**Software Requirements and Design Document**

**For**

**Group 11**

Version 3.0

**Authors**:

Tristan F

Kiara B

Jared R

Kaedon H

Jeffrey M

# **1.** **Overview**

The KitchGym is a fitness/health website that allows you to record your old workouts as well as create new ones. The results of these workouts can then transfer over to the health portion of the website where meals can be created depending on your required calorie intake.

# **2.** **Functional Requirements**

**High Priority**

1. Users must be able to create custom workouts based on desired target muscle groups that will then be saved to a database.
2. Users must be able to save their created workouts to a database and view them in another activity.
3. Website must be able to calculate the calories of food that the user inputs via the “Input Meal” button. This is based upon the given macros and uses this equation:  
    (4 \* protein macros) + (4 \* carb macros) + (9 \* fat macros)
4. Website must include a daily calorie tracker.
5. Must be functionality between activities in the website.

**Medium Priority**

1. A page on the site must display a visual progress report that includes weight changes and daily macros.
2. Allow for users to sign-up and log-in to website

# **3.** **Non-functional Requirements**

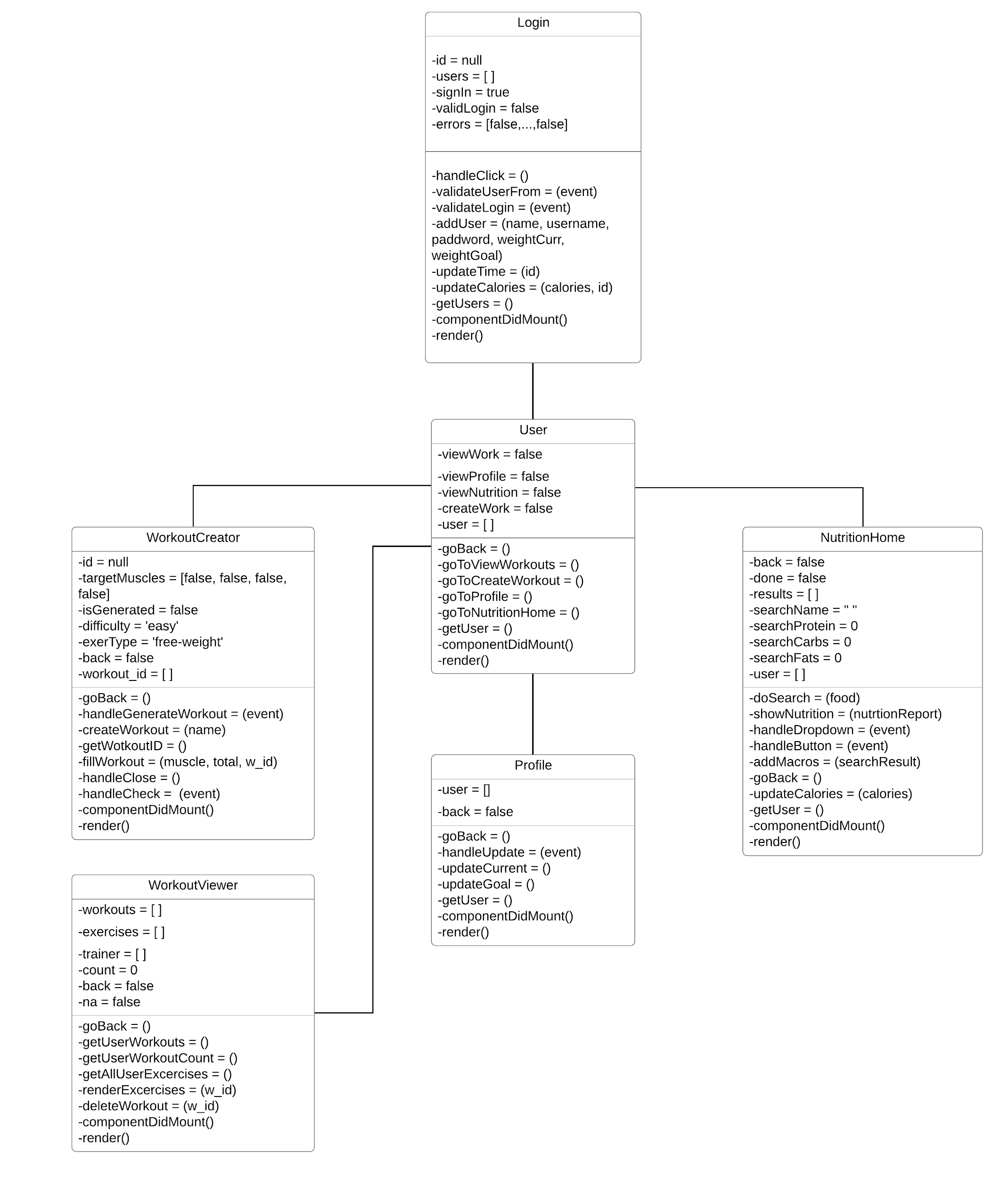
1. The website should be simple to access and navigate. The color scheme should also provide ease of readability (dark background, bright text/buttons).
2. All data will be stored in remote MySQL databases to reduce the amount of local storage space used.
3. Website should be compatible across multiple browsers (Chrome, Firefox, Safari, Edge).
4. NutritionFacts is a JavaScript wrapper for the USDA Food database API.  
   The USDA website can be found here: <https://fdc.nal.usda.gov/api-guide.html>  
   The GitHub can be found here: <https://github.com/eliashussary/nutrition-facts>
5. Pre-created exercises will be stored in the database, ensuring users will not be overwhelmed with confusing or more advanced exercises.
6. In the event that an API or database is unable to be accessed, the user will be notified by an error message.
7. Any code should be well documented and readable.
8. Website should be able to load in under 5 seconds on access. Transitions between pages should be as instant as possible.
9. Any caloric calculations will be rounded to the nearest whole number.
10. Make website look nice (color scheme, animations, icons, etc)

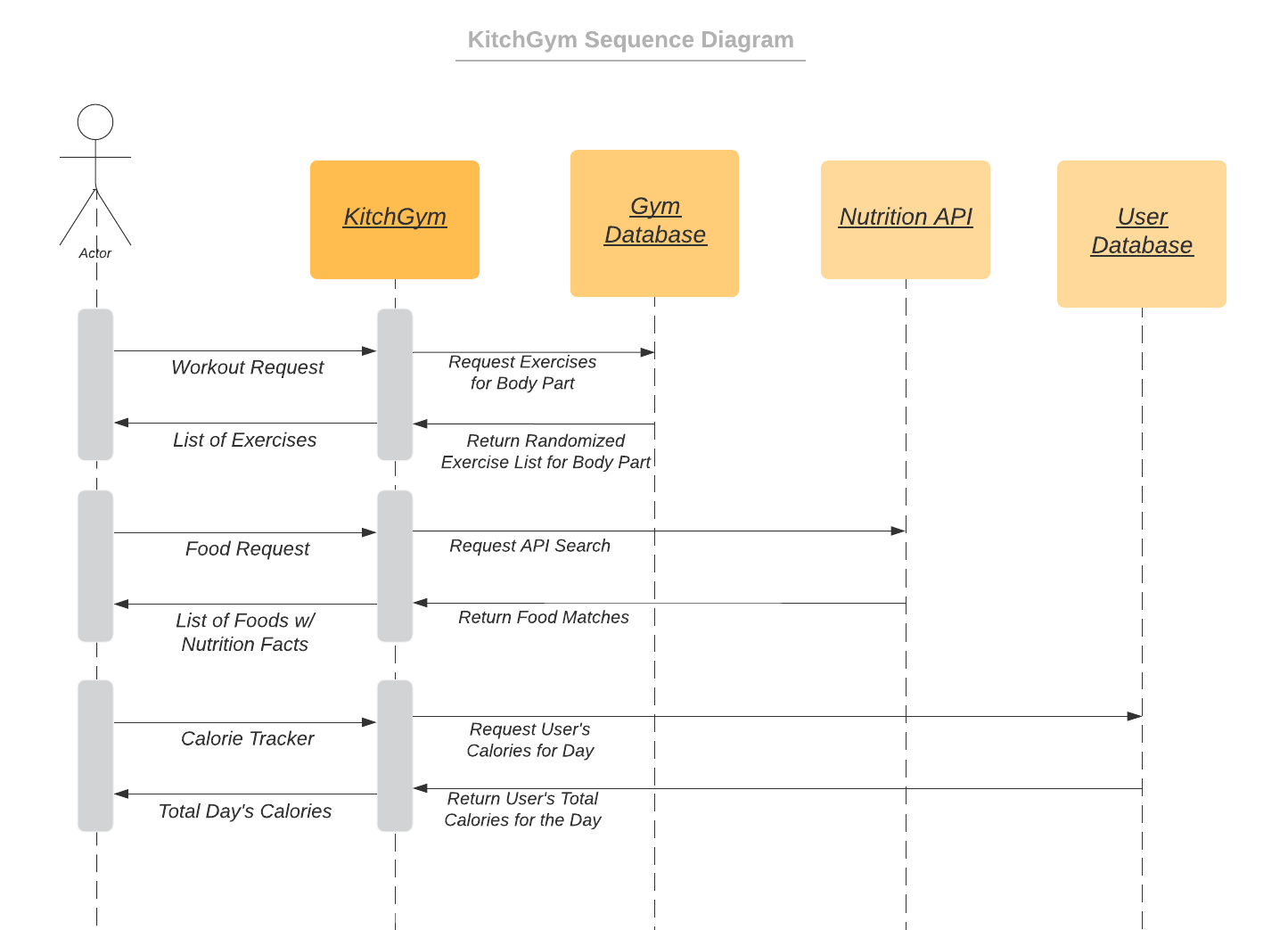
# **4.** **Use Case Diagram**

# 

1. Name: Login
   1. Participating actors: User
   2. Entry condition:
      1. User presses “Login”
   3. Exit condition:
      1. User is logged in
   4. Flow of events:
      1. User types in login information
      2. Presses “Login” button
      3. User is logged in
   5. Special Requirements: User has gone through Sign Up
2. Name: Sign Up
   1. Participating actors: User
   2. Entry condition:
      1. User presses: “Sign Up”
   3. Exit condition:
      1. User creates an account
   4. Flow of events:
      1. User presses Sign Up button
      2. User inputs information
      3. User presses Sign Up button
      4. An account is created
   5. Special Requirements: None
3. Name: Create Workout
   1. Participating actors: User, Exercise API
   2. Entry condition:
      1. User presses “Create Workout”
   3. Exit condition:
      1. Workout is generated
   4. Flow of events:
      1. User presses “Create Workout”
      2. User checks target muscle groups for this workout
      3. KitchGym accesses the database and retrieves exercises matching the desired muscle groups
      4. Workout is generated and displayed to User
      5. User does the workout
   5. Special requirements: User has an account
4. Name: Input Macros
   1. Participating actors: User, USDA FoodData API
   2. Entry condition:
      1. User presses “Input Macros”
   3. Exit condition:
      1. Calories are re-calculated
   4. Flow of events:
      1. User presses “Input Macros”
      2. User searches for food item and inputs amount of that results
      3. KitchGym calls USDA FoodData API to retrieve information for the food eaten
      4. Total calories is calculated and updated on the User on the Nutrition homepage
   5. Special requirements: User has an account
5. Name: Saved Workouts
   1. Participating actors: User
   2. Entry condition:
      1. User presses “Created Workouts”
   3. Exit condition:
      1. Workouts are displayed
   4. Flow of events:
      1. User presses “View Workouts”
      2. Saved workouts are displayed on ViewWorkouts page
   5. Special requirements: User has an account

# **5.** **Class Diagram and/or Sequence Diagrams**





# **6.** **Operating Environment**

The application will operate on an internet browser. The website is functional across Chrome, Firefox, Safari, and Edge. These browsers were chosen because they are the most commonly used and accessible browsers.

# **7.** **Assumptions and Dependencies**

* The browser being used to access the website is up to date
* The utilized APIs are going to stay public
* The user’s computer/browser will be able to connect to the internet when using KitchGym
* The users will only want to work out the main muscle groups, and have knowledge of basic exercises such as: bicep curls, squats, etc.
* Users actively input their food intake in order to accurately calculate calories.