

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
2	Scope and Purpose of Hazard Analysis	1
3	System Boundaries and Components	1
3.1	System Overview	1
3.2	System Boundaries	1
3.3	System Components	1
4	Critical Assumptions	2
5	Failure Mode and Effect Analysis	2
6	Safety and Security Requirements	3
7	Roadmap	3

[You are free to modify this template. —SS]

1 Introduction

[You can include your definition of what a hazard is here. —SS]

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

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 back-end.

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

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

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]

7 Roadmap

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

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:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
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 appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

Casey Francine Bulac - Reflection

1. What went well while writing this deliverable?

Writing this deliverable was organized as the team split the sections beforehand, so we had clear instructions on how and who is completing which sections. It was also helpful completing the SRS before this document as it helped define certain sections such as the System Boundaries and Components.

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

One challenge I ran into with this deliverable were keeping the language and details aligned with the SRS. Another challenge was having to finish one section before I could move on to others as some sections depended on previous ones. To address this, I double-checked the SRS and kept clear, timely communication with my team about what needed to be done and by when.

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 appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

One main risk in software products is in security and privacy. Situations such as unauthorized access and data leaks can harm users of the software and can violate laws. This can also lead to damage of the developer's reputation, as well as fines and lawsuits. Another risk in software products is failure to comply with certain required regulations. For example, violating licenses or accessibility rules can result in the delay or blocking of deployments, and may also lead to legal issues.