# Hazard Analysis
# Bridging Gaps: AI for Diagram Accessibility

Team 22, Reading4All
Nawaal Fatima
Dhruv Sardana
Fiza Sehar
Moly Mikhail
Casey Francine Bulaclac

Table 1: Revision History

| **Date** | **Developer(s)** | **Change** |
|---|---|---|
| Date1 | Name(s) | Description of changes |
| Date2 | Name(s) | Description of changes |
| ... | ... | ... |

# Contents

# 1 Introduction

## 1.1 Problem Statement

Students with disabilities confront major challenges to reading technical diagrams, which are often given in the form of static images that screen readers cannot interpret. Manually developing alternative (alt) text is possible, but it is resource-intensive, inconsistent, and not scalable for huge volumes of course content. This generates disparities in access to learning materials in postsecondary education.

To address this issue, this project proposes creating an AI/ML-powered application that automatically generates clear and detailed alt text for technical diagrams. The tool intends to increase accessibility, assure compliance with AODA regulations, and promote greater inclusion in higher education.

## 1.2 Hazard Analysis Introduction

A hazard is a system state or combination of conditions that, when paired with specific environmental or contextual factors, can result in an unwanted or adverse event in the system or its surroundings. Hazards in software and AI development encompass not only physical threats but also usability, ethical, and technological issues.

Hazards in this project may develop during the design, training, and implementation of the alt-text generating model. These risks include technical flaws in generated text, misinterpretations that may mislead learners, ethical concerns regarding bias in the dataset, and compatibility issues with assistive technology.

This hazard analysis identifies and evaluates such risks to guarantee that the system provides consistent accessibility gains while not mistakenly introducing new obstacles.

# 2 Scope and Purpose of Hazard Analysis

The objective of this hazard analysis is to analyze the potential hazards and damages related to the AI/ML alt-text generating tool throughout its development and operation. The most significant hazards are:

- **Technical Flaws:** Incorrect or inadequate alt text may mislead students, impair comprehension, or violate accessibility guidelines. This could result in a loss of confidence in the tool and poor results in education.
- **User Insensitivity:** Alt text that is excessively technical, unsophisticated, or inconsistent may not meet the needs of students. This may result in a loss of usefulness, reducing participation among students, teachers, and accessibility professionals.
- **External Dependencies:** Relying on third-party libraries, APIs, or screen reader compatibility can lead to external failures. Interruptions in these functions may result in a loss of functionality or downtime.

The purpose of this hazard study is to systematically identify these hazards, estimate their potential impact, and develop mitigation strategies. By doing this, we aim to minimize losses associated with time, trust, resources, and accessibility outcomes, ensuring that the tool contributes positively to equitable education.

# 3 System Boundaries and Components

[Dividing the system into components will help you brainstorm the hazards. You shouldn't do a full design of the components, just get a feel for the major ones. For projects that involve hardware, the components will typically include each individual piece of hardware. If your software will have a database, or an important library, these are also potential components. —SS]

## 3.1 System Overview

The system is a web-based AI tool that generates alternative text (alt text) for uploaded images or figures and integrated with screen readers to improve accessibility for visually impaired users.

## 3.2 System Boundaries

- **Internal Components:** Alternative Text Generation Machine Learning (ML) Model, User Interface, Session History Manager
- **External Components:** McMaster Authentication System, External AI/ML Frameworks, Screen Reader Software

## 3.3 System Components

1. **Alt Text Generation ML Model**
   - **Purpose:** To automatically generate accurate and descriptive alternative text for uploaded images using machine learning.
   - **Key Functions:**
     - Process image inputs received from the backend and extract key visual features.
     - Generate contextually relevant text descriptions.
     - Return the generated alternative text to the backend to display to the user interface.
2. **User Interface**
   - **Purpose:** To serve as the primary interaction point between the user and the system and allow users to upload images and view generated alt text.
   - **Key Functions:**
     - Enable users to upload images through an accessible web interface.
     - Display generated alt text and allow users to edit, copy, and download the text.
     - Provide features that are accessible and complies with the Web Content Accessibility Guidelines (WCAG) 2.1 standards.
     - Communicates user requests and display outputs from the backend.
3. **Session History Manager**

- **Purpose:** To ensure the continuity of the current session and manage user data during active use of the system.
- **Key Functions:**
  - Track unique user sessions throughout interaction with the web application.
  - Store previously uploaded images and generated alternative text for the current session to allow users to view history.

# 4 Critical Assumptions

This section documents the assumptions made during the hazard analysis of Bridging Gaps: AI for Diagram Accessibility (Reading4All). The number of assumptions is kept to a minimum to reduce the chance of overlooking potential hazards. Where assumptions are made, they set clear boundaries for the analysis and define the conditions under which the system is expected to operate safely.

- **Assumption 1: Input Integrity.** All image files provided to the system are assumed to be valid image formats and not corrupted or maliciously constructed to exploit parsing vulnerabilities.
- **Assumption 2: Standards Stability.** Accessibility guidelines (WCAG 2.1, AODA) are assumed to remain stable for the operational lifetime of the system.
- **Assumption 3: Model Performance.** The machine learning models used for visual recognition and natural language generation are assumed to operate within validated ranges of accuracy and reliability.
- **Assumption 4: Human Oversight.** It is assumed that alternative text generated by the system will undergo review by human instructors, teaching assistants, or accessibility specialists before being used in educational contexts.

Violations of these assumptions may introduce additional hazards outside the current scope of this analysis. Such cases would require re-evaluation of risks and system design updates.

# 5 Failure Mode and Effect Analysis

Table 2: Failure Mode and Effect Analysis (FMEA) for Reading4All

| **HA ID** | **Component** | **Failure Mode** | **Effects of Failure** | **Possible Causes** | **Recommended Action / Mitigation** | **SRS Ref.** |
|---|---|---|---|---|---|---|
| HA-1 | Frontend UI | All user interface components cannot be controlled using keyboard | Users using assistive technology cannot operate all features | Missing Accessible Rich Internet Applications (ARIA) roles, incorrect tab order | WCAG 2.1 Level AA evaluation; fix focus order and ARIA; add additional tests for accessibility needs | UHR-AR 1, UHR-AR 2 |
| HA-2 | Frontend UI | Alt text output not compatible with screen readers | Screen readers skip generated alt text | Missing ARIA labels | Validate with NVDA/JAWS/VoiceOver; use `aria-label` with user interface components | FR 3, UHR-AR 1 |
| HA-3 | Backend Controller | Accepts unsupported/corrupted files | System crash or user confusion | Missing images/type validation or file-size limits | Validate uploads; enforce size/type checks; detailed error messages | FR 1, PR-RFT 1, SR-IM 1 |
| HA-4 | Backend Controller | Timeout during image analysis | User perceives failure and is frustrated with system; repeated submissions; increased load on backend controller | Increased model latency; missing timeout handling | Add timeouts/retries for better user handling; progress indicator | PR-SL 1, PR-SL 2 |
| HA-5 | Alt Text Generation Model | Generates offensive/biased or Personal Interest Information (PII)–leaking text | Ethical/privacy risk; loss of user trust | Lack of Model Training and Accuracy; no output filtering | Add additional filters and checks for PII Data and offensive texts | SR-PR 2 |
| HA-6 | Alt Text Generation Model | No output or empty alt text | Users cannot use result; reduced learning impact | Interface failure or Application Programmable Interface (API) crash | Retry option; "No Text Generated" label; clear user feedback | PR-RFT 2 |

*Table continues on next page*

*Table 2 (continued)*

| HA ID | Component | Failure Mode | Effects of Failure | Possible Causes | Recommended Action / Mitigation | SRS Ref. |
|---|---|---|---|---|---|---|
| HA-7 | Session Storage Component | Images metadata or generated alt text not deleted after processing | Privacy exposure; increased storage/cost | Cleanup jobs fail or not configured | Auto-delete temp files; periodic cleanup; log storage usage; manual deletion triggered by deletion failure alarms | SR-PR 1 |
| HA-8 | Session Storage Component | Session history not found | Loss of user trust and session data causing frustration | Session key mismatch; write errors | Atomic writes; bind session to SSO token | FR 5, SR-AR 2 |
| HA-9 | Alt Text Model Output Training Evaluation | Evaluation metrics and scale linked to wrong model output | Inaccurate metrics | Race condition; wrong foreign key | Immutable IDs; transactional writes; enforce referential integrity | PR-PAR 1 |
| HA-10 | McMaster SSO (Access Control) | Session bypass | Unauthorized access | Token reuse; improper validation | Validate tokens on the server | SR-AR 1, SR-AR 2 |

# 6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

PR-SR-HA 1. *The system must notify the user when a timeout occurs during alternative text generation.*

**Rationale:** Users should be informed when the alternative text generation exceeds the expected amount of time. If users are not notified they may send multiple requests, leading the server to be overloaded; this would also lead to user frustration.
**Fit Criterion:** When a timeout occurs, the system displays a message indicating the timeout and a "Retry" option. The message must follow accessibility guidelines and be compatible with screen readers.
**Priority:** Medium
**Hazard Analysis Connected**: HA4

PR-SR-HA 2. *The system must safely exit when a timeout occurs and ensure that no user data or incomplete alternative text is stored or shown to the user.*

**Rationale:** Safely exiting during a timeout prevents users from seeing incomplete alternative text and mistaking it for a complete output, which may cause confusion. Leaving user data stored would also be a security violation.
**Fit Criterion:** When a timeout occurs, the system must stop processing and delete the users data and any incomplete alternative text that has been generated.
**Priority:** High
**Hazard Analysis Connected**: HA4

PR-SR-HA 3. *The system must ensure that messages notifying the user of failure, do not reveal any system code or data.*

**Rationale:** This will prevent internal data from being shown to users, which may lead to system and user security issues.
**Fit Criterion:** All error messages shown to the user only display the necessary information and do not contain any technical information.
**Priority:** High
**Hazard Analysis Connected**: HA4, HA7

# 7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

All safety and security requirements outlined in our SRS, as well as newly discovered requirements will be implemented as part of the capstone timeline. This includes the following requirements, which can be found in our SRS document:

- SR-AR 1
- SR-AR 2
- SR-IR 1
- SR-IR 2
- SR-PR 1
- SR-PR 2
- SR-AU 1
- SR-AU 2
- SR-IM 1
- SR-IM 2
- PR-SCR 1
- PR-SCR 2
- PR-SCR 3
- PR-SCR-HA 1
- PR-SCR-HA 2
- PR-SCR-HA 3

# Appendix — Reflection

[Not required for CAS 741 —SS]

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:

**Moly Mikhail - Reflection**

1. What went well while writing this deliverable?

   Completing part 6, where we outlined the Safety and Security requirements, went well for several reasons. Firstly, completing the previous section 5 helped us directly derive the new Safety and Security requirements that needed to be outlined. Additionally, working on this document after finishing the SRS document made the process much easier, as we had gained a lot of practice with determining and writing requirements from the SRS.

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

   One pain point we faced during this deliverable was the dependency on other sections within the HA document and the SRS document. For example, to complete section 5 in the HA document, we needed the requirements in SRS to be completed so they can be referenced as needed. This was challenging as developing our SRS requirements was a long process that required significant work and detail; meanwhile, the HA Failure Mode and Effect analysis also needed extensive work. This made it hard to manage our time effectively, as the two documents couldn't be worked on in parallel at times.

3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

4. Other than the risk of physical harm (some projects may not have any

5. What went well while writing this deliverable?

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

7. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

8. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

**Nawaal Fatima - Reflection**

1. **What went well while writing this deliverable?**
   I think the easiest part was coming up with the assumptions themselves. From my summer work experience with Ms. Sui, I was already familiar with the main areas where Reading4All could run into issues (inputs, standards, model accuracy, and human review), it felt pretty natural to turn those into clear assumptions. It also helped that the section didn't need to be super long, so I could keep it focused and to the point.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   The tricky part was figuring out how much detail to put in without overcomplicating things. At first, I thought about writing assumptions that basically ruled out certain failures, but I realized that would go against the whole idea of hazard analysis. To fix that, I stuck to assumptions that made sense for defining boundaries without pretending hazards don't exist. Another small challenge was making the wording simple enough - so I rewrote a couple of the assumptions to sound clearer and less "technical report" heavy.