

# Verification and Validation Report: Software Engineering

Team 15, ASLingo

Andrew Kil

Cassidy Baldin

Edward Zhuang

Jeremy Langner

Stanley Chan

March 8, 2024

# 1 Revision History

Date	Contributors	Notes
Feb 29, 2024	Cassidy	Initial draft and formatting
Mar 4, 2024	Andrew	Filled out table for learning progression
Mar 4, 2024	Stanley	Added test results for performance requirements
Mar 4, 2024	Cassidy	Added test results for tables 2,4,5 and 6
Mar 5, 2024	Stanley	Added test to modules traceability matrix
Mar 6, 2024	Andrew	Added automated testing information
Mar 6, 2024	Cassidy	Added reflection, fixed table 6, added to automated testing section
Mar 7, 2024	Edward	Added unit testing section

## 2 Symbols, Abbreviations and Acronyms

Table 1: Naming Conventions and Terminology

Term, Abbreviation, or Acronym	Description
A	Shorthand for Assumption
ASL	Shorthand for American Sign Language. It is a form of sign language primarily used in the US and in parts of Canada
ASLingo	The commercial name for the project
CV	Refers to Computer Vision, the field of technology that involves processing visual input to achieve various means.
CR	Shorthand for 'Cultural Requirements', a subsection of Non-Functional Requirements.
HSR	Shorthand for 'Health and Safety Requirements', a subsection of Non-Functional Requirements.
FR	Shorthand for Functional Requirements
LR	Shorthand for 'Legal Requirements', a subsection of Non-Functional Requirements.
LFR	Shorthand for 'Look and Feel Requirements', a subsection of Non-Functional Requirements.
MSR	Shorthand for 'Maintainability and Support Requirements', a subsection of Non-Functional Requirements.
OER	Shorthand for 'Operational and Environmental Requirements', a subsection of Non-Functional Requirements.

OpenCV	Refers to the Open Computer Vision Library library available for free to developers in order to develop Computer Vision applications.
PR	Shorthand for 'Performance Requirements', a subsection of Non-Functional Requirements.
SR	Shorthand for 'Security Requirements', a subsection of Non-Functional Requirements.
UHR	Shorthand for 'Usability and Humanity Requirements', a subsection of Non-Functional Requirements.

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>General Information</b>	<b>1</b>
3.1	Summary . . . . .	1
3.2	Objectives . . . . .	1
3.3	Relevant Documentation . . . . .	1
<b>4</b>	<b>Functional Requirements Evaluation</b>	<b>2</b>
4.1	System Tests for Authentication . . . . .	2
4.2	System Tests for ASL Learning Progression . . . . .	3
4.3	System Tests for Web Application . . . . .	4
4.4	System Tests for Hardware . . . . .	4
<b>5</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>6</b>
5.1	System Tests for Usability . . . . .	6
5.2	System Tests for Performance . . . . .	7
<b>6</b>	<b>Unit Testing</b>	<b>9</b>
6.1	Web Socket Unit Tests . . . . .	9
<b>7</b>	<b>Changes Due to Testing</b>	<b>9</b>
<b>8</b>	<b>Automated Testing</b>	<b>10</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>10</b>
<b>10</b>	<b>Trace to Modules</b>	<b>12</b>
<b>11</b>	<b>Appendix</b>	<b>13</b>
11.1	Usability Survey Questions . . . . .	13
11.2	Appendix — Reflection . . . . .	13

## List of Tables

1	Naming Conventions and Terminology . . . . .	ii
---	--	----

2	System Tests for Authentication . . . . .	2
3	System Tests for ASL Learning Progression . . . . .	3
4	System Tests for Web Application . . . . .	4
5	System Tests for Hardware . . . . .	5
6	System Tests for Usability . . . . .	6
7	System Tests for Performance . . . . .	7
8	System Tests for Performance . . . . .	9

## 3 General Information

### 3.1 Summary

As a machine learning-based image recognition web app, ASLingo has many areas to be tested. The overall software will be broken down into modules. There will be a front-end, a back-end, a database, and a machine learning model which all need to be separately tested, along with physical hardware and compatibility. This document serves as a report of the testing done to ensure that this system has been properly and thoroughly tested to meet the requirements set by the [Software Requirements Specification](#).

### 3.2 Objectives

This document aims to outline the testing plan for ASLingo in order to create a functional and reliable product for users that aligns with the specified requirements. The team seeks to build confidence in stakeholders and users that the software is correct and meets or exceeds the initial intended goals, resulting in an overall satisfactory user experience.

### 3.3 Relevant Documentation

Below is a list of the relevant documentation referenced within the Verification and Validation Plan.

The [Development Plan](#) outlines the roles of each team member and the areas that each member will focus on. This breakdown of team responsibilities allows the team to assign testing roles accordingly. This document also contains the tools that the team plans on using for testing.

The [VnV Plan](#) outlines the testing plan for the system, as well as outlining the test cases that the team will perform to ensure the project has been properly and thoroughly tested. This document also contains the tools that the team plans on using for testing.

The [Software Requirements Specification](#) lists the functional and non-functional requirements which will aid in testing by formulating a testing plan to meet each requirement. Non-functional requirements should be tested such that

the fit criteria are met.

The [Hazard Analysis](#) identifies failure modes to determine the implementation strategies to mitigate them. These will be used as a part of the testing plan to ensure that the failures are covered.

The [Module Guide](#) divides the software into modules. The team will build the testing plan around the modules.

The [Module Interface Specification](#) further decomposes the software's modules into specific access routines. The team will build the testing plan such that each function and routine works as intended.

## 4 Functional Requirements Evaluation

### 4.1 System Tests for Authentication

Many of these tests fails since the full implementation of the Authentication module has not been fully completed yet. This will be changed before Rev1 to ensure full functionality of the system and all tests should pass.

Table 2: System Tests for Authentication

Test ID	Description	Input	Expected Output	Actual Output	Result	Req ID
FRT1-A1	User can make their account	User inputs username and password, then selects 'Sign In'	User successfully makes their account	User is shown successful account creation page	Pass	FR3



FRT1-A2	User can sign into account made previously	Previous user inputs username and password	User successfully signs into their account	User is shown successful account creation page	Fail	FR4
FRT1-A3	User can sign out of their account	User selects 'Logout' header	User signs out of account and is taken back to home page	None	Fail	FR5
FRT1-A4	User can reset password	User requests reset password then inputs email	System changes stored password to new password	None	Fail	FR13

## 4.2 System Tests for ASL Learning Progression

Many of the learning progression tests are redundant, as the development team and some participating users tested multiple ASL signs using the applications 'Quiz' and 'Practice' sections. All letters of the alphabet were tested to ensure correctness of the system, and all quiz levels were tested to ensure their functionality.

Table 3: System Tests for ASL Learning Progression

Test ID	Description	Input	Expected Output	Actual Output	Result	Req ID
FRT2-LP1	User performs ASL signs	Alphabetical signs 'a', 'b', 'c'	The letters 'a', 'b', 'c' are correct from user	The letters 'a', 'b', 'c' are shown by user	Pass	FR2

FRT2-LP2	Complete diagnostic quiz	User goes to quiz page	System starts diagnostic quiz until user completes it	Quiz is started and completed by user	Pass	FR6
FRT2-LP3	User attempts progression based course	User completes their diagnostic quiz	System generates new quiz for user based on results	No new course is generated	Fail	FR7
FRT2-LP4	User tracks their progression	User goes to progression tab	Views their progress	None	Fail	FR7, FR12
FRT2-LP5	System Saved User Progress	User completes a quiz	Result is stored and saved	Not saved	Fail	FR8
FRT2-LP6	Get live feedback from user signs	User signs letter 'a'	Systems outputs letter 'A'	System shows user letter 'A'	Pass	FR10

### 4.3 System Tests for Web Application

Table 4: System Tests for Web Application

Test ID	Description	Input	Expected Output	Actual Output	Result	Req ID
FRT3-U1	User can access the web application	User opens web browser with the ASLingo application	User is able to access all pages of the application	User is able to access all pages of the application	Pass	FR9

### 4.4 System Tests for Hardware

Table 5: System Tests for Hardware

<b>Test ID</b>	<b>Description</b>	<b>Input</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result</b>	<b>Req ID</b>
FRT4-HW1	User is able to access the web camera	User goes to quiz page and starts a quiz	System is able to recognize user signs from their camera	System is able to recognize user signs from their camera	Pass	FR1
FRT4-HW2	Monitor web camera usability	User goes to quiz page and starts a quiz	User is notified if their camera is not working correctly	User is notified if their camera is not working correctly	Pass	FR11

## 5 Nonfunctional Requirements Evaluation

### 5.1 System Tests for Usability

We will be testing our usability requirements using a survey for a group of testers to fill out after using the application for 15 minutes. The group of users will have an interest in learning ASL, and will be willing to fill out this questionnaire to give some perspective on the usability of our application. The list of survey questions and the link to the survey can be found in the Appendix [11.1](#).

Table 6: System Tests for Usability

Test ID	Description	Input	Expected Output	Actual Output	Result	Req ID
NFRT1-UT1	User is able to start application with no training	User opens web application	User is able to use application, completes question 1 of survey with result over 75%	User is able to use application, do not have a large enough sample of users yet	Fail	UHR1
NFRT1-UT2	User is able to complete a quiz with no training	User opens quiz page and starts a quiz	User is able to complete a quiz, completes question 3 of survey with result over 75%	User is able to complete a quiz, do not have a large enough sample of users yet	Fail	UHR1

NFRT1-UT3	User is able to use application with various hearing abilities	User opens application	User is able to use application, completes question 2 of survey with result over 75%	User is able to use application, do not have a large enough sample of users yet	Fail	UHR2
NFRT1-UT4	User can personalize their account	User is signed into their account	User can change some personal settings	None	Fail	UHR3
NFRT1-UT5	System should show user if input needs to be adjusted	User tries to complete quiz but their camera is not set up properly	System prompts user to fix camera settings	System prompts user to fix camera settings	Pass	UHR4

## 5.2 System Tests for Performance

Table 7: System Tests for Performance

Test ID	Description	Input	Expected Output	Result	Req ID	
NFRT2-PT1	The application should respond to user input within 1 second.	The user should respond to the application's prompt.	The system should register the user's input and respond to the user quickly.	The system responded with the detected sign almost instantly.	Pass	PR1

NFRT2-PT2	The application should be able to accurately determine if the user has signed the correct response to the prompt 95% of the time.	The user should sign in response to the application's prompt.	The application should register the user's signed input and determine if they have signed the required action correctly with 95% overall accuracy.	Static hand signs are recognized with a total testing accuracy of around 98%. Dynamic hand signs are inconsistent, with the accuracy at around 50 - 60%.	Fail	PR2
-----------	---	---	--	--	------	-----

## 6 Unit Testing

### 6.1 Web Socket Unit Tests

Table 8: System Tests for Performance

Test ID	Test Name	Input	Expected Output	Result	Req ID
UT-WS1	test_dynamic_event	The system initializes a mock client and it sends dynamic hand sign landmark history back to the system.	The system should be able to successfully process the data and return the sign the data corresponds to.	Fail	
UT-WS2	test_stream_event	The system initializes a mock client and it sends landmark features back to the system.	The system should be able to successfully process the data and return the sign the data corresponds to.	Fail	

## 7 Changes Due to Testing

- The main change for final product will be to ensure user account functionality is working to allow users to grow and learn within the ap-

plication. This includes creating an account, signing in - out, having personalized learning, and progression opportunities.

- The second change due to testing involves ensuring efficiency of sign recognition is held to a high standard from NFRT2 which outlines having a 95% accuracy of determining the user's hand sign.
- From our user testing, we also want to ensure that new and existing users of our application can get the best learning experience possible through a responsive, well designed and tested application.

## 8 Automated Testing

Automated testing is taken care of by the automatic linter flake8 upon every push to the repository to ensure that our python code is in line with the styling guide of flake8. We are also using a local and automatic linter eslint for our Javascript code for the front end of our application. Currently a couple files are failing to meet the coding standard but these will be taken of upon Rev1.

## 9 Trace to Requirements

### Functional Requirements to System Tests



System Test	FR Req.												
	1	2	3	4	5	6	7	8	9	10	11	12	13
FRT1-A1			X										
FRT1-A2				X									
FRT1-A3					X								
FRT1-A4													X
FRT2-LP1		X											
FRT2-LP2						X							
FRT2-LP3							X						
FRT2-LP4							X					X	
FRT2-LP5								X					
FRT2-LP6									X				
FRT3-U1									X				
FRT4-HW1	X												
FRT4-HW2											X		

System Test	UHR				PR	
	1	2	3	4	1	2
NFRT1-UT1	X					
NFRT1-UT2	X					
NFRT1-UT3		X				
NFRT1-UT4			X			
NFRT1-UT5				X		
NFRT2-PT1					X	
NFRT2-PT2						X

#### Non Functional Requirements to System Tests

## 10 Trace to Modules

System Test	Module												
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13
FRT1-A1			X									X	X
FRT1-A2				X								X	X
FRT1-A3												X	X
FRT1-A4												X	X
FRT2-LP1	X	X	X			X		X					
FRT2-LP2									X	X			X
FRT2-LP3										X			X
FRT2-LP4										X			X
FRT2-LP5													X
FRT2-LP6	X	X	X							X			
FRT3-U1									X				
FRT4-HW1								X					
FRT4-HW2								X		X			
NFRT1-UT1									X	X	X	X	X
NFRT1-UT2										X	X		
NFRT1-UT3									X	X	X		
NFRT1-UT4												X	X
NFRT1-UT5	X	X	X					X					
NFRT2-PT1	X	X	X	X	X		X						
NFRT2-PT2	X	X	X	X	X					X			

### System Tests to Modules

## 11 Appendix

### 11.1 Usability Survey Questions

A link to the survey that participants will be given can be found [here](#). Participants will be asked to rank how they felt about the following statements, with the response options being Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree.

1. It was very easy to get right into a testing session with little to no hassle.
2. The User Interface is very friendly and it is easy to identify where everything is.
3. During a Quiz, its very easy to understand what to do and how to complete it.
4. While signing, it is very easy to see what sign I am making and whether to make adjustments or not.
5. At my current level of ASL knowledge, it is easy to use the application
6. On a scale of 1 to 10, how would you rate your experience with ASLingo?  
[ 1 = terrible, 10 = fantastic]

### 11.2 Appendix — Reflection

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

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

The Verification and Validation (VnV) plan is different than the tests that were actually conducted for the VnV Report in many ways. One main difference was that our team wrote more tests than the tests that were in the original VnV plan, and we also performed even more tests than the ones that were written. This is because when we wrote the VnV plan in November, we only had the proof of concept version of our project completed, so the tests that were written were for an earlier version of the project, or an ideal version of our project. While working on the project for our Rev0 demo, the team was constantly testing both the front and back end to ensure that the required functionality was working according to the specifications lead out in the SRS document. This testing was in line with some of the new test cases that were added to the VnV plan, but many were redundant and are not shown in the final report or plan (such as testing each letter of the alphabet individually multiple times with different people and in different environments) to reduce the length of the report. We think that some of these changes could be anticipated in future projects if a lot more time was given to all the intricacies of what the final project would look and perform like, but some changes to things like the usability of the project you can only really know after doing testing with users and other stakeholders and are harder to predict from a glance. There may be more changes to the VnV plan and report while the team works on the final product demo and as a result of more testing from stakeholders as well, which will be updated in this document over time.