Hazard Analysis
Software Engineering

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

Table 1: Revision History

Date	Developer(s)	Change
Oct 17 2022	Jeremy	Added 2 FMEA table entries related to web application hazards
Oct 17 2022	Andrew	Added Camera FMEA table entry
Oct 17 2022	Jeremy	Added 2 more FMEA table entries related to web application
Oct 17 2022	Stanley	Rearranged some FMEA table entries, added computer vision table entries
Oct 18 2023	Edward	Added sections 1, 2, 3, 4
Oct 19 2023	Everyone	Finished Section 6 and 7
Jan 23 2024	Cassidy	Addressed git issue #35 and #38 regarding system boundaries and typo
Jan 23 2024	Cassidy	Addressed git issue #39 regarding rationale and fit criterion of requirements
April 3 2024	Jeremy	Removed user auth hazard requirements per git issue #37 and clarification of critical assumptions for git issue #36 and #40

Contents

1 Introduction

This document aims to outline and analyze the potential hazards of ASLingo. A hazard can be defined as a system state or set of conditions, often arising from inherent risks or software anomalies, that, when coupled with particular worst-case environmental conditions or unexpected interactions, can lead to a loss or adverse outcomes. This embodies potential sources of harm due to software failures, bugs, or undesired system behavior, emphasizing proactive identification and mitigation to ensure software safety and functionality.

2 Scope and Purpose of Hazard Analysis

Hazard analysis is a fundamental aspect of the software development process, crucial for preventing losses or adverse outcomes that are undesirable for any product. It involves identifying areas where hazards may arise and determining steps to either reduce or eliminate their effects, making it an important part of the development journey. This analysis is closely tied to the safety and security requirements of the software. Ensuring these requirements are well met significantly contributes to enhancing the software's reliability, making it a more dependable product in the long run.

3 System Boundaries and Components

ASLingo's system will involve the following components:

1. A camera to allow for user input
2. A web frontend to provide user interface
3. A backend to process software logic
4. A machine learning model to interpret user hand signs

The system boundaries of ASLingo include our web application with the user's camera input, our machine learning model which is trained to learn hand signs, and our back and front end systems that will allow our model to interact with the user's inputs. Due to the limitations of these parts of our system, these may cause hazards to occur or cause our system to not function as intended, so these components of our system will be explored further in this document.

4 Critical Assumptions

1. Assume users are using ASLingo for its intended purpose of learning sign language
2. Assume users are able and willing to follow safety instructions
3. Assume user's web browser is supported and can connect to a fully functioning webcam

5 Failure Mode and Effect Analysis

Table 2: Failure Mode and Effect Analysis

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref.
Working Application	Error state	User cannot view any pages	Major system failure due to bugs	Display static error page and await application restoration	WAR1	H1-1
Camera	Visual feed is unable to be captured	User's sign cannot be perceived by the device	Poor Lighting Conditions	Instruct user to adjust their environment lighting or move to environment with sufficient lighting	HR1	H2-1
	Physical defect that impairs operation		Cracked/Filthy camera lenses	Notify user that camera lenses appear to be inoperable	HR2	H2-2
Machine Learning Model	Model fails to process/recognize camera input	User sign input cannot be processed accurately/correctly	Hand sign motions are too fast/slow	Interrupt the user and inform user to adjust hand sign motions accordingly	CVR1	H3-1
		Sign recognition works with group members and stakeholders, but fails when a new user uses the application	Model is trained on a specific set of training data and tested on a specific set of people (developers and stakeholders)	Rigorous testing on multiple testing sets and on users not affiliated with the project to ensure hands with varying qualities can be recognized	CVR2	H3-2

6 Safety and Security Requirements

6.1 Web Application Requirements

Table 3: Web Application Requirements

Requirement No.	**Description**	**Rationale**	**Fit Criterion**

WAR1	The system shall notify users of any expected down times for upgrades and be locked out of account until complete.	This is to allow the users to be notified of service outages and give them an estimate of when they will be able to return to the application.	The developers shall certify the product complies with these requirements by ensuring users are notified in advance of any service outages.

6.2 Hardware Requirements

Table 4: Hardware Requirements

Requirement No.	**Description**	**Rationale**	**Fit Criterion**
HR1	The system shall inform the user that the camera requires better lighting in their environment if there not enough currently.	This is to ensure that the user is aware of any system/camera adjustments that may be needed to capture their input correctly, as well as comply with FR1 and FR4.	The developers shall certify the product complies with these requirements by ensuring the user is notified if there are any errors associated with their camera.
HR2	The system shall inform the user that the camera is inoperable.	This is to ensure that the user is informed if the system cannot access the camera feed for any particular reason.	The developers shall certify the product complies with these requirements by ensuring the user is notified if the camera is inoperable.

6.3 Computer Vision Requirements

Table 5: Computer Vision Requirements

Requirement No.	**Description**	**Rationale**	**Fit Criterion**
CVR1	The model will be trained on multiple sign language data sets from varying sources and characteristics.	This is to ensure that the system is trained using a wide range of signs, as well as different angles and lighting conditions of the training data. This will allow the model to have ample training data to correctly recognize signs using ASL.	The developers shall certify the product complies with these requirements by ensuring the machine learning model is trained on a variety of trusted ASL sources.

CVR2	The system will notify the user if it is unable to detect a hand within frame.	This is to ensure that a user's input will be correctly identified to the best of the model's ability, and the user will be informed if they have to sign more clearly into their camera.	The developers shall certify the product complies with these requirements by ensuring the user is notified if the system cannot detect user's hand.

7 Roadmap

After careful consideration and reassessment, we see that there are many new requirements for us to take into consideration that weren't initially apparent when writing the Software Requirement Specification. Ideally, we will aim to implement every safety requirement, but realistically when taking into account time constraints, those requirements that are strictly necessary for system functionality may be the only ones that get implemented. Primarily, we will be focusing in on the WA and CV requirements, with a majority of the effort being channeled into the CV requirements since that is the main basis that will be providing the functionality for sign recognition. The hazard analysis will be used as a reference throughout the development process and may be amended when necessary in the future.