System Verification and Validation Plan for Software Engineering

Team #22, TeleHealth Insights
Mitchell Weingust
Parisha Nizam
Promish Kandel
Jasmine Sun-Hu

March 23, 2025

Revision History

Table 1: Revision History

Date	Vers.	Contributors	Notes
10/30/24	1.0	Mitchell Weingust	• Added: 3.1 Verification and Validation Team • Added: 3.4 Verification and Validation Plan • Added: 6.2 Usability Survey Questions
10/30/24	1.1	Promish Kandel	• Added: 3.6 Automated Testing and Verification Tools • Added: 3.7 Software Validation Plan • Added: 4.1 Data Collection and Storage • Added: Video and Audio Data Analysis FR tests • Added: 4.2 Performance NFR tests
10/30/24	1.2	Parisha Nizam	• Added: 2.1 Summary • Added: 2.2 Objectives • Added: 2.3 Challenge Level and Extras • Added: 2.4 Relevant Documentation
11/01/24	1.3	Jasmine Sun-Hu	• Added: 3.2 SRS Validation Plan • Added: 4.1 Data Processing and Display • Added: 4.2 Look and Feel FR tests • Added: 4.2 Maintenance and Support NFR tests • Added: 4.2 Cultural NFR tests
11/02/24	1.4	Mitchell Weingust	• Added: 4.1 Authentication FR tests • Added: 4.2 Usability and Humanity NFR tests • Added: 4.2 Operational and Environmental NFR tests
11/02/24	1.5	Parisha Nizam	• Added: 4.1 System Set Up & Assessment Interface FR tests • Added: 4.2 Security and Compliance NFR Tests
11/04/24	1.6	Mitchell Weingust	• Added: Reflection

Date	Vers.	Contributors	Notes
11/04/24	1.7	Jasmine Sun-Hu	• Added: 6.1 Symbolic Parameters • Added: Reflection
11/04/24	1.8	Promish Kandel	• Added: 1 Symbols, Abbreviations and Acronyms • Added: Reflection
11/04/24	1.9	Parisha Nizam	• Added: Reflection
03/22/25	2.1	Jasmine Sun-Hu	• Implemented TA Feedback: Adjusted formatting and added subsection intro text • Added HIPAA adherence
03/23/25	2.2	Jasmine Sun-Hu	• Implemented Peer Feedback: Added test case derivations to NFRs • Clarified objectives • Adjusted FR-ST-DSC5 test case method • Clarified test case details • Added role descriptions to team table • Updated test case types • Added test case names and sentence descriptions

Contents

List of Tables

1 Symbols, Abbreviations, and Acronyms

symbol	description
SRS	Software Requirements Specification
VnV	Verification and Validation
PII	Personally Identifiable Information

This document contains the team's verification and validation plan for the TeleHealth Insights project. This document features general information, overall plan, system tests, the eventual addition of unit tests, and usability survey questions.

2 General Information

2.1 Summary

The software being tested TeleHealth Insights, is a web-based at-home bilingual speech assessment system with video and audio analysis features. The system is designed to provide clear guidance to parents when administering the assessment to their children, in an environment where speech-language pathologists (SLPs) are unavailable. By streamlining the assessment pro- cess, the project aims to provide a convenient and comprehensive solution for SLPs to assess and support their patient's speech and language development remotely.

TeleHealth System comprises of several software components to efficiently provide a platform for children assessments to occur while actively recording for clinicians to review and analyze.

2.1.1 User Interface

We must ensure that the interface is easy to navigate for all stakeholders involved, the parents, clinicians as well as the parents. The system should be simple and testing will include ensuring there is an intuitive layouts, clear instructions and labels, as well as a logical workflow.

2.1.2 Video and Audio Recording

Ensure set up of hardware devices used on application is intuitive and done properly.

2.1.3 Video and Audio Analysis

Integrate web-based application to store recorded audio and video footage. The system will be tested to ensure it is able to appropriately analyze and identify flags and bias when conducting assessments. The analyzed data should be presented in a easy to understand representation

2.2 Objectives

The VnV Plan aims to evaluate whether the system meets key project goals in terms of recording quality, AI performance, data security, and ease of use. The following subsections clarify the scope and definitions of the main objectives within the context of this project.

2.2.1 Ensure Accuracy and Correctness of Software Reading and Identifying Bias

The primary objective of this VnV Plan is to ensure that the system can:

- Correctly record video and audio: The system should capture audio and video that meet minimum quality thresholds, such as resolution, clarity, and synchronization between video and audio streams, as defined in performance requirements (Section 4.2.3).
- Accurately detect disturbances or bias: The AI analysis models should flag disturbances (e.g., background noise, occluded faces, off-task behavior) and potential assessment biases (e.g., speech delays, distracted environment) with a predefined success rate (e.g., `VERY_HIGH_SUCCESS_RATE`) in comparison with human-reviewed ground truth.
- Provide valid outputs: Results of the analysis should be correctly summarized in clinician-facing dashboards or reports, with visualizations that align with recorded content.

2.2.2 Security and Authentication

The second objective is to verify that the system enforces strong authentication protocols and protects sensitive data throughout its lifecycle. Specifically:

- Data privacy and encryption: All assessment recordings and user data must be securely stored and encrypted both at rest and in transit, meeting compliance requirements (e.g., HIPAA, as referenced in Section 4.2.8).
- Role-based access control: Only authorized users should be able to perform role-specific tasks:
  - Parents can complete assessments with their children.
  - Clinicians can view assessment results and recordings.
  - Admins can manage account creation and system configuration.
- Access restrictions: Attempts by unauthorized users to access protected interfaces or data must be detected and blocked.

2.2.3 Demonstrate Usability

The final objective is to confirm that the system is intuitive and accessible for parents, children, and clinicians. In this context:

- User-friendly interface: The UI should follow best practices in accessibility, clarity, and layout. Users should be able to perform common tasks (e.g., starting an assessment, reviewing results) with minimal instructions and no more than a defined number of clicks (`MAX_CLICKS`).
- Efficient task completion: Success is measured by user testing metrics (e.g., task completion rate, time to complete), and feedback collected via the usability survey in the Appendix (Section 6.2).
- Child accessibility: The assessment interface must be suitable for children aged 6–12, who should be able to navigate the assessment independently (Section 4.2.2).

Out of Scope

Some objectives that are considered out of scope for this VnV plan due to resource limitations include:

- Adherence to UI Industry Standards: Full compliance with all published UI standards is not feasible due to time constraints. The focus is on practical usability and functional aesthetics rather than exhaustive WCAG or HIG compliance.
- Verification of Third-Party Libraries: External open-source libraries used in the frontend or AI model will not be independently audited. Instead, system-level testing will ensure that their integration functions as expected.

2.3 Challenge Level and Extras

2.3.1 Challenge Level

Our challenge level is general as the project scope is limited in terms of how much research is required. The required domain knowledge is basic web-design in a stack of our choice. We are also planning on using open-source large language models for audio and video processing.

2.3.2 Extras

Our project has the following extras;

- **User Documentation:** Providing users with a guide on how to use and better understand the system.
- **Usability Testing:** Receive user feedback on usability of design of the application, including improvements on how the system looks and functions. This will help to ensure an intuitive and easy system for both clinicians and parents/children to navigate through.

2.4 Relevant Documentation

The three main relevant documentation that helped guide us in System Verification and Validation Plan (VnV) was Software Requirements Specification (SRS), Development plan and MIS.

- The SRS document was crucial to define descriptions, rationals and fit criteria for all of the systems main functional and non functional requirements. This outlines the basis of what needs to be verified and tested in the VnV Plan. It also outlines the traceability matrix that needs to be followed.
- Development Plan: Defined the main Goals, Objectives and extras for the system, aiding the team to understand what needs to be done and focused on.
- MIS : The Module Interface Specification (MIS) will shape the scope of unit testing in the VnV plan and guides the development of both functional and nonfunctional tests to verify specific software functions and attributes.

3 Plan

This section includes plans for how the team's verification and validation processes will be implemented. This covers the team and its roles, the SRS verification plan, the design verification plan, the verification and validation plan verification plan, the implementation verification plan, automated testing and verification tools, and the software validation plan.

3.1 Verification and Validation Team

The following table displays the roles and responsibilities of each team member.

Name	Roles and Responsibilities
Mitchell Weingust	• Audio Analysis Model verification • System Architecture and Design Validation • SRS Verification
Parisha Nizam	• Frontend Interface Verification • Backend Database Verification • VnV Verification
Promish Kandel	• Frontend Interface Verification • Video Analysis Model verification • VnV Verification
Jasmine Sun-Hu	• Backend Database Verification • System Architecture and Design Validation • SRS Verification
Dr. Irene Yuan	• Providing feedback (including Hands-On) during project development
Dr. Yao Du	• Providing written feedback on user experiences and testing
Chris Schankula	• Providing feedback during project development • Revision recommendations

Table 2: Verification and Validation Team Table

Role Descriptions:

- **Audio Analysis Model Verification:** Evaluate the model's ability to detect disturbances, flag bias, and process audio data accurately.
- **Video Analysis Model Verification:** Validate that the model correctly identifies gestures and mouth movements in accordance with assessment needs.

- **Frontend Verification:** Review and test UI components for usability, responsiveness, design consistency, and accessibility across devices.
- **Backend Database Verification:** Confirm secure storage, proper schema design, access control, encryption, and compliance with data protection standards.
- **System Architecture and Design Validation:** Validate overall design structure, integration between modules, and consistency with the system requirements specification (SRS).
- **SRS Verification:** Ensure all functional and nonfunctional requirements are clear, testable, and traceable in the VnV plan.
- **VnV Plan Development and Review:** Draft, organize, and review test cases, structure testing phases, and maintain traceability between test results and requirements.
- **Usability and UX Feedback:** Provide evaluations of system usability, especially regarding child-friendly design and clinician interaction.
- **Project Feedback and Documentation Review:** Offer structured input on development direction and suggestions for improving documentation clarity and completeness.

3.2 SRS Verification Plan

The following approaches will be used to verify the SRS:

- **Ad hoc Peer Review**: Informal reviews of the SRS will be conducted after every major revision to the SRS with classmates who will serve as peer reviewers. This will provide new perspectives that can help identify ambiguities, missing requirements and/or areas of improvement.
  - Peer reviewers will submit feedback using GitHub issue tracker for organized task assignment and tracking.
- **Supervisor Review**: Before every major SRS revision submission, the team will send a copy of the SRS to the project supervisor a week before meeting, along with a checklist highlighing priority areas for their feedback. The team will also prepare questions about the requirements related to interfaces and usability which is the supervisor's area of expertise. During the review meeting, the team will first review the supervisor's initial feedback with the supervisor, and then ask the prepared questions.
  - The meeting notes will be documented using GitHub issue tracker.
- **Internal Team Walkthrough**: The team will hold a collaborative session before every major SRS revision submission where the team will discuss each section one by one to verify that all requirements are understood by all members, and that each section is consistent with the project goals/objectives. A checklist will act as a guide to highlight the key concepts the review should focus on, and will help ensure no critical areas are missed in review. The checklist can be found below.

  - Any corrections or modifications that need to be made will be noted in the team meeting's GitHub issue tracker.

The following checklist will be used for the internal team walkthrough:

- ☐ Are all major functions required for the website (interface, backend, recording, analysis, storage) covered?
- ☐ Does each function have clear input and output specifications?
- ☐ Are all requirements written with consistent terminology?
- ☐ Do all requirements avoid conflict with each other?
- ☐ Does each requirement avoid ambiguous language?
- ☐ Are all requirements verifiable and testable?
- ☐ Is each requirement written in a way all team members and stakeholders can understand?
- ☐ Are any assumptions about user behavior or system behaviour explicitly stated?

3.3 Design Verification Plan

The design verification plan outlines the strategies that the team will use to verify the usability and correctness of our system. The following outlines the plan for verifying the User Interface of the software as captured by system design

- Shall conduct a quick review with supervisor after design documents (MIS) have been completed
- Conduct peer review sessions from classmates to provide critical suggestions on improvements of the system.
- Conduct final formal review with clinician Dr. Du and the team, following defined SRS and MIS documents.
- Conduct ad-hoc review(s) with other teams.

The following checklist will be used to verify the system's design documents over the course of completion.

- ☐ Are all requirements traceable to at least one feature/module in the MIS?
- ☐ Do all modules and components follow SOLID design principles?
- ☐ Have the creation of all modules been tracked via issues and closed once reviewed?
- ☐ Are all inputs and outputs of system components defined well and correctly?

3.4 Verification and Validation Plan Verification Plan

As the verification and validation plan is an artifact, it must be verified too. The team's verification of the VnV plan follows:

- Peer reviews by classmates, including other teams' peer reviews, to identify areas of improvement and general feedback
- Documentation review by the project's supervisor, Dr. Irene Ye Yuan, to ensure that the team's planned verification and validation plan is realistic and feasible
- Teammate documentation reviews, to provide critical feedback and ensure that all intended goals and outcomes are met
- Mutation testing to ensure that changes to aspects of the source plan can be detected by test cases.

The below checklist will be used, in addition, to ensure the team's VnV plan is correct and complete.

- ☐ Does the VnV Plan verify all functional requirements are met?
- ☐ Does the VnV Plan verify all non-functional requirements are met?
- ☐ Have all peer-review issues been addressed and closed?
- ☐ Have all members of the Verification and Validation Team contributed to the review and approved the document?
- ☐ Are all aspects of the system boundary being verified, validated, and tested?
- ☐ Do the system tests cover all requirements mentioned in the SRS?
- ☐ Did the test cases detect mutations and give desired outputs?
- ☐ Did the test cases' expected output match the actual output?
- ☐ Is there a process for documenting and resolving defects?

3.5 Implementation Verification Plan

The following outlines the plan for verifying implementation of the system, using both static and dynamic techniques

- Walkthrough of key components of the system with the supervisor (UI design, user authentication, video and audio analysis)
- Walkthrough of each components with other teammates
- Running unit tests (to be implemented in VnV Plan) to verify that the implementation matches the specified design outlined in MIS. This will be done automatically (dynamic) using Github Actions for each pull request made, following CI/CD principles.

- Running system tests (both functional and non functional) described in the VnV plan to verify that the implementation meets requirements
- Running Static analyzers including linters to help discover any potential errors or bugs in the code. Linters and tools will also be used to help make the code more organized and readable.
- Major code commits will be reviewed by at least one other team member before merging to main branch to ensure consistency and minimize chance of conflicts.

3.6 Automated Testing and Verification Tools

The following are the automated testing and verification tools to be used during the validation and verification process for the software being tested in this VnV plan:

- Unit Tests: Jest, Pytest
- Linters: Flake9, Prettier, ESLint
- Continuous Integration: GitHub Actions

For our code coverage, we will use Istanbul and Coverage.py. Istanbul is a code coverage tool that works with JavaScript testing frameworks like Jest. It helps developers see how much of their code is tested by creating reports that show untested lines and functions. Coverage.py is a code coverage tool for Python, which we use for our machine learning model. It measures how much of the code runs during tests and generates reports in different formats, helping developers find untested parts of their Python applications and improve test coverage.

3.7 Software Validation Plan

Our plan for validating the software includes review sessions with stakeholders and extensive user testing. Dr.Yao Du, one of our key stakeholders, will provide the software to clinicians and patients for real-world testing in clinical settings. This will allow us to gather valuable feedback on the software's functionality and usability in actual healthcare environments. In addition to this field testing, we will conduct structured user testing sessions where participants will simulate the experience of being a patient. During these sessions, users will navigate through the software, interacting with its features, and afterward, they will share their insights on what worked well and what didn't. Overall, using both field testing and targeted user testing, the information gathered will help us refine the software and ensure it meets the needs of its intended users.

4 System Tests

This section includes systems tests for functional requirements and nonfunctional requirements mentioned in the SRS. For traceability, the team created a traceability matrix to correlate the system tests to the requirement(s) they cover.

4.0.1 Authentication

The test cases below focus on ensuring users can safely and securely login, create and access their accounts without worrying about others accessing their information.

- FR-ST-A1: **Select Parent Role for Login**

  Purpose: Verifies that selecting the Parent role from the login screen correctly redirects the user to the Parent login page.

  Type: Manual, Dynamic

  Initial State: User has a Parent account already created and stored in the database

  Input: Selection of Parent account role for login

  Output: The expected result is the Parent account role is selected and User is brought to the Parent login screen

  Test Case Derivation: The expected output is justified based on FR-A1 in the SRS document

  How the test will be performed:

  1. Select 'Login' to go to login screen
  2. Select 'Parent' when prompted to select between Parent and Clinician roles
  3. User is brought to the Parent login screen

- FR-ST-A2: **Select Clinician Role for Login**

  Purpose: Verifies that selecting the Clinician role redirects the user to the appropriate Clinician login screen.

  Type: Manual, Dynamic

  Initial State: User has a Clinician account already created and stored in the database

  Input: Selection of Clinician account role for login

  Output: The expected result is the Clinician account role is selected and User is brought to the Clinician login screen

  Test Case Derivation: The expected output is justified based on FR-A1 in the SRS document

How the test will be performed:

1. Select 'Login' to go to login screen
2. Select 'Clinician' when prompted to select between Parent and Clinician roles
3. User is brought to the Clinician login screen

- FR-ST-A3: **Create New Parent Account (Valid Username)**

Purpose: Confirms that a new Parent account is successfully created when the entered username is not already in use.

Type: Manual, Dynamic

Initial State: User does not have a Parent account stored in the database

Input: Selection of 'Create Account', with a username that does not exist in the database, upon attempting to access the system

Output: The expected result is a new Parent account is created

Test Case Derivation: The expected output is justified based on FR-A2 in the SRS document

How the test will be performed:

1. Select 'Create Account' to go to create account screen
2. Enter unique username that is not in the database
3. Enter account credentials (to complete account create process)
4. Parent account is created

- FR-ST-A4: **Fail to Create Parent Account (Username Already Exists)**

Purpose: Ensures that account creation fails when the Parent username already exists in the system.

Type: Manual, Dynamic

Initial State: User does not have a Parent account stored in the database

Input: Selection of 'Create Account', with a username that exists in the database, upon attempting to access the system

Output: The expected result is a new Parent account fails to be created

Test Case Derivation: The expected output is justified based on FR-A2 in the SRS document

How the test will be performed:

1. Select 'Create Account' to go to create account screen
2. Enter username that already exists in the database
3. System communicates the account could not be created

4. System prompts user to select a new username
5. Parent account is not created

- FR-ST-A5: **Admin Creates New Clinician Account (Valid Username)**

Purpose: Confirms that Admins can create new Clinician accounts when a unique username is provided.

Type: Manual, Dynamic

Initial State: User has Admin privileges, attempting to create a new Clinician account

Input: Admin user selects option to 'Create Account', with a username that does not exist in the database, upon attempting to access the system

Output: The expected result is a new Clinician account is created

Test Case Derivation: The expected output is justified based on FR-A3 in the SRS document

How the test will be performed:

1. Admin user is logged into their account
2. Select 'Create Account' to go to create account screen
3. Enter unique username that is not in the database
4. Enter account credentials (to complete account create process)
5. Clinician account is created

- FR-ST-A6: **Fail to Create Clinician Account (Username Already Exists)**

Purpose: Ensures the system prevents Admins from creating duplicate Clinician accounts using existing usernames.

Type: Manual, Dynamic

Initial State: User has Admin privileges, attempting to create a new Clinician account

Input: Admin user selects option to 'Create Account', with a username that exists in the database, upon attempting to access the system

Output: The expected result is a new Clinician account fails to be created

Test Case Derivation: The expected output is justified based on FR-A3 in the SRS document

How the test will be performed:

1. Admin user is logged into their account
2. Select 'Create Account' to go to create account screen
3. Enter username that already exists in the database

4. System communicates the account could not be created
5. System prompts admin user to select a new username
6. Clinician account is not created

- FR-ST-A7: **Login with Valid Credentials**

Purpose: Confirms that users can successfully log in using valid usernames and matching passwords.

Type: Manual, Dynamic

Initial State: User is on their corresponding role's login page, with an account already created and stored in the database

Input: Unique username and corresponding password that exists in the database

Output: The expected result is a successful login to a user's account

Test Case Derivation: The expected output is justified based on FR-A4 in the SRS document

How the test will be performed:

1. On login screen
2. Enter unique username
3. Enter corresponding password
4. Select login to enter account
5. Logged into account

- FR-ST-A8: **Logout of User Account**

Purpose: Verifies that users are successfully logged out when selecting the logout option and receive confirmation.

Type: Manual, Dynamic

Initial State: User is logged into their account

Input: Selection of 'logout'

Output: The expected result is a successful logout from a user's account

Test Case Derivation: The expected output is justified based on FR-A5 in the SRS document

How the test will be performed:

1. Logged into account
2. Select 'logout'
3. System logs user out of their account

4. Logout confirmation is displayed to the user

4.0.2 Data Collection and Storage

The test cases below focus on ensuring reliable storage and retrieval of multimedia data, privacy compliance by excluding PII, accurate grouping under unique identifiers, and long-term report accessibility, meeting all data storage and organization requirements.

- FR-ST-DSC1: **Store and Retrieve Multimedia Session Data**

Purpose: Ensures the database can store and retrieve complete, uncorrupted multimedia session files (video, audio, JSON).

Type: Automatic, Dynamic

Initial State: Database is empty or initialized with test data.

Input: Multimedia files (video, audio, and structured data files).

Output: The expected result is a success message in console for both storing data and retrieving data

Test Case Derivation: Ensures the database meets storage capacity and integrity requirements as per FR-DSC1 in SRS document.

How the test will be performed:

1. Insert a session containing multimedia files (video, audio, JSON) into the database.
2. Retrieve the session files and check for data integrity by comparing size of stored data with retrieved data.
3. Verify that the retrieved files are uncorrupted and correctly match the original files.

- FR-ST-DSC2: **Store JSON with Flagged Occurrences and Timestamps**

Purpose: Verifies that a JSON file is created and stored alongside each assessment with accurate flagged events and timestamps.

Type: Manual, Dynamic

Initial State: Database is set up to store assessment session data.

Input: Video, audio files, flagged occurrences, and timestamps for each assessment question.

Output: The expected result is the creation of a JSON file that contains flagged occurrences and timestamps which is stored alongside the data.

Test Case Derivation: Allows the system to store all important information for clinicians to use.

How the test will be performed:

1. Insert a test assessment session with video and audio files, flagged occurrences, and timestamps.
2. Query the database to retrieve the session's data and verify the presence of a JSON file with accuracy of flagged occurrences and timestamps.

- FR-ST-DSC3: **Prevent PII Storage in Database**

Purpose: Confirms that any record containing personally identifiable information is either rejected or anonymized.

Type: Automated, Static

Initial State: Database configured to prevent storage of personally identifiable information (PII).

Input: Attempted insertion of a record containing personally identifiable information.

Output: The expected result is that the database rejects any PII-containing records and stores only anonymized data.

Test Case Derivation: Ensures compliance with privacy standards, verifying that no PII is stored in the database, in alignment with HIPAA data minimization and protection requirements.

How the test will be performed:

1. Attempt to insert a record with PII (e.g., name, address).
2. Verify that the system blocks or anonymizes PII, preventing its storage.
3. Retrieve all clinician-accessible data and confirm the absence of PII.

- FR-ST-DSC4: **Group Sessions by Unique User ID**

Purpose: Confirms that session data is grouped and retrieved correctly by the associated user identifier.

Type: Manual, Dynamic

Initial State: Database initialized and ready for storing user session data.

Input: Multiple sessions, each with unique user identifiers.

Output: The expected result is that all session data is stored and grouped correctly according to the unique user identifiers.

Test Case Derivation: Confirms database's grouping and retrievable capabilities, ensuring accurate data organization.

How the test will be performed:

1. Insert multiple sessions into the database, each tagged with a unique user

identifier.
2. Query the database for each user identifier and verify that all associated session data is correctly grouped.
3. Confirm no data is incorrectly associated or left unassociated.

- FR-ST-DSC5: **Retain Reports Over Time by Patient ID**

Purpose: Verifies that each stored report remains accessible and correctly linked to its user throughout the storage period.

Type: Manual, Dynamic

Initial State: Database initialized and ready to store reports with unique identifiers.

Input: Assessment report linked to a patient's unique identifier.

Output: The expected result is that the report is successfully stored, linked to the corresponding patient identifier, and retrievable for at least MAX_STORAGE_TIME.

Test Case Derivation: Verifies long-term storage and retrievability of assessment reports by simulating a shortened retention period (e.g. 7 days), and confirming report accessibility at the start, midpoint, and just before expiration. This ensures confidence in meeting the real retention requirement.

How the test will be performed:

1. Insert a test report with a unique patient identifier into the database.
2. Simulate passage of time or use timestamps to emulate 3 key stages: day 1, midpoint, and just before test retention period ends.
3. At each point, retrieve the report using the identifier and confirm its presence and correctness.
4. Optionally, test behavior just after simulated expiration to ensure proper handling (retained if policy allows or removed if enforced).

4.0.3 Video and Audio Data Analysis

The test cases below focus on ensuring the video analysis model can reliably access session recordings, accurately detect and log speech disturbances, and correctly flag disturbances with timestamps, questions, and user responses to support efficient clinical review.

- FR-ST-VDA1: **Model Accesses and Loads Multimedia Files**

Purpose: Confirms that the analysis model can successfully access and load video/audio files without error.

Type: Dynamic, Automatic

Initial State: Completed assessment sessions are available in the database, with video and audio recordings accessible for processing.

Input: Request by the analysis model to access video and audio data from a completed session.

Output: The expected result is that all videos requested are processed with a success message in the logs

Test Case Derivation: Verifies that the model has reliable access to stored multimedia data, which is critical for processing and analysis.

How the test will be performed:

1. Retrieve the video and audio recordings from several completed sessions.
2. Check that the model successfully accesses the multimedia files for each session without data access errors.
3. Verify that the files are correctly loaded for analysis with no corruption or access issues.

- FR-ST-VDA2: **Detect and Log Disturbances with High Accuracy**

Purpose: Verifies that the analysis model can identify speech disturbances in recordings with high accuracy.

Type: Dynamic, Automatic

Initial State: The analysis model is initialized and ready to process test video and audio data.

Input: Video and audio data containing speech disturbances, interruptions, and other irregularities for analysis.

Output: The expected results is an accuracy of VERY_HIGH_SUCCESS_RATE in a JSON file for number of disturbances found by the model

Test Case Derivation: Confirms that the model's disturbance detection meets the accuracy requirement, reducing bias in the analysis process.

How the test will be performed:

1. Run the model on a test dataset containing known speech disturbances.
2. Compare the disturbances identified by the model with human observations for accuracy validation.
3. Verify that the model achieves at least VERY_HIGH_SUCCESS_RATE accuracy in identifying and logging disturbances.

- FR-ST-VDA3: **Flag Disturbances with Timestamps and Context**

Purpose: Confirms that the model logs timestamps, questions, and user responses with each detected disturbance for clinical review.

Type: Dynamic, Automatic

Initial State: Video and audio data with disturbances has been processed by the analysis model.

Input: Disturbances identified by the model, requiring flags with associated timestamps, assessment questions, and user answers.

Output: The expected result is an accuracy of VERY_HIGH_SUCCESS_RATE in a JSON file for timestamp accuracy

Test Case Derivation: Ensures clinicians can quickly access relevant parts of the assessment with accuracy, aiding in efficient diagnosis.

How the test will be performed:

1. Process a test assessment session with the model, identifying and flagging disturbances.
2. Retrieve flagged disturbances and confirm each has an accurate timestamp, associated question, and user response.
3. Compare the flagged data with human observations and verify at least VERY_HIGH_SUCCESS_RATE accuracy in the model's associations.

4.0.4 Data Processing and Display

This set of test cases will help confirm the system's data retrieval, report generation, and display functionalities to ensure the clinician experience aligns with the project's goals.

- FR-ST-DPD1: **Retrieve Processed Assessment Data from Database**

Purpose: Verifies that the system can accurately retrieve a patient's processed data for report generation within performance limits.

Type: Dynamic, Automatic

Initial State: Database is populated with processed assessment data.

Input: Query request for a specific patient's processed assessment data.

Output: The expected result is the successful retrieval of all relevant assessment data, displayed without errors within MAX_PROCESSING_TIME.

Test Case Derivation: Ensures the system can retrieve data efficiently for report generation, meeting retrieval speed and completeness requirements.

How the test will be performed:

1. Query the database for a test patient's processed assessment results.
2. Measure and record retrieval time, ensuring it does not exceed MAX_PROCESSING_TIME.
3. Verify that all required data is retrieved and matches the stored information.

- FR-ST-DPD2: **Generate Reports from Assessment Data**

Purpose: Confirms that the system can generate full reports containing performance metrics, timestamps, and flags within set time limits.

Type: Dynamic, Automatic

Initial State: Database has assessment data including flagged occurrences, timestamps, and patient performance metrics.

Input: Trigger for report generation based on a retrieved assessment dataset.

Output: The expected result is a generated report containing all required data within MAX_PROCESSING_TIME.

Test Case Derivation: Confirms that report generation is complete, accurate, and within performance constraints.

How the test will be performed:

1. Retrieve a patient's assessment data from the database.
2. Trigger report generation.
3. Confirm the report includes flagged occurrences, timestamps, and performance metrics.
4. Measure and confirm report generation time does not exceed MAX_PROCESSING_TIME.

- FR-ST-DPD3: **Display Report on Clinician Dashboard**

Purpose: Validates that the clinician dashboard displays generated reports with proper formatting and within acceptable loading time.

Type: Dynamic, Manual

Initial State: Generated report available in the database.

Input: Clinician dashboard query to display the generated report.

Output: Report displayed in the clinician's dashboard with accurate formatting, charts, and tables, fully loaded within MAX_PROCESSING_TIME.

Test Case Derivation: Validates the report display function, ensuring usability and speed requirements are met.

How the test will be performed:

1. Query the clinician's dashboard to load the report.
2. Confirm that the report is displayed with correct charts, tables, and formatting.
3. Verify full loading of the report within MAX_PROCESSING_TIME.

- FR-ST-DPD4: **Access and Display Historical Reports**

Purpose: Ensures clinicians can access previously generated reports accurately and quickly for longitudinal patient tracking.

Type: Dynamic, Manual

Initial State: Database has stored reports for previous sessions, each with a unique patient identifier.

Input: Clinician request to access a specific previously generated report.

Output: The expected result is successful retrieval and display of the requested report without errors, within MAX_PROCESSING_TIME.

Test Case Derivation: Ensures that clinicians can reliably access and view past reports, supporting longitudinal patient assessment.

How the test will be performed:

1. Query the database for a stored report using a unique patient identifier.
2. Verify that the retrieved report is complete and accurate.
3. Confirm that the report is displayed within MAX_PROCESSING_TIME.

4.0.5 System Set Up

This set of test cases verifies that the system provides users with the ability to review assessment information, complete necessary hardware checks, and receive guidance before beginning an assessment.

- FR-ST-SS1: **View Pre-Assessment Information**

Purpose: Verifies that users can access important assessment overview content before setup begins.

Type: Static, Manual

Initial State: User is logged into the system and has access to the assessment information page.

Input: User navigates to the page where assessment information is displayed before starting hardware checks.

Output: User is able to view relevant information about the assessment before beginning any hardware checks.

Test Case Derivation: Users need to be aware of the assessment requirements and goals prior to starting the hardware setup to ensure they are prepared. This follows expected output found in FR-SS1 in (SRS) Document.

How the test will be performed:

1. Log in as a user and navigate to the assessment information section.

2. Verify that the page displays necessary details about the assessment, such as the purpose, requirements, and overview.
3. Confirm that the information is accessible and readable to the user.

- FR-ST-SS2: **Run Audio Hardware Check**

Purpose: Ensures the system verifies the user's audio input/output devices are working before assessment.

Type: Dynamic, Manual

Initial State: User is on the hardware check section of the system with audio devices connected.

Input: User initiates the audio hardware check through the system.

Output: User receives confirmation that the audio input and output devices are functioning correctly.

Test Case Derivation: Verifying that audio devices are functional before the assessment prevents issues of recording audio during the assessment, which could affect proper identification of bias. This follows expected output found in FR-SS2 in (SRS) Document.

How the test will be performed:

1. Start the audio hardware check.
2. Confirm that the system prompts the user to test both audio input (microphone) and output (speakers/headphones).
3. Verify that the system indicates successful audio detection when the test is performed.
4. Test with a non-functional audio device and confirm that the system displays an appropriate error.

- FR-ST-SS3: **Run Video Hardware Check**

Purpose: Confirms that the user's video capture device is functional and recognized before beginning an assessment.

Type: Dynamic, Manual

Initial State: User is on the hardware check section of the system with a video device connected.

Input: User initiates the video hardware check through the system.

Output: User receives confirmation that the video capturing device is functioning correctly.

Test Case Derivation: Ensuring video functionality prevents disruptions in assessments that require visual input or interaction and ensures proper recording

is taken for accurate data analysis. This follows expected output found in FR-SS3 in (SRS) Document.

How the test will be performed:

1. Start the video hardware check.
2. Confirm that the system activates the video device and displays a live feed.
3. Verify that the system confirms successful video detection if the feed is displayed correctly.
4. Test with a non-functional video device and confirm that the system displays an error.

- FR-ST-SS4: **Display Assessment Tutorial After Setup**

Purpose: Confirms users are shown a helpful step-by-step tutorial immediately after completing device setup.

Type: Dynamic, Manual

Initial State: User has successfully completed audio and video hardware checks.

Input: User proceeds to the tutorial section after completing hardware checks.

Output: User is directed to a tutorial that explains the assessment process in a step-by-step manner.

Test Case Derivation: Providing a tutorial ensures that users understand the assessment process, improving accuracy and compliance. This follows expected output found in FR-SS4 in (SRS) Document.

How the test will be performed:

1. Complete the audio and video hardware checks.
2. Verify that the system automatically navigates the user to a tutorial section upon completion of the hardware checks.
3. Confirm that the tutorial provides clear, step-by-step instructions on how to proceed with the assessment.

- FR-ST-SS5: **Begin Assessment After Tutorial Completion**

Purpose: Verifies that users can begin the assessment only after they complete all setup steps, including the tutorial.

Type: Dynamic, Manual

Initial State: User has completed the tutorial and is ready to begin the assessment.

Input: User initiates the start of the assessment through the system.

Output: User is taken to the first assessment question, and the assessment begins.

Test Case Derivation: Allowing users to start the assessment on their own

terms helps them feel prepared and reduces errors. This follows expected output found in FR-SS5 in (SRS) Document.

How the test will be performed:

1. After completing the tutorial, select the option to start the assessment.
2. Confirm that the system directs the user to the first assessment question.
3. Verify that the assessment interface is properly displayed and ready for the user to begin answering.

4.0.6 Assessment Interface

This set of test cases ensures the assessment interface supports a smooth and accurate user experience by verifying the functionality of question presentation, response recording, navigation, and feedback mechanisms throughout the assessment.

- FR-ST-AI1: **Start Recording at Assessment Launch**

Purpose: Confirms the system begins recording audio and video when the assessment starts, with visual feedback to the user.

Type: Dynamic, Automatic

Initial State: User has started the assessment and is ready to begin answering questions.

Input: User initiates the assessment.

Output: System begins recording both audio and video, with an indicator showing recording is active.

Test Case Derivation: Audio and video recording are essential for capturing user responses accurately, and an indicator reassures users that recording is in progress. This follows expected output found in FR-AI1 in (SRS) Document.

How the test will be performed:

1. Start the assessment as a user.
2. Confirm that the system begins recording audio and video.
3. Verify that a visible indicator shows that recording is ongoing.

- FR-ST-AI2: **Play Audio Prompt for Each Question**

Purpose: Verifies that each question is introduced with an audio prompt to guide users.

Type: Dynamic, Automatic

Initial State: User is on a new question within the assessment.

Input: System progresses to a new question in the assessment.

Output: The system plays the corresponding audio prompt for the new question.

Test Case Derivation: The audio prompt ensures users understand each question and assessment is running as intended. This follows expected output found in FR-AI2 in (SRS) Document.

How the test will be performed:

1. Progress to a new question in the assessment.
2. Verify that the system automatically plays the corresponding audio prompt.

- FR-ST-AI3: **Display Answer Options for Each Question**

Purpose: Ensures that all expected answer options are visible for each question.

Type: Dynamic, Automatic

Initial State: User has progressed to a new question in the assessment.

Input: System loads a new question.

Output: System displays all possible answer options for the user to select from.

Test Case Derivation: Presenting options ensures the user can make a response selection based on the question, system must mark answer select as right or wrong. This follows expected output found in FR-AI3 in (SRS) Document.

How the test will be performed:

1. Go to a new question in the assessment.
2. Confirm that all answer options associated with the question are displayed.

- FR-ST-AI4: **Highlight Selected Answer**

Purpose: Confirms that selected answers are visually indicated to reduce confusion.

Type: Dynamic, Manual

Initial State: User is viewing the answer options for a question in the assessment.

Input: User selects one of the displayed answer options.

Output: System highlights or otherwise indicates the user's selected option.

Test Case Derivation: Visual confirmation of selection minimizes user error and allows the user to review their choice before confirmation. This follows expected output found in FR-AI4 in (SRS) Document.

How the test will be performed:

1. Select an answer option for a question.
2. Verify that the system indicates the selected option visually (e.g., highlighting).

- FR-ST-AI5: **Confirm Answer and Proceed to Next**

Purpose: Verifies that users are moved to the next step only after confirming their answer.

Type: Dynamic, Manual

Initial State: User has selected an answer option and is ready to proceed.

Input: User confirms their selection.

Output: System moves the user to the next question or stage.

Test Case Derivation: Confirmation helps prevent unintended answers, ensuring accuracy in user responses. This follows expected output found in FR-AI5 in (SRS) Document.

How the test will be performed:

1. Select and confirm an answer for a question.
2. Verify that the system progresses to the next question or stage upon confirmation.

- FR-ST-AI6: **Record Entry and Exit Timestamps**

Purpose: Ensures each question interaction is timestamped to support later review or syncing.

Type: Dynamic, Automatic

Initial State: User is in the process of answering questions within the assessment.

Input: User enters and exits each question.

Output: System records timestamps for entry and exit for each question.

Test Case Derivation: Timestamping provides valuable tracking information for synchronizing recordings and analyzing response timing. This follows expected output found in FR-AI6 in (SRS) Document.

How the test will be performed:

1. Begin the assessment and progress through several questions.
2. Verify that the system logs entry and exit timestamps for each question.

- FR-ST-AI7: **Display Assessment Completion Message**

Purpose: Verifies that the system clearly informs the user when the assessment is complete.

Type: Dynamic, Automatic

Initial State: User has reached the final question in the assessment.

Input: User completes the final question and confirms the selection.

Output: System displays a message informing the user that the assessment is complete.

Test Case Derivation: Notifying the user of completion provides a clear end to the assessment and allows users to exit confidently. This follows expected output found in FR-AI7 in (SRS) Document.

How the test will be performed:

1. Complete the final question in the assessment and confirm the selection.
2. Verify that the system displays a completion message.

4.1 Tests for Nonfunctional Requirements

The following section covers all the nonfunctional requirements specified in the project's SRS document. Each of the nonfunctional requirements are covered by a test in section 4.2. The coverage can be traced in Table 4.

4.1.1 Look and Feel Requirements

These test cases ensure that all appearance and style requirements are addressed effectively, covering navigation, user-friendliness, brand consistency, visual appeal, and responsiveness.

- LF-ST-LFR1: **Navigation Simplicity and Child Usability**

  Purpose: Validates that users can navigate and complete key tasks easily, and that children aged 6–12 can use the system without assistance.

  Type: Dynamic, Manual

  Initial State: Platform initialized and navigable on a test device, prepared for user testing with adults and children.

  Input/Condition: Conduct user tests with participants performing core tasks like starting an assessment, navigating menus, and viewing results.

  Output/Results: Expected results include:

  - No more than three levels of navigation depth, and each screen presents no more than six main options.
  - VERY_HIGH_SUCCESS_RATE of users complete core tasks within MAX_CLICKS clicks.
  - HIGH_SUCCESS_RATE of children aged 6-12 can complete a sample assessment independently.

  Test Case Derivation: This test ensures that the user interface design meets key look-and-feel requirements by validating navigation simplicity, ease of use, and child accessibility. It covers constraints on UI depth and option count (LF-AR1), intuitive navigation and task flow (LF-AR2, LF-AR4), and confirms that the platform is usable without adult guidance by children aged 6-12.

  How the test will be performed:

  1. Observe and record the number of clicks taken by each user to complete key tasks.
  2. Inspect the navigation structure to ensure it meets depth and option constraints.
  3. Test with children to verify completion of assessments independently.

- LF-ST-LFR2: **Visual Design Consistency and Feedback Timing**

Purpose: Ensures that branding and design are consistent and calming, and that user actions receive quick system feedback.

Type: Static and Dynamic, Manual

Initial State: Platform with finalized colour schemes, fonts, and brand assets loaded and available for inspection and user interaction tests.

Input/Condition: Perform visual inspection and feedback collection, along with response-time measurements for interactive elements.

Output/Results: Expected results include:

- Compliance with brand guidelines, style guide, and use of no more than three calming, neutral/pastel tones.
- Positive feedback from child participants indicating a calm, non-stressful environment.
- VERY_HIGH_SUCCESS_RATE consistency in design across all pages.
- HIGH_SUCCESS_RATE of user interactions provide immediate feedback within SHORT_PROCESSING_TIME.

Test Case Derivation: This test validates the system's adherence to aesthetic and branding requirements (LF-AR3), confirms user interaction feedback timing (LF-AR5), and ensures a calm and accessible visual experience for children (LF-SR1, LF-SR2). Both objective checks and subjective user feedback are used to confirm compliance and usability standards.

How the test will be performed:

1. Select at least 30% of platform screens across key user flows (e.g., login, assessment, results) for visual inspection.
2. Compare fonts, colour schemes, and layout elements against the brand style guide and confirm use of no more than three calming colour tones.
3. Conduct usability testing sessions with at least 5 child participants. Ask each participant to describe how they felt about the interface visuals and whether anything was confusing or distracting.
4. Identify at least five types of interactive elements (e.g., buttons, form submissions) and measure response feedback time for each using a stopwatch or browser dev tools. Confirm response occurs within SHORT_PROCESSING_TIME.

4.1.2 Usability and Humanity

The test cases below ensure that the system meets usability and humanity requirements for users to have an enjoyable and accessible experience.

- UH-ST-EOU1: **Ease of Use and Accessibility Feedback**

Purpose: Validates overall usability, clarity, and user satisfaction through a post-assessment survey.

Type: Dynamic, Manual

Initial State: System is complete, functional, and ready for user interaction.

Input/Condition: Users complete one full assessment using the system.

Output/Results: User answers questions in the Usability Survey (6.2), and results are culminated and averaged.
Averages should be at least 'Agree' on the answer scale in section 6.2.

Test Case Derivation: This test confirms that the system is easy to use, accessible, and intuitive (UH-EOU1, UH-EOU2, UH-LI1, UH-UP1, UH-AR1) based on user feedback collected after completing an assessment.

How the test will be performed:

1. User has access to the system
2. User completes one full assessment using the system
3. Upon completion of the assessment, user is requested to fill out a usability survey
4. Results are stored
5. Usability scores for questions 1 through 8 are averaged across users

- UH-ST-PI1: **Multilingual Support Availability**

Purpose: Verifies the system supports multiple languages, enhancing accessibility and inclusivity.

Type: Static, Manual

Initial State: System, including assessments, has been completed.

Input/Condition: List of available languages to perform assessments in is available to be selected and listed

Output/Results: Count the number of available languages for the assessment

Test Case Derivation: This test ensures that users have access to multiple supported languages as outlined in UH-PI1.

How the test will be performed:

1. View list of available languages
2. Count how many languages are available for the assessment

- UH-ST-LI2: **Accessible Help Documentation Link**

Purpose: Ensures that users can easily access documentation from the interface for help and guidance.

Type: Static, Manual

Initial State: User documentation has been completed and made available to users.

Input/Condition: Link to documentation is available on the system's frontend interface, and can be accessed

Output/Results: Verify link takes user to access documentation

Test Case Derivation: This test validates that accessible help documentation is present and reachable from the system interface as required in UH-LI2.

How the test will be performed:

1. Select 'documentation'
2. User goes to documentation screen
3. User has access to view up-to-date, available documentation

4.1.3 Performance

The test cases below ensures that the system meets essential performance metrics, including quick page load times, low latency in video and audio recording, high video resolution, and efficient report generation.

- PR-ST-SL1: **System Page Load Speed Test**

Purpose: Ensure web pages load with full functionality within MAX_LOAD_TIME under normal internet conditions.

Type: Dynamic, Automatic

Initial State: System initialized, strong internet connection

Input/Condition: User navigates to different web pages within the system.

Output/Results: The expected output is that each web page loads fully with all functionalities within MAX_LOAD_TIME.

Test Case Derivation: Verifies system responsiveness and performance under expected usage conditions as required by PR-SL1.

How the test will be performed:

1. Identify the full set of web pages in the system.
2. Randomly select a subset of at least 30% of those pages for testing.

3. Navigate to each selected page and measure the time taken for full load and interactive readiness.
4. Confirm that each page meets the MAX_LOAD_TIME requirement.

- PR-ST-SL2: **Video-Audio Latency Synchronization Test**

Purpose: Verify that recorded video and audio remain in sync and latency is within SHORT_PROCESSING_TIME.

Type: Dynamic, Manual

Initial State: Video and audio recording session initialized.

Input/Condition: Audio and video session of user performing gestures while talking

Output/Results: The expected output is the latency between actions and recorded playback remains under SHORT_PROCESSING_TIME, ensuring synchronization.

Test Case Derivation: Ensures time-sensitive performance requirements for synchronization between user behavior and system response as required by PR-SL2.

How the test will be performed:

1. Begin recording a session.
2. Have the user perform timed actions while recording.
3. Play back the recording and measure the latency between actions and their corresponding timestamps.
4. Confirm latency does not exceed SHORT_PROCESSING_TIME.

- PR-ST-SL3: **Video Quality Storage Test**

Purpose: Ensure videos recorded and retrieved from the system are stored at a resolution of at least AVERAGE_RESOLUTION.

Type: Dynamic, Manual

Initial State: System configured to store video data.

Input/Condition: Video recorded and stored from assessment session.

Output/Results: The expected output is the video quality is at least AVERAGE_RESOLUTION when retrieving or storing.

Test Case Derivation: Ensures the system maintains minimum quality standards for video data as required by PR-SL3.

How the test will be performed:

1. Record a session and store the video.
2. Retrieve the stored video and verify its resolution.

3. Confirm the resolution is AVERAGE_RESOLUTION or higher.

The test cases below focus on verifying the system's precision in detecting and analyzing speech and gesture disturbances, ensuring timestamp alignment, and maintaining 100% accuracy in assessment data.

- PR-ST-PA1: **Disturbance Detection Accuracy Test**

Purpose: Ensure model achieves VERY_HIGH_SUCCESS_RATE in detecting speech and gesture disturbances.

Type: Manual, Dynamic

Initial State: Video and Audio analysis model is loaded with sample audio data.

Input/Condition: Data containing speech patterns and video with known disturbances.

Output/Results: The expected output is that the analysis achieves VERY_HIGH_SUCCESS_RATE in detecting speech and gesture disturbances.

Test Case Derivation: Verifies the precision of detection and classification in the audio/video analysis model, fulfilling PR-PA1.

How the test will be performed:

1. Load test data files with known disturbances into the analysis model.
2. Compare detected disturbances with human-reviewed observations.
3. Verify that the model correctly identifies at least VERY_HIGH_SUCCESS_RATE of disturbances.

- PR-ST-PA3: **Timestamp Synchronization Accuracy Test**

Purpose: Confirm timestamps for user actions are accurate within SHORT_PROCESSING_TIME margin.

Type: Dynamic, Manual

Initial State: Timestamp function is synchronized with real-time actions.

Input/Condition: User performs actions in the recorded session.

Output/Results: The expected output is that the timestamps delay within SHORT_PROCESSING_TIME of the real-time action.

Test Case Derivation: Validates synchronization accuracy between user actions and system-generated timestamps per PR-PA3.

How the test will be performed:

1. Record a session with specific user actions.
2. Analyze timestamps and compare them to the actual timing of the actions.

3. Confirm each timestamp falls within SHORT_PROCESSING_TIME margin of the real action.

- PR-ST-PA4: **Assessment Answer Key Accuracy Test**

Purpose: Ensure answer key accuracy reaches MAX_SUCCESS_RATE and is free of inconsistencies.

Type: Manual, Static

Initial State: Assessment answer key is loaded.

Input/Condition: Manual verification of the answer key's accuracy.

Output/Results: The expected output is that the answer key is MAX_SUCCESS_RATE.

Test Case Derivation: Confirms correctness and completeness of stored answer keys against expected outputs as defined in PR-PA4.

How the test will be performed:

1. Manually review each entry in the assessment answer key.
2. Check for errors or inconsistencies in each answer.
3. Confirm all answers are correct, ensuring MAX_SUCCESS_RATE.

The test cases below validate the system's ability to handle errors, back up data reliably, and enforce strict input validation.

- PR-ST-RFT1: **Error Handling and Message Clarity Test**

Purpose: Ensure the system handles common user errors and displays clear, informative error messages for at least VERY_HIGH_SUCCESS_RATE of cases.

Type: Dynamic, Automatic

Initial State: System is operational with error handling in place.

Input/Condition: User initiates actions known to cause common errors.

Output/Results: The expected output is the system displays clear error messages for at least VERY_HIGH_SUCCESS_RATE of the common errors encountered.

Test Case Derivation: Verifies the system's ability to handle errors with informative feedback, ensuring robustness and user understanding as required by PR-RFT1.

How the test will be performed:

1. Simulate common user errors, such as invalid inputs or incorrect file uploads.
2. Observe system response and displayed error messages.
3. Verify clarity and accuracy of error messages in at least

VERY_HIGH_SUCCESS_RATE.

- PR-ST-RFT2: **Automated Data Backup Reliability Test**

Purpose: Ensure automated data backups run successfully on the first of each month and complete within MONTHLY_BACKUP.

Type: Dynamic, Automatic

Initial State: Database backup processes configured and operational in the system.

Input/Condition: Monthly data backup event.

Output/Results: The expected output is that the system performs a data backup within a MONTHLY_BACKUP timeframe on the first of each month.

Test Case Derivation: Ensures system reliability through automated data backups to prevent loss of critical assessment data as required by PR-RFT2.

How the test will be performed:

1. A data backup is triggered.
2. Record the duration of the backup process.
3. Confirm that backup completes within MONTHLY_BACKUP.

The test cases below ensure that the system can handle the expected user load, data storage needs, and simultaneous uploads without performance degradation.

- PR-ST-CR1: **Minimum User Load Stability Test**

Purpose: Confirm system can support MIN_USERS simultaneously with no performance degradation.

Type: Dynamic, Automatic

Initial State: System initialized with maximum user capacity parameters.

Input/Condition: System loaded with MIN_USERS accounts.

Output/Results: The expected result is that the system operates stably and manages all accounts without issues.

Test Case Derivation: Validates the system's ability to scale under a minimum expected user load while maintaining stable performance as described in PR-CR1.

How the test will be performed:

1. Create and load MIN_USERS accounts into the system.

2. Monitor system performance metrics, including stability and response time.
3. Confirm system maintains stable performance.

- PR-ST-CR2: **Annual Data Storage Capacity Test**

Purpose: Ensure the system handles at least MIN_STORAGE of annual data without degradation.

Type: Dynamic, Automatic

Initial State: System is initialized with required storage capacity.

Input/Condition: Data stored in the database approaches MIN_STORAGE annually.

Output/Results: The expected result is that the system accommodates MIN_STORAGE of data without loss of performance.

Test Case Derivation: Confirms that the database can handle expected yearly data growth without system slowdowns or data loss, fulfilling PR-CR2.

How the test will be performed:

1. Load MIN_STORAGE data into database.
2. Monitor database performance metrics, such as access time and error rate.
3. Confirm system's ability to manage MIN_STORAGE without performance impact.

The test cases below confirm that the system can scale effectively to accommodate an increasing user base, data volume, and computational needs over time.

- PR-ST-SE1: **Scalability Under User Growth Test**

Purpose: Ensure system maintains performance with a YEARLY_INCREASE_PERCENTAGE growth in user base.

Type: Dynamic, Automatic

Initial State: System operational with current allocated user amount.

Input/Condition: Increase user base by YEARLY_INCREASE_PERCENTAGE.

Output/Results: The expected result is that the system maintains performance while handling user growth.

Test Case Derivation: Ensures the system is scalable and can maintain acceptable performance under anticipated user base growth as required by PR-SE1.

How the test will be performed:

1. Simulate a YEARLY_INCREASE_PERCENTAGE increase in the user base

by increasing user base parameters.
2. Monitor system metrics such as response time and error rate.
3. Confirm that the system operates within acceptable performance metrics post-increase.

The test cases below verify the system's reliability across updates and compatibility with major operating systems.

- PR-ST-LR1: **System Reliability Across Updates Test**

Purpose: Ensure release builds remain stable with a failure rate below LOW_FAILURE_RATE during ongoing development updates.

Type: Dynamic, Automatic

Initial State: System release build in use, with ongoing development updates.

Input/Condition: System stability monitored over successive updates.

Output/Results: The expected result is that the system maintains a failure rate below LOW_FAILURE_RATE in release builds.

Test Case Derivation: Ensures that new updates do not compromise system reliability and meet the fault tolerance criteria of PR-LR1.

How the test will be performed:

1. Conduct routine tests on the release build during ongoing development updates.
2. Monitor system logs for errors or malfunctions.
3. Verify that the failure rate remains below LOW_FAILURE_RATE.

- PR-ST-LR2: **Cross-Platform Compatibility Test**

Purpose: Ensure system operates without functional issues across major platforms (Windows, macOS, Linux, Android, iOS).

Type: Dynamic, Manual

Initial State: System configured for compatibility testing across platforms.

Input/Condition: System loaded on Windows, macOS, Linux, Android, and iOS.

Output/Results: The expected result is that the system functions correctly across all platforms.

Test Case Derivation: Validates cross-platform operability to ensure consistent performance across major operating systems, as required by PR-LR2.

How the test will be performed:

1. Run the system on each specified operating system.
2. Perform standard operations and monitor user experience on each platform.
3. Confirm that the system operates without issues on all platforms.

4.1.4 Operational and Environmental

The test cases below ensure that the system can be used in a variety of environments, adhere to conditions users are expected to operate under, and maintain the necessary capabilities to interact with external systems and devices within those environments.

- OE-ST-EPE1: **Responsive Display Across Devices Test**

Purpose: Verifies that the system scales correctly across a variety of screen sizes, providing a consistent and usable UI experience.

Type: Dynamic, Manual

Initial State: System is complete, functional, and ready for user interaction.

Input/Condition: Testing the system, including the assessment, on a variety of screen sizes.

Output/Results: The system's displayed elements will scale appropriately to different screen sizes.

Test Case Derivation: Validates OE-EPE1 by confirming the system provides a consistent and usable UI across different device screen sizes.

How the test will be performed:

1. User logs into the system.
2. User completes one full assessment using the system.
3. Upon completion of the assessment, user is requested to fill out a usability survey.
4. User answers questions about their screen size and whether the test scaled accordingly.
5. Results are stored for review.

- OE-ST-WE1: **Environmental Setup and Input Quality Verification**

Purpose: Ensures the system checks for environmental readiness, including internet, audio, and video conditions.

Type: Dynamic, Manual

Initial State: System, including assessments, has been completed.

Input/Condition: User attempts to start system setup.

Output/Results: Device verification is displayed on-screen, informing the user that the environment is suitable for the assessment.

Test Case Derivation: Confirms OE-WE1 and OE-WE2 by testing the system's ability to verify internet connectivity and input quality, ensuring environmental readiness.

How the test will be performed:

1. Select 'system setup'.
2. System checks if connected to the internet.
3. System checks audio input is not noisy.
4. System checks video input is clear.
5. System displays to the user their device is ready for the assessment to be used in the current environment.

- OE-ST-IA1: **External Server Interoperability Test**

Purpose: Ensures that the system can accurately store and retrieve assessment results using external storage systems.

Type: Dynamic, Manual

Initial State: System is connected to an external server for retrieving and storing data.

Input/Condition: Assessment is complete, and results need to be stored.

Output/Results: Verify results are stored in the external server.

Test Case Derivation: Ensures OE-IA1 by confirming system interoperability with external storage services for accurate assessment result storage.

How the test will be performed:

1. Complete the assessment.
2. Access the external server.
3. Check if results have been uploaded to the server.
4. Access results to ensure data has been uploaded successfully.

4.1.5 Maintainability and Support Requirements

These test cases ensure the platform meets its maintenance, support, and adaptability requirements effectively, including modular design, tutorial clarity, and accessibility across devices.

- MS-ST-MSA1: **Modular Component Update and GitHub Feedback Test**

Purpose: Confirms that platform modules can be updated independently with minimal external changes and that user feedback mechanisms through GitHub are functional.

Type: Static and Dynamic, Manual

Initial State: Modular platform architecture with access to the component's source code. A direct link to GitHub is also available on the platform.

Input/Condition: Perform updates to individual components and simulate user feedback submissions via the GitHub repository.

Output/Results: Expected results include:

- Each component update does not exceed NUM_CODE_LINES lines of code edited outside the updated module.
- Users can submit issues and feature requests directly to GitHub, categorized as issues, feature requests, or feedback.

Test Case Derivation: Validates MS-MR1 and MS-SR1 by confirming modularity of component updates and the presence of a support feedback mechanism via GitHub.

How the test will be performed:

1. Perform code updates on isolated components and verify changes are contained within NUM_CODE_LINES lines outside the component.
2. Test submission flow to GitHub, verifying links are accessible and that issues and requests are categorized correctly.

- MS-ST-MSA2: **Tutorial Effectiveness and Onboarding Success Test**

Purpose: Validates that new users can follow the tutorial to complete essential platform tasks without prior training.

Type: Dynamic, Manual

Initial State: Platform initialized with a tutorial accessible from the homepage.

Input/Condition: New user group follows the tutorial to complete primary tasks (e.g., starting an assessment).

Output/Results: Expected results include:

- HIGH_SUCCESS_RATE of users can complete core tasks correctly after following the tutorial.

Test Case Derivation: Supports MS-SR2 by confirming that the platform's tutorial is effective in helping new users complete essential tasks without prior training.

How the test will be performed:

1. Guide users through the tutorial and observe task completion rates.

2. Collect feedback on tutorial clarity and assess if 90% of users can independently complete tasks.

- MS-ST-MSA3: **Cross-Device Feature Accessibility and Responsiveness Test**

Purpose: Ensures essential features are accessible and functional on a wide range of devices and screen sizes.

Type: Dynamic, Manual

Initial State: Platform accessible across multiple devices and screen sizes, from mobile (MIN_SCREEN_SIZE) to desktop (MAX_SCREEN_SIZE).

Input/Condition: Load and navigate the platform across multiple devices to evaluate responsive design and functionality.

Output/Results: Expected results include:

- MAX_SUCCESS_RATE of essential features are fully functional and readable across all screen sizes tested.

Test Case Derivation: Ensures MS-AR1 by confirming that the platform is accessible and functions consistently across all screen resolutions and device types.

How the test will be performed:

1. Access the platform on multiple screen sizes and test for display, layout, and functionality.
2. Verify that all features are usable and adapt responsively without readability or functionality loss.

4.1.6 Cultural Requirements

These tests ensure that the platform respects cultural sensitivities and provides full bilingual support, enhancing inclusivity and accessibility for diverse user groups.

- CU-ST-CUR1: **Cultural Sensitivity and Content Review Test**

Purpose: Confirms all visual and textual content is culturally sensitive and approved by both expert reviewers and diverse user feedback.

Type: Static and Dynamic, Manual

Initial State: Platform content (language and imagery) is finalized and presented for review.

Input/Condition: A cultural consultant reviews all language and imagery, and user acceptance testing gathers feedback from a diverse set of users.

Output/Results: Expected results include:

– MAX_SUCCESS_RATE of reviewed content is confirmed as culturally sensitive with no instances of offensive language or imagery.

Test Case Derivation: Verifies CU-CR1 by confirming the platform's content is reviewed for cultural appropriateness and validated by user feedback from diverse backgrounds.

How the test will be performed:

1. A cultural consultant examines all text and imagery for potential cultural insensitivity.
2. Conduct user acceptance testing with a diverse group and gather feedback on the platform's cultural sensitivity.
3. Validate that all feedback confirms no culturally insensitive content.

- CU-ST-CUR2: **Bilingual Functionality and Translation Accuracy Test**

Purpose: Confirms that all platform content, assessments, and instructions are fully translated and accurate in both English and Mandarin.

Type: Static, Manual

Initial State: Platform is available in both English and Mandarin, with all interface elements and assessments translated.

Input/Condition: Switch between language settings to review each text, instruction, and assessment in both languages.

Output/Results: Expected results include:

– MAX_SUCCESS_RATE of assessment content is fully translated and functional in both English and Mandarin with no untranslated elements.

Test Case Derivation: Addresses CU-CR2 by confirming full bilingual support across the entire platform, ensuring accessibility for both English and Mandarin speakers.

How the test will be performed:

1. Navigate through the platform in both English and Mandarin settings, verifying translations for each section.
2. Confirm that all assessments, instructions, and interface elements are accurately translated and free from language discrepancies.

4.1.7 Security

The test cases below ensure that the system meets essential security requirements including authentication of users and encryption of confidential data in the system.

- SR-ST-AC1: **Admin-Only Clinician Account Creation Test**

Purpose: Ensures that only Admin users can create and manage clinician accounts.

Type: Dynamic, Manual

Initial State: System has multiple user roles: Admin, Parent, and Clinician.

Input/Condition: User with Admin role attempts to create and assign accounts to clinicians

Output/Results: Only Admin users can access and execute functions related to clinician account creation

Test Case Derivation: Ensures SR-AC1 by confirming that role-based access control restricts clinician account creation to Admin users only.

How the test will be performed:

1. Log in as an Admin and attempt to create and view clinician accounts. Verify that the action succeeds.
2. Log in as a non-Admin (e.g., Parent or Clinician) and try to access the same function. Confirm that access is denied with an appropriate error message.

- SR-ST-AC2: **Parent User Functional Access Control Test**

Purpose: Validates that Parent users can complete assessments but cannot access restricted functions.

Type: Dynamic, Manual

Initial State: Parent role created and available for testing.

Input/Condition: User with Parent role logs in and attempts to complete assessments.

Output/Results: Parent users can create their account, complete assessments, and log out successfully.

Test Case Derivation: Validates SR-AC2 by ensuring that Parent users can perform all necessary functions without elevated privileges.

How the test will be performed:

1. Log in as a Parent user and attempt to start and complete an assessment. Verify successful completion and logout.
2. Attempt to access administrative functions (e.g., creating clinician accounts). Verify that access is denied with an appropriate message.

- SR-ST-AC3: **Clinician Role Access Restriction Test**

Purpose: Confirms that clinicians can view results but not initiate or take assessments.

Type: Dynamic, Manual

Initial State: Clinician role with restricted access is created and available for testing.

Input/Condition: User with Clinician role logs in and attempts to view assessment results.

Output/Results: Clinician users can view assessment results but cannot start or complete assessments as a Parent user would.

Test Case Derivation: Confirms SR-AC3 by testing access control boundaries for the Clinician role, ensuring view-only access.

How the test will be performed:

1. Log in as a Clinician and attempt to view completed assessment results. Confirm access is granted.
2. Attempt to start or complete an assessment and confirm access is denied with an error message indicating unauthorized action.

- SR-ST-AC4: **Authentication Success and Failure Test**

Purpose: Ensures login is only allowed with correct credentials across all user roles.

Type: Dynamic, Manual

Initial State: System with user login functionality.

Input/Condition: Users attempt to log in with correct and incorrect credentials.

Output/Results: Users can only log in with correct credentials; unauthorized access attempts are denied.

Test Case Derivation: Ensures SR-AC4 by verifying secure authentication mechanisms that prevent unauthorized access.

How the test will be performed:

1. Attempt to log in with valid credentials for multiple roles (Admin, Parent, Clinician). Confirm successful login.
2. Attempt to log in with incorrect credentials (e.g., incorrect username or password). Confirm that login is denied and an error message is shown.

- SR-ST-P1: **Privacy and Data Protection Law Compliance Test**

Purpose: Verifies legal compliance with data privacy laws and regulations.

Type: Static, Manual

Initial State: System ready for pre-release review.

Input/Condition: Review documentation to ensure adherence to data protection and privacy laws in the region.

Output/Results: Confirm all applicable data protection requirements are met.

Test Case Derivation: Validates SR-P1 by confirming the system adheres to legal and regulatory privacy frameworks such as HIPAA.

How the test will be performed:

1. Conduct a documentation review with legal and compliance teams, focusing on privacy policies, data handling, and retention practices.
2. Verify that all data collection, storage, and usage adhere to applicable privacy laws (e.g., HIPAA).

- SR-ST-P2: **Data Encryption in Transit and at Rest Test**

Purpose: Confirms that sensitive data is protected through encryption both during transmission and while stored.

Type: Static, Automated

Initial State: System with sensitive data handling enabled.

Input/Condition: Examine data in transit and at rest.

Output/Results: Data remains encrypted according to standard encryption protocols during transit and at rest.

Test Case Derivation: Verifies SR-P2 by ensuring encryption mechanisms are implemented and meet security best practices for sensitive data.

How the test will be performed:

1. Use network monitoring tools to capture data packets during transmission to ensure data is encrypted.
2. Review database configuration to verify that data at rest is encrypted. Decrypting should only be possible by authorized clinician accounts.

- SR-ST-P3: **PII-Free Storage Verification Test**

Purpose: Ensures no unnecessary personally identifiable information (PII) is stored.

Type: Static, Manual

Initial State: System configured for data collection.

Input/Condition: Examine data storage for PII.

Output/Results: System does not store any personal identifiable information beyond username and assessment recordings.

Test Case Derivation: Ensures SR-P3 by confirming that only non-sensitive identifiers are retained, reducing privacy risk.

How the test will be performed:

1. Conduct a database inspection and audit to ensure no PII (e.g., address, date of birth, names) is stored.
2. Verify data schema to confirm that only usernames and recordings are stored.

- SR-ST-IM1: **Password Strength Enforcement Test**

Purpose: Verifies the system enforces strong password policies to enhance account security.

Type: Dynamic, Manual

Initial State: System is in the account creation phase, requiring users to set up passwords.

Input/Condition: User attempts to create an account with both weak and strong passwords.

Output/Results: Account creation is completed only when a strong password, meeting specified security criteria, is entered. Weak passwords are rejected with an error message detailing the password requirements.

Test Case Derivation: Validates SR-IM1 by testing enforcement of strong password policies to protect against unauthorized access.

How the test will be performed:

1. Attempt to create an account with a weak password (e.g., fewer than 8 characters, no special characters or numbers). Verify rejection and error message.
2. Attempt to create an account with a valid strong password and verify that creation succeeds.
3. Repeat with multiple variations to ensure consistent enforcement.

4.1.8 Compliance

The test cases below ensure that the system meets essential compliance requirements including security of the system, following the rules of the clinician and law.

- CR-ST-D1: **HIPAA-Compliant Secure Data Storage Test**

Purpose: Verifies that assessment data is stored securely using encryption and minimal identifiers to meet HIPAA compliance.

Type: Static, Automated

Initial State: User assessment data, including video and audio recordings, is stored in the system.

Input/Condition: Examine the storage configuration and security measures applied to user assessment data.

Output/Results: User assessment data is securely stored and associated only with usernames, ensuring no additional personally identifiable information is included. Data at rest is protected by strong encryption in compliance with HIPAA Security Rule standards.

Test Case Derivation: Verifies CR-STD1 by confirming the system enforces HIPAA-compliant storage standards, protecting sensitive health data through encryption and limiting identifiers.

How the test will be performed:

1. Review database schema to confirm that assessment data is linked only to usernames and does not include any additional identifiable information.
2. Conduct an inspection of the encryption protocols used for stored data to verify compliance with security standards.
3. Attempt unauthorized access to stored assessment data to ensure encryption and security measures prevent access by unauthorized users.

4.2 Traceability Between Test Cases and Requirements

Table 3: Traceability Table Between System Test Cases and Functional Requirements

	FR-ST-A1	FR-ST-A2	FR-ST-A3	FR-ST-A4	FR-ST-A5	FR-ST-A6	FR-ST-A7	FR-ST-A8	FR-ST-DSC1	FR-ST-DSC2	FR-ST-DSC3	FR-ST-DSC4	FR-ST-DSC5	FR-ST-VDA1	FR-ST-VDA2	FR-ST-VDA3	FR-ST-DPD1	FR-ST-DPD2	FR-ST-DPD3	FR-ST-DPD4	FR-ST-SS1	FR-ST-SS2	FR-ST-SS3	FR-ST-SS4	FR-ST-SS5	FR-ST-AI1	FR-ST-AI2	FR-ST-AI3	FR-ST-AI4	FR-ST-AI5	FR-ST-AI6	FR-ST-AI7
FR-A1	X	X																													
FR-A2			X	X																											
FR-A3					X	X																									
FR-A4							X																								
FR-A5								X																							
FR-SS1																					X										
FR-SS2																						X									
FR-SS3																							X								
FR-SS4																								X							
FR-SS5																									X						
FR-AI1																										X					
FR-AI2																											X				
FR-AI3																												X			
FR-AI4																													X		
FR-AI5																														X	
FR-AI6																															X
FR-AI7																															X
FR-DSC1									X																						
FR-DSC2										X																					
FR-DSC3											X																				
FR-DSC4												X																			
FR-DSC5													X																		
FR-VADA1														X																	
FR-VADA2															X																
FR-VADA3																X															
FR-DPD1																	X														
FR-DPD2																		X													
FR-DPD3																			X												
FR-DPD4																				X											

Table 4: Traceability Table Between System Test Cases and Nonfunctional Requirements

System test	LF-ST-LFR1	LF-ST-LFR2	UH-ST-EOU1	UH-ST-PI1	UH-ST-LI2	PR-ST-SL1	PR-ST-SL2	PR-ST-SL3	PR-ST-PA1	PR-ST-PA3	PR-ST-PA4	PR-ST-RFT1	PR-ST-RFT2	PR-ST-CR1	PR-ST-CR2	PR-ST-SE1	PR-ST-LR1	PR-ST-LR2	OE-ST-EPE1	EO-ST-WE1	OE-ST-IA1	MS-ST-MSA1	MS-ST-MSA2	MS-ST-MSA3	CU-ST-CUR1	CU-ST-CUR1	SR-ST-AC1	SR-ST-AC2	SR-ST-AC3	SR-ST-AC4	SR-ST-IM1	CR-ST-D1	SR-ST-P1	SR-ST-P2	SR-ST-P3
LF-AR1	X																																	
LF-AR2	X																																	
LF-AR3		X																																
LF-AR4	X																																	
LF-AR5		X																																
LF-SR1		X																																
LF-SR2		X																																
UH-EOU1			X																															
UH-EOU2			X																															
UH-PI1				X																														
UH-LI1			X																															
UH-LI2					X																													
UH-UP1			X																															
UH-AR1			X																															
PR-SL1						X																												
PR-SL2							X																											
PR-SL3								X																										
PR-SL4																																		
PR-SCL1																																		
PR-PA1									X																									
PR-PA2									X																									
PR-PA3										X																								
PR-PA4											X																							
PR-PA5											X																							
PR-RFT1												X																						
PR-RFT2													X																					
PR-RFT3												X																						
PR-CR1														X																				
PR-CR2															X																			
PR-CR3														X																				
PR-CR4														X																				
PR-SE1																X																		
PR-SE2															X																			
PR-SE3															X																			
PR-LR1																	X																	
PR-LR2																		X																
OE-EPE1																			X															
OE-WE1																				X														
OE-WE2																				X														
OE-IA1																					X													
MS-MR1																						X												
MS-SR1																						X												
MS-SR2																							X											
MS-AR1																								X										
SR-AC1																										X								
SR-AC2																											X							
SR-AC3																												X						
SR-AC4																													X					
SR-P1																																X		
SR-P2																																	X	
SR-P3																																		X
SR-IM1																														X				
CU-CR1																								X										
CU-CR2																									X									
CR-STD1																															X			

5 Unit Test Description

Section 5, Unit Test Description, will be filled in prior to the revision 0 deliverable. The team decided to leave in the template instructions as a guide to indicate our desired return to the document.

All the modules described in the MIS will be within the scope of unit testing for the following 4 services our application comprises of; Authentication Service, Question Bank Service, Result Storage Service, and Media Processing Service.

5.1 Unit Testing Scope

All modules described in the MIS fall under the scope of unit tests and will be tested in some way, either with individual tests or tests that touch multiple components.

5.2 Tests for Functional Requirements

Authentication Service Tests

Unit Test Name: Should Register a New Clinician

Function(s) Tested: `clinicianSignup` (in `clinician.controller.js`)

Input: Valid clinician registration data (firstname, lastname, username, email, password)

Expected Output: JSON response "message": "Clinician created", "user": "username": "drjohndoe"

Relevant Test Case(s): FR-ST-A6

Unit Test Name: Should Return Error for Existing Username

Function(s) Tested: `clinicianSignup` (in `clinician.controller.js`)

Input: Clinician registration data with an existing username

Expected Output: HTTP error response with status 400

Relevant Test Case(s): FR-ST-A4

Unit Test Name: Should Login a Clinician with Valid Credentials

Function(s) Tested: `clinicianLogin` (in `clinician.controller.js`)

Input: Valid login credentials (username and password)

Expected Output: JSON response indicating successful login (i.e., `res.json` is called)

Relevant Test Case(s): FR-ST-A7

Media Processing Service Tests

Unit Test Name: Should Upload and Process Media

Function(s) Tested: POST `/media/upload` endpoint (in media processing controller)

Input: Valid video file upload (if file exists)

Expected Output: HTTP 200 response with JSON "message": "Processing started"

Relevant Test Case(s): FR-ST-DSC1

Unit Test Name: Should Get Processed Media Successfully

Function(s) Tested: GET `/media/processed` endpoint (in media processing controller)

Input: No input (simple GET request)

Expected Output: HTTP 200 response with JSON containing a valid media URL

Relevant Test Case(s): FR-ST-VDA1

Question Service Tests

Unit Test Name: Should Return Correct JSON When Retrieving a Test Question Information

Function(s) Tested: GET `/questions/english/matching/1` endpoint (in question controller/route)

Input: HTTP GET request to `/questions/english/matching/1`

Expected Output: HTTP 200 response with JSON containing keys: `id` (Number), `title` (String), `sound` (String), `options` (Array of objects with `id` and `image`), and `correctAnswer` (String, e.g., "b")

Relevant Test Case(s): FR-ST-DPD1

Result Service Tests

Unit Test Name: Should Return Calculated Result of a Specific Test Result.

Function(s) Tested: `calculateResult` (in `results.controller.js`)

Input: JSON body with "score": 10, "multiplier": 2

Expected Output: JSON response "result": 20

Relevant Test Case(s): FR-ST-DPD2

5.3 Tests for Nonfunctional Requirements

5.3.1 Performance and Security Tests

Unit Test Name: Should Encrypt Data Before Storing

Function(s) Tested: Database insertion function (e.g., `storeEncryptedData`)

Input: Plain text user data

Expected Output: Data stored in an encrypted format, verified by retrieval and decryption

Relevant Test Case(s): SR-ST-P2

Unit Test Name: Should Prevent Unauthorized Access to User Data

Function(s) Tested: Authentication middleware (e.g., `verifyToken`)

Input: Unauthorized request to access protected resources

Expected Output: HTTP 401 response with an error message

Relevant Test Case(s): SR-ST-AC3

5.4 Traceability Between Test Cases and Modules

Table 5: Traceability Table Between System Test Cases and Modules

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17
FR-ST-A1	X	X	X		X			X									
FR-ST-A2	X	X	X		X			X									
FR-ST-A3		X	X		X			X									
FR-ST-A4		X	X		X			X									
FR-ST-A5	X		X		X			X									
FR-ST-A6	X		X		X			X									
FR-ST-A7					X			X									
FR-ST-A8					X			X									
FR-ST-DSC1				X		X		X									
FR-ST-DSC2				X		X	X			X		X	X				
FR-ST-DSC3				X	X												
FR-ST-DSC4				X	X												
FR-ST-DSC5				X				X		X	X						
FR-ST-VDA1				X			X					X	X				
FR-ST-VDA2				X			X	X				X	X				
FR-ST-VDA3				X			X	X				X	X				
FR-ST-DPD1				X		X		X		X	X						
FR-ST-DPD2				X		X		X		X	X						
FR-ST-DPD3	X			X		X		X		X	X						
FR-ST-DPD4	X			X		X		X		X	X						
FR-ST-SS1		X	X														
FR-ST-SS2		X	X														
FR-ST-SS3		X	X														
FR-ST-SS4		X	X														
FR-ST-SS5		X	X						X								
FR-ST-AI1		X	X	X			X		X			X	X	X	X	X	
FR-ST-AI2		X	X	X			X		X			X	X	X	X	X	
FR-ST-AI3		X	X	X			X		X			X	X	X	X	X	
FR-ST-AI4		X	X	X			X		X			X	X	X	X	X	
FR-ST-AI5		X	X	X		X	X		X			X	X	X	X	X	
FR-ST-AI6		X	X	X			X		X			X	X	X	X	X	
FR-ST-AI7		X	X	X			X		X			X	X	X	X	X	X

6 Appendix

Below is additional information relevant to the document.

6.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

Variable Name	Value
MAX_PROCESSING_TIME	10 seconds
SHORT_PROCESSING_TIME	0.5 seconds
MAX_SUCCESS_RATE	100%
VERY_HIGH_SUCCESS_RATE	95%
HIGH_SUCCESS_RATE	90%
NUM_CODE_LINES	10
MIN_SCREEN_SIZE	4 inches
MAX_SCREEN_SIZE	27 inches
MAX_STORAGE_TIME	7 years
MAX_LOAD_TIME	3 seconds
AVERAGE_RESOLUTION	720p
MONTHLY_BACKUP	4 hour
MIN_USERS	2000 users
MIN_STORAGE	10TB
YEARLY_INCREASE_PERCENTAGE	10%
LOW_FAILURE_RATE	1%
MAX_CLICKS	5

Table 6: Variable Names and Values

6.2 Usability Survey Questions

The following questions depict the first draft of the team's usability.
The usability survey will be conducted after participants have engaged with testing and using the system for at least one iteration of the assessment.

Please select the statement that best describes your experience for each of the following:

1. Learning how to use the system was easy:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

2. Setting up the system was easy:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

3. I found the assessment interface easy to use:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

4. Navigating the interface was easy:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

5. All the button interactions reacted and responded how I thought they should:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

6. The information on screen was easy to read and understand:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

7. I like the organization of the assessment interface:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

8. I enjoyed my overall experience using the TeleHealth Insights platform:

   Strongly Disagree Disagree Neutral Agree Strongly Agree

Answer the following:

9. What was the most difficult part of using the platform?

Insert answer here...

10. Did you encounter any bugs/problems while using the platform? If so, what were they?

Insert answer here...

11. What was your favourite part of the experience? Why?

Insert answer here...

12. What was your least favourite part of the experience? Why?

Insert answer here...

13. Were there any aspects of the platform that you found unnecessary? Why?

Insert answer here...

14. Which part of the platform needs the most improvement? Why?

    Insert answer here...

15. What would you like changed to make the platform easier to use?

    Insert answer here...

16. What is the device you ran the system on?

    Insert answer here...

17. Did the screen's visuals scale appropriately to the screen size?

    Insert answer here...

18. Do you have any additional feedback?

    Insert answer here...

Appendix — Reflection

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

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

   Parisha: The work of the VnV plan was divided very well, with team members working on testing of non functional and functional requirements that match the ones they worked on for the SRS document. This made it a lot more efficient and easier to understand what to write for the testing requirements.

   Mitchell: One of the things that went well during this deliverable was splitting up the System Tests for both Functional Requirements and Nonfunctional Requirements. The team decided to assign the functional and nonfunctional tests to the team members that wrote those particular requirements in the SRS. This allowed the team to use their prior knowledge on the requirements to develop detailed test plans to accurately test the requirements. This also meant that team members that were familiar with the requirements knew the limitations of the tests, and could improve upon given feedback to strengthen the plan further.

   Jasmine: I think the team worked well together as usual, especially when it came to distributing tasks. Our team was very efficient and decided to split test cases for the functional and nonfunctional requirements in the same way we split up writing the requirements for the SRS, and the rest of the document was split up fairly quickly within a team meeting. Another thing that went well while writing this deliverable is organization. Now that we have written several pieces of documentation, the team is comfortable navigating LaTex formatting, as well as GitHub project and issue tracking.

   Promish: One thing that went well during this deliverable was how comfortable everyone became using GitHub. We created issues for each task, and handling merge conflicts was faster. As a team, we quickly divided the work, understanding that writing system tests for the requirements each of us specialized in was the most efficient approach.

We also recognized the importance of consistent naming for our tests to maintain traceability with our SRS, so we decided to include "ST" in the middle of our SRS requirements tags.

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

   Parisha: A challenge I found was due to time constraints and responsibilities in other classes it become very hard to coordinate meetings with all members of the team. In order to optimize our time, team members communicated progress consistently and we worked together in the end to merge branches (a lot of merge conflicts due to structure) and then reviewed the documentation against rubric and added reflection

   Mitchell: One of the pain points I experienced during this deliverable was understanding the format needed for the System Tests section. I found it difficult to get started because of the formality of the tests. As well, I wanted to make sure that the tests were consistent among team members, and a difference in formatting would be difficult for someone reading the document to understand. To resolve this, the team decided to follow a consistent formatting outline, and follow along a sample so that everyone knew what test cases should look like for this deliverable.

   Jasmine: One pain point during this deliverable was clarification of document instructions, such as what area testing is or what exactly the symbolic constraints were. We resolved this by preparing questions to ask the TA during our informal TA meeting, and looking at examples from other capstone projects from the same course completed in previous years.

   Promish: A challenge we faced during this deliverable was the large number of tests we had to write within a limited timeframe. With only a week to work on the document, we didn't have time for extra meetings with our advisor and TA to clarify certain sections as much as I would have liked. Overall, the team did a great job with the time and resources available, but some parts of the document were a bit unclear, especially regarding formatting and focus. The rubric was also quite general, making it harder to understand the exact requirements.

3. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage, Valgrind etc. You should look to identify at least one item for each team member.

   Parisha: One piece of knowledge and skills the team should acquire for the project is

a good understanding of all the different types of testing like static, dynamic, manual, automatic and functional testing. Understanding the difference between these types and what cases are the best to use them are important to design correct and efficient tests for the system.

Mitchell: One of the pieces of knowledge and skills the team collectively needs to acquire to successfully complete the verification and validation of the project is understanding how to properly perform usability testing. This will be important to the team because it is one of the project's chosen extras, so understanding usability testing is crucial for the success of the project. As well, it will inform us on how users actually engage, interact, and understand our system. Another skill the team will need to acquire is dynamic testing using different frameworks. It will be beneficial to perform dynamic testing to get verify each of our desired outcomes of tests against the actual outcomes. This will provide the team with accurate, reliable results and conclusions.

Jasmine: The team will need skills in both dynamic and static testing to successfully complete the verification and validation of the project, including knowledge of effective test case creation and debugging. Being familiar with tools like automated test frameworks and static analysis tools that our team can use for our project will be important to improving efficiency and accuracy in our testing processes. Additionally, understanding quality assurance practices and common code review techniques will help make sure our validation is thorough and reliable.

Promish: As a team, we'll need to focus on learning to use the frameworks outlined in section 3.6 to automate our unit tests. We'll also need to understand how to evaluate our SRS and the processes for making changes. Lastly, we'll need to know where testing fits into continuous development and how to follow Agile practices effectively.

4. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

   Parisha: One approach to acquire knowledge specially in manual vs automatic testing is using course material from our Software Testing course SFWRENG 3S03 to get a good summary of the different between the two and find some small examples/ practice problems to go through from tutorial to refresh our memory/ Another good approach would be to use testing modules online (such as Geeks4Geeks) to understand the different types of testing with good examples. I will be using approach 2 to acquire knowledge as I learn best with a combination of reading, watching and doing.

   Mitchell: One approach to acquire knowledge for usability testing is referring to SFWRENG 4HC3 course notes, as usability testing is one of the major topics the

course focuses on, with lots of examples and details. To obtain knowledge for dynamic tests, I plan on learning pytest to learn about dynamic testing in Python, which will be the team's language of choice for developing machine learning models.

Jasmine: I will choose to focus on static testing knowledge since it is something I usually overlook and would like to improve on. To acquire knowledge in static testing, one approach would be to use online static testing tutorials and resources, especially platforms with specialized online courses such as Coursera or Udacity. Another approach would be to use an open-source static analysis tool and apply it to previous projects or example codebases to get hands on experience identifying common code issues, warnings, and even security vulnerabilities.

Promish: One of the best ways to learn a framework is to read its documentation or watch YouTube tutorials. After that, testing it out in a separate IDE with a small example would be helpful for something like Pytest. For a better understanding of Agile development, we could look into resources that explain Agile principles and methods. Additionally, consulting our TA, supervisor, and professor would be valuable for understanding how our team can function effectively.