

# Problem Statement and Goals

## 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
September 17th 2025	Fiza Sehar	First draft of document
Date2	Name(s)	Description of changes
...	...	...

## 1 Problem Statement

### 1.1 Problem

Students with disabilities face barriers to equitable education, with access to technical diagrams being a major challenge. These diagrams are often distributed as static images, making them unreadable by assistive technologies such as screen readers. Manually creating detailed alternative text is time-consuming, resource-intensive, and inconsistently implemented, resulting in inequitable access to learning materials across courses. This project aims to develop an ML/AI-driven tool that automatically generates clear, descriptive alternative text for technical diagrams, ensuring compatibility with screen readers, compliance with AODA standards, and improved inclusion within post-secondary education.

### 1.2 Inputs and Outputs

**Inputs:** Technical diagrams requiring alternative text descriptions.

**Outputs:** Automatically generated descriptive alt text suitable for screen readers.

## 1.3 Stakeholders

### Direct Stakeholders

1. **Students with Disabilities:** Primary beneficiaries who will gain access to previously inaccessible technical diagrams.
2. **Instructors and TAs:** Use the tool to generate accurate alt text for course materials.
3. **Accessibility Services Staff:** Ensure that generated descriptions meet AODA standards and institutional accessibility requirements.

### Indirect Stakeholders

1. **University Administration:** Interested in improving overall accessibility compliance across courses.
2. **Curriculum Designers:** Can integrate the tool into course development workflows for consistent accessibility.
3. **Assistive Technology Vendors:** Benefit from improved compatibility and user experience when screen readers are used with accessible diagrams.

## 1.4 Environment

### Development Frameworks and Tools:

1. *GitHub* will be used for version control and collaboration.
2. *Visual Studio Code* will be used as the IDE for development.
3. *Python and Machine Learning Libraries* will be used to implement and train the alt text generation model.

## 2 Goals

Our goal is to make technical diagrams accessible by generating high-quality alternative text. We plan to achieve this through the following goals:

1. **Generate Clear and Concise Descriptions:** Produce alt text that captures details of the diagram while remaining easy to read and understand.
2. **User Satisfaction and Learning Outcomes:** Improve Learning Impact and Length Scores while maintaining high Accessibility and Usability Scores, ensuring descriptions remain helpful and consistent.
3. **Browser Extension:** Provide a easy-to-use web browser extension to generate alt text directly within the user's workflow.
4. **Hotkeys:** Allow users to navigate, review, and swap descriptions using customizable keyboard shortcuts.

### 3 Stretch Goals

1. **Multi-Language Support:** Extend the alt-text generation model to support multiple languages, improving accessibility for a broader range of students. This feature would include:
  - **Language Coverage:** Add support for languages such as French to meet bilingual education requirements.
  - **Localization:** Ensure that generated descriptions maintain context, clarity, and technical accuracy across languages.
2. **Batch Processing:** Allow students to process multiple diagrams or entire sets of study materials at once, making it faster to generate alt text for all content they need to review. This goal would involve:
  - **Bulk Upload:** Provide an option for students to select multiple images or documents and generate alt text for all of them in one step.
  - **Performance Optimization:** Ensure smooth handling of large batches so students can access their study materials without delays.

### 4 Extras

The overall challenge level of this project is moderate. While developing an ML-based system for generating alt text involves model design, training, and evaluation, there are well-established frameworks and research papers that guide these processes, making them approachable for our team. Our collective experience with Python and machine learning from coursework provides a solid foundation for completing this work successfully. The project will require balancing technical development with accessibility considerations, but its scope remains feasible within the given timeline.

To strengthen the project, we will include two additional deliverables:

- **Norman's Principles Report:** An evaluation of the tool against Norman's design principles to ensure usability.
- **User Manual:** Comprehensive documentation to guide students in using the tool effectively and to support long-term adoption.

These deliverables will help ensure that the final tool is both usable and well-documented for future use.

## Appendix — Reflection

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:

### **Nawaal Fatima - Development Plan Reflection**

**1. Why is it important to create a development plan prior to starting the project?**

Our project is very user-oriented and has more risks when it comes to reliability and functionality as Group 22 is designing for a demographic with which we have little/no working experience with. Knowing this, it is very important to have a blueprint of what we are building before we waste resources and cause our testers/end-users any unnecessary frustration. When we have a plan, we can also make sure that we are all working towards the same goal. It also helps us to identify potential challenges and risks early on, allowing us to develop strategies to mitigate them. The development plan asked a couple questions we didn't consider, which helped us to think more critically about our project and how we can make it successful.

**2. In your opinion, what are the advantages and disadvantages of using CI/CD?**

I think there are more advantages than disadvantages when it comes to CI/CD. The main advantage is that it allows for faster and more frequent releases, which can lead to quicker feedback from users and a more responsive development process. It also helps to catch bugs and issues early in the development process, which can save time and resources in the long run. However, one disadvantage is that it can be difficult to set up and maintain, especially for smaller teams or projects with limited resources. It also requires a certain level of discipline and commitment from the development team to ensure that code is properly tested and reviewed before being merged into the main branch. Overall, I think the benefits of CI/CD outweigh the challenges, and it is a valuable practice for modern software development.

**3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?**

We're in agreement about most aspects of the development plan. Most 'disagreements' we had were minor - such as naming conventions to follow or what processes to establish to ensure everything remains organized. We resolved these disagreements through open communication, making sure to listen to each other's perspectives and find solutions that worked for everyone. I imagine as we continue to work together, we may have more disagreements, but I am confident that we will be able to resolve them in a similar manner.

1. *What went well while writing this deliverable?*

Our team collaborated smoothly and was able to create a clear plan of action early on. We held multiple meetings to discuss the problem, inputs, and outputs, as well as the overall vision for what our product should look like. Work was divided efficiently, and we supported each other by sharing ideas and reviewing one another's contributions. Our supervisor provided us with useful information and previously researched data, which was particularly insightful and helped shape the initial proof of concept for our project.

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

We did not struggle with defining the problem itself, as we quickly aligned on the key issue of improving accessibility for technical diagrams. The main challenge we faced was uncertainty about the implementation details, particularly what kind of self-training ML model to use and how we would gather or prepare input data (e.g., whether we would use real diagrams for training). There were still many unknowns about the exact approach, but we had enough foundational information to proceed confidently. Meeting with our supervisor was very helpful in this stage as we were introduced to a data analysis study they had conducted, which provided valuable insight into the ideal length of alt text and highlighted weaknesses in e

3. *How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?*

We intentionally avoided designing an overly complex solution by leveraging existing technology as a foundation. Instead of building everything from scratch, we chose to use a web browser extension framework as our baseline, since extensions are widely used and well-documented, making information and implementation examples easy to find. This allowed us to spend less time reinventing the wheel and more time focusing on the core technical challenge of integrating and training the ML model to generate high-quality alt text. This approach keeps the project achievable within the timeline while still providing significant technical depth through model development and accessibility considerations. By leaving advanced features such as multi-language support and batch processing as stretch

goals, we ensured that our scope remains realistic but still challenging enough for a senior design project.