

***Software Engineering***

***Test Plan Document:***

***Iteration 1***

*James Dadson, Jonathan Miller, Colin Newton, Michael Yilma*  
*Affiliation*  
*teammapletree.weebly.com*  
*3-4-2019*

**Table of Contents**

**1** **Introduction 3**

1.1 Iteration Overview 3

1.2 Items Not Covered by These Test Cases 3

1.3 Bug Tracking 3

1.4 Quality Control 3

1.5 Adequacy Criterion 3

**2** **Test Cases 4**

**3** **Iteration 0 – Bug Reports 5**

**4** **Customer Acceptance Testing 6**

# 

# **Introduction**

## 

## **Iteration Overview**

Iteration 1 includes testing the login, register, and add an exercise pages based on inputs and overall behavior. The login page involves testing for correct and incorrect username and password combinations and creating a session that will be held throughout until the user logs out or exits the application. The register page will be tested for correct and incorrect input to make sure the application behaves in the way we expect. The add an exercise page will also be tested for correct and incorrect input to make sure the application behaves in the way we expect.

***Guidelines:***

* On the select an exercise page the field should accept numbers and letters. This field will not accept any special characters.
* On the select an exercise page the sets, reps, and weight should only accept numbers in their field. They will not accept letters or special characters.
* The login page will be tested to confirm the username and password.
* The register page will not allow users to do code injections. Will also only allow letters for the name field and will do pattern checking for passwords.

## **Items Not Covered by These Test Cases**

The create a profile function will not be tested, since the user’s profile will be populated automatically from the information they provide when they create an account, therefore the testing for that input will be tested on the create an account page.

## **Bug Tracking**

We will be using a database management system to track bugs in the database, namely, PHPmyAdmin. The DBMS will make it easier to identify where these bugs are occuring while practically testing features will be our main method of reporting the bugs*.*

## **Quality Control**

We will develop quality standards that every member of the group will adhere to during the testing process. We will confirm that we have a good test plan by ensuring that we cover every function of the iteration.

## **Adequacy Criterion**

As a team, we decided that if 95% of the tests pass, then testing will be complete for this iteration. We plan to discuss this number with our client to decide if they agree.

# **Test Cases**

*This section details for each test the exact steps necessary to perform the test and the expected results, and provides a section for recording the observed results. This table is the most important section of the document. It comprises both the plan and the results of executing the plan. That is, the testing team fills it in as the testing process proceeds. Failed tests should result in bug reports.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test #** | **User Story** | **Action / Input** | **Expected Result** | **Actual Result** | **P/F** | **Notes** | **Date test performed** | **Tester** |
| 1.a | Create An Account | Add first name data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan  Miller |
| 1.b | Create An  Account | Add last name data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 1.c | Create An  Account | Add user name data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 1.d | Create An  Account | Add RU ID data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 1.e | Create An  Account | Add E-Mail address data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 1.f | Create An  Account | Add password data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 1.g | Create An  Account | Add confirm password data | Data is added to the database, only if the passwords match | Confirms password correctly | P | Confirms correctly | 3/4/19 | Jonathan Miller |
| 1.h | Create An Account | Attempt code injection | Application doesn’t allow code injection | Data not matching pattern is not added. | P | Detects Correctly | 3/4/19 | Jonathan Miller |
| 2.a | Select An Exercise | Add Exercise type data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Colin Newton |
| 2.b | Select An Exercise | Attempt code injection | Application doesn’t allow code injection | Application allows code injection | F | Doesn’t sanitize input | 3/4/19 | Colin Newton |
| 3.a | Add a Set | Add set type data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Colin Newton |
| 3.b | Add a Set | Attempt code injection | Application doesn’t allow code injection | Application allows code injection | P | Doesn’t sanitize input | 3/4/19 | Colin Newton |
| 3.c | Add a Set | Provides client and server side validation for input | Application doesn’t allow incorrect input | Application doesn’t allow incorrect input | P | Doesn’t add incorrect input | 3/4/19 | Colin Newton |
| 4.a | Record Weight | Add weight data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Colin Newton |
| 4.b | Record Weight | Attempt code injection | Application doesn’t allow code injection | Application allows code injection | P | Doesn’t sanitize input | 3/4/19 | Colin Newton |
| 4.c | Record Weight | Provides client and server side validation for input | Application doesn’t allow incorrect input | Application doesn’t allow incorrect input | P | Doesn’t add incorrect input | 3/4/19 | Colin Newton |
| 5.a | Record Reps | Add reps data | Data correctly adds to database | Data correctly adds to database | P | Adds Correctly | 3/4/19 | Jonathan Miller |
| 5.b | Record Reps | Attempt code injection | Application doesn’t allow code injection | Application allows code injection | P | Doesn’t sanitize input | 3/4/19 | Jonathan Miller |
| 5.c | Record Reps | Provides client and server side validation for input | Application doesn’t allow incorrect input | Application doesn’t allow incorrect input | P | Doesn’t add incorrect input | 3/4/19 | Colin Newton |
| 6.a | Login Username | Enter Username | Valid username | Data is correctly verified and user is logged in | P | User is logged in | 3/4/19 | Jonathan Miller |
| 6.b | Login Username | Attempt code injection | Application doesn’t allow code injection | Application allows code injection | F | Doesn’t sanitize input | 3/4/19 | Jonathan Miller |
| 7.a | Login Password | Enter Password | Valid password | Data is correctly verified and user is logged in | P | User is logged in | 3/4/19 | Jonathan Miller |
| 7.b | Login Password | Attempt Code injection | Application doesn’t allow code injection | Application allows code injection | F | Doesn’t sanitize input | 3/4/19 | Jonathan Miller |

***Guidelines:***

* *Although the information in the table above indicates that each test comprises a line, actual descriptions may require multiple lines.*
* *The* ***Test #*** *can represent a category or set of tests related to a user story. In these situations, add a suffix to the number using small Roman letter. For example, use Test 2.a, Test 2.b, …*
* *All tests should have a specific purpose. For example, a test may evaluate the satisfaction of a specific requirement, in which case, the user story number should be included.*
* *A test (****Action****) may comprise a series of steps. Each step should be specific with regard to relevant user input and data files. That is, the information in this document is a prescription for executing the test.*
* *Some steps may be common to more than one Test. Some Tests may take up where others left off. There is no need to duplicate this content. In these cases, refer in the* ***Action*** *section to the* ***Test #*** *and Step Numbers that should be performed.*
* ***Expected Results*** *describe the exact output that should be produced by the test.* ***Actual Output*** *is what happened when the test was run. Note that for non-functional tests, the contents of this field might include elapsed time, memory consumed, etc. For failing tests, the output might include information about exceptions raised.*
* ***P/F*** *denotes Pass / Fail. A test should only pass if its expected output matches its actual output.*
* *The* ***Notes*** *field can provide details so that the team can repeat the test in future iterations.*

# **Iteration 0 – Bug Reports**

*This section details bugs (or failed tests) found during the current iteration..*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Originator** | **Submit Date** | **Summary** | **Severity**  [Critical | major | normal | minor | trivial | enhancement] | **Platform**  [PC | Mac] | **OS**  [Windows | Mac | Linux] | **Browser**  [IE | Firefox | Chrome | Safari] | **Completion Date** |
| 2.b | Colin Newton | 2/25/19 | Select An Exercise allows code injection | Critical | Both | All | All |  |
| 1.a | Colin Newton | 3/4/19 | Datalist doesn’t work on Safari | Normal | Mac | Mac | Safari |  |
| 1.b | Colin Newton | 3/4/19 | Datalist doesn’t work on iPhone | Critical | Mac | iOS | All |  |

# **Customer Acceptance Testing**

*This section details for each test the steps necessary for the customer to perform acceptance testing and the expected results, and provides a section for recording the observed results. The customer should fill in this table at then end of the iteration/phase. Failed tests should result in bug reports.*

Customer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test #** | **Feature** | **Action / Input** | **Expected Result** | **Actual Result** | **P/F** | **Notes** | **Date test performed** |
| 1.a | Create An Account | Add first name data | Data correctly adds to database |  |  |  |  |
| 1.b | Create An  Account | Add last name data | Data correctly adds to database |  |  |  |  |
| 1.c | Create An  Account | Add user name data | Data correctly adds to database |  |  |  |  |
| 1.d | Create An  Account | Add RU ID data | Data correctly adds to database |  |  |  |  |
| 1.e | Create An  Account | Add E-Mail address data | Data correctly adds to database |  |  |  |  |
| 1.f | Create An  Account | Add password data | Data correctly adds to database |  |  |  |  |
| 1.g | Create An  Account | Add confirm password data | Data is added to the database, only if the passwords match |  |  |  |  |
| 2.a | Select An Exercise | Add Exercise type data | Data correctly adds to database |  |  |  |  |
| 3.a | Add a Set | Add set type data | Data correctly adds to database |  |  |  |  |
| 4.a | Record Weight | Add weight data | Data correctly adds to database |  |  |  |  |
| 5.a | Record Reps | Add reps data | Data correctly adds to database |  |  |  |  |
| 6.a | Login Username | Enter Username | Valid username |  |  |  |  |
| 7.a | Login Password | Enter Password | Valid password |  |  |  |  |