

Verification and Validation Report: Software Engineering

Team 14, Reach
Aamina Hussain
David Moroniti
Anika Peer
Deep Raj
Alan Scott

March 6, 2024

1 Revision History

Date	Version	Notes
03/04/2024	1.0	Add results for some functional tests
03/06/2024	1.1	Add results for unit tests + reflection

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
UNT	Unit Test
UT	Usability Test
ST	Searching Test
DT	Data Test
BT	Bookmarking Test
PIT	Program Info Test
PT	Performance Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
4	Nonfunctional Requirements Evaluation	2
4.1	Usability	2
4.2	Performance	2
4.3	Maintainability	3
4.4	Security	3
5	Comparison to Existing Implementation	3
6	Unit Testing	3
7	Changes Due to Testing	4
8	Automated Testing	5
9	Trace to Requirements	5
10	Trace to Modules	5
11	Code Coverage Metrics	5

List of Tables

List of Figures

3 Functional Requirements Evaluation

User Data

1. DT-1 Result: **Passed** - New account can successfully be created, and data is stored successfully in the database. Therefore, the tester was able to verify that the data is persistent, and so this test was successful.
2. DT-2 Result: **Passed** - Account information can successfully be updated, and the changes are reflected in the database. Therefore, the tester was able to verify that the updates are persistent, and so this test was successful.
3. DT-3 Result: **Not run** - reason being that the database application we used will ensure the issue brought up by this test can never happen.

Program Information

1. PIT-1 **Passed** - The tester was able to verify that the FAQ system can be viewed by a user (whether they are logged in or not), and so this test was successful.

Searching

1. ST-1 **Not run** - Functionality tested by this test was removed from the requirements prior to conducting testing.
2. ST-2 **Passed** - The tester was able to conduct a search based on a patient profile, and view the resulting trials. Therefore, this test was successful.
3. ST-3 **Failed** - When attempting to search for trials with no profile data selected, the app did not behave as intended (it still tried to conduct a search), therefore this test was unsuccessful.
4. ST-4 **Passed** - The tester was able to switch between multiple profiles, and conduct a search based on each respective profile.

5. ST-5 **Passed** - Upon selecting a displayed trial, the tester was able to verify that the location displayed on the map updates and displays correctly.

Bookmarking

1. BT-1 **Passed** - The tester was able to save a trial, and view the trial in the bookmarks page. Additionally, there was an indication that the trial was saved, upon saving the trial. Therefore, this test was successful.
2. BT-2 **Passed** - The tester was able to delete a trial that was previously saved, and verified that the deletion was recognized in the frontend (i.e., by no longer being part of the list of saved trials). Therefore, this test was successful.
3. BT-3 **Passed** - The tester was able to verify that the details provided by the saved trial are the same as the details related to the trial on the search trial page.
4. BT-4 **Passed** - The tester was able to verify that the saved trials are filtered correctly upon selecting a profile to filter by. When selecting a profile, only the trials saved under that profile are displayed, and when selecting no profile, all saved trials are displayed to the user.

4 Nonfunctional Requirements Evaluation

4.1 Usability

4.2 Performance

1. PT-1 **Passed** - On average, the tests to load trials take on average 4.1 seconds. As this is below the benchmark of 5 seconds, the test passes.
2. PT-2 **Passed** - All endpoints were tested manually to measure the time taken to establish connections between endpoints. All tests succeeded in under 1 second, so the tests pass.

4.3 Maintainability

1. MT-1 **Failed** - Modified code in the pull request has been routinely and automatically linted by a Github CI component upon creation of pull requests. Some pull requests have been rejected by the linter, so the test fails.

4.4 Security

1. SEC-1 **Failed** - The tester entered the password "pass" which does not conform to the NIST guidelines. The password was accepted and the account was created, so the test fails.
2. SEC-2 **Passed** - Two tests were run. Firstly, a tester with the correct password but incorrect username attempted to access the database. This test was successful as the tester was denied access to the database. Secondly, a tester with correct credentials attempted to log into the database. This attempt was successful, so the second part of the test was also successful.

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

1. UNT-1 Result: **Passing**
2. UNT-2 Result: **Passing**
3. UNT-3 Result: **Passing**
4. UNT-4 Result: **Passing**
5. UNT-5 Result: **Passing**
6. UNT-6 Result: **Passing**
7. UNT-7 Result: **Passing**

8. UNT-8 Result: **Passing**
9. UNT-9 Result: **Failing** - Functionality that is tested by this unit test is currently under a slight rework, which is causing the test to fail.
10. UNT-10 Result: **Failing** - Functionality that is tested by this unit test is currently under a slight rework which is causing the test to fail.
11. UNT-11 Result: **Passing**
12. UNT-12 Result: **Passing**
13. UNT-13 Result: **Passing**
14. UNT-14 Result: **Passing**
15. UNT-15 Result: **Passing**
16. UNT-16 Result: **Passing**
17. UNT-17 Result: **Passing**
18. UNT-18 Result: **Passing**
19. UNT-19 Result: **Passing**
20. UNT-20 Result: **Passing**
21. UNT-21 Result: **Passing**

7 Changes Due to Testing

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

- 8 Automated Testing
- 9 Trace to Requirements
- 10 Trace to Modules
- 11 Code Coverage Metrics
- References

Appendix — Reflection

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

1. 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.)

There were a few ways in which the VnV plan was different from the activities that were actually conducted. First, there were a few tests written for functionality that is no longer planned to be in the app. For example, the email notification system/module is no longer in scope, and as a result, tests were obviously not conducted for this module. Furthermore, there were some tests that did not need to be done as a direct result of the implementation, and more specifically, some of the tools that were used in the implementation. For example, a good chunk of the authentication tests did not actually need to be done, since Django provides an out of the box solution to authenticating users (which we were able to take advantage of). It would be very difficult to anticipate all of these changes in a future project, since there are so many variables affecting the testing stage (i.e., change in scope, implementation details, etc.). Fortunately, this is why we follow the iterative process, since it allows us to make continuous updates to our documentation, ensuring there is always consistency between the plan, report, the app itself, and the other documents (such as SRS, MIS, etc.).