Verification and Validation Report: Software Eng 4G06

Team 2, Parnas' Pals
William Lee
Jared Bentvelsen
Bassel Rezkalla
Yuvraj Randhawa
Dimitri Tsampiras
Matthew McCracken

March 8, 2023

1 Revision History

Date	Version	Notes
March 8 2023	1.0	Initial draft

Contents

1 Revision History **i**

2 User Testing Summary **1**

3 Functional Requirements Evaluation **1**

3.0.1 Workout Routine Tests . . . 2
3.0.2 Exercise Tests . . . 2
3.0.3 Quantifier Tests . . . 3
3.0.4 Publicity Tests . . . 3
3.0.5 Workout Routine Saving Tests . . . 4
3.0.6 Browsing Workout Routine Tests . . . 4
3.0.7 User Profile Tests . . . 4
3.0.8 Fitness Goal Tests . . . 5
3.1 Functional Requirements Results . . . 5

4 Nonfunctional Requirements Evaluation **6**

4.1 Look and Feel Testing . . . 6
4.1.1 Usability and Humanity Tests . . . 6
4.1.2 Performance Tests . . . 7
4.1.3 Operational and Environment Tests . . . 7
4.1.4 Maintainability and Support Tests . . . 7
4.1.5 Security Tests . . . 7
4.1.6 Cultural Requirements Tests . . . 8
4.1.7 Legal Requirements Tests . . . 8

5 Unit Testing **8**

6 Changes Due to Testing **8**

6.1 Timer . . . 9
6.2 Auto Login . . . 9
6.3 Styling . . . 9
6.4 Tracking user Fitness Goals . . . 9

7 Automated Testing **9**

8 Trace to Requirements **9**

9 Trace to Modules **12**

10 Code Coverage Metrics **12**

List of Tables

1 Functional System Tests to Functional Requirement Matrix . . 10
2 Non-Functional System Tests to Non-Functional Requirement Matrix 11
3 Non-Functional System Tests to Non-Functional Requirement Matrix 11
4 Test to Module Traceability Matrix 12

List of Figures

This document serves to record the results of testing and validation performed on the Olympian system, as well as any design decisions or changes made due to the results of said testing.

2 User Testing Summary

Unless Specified these tests were performed with a semi-structured interview where users were brought scenarios then asked questions relevant to that scenario. Most Scenarios followed the key use cases of the application.
Examples include:

1. Creating an Account.
2. Logging into an account.
3. Creating a Program.
4. Creating a Workout.
5. Creating an Exercise.
6. Searching for an Exercise.
7. Using the Discovery page to browse workouts and programs.
8. Finding new workouts from the Discovery page.
9. Starting a workout (as if the user were to use the application during their workout).

3 Functional Requirements Evaluation

During the user tests, users were asked to preform extra actions based on the FR test cases displayed in the VnV Plan. Some example actions included:

1. During workout creation, could you make your workout private?
2. Could you try failing your password?
3. Asking the user to attempt all actions when starting a workout.

As for the non-user tests, manual unit tests were preformed in order to measure metrics.

The summary of the test cases are displayed below:

3.0.1 Workout Routine Tests

1. **test-WR-1**:A Workout Routine can be created.
   Input: User inputs required data for a workout routine.

   Expected Output: A workout routine is stored in the database and is accessible to the user that created it. If the user determined that the workout to be public then it should be publicly visible.

   Test result: [PASS] Workout routines can be created and are added to an existing Program.

2. **test-WR-2**: Editing a Workout Routine.
   Input: User edits a workout routine with new values.

   Expected Output: A workout routine is updated with new values in the database and new values are visible to accessible users.

   Test result: [PASS] Workout routines can be edited, including changing the names, tags, and amending and removing exercises.

3.0.2 Exercise Tests

3. **test-EX-1**:Adding an Exercise to a Workout Routine.
   Input: A user created exercise with the parameters required for the exercise.

   Expected Output: The workout routine should include the added exercise. **Test result**: [PASS] Users can add Exercises to Workout Routines and customize reps and sets.

4. **test-EX-2**:Removing an Exercise from a Workout Routine.
   Input: The user chooses to remove an exercise.

   Expected Output: Some feedback indicating that an exercise has been removed. The workout routine no longer contains the removed exercise.

Test result: [PASS] Users can remove exercises from Workout Routines.

5. **test-EX-3**:Limiting Exercises to a Workout Routine.
   Input: The number of exercises required to reach the limit of exercises per workout routine.

   Expected Output: A notification to the user, before the limiting exercise notifying them of the limit, and preventing them from adding another exercise.

   Test result: [FAIL] Users cannot limit exercises to workout routines.

3.0.3 Quantifier Tests

6. **test-QT-1**:Adding Quantifiers to an Exercise.
   Input: A number and unit of measurement describing an exercise.

   Expected Output: The exercise now holds the given quantifier and is displayed to the user.

   Test result: [PASS] Users can add quantifiers to their exercises.

7. **test-QT-2**:Removing Quantifiers from an Exercise.
   Input: Removal of a quantifier.

   Expected Output: The exercise no longer holds a quantifier and is not displayed to the user.

   Test result: [FAIL] Users cannot remove their quantifiers from their exercises. These come as default.

8. **test-QT-3**:Editing Quantifiers of an Exercise.
   Input: A number and unit of measurement describing an exercise.

   Expected Output: The exercise now holds the updated quantifier.

   Test result: [PASS] Users can edit their quantifiers.

3.0.4 Publicity Tests

9. **test-PB-1**:Publicizing a Workout Routine.
   Input: An edit or addition to a workout routine to make it public.

Expected Output: The workout routine is declared public in the database and is now visible to all users.

Test result: [PASS] Users can publicize a workout routine.

10. **test-PB-2**:Privatizing a Workout Routine.
    Input: An edit or addition to a workout routine to make is private.

    Expected Output: The workout routine is declared private in the database and is no longer visible to any user except the creator.

    Test result: [PASS] Users can privatize a workout routine.

3.0.5 Workout Routine Saving Tests

11. **test-WS-1**: Saving a Public Workout Routine.
    Input: The other user workout routine is saved.

    Expected Output: The workout routine is now visible and accessible under the saved workout routines.

    Test result: [PASS] Users can save a public workout routine.

3.0.6 Browsing Workout Routine Tests

12. **test-BS-1**:Browsing Workout Routines.
    Input: navigation movements to view the public workout routines.

    Expected Output: Multiple public workout routines should be displayed.

    Test result: [PASS] Users can browse public programs according to specific categories.

13. **test-BS-2**:Search Workout Routines.
    Input: Search string inputs to view the public workout routines.

    Expected Output: Public workout routines that match the search criteria should be displayed.

    Test result: [PASS] Users can search publicly published routines using custom search text.

3.0.7 User Profile Tests

14. **test-UP-1**:Creating a User Profile.
    Input: The required parameters for creating a profile.

    Expected Output: The created profile is stored in a user database and the user should be logged into their profile.

    Test result: [PASS] Users can create their profiles when signing up.

15. **test-UP-2**:Viewing Other User Profile.
    Input: Search criteria for the searched user profile.

    Expected Output: A user profile is displayed with their public routines, fitness goals, and other public profile data.

    Test result: [FAIL] Users cannot view other user profiles.

3.0.8 Fitness Goal Tests

16. **test-FG-1**:Creating a Fitness Goal.
    Input: Progress points towards a given fitness goal at a given date.

    Expected Output: The progress displayed towards a fitness goal should be visually updated and numerically updated in the database with a date

    Test result: [FAIL] Users cannot create a fitness goal.

17. **test-FG-2**:Progressing a Fitness Goal.
    Input: Progress points towards a given fitness goal at a given date.

    Expected Output: The progress displayed towards a fitness goal should be visually updated and numerically updated in the database with a date

    Test result: [FAIL] Users cannot progress fitness goals.

3.1 Functional Requirements Results

Tests Passed: 12
Tests Failed: 5
Total Tests: 17

4 Nonfunctional Requirements Evaluation

After users were brought through the test case scenarios they were asked questions based on the NFR test cases displayed in the VnV Plan. Some example questions included:

1. How did you find the speed of the application? Did you notice any latency or lagging?
2. Any time the application prompted you to do an action such as find an exercise, was it clear on what to do?
3. Considering the navigation of this program was it hard to locate various operations and actions from any given point?

As for the non-user tests, unit tests were preformed in order to measure metrics based on the application performance.

The summary of the test cases are displayed below:

4.1 Look and Feel Testing

1. **test-LF-1**: Style.
   Test result:

4.1.1 Usability and Humanity Tests

2. **test-UH-1**: Text Sizing and Visibility.
   Test result:
3. **test-UH-2**: Text Language.
   Test result:
4. **test-UH-3**: Learnability.
   Test result:
5. **test-UH-4**: Understandability.
   Test result:

6. **test-UH-5**: Hearing and Audio considerations.
   Test result:

7. **test-UH-6**: Use of Colour and Contrast.
   Test result:

4.1.2 Performance Tests

8. **test-PF-1**: Speed and Latency.
   Test result:

9. **test-PF-2**: Accuracy and Precision of Quantifiers.
   Test result:

10. **test-PF-3**: Availability and Uptime.
    Test result:

11. **test-PF-4**: User Capacity.
    Test result:

12. **test-PF-5**: Scalability of User Capacity.
    Test result:

4.1.3 Operational and Environment Tests

13. **test-OE-1**: Supported Systems.
    Test result:

4.1.4 Maintainability and Support Tests

14. **test-MS-1**: Maintenance.
    Test result:

4.1.5 Security Tests

15. **test-SEC-1**: Private and Public Details.
    Test result:

16. **test-SEC-2**: Passwords.
    Test result:

17. **test-SEC-3**: Client Server Privacy.
    Test result:

18. **test-SEC-4**: Data storage and logging.
    Test result:

19. **test-SEC-5**: Data Backups.
    Test result:

4.1.6 Cultural Requirements Tests

20. **test-CR-1**: Profanity and Inappropriate Language.
    Test result:

21. **test-CR-2**: Reporting Offensive Language.
    Test result:

4.1.7 Legal Requirements Tests

22. **test-LR-1**: Age and Gender Use.
    Test result:

23. **test-LR-2**: Data Protection.
    Test result:

5 Unit Testing

6 Changes Due to Testing

Listed below are some of the changes that are planned (some implemented) as a result of the testing and user testing.

6.1 Timer

During user testing one of the most desired additions to this application was a built in timer that would help for rest period and counting during workout sets. This addition makes sense as it helps users keep track of their timings without having the close the application to open up a timer.

6.2 Auto Login

Another user suggestion was that when logging into the application from a point where the user has already logged in (e.g. logged in then closed the application then open again) then it is 'tedious' to log in again. There were many reasons for this such as users closing the application to change their music or check a text etc. Overall this change will help with the overall efficiency and user experience by bringing a more user friendly environment with less user actions in order to login and continue working out.

6.3 Styling

There were many positive reviews with the looks and styling of the application, However one of the minor changes that was mentioned was that some of the designed were 'too close together'. Taking in this feedback, the visual components may need to be spaced out more or even re-formatted in such a way that everything is less squashed together.

6.4 Tracking user Fitness Goals

Tracking Fitness goals has always been a stretch goal of this application. Some users asked to see a visual representation such as a graph over time or a numerical increment to help them see progress. This change will help users visually see and track their fitness goals and progress easier.

7 Automated Testing

8 Trace to Requirements

Below is the link from requirements to test cases.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
WR1	X										
WR2	X										
EX1		X									
EX2		X									
EX3		X									
QT1			X								
QT2			X								
QT3			X								
PB1				X							
PB2				X							
WS1					X						
BS1						X					
BS2						X					
UP1							X				
UP2								X			
FG1									X		
FG2										X	X

Table 1: Functional System Tests to Functional Requirement Matrix

	NFR1	NFR2	NFR3	NFR4	NFR5	NFR6	NFR7	NFR8	NFR9	NFR10	NFR11
LF1	X										
UH1		X									
UH2			X								
UH3				X							
UH4					X						
UH5						X					
UH6							X				
PF1								X			
PF2									X		
PF3										X	
PF4											X

Table 2: Non-Functional System Tests to Non-Functional Requirement Matrix

	NFR12	NFR13	NFR14	NFR15	NFR16	NFR17	NFR18	NFR19	NFR20	NFR21	NFR22
PF5	X										
OE1		X									
MS1			X								
SEC1				X							
SEC2					X						
SEC3						X					
SEC4						X					
SEC5							X				
CR1								X			
CR2									X		
LR1										X	
LR2											X

Table 3: Non-Functional System Tests to Non-Functional Requirement Matrix

9 Trace to Modules

Module Name	Associated Test Ids
Exercise Module	
Workout Module	
Workout Routine Module	
User Login Module	
User Registration Module	
User Profile Module	
User Fitness Goal Module	
Workout Browser Module	
Creation Module	
Workout Creation Module	
Workout Routine Creation Module	
Exercise Creation Module	
Timed Sequence Module	
Database Communicator Module	

Table 4: Test to Module Traceability Matrix

10 Code Coverage Metrics

References

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. **Question**

   In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

2. **Answer**

   The VnV plan differed from the actual activities that were performed in many ways, most notably the user testing. During user testing some actions that were a part of the test plan e.g. create a workout, the user may not know where to go. This seemed to be from a lack of software to user communications but this caused some of the testing to be performed a little differently. Occasionally we as testers would have to step in and assist the user (which is not what we wanted) to help them out.

   In the VnV plan we stated that we would conduct surveys on some of the 'Aesthetics' of the design, however this was very tedious to do as a tester so we ended up asking the individuals we tested on about the look and feel of the application. As for looking at modifications towards the VnV plan and future plans, we should focus on the feasibility of what we say we are going to do vs what will actually be done. To give our team the benefit of the doubt there were many midterms and course work load build ups during this time that we did not anticipate for as well. Another way we deviated from the VnV plan was that large data-management testing is very difficult and takes some time. We were not able to complete certain tests such as data backups due to the complexity that this requires.