saved in the workout list where it can be updated to change these values or deleted if the user no longer wants to have this in their workout list.

A screenshot of a computer

Description automatically generated

Demonstration of the application will show that the main Workout list can support more than one workout at a time. As it is a data structure, or Array/ArrayList that can hold more than one piece of data.

Goals:

Much like the workout list the user can create goals for themselves to achieve. They can enter the goals into the goal list and then update them or delete them at their pleasure. There is also the possibility for these to be updated automatically based off workout data entry. That feature will be determined while development is being done on the code.

A screenshot of a computer

Description automatically generated

UML Section 3:

A screenshot of a computer screen

Description automatically generated

This section of the UML focuses on the interfaces that define the workouts and Goals. As seen the structure of these is straightforward defining the structure of the workout and goals object. As seen the workouts have id, name, duration, calories. While the goals have id, name, and goal. Also, these interfaces directly follow the data structures that flow into the database. The tables of the database will follow this same structure.

UML Section 3 cont.:

A screenshot of a computer screen

Description automatically generated

This continued section of the UML defines the structures of the application for the WorkoutRoutes and GoalRoutes. These components of the application allow the data entered by the user to be routed correctly to their intended endpoints.

Site Map:

Below is the finalized site map. Goals are still an optional aspect based on development time

A diagram of a company

Description automatically generated

This directly correlates with the Top-Down Chart provided in the requirements document but can also be seen here(Goals remain Optional)

A diagram of a workflow

Description automatically generated

ERD:

As seen below the finalized ERD is shows now the user’s data is tracked for each workout and goal that they user enters.

A screenshot of a graph

Description automatically generated

System Design Diagram:

Below is the system design diagram for this application. As you can see the user will access the client and then the data will flow from there to the webserver. From the server the user will have the ability to read and write api functions for their workouts and goals to the database. This is the ideal situation and workarounds can be instituted in needed. Once the data is in the database the user can then fetch or read their input data as you can see it flowing bac to the user.

A diagram of a computer

Description automatically generated

Design Planning Summary:

In summary this document outlines all the finalized architectural plans for this application. Using the design principles set forth the application will be built to be easy to use and keep the user on track for their fitness goals. The Proposal document has been updated to include corrections to the prior revision. The Requirements document remains un-unchanged as no technical requirements have been altered. The system logic is as outlined in this document. Based on user data input of their workouts and goals the application will respond by saving their workouts and goals data. This data can be changed by the user and updated at the user’s pleasure. The aspect of automatically updating goals is something the development team may institute if time permits. The main goal of the application will be met before further enhancements can be made.