**Testing & Integration Plan**

FH Mobile Application

Version 1.0

Prepared by

Omar Rivera

Andrew Poirier

Daven Amin

Rick Rejeleene

Table of Contents

[1.0 Introduction 3](#_Toc384900008)

[2.0 Modular Test 3](#_Toc384900009)

[2.1 User Platform Requirements 3](#_Toc384900010)

[2.1.1 Bullet/Table 1 – Title of Test Case 3](#_Toc384900011)

[2.2 Account Requirements 3](#_Toc384900012)

[2.2.1 Test account login 3](#_Toc384900013)

[2.2.2 Test account registration 4](#_Toc384900014)

[2.2.3 Test account persistence 4](#_Toc384900015)

[2.2.4 Test account updates 5](#_Toc384900016)

[2.3 Workout schedule requirements 5](#_Toc384900017)

[2.3.1 Bullet/Table 1 – Title of Test Case 5](#_Toc384900018)

[2.4 Validation Requirements 5](#_Toc384900019)

[2.4.1 Bullet/Table 1 – Title of Test Case 5](#_Toc384900020)

[2.5 Server Platform Requirements 6](#_Toc384900021)

[2.5.1 Bullet/Table 1 – Title of Test Case 6](#_Toc384900022)

[2.6 Server Requirements 6](#_Toc384900023)

[2.6.1 Bullet/Table 1 – Title of Test Case 6](#_Toc384900024)

[2.7 User Account Requirements 6](#_Toc384900025)

[2.7.1 Bullet/Table 1 – Title of Test Case 6](#_Toc384900026)

[2.8 User Interface Requirements 7](#_Toc384900027)

[2.8.1 Bullet/Table 1 – Title of Test Case 7](#_Toc384900028)

[2.9 Database Requirements 7](#_Toc384900029)

[2.9.1 Bullet/Table 1 – Title of Test Case 7](#_Toc384900030)

[2.10 Performance Requirements 7](#_Toc384900031)

[2.10.1 Bullet/Table 1 – Title of Test Case 7](#_Toc384900032)

# 1.0 Introduction

This document describes the test plan and integration for the FH mobile application.

# 2.0 Modular Test

This section is broken down in to the different modular that will be tested in this section

## 2.2 Account Requirements

### 2.2.1 Test account login

|  |  |
| --- | --- |
| Requirement Number | 3.2.2 |
| Requirement Name | Account Requirements – Allow the user to log into the service with account credentials |
| Pre-conditions | User has a valid account (username and password) on the service |
| Procedures | |  |  | | --- | --- | |  | * Open the FH mobile application * Type in valid username into “username” field * Type in valid password into “password” field * Press “Login” button | |
| Post Conditions | Mobile application proceeds to Workouts page |
| Test Results | PASS if post conditions are true, FAIL if application hangs or returns an error/alert. |

### 2.2.2 Test account registration

|  |  |
| --- | --- |
| Requirement Number | 3.2.2 |
| Requirement Name | Account Requirements – Allow the user create a new account in the service |
| Pre-conditions | NONE |
| Procedures | |  |  | | --- | --- | |  | * Open the FH mobile application * Type in a desired username into “username” field * Type in a desired password into “password” field * Press “Register” button | |
| Post Conditions | If password does not meet the requirements (is not between 1 and 20 alphanumeric characters), an alert is displayed requesting the password be modified to meet the password requirements.  If username does not meet the requirements (is not between 1 and 10 alphanumeric characters), an alert is displayed requesting the username be modified to meet the username requirements.  If the username already exists in the system, an alert is displayed requesting another valid username be chosen.  If username and passwords meet the requirements and username does not already exist in the system, an alert is displayed stating “success” and application proceeds to Workouts page |
| Test Results | PASS only if all post conditions are true, FAIL otherwise. |

### 2.2.3 Test account persistence

|  |  |
| --- | --- |
| Requirement Number | 3.2.2 |
| Requirement Name | Account Requirements – Persist user account information when user is not logged into service. |
| Pre-conditions | All tests in section 2.9 – Database Requirements PASS |
| Procedures | * Verify that database tables persist between reboots of database machine |
| Post Conditions | Database tables remain consistent between machine reboots |
| Test Results | PASS if post conditions are true, FAIL database tables are dropped between reboots. |

### 2.2.4 Test account updates

|  |  |
| --- | --- |
| Requirement Number | 3.2.2 |
| Requirement Name | Account Requirements – Update user account information when user is logged into service. |
| Pre-conditions | All tests in section 2.3 – Workout schedule requirements PASS  All tests in section 2.4 – Validation requirements PASS  All tests in section 2.5 – Server requirements PASS  All tests in section 2.7 – User account requirements PASS |
| Procedures | N/A |
| Post Conditions | N/A |
| Test Results | PASS if all tests referenced in pre-conditions PASS, otherwise FAIL. |

## 2.3 Workout schedule requirements

### 2.3.1 Test account registration

|  |  |
| --- | --- |
| Requirement Number | 3.2.3 |
| Requirement Name | Workout Schedule Requirements – Allow the user to create one or more workout schedules composed of a workout metric, a frequency per week, and a total number of weeks  Workout Schedule Requirements – Allow the user to specify a workout metric as a number and a textual description |
| Pre-conditions | User is logged in and at the Add Workout screen |
| Procedures | |  |  | | --- | --- | |  | * Select a start date * Select an end date * Type in a number of workouts per week (Frequency) * Select a metric (Repetitions, Distance, or Time) * Type a description * Press “Submit” | |
| Post Conditions | If the Start Date is not a valid (a calendar date at the current date or later), an alert is displayed requesting the field be changed to indicate a valid date.  If the End Date is not valid (a calendar date later than the Start Date), an alert is displayed requesting the field be changed to indicate a valid date.  If the Frequency is not a positive integer number, an alert is displayed requesting the field be changed to a positive integer number.  If Description is blank or contains non ASCII characters, an alert is displayed requesting the field be changed to contain at least one character and only valid ASCII characters.  If all fields contain valid input, an alert is displayed stating “success”, and the application proceeds to the Workouts page. |
| Test Results | PASS only if all post conditions are true, FAIL otherwise. |

### 2.3.2 Test user authentication preferences

|  |  |
| --- | --- |
| Requirement Number | 3.2.3 |
| Requirement Name | Workout Schedule Requirements – Allow the user to provide a user name to authenticate that workout metrics have been met for a particular workout schedule |
| Pre-conditions | System contains more than one user account.  User is logged in and at User Information page |
| Procedures | |  |  | | --- | --- | |  | * Verify that “Trainer” dropdown field contains all other user account names. * Verify that selection of “Trainer” remains consistent after leaving and re-entering User Information page * Verify that selection of “Trainer” remains consistent after restarting application | |
| Post Conditions | Trainer field contains all other user account names and field selection persists between application sessions and navigation |
| Test Results | PASS if post conditions are true, else FAIL |

## 2.6 Server Requirements

### 2.6.1 Test workout creation/retrieval

|  |  |
| --- | --- |
| Requirement Number | 3.2.6 |
| Requirement Name | Server Requirements - Server shall allow for the creation, storage, and retrieval of *workout schedule* information associated with a specific user account.  Server Requirements -Server shall allow for the creation, storage, and retrieval of *workout metric* information associated with a specific user account and *workout schedule*. |
| Pre conditions | Tester has a Javascript execution environment and is connected to the server machine – able to communicate with server via HTTP(S)  Tester is logged into server |
| Procedures | * Run fitnesshealth.backend.js “createWorkout” with valid arguments. * Run fitnesshealth.backend.js “getWorkouts” with valid arguments, and dump received data to the console. |
| Post Conditions | Console contains workout data submitted during the “createWorkout” call. |
| Test Results | PASS if post-condition holds, FAIL if function throws an error or if console output is inconsistent with function input. |

### 2.6.2 Test workout creation/retrieval

|  |  |
| --- | --- |
| Requirement Number | 3.2.6 |
| Requirement Name | Server Requirements - Server shall allow for the creation, storage, and retrieval of user account information. |
| Pre-conditions | Tester has a Javascript execution environment and is connected to the server machine – able to communicate with server via HTTP(S) |
| Procedures | * Run fitnesshealth.backend.js “register” with a new username and password. * Run fitnesshealth.backend.js “login” with same username and password. * Run fitnesshealth.backend.js “refresh” and dump received data to the console. |
| Post Conditions | Console contains the username specified by the tester, as well as default user information. |
| Test Results | PASS if post-condition holds, FAIL if any of the functions throw an error or if console output is empty or inconsistent with post-condition. |

### 2.6.3 Test user/trainer association

|  |  |
| --- | --- |
| Requirement Number | 3.2.6 |
| Requirement Name | Server Requirements - Server shall allow for many-to-one *authenticator* associations of one or more user accounts with a single user account. |
| Pre-conditions | Tester has a Javascript execution environment and is connected to the server machine – able to communicate with server via HTTP(S)  Server contains two newly registered user accounts (account1 and account2) with existing and known workout information |
| Procedures | * Login to account1 * Run fitnesshealth.backend.js “addValidatedUser” and provide account2 as an argument * Login to account2 * Run fitnesshealth.backend.js “getValidatableWorkouts” and dump received data to console. |
| Post Conditions | Console contains workout information associated with account1. |
| Test Results | PASS if post-condition holds, FAIL if any of the functions throw an error or if console output is empty or inconsistent with post-condition. |