Verification and Validation Report: Software Engineering

Team 15, ASLingo
Andrew Kil
Cassidy Baldin
Edward Zhuang
Jeremy Langner
Stanley Chan

April 4, 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
Apr 4, 2024	Jeremy and Cassidy	Updated system test results and tables according to VnVPlan

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

List of Tables

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 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 2: System Tests for ASL Learning Progression

Test ID	**Description**	**Input**	**Expected Output**	**Actual Output**	**Result**	**Req ID**
FRT1-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	FR1, FR2, FR4
FRT1-LP2	Complete quiz	User goes to quiz page	System starts diagnostic quiz until user completes it	Quiz is started and completed by user	Pass	FR5, FR6, FR7

FRT1-LP3	Get live feedback from user signs	User signs letter 'a'	Systems outputs letter 'A'	System shows user letter 'A'	Pass	FR1, FR2, FR3

4.2 System Tests for Web Application

Table 3: 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	FR1

4.3 System Tests for Hardware

Table 4: System Tests for Hardware

Test ID	**Description**	**Input**	**Expected Output**	**Actual Output**	**Result**	**Req ID**
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	FR1

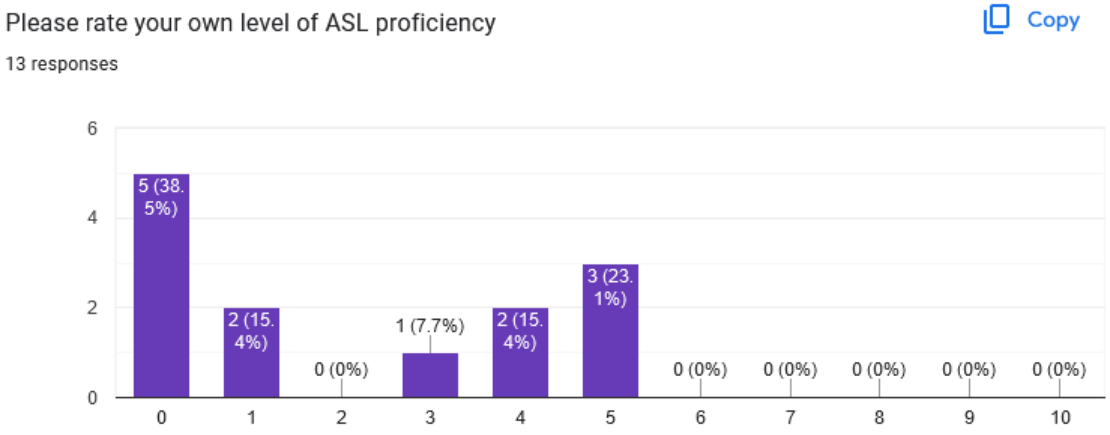
5 Nonfunctional Requirements Evaluation

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.

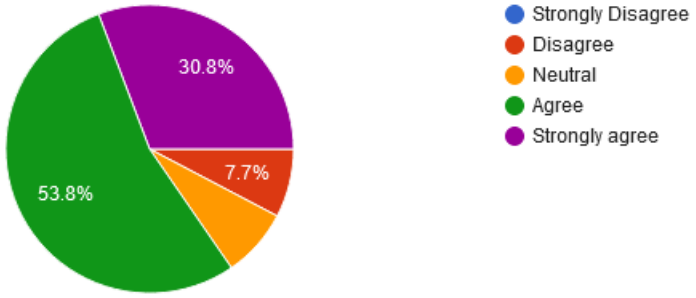
5.2 Usability Testing Results

Results from the Usability Survey conducted in partnership with the McMaster Sign Language Club. Participants had a mid to low range of skill levels, aka our target demographic.



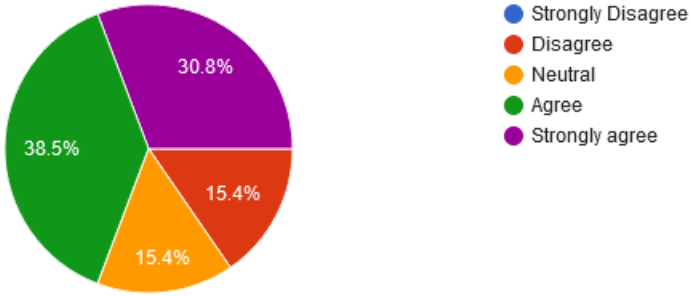
It was very easy to get right into a testing session with little to no hassle

13 responses



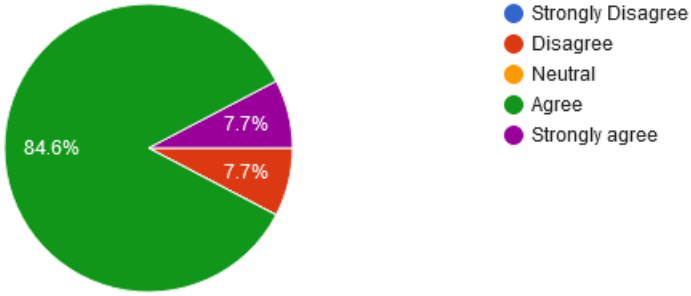
The User Interface is very friendly and it is easy to identify where everything is.

13 responses



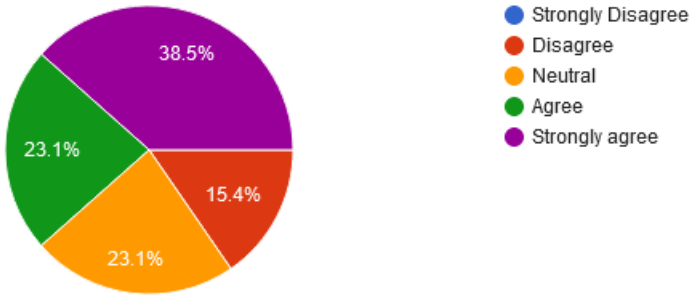
During a Quiz, its very easy to understand what to do and how to complete it

13 responses



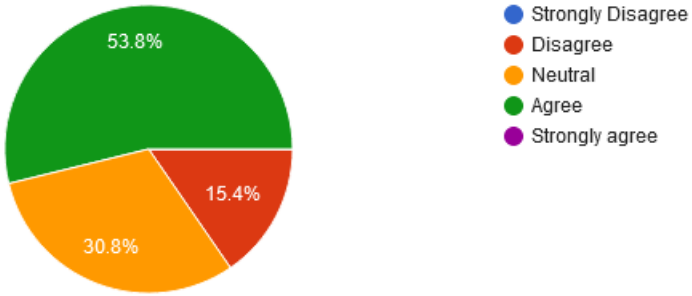
While signing, it is very easy to see what sign I am making and whether to make adjustments or not

13 responses



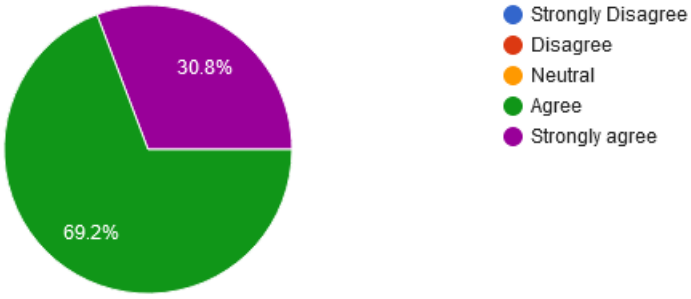
After completing a Quiz, I receive useful feedback

13 responses



At my current level of ASL knowledge, it is easy to use the application

13 responses



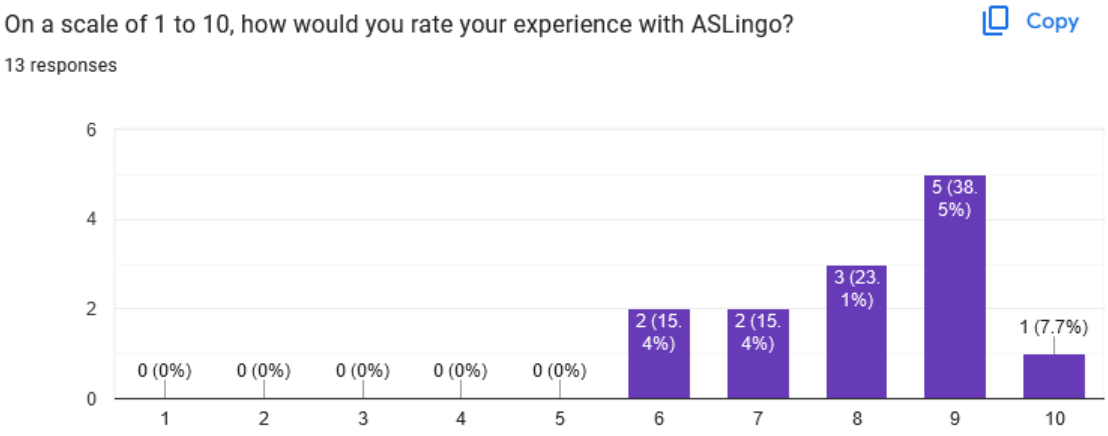


Table 5: System Tests for Usability

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 2, 3 of survey with positive result	User is able to use application, results of question 2, 3 of survey sow most participants Agree or Strongly Agree	Pass	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 4, 6 of survey with positive result	User is able to complete a quiz, results of question 4, 6 of survey sow most participants Agree or Strongly Agree	Pass	UHR1
NFRT1-UT3	User is able to use application with various hearing abilities	User opens application	User is able to use application, completes question 8 of survey with result over 75%	User is able to use application, results of question 8 overall was 80% from our user testing	Pass	UHR2
NFRT1-UT4	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	UHR3

5.3 System Tests for Performance

Table 6: System Tests for Performance

Test ID	Description	Input	Expected Output	Actual 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 very consistent, with the accuracy at around 94%.	Pass	PR2

6 Unit Testing

Test ID	Description	Input	Expected Output	Actual Output	Result	Req ID
DPM-UT1	Tests the vector normalization function	[1.0, 2.0, 3.0, -4.0], [4.0, -6.0, 3.0, -1.0]	[0.25, 0.5, 0.75, -1], [4.0/6.0, -6.0/6.0, 3.0/6.0, -1.0/6.0]	[0.25, 0.5, 0.75, -1], [4.0/6.0, -6.0/6.0, 3.0/6.0, -1.0/6.0]	Pass	FR2
DPM-UT2	Tests the feature processing function	[[1.0, 2.0, 5.0], [3.0, 4.0, -2.0], [5.0, 6.0, -3.0]], [[-1.0, 1.0, 0.0], [2.0, 1.0, -3.0], [-3.0, 2.0, 1.0]]	[[0.0, 0.0, -2.0], [-2.0, -2.0, 5.0], [-4.0, -4.0, 6.0]], [[0.0, 0.0, 0.0], [-3.0, 0.0, 3.0], [2.0, -1.0, -1.0]]	Correct	Pass	FR2

7 Changes Due to Testing

- A 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.

9 Trace to Requirements

	FR Req.						
System Test	1	2	3	4	5	6	7
FRT1-LP1	X	X		X			
FRT1-LP2					X	X	X
FRT1-LP3	X	X	X				
FRT2-U1	X						
FRT3-HW1	X						
FRT3-HW2	X						

Functional Requirements to System Tests

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

Non Functional Requirements to System Tests

10 Trace to Modules

	Module											
System Test	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
FRT2-LP1	X	X						X				
FRT2-LP2	X							X	X		X	
FRT2-LP3	X	X	X					X		X		
FRT3-U1								X				
FRT4-HW1								X				
FRT4-HW2								X				
NFRT1-UT1									X			
NFRT1-UT2									X			
NFRT1-UT3									X			
NFRT1-UT4	X							X				
NFRT2-PT1			X		X							
NFRT2-PT2	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. Please rate your own level of ASL proficiency on a scale of 1-10

   [1 = I know absolutely nothing, 10 = I sign at a high level]

2. From Strongly Disagree to Strongly Aggree, rate the following sentence: It was very easy to get right into a testing session with little to no hassle

3. From Strongly Disagree to Strongly Aggree, rate the following sentence: The User Interface is very friendly and it is easy to identify where everything is

4. From Strongly Disagree to Strongly Aggree, rate the following sentence: During a Quiz, its very easy to understand what to do and how to complete it

5. From Strongly Disagree to Strongly Aggree, rate the following sentence: While signing, it is very easy to see what sign I am making and whether to make adjustments or not

6. From Strongly Disagree to Strongly Aggree, rate the following sentence: After completing a Quiz, I receive useful feedback

7. From Strongly Disagree to Strongly Aggree, rate the following sentence: At my current level of ASL knowledge, it is easy to use the application

8. How would you rate your overall experience with ASLingo on a scale of 1-10

   [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.