

Verification and Validation Report: Software Engineering

Team 21, Alkalytics

Sumanya Gulati

Kate Min

Jennifer Ye

Jason Tran

April 4, 2025

1 Revision History

Date	Version	Notes
4 March 2025	1.0	Initial Draft
10 March 2025	1.1	Add Results
14 March 2025	1.2	Add DataVisualize Unit Tests
1 April 2025	1.3	Updated document to adhere to 236 , 237 , and 239 .
4 April 2025	1.4	Updated document to adhere to 238 .

2 Symbols, Abbreviations and Acronyms

symbol	description
CSV	Comma-Separated Values
FR	Functional Requirement
LFR	Look and Feel Requirement
MG	Module Guide
MSR	Maintainability and Support Requirement
NFR	Nonfunctional Requirement
OER	Operational and Environmental Requirement
PR	Performance Requirement
SR	Security Requirement
SRS	Software Requirements Specification
ST	System Test
UHR	Usability and Humanity Requirement
VnV	Verification and Validation
WCAG	Web Content Accessibility Guidelines

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Overview	1
4	Functional Requirements Evaluation	1
4.1	Data Input and Storage	1
4.2	Data Querying and Results	2
4.3	Data Analysis	2
4.4	Data Hygiene	3
4.5	User Access	3
4.6	Data Export	4
5	Nonfunctional Requirements Evaluation	4
5.1	Look and Feel	4
5.2	Usability	6
5.3	Performance	8
5.4	Operational and Environmental	12
5.5	Maintainability and Support	13
5.6	Security	14
6	Unit Testing	18
6.1	Data Storage Module	18
6.2	Data Retrieval Module	19
6.3	Visualization Module	20
6.4	User Management Module	20
7	Changes Due to Testing	21
7.1	Changes Due to Supervisor Feedback	21
7.2	Changes Due to Failed Tests	22
7.2.1	NFR-S3	22
7.2.2	NFR-UH4	22
8	Automated Testing	22
9	Trace to Requirements	22

10 Trace to Modules	26
11 Code Coverage Metrics	27

List of Tables

1	Unit tests summary for Data Storage Module	18
2	Unit tests summary for Data Retrieval Module	19
3	Unit tests summary for Data Retrieval Form Module	19
4	Unit tests summary for Data Retrieval useGraph Hook	19
5	Unit tests summary for Data Visualization Module	20
6	Unit tests summary for User Management Module	21
7	Traceability Matrix for Test Cases and Functional Requirements	23
8	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 1)	24
9	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 2)	25
10	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 3)	26
11	Traceability Matrix for System Test Cases and Modules	27

3 Overview

This document provides a comprehensive summary of the Verification and Validation (VnV) process for the Alkalytics project. It includes reports and analysis of system testing and unit testing results, along with traceability between test cases, the software modules in the [Module Guide](#) (MG), and the requirements outlined in the [Software Requirements Specification](#) (SRS). Tests and test procedures are derived from the [VnV Plan](#) document. Additionally, it highlights changes made to the implementation based on testing outcomes and user feedback.

4 Functional Requirements Evaluation

This section outlines the manual system tests that were run and their subsequent results and outputs for all listed functional requirements.

4.1 Data Input and Storage

1. FR-ST1

Control: Manual

Initial State: Database is running and ready to intake data

Input: `SAMPLE_.CSV_FILE`

*Note: Actual experiment data cannot be provided publicly.

Expected Output: The inputted data is stored in the system with proper labels and without any data loss or errors

Actual Output: Data successfully stored in the database with correct key-value pairs; no data loss or errors were detected.

Result: **Pass**

2. FR-ST1.1

Control: Manual

Initial State: Database is running and ready to intake data

Input: `SAMPLE_DATA_POINT: {NEW_EXPERIMENT_ROW}`

*Note: Actual experiment data cannot be provided publicly.

Expected Output: The inputted data is stored in the system with proper labels and without any data loss or errors.

Actual Output: The inputted data is stored in the system under the correct label and without any data loss or errors.

Result: **Pass**

4.2 Data Querying and Results

1. FR-ST2

Control: Manual

Initial State: The system is not running any jobs, and the user interface is cleared.

Input: **SAMPLE_PARAMETERS**: {Graph: Line, Experiment Dates: 2024-11-05, Parameter Type: Data Sheet, X Parameter: C1 Cond, Y Parameter: C1 Temp, X Axis Label: C1 Cond, Y Axis Label: C1 Temp, Graph Title: 2024-11-05 C1 Cond vs C1 Temp }

Expected Output: A human-readable and customizable visualization of correct results corresponding to the selected parameters from the input

Actual Output: Line graph visual displaying the data points from the data sheet with the axis custom labeled and graph title. A line of best fit is also displayed on the graph.

Result: **Pass**

4.3 Data Analysis

1. FR-ST3

Control: Manual

Initial State: The application's cleared user interface which has not yet been used to query data, with no graph showing yet.

Input: **SAMPLE PARAMETERS:** {Graph: Scatter, Experiment Dates: 2024-09-26, Parameter Type: Data Sheet, X Parameter: C1 pH, Y Parameter: C1 Temp, X Axis Label: C1 pH, Y Axis Label: C1 Temp}

Expected Output: ~~A small written human-readable paragraph explaining the input data.~~ A regression analysis that provides a visualization of the data's linearity. This was updated from the original VnV Plan after a discussion with the client.

Actual Output: A linear regression line is plotted with the resulting graph output, along with information about how well the regression line fits with the data.

Result: **Pass**. The actual output does not match the original expected output, however, the linear regression line provides a clear trend analysis and has been determined to be more useful than a text-based analysis, thus the expected output was updated to reflect these needs.

4.4 Data Hygiene

1. **FR-ST4**; Stretch Goal Requirement

Control: Manual

Initial State: The application's cleared user interface which has not yet been used to query data, with no graph showing yet.

Input: **SAMPLE_.CSV_FILE**

Expected Output: A log file documenting errors found in the input data and/or removals of missing data.

Actual Output: Feature not implemented; as this is a stretch goal, the functionality is currently not available.

Result: **Fail**

4.5 User Access

1. **FR-ST5**

Control: Manual

Initial State: User interface shows a login page, with no login credentials currently used

Input: `SAMPLE_VALID_EMAIL`, `SAMPLE_VALID_PASSWORD`

Expected Output: The page redirects to the page designated after login

Actual Output: Once credentials are submitted, the page redirects to the main home page of the application.

Result: **Pass**

4.6 Data Export

1. **FR-ST6**; Stretch Goal Requirement

Control: Manual

Initial State: User interface after multiple usages of data queries

Input: `USER ACTION => BUTTON CLICK`

Expected Output: Query report will be downloaded to the user's device

Actual Output: Feature not implemented; as this is a stretch goal, the functionality is currently not available.

Result: **Fail**

5 Nonfunctional Requirements Evaluation

This section outlines the manual system tests that were run and their subsequent results and outputs for all listed non-functional requirements.

5.1 Look and Feel

1. **NFR-LF1**

Type: User Demo, Manual

Initial State: Fully functional application ready for user interaction, starting at the login page

Input/Condition: User engagement with application

Expected Output: 90% of responses in the Navigation and Ease of Use section of the usability survey are at least 'Neither easy nor difficult'; 85% of users navigate the entire application within 10 minutes.

Actual Output:

The following answers are based on the Usability Survey provided in the [Verification and Validation Plan](#). The test was conducted on 2 users.

- Navigation and Ease of Use

(a) **User 1:** *Very Easy*, **User 2:** *Somewhat Easy*

(b) **User 1:** *Somewhat Easy*, **User 2:** *Somewhat Easy*

Calculated percentage of user responses that are at least 'Neither easy nor difficult' = 100%.

- Time to navigate the application end to end: **User 1:** 7.63 minutes, **User 2:** 8.5 minutes.

Calculated percentage of users that are able to navigate the application in under 10 minutes = 100%.

Result: **Pass**

2. NFR-LF2

Type: Manual

Initial State: Application running on a 15" laptop, 24" monitor, standard smart phone and/or tablet

Input/Condition: Manual tester's engagement with application

Expected Output: Number of inconsistencies found across all devices ≤ 10 .

Actual Output:

Dashboard: 2 inconsistencies on mobile, 1 inconsistency on tablet. Not full screen cuts off half way on both tablet and mobile. Table quick view is not mobile compatible as the table is squished and not readable.

Query Page: 4 inconsistencies on mobile, 1 inconsistency on tablet. The table is not horizontally scrollable on mobile view as such cannot

see the search bar and column drop down. Also cannot type in the equations field to apply them to the table. On tablet view there is a bit of the table that is cut off on the right.

Upload Page: No inconsistencies.

Graphs Page: 1 inconsistency on mobile. The modal is cut off on the right. Although it does not affect the functionality as all the buttons are still reachable, it is not visually appealing.

Login Page: 0 inconsistencies.

Result: **Pass**

Although there are still inconsistency, they are mainly due to the fact that mobile compatibility is a nice to have. It is expected that almost always users will be using the app on a standard sized laptop or monitor. Hence, why there are only inconsistencies present in tablet and mobile views and not the laptop or monitor views.

5.2 Usability

1. NFR-UH1

Type: User Demo, Manual

Initial State: Fully functional application ready for user interaction, starting at the login page

Input/Condition: User engagement with application, set of tasks to complete

Expected Output: 85% of users ask for help no more than 3 times; 90% of responses to the Learning section of the usability survey are at least 'Neither easy nor difficult'.

Actual Output:

The answers for the Usability Survey have been provided in test output for test 1 in section 5.1.

The users did not have to ask the usability test conductor for help at all. With that said, however, it must be noted that the users had been walked through a virtual demo of the application a couple of weeks prior to the usability testing.

Result: **Pass**

2. NFR-UH2

Type: Manual

Initial State: Fully developed application, starting at the login page

Input/Condition: Manual tester using application

Expected Output: 0 language discrepancies found in the application.

Actual Output: 0 language discrepancies found in the application.

Result: **Pass**

3. NFR-UH3

Type: Manual

Initial State: Application ready to take in data

Input/Condition: **SAMPLE..CSV_FILE**

Expected Output: Error logs from unrecognized characters, or successful upload

Actual Output: The upload csv will still upload the unrecognized symbols without any issues. The type will just be a string. As a result, there is nothing to log.

Result: **Pass**

4. NFR-UH4

Type: Manual

Initial State: Application ready for use

Input/Condition: Tester engagement using third party tools

Expected Output: List of accessibility issues in accordance to Web Content Accessibility Guidelines (WCAG), and checklists for if page is screen-readable

Actual Output:

Dashboard: All buttons and form fields are do not have alt text or context.

Query Page: All buttons and form fields are do not have alt text or

context. Scrolling is also not possible from keyboard shortcuts.

Upload Page: Upload button does not have alt text.

Graphs Page: All buttons and form fields are do not have alt text or context.

Login Page: Is accessible

Result: **Fail**

Most pages do not comply with the WCAG guidelines. Many of the components and buttons are missing alt text which makes the page screen readable.

5.3 Performance

1. NFR-P1

Type: Manual

Initial State: Application navigated to upload page, ready to upload file

Input/Condition: `SAMPLE..CSV_FILE`

Expected Output: Average upload duration \leq `UPLOAD_TIME`

`UPLOAD_TIME` = 60 seconds

Actual Output: File uploaded in 48 seconds on average over 20 sample uploads, satisfying the performance requirements.

Result: **Pass**

2. NFR-P2

Type: Manual

Initial State: Application navigated to querying page, ready to make query

Input/Condition: `SAMPLE_QUERY: {}`

Expected Output: Average query response duration \leq `QUERY_TIME` where `QUERY_TIME` = 3 seconds

Actual Output: Query response time averaged less than 0.1, satisfying the performance requirement.

Result: **Pass**

3. NFR-P3

Type: Manual

Initial State: Application ready to use

Input/Condition: Tester engagement

Expected Output: Average response time of buttons on website \leq RESPONSE_TIME

RESPONSE_TIME = 2 seconds

Actual Output:

Dashboard: Dropdowns have a response time of 0.1 seconds

Query Page: All buttons and form fields are do not have alt texts or context. Scrolling is also not possible from keyboard shortcuts.

Upload Page: Upload button takes about 2 seconds if there is an error and on average 1.75 seconds for a successful upload. It also takes 0.1 seconds to being uploading.

Graphs Page: Graph, parameter type drop downs have a response time of 0.1. Date, x and y parameter drop downs has an average response time of 1.25 seconds

Login Page: Login and register button have a response time of 0.1 seconds

Result: **Pass**

4. NFR-P4

Type: Manual

Initial State: Application navigated to graphs page, ready to make query to generate graphs

Input/Condition: SAMPLE_QUERY: {}

Expected Output: Average time taken to generate graphs/visualizations \leq GRAPH_TIME

GRAPH_TIME: 10 seconds

Actual Output:

- Test Execution 1: 8.2 seconds
- Test Execution 2: 9.1 seconds
- Test Execution 3: 7.9 seconds
- Test Execution 4: 8.7 seconds
- Test Execution 5: 9.5 seconds

Average Time Taken = 8.7 seconds

Result: **Pass**

5. **NFR-P5**

Type: Manual

Initial State: Application navigated to query page, ready to make query

Input/Condition: `SAMPLE_QUERY : {NaCl ≥ 1.0}`

Expected Output: A check for precision of numbers in different components of system. Values should be accurate to 4 decimal places.

Actual Output: Parameter values displayed with an accuracy matching that of the original spreadsheet data. All parameters exceeded the accuracy of 4 decimal places, satisfying the requirement.

Result: **Pass**

6. **NFR-P6**

Type: Manual

Initial State: Application running as normal

Input/Condition: Tester temporarily taking down back end

Expected Output: Error messages properly displayed

Actual Output: Displayed appropriate error messages for all components indicating that data could not be retrieved from backend services.

Result: **Pass**

7. **NFR-P7**

Type: Manual

Initial State: Application running as normal

Input/Condition: Tester temporarily disconnects internet connection

Expected Output: Previously generated plots and previous queries still working

Actual Output:

- Previously generated plots remain visible and interactive.
- Queries executed before disconnection are still accessible and functional.
- No errors or crashes observed during the disconnection period.

Result: **Pass**

8. **NFR-P8**

Type: Manual

Initial State: Three devices ready to run application

Input/Condition: `SAMPLE_QUERY: {}`

Expected Output: System response time while under load \leq `RESPONSE_TIME_THRESHOLD`

Actual Output:

`RESPONSE_TIME_THRESHOLD = 3` seconds

- Device 1 Response Time: 1.8 seconds
- Device 2 Response Time: 1.9 seconds
- Device 1 Response Time: 1.7 seconds

Average Response Time Under Load = 1.8 seconds

Result: **Pass**

9. **NFR-P9**

Control: Data Generation, Automated

Initial State: A database with a known amount of experiment data

Input/Condition: `DUMMY_DATA`

Expected Output: Observations on system health after large payload

Actual Output: Large payload are processed with slight performance degradation, but still satisfying performance requirements.

Result: **Pass**

5.4 Operational and Environmental

1. NFR-OE1

Type: Manual

Initial State: Application running on a windows device as a web application on a Chromium based browser.

Input/Condition: Tester engagement

Expected Output: A list of all discovered issues with the application that arise due to environment compatibility

Actual Output: No issues discovered when using Google Chrome. The experience is same as the expected.

Result: **Pass**

2. NFR-OE2

Type: Manual, User Demo

Initial State: Application running and ready for use on home screen

Input/Condition: User engagement

Expected Output: Survey results depicting subjective complexity of onboarding process

Actual Output:

Based on the usability testing results provided in section 5.1 for test 1, as a 100% of the users found the navigation process to be at least ‘somewhat easy’, it can be concluded that the onboarding process is intuitive and user-friendly.

Result: **Pass**

5.5 Maintainability and Support

1. NFR-MS1

Type: Manual, User Demo

Initial State: Application running, ready for use

Input/Condition: User engagement

Expected Output: Observations on user's ability to complete tasks without support

Actual Output:

Observations from Usability Testing

(a) User 1:

- Successfully completed all assigned tasks such as navigating to the graphs page, generating a graph etc. without any assistance.
- Did not ask for help or clarification during the session.
- Completed the tasks in 7.63 minutes.

(b) User 2:

- Successfully completed all assigned tasks such as navigating to the graphs page, generating a graph etc. without any assistance.
- Did not ask for help or clarification during the session.
- Completed the tasks in 8.5 minutes.

Percentage of users who completed tasks without support = 100%.
Average Time to complete tasks = 8.07 minutes. A complete list of the tasks each user was asked to perform can be found in the [Sample Usability Testing Survey](#).

Result: **Pass**

2. NFR-MS2

Type: Manual

Initial State: Multiple Chromium-based web browsers open

Input/Condition: Tester Engagement

Expected Output: All abnormal behaviour of web pages observed on each different web browser

Actual Output: The application ran on Firefox, Chrome and Microsoft Edge. There was no abnormal behaviour across the browsers running the app.

Result: **Pass**

5.6 Security

1. NFR-SR1

Type: Manual

Initial State: Application login page is displayed.

Input/Condition: **SAMPLE_VALID_CREDENTIALS**, **SAMPLE_INVALID_CREDENTIALS**

Expected Output: Access is granted or denied based on the validity of the credentials provided.

Actual Output: Access is granted for the **SAMPLE_VALID_CREDENTIALS**, which consisted of a registered email and correct password, and consistently denied for the **SAMPLE_INVALID_CREDENTIALS**, which consisted of the following test inputs:

- Registered email, incorrect password
- Unregistered email, correct password
- Unregistered email, incorrect password

Result: **Pass**

2. NFR-SR2

Type: Manual

Initial State: Application logged in with multiple user roles (e.g., admin, researcher, research assistant).

Input/Condition: Tester interacts with the application using on different user roles.

Expected Output: Access to query or modify data and perform sensitive operations is restricted according to user roles.

Actual Output: Only admin and researchers were able to modify data and these operations were not available to users with the role of research assistant.

Result: **Pass**

3. **NFR-SR3**

Type: Manual

Initial State: User is logged into the application.

Input/Condition: User remains inactive for *TIMEOUT* minutes, where *TIMEOUT* = 15 minutes.

Expected Output: User is automatically logged out after the predefined period of inactivity.

Actual Output: The user was not logged out after 15 minutes of inactivity.

Result: **Fail**

4. **NFR-SR4**

Type: Manual

Initial State: Application is open, ready for data entry and CSV upload.

Input/Condition: `VALID_DATA`, `INVALID_DATA`, `CORRUPTED_DATA`

Expected Output: All invalid inputs and CSV uploads are rejected, and only valid data entries are processed.

Actual Output: Valid data entries were processed correctly. All invalid inputs were rejected with a visual cue, and/or error message.

Result: **Pass**

5. **NFR-SR5**

Type: Manual

Initial State: Application database contains a set of unique, validated records. Application interface is open for data entry, processing, and transfer actions.

Input/Condition: `SAMPLE_DUPLICATE_RECORD`

Expected Output: Duplicate records are detected and prevented, and data accuracy is maintained during all transfer operations.

Actual Output: Duplicate records are detected and filtered out through indexing, thus data accuracy maintained throughout the transfer process.

Result: **Pass**

6. **NFR-SR6**; Out of Scope

Type: Manual/Automated

Initial State: Database is operational, with storage capacity at or below normal usage. Alert system is configured, and the administrator contact information is set up to receive notifications.

Input/Condition: Tester simulates increasing database storage usage to exceed the `STORAGE_THRESHOLD`.

Expected Output: System successfully detects when storage usage exceeds `STORAGE_THRESHOLD` and sends a timely alert to administrators.

Actual Output: Feature not implemented; notification system is considered a stretch goal.

Result: **Fail**

7. **NFR-SR7**; Out of Scope

Type: Manual/Automated

Initial State: The application is operational, with logging features configured and permissions for accessing logs assigned to administrators only.

Input/Condition: Tester performs various access and modification actions within the application, then attempts to access the audit logs

with both authorized and unauthorized user accounts.

Expected Output: All actions are logged with timestamps and user identities, and audit logs are accessible only to authorized users, with proper encryption.

Actual Output: Feature not implemented; audit system is considered a stretch goal.

Result: **Fail**

8. **NFR-SR8**; Out of Scope

Type: Manual

Initial State: Application login screen is open. Administrator contact information is configured to receive security alerts.

Input/Condition: Tester attempts multiple failed logins to trigger the suspicious activity detection mechanism.

Expected Output: Application detects and blocks access after three failed attempts, sends an alert to administrators, and locks out the user temporarily.

Actual Output: Application detects failed login attempts but does not send alerts. Notification system is considered a stretch goal.

Result: **Fail**

9. **NFR-SR9**; Out of Scope

Type: Automated/Manual

Initial State: Application is running under typical workload conditions. Monitoring tools and alerts for CPU and memory usage are enabled.

Input/Condition: Tester simulates a high workload to approach system resource limits, observing the system's monitoring and optimization response.

Expected Output: The system actively manages CPU and memory usage, preventing overload and maintaining performance stability.

Actual Output: Not conducted due to resource constraints and complexity in simulating high workload conditions in the current testing

environment.

Result: **Fail**

6 Unit Testing

The following subsections outline the unit tests relevant to each module. Each subsection includes a table or table(s) with a summary of the results for each test, where each test includes a link to its corresponding test file in the project repository.

6.1 Data Storage Module

Table 1 provides a summary of the results for each test that is relevant to the Data Storage Module.

Test ID	Test Name	Result
6.1-UT1	Uploads experiments and data	Pass
6.1-UT2	Links data to experiments	Pass
6.1-UT3	Uploads data when experiment exists	Pass
6.1-UT4	Skips data upload if no experiment exists	Pass
6.1-UT5	Cleans data by removing empty values	Pass
6.1-UT6	Checks for duplicate experiments	Pass
6.1-UT7	Handles multiple matching experiments	Pass
6.1-UT8	Handles no matching experiments	Pass
6.1-UT9	Runs full migration process	Pass
6.1-UT10	Handles experiment upload errors	Pass
6.1-UT11	Handles data upload errors	Pass
6.1-UT12	Shows upload status indicators	Pass
6.1-UT13	Handles drag over and drop event with valid file	Pass
6.1-UT14	Handles file input change event with valid file	Pass
6.1-UT15	Handles cell editing	Pass

Table 1: Unit tests summary for Data Storage Module

6.2 Data Retrieval Module

Tables 2, 3, and 4 provide a summary of the results for each test that is relevant to the Data Retrieval module.

Test ID	Test Name	Result
6.2.1-UT1	Initialize with default values	Pass
6.2.1-UT2	Fetch experiment IDs	Pass
6.2.1-UT3	Fetch experiments	Pass
6.2.1-UT4	Fetch data	Pass
6.2.1-UT5	Handle Apollo error states	Pass
6.2.1-UT6	Refetch experiments and data	Pass
6.2.1-UT7	Call RefetchExperiments function	Pass
6.2.1-UT8	Call RefetchData function	Pass
6.2.1-UT9	Filter data by search keyword	Pass

Table 2: Unit tests summary for Data Retrieval Module

Test ID	Test Name	Result
6.2.2-UT1	Initialize with default step	Pass
6.2.2-UT2	Navigate through all steps correctly	Pass
6.2.2-UT3	Validates required fields at each step	Pass
6.2.2-UT4	Shows error for invalid parameter selection	Pass
6.2.2-UT5	Validates axis ranges and shows errors	Pass
6.2.2-UT6	Handles form submission successfully	Pass

Table 3: Unit tests summary for Data Retrieval Form Module

Test ID	Test Name	Result
6.2.3-UT1	should fetch all graphs when latest=0	Pass
6.2.3-UT2	should fetch latest graph when latest=1	Pass
6.2.3-UT3	should handle network errors	Pass

Table 4: Unit tests summary for Data Retrieval useGraph Hook

6.3 Visualization Module

Table 5 provides a summary of the results for each test that is relevant to the Data Visualization module.

Test ID	Test Name	Result
6.3-UT1	Renders without crashing	Pass
6.3-UT2	Renders GraphSideBar component	Pass
6.3-UT3	Renders correct graph type	Skipped
6.3-UT4	Renders with correct graph title	Skipped
6.3-UT5	Renders with correct axis labels	Skipped

Table 5: Unit tests summary for Data Visualization Module

Tests 6.3-UT3, 6.3-UT4, and 6.3-UT5 were skipped due to ongoing debugging and recalibration of the module. Although they are included in the test suite for full test coverage, the team is already aware of the existing issues and is actively addressing them. Running these tests only to deliberately fail them is not meaningful at this stage. The team will rerun the skipped tests once the debugging process is complete and will update their results at that time.

6.4 User Management Module

Table 6 provides a summary of the results for each test that is relevant to the User Management module.

Test ID	Test Description	Result
6.4-UT1	Creates a new user	Pass
6.4-UT2	Skips user creation if already registered	Pass
6.4-UT3	Validates login attempt with correct password	Pass
6.4-UT4	Invalidates login attempt with incorrect password	Pass
6.4-UT5	Retrieves current user's email and role	Pass
6.4-UT6	Creates session information for current user	Pass
6.4-UT7	Removes session information for current user	Pass
6.4-UT8	Updates current user's password	Pass
6.4-UT9	Updates current user's role	Pass
6.4-UT10	Removes existing user	Pass
6.4-UT11	Handles removal of nonexisting user	Pass
6.4-UT13	Handles logout and user redirected to login page	Pass
6.4-UT14	Handles page access when not logged in	Pass
6.4-UT15	Checks for session creation when logged in	Pass
6.4-UT16	Checks for session removal when logged out	Pass

Table 6: Unit tests summary for User Management Module

7 Changes Due to Testing

This section outlines all of the changes that should be made to the implementation based on the results of the testing. This includes changes from the supervisor's feedback and from failed tests.

7.1 Changes Due to Supervisor Feedback

Our supervisor suggested the current data visualization form intake could be improved by allowing the user to filter the experiment data by additional parameters, rather than only by the date it was performed. This feature will be added to the final product.

They have also requested to add a new feature that was not originally part of the requirements. This feature allows the user to compute a variety of effi-

ciency factors related to the data, measuring the efficiencies of the membrane used in the experiment. This feature will be added to the final product, but thorough validation testing will be limited due to time constraints.

7.2 Changes Due to Failed Tests

The following sections outline changes that will be made to the implementation following a failed test or tests. This does not include tests for stretch goal requirements that failed.

7.2.1 [NFR-S3](#)

The system did not automatically log out the user after a period of inactivity. This was due to the lack of a session timeout manager, which will be implemented in the final product.

7.2.2 [NFR-UH4](#)

The system was not compliant to with the WCAG standards. This is due to the fact that the current research team using the project does not need any accessibility considerations when using the app. However this may not be true in the future, as such, updates to the current pages will be implemented using the feedback from SiteImprove in the final product.

8 Automated Testing

Pytest was used as the automated testing framework for Python. Unit test suites were executed using Pytest, which supports fixtures, parameterization, and code coverage metrics, making automated testing more efficient and flexible. Jest was used as the automated testing framework for React components.

9 Trace to Requirements

Tables [7](#), [8](#), [9](#), and [10](#) provide a traceability matrix for the functional and nonfunctional requirements to the test cases.

Test ID	FR-ST1	FR-ST1.1	FR-ST2	FR-ST3	FR-ST4	FR-ST5	FR-ST6
FR-1	X	X					
FR-2	X	X					
FR-3	X	X					
FR-4	X	X					
FR-5			X				
FR-6			X				
FR-7			X				
FR-8			X				
FR-9			X				
FR-10				X			
FR-11				X			
FR-12					X		
FR-13					X		
FR-14						X	
FR-15							X

Table 7: Traceability Matrix for Test Cases and Functional Requirements

Test ID {NFR-}	LF1	LF2	UH1	UH2	UH3	UH4	OE1	OE2	MSR1	MSR2
LFR-1	X									
LFR-2		X								
LFR-3	X									
LFR-4	X									
LFR-5	X									
LFR-6	X									
LFR-7	X									
UHR-1			X							
UHR-2				X						
UHR-3					X					
UHR-4			X							
UHR-5			X							
UHR-6						X				
OER-2							X			
OER-3							X			
OER-4							X			
OER-5								X		
MSR-4									X	
MSR-6										X

Table 8: Traceability Matrix for Test Cases and Nonfunctional Requirements
(Part 1)

Test ID {NFR-}	P1	P2	P3	P4	P5	P6	P7	P8	P9
PR-1	X								
PR-2		X							
PR-3			X						
PR-4		X							
PR-5				X					
PR-6					X				
PR-7					X				
PR-8					X				
PR-9						X			
PR-10							X		
PR-11								X	
PR-12									X

Table 9: Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 2)

Test ID {NFR-}	SR1	SR2	SR3	SR4	SR5	SR6	SR7	SR8	SR9
SR-1	X								
SR-2		X							
SR-3		X							
SR-4			X						
SR-5				X					
SR-6					X				
SR-7					X				
SR-8					X				
SR-9				X					
SR-12						X			
SR-13							X		
SR-14							X		
SR-15							X		
SR-16								X	
SR-17									X

Table 10: Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 3)

10 Trace to Modules

Table 11 provides a traceability matrix for the system test cases to the modules.

Test ID	FR-ST1	FR-ST1.1	FR-ST2	FR-ST3	FR-ST4	FR-ST5	FR-ST6
M6	X	X			X		
M7			X	X			
M11			X				
M12			X				
M13						X	
M15	X						
M16					X		
M17							X

Table 11: Traceability Matrix for System Test Cases and Modules

11 Code Coverage Metrics

The team has determined code coverage metrics were not indicative of valuable information in relation to the validation and verification of the implementation, thus this section is omitted.

Appendix — Reflection

1. What went well while writing this deliverable?

Writing the general structure of the report was straightforward as we could refer back to the tests we had written in the VnV Plan. Having detailed system tests planned out made testing the system a lot easier given that we had a well established jumping ground.

2. What pain points did you experience during this deliverable, and how did you resolve them?

Constructing and executing all of our unit tests was the biggest challenge primarily due to time constraints. To combat this, the team lead held regular check-ins to ensure that everyone's individual timelines were aligned.

We also ran into the issue of interpreting ambiguous user feedback during the usability testing. This was resolved by consulting back to our course 4HC3 - Human Computer Interaction and changing the style of the session so that instead of open-ended feedback, the user is asked to choose from pre-determined options with the choice to provide additional feedback later on.

3. Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?

Most of the non-functional requirements stemmed from speaking to our clients who acted as the users for the usability test. They were consulted when conducting system tests and ensuring that the outlined tests were in alignment with the application requirements.

The parts of the document related to automated testing and unit testing such as testing the hooks and APIs and more did not stem from speaking to the clients or peers. This is because most of the tests were to ensure that the system compiles and functions correctly from a technical perspective rather than focusing on user experience or usability. These tests were derived from the development team's understanding of the system architecture, coding standards, and best practices for ensuring reliability, performance and maintainability.

4. 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.)

The primary way in which the VnV Plan is different from the activities that were actually conducted for the VnV Report is implementation of features. After Revision 0, the scope of the project was changed after consulting the professor and given the change in the priority of requirements from the client. This resulted in a number of features now being labelled as stretch goals which were, at the time of testing, not implemented. Thus, the tests for these features had to be labelled as 'Out of Scope'.

Given more experience in the iterative nature of a software project of this scale, we would like to believe that we can manage the project expectations better and define a more realistic implementation timeline at the get go, in the future.

Additionally, for a few features, the scope of implementation had to be reduced. One example is the user management system and authorization protocols. These were not implemented in as robust a fashion as earlier expected due to the limited resources at the team's disposal. For instance, it was mutually decided that it would be better to allocate the time and effort required to make the protocol more robust to instead implement additional analysis features.