| Use Case Name: **Workout Tracking** | ID: **1** | Importance: **High** |
| --- | --- | --- |

| Primary Actor: **User** | Use Case Type: **Functional** |
| --- | --- |

| Stakeholders: Admins might be interested in looking at the data/construction stats. Users will want to directly interact with the workout tracking, adding to it with new workouts, monitoring the workouts they’ve done, and maybe take advantage of the predictions. |
| --- |
| Brief Description: Workout tracking will allow user to complete a whole workout tracking everything they did. At the end it will compile a workout data and use it in certain ways to improve the users future workouts. |
| Trigger: User starts workout |
| Relationships: User -> System |
| Normal Flow of Events:   1. The user navigates to Start workout page 2. The system present a selection of preset exercise plans 3. The user completes each exercise of the workout plan 4. The user completes the entire workout by pressing an end button 5. The system stores workout data |
| Subflows: N/A |
| Alternate Flows:   1. The user starts a blank workout    1. The user clicks add new exercise    2. The user inputs the weight and reps they did for that exercise 2. The user edits a exercise in their preset while doing the workout    1. They did more reps    2. They did more weight |

| Use Case Name: **Built in workout templates** | ID: **2** | Importance: **High** |
| --- | --- | --- |

| Primary Actor: User | Use Case Type: Functional |
| --- | --- |

| Stakeholders: Users and admins. The templates provide a good starting point for the users, and the admins might be interested in which templates get the most attention/uses for future changes. |
| --- |
| Brief Description: The workout program templates serve as a way for users to download and use “official” templates without having to search on their own for one. |
| Trigger: 1. The user interacts with the “Find Template” button. 2. The system has a popup with the recommended starting template based on the user’s goals. |
| Relationships: User<->System |
| Normal Flow of Events:   1. The user navigates to the template page 2. The user chooses to view the apps built in templates via a selector 3. A range of default templates are presented to the user 4. The user selects a template 5. The user is given the options to open, modify, or start a workout with the selected template |
| Subflows: The system keeps track of which template was chosen and stores the data. |
| Alternate Flows: A pop-up with the recommended template once the user chooses their goals |

| Use Case Name: **User Profile Creation** | ID: **3** | Importance: **High** |
| --- | --- | --- |

| Primary Actor: User | Use Case Type: Functional |
| --- | --- |

| Stakeholders: Everyone, Users will have their data tied the account created and business side all data tracking will be tied to the account created |
| --- |
| Brief Description: Profile creation will create a unique identifier for a user to be able to have a personalized experience that is able to store data between app uses |
| Trigger: Login, Create Account, Save Workout when not logged in |
| Relationships:  Users<->Everyone |
| Normal Flow of Events:   1. The user is navigated to the profile creation page 2. The web app presents a form to enter profile information 3. The user fills in the form with the following details    1. First Name    2. Last Name    3. Username    4. Password    5. Email    6. Preferred workout method (optional) 4. The user submits the form 5. Web application validates the information 6. If the information is valid, store information and direct user to profile page |
| Subflows: Changing password/email/preferences |
| Alternate Flows:  Username already exists  Password is insecure  Bad email  User logs in |

| Use Case Name: **Workout Progress Page** | ID: **4** | Importance: **High** |
| --- | --- | --- |

| Primary Actor: **User** | Use Case Type: **Functional** |
| --- | --- |

| Stakeholders: Users will have their data collected from their workouts and on the functional side all data will be tied to the users account. |
| --- |
| Brief Description: The workout progress page gathers metrics from all previous workouts and calculates metrics like personal records, then displays them in a report. |
| Trigger:The button to access the workout progress page was clicked |
| Relationships: User<-> System |
| Normal Flow of Events:   1. Button to access progress page is clicked 2. The system makes the necessary calculations to calculate personal records, weight increase in the last month, in the last 3 months, and in the last year. The system also calculates the amount left to hit their goals. 3. The user is allowed to choose which exercises show up on the progress page. 4. The user can input goals for each exercise 5. The system shows the report. 6. The page is exited once the save button is clicked |
| Subflows: N/A |
| Alternate Flows:  Input goals  Choose exercises |

| Use Case Name: **After Action Report** | ID: **5** | Importance: **High** |
| --- | --- | --- |

| Primary Actor: System | Use Case Type: Functional |
| --- | --- |

| Stakeholders: Everyone, users will have their data collected from the workout that they just completed and on the functional side all data will be tied to the users account. |
| --- |
| Brief Description: The after action report generates a report of how long the workout lasted, exercises performed, weight for each exercise, and other metrics pertaining to the workout that was just completed. |
| Trigger: The end workout button was pressed. |
| Relationships:  User<->System |
| Normal Flow of Events:   1. The user clicks the end workout button. 2. The system performs the necessary calculations for time elapsed, what exercises were performed, sets and reps of those exercises, the highest weight performed for each exercise, and other metrics generated on the after action report. 3. The system then displays the report. 4. The user clicks the finish button. |
| Subflows: N/A |
| Alternate Flows: N/A |