GLUB TRAINING MANAGER

**TEST PLAN**

Cristin Rusnac | Fontys University of Applied Science

c

Contents

[Introduction 2](#_Toc75109616)

[Testing Types 3](#_Toc75109617)

[Unit Testing 3](#_Toc75109618)

[End to End Testing 3](#_Toc75109619)

[User Acceptance testing 4](#_Toc75109620)

[Test Strategy 5](#_Toc75109621)

[Test Case: User CRUD and Authentication tests 5](#_Toc75109622)

[Test Case: User can access the data of all the trainings 6](#_Toc75109623)

[Test Case: User can add exercise to the training 6](#_Toc75109624)

[Test Case: User can edit any exercise data 6](#_Toc75109625)

[Test Case: User can delete exercise 6](#_Toc75109626)

[Test Case: The exercise and training calories are calculated and displayed 7](#_Toc75109627)

[Test Case: User Logout 7](#_Toc75109628)

[User Acceptance Tests 8](#_Toc75109629)

[Requirements 8](#_Toc75109630)

[Plan 8](#_Toc75109631)

[Confirmation of Business Objectives 8](#_Toc75109632)

# Introduction

This document formally describes the test plan for the solution. This is a systematic approach to testing the system. Currently, this plan contains a detailed understanding of how the software will respond to different situations it encounters and all information regarding the testing phase. There will be implemented three testing types which are Unit Testing, End to End Testing and User Acceptance tests.

The Test Strategy is based on the User Stories and is categorized based on the functionality and features of the application. Also the User Acceptance Test Requirements, Plan are described as well as the Confirmation of the Business Objectives.

# Testing Types

## Unit Testing

Unit tests are going to be implemented based on the user story acceptance criteria and based on the main features that affects the usage of the app. Unit tests will ensure that the application functions accordingly and that all the functionality is intact.

**Tested Features:**

* Retrieving all users, finding users by ID, deleting users, deleting users by ID and user Authentication to the system.
* Training retrieving, retrieving exercise by ID from the training, deleting exercise, editing exercise, adding exercise and the logic that determines the total calories of the training.
* Exercise creation, deletion of the exercise, fetching all exercises, and fetching a specific exercise based on the ID.

## End to End Testing

End to end testing will perform all possible user interactions with the application and check if all the logic and procedure works as supposed. End to end testing will ensure that all the layers of the application work accordingly (database, backend, frontend).

**Tested Features:**

* User Authentication
* Navigation
* Data retrieve
* Adding exercise to training
* Editing exercise from the training
* Deleting exercise from the training
* Sign out

**Testing Plan:**

* Test accesses the Homepage
* Test checks if the appropriate element is displayed
* Test accesses the Login page through the navbar
* Test fills in the credentials in the form and clicks the appropriate button
* Test navigates to the Add Exercise page, fills in the data and submits the exercise
* Test navigates to the Training page and checks if the submitted exercise exists
* Test clicks the appropriate button to access the edit of the submitted exercise
* Test changes the data, updates the exercise and checks if the changes appear in the training
* Test sings out

## User Acceptance testing

The user acceptance testing is also an important part of the testing phase. A user that knows how the features of the app work and the purpose of each element can determine unexpected or unwanted behavior by using the application.

Tested Features:

* User will check if all the buttons work accordingly
* User will determine if the logic and the procedure of the app is correct
* User will check if the logic is correct based on the input
* User will check the validation of the input data

# Test Strategy

Test Cases will represent the procedure of testing specific functions of the application using mock data. The purpose of testing is to identify unusual or unexpected behavior. Which in this specific case is focused on the CRUD that let us know if the application behaves different or not as expected after changes are made or new features are added. Since the tests are made with mocked data the unwanted behavior will be identified easily.

## Test Case: User CRUD and Authentication tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 1** | Find All Users | There is a user table in the database, and it stores user data. | Users | The addition of two users to the database is mocked and then the test checks how many entities are stored and if it is possible to get them. |
| **TC-02** | Find User by ID | There is a user table in the database and the user data can be fetched based on their id. | Users | A user with a specific id is added to a mocked directory and the test fetches it by the users id. |
| **TC-03** | Save User | There is a repository that stores the user data and new users can be added. | Users | A repository is mocked and the test adds a user and checks if the user is present in the repo. |
| **TC-04** | Delete User by ID | There is a user stored in the database and a method to delete the data by ID | User | The user is removed from the repository based on the id passed. |
| **TC-05** | Authenticate user | The user data is stored in the database and can be fetched to authenticate in the system | Users | The user data that is filled in in the form is processed by the backend logic and a token is generated that is added to all the users CRUD operations. |

## Test Case: User can access the data of all the trainings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 2** | Get Training | There is a Training table in the database that stores data based of exercises based on the exercise ID. | Training | The user retrieves the data of the training containing all exercise details in an interactive Table. |

## Test Case: User can add exercise to the training

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 3** | Add Exercise to Training | There is an appropriate list of exercises that can be selected by the user and adding the duration parameter and position it is added to the Training. | Training | The user can fill in the exercise details in the form and submit it and they will be stored in the training table. |

## Test Case: User can edit any exercise data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 4** | Edit existing Exercise | There is at least one exercise part of the training and the user can access the edit form of it. | Training | After clicking the appropriate button the user is forwarded to the form that contains the data of the exercise and can make changes and submit them. |

## Test Case: User can delete exercise

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 5** | Edit existing Exercise | There is at least one exercise part of the training and the user can delete it by clicking the appropriate button. | Training | After clicking the appropriate button the exercise is removed from the training list. |

## Test Case: The exercise and training calories are calculated and displayed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 7** | Calculate Exercise Calories | There is a logic that calculates the exercise calories after it is added to the training. | Exercise | After the exercise is submitted the calories are calculated based on the exercise type and duration. |
| **TC-02** | Calculate Training Calories | There is a logic that calculates and displays the total calories of the training based on the sum of all exercise’s calories. | Training | Each time any changes are made to the exercises of a training the total amount of calories are being calculated and displayed in the top corner of the table. |

## Test Case: User Logout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **US** | **Name** | **Pre-condition** | **Test data** | **Expected result** |
| **TC-01** | **User Story 8** | User Logout | There is a logout button in the navbar and a logic that logs out the user. | User | After clicking the logout button the user is logged out and forwarded to the login page. |

# User Acceptance Tests

## Requirements

* User is able to access the web-application.
* User can view the content of the homepage.
* User can authenticate in the system.
* User can view the exercises of the training and the details of the exercises.
* User is able to access the form to add new exercises.
* User can select a exercise from the dropdown, can specify the duration of the exercise, the position and submit the exercise.
* The exercise appears on the training plan.
* User can edit a specific exercise in a form with the data persistent already in the form.
* User can delete any of the exercises from the training.
* User can sign out of the system.

## Plan

1. User authenticates in the system.
2. User submits a new exercise to the training through the appropriate form.
3. User can view the added exercise in the training and can edit it.
4. The modification user made is updated in the training and specific exercise.
5. User deletes a specific exercise from the training.
6. User signs out of the system.

## Confirmation of Business Objectives

When the user accesses the application he is forwarded to the homepage, he can access the authentication menu through a navbar, after authenticating the user has access to the features of the application. The Training page displays the list of the exercises that forms it with all the details and the possibility to delete or edit exercises. The submission of a new exercise is done in a separate form. Any exercise can be edited by pressing the appropriate button and all the details are prefilled in the form. Any exercise can be deleted from the training using the appropriate button. It is possible to sign out and the user is forwarded to the appropriate page.