Breazy Fit

Business Requirements Document (BRD)

Github Link:
https://github.com/ChazArvizu/CECS491_Hexadecimators.git

Team Name: Hexadecimators
Team Leader: Chaz Arvizu
Team Members: Carlsean Claricia, Tania Adame, Tyler Kelsey, Andrew De La Rosa, Sean Iida

Submission Date: 10/05/2022

Version History

Version 1.0 - Initially Created: 09/28/2022

Table of Contents

Page Number

1. Project Summary

This project aims to create a personalized regimen to achieve user’s fitness goals by creating unique features that are geared to effortless tracking of workouts and calories based on user objectives. Users registered as trainees can find other users registered as trainers who are near their location. This app will be up for release next year, where users like daily gym goers can test the product. We are looking to achieve being able to stay on top of goal management, and all milestones created for this web application.

2. Project Objectives

When releasing the web application, we hope to achieve 150 users who are utilizing the application for their fitness needs after the two weeks of release. Along with the users we are hoping to achieve having 50 trainers that are registered and certified after the three weeks of release who specialize in different areas of fitness. Our goal is to have a web application that will be able to fit the needs of users who are looking to better their fitness habits and goals. We will have an organized graphical user interface that will contain a calendar, workout routines, and trainer recommendations. Users will set up their own Breazy Fit profile personalized for themselves.

3. Project Scope

The application that will be delivered is a fitness maintaining application designed to suit the user’s needs in a way that is convenient and customizable. The finished application will be directed towards people who want to maintain a balanced regimen with daily routines, specialized personal help, and who want a more organized experience. The application that will be developed is a single page web application designed to run on Chrome 104.x (64 bit) while using various front-end and back-end languages, such as C#, SQL, and HTML. The data that we are collecting from the users are as follows:

- User’s personal emails
    - Will be used when creating accounts to make each user account unique
- User’s location
    - Will be used to find various nearby fitness events
    - Will be used by trainees to find nearby trainers
- User’s phone number

    - Will be used for users to send their phone number to another user for training purposes
- User’s height and weight for calculating BMI for certain features
    - Will be used to calculate BMI to accurately calculate the amount of calories and macronutrients they should consume.

4. Business Requirements

4.1 Core Components

Login

Function: A user is able to enter in their credentials (email and password) to access their account

User Stories: “As a trainee I would like to log into my account”, “As a trainer I would like to log into my account”

Business Rules:

- Users must use a valid email and password combination to log into their account
- The password must be between 8 - 20 characters and must contain a capital letter and a special character
- User is given five attempts to log into their account, before having to reset their password

Pass Criteria:

- User is able to log into their account with their email and password
- User is not able to login without a correct password
- A user friendly error message is displayed if they input the wrong password

Fail Criteria:

- User is able to log into an account with an invalid password
- User is not able to log into an account with a valid email and password
- User is able to have unlimited attempts to login to an account

Functional Requirements:

- Authenticate when user logs in with their respective password and email for their account
- Fail to authenticate when a user enters the incorrect email and password password sequence

Non-Functional Requirements:

- Website pages should be able to load in 3 seconds on Chrome 104.x(64 bit)
- The system should be able to recognize proper credentials and allow access or deny access based on improper credentials at a quick speed

Logout

Function: A user is able to gracefully exit their account.

User Stories: “As a trainee I would like to log out of my account”, “As a trainer I would like to log out of my account”

Business Rules:

- User must press a logout button to log out of their account
- User will be logged out after 15 minutes of inactivity

Pass Criteria:

- User is able to log out their account, causing them to no longer be able to view their account

Fail Criteria:

- When logging out, user is still able to access account and account data

Functional Requirements:

- User will be logged out after 15 minutes of inactivity
- After logging out by user choice or not the user should be returned to the homepage
- Prevents all access to account specific actions after logging out including on already open pages

Non-Functional Requirements:

- Returns the user to the homepage after logging out within 3 seconds
- Refreshes other pages to reflect having logged out within a quick time

User Access Control

Function: A user is able to access certain features as a trainer or a trainee and access their own private information

User Stories: “As a trainee, I would like to be able to prepare my meals ahead of time, attend fitness events, and hire a trainer”, “As a trainer, I would like to be able to train those looking to hire a trainer and keep notes on my trainee(s)”

Business Rules:

- Users cannot modify another user's account
- User’s account must be able to differentiate between a trainer and trainee
- When searching for trainer, user’s must only be able to see nearby trainers
- Trainers and trainees must have access to differentiating features
- User will have access to their own private information

Pass Criteria:

- Trainers and Trainees are visibly distinguishable on the application
- Trainers and Trainees have access to different features
- User can only see their own private information

Fail Criteria:

- Users cannot see a difference between a trainer and a trainee
- All users have access to every feature regardless of account type
- User can see every users private information

Functional Requirements:

- While creating an account, user can select between a trainer and a trainee
- User can update their own private information

Non-Functional Requirements:

- Modified data should be updated in the database within 3 seconds

User Management

Function: A user is able to reset and recover their account, update their account, and delete their account.

User Stories: “As a user, I would like to reset my forgotten password”

Business Rules:

- User’s account email must be a valid email
- User’s account must be erased if they want to delete their account
- User’s account must be updated with correct data

Pass Criteria:

- Reset password is updated as new account password
- Reset email should have the correct link to reset password
- Deleting an account gets rid of all user data

- Updates to user account are properly shown

Fail Criteria:

- Reset password is not updated as new account password
- Email link to reset password links to wrong page
- Deleting an account keeps user data
- Updates to user account are not shown

Functional Requirements:

- Entering a valid email into a validator sends a reset password link to the associated email
- Resets the password within the database upon success of making a new password
- Update and change user account’s email, username, password, weight, and height
- Delete user account from database along with all data tied to it

Non-Functional Requirements:

- Password gets changed within the database in 2 seconds
- Account updates within the database in 2 seconds

4.2 Unique Features

User Trainer Lookup

Function: Users registered as trainees can search for a personal trainer who is specialized in an area of fitness

User Stories: **“**As a trainee, I would like look for a personal trainer who specializes in weight loss **”,** “As a trainer I would like to verify that I am a professional trainer”

Business Rules:

- User’s account must be able to differentiate between a trainer and trainee via a badge next to trainers name
- When searching for trainer, user’s must only be able to see nearby trainers
- Only verified trainers are able to be contacted by a trainee and vice versa

Pass Criteria:

- Trainers and Trainees are visibly distinguishable on the application
- Trainer search outputs only trainers that are available in the area

- Only certified trainers can be searched and displayed

Fail Criteria:

- Non Certified trainer is displayed
- Trainee and Trainers are able to see the same features
- All trainers in a specialized fitness field are displayed

Functional Requirements:

- Trainees can look up trainers using the search bar
- Trainees can add their trainer schedule to their calendar
- Users general location is used to look up general trainers in the area

Non-Functional Requirements:

- Trainers will be displayed in 2 seconds after the search up
- Database accepts only certified trainers

Event Planning

Function: Users can look through a variety of events that are nearby to them, such as marathons, weightlifting events, and other types of fitness events

User Stories: “As a user, I would like to see marathon events that are happening in my area”

Business Rules:

- Use user location to provide fitness events near them
- Give event recommendations based on user information
- Caloric and Macronutrient Recommendation will give recommendations to get user prepared for events

Pass Criteria:

- Events displayed match the user location
- Event recommendations match user’s event interests
- The correct Caloric and Macronutrient Recommendations are given

Fail Criteria:

- Event search displays events that are miles away from user
- Recommendation events do not match user information
- The wrong Caloric and Macronutrient Recommendations are given

Functional Requirements:

- Users can search up nearby event using the search bar
- Events are displayed in users calendar
- User homepage displays event chosen by user

Non-Functional Requirements:

- Events are displayed in 2 seconds after search
- Refreshed homepage displays all events user will participate in

Meal Scheduling

Function: Users will see a display of a calendar where each day of the week contains the individual days worth of food along with scheduled eating time

User Stories: “As a user, I would like to add my own recipe to the calendar and schedule my meal for every Monday and Wednesday.”

Business Rules:

- Calendar displays all recipe from either the user or database
- Users will be able to plan meal plans up to a month
- Recipe database gives meal suggestions to the user via a filter

Pass Criteria:

- Correct meal plan is displayed on the calendar
- Recipe database correctly imports meal

Fail Criteria:

- Meals are not displayed correctly on the calendar
- Recipe database does not import meals
- Another user is able to modify a users calendar

Functional Requirements:

- Users can schedule meal plans for certain days
- Calendar displays all meals scheduled
- Users can search for recipes via a search bar

Non-Functional Requirements:

- Adding a meal to certain dates updates calendar in 2 seconds
- Retrieving recipes takes less than 5 seconds

Caloric and Macronutrient Recommendation

Function: Users will get caloric and macronutrient recommendations based on the body mass index (BMI) and the body goals of the user

User Stories: “As a user, I would like to get recommendations on how many calories I should intake to lose weight ”

Business Rules:

- Take in user input for body mass index
- Recommendations are based on the fitness goals of the user
- Allows users to rate the recommendations

Pass Criteria:

- User input is correctly displayed in users account
- BMI is correctly calculated and displayed
- Correct recommendations and ratings are showed based on users fitness goals

Fail Criteria:

- User input does not display on users account
- BMI is not correctly calculated
- Application gives out recommendations that do not match the user’s fitness goals
- Ratings do not show up properly

Functional Requirements:

- User account will calculate and display user BMI
- Recommendations will be displayed to user

Non-Functional Requirements:

- Modified data should be modified within the database in less than 5 seconds

Workout Challenge Database

Function: Users can search in a database of different workout routines/regimens to use at home or the gym

User Stories: “As a user, I would like to look up a routine at home for weight loss”

Business Rules:

- Database provides workout routines for use at home or the gym
- Routines will be recommended based on user fitness goals
- Users will be able to submit their own workout regimens

- User can rate workout routines

Pass Criteria:

- Routines that recommended are based on the users fitness goals
- Users submitted workout routine correctly displays in users account
- Workout routines show their accurate rating

Fail Criteria:

- Routines recommended do not meet user fitness goals
- Uploaded workout does not display on users account
- rating s do not show up properly

Functional Requirements:

- Workout routines are displayed on users calendar
- Users can upload and rate workout routines for other users to see

Non-Functional Requirements:

- Workout routines display in two seconds onto the calendar
- Uploaded workout routines are updated in the database in two seconds

Workout/Calories Tracker

Function: Users can keep track of workouts done and the amount of calories they have consumed or lost

User Stories: “As a user, I would like to know how many calories I lost during my monday workout”

Business Rules:

- Keep track of all calories gained and lost
- Keep track of workouts that need to be done and ones that are finished
- Calculate calories burned by using the Metabolic Equivalent (MET) formula

Pass Criteria:

- User is able to write down and save calorie information onto the program
- User is able to download user calorie information as a document or pdf
- Calories burned are displayed accurately

Fail Criteria:

- Information that the user writes down is not saved
- User isn’t able to download their calorie information
- Calories burned are not displayed or not displayed correctly

Functional Requirements:

- Display users recap of calorie intake or loss for the day
- DIsplay users finished workout routines
- Update workout routines after user is finished

Non-Functional Requirements:

- Updating workout routines and calorie information takes 2 seconds

5. Glossary

Body Mass Index (BMI) - Body Mass Index takes in a person's height and weight to determine what weight category they fall into. The equation to calculate BMI is: BMI = 703 * ((weight(lb.)) / (height(in.))^2).
Macronutrient - includes the main three nutrients your body needs for energy; carbohydrates, fat and protein.
Metabolic Equivalent (MET) formula - Each exercise has a corresponding MET value which can be used in the following formula to calculate the calories burned:
calories burned per minute = MET x 3.5 x ((body weight(lb)) * 0.454) / 200.