**Calisthenics Workout Application**

**Software Requirements Specification**

**Version #2**

**Team Number: 14**

**Project Manager: Tyler Walston**

**Mentor: Subbu Devarajan**

**Team Members: Alexandra Craig, Michael Mears, Thomas Garland, Cody Selph, In Yip**

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| 1 | Tyler Walston, Alexandra Craig | Initial draft of specifications | 10/16/22 |
| 1.1 | Tyler Walston | Revised draft with project team expanding on application specifications | 10/23/22 |
| 2 | Tyler Walston, Alexandra Craig | Added implementation of high-level design | 10/30/22 |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
| Subbu Devarajan | Version 1 – Initial draft | 10/19/22 |
|  |  |  |

**Table of Contents**

1. Introduction

1.1. Project Objectives

1.2. Project Scope

1.3. Project Overview

1.4. Project Management/Collaboration

2. General Overview

2.1. Major Features

2.2. Business Requirements

2.3. Scope of Release

3. Requirement Specifications

3.1. Functional Requirements

3.2. Nonfunctional Requirements

4. High-Level Design

4.1. Security

4.2. Hardware/Software

4.3. User Interface

4.4. Architecture

4.5. Database

4.6. Top-Level Classes

4.7. Data Flow and States

4.8. Reports

5. Appendix

5.1. Interface Requirements

5.2. Data Flow Diagrams

5.3. Wireframes

**1. Introduction**

1.1. Project Vision

To create a free mobile calisthenics application to fill the current gap in the mobile workout application market. The app would generate calisthenics workouts based on a user’s level of experience and desired muscular focus and would also encourage users to adjust exercises within prescribed workouts to better match their individual goals. The app archives past workouts (exercises, target goals, actual results, etc.) for the user’s review, charts progress graphs, and allows users to plan future workouts by designing and implementing workout templates. These templates, based on past workouts, reflect the user’s progression to meeting their goals. The application will also include two monthly paid subscription options that provide different levels of access to professional trainers and customized workouts.

1.2. Project Scope

The application’s focus is not to educate users on calisthenics in general (history, particular benefits) or muscular anatomy, but to demonstrate the correct way to perform a calisthenics exercise and to reflect the number of sets and reps of that exercise that matches the experience level chosen by the user. The application will not include any social aspect, will not be connected to any social platform, and will not send out any notifications to the user. The application is a tool for creating, performing, and recording calisthenics workouts based on a user’s goals only. “Goals” refers to a user’s target number of sets and reps for given exercises, not larger goals such as weight loss. The free portion of the application will not address larger fitness goals. The current scope will not include ads, a purchase of the application, or subscription payments.

1.3. Project Overview

By assigning a fitness level (beginner, intermediate, advanced) and category (full body, legs, arms, etc.) to a specific calisthenics exercise with a specific number of sets and reps, the workout application will be able to identify the appropriate exercises to add to a user’s workout. This is based upon the fitness level they’ve selected and their desired category focus. The number of exercises added to a user’s workout, as well as exercise duplication within a workout, will be determined by the category selected.

1.4. Project Management

This project aims to use ClickUp as the main collaboration resource. It will also use Microsoft Word and Microsoft Teams as needed for communication and documentation.

**2. General Overview**

**Major Features**

|  |  |
| --- | --- |
| FE-1 | Create a member account to generate and save workouts |
| FE-2 | Have a workout generated from given workout category and level |
| FE-3 | View, edit, and reuse prior workouts |
| FE-4 | Create workout templates from prior workouts and save with desired name |
| FE-5 | Create multiple future workouts from prior workouts and templates and save by name or date to be used |

**Business Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Description | MOSCOW |
| BR1 | We will track member ages, gender identification, and location to for user analysis within one year. | M |
| BR2 | We will monitor how often each workout category is chosen within one year. | M |
| BR3 | We will monitor how often each workout level is chosen within one year. | M |
| BR4 | We will monitor which exercises are consistently removed from workouts by members and begin making adjustments within a year. | M |
| BR5 | We will increase our number of workout levels from three to five within two years. | S |
| BR6 | We will add two workout subcategories within three years. | C |
| BR7 | We will increase the number of exercises in each category such that a member can have a unique workout everyday within five years. | C |
| BR8 | We will add an alternative exercise for every current exercise that requires equipment within one year. | M |
| BR9 | We will add a monthly subscription option that gives members access to advice from professional trainers within two years. | S |
| BR10 | We will add a monthly subscription option that gives members the ability to have customized workouts created for them by professional trainers to meet their specific needs and goals within three years. | C |

**Scope of Release**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Sprint 1** **(Must)** | **Sprint 2 (Must)** | **Sprint 3 (Should)** |
| Member Account |  | Member Registration  Log in, Log out.  Delete Account.  Change Workout Level | Member Workout:  Select Category  Generate (same function as guest)  Confirm Email  Confirm Phone Number  Forgot Password  Change Password |
| Provide Workout | Guest Workout:  Select Level and Category  Generate Read-Only Workout  Close Workout. | Reorder, Remove Exercises  Edit Exercise Targets.  Record Exercise Actuals. | Give Guest Option to Register  Change Workout Level  Change Workout Category  Request New Workout |
| Workout Log | Not Implemented | Not Implemented | View Prior Workouts |
| Workout Template | Not Implemented | Not Implemented | Not Implemented |
| Future Workout Queue | Not Implemented | Not Implemented | Not Implemented |

|  |  |
| --- | --- |
| **Feature** | **Only If Time Allows (Could)** |
| Member Account | Fully Implemented |
| Provide Workout | Fully Implemented |
| Workout Log | Sort, Filter, Edit Prior Workouts |
| Workout Template | Create, Edit, Delete, View, Use Template |
| Future Workout Queue | Create, Edit, Delete, View, Use Future Workout |

**3. Requirement Specifications**

3.1 Functional Requirements

**All User Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Requirement ID | User Requirements | MOSCOW |
| Member Account Registration | | | |
|  | UR1 | Create Member Account | M |
|  | UR2 | Confirm Member Phone Number | S |
|  | UR3 | Confirm Member Email Address | S |
|  | UR4 | Password Protect Account | M |
|  | UR5 | Optional Demographics | M |
|  | UR6 | Set Workout Level | M |
|  |  |  |  |
| Member Account | | | |
|  | UR7 | Delete Member Account | M |
|  | UR8 | Log into Account | M |
|  | UR9 | Log out of Account | M |
|  | UR10 | Change Password | S |
|  | UR11 | Change Workout Level | S |
|  | UR12 | Edit Demographics | S |
|  | UR13 | Change Email Address | S |
|  | UR14 | Confirm Changed Email Address | S |
|  | UR15 | Change Phone Number | S |
|  | UR16 | Confirm Changed Email Address | S |
|  |  |  |  |
| Other Member Account | | | |
|  | UR17 | Provide Forgot Password Process | S |
|  | UR18 | Allow User to Choose Between Email and Phone Number for Forgot Password Process | S |
|  |  |  |  |
| Guest Workout | | | |
|  | UR19 | Can Choose Workout Level | M |
|  | UR20 | Can Choose Workout Category | M |
|  | UR21 | Generate Workout | M |
|  | UR22 | Save Workout Data with Immediate Registration | S |

|  |  |  |  |
| --- | --- | --- | --- |
| Member Workout | | | |
|  | UR23 | Can Choose Workout Category | S |
|  | UR24 | Generate Workout | S |
|  |  |  |  |
| User Workout | | | |
|  | UR25 | Reorder Exercises in the Workout | M |
|  | UR26 | Remove Exercises in the Workout | M |
|  | UR27 | Edit the Target Sets and Reps | M |
|  | UR28 | Record Actual Sets and Reps | M |
|  | UR29 | Change Level for Specific Workout | S |
|  | UR30 | Change Category for Specific Workout | S |
|  | UR31 | Request Different Workout | S |
|  | UR32 | Watch Visual Demonstration of Exercise | M |
|  | UR33 | Workout Data Auto-Saved During Workout | S |
|  | UR34 | Completed Workouts Automatically Saved to Log | S |
|  | UR35 | Cancel Workout | M |
|  |  |  |  |
| Workout Log | | | |
|  | UR36 | View Prior Workouts | M |
|  | UR37 | Sort Prior Workout Log | C |
|  | UR38 | Filter Prior Workout Log | C |
|  | UR39 | Select Prior Workout | C |
|  | UR40 | Edit Prior Workout | C |
|  | UR41 | Remove Prior Workout | C |
|  | UR42 | Use Prior Workout for Current Workout | C |
|  | UR43 | Use Prior Workout to Create Template | C |
|  | UR44 | User Prior Workout to Create Future Workout | C |
|  |  |  |  |
| Workout Templates | | | |
|  | UR45 | View a Template | C |
|  | UR46 | Create a Template | C |
|  | UR47 | User Names Created Template | C |
|  | UR48 | Edit a Template | C |
|  | UR49 | Save Edited Template as New Template | C |
|  | UR50 | Remove a Template | C |
|  | UR51 | Use a Template for Current Workout | C |
|  | UR52 | Use a Template to Create Future Workout | C |
|  |  |  |  |
| Workout Queue | | | |
|  | UR53 | View Future Workouts | C |
|  | UR54 | Create a Future Workout | C |
|  | UR55 | User Can Name Created Future Workout by Name or Associate with a Date | C |
|  | UR56 | Edit a Workout | C |
|  | UR57 | Save Edited Template as New Future Workout | C |
|  | UR58 | Remove a Future Workout | C |

**Feature: Member Account**

|  |  |
| --- | --- |
| User Story ID | User Story |
| US-1 | As a potential member, I want to be able to create a member account so I can track my workouts. |
| US-2 | As a potential member, I want to be able to choose between communication by email or phone number so I can pick what is convenient for me. |
| US-3 | As a member, I want to be able to create a strong password with definitive password requirements so I can ensure the safety of my account. |
| US-4 | As a potential member, I only want to give the necessary required personal information, so that my personal information is safe. |
| US-5 | As a member, I want to be able to edit my information in a single location so that I do not have to constantly move between app screens. |
| US-6 | As a member, I want to stay signed into my account unless I choose to log out, so that I can use the application quicker. |
| US-7 | As a member, I want to be able to delete my account and associated information so that if I no longer want to use the application, my information is deleted. |

**Member Account Use Cases**

|  |  |
| --- | --- |
| **UC-1** | **Register Account** |
| Primary Actor | Potential Member (PM) |
| Trigger | Potential Member indicates that they want to register for an account |
| Preconditions | PRE-1 Application has been downloaded onto device  PRE-2 Application has been opened on device  PRE-2 Potential Member not already registered |
| Postconditions | POST-1 Account Information stored in the database |
| Normal Flow | 1. PM chooses “Register” option  2. PM enters required information: email address, phone number, password, password (second time for confirmation)  3. PM selects a workout level  4. PM enters optional demographics: year of birth, gender identification, and zip-code  5. PM receives six-digit code in email  6. PM enters code into application  7. PM receives six-digit code text on phone  8. PM enters code into application  9. Member Account Page is displayed |
| Alternative Flows | N/A |
| Exceptions | E1 Email Address is already associated with an account  1. Application informs PM that email address is already in use  2a. PM cancels registration process  2b. PM supplies a different email address  E2 Phone Number is already associated with an account  1. Application informs PM that phone number is already in use  2a. PM cancels the registration process  2b. PM supplies a different phone number  E3 Six-digit code entered is incorrect  1. Application informs PM that code is incorrect  2a. PM attempts to enter the number again  2b. PM requests that a new code be sent  2c. PM cancels registration |
| Priority | High |
| Other Info | 1. PM will be able to cancel the registration process at any time  2. PM will be able to navigate the registration steps with back and next links |

|  |  |
| --- | --- |
| **UC-2** | **Change Email Address** |
| Primary Actor | Member |
| Trigger | Member indicates in their member profile that they want to change their email address |
| Precondition | 1. Member is logged into their account |
| Postconditions | 1. Member’s saved email address is updated in the database |
| Normal Flow | 1. Member indicates they want to go to their profile  2. Member indicates they want to edit their profile  3. Member enters a new email address  4. Member saves change.  5. A six-digit code is sent to the email address  6. Member enters in six-digit code  7. Confirmation message displayed |
| Alternative Flows | N/A |
| Exceptions | E1 Email Address is already associated with an account  1. Application informs member that email address is already in use  2a. Member cancels edit  2b. Member enters a different email address  E2 Six-digit code entered is incorrect  1. Application informs member that code is incorrect  2a. Member attempts to enter the number again  2b. Member requests that a new code be sent  2c. Member cancels the edit |
| Priority | Medium |
| Other Info | N/A |

|  |  |
| --- | --- |
| **UC-3** | **Change Phone Number** |
| Primary Actor | Member |
| Trigger | Member indicates in their member profile that they want to change their phone number |
| Precondition | 1. Member is logged into their account |
| Postconditions | 1. Member’s saved phone number is updated in the database |
| Normal Flow | 1. Member indicates they want to go to their profile  2. Member indicates they want to edit their profile  3. Member enters in a new phone number  4. Member saves change  5. A six-digit code is sent to the phone number  6. Member enters in six-digit code  7. Confirmation message displayed |
| Alternative Flows | N/A |
| Exceptions | E1 Phone Number is already associated with an account  1. Application informs member that phone number is already in use  2a. Member cancels the edit  2b. Member supplies a different phone number  E2 Six-digit code entered is incorrect  1. Application informs Member that code is incorrect  2a. Member attempts to enter the number again  2b. Member requests that a new code be sent  2c. Member cancels the edit |
| Priority | Medium |
| Other Info | N/A |

|  |  |
| --- | --- |
| **UC-4** | **Change Workout Level Settings** |
| Primary Actor | Member |
| Trigger | Member indicates in their member profile that they want to change their workout level |
| Precondition | 1. Member is logged into their account |
| Postconditions | 1. Member’s saved workout level setting is updated in the database  2. All future workouts will be generated using the new level |
| Normal Flow | 1. Member indicates they want to go to their profile  2. Member indicates they want to edit their profile  3. Member chooses a different workout level  4. Member saves the change |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | Medium |
| Other Info | N/A |

|  |  |
| --- | --- |
| **UC-5** | **Forgot Password** |
| Primary Actor | Member |
| Trigger | Member indicates that they have forgot their password on the login screen |
| Precondition | 1. Member has already registered |
| Postconditions | 1. The Member’s saved password is updated  2. Member is logged into account |
| Normal Flow | 1. Member indicates they want to log in on the home screen  2. Member indicates they have forgotten their password on the login screen  3. Request for email address is displayed  4. Member enters email address  5. Options for where to have six-digit code sent, email or phone number, is displayed  6. Member indicates where they would like the six-digit code sent  7. The six-digit code is sent  8. Member enters the six digit-code  9. Request for new password is displayed  10. Member enters new password twice (second for confirmation)  11. Member submits new password |
| Alternative Flows | N/A |
| Exceptions | E1 Six-digit code entered is incorrect  1. Application informs Member that code is incorrect  2a. Member attempts to enter the number again  2b. Member requests that a new code be sent |
| Priority | Medium |
| Other Info | N/A |

**Member Account Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR-1 | The user will need to supply their email address, phone number, password, and workout level | M |
| FR-2 | The user must input a password of at least 8 characters with a letter, number and special character (i.e. @#$%). | M |
| FR-3 | The system will send the new account user a confirmation key to their email. The key will be valid for 1hr after being sent. | S |
| FR-4 | The system will send the new account user a confirmation key to their phone. The key will be valid for 15mins after being sent. | S |
| FR-5 | The system will check for accounts already in the database to see if the email and phone number are unique. If not, the system will reject that particular field. | M |
| FR-6 | The system compares what is entered into the database to compare validate a match and login the current user. | M |
| FR-7 | The user inputs a new email into their account and a new confirmation code is sent to the updated email to validate. The system also sends an update notice to the old email with a link to override the email update in case of a bad actor. | S |
| FR-8 | If the user forgot their password, they will be sent an email with a substitute password and after login in the user will be prompted to update their current password. If the user chooses to update their password while logged in they will update their current password and an email will be sent to ensure that the correct user changed the password and if not there will be a link to reverse the change. | S |

**Feature: Create Workout**

|  |  |
| --- | --- |
| User Story ID | User Story |
| US-1 | As a potential user, I want to try a workout before creating an account to see if I would use and enjoy it. |
| US-2 | As a user, I want the application to create a workout for me so I can spend my time performing the workout instead of thinking about it. |
| US-3 | As a user, I want to determine the focus of my workout so that I’m using my time for what is important to me. |
| US-4 | As a user, I want to be able to adjust the order of the exercises to better match my needs. |
| US-5 | As a user, I want to remove an exercise that I cannot perform from my workout so the recorded workout matches what I actually accomplished. |
| US-6 | As a user, I want to adjust the prescribed target sets and reps of an exercise to better reflect my goals. |
| US-7 | As a user, I want to record my actual sets and reps to record how close I am to my goal. |
| US-8 | As a user, once I see the workout generated, I want to be able to request a new one with the same or different level and category until I am satisfied with the result. |
| US-9 | As a calisthenics beginner user, I want the workout exercises to be visually demonstrated so I am confident that I am performing them correctly. |
| US-10 | As a user, if I have to stop a workout, I want the option to cancel instead of close the workout so I can start fresh the next time. |

**Create Workout Use Cases**

|  |  |
| --- | --- |
| **UC-1** | **Generate Guest Workout** |
| Primary Actor | Potential Member (PM) |
| Trigger | Potential member indicates on the application home screen that they want to use the application as a guest |
| Precondition | 1. Application has been downloaded to a device  2. Application has been opened on the device |
| Postconditions | 1. PM can use all features on workout screen |
| Normal Flow | 1. PM indicates they want to use the application as a guest  2. The workout level options are displayed  3. PM chooses a workout level option  4. The workout categories are displayed  5. PM chooses a workout category  6. The workout is generated and the list of exercises displayed |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | High |
| Other Info | 1. The PM will be able navigate through the steps using the back and next links  2. The workout screen for a guest will have a registration option. If the PM registers directly after their workout, the workout will be saved. |

|  |  |
| --- | --- |
| **UC-2** | **Generate Member Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to start a new workout |
| Precondition | 1. Member is logged into their account |
| Postconditions | 1. Member’s workout log will automatically be undated with workout once completed |
| Normal Flow | 1. Member indicates they want to start a new workout  2. The workout category options are displayed  3. The workout is generated and the list of exercises displayed |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | High |
| Other Info | 1. The workout level associated with the member’s account will be used as a factor when generating the workout.  2. The member will be able to navigate back to the workout category options from the workout screen  3. Member will have the option to cancel the workout at any time  4. Member will have be ability to indicate that the workout has been completed |

|  |  |
| --- | --- |
| **UC-3** | **Edit Workout** |
| Primary Actor | Member |
| Trigger | Member indicates they wish to change the workout they are currently using |
| Precondition | 1. Member is logged into their account  2. Member has generated a new workout or opened a saved workout from log, templates, or the queue |
| Postconditions | N/A |
| Alternative Flows | Remove an Exercise  1. Member chooses an exercise in the workout  2. Member indicates they want the exercise removed  3. The exercise is no longer displayed in the current workout  Reorder the Exercises  1. Member chooses an exercise in the workout  2. Member indicates that they would like in to be the last exercise of the workout  3. The exercise is removed from its current position  4. The rest of the exercises move up in position  5. The chosen exercise is placed at the end position |
| Exceptions | N/A |
| Priority | High |
| Other Info | 1. The change in order or removal of an exercise will be reflected in the workout saved to the member’s workout log |

|  |  |
| --- | --- |
| **UC-4** | **Edit Exercise** |
| Primary Actor | Member |
| Trigger | Member indicates they want to edit an individual exercise within the current workout |
| Precondition | 1. Member has generated a new workout or opened a saved workout from log, templates, or the queue |
| Postconditions | 1. The database will be automatically updated to reflect the member's exercise edits |
| Normal Flow | 1. Member indicates which exercise they want to change  2. Member indicates that they want to edit  3. Member increases or decreases the target number of sets for the exercise  4. Member increases or decreases the target number of reps for the exercise |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | High |
| Other Info | 1. The change in the number of target sets and target reps of an exercise will be reflected in the workout saved to the member’s workout log  2. Member will be unable to decrease the number of target sets and target reps to below 0. |

|  |  |
| --- | --- |
| **UC-5** | **Record Exercise Actuals** |
| Primary Actor | Member |
| Trigger | Member indicates they want to edit an individual exercise within the current workout |
| Precondition | 1. Member has completed the sets and reps for an exercise |
| Postconditions | 1. The database will be automatically updated to reflect the member's exercise edits |
| Normal Flow | 1. Member indicates which exercise they want to change  2. Member indicates that they want to edit  3. Member increases the number of actual sets to match what they completed  4. Member increases the number of actual reps to match what they completed |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | High |
| Other Info | 1. The default value of the actual sets and actual reps will be 0.  2. If the member records incorrect actuals during the workout, the member will have the option to correct the saved numbers in the prior workout log  3. Member will be unable to decrease the number of actual sets and actual reps to below 0. |

|  |  |
| --- | --- |
| **UC-6** | **Request New Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want a new workout to be generated instead of using the one currently being displayed |
| Precondition | 1. Member has generated a new workout or opened a saved workout from log, templates, or the queue |
| Postconditions | 1. The database is updated to reflect any changes to level and category as well as the new workout exercises. |
| Alternative Flows | Same Level and Category  1. Member indicates on the workout screen that they want a new workout generated  2. A new workout is generated and displayed  Different Level  1. Member chooses new workout level on the workout screen  2. Member indicates that they want a new workout generated using the new level  3. A new workout is generated and displayed  Different Category  1. Member chooses new workout category on the workout screen  2. Member indicates that they want a new workout generated using the new category  3. A new workout is generated and displayed  Different Level and Category  1. Member chooses a new workout level on the workout screen  2. Member chooses a new workout category on the workout screen  3. Member indicates that they want a new workout generated using the new level and category  4. A new workout is generated and displayed |
| Exceptions | N/A |
| Priority | Medium |
| Other Info | 1. Because the workout is not given an id within the database until after the member has completed it, discarded workouts will not be recorded. |

**Create Workout Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR-9 | The System will prompt the Guest for their desired workout level. | M |
| FR-10 | The System will prompt the Guest for their desired workout category. | M |
| FR-11 | The System will identify exercises that match the desired workout level and category. Based on the desired workout and category, a specific number of the identified exercises are selected and displayed. The relevant level and category are also displayed. | M |
| FR-12 | If the Guest indicates they want to cancel the workout, the workout information will not be saved and the application home page will be displayed. | M |
| FR-13 | If the Guest indicates the workout is complete, the System will ask the Guest if they want to register an account. If the Guest declines, the workout information is not saved and the application home page will be displayed. | S |
| FR-14 | If the Guest indicates they want to register for a Member Account, the workout information will be saved and added to the new account as a prior workout once their registration is complete. | S |
| FR-15 | While signed in, the Member will request a workout. | S |
| FR-16 | The System will prompt the Member for their desired workout category. | S |
| FR-17 | The System will identify exercises that match the Member’s account workout level and the category they selected for the new workout. Based on the level and category, a specific number of the identified exercises will be selected and displayed. The relevant level and category will also be displayed. | S |
| FR-18 | If the Member indicates they want to cancel the workout, the workout information will not be saved and the member’s account page will be displayed. | S |
| FR-19 | If the Member indicates the workout is complete, the saved workout information is moved to the Member’s workout log. The Member’s account page is displayed. | S |
| FR-20 | The System will allow the User to choose an exercise from the workout list. | M |
| FR-21 | The System will indicate that the exercise can be removed. | M |
| FR-22 | The System will indicate that the exercise can be moved to a different position in the list. | M |
| FR-23 | If the User chooses to remove an exercise, the remaining exercises will shift positions to fill the empty spot. | M |
| FR-24 | If the User attempts to insert the exercise into a position already filled, the positioned exercise will shift down one position allowing the user to complete the insert. | M |
| FR-25 | The System will display the prescribed number of sets and reps for the exercise based on their chosen workout level. The System will indicate that the prescribed numbers can be changed. | M |
| FR-26 | The System will provide an area for the User to record the actual number of sets and reps that they completed for the exercise. | M |
| FR-27 | The System will indicate that the current workout can be replaced. | S |
| FR-28 | The System will use the displayed workout level and category to identify matching exercises and the number of exercises to select. If the User has indicated that they wish to change the level or category, the new list of exercises will match those. | S |
| FR-29 | The System will save the User’s workout as they perform it. | S |
| FR-30 | If the application or device shuts down while a workout was in use, the workout will be displayed at its last saved point when the next time the User opens the application. | S |
| FR-31 | Close captions will be available for the exercise demonstrations, and they will not block the User’s view of the demonstration. | C |
| FR-32 | A transcript of the demonstration and any additional notes needed to provide a full understand of the safe way(s) to perform the exercise will be available. | C |

**Feature: Workout Log User Stories**

|  |  |
| --- | --- |
| US-1 | As a member, I want to view my past workout to see my progression. |
| US-2 | As a member, I want to be able to edit my past workout to suit my current needs. |
| US-3 | As a member, I want to be able to sort through my workout log. |
| US-4 | As a member, I want to be able to remove prior workouts. |
| US-5 | As a member, I want to be able to use a past workout plan for my current workout so that I can do a workout I know I will enjoy. |
| US-6 | As a member, I want to be able to filter my prior workout logs to easily access past workouts. |

**Workout Log Use Cases**

|  |  |
| --- | --- |
| **UC-1** | **View Prior Workout/s** |
| Primary Actor | Member |
| Trigger | Member indicates, from their account, that they want to view their prior workouts |
| Precondition | 1. Member is logged into their account  2. Member on member account home page  Assumption: Member has completed at least one workout |
| Postconditions | N/A |
| Normal Flow | 1. Member indicates that they want to view their prior workouts  2. Prior workouts are displayed, ordered by date time descending  3. Member indicates which workout they are interested in  4. Member indicates they wish to view the workout  5. The workout's data and the list of completed exercises is displayed |
| Alternative Flows | Sort (Priority Low)  3. Member indicates they want to change the order of the workouts displayed  4. Member indicates whether they want workouts to be date ascending, sorted by category, or sorted by level.  5. The workouts in the log are re-positioned for this view  Filter(Prior Low)  3. Member indicates they want to change which prior workouts are displayed  4. Member indicates either a date range, category, or level to use for the filtering  5. Workouts that don't match the desired criteria are removed from the current view |
| Exceptions | N/A |
| Priority | Medium |
| Other Info | 1. Workout and exercise information that are being viewed will be in a read-only status  2. The prior workout data that will be reflected includes: the date and time the workout was completed, the workout level, and workout category.  3. If the workouts are sorted by level or category, the second sort level will be alphabetical ascending  4. The workouts remaining after a filter will be automatically sorted by date time descending  5. Any change to sorting will only be active during the current viewing session  6. Any filtering requested will only be active during the current viewing session  7. If a member attempts to filter or sort when no prior workout are present, no action will occur but no error message will be displayed. |

|  |  |
| --- | --- |
| **UC-2** | **Edit Prior Workout** |
| Primary Actor | Member |
| Trigger | Member indicates they want to change a prior workout within their log |
| Precondition | 1. Member has opened their account prior workout log  Assumption: Member has completed at least one workout |
| Postconditions | 1. Workout information in database is updated to reflect changes. |
| Normal Flow | 1. Member indicates the prior workout they are interested in  2. Member indicates they want to edit the workout  3. The workout data and list of exercises is displayed with the exercise target and actual sets and reps numbers in an editable state  4. Member increases or decreases the target or actual sets and reps numbers  5. Member indicates they want to save changes. |
| Alternative Flows | 2. Member indicates they want to view the workout  3. The workout's data and the list of completed exercises is displayed  4. Member indicates they want to edit the workout they are currently viewing  5. The screen is refreshed, changing the exercise target and actual sets and reps numbers into an editable state  6. Member increases or decreases the target or actual sets and reps numbers  7. Member indicates they want to save changes. |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. Member will be unable to decrease the number of actual sets and actual reps to below 0.  2. The prior workout data that will be reflected includes: the date and time the workout was completed, the workout level, and workout category. |

|  |  |
| --- | --- |
| **UC-3** | **Remove Prior Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to remove a prior workout within their log |
| Precondition | 1. Member has opened their account prior workout log  Assumption: Member has completed at least one workout |
| Postconditions | 1. The database is updated to reflect the state of the removed workout as hidden |
| Normal Flow | 1. Member indicates the prior workout they are interested in  2. Member indicates that they want to delete the workout  3. The log is refreshed to display the prior workouts without the one removed |
| Alternative Flows | 2. Member indicates they want to view the workout  3. The workout's data and the list of completed exercises is displayed  4. Member indicates they want to remove the workout they are currently viewing  5. The prior workout log is displayed again without the removed workout |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. A member will have the option to remove a prior workout from their own log, but the workout itself will not be removed from the database. |

|  |  |
| --- | --- |
| **UC-4** | **Use Prior Workout for Current Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to use a prior workout for a current workout |
| Precondition | 1. Member has opened their account prior workout log  Assumption: Member has completed at least one workout |
| Postconditions | N/A |
| Normal Flow | 1. Member indicates the prior workout they are interested in  2. Member indicates that they want to use the prior workout to start a new workout  3. The workout screen displays a list of exercises that match those in the prior workout, including the target sets and reps numbers |
| Alternative Flows | 2. Member indicates they want to view the workout  3. The workout's data and the list of completed exercises is displayed  4. The member indicates they want to use the workout they are currently viewing to start a new workout  5. The workout screen displays a list of exercises that match those in the prior workout, including the target sets and reps numbers |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. The copy of the prior workout being used for the current workout will have the actual sets and reps numbers for all the exercises changed back to 0. |

**Workout Log Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR-33 | The System will display the Date, Time, Workout Category, Workout Level of each prior workout. The most recently finished will be at the first. | C |
| FR-34 | If the user indicates that they want to view a specific workout, the System will expand the section so that the name, target sets & reps, and the actual sets & reps for each exercise is displayed. | C |
| FR-35 | The system will indicate to the user that they’re done with the opened workout. | C |
| FR-36 | When the user indicates they are done with the opened workout, the System will collapse the workout back to its original size. | C |
| FR-37 | The system will indicate that workouts can be selected. | C |
| FR-38 | The system will select the workout, once indicated by user. | C |
| FR-39 | The system will indicate that exercises can be selected to be edited. | C |
| FR-40 | The system will edit selected exercises, once indicated by user. | C |
| FR-41 | The system will indicate that the log can be sorted by ascending and descending date, category, and level. | C |
| FR-42 | The system will then sort by ascending and descending date, category, or level once indicated by the user. | C |
| FR-43 | The system will indicate that the log can be filtered by various metrics. | C |
| FR-44 | The system will then filter as indicated by the user. | C |
| FR-45 | The system will indicate that the prior workouts can be selected by name or date. | C |
| FR-46 | The system will select the workout by name or date, once indicated by the user. | C |
| FR-47 | The system will indicate the user, that the selected workout will now be used. | C |
| FR-48 | The system will then use the selected workout and replace the current one, once indicated by the user. | C |

**Feature: Templates**

|  |  |
| --- | --- |
| US-1 | As a member, I want to save a past workout as a template so I can easily find and use it for future workouts. |
| US-2 | As a member, I want to be able to take a template and edit it so that I can progress in difficulty as I continue to do the same workout every week. |
| US-3 | As a member, I would like to be able to view all my templates and delete the ones that I no longer want or need. |
| US-4 | As a member, I’d like to be able to sort my templates by category, so that I can easily find what I need. |

**Templates Use Cases**

|  |  |
| --- | --- |
| **UC-1** | **View Template/s** |
| Primary Actor | Member |
| Trigger | Member indicates they want to view one of their saved templates |
| Precondition | 1. Member signed into their account  2. Member on their account homepage |
| Postconditions |  |
| Normal Flow | 1. Member indicates they want to view their workout templates  2. Templates are displayed, ordered by name alphabetical ascending  3. Member indicates which template they are interested in  4. Member indicates they want to view the template  5. The workout template data and the list of exercises |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. When a member requested to view a template, the template will be displayed in read-only status  2. The template data that will be reflected includes: the name of the template, the date and time the template was created, the workout level, and workout category. |

|  |  |
| --- | --- |
| **UC-2** | **Create a Template** |
| Primary Actor | Member |
| Trigger | Member indicates they want to create a new template |
| Precondition | 1. Member has opened the templates feature in their account |
| Postconditions | 1. Database is updated to include new template |
| Normal Flow | 1. Member indicates they want to create a new template  2. The member's prior workouts are displayed  3. Member indicates which prior workout they want to use  4. Member enters what name the workout should be saved as in templates |
| Alternative Flows | N/A |
| Exceptions | E1 Name already in use  1. Application informs member that name given for saving is already in use  2. Application gives member the option to replace existing template with new template  3. If member declines replacement option, member can enter new name or cancel template creation |
| Priority | Low |
| Other Info | 1. The copy of the prior workout being used for the template will have the actual sets and reps numbers for all the exercises changed back to 0. |

|  |  |
| --- | --- |
| **UC-3** | **Edit a Template** |
| Primary Actor | Member |
| Trigger | Member indicates they want to change a template |
| Precondition | 1. Member has opened their account template feature |
| Postconditions | 1. Template information in database is updated to reflect changes. |
| Normal Flow | 1. Member indicates the template they are interested in  2. Member indicates they want to edit the template  3. The template data and list of exercises is displayed with the exercise target sets and reps in an editable state  4. Member increases or decreases the target sets and reps numbers  5. Member can save the changes to the current name or choose to save as a new name for a new template. |
| Alternative Flows | 2. Member indicates they want to view the template  3. The template data and list of exercises is displayed in a read-only state  4. Member indicates they want to edit the template they are currently viewing  5. The screen is refreshed, changing the exercise target sets and reps numbers into an editable state  6. Member increases or decreases the target sets and reps numbers  7. Member can save the changes to the current name or choose to save as a new name for a new template. |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. Member will be unable to decrease the number of actual sets and actual reps to below 0.  2. The template data that will be reflected includes: the name of the template, the date and time the template was created, the workout level, and workout category. |

|  |  |
| --- | --- |
| **UC-4** | **Remove Template** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to remove a template |
| Precondition | 1. Member has opened their account template feature |
| Postconditions | 1. Template is removed from database |
| Normal Flow | 1. Member indicates the template they are interested in  2. Member indicates that they want to remove the template  3. The screen is refreshed to display the templates without the one removed |
| Alternative Flows | 2. Member indicates they want to view the template  3. The template's data and the list of completed exercises is displayed  4. Member indicates they want to remove the template they are currently viewing  5. All the templates are displayed again without the removed workout |
| Exceptions | N/A |
| Priority | Low |
| Other Info | N/A |

|  |  |
| --- | --- |
| **UC-5** | **Use Template for Current Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to use a template for a current workout |
| Precondition | 1. Member has opened their account template feature |
| Postconditions | N/A |
| Normal Flow | 1. Member indicates the template they are interested in  2. Member indicates that they want to use the template to start a new workout.  3. The workout screen displays a list of exercises that match those in the template, including the target sets and reps numbers |
| Alternative Flows | 2. Member indicates they want to view the template  3. The template's data and the list of completed exercises is displayed  4. The member indicates they want to use the template they are currently viewing to start a new workout  5. The workout screen displays a list of exercises that match those in the template, including the target sets and reps numbers |
| Exceptions | N/A |
| Priority | Low |
| Other Info | N/A |

**Templates Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR-49 | System will allow the Member to choose a workout from their previous workouts list | C |
| FR-50 | System will give the Member the option to save a copy of the selected workout as a template | C |
| FR-51 | System will allow the Member to name the workout template. If the name given is already taken in their list, the system will notify the member and give them the option to replace the current template, choose a different name, or cancel the save. | C |
| FR-52 | System will allow the Member to save the workout template to their profile. It will then return to the templates screen and display the new template, confirming that the workout was saved. | C |
| FR-53 | System will allow the Member to view and scroll through a list of saved templates. | C |
| FR-54 | System will supply a drag and drop system for the user to delete templates. | C |
| FR-55 | If the member has the template open, the System will provide an option to delete it. | C |
| FR-56 | System will allow the member to view a list of saved templates. | C |
| FR-57 | If the Member indicates they want to edit a template, the System will open and display the template in an editable state. | C |
| FR-58 | If the Member has opened a template to view, the System will provide an option to edit. | C |
| FR-59 | Member can change the name of the template. | C |
| FR-60 | Member can change the order of the exercises within the template | C |
| FR-61 | Member can change the number of target sets of any exercise within the template workout. | C |
| FR-62 | Member can change the number of target reps of any exercise within the template workout. | C |
| FR-63 | System will allow edited template to be saved under the current name or new name for a new template. | C |
| FR-64 | System will provide a read-only view option for all templates | C |
| FR-65 | System gives Member the option to select a template from the list of templates to immediately start a workout. | C |
| FR-66 | If a template is open for preview, the system will allow the user to start the selected template as a workout. | C |

**Feature: Future Workout Queue**

|  |  |
| --- | --- |
| US-1 | As a member, I want to create and save workout to a queue for future workouts so I can set my workout goals ahead of time. |
| US-2 | As a member, I want to remove a workout from a queue for future workouts in case I change my mind. |
| US-3 | As a member, I want to be able to modify a workout in a queue for future workouts so that I can make desired changes to workout goals. |
| US-4 | As a member, I want to be able to review the workouts I saved to the future workout queue in case I forget the workout details. |
| US-5 | As a member, I want to be able to use a workout I previously prepared so that I can be sure that I’m progressing in the exercises as planned. |
| US-6 | As a member, I want to be able to open and review future workouts so that I can view the exercises planned. |
| US-7 | As a member, I want to be able to view future workouts so that I can see each workout in the desired order. |

**Queue Use Cases**

|  |  |
| --- | --- |
| **UC-1** | **View Queue Workout/s** |
| Primary Actor | Member |
| Trigger | Member indicates they want to view one of their saved workouts in their future workout queue |
| Precondition | 1. Member signed into their account  2. Member on their account homepage |
| Postconditions | N/A |
| Normal Flow | 1. Member indicates they want to view their workout queue  2. Queue workouts are displayed  3. Member indicates which queue workout they are interested in  4. Member indicates they want to view the workout  5. The queue workout data and the list of exercises is displayed |
| Alternative Flows | N/A |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. When a member requested to view a queue workout, the workout will be displayed in read-only status  2. The queue workout data that will be reflected includes: the name of the workout, the date and time it was added to the queue, the workout level, and workout category.  3. The home screen for the queue feature will display workouts saved by name and display workouts saved by date, but in separate tables. |

|  |  |
| --- | --- |
| **UC-2** | **Create a Future Workout** |
| Primary Actor | Member |
| Trigger | Member indicates they want to create a new future workout for the queue |
| Precondition | 1. Member has opened the queue feature in their account |
| Postconditions | 1. Database is updated to include new queue workout |
| Normal Flow | 1. Member indicates they want to create a new future workout for the queue  2. Application request location of workout they want to use: Workout Log or Templates  3. Member indicates the location  4. The workouts from that location are listed  5. Member indicates which workout they want to use  6. Member enters the name or date they want associated with the workout |
| Alternative Flows | N/A |
| Exceptions | E1 Name already in use  1. Application informs member that name given for saving is already in use  2. Application gives member the option to replace existing template with new template  3. If member declines replacement option, member can enter new name or cancel template creation |
| Priority | Low |
| Other Info | 1. The copy of the prior workout being used for the template will have the actual sets and reps numbers for all the exercises changed back to 0. |

|  |  |
| --- | --- |
| **UC-2** | **Remove Future Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to remove a workout from the queue |
| Precondition | 1. Member has opened their account queue feature |
| Postconditions | 1. Workout is removed from queue in the database |
| Normal Flow | 1. Member indicates the future workout they are interested in  2. Member indicates that they want to remove the workout  3. The screen is refreshed to display the queue workouts without the one removed |
| Alternative Flows | 2. Member indicates they want to view the queue workout  3. The workout's data and the list of completed exercises is displayed  4. Member indicates they want to remove the queue workout they are currently viewing  5. The queue home page is displayed with the workout removed |
| Exceptions | N/A |
| Priority | Low |
| Other Info | N/A |

|  |  |
| --- | --- |
| **UC-3** | **Edit a Future Workout** |
| Primary Actor | Member |
| Trigger | Member indicates they want to change a queue workout |
| Precondition | 1. Member has opened their account queue feature |
| Postconditions | 1. Queue workout information in database is updated to reflect changes. |
| Normal Flow | 1. Member indicates the queue workout they are interested in  2. Member indicates they want to edit the workout  3. The queue data and list of exercises is displayed with the exercise target sets and reps in an editable state  4. Member increases or decreases the target sets and reps numbers  5. Member can save the changes to the current name or choose to save as a new name/date for a new queue workout. |
| Alternative Flows | 2. Member indicates they want to view the workout  3. The queue workout data and list of exercises is displayed in a read-only state  4. Member indicates they want to edit the workout they are currently viewing  5. The screen is refreshed, changing the exercise target sets and reps numbers into an editable state  6. Member increases or decreases the target sets and reps numbers  7. Member can save the changes to the current name or choose to save as a new name/date for a new queue workout. |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. Member will be unable to decrease the number of actual sets and actual reps to below 0.  2. The queue workout data that will be reflected includes: the name/date associated with the workout, the date and time the queue workout was created, the workout level, and workout category. |

|  |  |
| --- | --- |
| **UC-5** | **Use Workout in Queue for Current Workout** |
| Primary Actor | Member |
| Trigger | Member indicates that they want to use a queue workout for a current workout |
| Precondition | 1. Member has opened their account queue feature |
| Postconditions | N/A |
| Normal Flow | 1. Member indicates the queue workout they are interested in  2. Member indicates that they want to use the queue workout  3. The workout screen displays the workout |
| Alternative Flows | 2. Member indicates they want to view the queue workout  3. The queue workout data and the list of completed exercises is displayed  4. The member indicates they want to use the workout they are currently viewing  5. The workout screen displays the workout |
| Exceptions | N/A |
| Priority | Low |
| Other Info | 1. Once the member has completed the workout, the workout will be removed from the queue. |

**Queue Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR-66 | If the workout is from the Workout Log, the System will clear the actual sets fields when saved to the queue | C |
| FR-67 | If the Member indicates to cancel the creation of a new future workout, the future workout will not be saved and the workout queue page will be displayed | C |
| FR-68 | If the Member indicates they wish to save a name or date for a future workout, the System saves that name or date | C |
| FR-69 | If the Member indicates they wish to create a new future workout, the system prompts the user to choose between the workout log or a template | C |
| FR-70 | When the use Member selects a workout in the log or templates, a copy of the workout is created for the queue | C |
| FR-71 | If the Member indicates the deletion of a future workout, the system deletes the future workout | C |
| FR-72 | If the Member indicates to cancel the removal of a future workout, the system will revert the future workout to the previous condition | C |
| FR-73 | If the Member indicates they wish to remove a future workout, the system prompts the Member to choose which workout to delete. | C |
| FR-74 | If the Member indicates they wish to edit a future workout, the system prompts the Member to choose which workout to modify. | C |
| FR-75 | If the Member indicates they wish to save their changes for a future workout the system saves that name or date. | C |
| FR-76 | If the Member indicates they wish to view a future workout, the system displays the user their queue of future workouts to view | C |
| FR-77 | If the Member indicates they wish to back out of viewing a workout, the system returns the user to the select portion | C |
| FR-78 | If the Member indicates they wish to use a future workout, the System prompts the Member to choose which future workout they desire to use | C |
| FR-79 | If the Member indicates they wish to start the future workout, the System starts the workout | C |
| FR-80 | If the Member indicates they wish to open and review a saved future workout, the System prompts the Member to choose which future workout they desire to open and review | C |
| FR-81 | If the Member indicates they wish to back out of viewing a workout, the System returns the Member to the selected portion | C |
| FR-82 | If the Member is working in an existing file and chooses to save the file, the System shall save it with the same name. | C |
| FR-83 | If the System is unable to save a file using a specific name, the System shall give the Member the option to save it with a different name or cancel the save operation | C |

**3.2 Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Availability |  |  |
| NF-1 | Updates will initially take place between 3am and 4am, East Cost US time-allocated |  |
| NF-2 | Users should be able to access their account information as long as the server is running. |  |
| NF-3 | Greats efforts should be made to ensure that the software is available with an uptime of at least 99%. |  |
| NF-4 | You should expect fourteen minutes and twenty-four seconds of downtime every day. |  |
| NF-5 | Expect an hour, forty minutes, and 24 seconds of downtime every week. |  |
| NF-6 | You should expect six hours, forty-three minutes, and 12 seconds of downtime every month. |  |
| NF-7 | Usage will be tracked to identify when traffic is at its lowest and the update schedule will be adjusted accordingly. |  |
|  |  |  |
| Installability |  |  |
| NF-8 | The installation will take no more than three minutes, two minutes ideally. |  |
| NF-9 | Installation is done locally only by the owner of the relevant device. |  |
| NF-10 | The application installer should only be accessible through the following approved locations: Apple store, Google play store, Windows store. |  |
| NF-11 | Requirements for a successful user installation will be provided in the application information. displayed in the app stores. |  |
| NF-12 | The installer will request use of mobile devices ‘s keyboard. |  |
| NF-13 | The installer will request use of mobile device's audio. |  |
| NF-14 | Following successful installation, the installation program shall delete all temporary, backup, obsolete, and unneeded files associated with the application |  |
|  |  |  |
| Data Integrity |  |  |
| NF-15 | All Dates will be formatted DD/MM/YYYY. |  |
| NF-16 | Time will be formatted HH:MM using military time |  |
| NF-17 | The time reflected will be specified by the device that is being used for the application |  |
| NF-18 | During registration and in the member’s profile, the format of the entered email address will be checked for accuracy |  |
| NF-19 | The email address field will be 40 characters long |  |
| NF-20 | During registration and in the member’s profile, the format of the phone number entered will be checked for accuracy |  |
| NF-21 | Dashes are allowed in the phone number field |  |
| NF-22 | The phone number field will be 20 characters long |  |
| NF-23 | If a failure is encountered partway through changes to data, the system will reset back to the last saved point and notify the user of the issue |  |
| NF-24 | Hardware security will be the responsibility of the cloud provider |  |
| NF-25 | Backups of the database and media will be saved on a separate cloud server automatically once a week |  |
|  |  |  |
| Performance |  |  |
| NF-26 | The response time for creation of a workout will be no more than two seconds |  |
| NF-27 | The video exercise demonstrations will be ready to play within four seconds. The video play time will be kept under 30 seconds. Currently, the largest device that will be playing the videos will be a tablet |  |
| NF-28 | Response time for all other interactions will be under one second |  |
| NF-29 | As the member information does not require much computation or heavy resource consumption the performance of retrieving and updating information on the database should be very quick (under 5 secs on user end) and confirmation emails should take no later than 3 mins during high traffic times. |  |
| NF-30 | The system must be able to provide the workouts to the client, they should be able to start, change, view, and access a workout. |  |
| NF-31 | Clients should be able to modify existing workouts. Change their names and arrange it via name, time, date. |  |
|  |  |  |
| Reliability |  |  |
| NF-32 | In the member registration, the birth year options will be in a drop-down to prevent user error. |  |
| NF-33 | In the member registration, the gender identification options will be in a drop-down to prevent invalid input. |  |
| NF-34 | In the member registration, zip code field will only accept integer values. |  |
| NF-35 | The target and actual sets and reps fields will only accept integer or floating-point numbers. No other special characters, besides decimal, are allowed. |  |
| NF-36 | When saving a template or future workout, the system will only accept letters, numbers, underscores and hyphens. |  |
| NF-37 | All error messages will explicitly state what the user entered and why it was considered invalid. |  |
| NF-38 | A message will always be displayed after the user saves edits to a workout in the log, template, or queue to confirm the data was successfully saved. |  |
|  |  |  |
| Robustness |  |  |
| NF-39 | If the application is unable to reach the database, the user will be informed, and the application will continue to try to reach the database every ten minutes that it is open. If the user’s settings allow the app to work in the background, it will continue to try even while closed. If the user’s settings do not allow this, then the application will attempt a connection the next time it is opened. Once a connection has been made, the application will inform the user. |  |
| NF-40 | A prior workout must be saved as a template or queue workout before it can be edited or used within the template or queue feature. |  |
| NF-41 | When a user has a workout open for use, the application will autosave progress every five minutes. |  |
| NF-42 | Until a user has entered an integer or floating-point number into the actual sets and reps fields of an exercise, the default value will be zero. |  |
| NF-43 | If the database is down the user should still be able to access the front end if it is operational as a nonmember. If a wrong password is used at login the operation should fail with a warning to retry the correct password. |  |
| NF-44 | The client should not terminate in case of invalid API-requests; the listener shall reject those. |  |
| NF-45 | The client shall be resistant against unexpected flood of requests. |  |
| NF-46 | The client shall never spawn that many application processes that the server hits maximum process limit or runs out of memory. Or the dedicated database sessions created by the application processes cause the databases to extend the maximum session limit. |  |
| NF-47 | The database shall be protected from reaching the session limit caused by unexpected number of database connections. |  |
|  |  |  |
| Safety |  |  |
| NF-48 | A guest user will be required to agree to a liability waiver before being allowed to use a workout. It should specify that since the user is a guest, the liability waiver will only be applicable for the current workout unless the guest registers before closing the application. |  |
| NF-49 | A member will be required to agree to a liability waiver when they register if it has not already been done as a guest. |  |
| NF-50 | The only information that will be required when a member registers for an account is an email address, phone number, and password. All demographic information will be optional and, it will be noted, for our own statistical purposes only. User information will not be sold. |  |
| NF-51 | Passwords will only be accepted if they are at least nine characters in length. |  |
| NF-52 | Passwords will only be accepted if they include lower case letters, upper case letters, numbers, and select specified special characters. |  |
| NF-53 | A password will be accepted if it is copied and pasted into the field. |  |
| NF-54 | A member’s email address and phone number will be verified by the user entering six-digit codes into the application that were sent to the email address and mobile phone number on file. |  |
| NF-55 | Members will never be required to change their password. |  |
| NF-56 | Members will not have to complete any security questions. |  |
| NF-57 | If a member forgets their password, they will have the option to have a six-digit code sent to the email address or phone number on file. Once the code has been entered, the user will be allowed to create a new password. |  |
| NF-58 | Help Desk Support will have the authorization to view a member’s profile, workout log, workout templates, workout queue, and the individual workouts within them in a read-only state. |  |
| NF-59 | Member passwords will be hashed. |  |
| NF-60 | Help Desk Support will be able to view the dates, times, destinations, and use of any six-digit codes sent, but not the code itself. |  |
| NF-61 | Help Desk Support will have an area within a member’s account to record notes and attach documents related to interactions with the member. |  |
| NF-62 | Administrators will be authorized to view, add, update, and delete application exercises. |  |
| NF-63 | The account should not stay logged in unless it is in the mobile app or the user specifies on desktop. The password should appear as only asterisks of the correct number of characters on login or viewing on the account screen. |  |
| NF-64 | Client should be notified of the space that each exercise takes. |  |
| NF-65 | Client should be warned to take breaks after every few reps/sets. |  |
| NF-66 | Client should examine equipment before use. |  |
| NF-67 | Client should put equipment back where it belongs. |  |
| NF-68 | Client should do stretches/warmups first before workouts. |  |
|  |  |  |
| Security |  |  |
| NF-69 | Security should be of highest concern all information should be hashed on the server to guarantee safety if data breach occurs. |  |
| NF-70 | The software system defined in this SRS must follow industry recommended practices for secure software development. |  |
| NF-71 | At a minimum, the software development must practice least privilege for defining for access-level requirements of the software system and its associated services. |  |
| NF-72 | The production-release version of the software system must pass an automated dynamic application security testing tool. |  |
|  |  |  |
| Usability |  |  |
| NF-73 | It will take five or fewer required interactions to complete an application operation. |  |
| NF-74 | It will take no longer than 20 seconds to complete required interactions for an application operation. |  |
| NF-75 | Buttons, menus, and drop-downs will always be used in place of user entry when feasible to cut down on user typing. |  |
| NF-76 | Buttons, menus, and drop-downs will always be used in place of user entry when feasible to cut down on user typing. |  |
| NF-77 | Tooltips will be distributed throughout the application to provide instructional or informative information. |  |
| NF-78 | Exercise video demonstrations will not be set to auto-play. |  |
| NF-79 | The exercise video demonstration audio will initially be muted. The user will be prompted to turn up the volume if they desire. |  |
| NF-80 | Close captions will be provided on the videos, and they will not block the exercise demonstration view.  If the video is unable to be displayed, a detailed description of the exercise and any relevant safety information will be displayed. |  |
| NF-81 | The member account section will work with anyone who can read and has a working phone number or email. |  |
| NF-82 | Have a phone or device that can emulate or is able to access the app. The app is very simple. |  |
|  |  |  |
| Portability |  |  |
| NF-83 | The application will be built to be cross-platform. |  |
| NF-84 | It will initially be available on mobile smartphones and tablets. It will then extend to laptops and PCs. |  |
| NF-85 | The application will be built and run in the cloud. |  |
|  |  |  |
| Constraints |  |  |
| NF-86 | The display will be designed to fit on the currently smallest smartphone. |  |
| NF-87 | Because touchscreens will be used, areas of user interaction will be large enough for a finger to activate without affecting neighboring sections. |  |
| NF-88 | The project cannot have too many workouts/ workout videos to start due to lack of manpower and time on the project. Users could submit their own workouts (i.e. burpees, pushups, etc.) if it is not already loaded in to widen the database available to all users while videos would just have to slowly be rolled out along various updates. |  |
| NF-89 | Project would most likely need to be implemented in a “one size fits all” framework instead of having a nicely made custom application for mobile and desktop use. Most likely a framework like electron would be used. While this is useful in terms of time it can mean that each application would not be a nice looking and could share bugs. |  |
| NF-90 | Users may store their data using QDR (Query-able Data Repository), which will use a data store to curate data and template information. All QDR and administrative services must be available over RESTful API’s. |  |
|  |  |  |
| Assumptions and Dependencies |  |  |
| NF-91 | Service Email (for updates and verification confirmations) |  |
| NF-92 | Service Phone Number (for notifications and confirmations) |  |
| NF-93 | Assumptions: Application is user-friendly and provides useful resources for experienced as well as inexperienced users. |  |
| NF-94 | Dependencies: Use of app by users, operating systems, customer support |  |

**4. High-Level Design**

4.1 Security

1. Application

Within the application, certain privileges will be secured behind the need for a username and password. Without an account, the user will only be permitted to use the basic, necessary actions that are accessible client-side. These actions include: 1) The ability to pick a preloaded workout, and 2) Navigate the app without connection to the server. Upon creating and verifying an account, the user will receive access to the servers which allow access to the account and to the other features of the application. This measure of security lowers the likelihood of unnecessary access to the servers.

1. Accounts

Accounts will be created using a password consisting of at least 8 characters, including one letter, one number, and one special character. These standards will increase the security of user accounts.

1. Data

Data will be protected using a hashing algorithm when passing information from the client to the server. This will help protect against hacking attempts in relation to the server. Application users will be notified immediately if a breach of data is detected so that further measures can be taken by users to protect their information/data.

4.2 Hardware/Software

The application will primarily be usable on handheld devices, such as smartphones and tablets. The application should also have have functionality on both desktop and laptop computers, though not the main source of access.

For the server and database back-end processes, the application will use Amazon Web Services (AWS) and Amazon’s Relational Database (RDS). Both of these services will help to ensure the safety of user personal information as well as allow for management tasks such as migration, back-up, recovery, and patching.

4.3 Interfaces

The user will be able to view several interfaces while using the application. Each interface allows the user to perform certain actions and/or access certain features of the application. A few of the interfaces to be included are: 1) Registration, 2) Sign-in, and 3) Account. A table of user interface requirements is included in the appendix at the conclusion of this document.

4.4 Architecture

1. Client/Server

A multi-tier client/server architecture will be used for the application. This allows users to have access to the user interface through the client application, while also allowing the database to be held on servers with API server requests in the middle. This approach allows the client application to be accessed by the user while lowering a portion of the over-head on the server-side.

1. Client Responsibilities

The client shall provide a functional user interface which can be accessed without network connectivity. The client will be responsible for loading and running workout plans.

1. With connectivity:

* The client application will hold user credentials and will send them to the API to validate user information, allowing for function log-in.
* The application will send and receive API calls to update user account information, as well as allow the user to get, create, or update workout plans.

1. Without connectivity:

* The client will open in guest mode as the account cannot be verified, unless the account was logged in and verified prior to loss of connectivity.
* If in guest mode, the client will feature a limited list of exercise options to choose from.
* If log-in was verified before loss of connectivity, the client will have cached the user’s most used and most recent workouts as well as the default list of limited exercises referred to above for the user to select from.

1. Server Responsibilities

The server shall provide a database of workouts for all users as well as save workout plans and account details for users. The data will be called from the API and all generic user information will be sent without the need for user validations. User account information and custom workout plans will need user account validation before being accessible to the API.

The API will be responsible for connecting each user to the necessary data from the database. The API commands are as follows:

1. Create
   1. User accounts
   2. New exercise
   3. New workout plan
2. Read
   1. Exercise data (name, details, video)
   2. Workout plan queue
   3. User account information
   4. Workout routine details
3. Update
   1. User account information
   2. User workout plans
   3. Automated workout routine updates
4. Delete
   1. User account
   2. User workout plans

4.4 Database

A screenshot of a computer

Description automatically generated with medium confidence

4.5 Top-Level Classes

Text

Description automatically generated

4.6 Data Flow and States

Data will flow between users and the client/server to allow users to create accounts, workout plans, and use features associated with the application. Diagrams of data flow are outlined in the appendix section of this document.

4.7 Reports

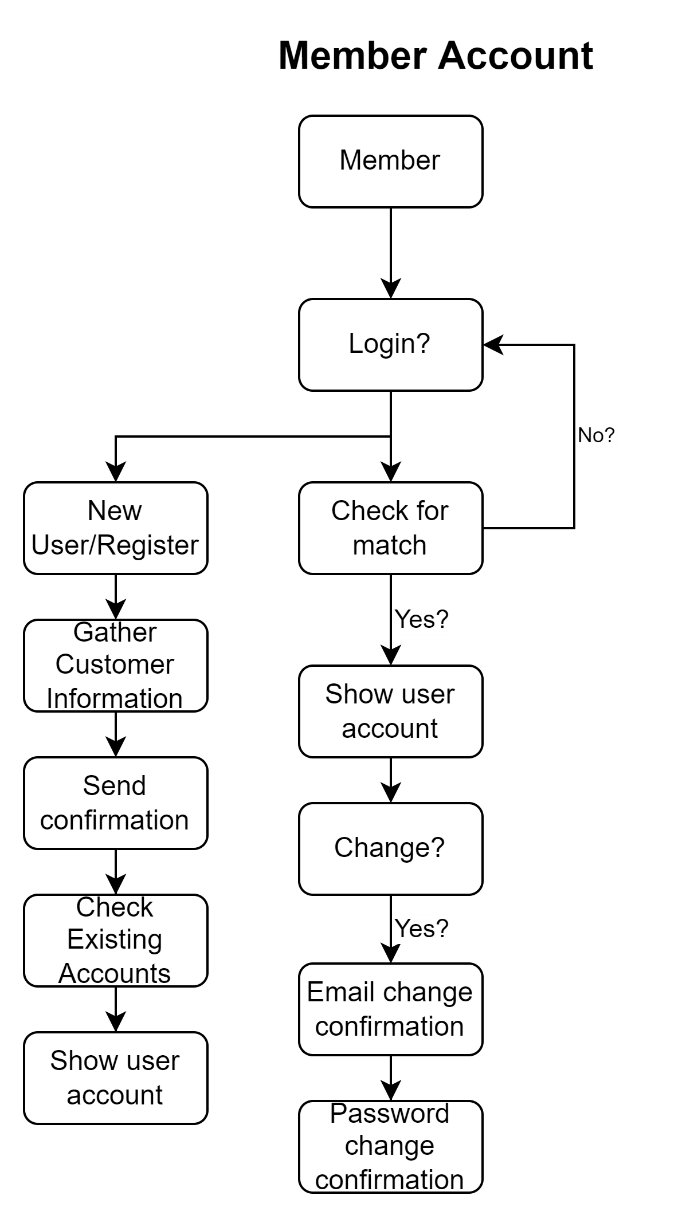
The business requirements of the project plan reflect our desire to monitor member demographics, chosen workout levels, chosen workout categories, and individual exercise popularity. The reports generated from this data will be used to identify areas in which we can improve the application. Further, users will be able to create individual reports which can show progress over time for specific workouts or as a whole.

**5. Appendix**

5.1 Interface Requirements

|  |  |  |
| --- | --- | --- |
| User Interfaces |  |  |
| EI-1 | A Help link will be visible on each page to direct users to our contact form and FAQs. |  |
| EI-2 | Users will have the ability to increase the font size to make it easier to read. |  |
| EI-3 | Users will have the ability to change the display to one of high contrast. |  |
| EI-4 | The application will conform to the Web Content Accessibility Guidelines. |  |
| EI-5 | Register Interface:  After clicking register account the app will run the user through a series of “slide” pages with the first labeled Credentials asking Email, Phone, and password and asking the user if they want to verify their account with email or phone. The next slide will be Demographics asking for height, weight, age, etc. (listed above). The next slide will be to enter the confirmation code from the email or phone number the user selected. |  |
| EI-6 | Sign-In Interface:  The sign-in Interface can be selected from a banner at the top of the screen and will appear as a popup asking for the user to input either email or phone number and password. |  |
| EI-7 | Account Interface:  The account interface will have the users name up top with sections of “Account Information” containing email and phone number as well as password which the user can edit. The next section will be “Demographics” With the Demographic information in it. At the bottom of the page will be a delete account button where the user can remove their information from the servers. |  |
| EI-8 | Should be organized and obvious as to what each thing does. The user interface should properly show all the data for the member and or guest. The videos/photos showing the exercises should be displayed properly. |  |
|  |  |  |
| Software Interfaces |  |  |
| EI-9 | The application will communicate with the Database for the following operations:   * To allow a user to create, modify, and delete a member account. * To allow a member to sign in and bring up their account information. * To generate a workout for a guest or member * To save, view and edit prior workouts * To create, edit, view, and delete template workouts * To create, edit, view, and delete future workouts |  |
| EI-10 | The software should be able to connect to the database to store member and guest generated data objects. Source and destination formats for data must include XML and may also include Extensible Stylesheet Language Transformation (XSLT), JavaScript Object Notation (JSON), Comma Separated Value (CSV), and American Standard Code for Information Technology (ASCII). |  |
|  |  |  |
| Communications Interfaces |  |  |
| EI-11 | The application will send an email to a member to verify the email address, assist in changing their password and notify of suspicious login attempts. |  |
| EI-12 | The application will send a text message to a member to verify the mobile phone number and assist in changing their password. |  |
|  |  |  |
| Hardware Interfaces | Mobile: The user must have a smartphone which can connect to an app store to download the application. The device must also have a working phone number if the user is using it for verification. |  |
| EI-13 | All server-side components must execute on server-class computers. All client-side components must execute on phones, computers, and other devices. |  |

5.2 Data Flow Diagrams



Diagram

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Diagram

Description automatically generated

Diagram, Teams

Description automatically generated

5.3 Wireframes

Graphical user interface, diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Diagram

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Graphical user interface, diagram

Description automatically generated