Software Requirements Specification for Bridging Gaps: AI for Diagram Accessibility: subtitle describing software

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

October 10, 2025

Contents

Symbolic Constants

Name	Value
Symbol	1.0
Symbol	1.1

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Introduction

1.1 Purpose of Document

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for the Bridging Gaps: AI for Diagram Accessibility system, **Reading4All**. The document explains the project's aims, context, restrictions, and expected behavior in order to ensure that developers, accessibility professionals, instructors, and project stakeholders all have a shared understanding.

This SRS provides a formal reference for Reading4All's design, implementation, testing, and validation. It establishes a clear link between user requirements, accessibility standards (such as AODA and WCAG 2.1), and system features that enable reliable alternative text generation for technical diagrams.

1.2 Scope of Project

Reading4All is an artificial intelligence/machine learning tool that provides detailed and contextually aware alternative text for complicated technical graphics, notably those found in postsecondary STEM course materials. The system combines machine vision and natural language generation models to assess diagram content, identify key parts, and provide text descriptions that are compatible with screen readers and assistive technology.

Core objectives:

- Automate the creation of comprehensive and accurate alternative text for technical diagrams.
- Maintain compliance with the **Accessibility for Ontarians with Disabilities Act (AODA)** and the **WCAG 2.1** accessibility standards.
- Improve the learning experiences and inclusion of students with visual challenges in higher education.
- Reduce the instructor effort and institutional costs related to manual alt-text creation.

Final deliverable includes:

- A web-based or locally hosted interface for uploading images and creating alternative-texts.
- A backend service that combines AI models for diagram analysis and generation of languages.
- Structured output formats are suitable with systems for learning management and screen readers.

1.3 Characteristics of Intended Reader

This document is intended for:

- **Developers** who are responsible for implementing and testing the system.
- **Accessibility Professionals** to ensure AODA and WCAG compliance.
- **Academic Stakeholders** including educators, instructional designers, and content creators.
- **Supervisors and Assessors** who assess project completion and design quality.

Readers should have an overall knowledge of software engineering, web development, and fundamental machine learning techniques. Accessibility reviewers should understand digital accessibility principles and standards.

1.4 Organization of Document

This Software Requirements Specification (SRS) is divided into twenty-six sections following the IEEE standard structure.

- **Sections 1–5:** Establish the foundation of the document, including the project scope, stakeholders, assumptions, and terminology.
- **Sections 6–10:** Define the overall system context, business model, product scope, and high-level functional requirements.
- **Sections 11–17:** Specify detailed non-functional requirements such as usability, performance, maintainability, security, and compliance.

- **Sections 18–26:** Present supporting material including open issues, off-the-shelf components, project tasks, migration plans, documentation, and solution ideas.

This organization ensures logical flow from project background to detailed requirements and supporting documentation.

1.5 System Context

Inputs: Technical diagrams or schematics (JPEG, JPG, and PNG) provided for alt-text generation.

Processing: The system analyzes visual elements using a vision model and generates textual descriptions through a language model.

Outputs: Descriptive alternative text formatted for screen readers, including HTML `alt` attributes, text files, and ARIA labels.

1.6 Document Conventions

This document follows the structure and formatting guidelines of the McMaster University SFWRENG 4G06 Capstone SRS template. All section numbers, requirement identifiers, and tables conform to the IEEE SRS format. Standard SI units are used where applicable. Technical terms, acronyms, and variables are presented in `monospaced` or *italic* text for clarity.

1.7 Reference Material

Relevant Standards and Reference Documents:

- Accessibility for Ontarians with Disabilities Act (AODA, 2005)
- Web Content Accessibility Guidelines (WCAG 2.1)
- *SRS Template*, McMaster University

Insert your content here.

2 Stakeholders

The project stakeholders consist of people who have a need or interest, whether direct or indirect, for alternative text generation for visual, idle content (such as images or diagrams). These stakeholders will influence and be affected by the project's development decisions and progress. To meet user needs, it is vital to understand the stakeholder roles and expectations.

First, this section introduces the client, customer and other stakeholders involved in this project. Then, the product users are described, specifically the hands-on users of the project. Finally, personas, priority levels and anticipated participation levels are listed for each stakeholder.

2.1 Client

This project's client is Ms. Jingchuan Sui who works as a Media Lab Specialist Supervisor at the Faculty of Engineering, McMaster University. As this project's supervisor, her main role is to provide guidance and voice any concerns during the development phase with her technical and domain expertise. She will be the main source for setting requirements while also being directly involved in the development of this project, providing feedback and opinions on the Human-Computer interface components.

2.2 Customer

The customers of this product are McMaster users, specifically, McMaster students, staff and teaching instructors who are directly involved in learning from course content or making them accessible as per the Accessibility for Ontarians with Disabilities Act (AODA). In other words, McMaster University stakeholders who benefit directly from accessible course content. For example, primary customers can include students who use a screen reader for learning purposes or a teaching assistant who is making a course's content AODA compliant. Furthermore, under her position at McMaster University, Ms. Sui can also be considered a customer, as she aids in course content remediation and thus, is also one of the intended end-users.

For development, Group 22 is tailoring the solution to the McMaster demographic in line with Ms. Sui's requirements. However, the product has the potential to support any users who require alternative text generation

for visual content. Feedback from McMaster stakeholders will be prioritized to maintain a clear and manageable scope.

2.3 Other Stakeholders

This subsection discusses other groups that are indirectly impacted or who contribute to the ecosystem of accessibility, content creation, and AODA compliance.

2.3.1 Faculty of Engineering Instructors

This group consists of professors and lecturers responsible for creating and maintaining course content. They may benefit from automated alternative text generation to ensure their teaching materials are accessible.

2.3.2 Teaching Assistants (TAs)

As part of their work, TAs are often responsible for preparing, modifying, and uploading course content. They are stakeholders as they could use the system to simplify accessibility compliance.

2.3.3 Accessibility Services Office at McMaster

This group includes staff members who oversee accessibility compliance and provide accommodations for students. They have a strong interest in ensuring tools meet AODA standards.

2.3.4 McMaster IT Services / Media Production Services

These teams may be involved in system integration, technical support, and maintenance of the product within the university's digital infrastructure.

2.3.5 Students with Accessibility Needs

These students are those who may not be primary testers but are indirectly impacted by improved accessibility of course materials.

2.4 Hands-On Users of the Project

2.4.1 Students with Accessibility Needs

In some cases, students who use screen readers may provide feedback loops to improve generated alternative text. While they are customers, they may also be "hands-on" users if they test or adjust alt text themselves.

2.4.2 Teaching Staff

This group consists of TAs and instructors. As mentioned above, TAs Frequently upload, adapt, and remediate course content. They would be interacting directly with the tool to generate and refine alt text. On the other hand, some instructors (especially those who prepare their own slides, diagrams, or assignments) would use the system to add or edit alternative text.

2.5 Personas

Persona: Alice Bayes

Age: 27
Job Title: Teaching Assistant at McMaster University
Education: Bachelor's in History
Work Environment: Alice works under several teaching instructors to help deliver course content to students. She is in charge of marking assignments and has recently been tasked with auditing then remediating any inaccessible learning content.
Professional Background: Alice graduated two years ago, and as part of her undergraduate career, she has experience in working with students with disabilities. She is trained on making content accessible and AODA compliant.

Need: With so many courses to grade student work for, Alice needs a tool that can easily and quickly generate content for her to use as alternative text while she can ensure that her boss' teaching content meets AODA compliance.
Challenges: Balancing her work and life has been difficult as there are multiple images per document, and several documents per course. She is overwhelmed with the amount of grading she has to do on top of manually writing alternative text for over 50 images.

Persona: Chetan Dakshesh

Age: 20
Job Title: Chetan is a student at McMaster University.
Education: He is currently pursuing a Bachelor's in Electrical Engineering
Work Environment: Chetan has a super busy course load with six courses and volleyball club!
Professional Background:

Need: With so many courses and volleyball practice to keep up with, Chetan is finding it hard to keep track of course content. Furthermore, through his screen reader, he has picked up that there is no alternative text generated for several diagrams in a course he is taking. These diagrams are vital to his learning experience but he has little clue on what they indicate.
Challenges: Using large language models (LLMs) such as Chat-GPT doesn't work for him as the text generated is too generic and lacks substance. Chetan needs a tool that can effectively describe the diagram to him while staying relevant to the course material.

Persona: Eyad Fahim

Age: 40
Job Title: Professor at McMaster University
Education: Doctor of Philosophy (PhD) in Engineering
Work Environment: Eyad works on a fast paced work environment, connecting with over 100 students. **Professional Background:** With over 20 years of experience both in the workforce and academic, Eyad loves to teach the next generation of leaders about various engineering techniques.

Need: With the goal to celebrate students of experiences, Eyad is looking for help to make his teaching content accessible for all.
Challenges: Eyad needs a fast tool that can help gap the accessibility knowledge he lacks. He wants to ensure all students can learn from his materials with little to no barriers, including alternative text but he has no idea how to get started.

2.6 Priorities Assigned to Users

Primary users:

- Students with accessibility needs
- Ms. Jingchuan Sui
- Teaching Staff

Secondary users:

- Accessibility Services Office at McMaster
- McMaster IT/Media Production Services

2.7 User Participation

During the development process, the requirements will be gathered mainly from Ms. Sui. During testing phase, Group 22 will conduct usability testing to ensure AODA compliance and to further refine the product.

2.8 Maintenance Users and Service Technicians

For this project, maintenance activities may involve updating alternative text generation models, fixing bugs, or upgrading dependencies.

Expected Maintenance Users and Roles

- **McMaster IT Services / Media Production Services**
  These teams may oversee deployment, integration with institutional systems, and technical support. They require access to configuration tools, diagnostic information, and documentation for updates or troubleshooting.
- **Accessibility Services Office Staff**
  Although initially secondary stakeholders, some staff may contribute to iterative refinement of alt-text generation accuracy or compliance updates. Their participation may prompt system adjustments or patches.
- **Development Team (Group 22) or Future Maintainership Team**
  During initial deployment and handover, the development team or a designated successor group may perform updates to improve usability, resolve technical issues, or adapt to new accessibility standards.

3 Mandated Constraints

3.1 Solution Constraints

MD-SL 1. *The solution design must comply with at least the Level AA of the Web Content Accessibility Guidelines (WCAG) 2.1 standards*

Rationale: This ensures that the solution demonstrates inclusivity for users with visual, auditory, or cognitive impairments.
Fit Criterion: The solution must past all tests using WCAG automated testing tools and manual tests.
Priority: High.

MD-SL 2. *The solution must be implemented as a web tool*

Rationale: A web tool will allow for automated testing against the WCAG standards which ensures accesibility for users and allow users to upload images/figures to generate alternative text.
Fit Criterion: The web tool must be functional and allow users to generate alternative text by uploading images and figures.
Priority: High.

MD-SL 3. *The solution must support common image formats (e.g. Joint Photographic Experts Group (JPEG), Portable Network Graphic (PNG), etc.)*

Rationale: The web tool will enable users to upload images or figures to generate the alternative text, therefore the solution must be able to handle the different types of image formats.
Fit Criterion: The product must successfully process at least one image of each required format including JPEG and PNG images and figures.
Priority: High.

3.2 Implementation Environment of the Current System

MD-IE 1. *The product must be able to run on standard of laptop environments, including operating systems (OS) such as macOS, Windows, and Linux*

Rationale: This ensures that the product is compatible with the latest and major operating systems to allow the product to be accessible to users, regardless of their laptop environment.
Fit Criterion: The product must successfully install and operate on the latest three releases of macOS, Windows, and Linux, verified through installation and functionality testing on each OS.
Priority: High.

3.3 Partner or Collaborative Applications

MD-PA 1. *The product must be compatible with other accessibility tools (e.g. screen readers, screen magnifiers, dictation software)*

Rationale: This is to ensure that the product does not limit or intefere with other accessibility tools that meets the users' needs.
Fit Criterion: The product must operate simultaneously with at least one other accessibility tool, verified through interoperability testing.
Priority: High.

3.4 Off-the-Shelf Software

There are a number of existing AI generated alternative text off-the-shelf software in the market today. The following highlights a few of these tools, including their functions, benefits, and limitations:

1. **Azure AI Vision Image Analysis**: This service by Microsoft can extract a wide variety of visual features from images. Image Analysis offers image captioning models that generate one-sentence descriptions of an image's visual content. Limitations of this product is that it only generates one simple sentence, and that the image captions are only available in English.

2. **ALTTEXT.AI**: This service allows users to upload images and generate alternative text. The website supports over 100 languages and many modern image formats. A significant limitation of this project is that it doesn't guarantee compliance with WCAG which limits accessibility.

3. **accessiBe**: This service is an accessibility platform built for developers and engineers that plugs into their SDLC to detect and remediate WCAG issues at code level. The tools offers AI alt-text descriptions for images and allows users to review and edit the alt text. A limitiation of this tool is that it uses overlays that sit on top of a website to fix issues at run-time. This is an issue because overlays can conflict with assistive technologies and miss context-specific WCAG requirements creating a false sense of real accessibility compliance.

3.5 Anticipated Workplace Environment

The anticipated workplace environment for this product is academic settings such as universities, where students may require alternative text to interpret images and figures within their coursework and study materials.

3.6 Schedule Constraints

MD-SC 1. *The final product must be completed and tested by the end of the academic term (April 2026)*

Rationale: This is to ensure that the final product is functional and meets all requirements at the end of the academic year.
Fit Criterion: All deliverables are submitted, and the final product is tested and operable by April 2026.
Priority: High.

3.7 Budget Constraints

MD-BC 1. *The project budget must include compensation for user testers, set at maximum $150 CAD per participant for two rounds of usability testing.*

Rationale: This is to ensure that user testers are compensated for their meaningful feedback, and that our testing aligns with ethical practices.
Fit Criterion: There must be record of participants being compensated between the range of $100 and $150 for two rounds of testing.
Priority: High.

3.8 Enterprise Constraints

MD-EC 1. *The product must comply with the Accessibility for Ontarians with Disabilities Act (AODA)*

Rationale: This ensures that the product meets the legal requirements in Ontario and guarantees that the product is accessible to users with diverse needs.
Fit Criterion: AODA requires compliance with WCAG standards, which ensures that the product meets AODA regulations. Compliance must be verified through both automated WCAG testing tools and manual accessibility testing.
Priority: High.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Insert your content here.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- This project is being developed for a Software Engineering Capstone course with a fixed timeline.
- The solution is targeted primarily for laptop and/or desktop environments, but can later be extended for mobile platforms use.

5.2 Business Rules

The business rules established among the team are as follows:

- **Adherence to Project Schedule**: All deliverables and milestones must be completed according to the established project schedule. Any anticipated delays must be communicated in advance.

- **Pull Request Requirement**: All pull requests made by a team member must be reviewed by three other members before being merged into the `main` branch. The reviewers must provide approval or feedback within 24 hours of the pull request.
- **Team Communication Standard**: All team members must communicate respectfully and professionally during all discussions, meetings, and written communication.
- **Testing Requirements**: All code contributions must include appropriate unit, intergration, and functionality tests to ensure correctness and reliability. Accessibility testing must also be performed for all product features.

5.3 Assumptions

The following assumptions are made when using the product:

- Users will be operating on the three latest releases of browsers including Chrome, Safari, and Firefox.
- Users will have access to stable internet connection when using the product.
- Users will have basic knowledge of installing and enabling web tools.
- Users will be using the three most commonly used and latest versions of screen readers.

6 The Scope of the Work

6.1 The Current Situation

Currently, alternative text generation tools are able to provide sufficient descriptions for simple images and figures. However, for more complex visuals such as engineering diagrams, the generated alt text is often misleading, incomplete, or inefficient at conveying the intended meaning.
Accurate alternative text is particularly essential for individuals with visual or cognitive impairments, as it enables fair access to academic content. Without reliable descriptions, students may experience barriers to learning and

miss critical information conveyed in diagrams and figures.
The current limitations of existing generated alternative text tools are as follows:

- **Inaccurate Alternative Text**: Generated alt text may emphasize unimportant details and overlook key elements, resulting in misleading or confusing interpretations.
- **Oversimplification of Complex Figures**: Current tools frequently oversimplify technical or academic diagrams, failing to capture essential details required for learning.
- **High Manual Effort**: In many cases, subject matter experts must manually create alt text, which is time-intensive and not scalable across large volumes of academic content.

6.2 The Context of the Work

The product will be in the form of a web tool that integrates into existing accessibility workflows by providing accurate descriptions from images that can be read aloud by screen readers. The product will complement existing screen readers by ensuring accurate generated alternative text from uploaded images and figures of academic work are available. Figure 1 shows how the product will integrate with existing screen readers.

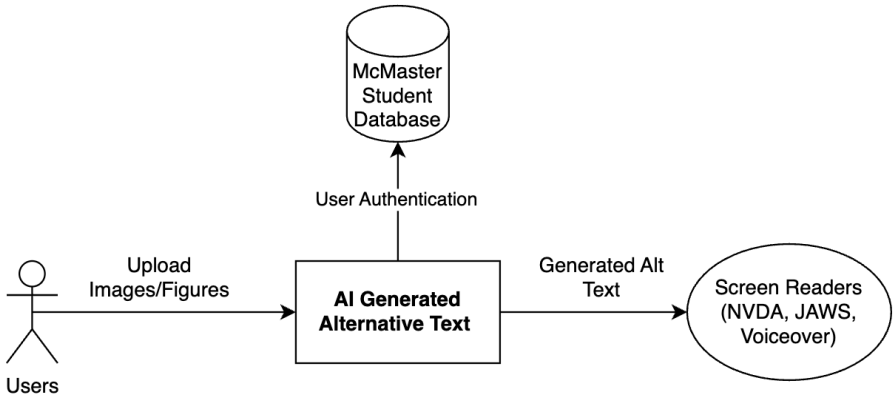

Figure 1: Work Context Diagram

6.3 Work Partitioning

Table 1 shows the work partitioning for completing the project. It includes major events, their inputs and outputs, and the summary of the event.

Table 1: Work Partition for the System

Event Name	Input	Output	Summary
Login	Username, Password		User logs in using their McMaster account
Upload Images/Figures	PNG/JPEG files	Uploaded File Reference	User uploads their files to generate alternative text
OCR Text Extraction	Uploaded Images/Figures	Detected Text	System reads the text embedded in the uploaded files
Generate Alternative Text	Uploaded Images/Figures, Extracted OCR, Model Parameters	Generated Alt Text, Quality Metric from Machine Learning (ML) Model	System analyzes the image and extracted OCR data to generate accurate alternative text
View History	User Login, Stored Uploads and Generated Alt Text	List of Previously Generated Alt Text	The system retrieves and displays a user's history of uploaded images along with their associated generated alt text

6.4 Specifying a Business Use Case (BUC)

The project has one primary business use case, which aims to achieve the goal of providing users with visual and cognitive impairments an efficient and accessible way to generate accurate alternative text for academic images and figures.

Preconditions:

- The user has access to the web tool
- The user has files containing diagrams or images requiring alternative text

Scenario:

1. The user logs into the system using their McMaster student credentials
2. The user uploads one or more files (PNG, JPEG) containing diagrams
3. The AI model analyzes the uploaded file(s), performs OCR to extract any visible text, and generates alternative text describing each image/figure accurately
4. Screen readers use the generated alternative text to read aloud and convey the uploaded image
5. The generated alternative text can be edited, copied, or dowloaded as a `.txt` file by the user if needed
6. The user can view previously uploaded files and generated alternative text for future reference

Postcondition:

- The user obtains accurate and accessible alternative text that complies with AODA and WCAG 2.1 standards.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

The diagram below shows the components within the system and how they connect. The components that this project will aim on building include a user

interface, an alternative text generation ML model, a session manager. Furthermore, these components will utilize or communicate with a screen reader software, McMaster Authentication system and external AI/ML Frameworks.

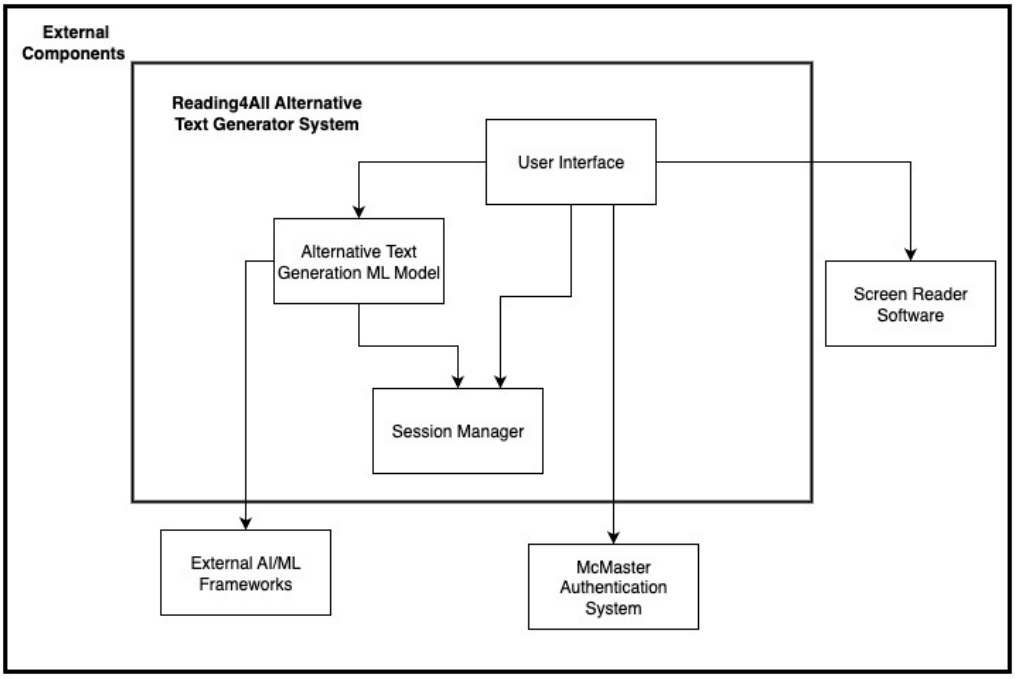

Figure 2: Product Boundary Diagram

8.2 Product Use Case Table

Table 2 summarizes the main product use cases for the systems. For each use case it describes the actors involved, inputs and outputs to the system and the related requirements.

Table 2: Labels and Their Usage

PUC#	PUC Name	Actor(s)	Input & Output(s)	Requirement
PUC 1	Login Using McMaster Credentials	McMaster Student and Faculty , McMaster Authentication System	User Credentials (input), Authentication Results (output)	FR 7
PUC 2	Upload Image	McMaster Student and Faculty	JPEG or PNG (input), Image upload status (output)	FR 1, UHR-EUR 3
PUC 3	Generate Alternative Text	McMaster Student and Faculty, Alternative Text Generation Model	Uploaded image (input), Generated alternative text	FR 2, FR 3
PUC 4	Copy or Download Text	McMaster Student and Faculty	User decision to copy or download (input), text copied to clipboard or downloaded	UHR-PIR 1
PUC 5	View History of Inputted Images and their Alternative Text	McMaster Student and Faculty	User request to view history (input), Display of previously inputted images and their generated text within a session	FR 5

8.3 Individual Product Use Cases (PUC's)

PUC 1: Login Using McMaster Credentials

Trigger: User selects "Login" and is directed to McMaster sign in page.
Preconditions:

- The user is registered person in McMaster system and has valid credentials.

Actors: McMaster Student or Faculty, McMaster Authentica-

tion System.
Outcome: McMaster validates user's credentials are validated and they are given access to system.
Input: McMaster System username and password.
Output: User enters system or an error message is displayed.

PUC 2: Upload Image

Trigger: User selects "Upload Image" and chooses a file
Preconditions:

- User successfully logged into the system.

Actors: McMaster Student or Faculty
Outcome: The selected image is uploaded and stored for later text generation.
Input: Image file (JPEG or PNG)
Output: A confirmation message is displayed if the image was successfully uploaded, or an error message otherwise is displayed.

PUC 3: Generate Alternative Text

Trigger: User selects *Generate Alternative Text* for an uploaded image.
Preconditions:

- A valid image has been successfully uploaded to the system.

Actors: McMaster Student or Faculty.
Outcome: The system generates a descriptive alternative text for the uploaded image.
Input: User selection to generate alternative text
Output: Generated alternative text is displayed to the user.

PUC 4: Copy or Download Generated Alternative Text

Trigger: User selects copy or download .txt after generating alternative text.
Preconditions:

- System has successfully generated alternative text.

- User is satisfied with generated alternative text and has made any desired changes.

Actors: McMaster Student or Faculty.
Outcome: The user receives the alternative text through their preferred method.
Input: User decision to copy or download.
Output: Text is copied or downloaded as .txt file on the users device.

PUC 5: View History of Uploaded Images and Generated Alternative Text

Trigger: User selects the view history option.
Preconditions:

- User is logged in with an active session.
- User has previously uploaded at least one image and generated text within the session.

Actors: McMaster Student or Faculty.
Outcome: The user views a list of their images and the corresponding generated alternative text within the current session.
Input: User request to view session history.
Output: Display of uploaded images and their corresponding generated alternative text.

9 Functional Requirements

9.1 Functional Requirements

FR 1. *The system must accept technical diagrams in the format of JPEG and PNG.*

Rationale: The system must process JPEG/PNG images in order to output alternative text.
Fit Criterion: The system successfully takes as accepts JPEG/PNG images and provides feedback to users when an invalid file type is

inputted.
Priority: High

FR 2. *The system shall generate alternative text of uploaded images.*

Rationale: The main purpose of the system is to make scientific diagrams more accessible by generating better alternative-text.
Fit Criterion: For a set of test diagrams, the alternative text generated must meet the pre-determined criteria.
Priority: High

FR 3. *The system shall output alternative text in a format compatible with screen readers.*

Rationale: Students with disabilities utilize screen readers to access digital content; therefore, the alternative text must be displayed in away that enables screen readers to read it correctly. Furthermore, if the alternative text output format is not compatible with screen readers, then students cannot benefit from the application output.
Fit Criterion: The alternative text output must be readable by at least three commonly used screen readers.
Priority: High

FR 4. *The system shall allow users to edit the outputted alternative texts.*

Rationale: Providing users with an option to edit the outputted text, enables them to adjust the output to better meet their needs if needed.
Fit Criterion: Users can add or delete text in any part of the outputted alternative text and save their changes.
Priority: High

FR 5. *The system shall store and display all inputted images and their generated alternative texts within a session.*

Rationale: Storing previously inputted images and their generated alternative texts, allows users to easily review or reuse them without re-uploading.
Fit Criterion: Users can see view all previously inputted images with their generated alternative texts during the same session.
Priority: Medium

FR 6. *The system must validate users during login to confirm they are McMaster University students.*

Rationale: User verification will ensure that only McMaster University students have access to the system, ensuring that the system is used by the intended users.
Fit Criterion: Users can only gain access to the system features after their McMaster University credentials are successfully validated.
Priority: High

10 Look and Feel Requirements

10.1 Appearance Requirements

LFR-AR 1. *The system must allow all text on the interface to be resized up to 200 %, without any loss of functionality or content.*

Rationale: Allowing text resizing will enable users with low vision to more easily utilize the system. This also ensures the system meets WCAG 2.1 Success Criterion 1.4.4 Resize Text. User verification will ensure that only McMaster University students have access to the system, ensuring that the system is used by the intended users.
Fit Criterion: All text, excluding any captions and images of text can be enlarged to 200 % on a standard browser zoom (ex. Google Chrome) without any overlapping, hidden content, or broken features.
Priority: High

LFR-AR 2. *The system must not use color as the only method to provide information, indicate actions or prompt user input.*

Rationale: Users with color vision deficiencies or other visual impairments may not detect color differences accurately. This also ensures the system meets WCAG 2.1 Success Criterion 1.4.1 Use of Color.
Fit Criterion: Any use of color communicates information to the user or requests information from he user must be appear with text.

Priority: High

LFR-AR 3. *The system must ensure sufficient contrasts of text and images of text.*

Rationale: Sufficient color contrast is important as it enables users with low vision or color vision deficiencies to easily read any system text. This also ensures the system meets WCAG Success Criterion 1.4.3 Contrast (Minimum)
Fit Criterion: All text and images of text in the system interfaces has a contrast ratio of at least 4.5:1
Priority: High

LFR-AR 4. *The system must provide alternative text for all non-text content.*

Rationale: Users with visual impairment often use screen readers to navigate through software systems; therefore, it is essential that all images have sufficient alternative text, so that the purpose of the images can understood. This also ensures the system meets WCAG Success Criterion 1.1.1 Non-text Content .
Fit Criterion: All decorative images and non-text elements have alternative text that communicate their meaning.
Priority: High

10.2 Style Requirements

LFR-SR 1. *The system interface must follow a simple and modern design style.*

Rationale: A simple interface will improve the systems usability as it better highlights the system's features, while also ensuring the system is visually appealing.
Fit Criterion: The system uses a clean layout with a maximum of three colors, consistent font styles and sizes, as well as only has key design elements that support usability.
Priority: High

LFR-SR 2. *The system interface must use McMaster University branding while maintaining accessibility standards and a modern style.*

Rationale: As the system is targeted towards McMaster University students, using the schools branding will build trust with users and ensure the system aligns with McMaster's identify. However, using McMaster branding must not interfere with usability and accessibility criteria..
Fit Criterion: The system interface includes McMaster University' official logo and meets the WCAG 2.1 contrast and non-text content success criteria.
Priority: High

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

UHR-EUR 1. *The system interface must allow users to efficiently use the system features.*

Rationale: It is important the users can quickly access and use the system features, as they may be generating multiple alternative text outputs in a single session.
Fit Criterion: Users can upload images to the system and generate alternative text in 5 steps or fewer.
Priority: High

UHR-EUR 2. *The system interface must be easy for users to remember how to use after not using it for some time.*

Rationale: Users should be able to quickly recall how to use the system without needing to relearn the features. An intuitive design will make it easier for returning users to find and use key features.
Fit Criterion: Users who have not used the system in a month, can successfully login, upload an image and generate alternative text within 5 minutes, without needing any assistance.
Priority: Medium

UHR-EUR 3. *The system interface must provide users with clear and immediate feedback for all actions.*

Rationale: Providing the users with feedback ensures they understand of the outcome of their actions and whether they are using the system correctly. This reduces confusion and makes users more confident while using the system.
Fit Criterion: The system provides textual feedback within 1 second after a user interaction, such as uploading an image.
Priority: High

UHR-EUR 4. *The system interface must provide clear instructions, prevent common errors and allow users to easily correct them.*

Rationale: Providing easy to follow instructions will help ensure that users can easily use the system features and prevent errors. Additionally, if a user makes a mistake, they should easily be able to revert it.
Fit Criterion: In user testing, at least 80% of users can complete tasks without errors. When a user error occurs, the system explains the issue and how to recover within 2 seconds.
Priority: High

11.2 Personalization and Internationalization Requirements

UHR-PIR 1. *The system interface must allow users to choose how generated alternative text is stored or copied.*

Rationale: Providing users with the option to either copy generated text to their clipboard or download it as file, helps tailor the output to the users specific needs.
Fit Criterion: After generating the alternative text users can choose to "Copy to Clipboard" or "Download as .txt" from the interface and system successfully completes the chosen option.
Priority: High

11.3 Learning Requirements

UHR-LR 1. *The system must be easy for low-vision users to learn and operate with screen readers.*

Rationale: The system should be intuitive for users with low

vision to use without prior training. Additionally, the system being highly compatible with screen readers, allows users to more easily navigate and use the system.
Fit Criterion: In user testing, at least 90% of first time users with low vision using a screen reader can upload an image and generate alternative text within 5 minutes without assistance.
Priority: High

11.4 Understandability and Politeness Requirements

UHR-LR 1. *The system must only display essential information and hide all technical details.*

Rationale: The system should only communicate the information needed to use the system. Displaying any technical details may cause the user to be confused and make the system less usable.
Fit Criterion: In user testing, users do not encounter any technical terms, code outputs or information that is not relevant to them.
Priority: High

11.5 Accessibility Requirements

UHR-AR 1. *The system must meet the WCAG 2.1 Level AA accessibility standards.*

Rationale: The Accessibility for Ontarians with Disabilities Act (AODA) requires organizations to meet WCAG 2.0 Level AA for web tools. Therefore, meeting WCAG 2.1 Level AA ensures the system meets AODA standards and is accessible for users with disabilities.
Fit Criterion: The system will be evaluated using an accessibility testing tool such as Pope Tech and Wave Web Aim to ensure WCAG 2.1 criteria is met.
Priority: High

UHR-AR 2. *The system must accept keyboard input for navigation.*

Rationale: Many users, including those with disabilities, use

keyboard inputs to navigate through applications, the system must support this as a way to navigate.
Fit Criterion: Users can navigate to all the main functions and areas of the system using their keyboard.
Priority: High

12 Performance Requirements

12.1 Speed and Latency Requirements

PR-SL 1. *The tool shall generate alt-text for uploaded images within a reasonable time frame.*
Rationale: Ensures users, including those using assistive technologies, do not experience delays that hinder accessibility.
Fit Criterion: The system shall return generated alt-text within **3 seconds** for images $\leq$2 MB and within **8 seconds** for images $\leq$10 MB under normal load conditions.
Priority: High

PR-SL 2. *The web interface shall load and render accessibility components efficiently.*
Rationale: Improves user experience and responsiveness for screen-reader users and keyboard navigation.
Fit Criterion: All interactive elements shall respond within **300 ms** of user input under typical conditions.
Priority: Medium

12.2 Safety-Critical Requirements

PR-SCR 1. *The tool shall ensure that no personally identifiable data from uploaded images is stored or shared without consent.*
Rationale: Protects user privacy and adheres to ethical AI standards.
Fit Criterion: Uploaded images are deleted from temporary storage within **60 seconds** of processing unless explicitly saved by the user.
Priority: High

PR-SCR 2. *The tool shall not produce alt-text containing offensive, biased, or harmful language.*

Rationale: Ensures ethical AI output and inclusivity.
Fit Criterion: 0 % of generated outputs shall contain content flagged by moderation filters as offensive or biased.
Priority: High

PR-SCR 3. *The interface shall adhere to WCAG 2.1 Level A accessibility guidelines to prevent stress or strain on users' eyes and ensure comfortable interaction.*
Rationale: Provides a visually safe, inclusive experience for all users, including those with visual or cognitive impairments.
Fit Criterion: Verified through front-end accessibility testing that confirms conformance with WCAG 2.1 Level A success criteria.
Priority: High

12.3 Precision or Accuracy Requirements

PR-PAR 1. *The generated alt-text shall adequately describe the image content with minimal omissions or irrelevant details.*
Rationale: Ensures the description fulfills its accessibility purpose.
Fit Criterion: At least **85 %** of outputs rated "Sufficient" or better on the sufficiency scale by testers.
Priority: High

PR-PAR 2. *The alt-text shall maintain appropriate length and readability.*
Rationale: Prevents overly short or verbose outputs that reduce usability.
Fit Criterion: ≥ 90 % of outputs rated "Proper Length" on the user-testing scale.
Priority: Medium

PR-PAR 3. *The overall accessibility and usability of the alt-text shall be acceptable to testers.*
Rationale: Evaluates real-world effectiveness of generated descriptions.
Fit Criterion: Median user rating ≥ 3 ("Mostly Accessible/Usable") on the 0–3 or 0–4 scales; no outputs below 2.
Priority: Medium

12.4 Robustness or Fault-Tolerance Requirements

PR-RFT 1. *The system shall gracefully handle unsupported or corrupted image inputs.*
Rationale: Prevents crashes and maintains system stability.
Fit Criterion: Invalid files trigger a clear error message within 2 seconds without interrupting service.
Priority: High

PR-RFT 2. *The backend shall recover automatically from isolated process failures.*
Rationale: Ensures continued operation without developer intervention.
Fit Criterion: System recovers within 5 seconds after fault detection.
Priority: High

12.5 Capacity Requirements

PR-CR 1. *The system shall support limited concurrent usage suitable for a proof-of-concept deployment.*
Rationale: Demonstrates feasibility and reliability for initial testing without production-level scaling.
Fit Criterion: Supports at least **5 simultaneous requests** with response times ≤ 10 seconds.
Priority: Medium

PR-CR 2. *Storage shall accommodate pilot testing datasets.*
Rationale: Ensures smooth prototype validation without capacity issues.
Fit Criterion: The system can temporarily store metadata for up to **500 images per day** without data loss.
Priority: Low

12.6 Scalability or Extensibility Requirements

PR-SER 1. *The architecture shall allow integration of improved ML models or multilingual capabilities in future phases.*
Rationale: Enables progressive enhancement and future accessibility expansion.
Fit Criterion: New models or language modules can be incorporated

without restructuring existing components.
Priority: Medium

12.7 Longevity Requirements

PR-LR 1. *The codebase shall be maintainable and adaptable to updates in WCAG guidelines, Python libraries, and ML frameworks.*
Rationale: Ensures long-term usability and compliance even after the pilot phase.
Fit Criterion: Minor updates or migrations require ≤ 2 person-days per quarter.
Priority: Medium

PR-LR 2. *The prototype shall maintain compatibility with at least the next two Python releases.*
Rationale: Ensures sustainability of the pilot for educational and testing purposes.
Fit Criterion: Verified through annual testing on supported Python versions.
Priority: Low

13 Operatinonal and Environment Requirements

The following operating environment requirements define the physical, network, interfacing, and deployment contexts in which the Reading4All system must operate reliably and efficiently.

13.1 Expected Physical Environment

OER-EP1. *The system shall be operable on standard devices (laptops, desktops, or servers) running common operating systems such as Windows, macOS, or Linux, or deployed to standard cloud environments (e.g., GCP, AWS, Azure).*

Rationale: Reading4All is intended for academic and institutional use, where users rely on a variety of systems. Supporting cross-platform and cloud deployment ensures accessibility and

flexibility.
Fit Criterion: The system runs successfully across all major operating systems and supported cloud environments without compatibility errors.
Priority: High

OER-EP2. *The system shall operate effectively under normal indoor conditions typical of academic or office environments.*

Rationale: As the system is designed for digital academic use within classrooms and offices, no specialized environmental setup is required.
Fit Criterion: The tool performs consistently at standard indoor temperatures (10°C–35°C) and lighting levels.
Priority: Low

13.2 Wider Environment Requirements

OER-WE1. *The system shall comply with accessibility and data privacy regulations, including AODA, WCAG 2.1 Level AA, and relevant institutional privacy policies such as FIPPA.*

Rationale: Compliance ensures digital inclusion and privacy protection while aligning with university and governmental accessibility mandates.
Fit Criterion: Independent evaluation verifies AODA and WCAG 2.1 Level AA compliance, and no accessibility blockers are found during testing.
Priority: High

OER-WE2. *The system shall operate effectively within standard academic network environments with stable internet connectivity.*

Rationale: Reading4All depends on cloud-based inference models for text generation, which require reliable network access.
Fit Criterion: The system maintains consistent API communication over typical university Wi-Fi or Ethernet with upload speeds of at least 10 Mbps.
Priority: Medium

13.3 Requirements for Interfacing with Adjacent Systems

OER-IAS1. *The system shall integrate with learning management systems (LMS) such as Avenue to Learn (D2L Brightspace) for image uploads and alt-text retrieval.*

Rationale: Direct LMS integration streamlines accessibility workflows for instructors and students.
Fit Criterion: Reading4All successfully connects to at least one LMS via secure API endpoints approved by the institution.
Priority: High

OER-IAS2. *The system shall support interoperability with assistive technologies such as screen readers (e.g., NVDA, JAWS, and VoiceOver).*

Rationale: Screen-reader compatibility ensures generated text can be read aloud for visually impaired users.
Fit Criterion: Descriptions produced by Reading4All are correctly parsed and read by major screen readers without formatting issues.
Priority: High

OER-IAS3. *The system shall support importing diagrams in common formats including JPG, JPEG, and PNG.*

Rationale: Supporting a range of standard formats allows instructors to use materials from multiple academic sources.
Fit Criterion: The system processes at least one valid sample of each format and produces accurate alternative text.
Priority: Medium

OER-IAS4. *The system shall optionally interface with automated accessibility validation tools (e.g., WAVE or Axe).*

Rationale: Integration with automated validators aids instructors in verifying alt-text accessibility compliance.
Fit Criterion: Validation reports are successfully generated and accessible through the user interface.
Priority: Low

13.4 Productization Requirements

OER-PR1. *The system shall be deployable as both a web application and an API service for institutional integration.*

Rationale: Dual deployment ensures accessibility for individual users and organizations integrating accessibility workflows.
Fit Criterion: A hosted web app and functional REST API are accessible and validated through institutional testing.
Priority: High

OER-PR2. *The system shall securely store configuration and model parameters in a version-controlled and scalable environment.*

Rationale: Proper configuration management ensures reproducibility and secure model versioning.
Fit Criterion: Configuration files are encrypted, tracked via version control, and retrievable for rollback or update.
Priority: Medium

13.5 Release Requirements

OER-RL1. *All major functionalities (image analysis, text generation, and accessibility validation) must be implemented, tested, and verified prior to release.*

Rationale: Ensures reliable, inclusive functionality before public or institutional rollout.
Fit Criterion: Verification and validation (V&V) documentation confirms that each functional requirement is successfully tested.
Priority: High

OER-RL2. *The system must be ready for release by March 18, 2026, aligned with the McMaster University SFWRENG 4G06 Capstone final demonstration schedule.*

Rationale: Aligns release timing with Capstone evaluation and stakeholder presentation.
Fit Criterion: The final deliverable is fully functional, accessible, and deployed for the 2026 demonstration.

Priority: Medium

14 Maintainability and Support Requirements

These requirements ensure the Reading4All system remains maintainable, well-supported, and adaptable to future accessibility standards and technical advancements.

14.1 Maintenance Requirements

MS-MNT1. *The system shall be designed with modular components to allow independent updates to subsystems such as the front-end interface, image-analysis model, language-generation model, or accessibility validation module.*

Rationale: Modularity minimizes maintenance effort and prevents regressions when individual components are updated.
Fit Criterion: Modules are separately deployable and testable with independent configuration files.
Priority: High

MS-MNT2. *All source code shall include inline documentation and external developer documentation stored in the project's GitHub Wiki or README.*

Rationale: Comprehensive documentation ensures maintainability and smooth developer onboarding.
Fit Criterion: Each function and class includes descriptive docstrings, and setup steps are verified by a peer review checklist.
Priority: Medium

MS-MNT3. *The system shall implement a CI/CD pipeline with automated regression and accessibility tests.*

Rationale: Continuous testing reduces maintenance time and ensures stability after updates.
Fit Criterion: Each code merge triggers automated unit, integration, and WCAG/AODA compliance tests.
Priority: High

14.2 Supportability Requirements

MS-SUP1. *The system shall allow users to report issues or feedback directly through the interface without revealing personal identifiers.*

Rationale: Enables user-driven improvements while maintaining privacy.
Fit Criterion: Feedback forms are logged securely without user PII and assigned unique issue IDs.
Priority: Medium

MS-SUP2. *The system shall log performance and error metrics for debugging and continuous improvement.*

Rationale: Monitoring key metrics helps identify bottlenecks and optimize performance.
Fit Criterion: Logs record API calls, latency, error rates, and model confidence distributions.
Priority: Medium

14.3 Adaptability Requirements

MS-AD1. *The system shall support integration of new AI models or components through a standardized interface schema.*

Rationale: A consistent interface allows easy replacement or upgrading of AI components.
Fit Criterion: New modules adhere to the existing JSON I/O structure and pass automated compatibility checks.
Priority: High

MS-AD2. *The system shall allow configuration updates without modifying source code.*

Rationale: Enables rapid adaptation to new accessibility or institutional requirements.
Fit Criterion: All configurable parameters are stored in editable YAML or JSON files.
Priority: Medium

MS-AD3. *All major updates shall be versioned and documented each academic term to ensure reproducibility.*

Rationale: Academic projects require transparent traceability across semesters.
Fit Criterion: Version logs include change summaries, affected modules, and verification results.
Priority: Medium

15 Security Requirements

15.1 Access Requirements

SR-AR 1. *The system shall restrict access exclusively to McMaster University users through institutional Single Sign-On (SSO) authentication.*
Rationale: Restricting access ensures only authorized users within McMaster can use the system during the pilot phase, reducing the risk of unauthorized use or data exposure.
Fit Criterion: All users must log in using verified McMaster SSO credentials before accessing the platform. Unauthenticated requests are automatically rejected.
Priority: High

SR-AR 2. *All actions performed by users shall be tied to their authenticated session.*
Rationale: Linking actions to a user's authenticated identity enables traceability and controlled access to system features.
Fit Criterion: Each upload or alt-text generation event is associated with a unique McMaster user ID through SSO session tracking.
Priority: Medium

15.2 Integrity Requirements

SR-IR 1. *All communication between the frontend, backend, and machine learning services shall use encrypted HTTPS (TLS 1.2 or higher).*
Rationale: Encryption prevents interception and tampering of sensitive data such as authentication tokens or image files.
Fit Criterion: All HTTP requests must be redirected to HTTPS; unencrypted requests are rejected by the web server.
Priority: High

SR-IR 2. *Uploaded images shall remain unmodified during processing and analysis.*
Rationale: Preserving file integrity ensures consistent and accurate generation of alt-text.
Fit Criterion: File hash comparison verifies that image files remain identical throughout the upload and analysis process.
Priority: High

15.3 Privacy Requirements

SR-PR 1. *Uploaded images shall be deleted immediately after processing is complete.*
Rationale: Protects user privacy and ensures compliance with institutional data governance policies.
Fit Criterion: Uploaded files are stored temporarily in memory or on a secure local directory and deleted within 60 seconds after alt-text generation.
Priority: High

SR-PR 2. *Generated alt-text shall not contain personally identifiable information (PII) or sensitive content.*
Rationale: Prevents disclosure of private information and ensures responsible AI usage.
Fit Criterion: The model output is passed through a content moderation filter that rejects or flags any alt-text containing PII or inappropriate language.
Priority: Medium

15.4 Audit Requirements

SR-AU 1. *System usage logs shall record authentication events, uploads, and generation activities for accountability and debugging.*
Rationale: Audit logs enable traceability, assist in debugging, and ensure ethical research practices.
Fit Criterion: Logs record timestamps, user IDs, and non-sensitive metadata while excluding image or generated text content.
Priority: Medium

SR-AU 2. *Access to audit logs shall be restricted to authorized project administrators.*
Rationale: Limits access to potentially sensitive operational data and protects user confidentiality.
Fit Criterion: Logs are stored in a restricted-access directory with read permissions granted only to administrators.
Priority: Medium

15.5 Immunity Requirements

SR-IM 1. *The system shall validate and sanitize all uploaded files to prevent malicious or unsupported file types.*
Rationale: Protects against injection attacks, corrupted uploads, or execution of non-image files.
Fit Criterion: Only files with valid image types (.png, .jpg, .jpeg, .svg, .webp) are accepted; unsupported or script files are automatically rejected.
Priority: High

SR-IM 2. *The system shall block access from networks or domains outside McMaster University's infrastructure.*
Rationale: Restricting network access minimizes exposure to external threats during the proof-of-concept phase.
Fit Criterion: Requests must originate from verified McMaster SSO tokens or IP ranges associated with university networks.
Priority: High

16 Cultural Requirements

The following list conists of cultural requirements the system shall follow:

CR 1. *The system shall generate alternative text using neutral and inclusive language appropriate for academic environments.*
Rationale: Ensures that generated content is respectful to diverse cultural and educational backgrounds.
Fit Criterion: Generated alt text contains no culturally biased, exclusionary, or inappropriate terminology.
Priority: High

CR 2. *The system shall avoid using culturally specific references unless the visual content explicitly requires it.*
Rationale: Prevents misinterpretation and maintains accessibility for a wide audience.
Fit Criterion: Alt text focuses on visual description and context without unnecessary cultural assumptions.
Priority: Medium

CR 3. *The system shall use professional and educationally appropriate tone in all generated content.*
Rationale: Maintains usability across academic departments and contexts.
Fit Criterion: Outputs remain formal, non-colloquial, and context-relevant.
Priority: Medium

17 Compliance Requirements

17.1 Legal Requirements

CR-LR 1. *The system shall comply with AODA standards for alternative text generation.*
Rationale: Ensures the tool supports institutional accessibility requirements and legal obligations.
Fit Criterion: All generated alt text meets WCAG 2.1 Level AA criteria for accuracy, clarity, and relevance.
Priority: High

17.2 Standards Compliance Requirements

CR-SCR 1. *The system shall follow institutional privacy and data-handling guidelines for uploaded teaching materials.*
Rationale: Prevents unauthorized distribution or mishandling of academic content.
Fit Criterion: No files are stored beyond active use unless explicitly authorized; logs exclude proprietary content.
Priority: High

CR-SCR 2. *The system shall provide verifiable documentation or statements of compliance upon request.*
Rationale: Facilitates audits, approvals, and integration into university workflows.
Fit Criterion: A compliance overview document or help section is available to stakeholders.
Priority: Medium

18 Open Issues

This section outlines unresolved questions and decisions that may impact the overall success of the system. The following items require additional research, testing, or discussion to ensure the project's successful completion.

- The ML/AI model architecture the team will use to generate alternative text will need research and testing to ensure optimal accuracy and correctness.
- The optimal length of the generated alternative text requires further research to determine how many characters provide an accurate description without causing confusion or distracting from the main idea of the diagram.

19 Off-the-Shelf Solutions

This section identifies existing tools, reusable components, and research products that can support or inspire the development of Reading4All. These solutions address various aspects of the system's vision analysis, language generation, and accessibility workflows.

19.1 Ready-Made Products

- **Google Cloud Vision API:** A commercial computer vision service capable of detecting objects, text, and structural elements within images. It can assist in preliminary diagram segmentation or optical character recognition for Reading4All's visual analysis component.

- **Microsoft Azure Cognitive Services:** Offers image captioning and scene-description APIs that generate textual summaries of visual content. These can serve as a performance benchmark for Reading4All's AI-generated academic alt text.
- **OpenAI GPT-4V (Vision):** A multimodal AI model capable of interpreting images and generating context-aware captions. It can be evaluated for use in academic diagrams and as a reference for Reading4All's natural-language generation pipeline.

19.2 Reusable Components

- **Hugging Face Transformers:** An open-source library offering pretrained multimodal models (e.g., BLIP-2, CLIP, and ViT-GPT2). These can be fine-tuned to identify structural and semantic relationships within STEM diagrams.
- **Albumentations:** A Python library for data augmentation, enhancing dataset diversity for diagram recognition tasks. It can be reused in Reading4All's preprocessing pipeline to improve robustness.
- **Pandas + Matplotlib AltText Plugin:** An open-source extension that generates descriptive alt text for statistical plots. Its modular logic can guide Reading4All's diagram-specific description component.

19.3 Products That Can Be Copied

- **Chart2Text:** A benchmark system that automatically converts data visualizations such as bar charts and line graphs into natural-language summaries. Its approach to aligning visual features with linguistic structures provides a strong foundation for Reading4All's diagram-description workflow.
- **SciA11y:** A research initiative by the Allen Institute for AI that generates accessible figure descriptions for scientific papers. Its methods for extracting captions, metadata, and contextual relationships between visual and textual elements align closely with Reading4All's academic accessibility objectives.

20 Tasks

20.1 Project Planning

- **Development Approach** The Reading4All team will follow an iterative agile workflow emphasizing continuous improvement and regular feedback from stakeholders. Development will progress through a series of short sprints that each target measurable goals, allowing flexibility as requirements evolve. The general process will involve:

  1. Requirement analysis and refinement of functional specifications
  2. Sprint planning and backlog prioritization
  3. Incremental coding and integration
  4. Verification, validation, and unit testing
  5. Periodic reviews with the project supervisor and accessibility lead
  6. System deployment and documentation finalization

- **Key Tasks**

  - Confirm system architecture and interface specifications
  - Establish a shared GitHub repository with branching standards
  - Configure CI/CD automation through GitHub Actions
  - Implement the vision module for diagram segmentation and labeling
  - Integrate the language generation component for alt-text synthesis
  - Conduct user testing with instructors and accessibility specialists
  - Evaluate outputs for WCAG 2.1 compliance and descriptive accuracy

- **Resource Estimates** The project involves a five-member team responsible for design, development, testing, and reporting. Shared tools will include:

  - Cloud computing resources (GPU-based inference environment)
  - Version control and issue tracking using GitHub Projects
  - Datasets of academic diagrams and verified alternative texts

- **Key Considerations**
  - Early integration testing to avoid dependency conflicts
  - Dataset bias evaluation to ensure inclusive model outputs
  - Consistent communication with the faculty advisor and accessibility consultant
  - Risk reduction through checkpoint testing and documentation reviews
- **Documentation Process**
  - Maintain all documentation collaboratively within the GitHub Wiki
  - Use meaningful commit messages and structured pull requests
  - Require automated testing to pass before merging any code
  - Apply peer review and version tagging for every major deliverable
  - Archive final reports and test results for supervisor review

20.2 Planning of the Development Phases

Deliverables and Tentative Schedule

- Problem Statement, Proof of Concept, and Development Plan — **Week 4 (September 24, 2025)**
- Software Requirements Specification (SRS) and Hazard Analysis — **Week 6 (October 8, 2025)**
- Verification and Validation Plan — **Week 8 (October 22, 2025)**
- Design Document (Revision 0) — **Week 10 (November 5, 2025)**
- Proof of Concept Demonstration — **Weeks 11–12 (November 12–19, 2025)**
- Design Document (Revision 1) — **Week 16 (January 21, 2026)**
- Revision 0 Demonstration — **Weeks 18–19 (February 4–11, 2026)**

- Verification and Validation Report — **Week 22 (March 4, 2026)**
- Final Demonstration (Revision 1) — **Week 24 (March 18, 2026)**
- Capstone EXPO — **Week 26 (April 1, 2026)**
- Final Documentation Submission — **Week 26 (April 1, 2026)**

Sprint and Review Cycle

- Each sprint will run for approximately two weeks, concluding with a progress review.
- Development milestones will be validated through automated testing and peer evaluation.
- Supervisor and accessibility advisor feedback will guide milestone adjustments and task prioritization.

By adopting this structured development plan, the team will ensure that Reading4All evolves into a reliable and maintainable system capable of producing accurate, accessible, and pedagogically useful alt text for academic diagrams.

21 Migration to the New Product

21.1 Requirements for Migration to the New Product

MNP-RMNP 1. *The system shall support a phased implementation to allow gradual adoption while minimizing disruptions.*
Rationale: Reduces organizational risk and allows controlled testing during rollout.
Fit Criterion: Each phase is deployed and validated independently before progressing to the next.
Priority: High

MNP-RMNP 2. *The organization shall operate the new system in parallel with the old product for a defined transition period.*
Rationale: Ensures continuity and confirms correct operation before full cutover.

Fit Criterion: Parallel operation lasts until all critical functions pass acceptance testing.
Priority: High

MNP-RMNP 3. *The system shall provide procedures and tools for manual backup during transition.*
Rationale: Maintains operational stability during migration.
Fit Criterion: Backup processes are documented, tested, and accessible to staff.
Priority: Medium

MNP-RMNP 4. *The transition plan shall identify and schedule major components and release phases.*
Rationale: Guides project planning and resource allocation.
Fit Criterion: A migration timeline with milestones and dependencies is documented.
Priority: Medium

21.2 Data That Has to be Modified or Translated for the New System

This section does not apply to this project as there is no current system to replace, thus, no data at all.

22 Costs

The total cost of developing this project is primarily based on the effort contributed by the student development team and faculty supervisors. As the project utilizes open-source technologies (e.g., Python, TensorFlow, Flask, and React) and university-hosted infrastructure, no direct monetary expenditure is incurred. The project is scheduled to be completed within the academic term (MVP ready by April 2026), and resource allocation is focused on efficient time management and workload balancing rather than financial cost.

22.1 Metrics for Estimation

To estimate overall development cost in terms of time and effort, the following key metrics have been considered:

- Number of image input/output workflows supported by the tool.
- Number of core functional requirements (e.g., image upload, alt-text generation, user authentication, evaluation metrics).
- Number of non-functional requirements (e.g., accessibility compliance, latency, privacy, and scalability).
- Number of deliverables and milestones within the development timeline.
- Team size and individual role distribution (frontend, backend, model integration, documentation).

22.2 Estimation Approach

Each deliverable has been estimated based on the effort required to implement, test, and document it within the given academic term. The estimates are derived from prior experience with similar web-based machine learning projects and adjusted for the learning and research effort required to integrate accessibility and WCAG compliance features. Time allocation also accounts for model fine-tuning, front-end accessibility testing, and usability evaluation with pilot users.

22.3 Cost Breakdown

• **Development Effort:** Based on a team of four student developers and one faculty supervisor, the estimated time allocation per team member is as follows:

- Initial research, planning, and requirement analysis: **40 hours per team member.**
- Model integration and backend implementation: **120 hours per team member.**

- Frontend development and accessibility compliance: **80 hours per team member.**
- Testing, debugging, and refinement: **60 hours per team member.**
- Documentation and presentation preparation: **30 hours per team member.**

Total estimated effort: 330 hours per team member.
• **Tools and Software:** All software components used in the project (Python, Flask, React, TensorFlow, Machine Learning Models, and GitHub) are open-source or free for academic use. Therefore, there are no direct licensing or software procurement costs.
• **Testing Environment:** Testing will be conducted using McMaster-hosted or open-source platforms for model inference and user testing. The team might decide to conduct user testing and award attendees 10 dollar Amazon.com gift cards as a reward for participation. No additional hardware purchases are required beyond existing student laptops and cloud compute credits provided for academic purposes.

22.4 Estimated Cost

The total development effort is estimated at approximately **1,320 person-hours** across all team members (4 × 330 hours). Assuming an average academic hourly equivalent of **$20 per hour** for estimation purposes, the notional cost of development would be approximately **$26,400 CAD**.

However, as this project is conducted as part of a capstone academic course and leverages free university and open-source resources, the **actual monetary cost is $0 CAD**. The effective cost of the project lies entirely in time, research, and human effort required to meet the performance, accessibility, and security requirements outlined in previous sections.

23 User Documentation and Training

23.1 User Documentation Requirements

1. User Manual

- **Purpose**: The user manual will serve as a user guide and provide detailed information and instructions on the final product and how to use it effectively
- **Target Audience**: Academic students, instructors, and other professionals
- **Content**: Web tool navigation and instructions, usage examples, product purposes, and best practices

23.2 Training Requirements

Users of the final product will require minimal to no training as we aim to ensure that the tool is as accessible, simple, and intuitive as possible. For any additional guidance, a user manual will be created along with any relevant tutorials on how to use the features within the web tool.

24 Waiting Room

This section lists potential ideas and features that are out of scope for the current project, however, may be valuable for future updates.

- Support for multilingual alternative text generation (e.g., French and Spanish)
- A browser extension that automatically generates alternative text on websites or learning platforms (e.g. *D2L*) using our model
- Compatibility with mobile platforms to extend accessibility across users' preferred devices

25 Ideas for Solution

This section discusses potential ways to acheive some of the functionality discussed throughout this report, including image upload and processing, alternative text generation and session history. These ideas have been documented so they can be referenced later during development.

Image Upload and Processing This functionality can be achieved through

a front-end interface, where users are prompted to upload an image using an upload button or by dragging their file into the drop box. Furthermore, to minimize errors, this will only allow JPEG and PNG image files. Once the image has been uploaded, it will be displayed to the user with the image file name, so users can confirm the correct file was chosen. If the upload fails, the system will display an error message explaining the issue. Furthermore, this can be achieved using the HTML5 File API, which supports reading and processing file data, specifically obtained through input or drag and drop.

Alternative Text Generation This functionality can be implemented using a vision-language model (VLM), which combines natural language models with computer vision. The model can learn from both images and text to solve various problems. The model can be trained using sample scientific images, paired with examples of descriptive alternative text, allow it to generate accurate and high quality descriptions for new images.

Session History After the user is satisfied with the generated alternative text, the system will store the image and its final description in the browser's session storage as a JSON record. This allows the data to be stored temporarily and can easily be displayed to the user when they request their history.

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:

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. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?
4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.
5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

Nawaal Fatima - Reflection

1. **What went well while writing this deliverable?**
   Our group divided the work efficiently, which made the writing process smoother. I found that once we agreed on the structure, it became easier to contribute my part because everyone had a clear understanding of what they were responsible for. We all came together towards the end and reviewed everyone's parts, ensuring coherency and consistency in our writing. Communication also went well, and we were able to clarify uncertainties quickly through discussions. Personally, I felt more confident writing my section because I understood how my contribution fit into the overall deliverable.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   One challenge I experienced was making sure my writing aligned with the tone and level of detail the rest of the team was using. At first, it was hard to tell how formal or detailed certain sections should be. I resolved this by checking in with my teammates' writing and reading over their parts so that my section matched in style. Another minor pain point was managing time alongside other coursework, but planning out smaller chunks helped me stay on track. Furthermore, my computer glitched during an update and I had to rewrite all of my sections which was super frustrating and delayed some teammate's sections. In the future I will ensure to commit all drafts to a remote branch before updating my computer.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?**
   A noticeable portion of the requirements came from talking to Ms. Sui and from my experiences working with her over the past three years. Even though we don't always interact with the customers directly, speaking to Ms. Sui and imagining how potential users would interact with the system helped shape several of the requirements. I'd estimate that roughly half of the requirements were influenced by those conversations or by feedback from people who could represent the end users.

4. **Which of the courses you have taken, or are currently taking, will help your team be successful with your capstone project?**
   Several courses connect directly to this project. Software engineering and requirements-focused courses helped with understanding how to draft clear specifications and think about users' needs. Any design or project-based courses gave me experience working in teams and coordinating deliverables. Courses that covered testing, human-computer interfaces, documentation, and development processes also helped ensure we follow good practices throughout the capstone.

References

Microsoft, "Azure AI Vision—Image Analysis," Microsoft Docs/Learn. Accessed: Oct. 9, 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-image-analysis?tabs=4-0

AltText.ai, "AltText.ai—Automatic Image Alt Text Generation," AltText.ai. Accessed: Oct. 9, 2025. [Online]. Available: https://alttext.ai/

accessiBe, "Building a more accessible web together", accessiBe. Accessed: Oct. 9, 2025. [Online]. Available: https://www.accessibe.com/about