Verification and Validation Report: SFWRENG 4G06 Capstone Design Project

Team #18, InfiniView-AI
Anhao Jiao
Kehao Huang
Qianlin Chen
Qi Shu
Xunzhou Ye

March 6, 2024

1 Revision History

Date	Developer(s)	Change
Mar 6	AJ, QS, KH, XY, QC	Initial draft

2 Symbols, Abbreviations and Acronyms

Table 1 lists all the symbols, abbreviations, and acronyms used throughout this document.

Symbol	Description
Tai Chi	A classical Chinese martial art system practiced for health promotion and rehabilitation.
Instructor	A person who teaches a Tai Chi class through an online conference system.
Practitioner	A person who learns Tai Chi through an online conference system.
Pipeline	A sequence of processing elements connected in series, which are responsible for generating annotations.
UI	User Interface.
TA	Teaching Assistant.
SRS	The Software Requirements Specification document.
POC	proof of concept.
SFU	Selective Forwarding Unit, a component in real-time communication systems like WebRTC that routes and selectively forwards audio and video streams from one participant to others.
V&V Plan	The verification and validation plan document.
Stress test	Testing the limits (often the amount of data) of a system.
Performance test	Testing the performance(includes different metrics) of a system.
MIN_RES	720p

Table 1: List of symbols, abbreviations, and acronyms

Contents

List of Tables

List of Figures

This document ...

3 Functional Requirements Evaluation

FR-T1 **Initial State** Client application is running on the user's device, but the user didn't do any operations yet.

Input/Condition User clicks on the applicable identity(instructor/practitioner) button

Expected Output The live stream video window pops out on the user's screen.

Actual Output The live stream video window shows on the instructor's screen, it doesn't show on the practitioner's screen.

Result Pass

FR-T2 **Initial State** Application running on user's computer, and the user has clicked on "the instructor identity button" to indicate they are a TaiChi instructor. A window asking for permission to use the camera on the instructor's device popped out.

Input/Condition User allow/deny the webcam permission

Expected Output The webcam on the instructor's device is turned on

Actual Output The webcam on the instructor's device is turned on after allowing webcam permission, and the webcam is not turned on after denying permission.

Result Pass

FR-T3 **Initial State** both client applications and the server are running.

Input/Condition The user clicks on the applicable identity button to indicate they are an instructor or a practitioner.

Expected Output A log message indicates connection between the user's device and the server has been established.

Actual Output Log message confirms a successful connection between the user's device and the server for both instructor and practitioner.

Result Pass

FR-T4 **Type** Functional, Dynamic, Automated

Initial State The live stream Window for practitioners.

Input/Condition The user's device has established a connection with the server as a practitioner device.

Expected Output A request from the client device to the server for accessing the list of available annotation configuration.

Actual Output The client device sends a request to the server for the list of available annotation configurations.

Result Pass

FR-T5 **Initial State** The selectable list of the type of annotations is rendered on the user's screen.

Input/Condition Practitioner's selection on the list of types of annotations.

Expected Output A request(that reflects user's annotation selection) from the client device to the server for updating the annotation configuration, with a log indicating the request is sent.

Actual Output The client device sends the selected annotation configuration update to the server, and a log entry confirms the request was sent.

Result Pass

FR-T6 **Initial State** The system is running and actively connected to practitioners.

Input/Condition Practitioners initiate updates to annotation configurations.

Expected Output The system receives and processes the updated annotation configurations.

Actual Output The server gets the request and user sees the annotation they selected.

Result Pass

FR-T7 **Initial State** The server is running and actively receiving annotation configuration updates.

Input/Condition In a controlled test environment, the practitioner-client initiates the update of an annotation configuration. The update is sent to the server for processing.

Expected Output The expected result is that the server correctly processes the received annotation configuration from the practitioner-client.

Actual Output The server is not able to receive annotation configuration updates while the connection has been established.

Result Fail

FR-T8 **Initial State** The server has received and processed the annotation configuration.

Input/Condition The server uses the received annotation configuration to configure machine learning pipelines.

Expected Output The machine learning pipelines are arranged and configured based on the annotation configuration.

Actual Output The server successfully arranges and configures machine learning pipelines according to the received annotation configuration.

Result Pass

FR-T9 **Initial State** The machine learning pipelines are configured and active.

Input/Condition The instructor's video stream is processed with the annotation configuration.

Expected Output The instructor's video stream is rendered with annotations.

Actual Output The instructor's video stream is processed with the annotations.

Result Pass

FR-T10 **Initial State** The server is actively connected to practitioner clients.

Input/Condition The annotated video stream is generated and ready for transmission.

Expected Output The annotated video stream is transmitted to each practitioner-client through their established connections.

Actual Output The annotated video stream is transmitted to practitioner clients.

Result Pass

FR-T11 **Initial State** The signaling server is running.

Input/Condition Signaling requests for WebRTC connections are initiated.

Expected Output The signaling server consistently responds to requests and establishes WebRTC connections.

Actual Output The signaling server responds to most requests and establishes WebRTC connections

Result Pass

FR-T12 **Initial State** The client application is running, but the user didn't do any operations yet

Input/Condition The user joining video stream session.

Expected Output A button to identify if a user is an instructor or a practitioner is rendered.

Actual Output The button to identify a user as an instructor or a practitioner is rendered without issues

Result Pass

4 Nonfunctional Requirements Evaluation

NFR-T1 **Initial State** The system is in a typical operational state with all components and services running, including the user interface.

Input/Condition User interactions such as button clicks and menu selections.

Expected Output The user is able to interact with the client application and understand the response from and results yielded by the system.

Actual Output All 5 users are able to successfully interact with the application with understandable responses and results from the system.

Result Pass

NFR-T2 **Initial State** The system is in a typical operational state with all components and services running, including the user interface.

Input/Condition User interactions such as button clicks and menu selections.

Expected Output System response to user interactions.

Actual Output The system responds to user interactions promptly and correctly.

Result Pass

NFR-T3 **Initial State** The system is in a stable operational state with necessary services and components active. No users are currently logged in.

Input/Condition 5 test accounts (or more) for users. Predefined user actions/scripts (relevant to normal system interactions during peak periods).

Expected Output The system remains stable and responsive. All user transactions are processed successfully.

Actual Output The system remains stable and responsive with all user transactions processed successfully.

Result Pass

NFR-T4 **Initial State** The system is operational, with all services and components running. User accounts for test are set up, and no tasks are being performed.

Input/Condition Erroneous user input data. Improper user actions.

Expected Output The system detects and rejects invalid inputs or actions, providing clear and helpful error messages to the user. Transactions or actions are either rolled back safely or do not proceed until valid input is provided.

Actual Output No invalid inputs can be made within the system.

Result Pass

NFR-T6 **Initial State** The system, with the SFU component, is in a stable, operational state. Initially, there is a base number of video streams being processed, which is within the SFU's current capacity.

Input/Condition A load testing tool or custom script capable of simulating multiple, simultaneous video streams.

Expected Output The SFU successfully processes an increasing number of video streams.

Actual Output The SFU processes an increased number of video streams, which was up to 5 users without significant issues.

Result Pass

NFR-T7 **Initial State** This system is in a typical operational state with all components and services running, the user is in a live session.

Input/Condition Network Interruption/Network Resumption

Expected Output The system attempts to resume the previous session.

Actual Output The system did not resume to the previous session.

Result Fail

NFR-T8 **Initial State** This system is in a typical operational state with all components and services running, the user is in a live session.

Input/Condition Network instability

Expected Output A pop up window for network instability warning.

Actual Output There was no pop up window for network instability warning.

Result Fail

NFR-T9 **Initial State** This system is in a typical operational state with all components and services running.

Input/Condition Turn down the primary signaling server.

Expected Output The redundant server takes over until the primary server resumes.

Actual Output Connections were able to be established without the primary signaling server.

Result Pass

NFR-T10 **Initial State** The system is in a typical operational state with all components and services running. The user is in a live session with a media capturing device plugged in.

Input/Condition Disconnect the media capturing device from the user's computing device.

Expected Output A pop up window which warns the user that the client application has lost connection to the media capturing device and prompts the user to reconnect the device to resume the live session.

Actual Output When the media capturing device is disconnected, a pop-up window alerts the user and prompts for reconnection.

Result Pass

NFR-T11 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition Connect a media capturing device to the user's computing device.

Expected Output A pop up window which warns the user that the video capturing device does not meet required specification if the client application detects that the video input stream has a resolution lower than MIN_RES.

Actual Output The system was not able to check the specification of the capturing device

Result Fail

NFR-T12 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition the user starts a live session and begins streaming.

Expected Output Accurate instructional annotations are rendered and overlaid on the instructor's video stream.

Actual Output The system renders accurate instructional annotations on the instructor's video stream.

Result Pass

NFR-T13 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition Connect more than one media capturing device to the user's computing device.

Expected Output A dialog listing all the detected capturing devices is displayed to prompt the user for selecting a device to use.

Actual Output Feature handled by external API.

Result Pass

NFR-T14 **Initial State** This system is in a typical operational state with all components and services running, with a live session created.

Input/Condition The live video and audio stream from instructor client application.

Expected Output Audio stream and video stream with annotations on practitioner client application.

Actual Output The practitioner client application receives only video streams with annotations.

Result Fail

NFR-T15 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition The user starts the client application for instructors.

Expected Output A message with detailed instruction to set up a media capturing device and a list of cautions is displayed.

Actual Output No message was displayed.

Result Fail

NFR-T16 **Initial State** The software system is fully developed, stable, and ready for testing. The different test environments for the latest versions of Windows, Linux, and macOS are set up, each with default settings.

Input/Condition The latest stable versions of Windows, Linux, and macOS. Test cases designed to cover all the main functionalities of the system.

Expected Output The system functions correctly on all mentioned operating systems without crashes, unexpected behaviour, or significant performance issues.

Actual Output The system functions correctly on all mentioned operating systems without any significant issues.

Result Pass

NFR-T17 **Initial State** The system is fully developed, with all features operational, and is hosted in a stable environment accessible via the web. Test environments with the latest versions of Chrome, Firefox, Safari, and Edge are established, each configured with default browser settings.

Input/Condition Latest stable versions of Chrome, Firefox, Safari, and Edge. A series of test cases designed to fully evaluate the functionalities and features of the system.

Expected Output The system operates as intended on all tested web browsers without significant functionality issues, crashes, or severe performance degradation. Features and visual elements are consistent across all browsers.

Actual Output The system operates as intended across all tested web browsers with consistent features and visual elements.

Result Pass

NFR-T18 **Initial State** The system is fully developed and operational. The testing environments represent a range of standard personal computer and laptop configurations (covering various manufacturers, system specifications, and age of devices) equipped with cameras and, where applicable, microphones. The devices are running compatible operating systems with the necessary drivers installed.

Input/Condition Range of standard computers or laptops with various specifications, but all within the commonly accepted 'standard' range for current users. Devices have functional cameras and optional microphones. Test script detailing system operation tasks.

Expected Output The system operates without significant delays, errors, or crashes.

Actual Output The system operates without significant delays, errors, or crashes across a range of standard computers and laptops.

Result Pass

NFR-T19 **Initial State** The system is fully developed, with comprehensive documentation on its architecture, codebase, dependencies, and update procedures. The development environment is available for testing maintenance procedures, including tools for version control, testing, and deployment.

Expected Output Successful completion of maintenance tasks without introducing significant issues or disruptions to the system's operation.

Actual Output It is impossible to perform maintenance without interrupting live users.

Result Fail

NFR-T22 **Initial State** The system is in a stable state, ready for testing. The user interface for account creation or any other information input is accessible, and documentation related to data handling is available.

Input/Condition Guidelines or criteria specifying what constitutes "essential" versus "nonessential" information for the system's purpose. Access to the system's user interfaces (UI) where data submission forms are present.

Expected Output Comparison of requested data against the criteria for essential information.

Actual Output The system requests only essential information from users.

Result Pass

NFR-T23 **Initial State** This system is in a typical operational state with all components and services running.

Input/Condition Attempted unauthorized modifications to the system including access to system files and modifications to system configurations.

Expected Output A determination of whether the system are able to prevent access or modifications from unauthorized users.

Actual Output The system prevents all unauthorized access and modifications during the test.

Result Pass

NFR-T24 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition The user starts a live session and begins streaming.

Expected Output A prompt for granting access to use the media capturing devices is displayed.

Actual Output A prompt appears for granting access to media capturing devices during a live session.

Result Pass

NFR-T25 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition The user starts a live session and begins streaming.

Expected Output An indicator that the media capturing device is in use and that the user is being recorded is visible in the client application.

Actual Output An indicator is visible in the client application when the media capturing device is in use.

Result Pass

NFR-T26 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition The user ends a live session and stops streaming.

Expected Output The indicator that the media capturing device is in use disappears. The media stream from the capturing device is no longer displayed on the screen.

Actual Output The indicator disappears, and the media stream is no longer displayed once the session ends.

Result Pass

NFR-T27 **Initial State** This system is in a typical operational state with all components and services running.

Input/Condition User inputs

Expected Output System response to user inputs

Actual Output The system responds effectively to user inputs.

Result Pass

NFR-T28 **Initial State** The system is in a typical operational state with all components and services running.

Input/Condition A set of ISO/IEC 12207 standards.

Expected Output A determination of whether the system and its associated processes, documentation, and activities comply with the ISO/IEC 12207 standards.

Actual Output The system and its processes comply with ISO/IEC 12207 standards as determined through evaluation.

Result Pass

NFR-T29 **Initial State** This system is in a typical operational state with all components and services running.

Input/Condition Various user interactions.

Expected Output The computers continue to function within acceptable performance parameters throughout the test.

Actual Output The computer's function is not affected throughout various user interactions.

Result Pass

NFR-T30 **Initial State** This system is in a typical operational state with all components and services running.

Input/Condition User inputs

Expected Output System response from user inputs

Actual Output The system responds appropriately and efficiently to user inputs.

Result Pass

5 Comparison to Existing Implementation

Our project shares similarities with two existing types of implementations. The first is a project developed by a group of researchers, focused on annotating Tai Chi videos with detailed features such as skeletal outlines and semantic segmentation. Like their summer Tai Chi annotation initiative, our project also annotates videos. However, ours operates in real-time, unlike the Tai Chi project which processes annotations offline, making ours more adept for live online learning contexts.

Additionally, our project parallels certain aspects of video conferencing tools like Zoom, which offer live annotation capabilities. While Zoom boasts a broader set of features due to its maturity as a platform, our project distinguishes itself in several key areas:

It is designed for one-directional streaming, in contrast to Zoom's multi-directional video conferencing. Unlike Zoom, where annotations are generated client-side, our project renders annotations server-side, minimizing the hardware demands on viewers' devices. Our system allows viewers the flexibility to choose their annotations, diverging from Zoom's model where annotations are unified for all viewers as determined by the video sender. These distinctions underscore our project's unique contributions to video annotation technology, especially in educational and collaborative online environments.

6 Unit Testing

6.1 User Authentication Module

UT-UA1 **Test Name** test_renders_auth_component

Initial State The system initializes the Auth component within a mock routing environment.

Input/Condition N/A

Expected Output The component should display the Motion Mingle banner and a logo image.

Actual Output The component displayed the Motion Mingle banner and a logo image.

Result Pass

UT-UB2 **Test Name** test_instructor_button

Initial State The system initializes the "Auth" component within a mock routing environment.

Input/Condition The user clicks on the Instructor button.

Expected Output The user should be navigated to the '/instructor' path.

Actual Output The user was navigated to the '/instructor' path.

Result Pass

UT-UC3 **Test Name** test_practitioner_button

Initial State The system initializes the "Auth" component within a mock routing environment.

Input/Condition The user clicks on the Practitioner button.

Expected Output The user should be navigated to the '/practitioner' path.

Actual Output The user was navigated to the '/practitioner' path.

Result Pass

UT-UD4 **Test Name** test_slogan

Initial State The system initializes the "Auth" component within a mock routing environment.

Input/Condition N/A

Expected Output The slogan should be visible within the component with the correct text.

Actual Output The slogan was displayed within the component with the correct text.

Result Pass

6.2 Instructor View Module

UT-M1 **Test Name** test_renders_instructor_component

Initial State The system is initialized with default state settings. The "Instructor" component is about to be rendered within a mock routing environment.

Input/Condition N/A

Expected Output The "Instructor" component should be rendered with the title "Motion Mingle".

Actual Output The "Instructor" component was rendered with the title "Motion Mingle".

Result Pass

UT-M2 **Test Name** test_select_annotation

Initial State The system has loaded the Instructor component, and the "SelectAnnotation" component is ready to receive user input.

Input/Condition A user selects "Skeleton" annotation from the dropdown menu.

Expected Output The dropdown menu should reflect that "Annotation 1" has been selected.

Actual Output "Skeleton" has been selected from the dropdown menu.

Result Pass

UT-M3 **Test Name** test_stop_broadcast

Initial State The Instructor component is rendered and operational, presumed to be in a state where broadcasting is occurring (mocked condition).

Input/Condition The user clicks the "Broadcast" button to simulate starting a broadcast and then clicks "Stop" to simulate ending the broadcast.

Expected Output The "MessageModal" mock should be displayed, indicating that the stopping of the video is to be confirmed.

Actual Output The "MessageModal" mock was displayed.

Result Pass

UT-M4 **Test Name** test_start_self_video

Initial State The Instructor component is rendered and operational. The self-video is initially off.

Input/Condition The user clicks "Start self video" to initiate the video stream.

Expected Output The system should transition to a state where the self-video is active.

Actual Output The system transitioned to a state where the self-video is active.

Result Pass

UT-M5 **Test Name** test_close_self_video

Initial State The Instructor component is rendered and operational. The self-video is active.

Input/Condition The user clicks "Close self video".

Expected Output The system should transition to a state where the self-video is inactive.

Actual Output The system transitioned to a state where the self-video is inactive.

Result Pass

6.3 Practitioner View Module

UT-PV1 **Test Name** test_renders_practitioner_component

Initial State The system is initialized with default state settings. The "Practitioner" component is about to be rendered within a mock routing environment.

Input/Condition N/A

Expected Output The "Practitioner" component should be rendered with the title "Motion Mingle", a dropdown for selecting annotations initialized to "None", and a "Connect" button.

Actual Output The "Practitioner" component was rendered with the title "Motion Mingle".

Result Pass

UT-PV2 **Test Name** test_select_annotation

Initial State The "Practitioner" component is rendered and displayed with the default selected annotation as "None".

Input/Condition The user changes the annotation selection from "None" to "Skeleton" using the dropdown.

Expected Output The dropdown menu for selecting annotations should reflect the change by displaying "Skeleton" as the current selection.

Actual Output "Skeleton" was displayed as selected in the dropdown menu.

Result Pass

UT-PV3 **Test Name** test_start_video_stream

Initial State The "Practitioner" component is rendered, displaying the "Connect" button while the video stream is not connected.

Input/Condition The "Connect" button was clicked to start the video stream.

Expected Output The "Connect" button should change to "Disconnect".

Actual Output The "Connect" button changed to "Disconnect".

Result Pass

UT-PV4 **Test Name** test_stop_video_stream

Initial State The "Practitioner" component is rendered, displaying the "Disconnect" button while the video stream is connected.

Input/Condition The "Disconnect" button was clicked to stop the video stream.

Expected Output The "Disconnect" button should change to "Connect".

Actual Output The "Disconnect" button changed to "Connect".

Result Pass

6.4 Annotation Configuration Module

UT-AC1 **Test Name** test_render_select_annotation_component

Initial State The "SelectAnnotation" component is initialized with default state settings.

Input/Condition The component is rendered without any preselected value.

Expected Output The component should display with the label "Annotation" and a dropdown button showing the default value "None".

Actual Output The component was rendered with the label "Annotation" and a dropdown button showing the default value "None".

Result Pass

UT-BC2 **Test Name** test_annotation_options

Initial State The "SelectAnnotation" component is rendered with default settings.

Input/Condition The user clicks on the dropdown menu to view available options.

Expected Output The dropdown menu should display four options: "None", "Skeleton", "Edges", and "Cartoon".

Actual Output The dropdown menu displayed four options: "None", "Skeleton", "Edges", and "Cartoon".

Result Pass

6.5 RTC Control Module

UT-RC1 **Test Name** test_create_peer_connection

Initial State Before creating a new "RTCPeerConnection", no connection exists.

Input/Condition The "createPeerConnection" function is called without any specific arguments since it uses predefined settings within its implementation.

Expected Output A new "RTCPeerConnection" should be created with the configuration specified for unified-plan SDP semantics and the Google STUN server.

Actual Output A new "RTCPeerConnection" was created with the configuration specified for unified-plan SDP semantics and the Google STUN server.

Result Pass

UT-RC2 **Test Name** test_connect_as_consumer

Initial State A "RTCPeerConnection" instance exists but has not yet established a connection nor created any offers.

Input/Condition The function "connectAsConsumer" is called with the created peer connection and "annotation" as arguments.

Expected Output The function should asynchronously create an SDP offer, set it as the local description, send it to the server, and set the received SDP answer as the remote description.

Actual Output Same as expected output.

Result Pass

UT-RC3 **Test Name** test_connect_as_broadcaster

Initial State A "RTCPeerConnection" instance is ready but not yet in a broadcasting state.

Input/Condition The function "connectAsBroadcaster" is called with the peer connection object.

Expected Output The function should asynchronously create an SDP offer for broadcasting, set it as the local description, send it to the server, and set the server's SDP answer as the remote description.

Actual Output Same as expected.

Result Pass

6.6 Other Modules

UT-OT1 **Test Name** test_display_message

Initial State The system initializes the "NotFound" component within a mock routing environment.

Input/Condition The component is rendered without any user interaction.

Expected Output The component should display the "Not Found" message.

Actual Output The component displayed the "Not Found" message.

Result Pass

UT-OT2 **Test Name** test_navigate_to_root

Initial State The system initializes the "NotFound" component within a mock routing environment.

Input/Condition The component is rendered, and the existence of a link is checked.

Expected Output There should be a link that reads "GO HOME" and navigates to the root path '/' upon clicking.

Actual Output Same as expected.

Result Pass

UT-OT3 **Test Name** test_close_message_modal

Initial State The "MessageModal" component is rendered with "isModalOpen" set to true.

Input/Condition The "Cancel" button was clicked.

Expected Output The "handleClose" mock function should be called once.

Actual Output The "handleClose" mock function was called once.

Result Pass

UT-OT4 **Test Name** test_stop_video

Initial State The "MessageModal" component is rendered with "isModalOpen" set to true.

Input/Condition The "Stop Video" button was clicked.

Expected Output The "handelStopVideo" mock function should be called once.

Actual Output The "handelStopVideo" mock function was called once.

Result Pass

UT-OT5 **Test Name** test_message_visible

Initial State The "MessageModal" component is initialized and rendered with "isModalOpen" set to true.

Input/Condition N/A

Expected Output The modal, including the warning message, should be visible.

Actual Output The modal, including the warning message, was visible.

Result Pass

UT-OT6 **Test Name** test_message_invisible

Initial State The "MessageModal" component is initialized and rendered with "isModalOpen" set to false.

Input/Condition N/A

Expected Output None of the modal's content should be present in the document.

Actual Output None of the modal's content was present in the document.

Result Pass

6.7 SFU Server Module

UT-S1 **Test Name** test_broadcast_options

Initial State Server is initialized and running with /broadcast endpoint available.

Input/Condition HTTP OPTIONS request to /broadcast.

Expected Output HTTP status code 200.

Actual Output HTTP status code 200.

Result Pass

UT-S2 **Test Name** test_broadcast_post

Initial State Server is initialized with the /broadcast endpoint ready to accept POST requests.

Input/Condition POST request to /broadcast with JSON body containing "sdp" and "type" fields.

Expected Output HTTP status code 200 and a JSON response containing "sdp" and "type".

Actual Output HTTP status code 200 and a JSON response containing "sdp" and "type".

Result Pass

UT-S3 **Test Name** test_broadcast_empty_body_post

Initial State Server is initialized with the /broadcast endpoint ready to accept POST requests.

Input/Condition POST request to /broadcast with an empty JSON body.

Expected Output HTTP status code 400

Actual Output HTTP status code 400

Result Pass

UT-S4 **Test Name** test_broadcast_post_with_invalid_data_types

Initial State Server is initialized with the /broadcast endpoint ready.

Input/Condition POST request to /broadcast with invalid data types for "sdp" and "type".

Expected Output HTTP status code 400

Actual Output HTTP status code 400

Result Pass

UT-S5 **Test Name** test_broadcast_post_missing_fields

Initial State Server is initialized with the /broadcast endpoint ready.

Input/Condition POST request to /broadcast with missing "type" field in the JSON body.

Expected Output HTTP status code 400

Actual Output HTTP status code 400

Result Pass

UT-S6 **Test Name** test_invalid_content_type

Initial State Server is initialized and ready to handle requests.

Input/Condition POST request to /broadcast with a "Content-Type" of "text/plain".

Expected Output HTTP status code 500

Actual Output HTTP status code 500

Result Pass

UT-S7 **Test Name** test_method_not_allowed

Initial State Server is initialized with the /broadcast endpoint not configured to accept GET requests.

Input/Condition GET request to /broadcast.

Expected Output HTTP status code 405

Actual Output HTTP status code 405

Result Pass

UT-S8 **Test Name** test_response_content_type

Initial State Server is initialized and ready to respond to POST requests.

Input/Condition HTTP status code 200 and the "Content-Type" header is "application/json; charset=utf-8"

Actual Output HTTP status code 200 and the "Content-Type" header is "application/json; charset=utf-8"

Result Pass

UT-S9 **Test Name** test_consume_options

Initial State Server is initialized and running with /consumer endpoint available.

Input/Condition HTTP OPTIONS request to /consumer.

Expected Output HTTP status code 200

Actual Output HTTP status code 200

Result Pass

UT-S10 **Test Name** test_consume_post

Initial State Server is initialized with the /consumer endpoint ready to accept POST requests.

Input/Condition POST request to /consumer with JSON body containing "sdp", "type", and "video_transform".

Expected Output HTTP status code 500

Actual Output HTTP status code 500

Result Pass

UT-S11 **Test Name** test_consume_empty_body_post

Initial State Server is initialized with the /consumer endpoint ready.

Input/Condition POST request to /consumer with an empty JSON body.

Expected Output HTTP status code 400

Actual Output HTTP status code 400

Result Pass

UT-S12 **Test Name** test_consume_post_unsupported_video_transform

Initial State Server is initialized with the /consumer endpoint ready.

Input/Condition POST request to /consumer with unsupported "video_transform" value.

Expected Output HTTP status code 400

Actual Output HTTP status code 400

Result Pass

6.8 Video Transform Module

Unit testing the video transform module is particularly challenging due to the nature of its operations which involve real-time video stream manipulation using complex image processing and machine learning models. These transformations, such as pose estimation and segmentation, rely on the input data's visual content, which can have near-infinite variability and require a visual context to determine correctness. The output is highly dependent on the specific content of the video frame being processed and the environmental conditions at the time of capture. Furthermore, such processes are typically non-deterministic due to the internal state management within the machine learning models (like MediaPipe's pose detection) and can be affected by the hardware acceleration capabilities of the environment where the tests are run.

Moreover, the transformations performed on video frames are designed to yield results that are subjectively evaluated visually, rather than through quantitative measures that could be programmatically asserted. While some aspects of the module could be unit tested by mocking dependencies and checking if the correct methods are called with expected arguments, the actual verification of the transformation's quality requires manual inspection or more sophisticated automated image comparison techniques that go beyond the scope of traditional unit testing. This inherent complexity and the need for a subjective assessment make the creation of automated unit tests for the video transform module both non-trivial and possibly less effective for ensuring the correctness of visual outputs.

7 Changes Due to Testing

7.1 Change due to supervisor's feedback

Andrew Mitchell, who is one of the supervisors of this project, has pointed out during the Revision 0 demo that the "check annotated video" button on the instructor page

should appear after the instructor has started broadcasting instead of showing it directly, as clicking the check annotated video can't show anything without input video stream. To follow the feedback from Andrew, the UI design will be changed so that the "check annotated video" button only shows up after the instructor starts broadcasting, and extra instructions will be added to the side of the instructor and practitioner view page.

7.2 Change due to reviewing the question in the usability survey

The VnV Plan, states that our group will review the questions in the usability survey as part of non-functional testing. One of the questions in the usability survey is: "Are there any specific content or learning materials you would like to see added to the application?" Some users state it is not convenient that they need to manually disconnect and reconnect to the server when selecting different annotations as a practitioner. The team decided to add an "auto-refresh" functionality to the system when an instructor or a practitioner changes their selected annotation would make the UI more intuitive and enhance user experiences.

7.3 Change due to failed tests

The VnV testing highlighted critical failures in server update processing and network resilience. Specifically, test case FR-T7 failed because the server could not receive annotation configuration updates while the connection was established. Additionally, the system did not handle network interruptions and resumptions as expected in test case NFR-T7, and it also failed to provide audio in the live video and audio streams in test case NFR-T14.

In response to these failures, the team will prioritize the development of an 'auto-reconnect' feature that ensures the system maintains or promptly re-establishes its connection to allow continuous update processing. This functionality will be designed to automatically resume the session and process updates without user intervention in case of network disruptions.

For the audio issue identified in test case NFR-T14, a feature addition is required. The team will integrate a reliable audio streaming capability to ensure both audio and video streams are consistently received and properly annotated in the practitioner client application.

The necessity of displaying instructional messages as indicated by the failure of test case NFR-T15 will also be addressed. The system will be updated to show de-

tailed instructions and cautions when starting the client application, thereby avoiding confusion and potential misuse.

These enhancements are crucial for improving the robustness, user experience, and functional completeness of the system. The adjustments made from these test failures will be documented and incorporated into the VnV Plan for future reference and to guide subsequent development cycles. The team will establish a rigorous testing procedure for these new features to ensure their effectiveness before deployment.

8 Automated Testing

8.1 Github Action

Github Action provides a seamless and efficient way to ensure code integrity with each push. Tests are executed automatically on the latest commits to the main and develop branches, verifying that new changes do not break existing functionality.

8.2 Pytest

PyTest is also used for the automatic testing. It is a robust and feature-rich Python library that streamlines the creation of test cases. By design, PyTest simplifies test authoring, allowing developers to write test functions that are concise and readable. It leverages the straightforward "assert" statement to check for expected outcomes, making test assertions both natural and easy to understand. Furthermore, PyTest's powerful parameterization capability, facilitated by the pytest.mark.parametrize decorator, permits the execution of a single test function against a variety of input sets. This not only enhances test coverage but also significantly boosts efficiency, as it allows for extensive testing without the need to write additional test cases. Overall, Pytest's intuitive approach and its ability to facilitate detailed and comprehensive testing with minimal code make it an invaluable tool for automated testing, contributing to faster development cycles and more reliable software.

9 Trace to Requirements

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11	FR12	FR13	FR14	FR15	FR16
FR-T1	X	X														
FR-T2			X													X
FR-T3				X	X											
FR-T4						X									X	
FR-T5								X								
FR-T6							X									
FR-T7									X							
FR-T8										X						
FR-T9											X					
FR-T10												X				
FR-T11													X			
FR-T12														X		

Table 2: Traceability matrix showing the connections between test cases and functional requirements

	LF1	UH1	UH2	PR1	PR2	PR3	PR4	PR5	PR6	PR7	PR8	PR9	PR10	PR11	PR12	PR13
NFR-T1	X	X	X													
NFR-T2				X	X											
NFR-T3					X	X										
NFR-T4							X									
NFR-T6									X							
NFR-T7										X						
NFR-T8											X					
NFR-T9												X				
NFR-T10													X			
NFR-T11														X		
NFR-T12															X	
NFR-T13																X

Table 3: Traceability matrix showing the connections between test cases and non-functional requirements

	PR14	PR15	OE1	OE2	OE3	MS1	MS2	SR1	SR2	SR3	SR4	SR5	SR6	CR1	LR1	HS1	HS2
NFR-T14	X																
NFR-T15		X															
NFR-T16			X														
NFR-T17				X													
NFR-T18					X												
NFR-T19						X											
NFR-T22									X								
NFR-T23										X							
NFR-T24											X						
NFR-T25												X					
NFR-T26													X				
NFR-T27														X			
NFR-T28															X		
NFR-T29																X	
NFR-T30																	X

Table 4: Traceability matrix showing the connections between test cases and non-functional requirements continued

10 Trace to Modules

11 Code Coverage Metrics

Frontend Code Coverage

Module	%Statements	%Branches	%Functions	%Lines
User Authentication Module	100	100	100	100
Instructor View Module	43.22	33.33	42.84	43.22
Practitioner View Module	48.14	45	50	48.14
RTC Control Module	44.44	100	100	44.44
Annotation Configuration Module	100	100	100	100
Other Modules	100	100	100	100
Total	46.62	45	49.17	46.62

Table 5: Frontend Code coverage report

Backend Code Coverage

Module	statements	missing	excluded	coverage
SFU Server Module	118	20	0	83%
Video Transform Module	86	51	0	41%
Total	204	71	0	65%

Table 6: Backend Code coverage report

The low code coverage for SFU Server, Instructor View Module, Practitioner View Module and RTC Control Module can be attributed to the inherent complexity of testing asynchronous operations and external dependencies, such as WebRTC interactions(RTCPeerConnection and MediaStreamTrack), which are difficult to simulate in unit tests.

For the VideoTransformTrack module, the low coverage is due to the challenges in automatically testing image processing and machine learning functionalities that require subjective visual verification and can exhibit non-deterministic behavior.

Appendix A — Reflection

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

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

(a) One change we made to the VnV plan was in selecting the group for our usability survey. Initially, we intended to engage a focus group of Tai Chi learners to evaluate our system and provide feedback. However, due to our supervisor's research leave and the difficulty in assembling the specified focus group, we shifted our approach. Instead, we opted for a diverse group of test users, including those without Tai Chi experience, a PhD student conducting Tai Chi-related research, and senior citizens, to gain insights into the system's user experience. They were also able to provide valuable feedback on the system user experiences. In the future, we will make sure to conduct sessions to connect with our target user group early in the development phase.

(b) Another change we made to the VnV plan was we removed some test cases due to the evolving project changes. The differences between the VnV plan and the test cases are mainly due to the dynamic nature of software development and the challenges that come with it. In the future, we will make sure to conduct regular reviews of the VnV plan and update it as needed to reflect changes in the project scope, requirements, and constraints.

AppendixB — Usability Survey

The usability survey to MotionMingle can be found here.