Software Requirements Specification for Bridging Gaps: AI for Diagram Accessibility

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

October 28, 2025

Contents

Symbolic Constants

Name	Value
T_ALT_GEN_SMALL	3 seconds(s)
T_ALT_GEN_LARGE	8 s
T_UI_RESP	300 miliseconds (ms)
R_SUFFICIENCY	85%
R_LENGTH	90%
R_USABILITY_MEDIAN	3 (rating)
R_USABILITY_MIN	2 (rating)
T_ERROR_HANDLE	2 s
T_RECOVERY	5 s
CAP_CONCURRENT	2 requests
CAP_STORAGE	500 images/day
MAINT_TIME	2 person-days/quarter
COMPAT_VERSIONS	2 releases
IMG_SIZE_BIG	10 MEGABYTES (MB)
IMG_SIZE_SMALL	2 MEGABYTES
TLS_VERSION	1.2
FILE_DELETE_TIME	60 s
FILE_TYPES	.png, .jpg, .jpeg, .svg, .webp
NETWORK_SOURCE_POLICY	McMaster SSO tokens or IP ranges only
TEAM_SIZE	5 students
HOURS_RESEARCH	40 hours
HOURS_BACKEND	120 hours
HOURS_FRONTEND	80 hours
HOURS_TESTING	60 hours
HOURS_DOCS	30 hours
HOURS_TOTAL	330 hours
HOURS_PROJECT	1,320 person-hours
COST_PER_HOUR	$20/hour
COST_TOTAL	$26,400 CAD
COST_ACTUAL	$0 CAD
COST_INCENTIVE_MIN	$100 CAD
COST_INCENTIVE_MAX	$150 CAD
MAX_ZOOM_PERCENTAGE	200%

Table continues on next page

MIN_CONTRAST_RATIO	4.5:1
MAX_UPLOAD_STEPS	5 steps
MAX_MINUTES	5 minutes
USERS_SUCCESS_PERCENT	80%
MAX_ERROR_RECOVER	2 seconds
LEARNING_PERCENT	90%
MAX_LEARNING_MINUTES	5 minutes
MIN_COMPENSATION_DOLLARS	$100 CAD
MAX_COMPENSATION_DOLLARS	$150 CAD
LATEST_RELEASES_NUM	3
MOST_COMMON_SR	3
SFWR_RELEASES	2

Revision History

Date	Version	Notes
October 10 2025	1.0	Initial Version of SRS

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, and system features that enable reliable alternative (alt) text generation for technical diagrams.

1.2 Scope of Project

Reading4All is an artificial intelligence (AI)/machine learning (ML) tool that provides detailed and contextually aware alternative text for complicated technical graphics, notably those found in postsecondary Science, Technology, Engineering, and Mathematics (STEM) course materials. The system combines machine learning 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 **Web Content Accessibility Guidelines (WCAG) 2.1** accessibility standards.
- Improve the learning experiences and inclusion of students with disabilities 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 alt text generation.
- Generated alt text that are compatible with existing accessibility tools.

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 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.

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-Volere Template*, McMaster University

2 Stakeholders

The project stakeholders consist of people who have a need or interest, whether direct or indirect, for alternative text generation for visual and 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 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 Ms. Sui's position at McMaster University, she 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: 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: N/A

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 close the gap in 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 Maintainance 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 LATEST_RELEASES_NUM 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 MAX_COMPENSATION_DOLLARS canadian dollars 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 MIN_COMPENSATION_DOLLARS and MAX_COMPENSATION_DOLLARS 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

Accuracy The degree to which generated descriptions capture the image's content correctly.

AI (artificial intelligence) Techniques that enable computers to perform tasks that normally require human intelligence.

Alt Text (Alternative Text) Textual description of non-text content such as images that allow accessibility tools such as screen readers to convey the content.

AODA (Accessibility for Ontarians with Disabilities Act) Ontario law aimed at improving accessibility for people with disabilities by removing and preventing barriers when designing.

API (Application Programming Interface) Rules and protocols that allows different software programs to communicate with each other.

Backend Server components handling processes of an application that users don't see.

Benchmarking The comparing of performance or quality of one's system against known systems or datasets.

Contrast Ratio Luminance difference between text and background required by WCAG 2.1.

Dataset Bias Systematic skew in training data that can harm fairness or accuracy of the model.

Edge Case Uncommon input or scenario that the system must handle safely.

FIPPA (Freedom of Information and Protection of Privacy Act) Ontario privacy law affecting the university data in the Authentication process.

Frontend User interface in the browser that handles input, feedback, and accessibility features.

Git/Github Version control and collaboration platform.

HTTP/HTTPS Web protocols in which HTTPS adds a transport layer security encryption for integrity and privacy.

Issue (Github) Tracked unit of task, bug, or feature with discussion and linkage to commits in Github.

JAWS (Job Access with Speech) A screen reader software available on Windows.

JSON (JavaScript Object Notation) / YAML (Yet Another Markup Language) Human-readable data formats used for configs and API payloads.

Latency Time from user action such as uploading an image to a system response or alt text generation

Low Vision Reduced level of vision that interferes with daily activities and is to be considered in designing the user interface and testing.

Manual Accessibility Testing Human review or testing of user interface and alt text.

Modularity Separating user interface, vision, language, and validation for maintainability.

NVDA (NonVisual Desktop Access) A free screen reader available on Windows.

OCR (Optical Character Recognition) Extracts embedded text in images or diagrams.

PII (Personally Identifiable Information) Data that identifies a person and must not appear in outputs or logs for security.

Screen Magnifier Assistive technology to enlarge screen content.

Screen Reader Assistive technology that reads text aloud.

Session History Record of user uploads and generated alt text during the current session.

Stakeholder Anyone affected by or influencing the system.

Technical Diagram An informational visual used in post-secondary course materials.

TLS (Transport Layer Security) Protocol providing encryption and integrity.

WCAG 2.1 (Web Content Accessibility Guidelines) International standard for accessible web content.

WCAG Levels (A/AA/AAA) Different conformance tiers to WCAG 2.1 where Level AA is the target for the project.

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 the last LATEST_RELEASES_NUM of 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 LATEST_RELEASES_NUM 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 MOST_COMMON_SR 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.

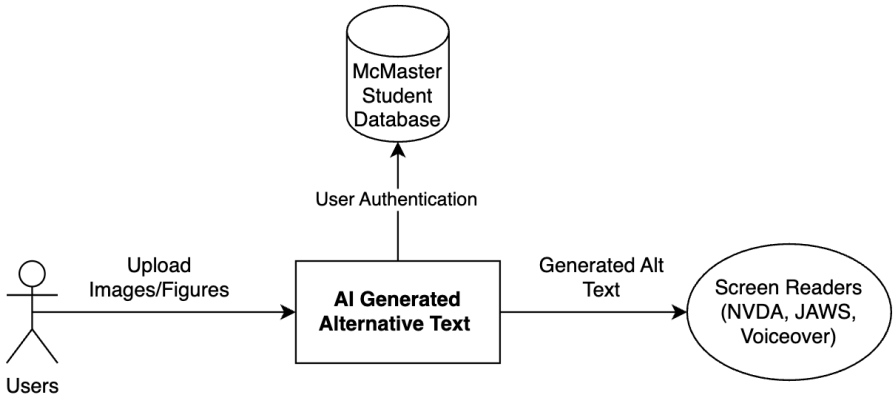

Figure 1: Work Context Diagram

6.3 Work Partitioning

Table 2 shows the work partitioning for completing the project. It includes major events, their inputs and outputs, and the summary of the event.

Table 2: 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
Optical Character Recognition 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	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

This section describes the structure and organization of the data flow throughout the system. It explains how the system's components interact with the stored data, ensuring a consistent and well-defined understanding of the information processed by the tool.

7.1 Business Data Model

The following diagram (Figure 2) illustrates the relationships between key components of the system as well as their interactions with external components.

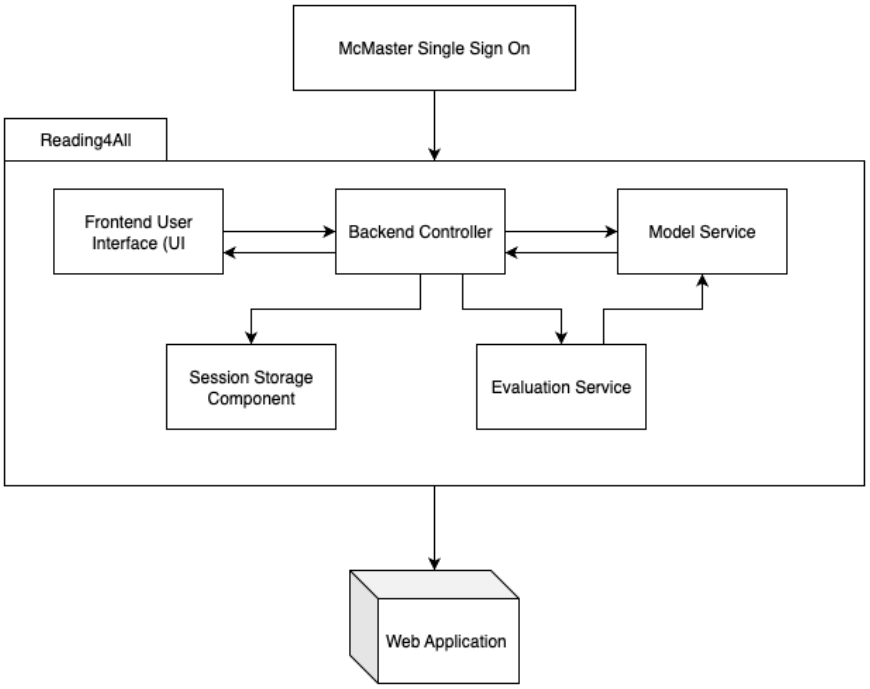

Figure 2: Business Data Model of alt text Generation System

7.2 Data Dictionary

Table 3: Data Dictionary for the Reading4All System

Name	Content	Type
Reading4All (container)	Logical package that contains the core components of the system (Frontend UI, Backend Controller, Model Service, Session Storage Component, Evaluation Service).	Package
Frontend User Interface (UI)	Uploads images, shows generated alt text, and captures tester evaluations. Provides keyboard/ARIA navigation and WCAG 2.1 Level AA compliant views.	Module
Backend Controller	Orchestrates requests: validates inputs (images/type/size), routes to the Model Service, manages temporary/session storage, records events, and returns results to the user interface (UI).	Module
Model Service	Machine-learning inference service that generates alt text from uploaded images and returns text plus timing/quality metadata. Exposed via an internal Application Programmable Interface (API).	External Service / Module
Session Storage Component	Short term temporary storage for uploaded image bytes and short-lived artifacts used during inference; configured to auto-delete shortly after processing.	Storage
Evaluation Service	Collects and persists pilot-testing ratings (sufficiency, length, accessibility/usability, learning impact) and optional notes for analysis.	Module / Dataset
McMaster Single Sign On (SSO)	Institutional authentication provider used to restrict access to McMaster users; issues and validates user sessions for the web application.	External Service Provider
Web Tool (deployment)	Deployed application entry point that exposes the Reading4All system to end users over web protocols such as Hypertext Transfer Protocol Secure (HTTPS); integrates Single Sign On (SSO) and serves the User/Application Interfaces.	Package / Deployment Target

8 The Scope of the Product

8.1 Product Boundary

Figure 3 below shows the components within the system and how they connect. The components that this project will aim on building include a user interface, backend component, alt text generation ML model, a session history manager. Furthermore, these components will utilize or communicate with a screen reader software, McMaster Authentication system and external AI/ML Frameworks.

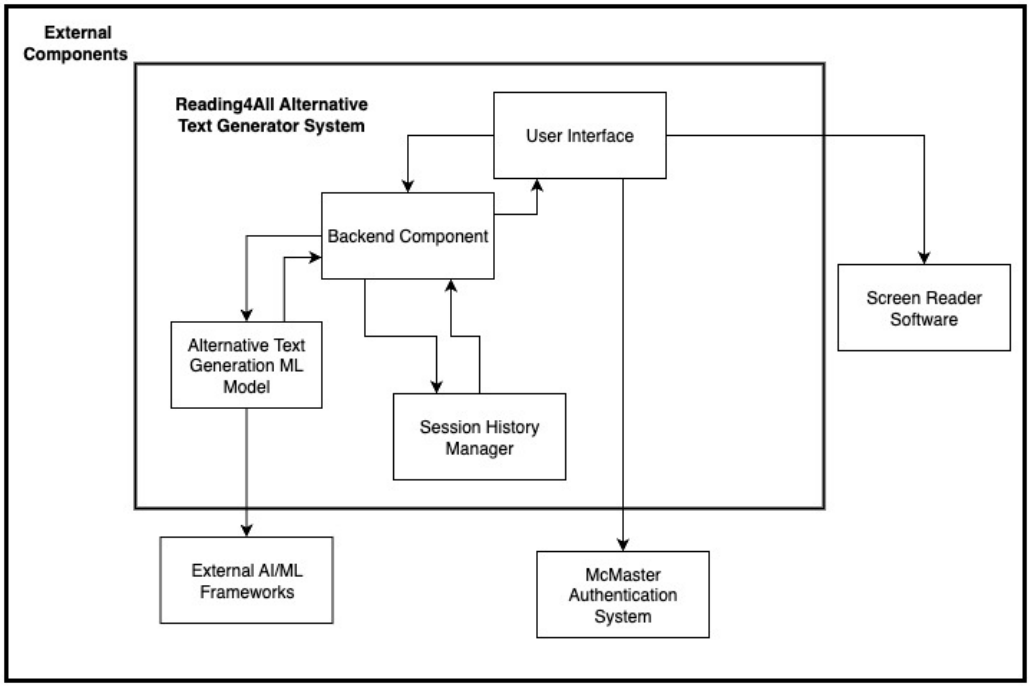

Figure 3: Product Boundary Diagram

8.2 Product Use Case Table

Table 4 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 4: Product Use Case Table

PUC#	PUC Name	Actor(s)	Input & Output(s)	Requirement
PUC 1	Login Using McMaster Credentials	McMaster Student and Faculty , McMaster Authentication System, UI, Backend Component	User Credentials (input), Authentication Results (output)	FR 6, SR-AR 1
PUC 2	Upload Image	McMaster Student and Faculty, UI, Backend Component	JPEG or PNG (input), Image upload status (output)	FR 1, UHR-EUR 3
PUC 3	Generate Alternative Text	McMaster Student and Faculty, Alt Text Generation ML Model, UI, Backend Component	Uploaded image (input), Generated alternative text	FR 2, FR 3
PUC 4	Copy or Download Text	McMaster Student and Faculty, UI, Backend Component	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, UI, Backend Component, Session History Manager	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 Authentication System, UI, Backend Component.
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, UI, Backend Component
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, Backend Component, Alt Text Generation ML Model.
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: Edit Generated Alternative Text

Trigger: User decides to modify the generated alternative text before copying or downloading it.
Preconditions:

- System has successfully generated alternative text for an uploaded image.

Actors: McMaster Student or Faculty, UI, Backend Component.
Outcome: The user edits and saves the updated alternative text.
Input: User edits to the generated alternative text.
Output: The edited alternative text is displayed to the user and stored within the session, to be copied or downloaded later.

PUC 5: Copy or Download Generated Alternative Text

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

- System has successfully generated alternative text for an uploaded image.
- User is satisfied with generated alternative text and has made any desired changes.

Actors: McMaster Student or Faculty, UI, Backend Component.
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 6: 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, Backend Component, Session History Manager.

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 MOST_COMMON_SR 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 MAX_ZOOM_PERCENTAGE, 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 MAX_ZOOM_PERCENTAGE 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 MIN_CONTRAST_RATIO.
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

LFR-SR 3. *The system interface must follow Don Norman's principles of designing an interface.*

Rationale: Applying Norman's principles, including visibility, feedback, constraints, mapping, consistency, and affordance, will make the system intuitive and user-friendly.
Fit Criterion: Evaluation against Norman's principles reveals no issues with any interface elements.
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 MAX_UPLOAD_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 MAX_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 the outcome of their actions and whether they are using the system correctly. This reduces the Gulf of Evaluation as it 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 USERS_SUCCESS_PERCENT of users can complete tasks without errors. When a user error occurs, the system explains the issue and how to recover within MAX_ERROR_RECOVER.
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 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 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 LEARNING_PERCENT of first time users with low vision using a screen reader can upload an image and generate alternative text within MAX_LEARNING_MINUTES without assistance.
Priority: High

11.4 Understandability and Politeness Requirements

UHR-UPR 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 T_ALT_GEN_SMALL for images $\leq$ IMG_SIZE_SMALL and within T_ALT_GEN_LARGE for images $\leq$ IMG_SIZE_BIG under normal load conditions.
Priority: High

PR-SL 2. *The web tool 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 T_UI_RESP 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 for the user session unless explicitly saved by the user. The images will not be stored in any database without user consent.
Priority: High

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

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 AA 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 AA success criteria.
Priority: High

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

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

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

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

alternative text that has been generated.
Priority: High
Hazard Analysis Connected: HA4.

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

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

12.3 Precision or Accuracy Requirements

Please refer to Appendix – Evaluation Metrics Summary (26) to understand the evaluation scales and metrics.

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 R_SUFFICIENCY 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: $\geq$ R_LENGTH 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 $\geq$ R_USABILITY_MEDIAN ("Mostly Accessible/Usable") on the 0–3 or 0–4 scales; no outputs below R_USABILITY_MIN.
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 T_ERROR_HANDLE 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 T_RECOVERY 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 CAP_CONCURRENT simultaneous requests with response times $\leq$ 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 CAP_STORAGE 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, software libraries, and ML frameworks.*

Rationale: Ensures long-term usability and compliance even after the pilot phase.
Fit Criterion: Minor updates or migrations require $\leq$ MAINT_TIME.
Priority: Medium

PR-LR 2. *The prototype shall maintain compatibility with at least the previous SFWR_RELEASES software 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 Operational and Environmental Requirements

The Reading4All system must operate reliably in typical university and institutional settings, both digitally and physically.

13.1 Expected Physical Environment

OER-EP1. *The system should be compatible with standard devices such as laptops, desktops, and cloud servers running Windows, macOS, Linux, or standard cloud platforms like AWS, GCP, or Azure.*

Rationale: This ensures that it will work properly in a variety of academic and institutional settings.
Fit Criterion: The system operates without compatibility issues across all supported platforms.
Priority: High

OER-EP2. *The system should function normally in common indoor conditions such as classrooms or offices with standard room temperatures and lighting.*

Rationale: Since no specialized hardware setup is required, it should perform consistently in regular academic and office environments.
Fit Criterion: The system performs reliably under standard indoor temperature (10°C–35°C) and lighting conditions.
Priority: Low

13.2 Wider Environment Requirements

OER-WE1. *The system should adhere to all applicable accessibility and privacy regulations, including AODA, WCAG 2.1 (Level AA), and institutional privacy policies like FIPPA.*

Rationale: This ensures that the system complies with legal and ethical standards regarding inclusion and data protection.
Fit Criterion: Independent evaluation confirms compliance with AODA and WCAG 2.1 Level AA.
Priority: High

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

Rationale: Reading4All depends on models for text generation, which may require reliable network access.
Fit Criterion: The system maintains consistent API commu-

nication over typical university Wi-Fi or Ethernet with upload speeds of at least IMG_SIZE_BIG Mbps.
Priority: Medium

13.3 Requirements for Interfacing with Adjacent Systems

OER-IAS1. *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-IAS2. *The system should support common image formats such as JPG, JPEG, and PNG to be compatible with the majority of course materials.*

Rationale: Supporting standard image formats ensures instructors can use materials from a variety of academic sources.
Fit Criterion: The system processes each supported image format correctly and generates accurate alternative text.
Priority: Medium

OER-IAS3. *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 tool for institutional integration.*

Rationale: This ensures accessibility for individual users and organizations integrating accessibility workflows.
Fit Criterion: A hosted web tool is accessible and validated through institutional testing.
Priority: High

13.5 Release Requirements

OER-RL1. *Core features such as image analysis, text generation, and accessibility validation should be implemented, tested, and verified prior to any release.*

Rationale: Ensuring these core components are functional before release guarantees system reliability and completeness.
Fit Criterion: Verification and validation documentation confirms that all functional requirements have been met.
Priority: High

OER-RL2. *The system must be ready for release by the end of March 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

Reading4All must be easy to maintain, update, and support after deployment, with structures that allow for future innovations.

14.1 Maintenance Requirements

MS-MNT1. *The system will be divided into modular components including the front-end interface, image analysis, language generation, and accessibility verification to ensure that each one can be updated or replaced without affecting the others.*

Rationale: Modular design simplifies maintenance by isolating potential issues and enabling component-specific updates without impacting the entire system.
Fit Criterion: Each subsystem (e.g., the image analysis or text generation module) can be modified or redeployed independently without causing failures in other modules.
Priority: High

MS-MNT2. *Every source file must include concise comments, and higher-level documentation such as installation guides and API references should be maintained in GitHub Wiki or README files.*

Rationale: Well-documented source code and external references make it easier for new developers to understand and maintain the project.
Fit Criterion: All functions and classes include docstrings, and complete setup instructions are verified through internal onboarding tests.
Priority: Medium

MS-MNT3. *Automated CI/CD pipelines should run unit, integration, and accessibility tests after each merge to ensure that updates do not disrupt existing functionality.*

Rationale: Continuous testing prevents regressions and ensures that all code changes maintain system stability and accessibility compliance.
Fit Criterion: Each code merge triggers automated tests confirming successful execution of core functions and compliance checks.
Priority: High

14.2 Supportability Requirements

MS-SUP1. *The system must record key metrics such as API calls, latency, error rates, and model confidence scores to support debugging and performance improvements.*

Rationale: Tracking performance data allows maintainers to identify bottlenecks, reduce errors, and optimize processing effi-

ciency.
Fit Criterion: Logged metrics are stored securely and reviewed periodically through a monitoring dashboard or exported reports.
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 data structures 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 files.
Priority: Medium

MS-AD3. *Every academic term, significant updates should be tracked and documented to maintain change records and ensure reproducibility.*

Rationale: Periodic documentation ensures version transparency and supports academic continuity for future development teams.
Fit Criterion: Version history includes changelogs, revision notes, and verification summaries for each update cycle.
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 TLS_VERSION 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 after each session.*

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 FILE_DELETE_TIME 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 FILE_TYPES 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 defined in NETWORK_SOURCE_POLICY.
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.
- Some users have pre-defined keyboard shortcuts, while others use a standard menu. Therefore, more research needs to be conducted on the most optimal and efficient keyboard navigation type with the least accessibility barriers.
- As mentioned in Table 5 of the Appendix, all these metrics need to be researched to further understand them.

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

Inspirations can be taken from existing software components and libraries including:

- **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 New Problems

20.1 Effects on the Current Environment

The current academic and developmental environment may undergo a number of changes as a result of the implementation of the Reading4All system. Among these effects are:

1. During analysis and description creation, the tool may use more memory and processing power, which could momentarily impair the functionality of other concurrently running applications.

2. To incorporate Reading4All into current authoring or content preparation workflows, extra setup or dependencies might be needed.

20.2 Effects on the Installed Systems

1. It might be necessary to assess whether current departmental or institutional systems are compatible with Reading4All. The necessary integrations might not be fully supported by older software environments.

2. Testing is necessary to make sure that routine document or media uploads continue to work as intended because the integration process may cause changes in system behaviour.

3. In order to handle new descriptive text outputs and associated accessibility metadata, more storage or indexing might be required.

20.3 Reusable Components

Inspirations can be taken from existing software components and libraries including:

- **Hugging Face Transformers:** An open-source library offering pre-trained 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.

20.4 Follow-Up Problems

1. Reading4All will require recurring updates as accessibility guidelines change in order to stay compliant and guarantee its ongoing efficacy.
2. Continuous user feedback might point out problems or areas that need work that weren't noticed during the first deployment.
3. To guarantee consistent performance and integration with institutional systems, long-term upkeep and assistance will be needed.

21 Tasks

21.1 Project Planning

- **Development Approach** The Reading4All team will adopt an iterative and adaptive workflow that promotes continuous improvement and consistent stakeholder engagement. Development will proceed through a series of short sprints, each focused on specific deliverables while allowing flexibility as project requirements evolve. The general development process will include:

1. Analyzing requirements and refining functional specifications
2. Prioritizing the backlog and conducting sprint planning
3. Incremental coding and component integration
4. Performing unit testing, validation, and verification
5. Holding review sessions with the accessibility lead and project supervisor
6. Completing final deployment and documentation preparation

- **Key Tasks** The main project activities will involve:
  - 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 TEAM_SIZE member team responsible for design, development, testing, and reporting. Shared tools will include:
  - Cloud-based GPU environments for AI model inference and experimentation
  - GitHub Projects for version control, issue tracking, and sprint management
  - Academic diagram datasets paired with validated alt-text examples for training and evaluation
- **Key Considerations** The following factors are critical to ensuring efficient development and accessibility compliance:
  - Detecting and resolving dependency conflicts through early integration testing

  - Monitoring and mitigating dataset bias to ensure inclusive model performance
  - Maintaining consistent communication with the accessibility consultant and faculty supervisor
  - Reducing project risks through incremental reviews, frequent documentation updates, and checkpoint testing

- **Documentation Process** All project documentation will be collaboratively created and maintained by team members. The documentation process will follow these practices:
  - Storing all documents in the shared GitHub Wiki to promote transparency and accessibility
  - Using clear and concise commit messages and structured pull requests
  - Reviewing automated test results before code integration
  - Applying version tagging and peer review for all major deliverables
  - Archiving finalized reports and test outcomes for submission and supervisor review

21.2 Planning of the Development Phases

Deliverables and Tentative Schedule

Deliverables	Due Date
Problem Statement, Proof of Concept, and Development Plan	Week 04
Software Requirements Specifications and Hazards Analysis (Revision 0)	Week 06
Verification & Validation Plan (Revision 0)	Week 08
Design Document (Rev-1)	Week 10
Proof of Concept Demonstration	Week 11 + 12
Design Document (Revision 0)	Week 16
Project Demonstration (Revision 0)	Week 18 + 19
Verification & Validation Report (Revision 0)	Week 22
Final Demonstration (Revision 1)	Week 24
Final Documentation	Week 26
Capstone EXPO	Week 26

Sprint and Review Cycle

- Each sprint will last approximately two weeks and conclude with a structured progress review.
- Milestones will be verified through peer review, supervisor evaluation, and automated testing.
- Task reprioritization and milestone adjustments will be based on feedback from the accessibility advisor and project supervisor.

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.

22 Migration to the New Product

22.1 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

22.2 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.

23 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 and university-hosted infrastructure, no direct monetary expenditure is incurred. The project is scheduled to be completed within the academic term (Minimum Viable Product (MVP) ready by April 2026), and resource allocation is focused on efficient time management and workload balancing rather than financial cost.

23.1 Metrics for Estimation

C-ME 1. *Number of image input/output workflows supported by the tool.*

Rationale: Defines the scope of functionality that affects the development and testing workload.
Priority: High

C-ME 2. *Number of core functional requirements (e.g., image upload, alt text generation, user authentication, evaluation metrics).*

Rationale: Determines the complexity of the implementation and testing processes.
Priority: High

C-ME 3. *Number of non-functional requirements (e.g., accessibility compliance, latency, privacy, and scalability).*

Rationale: Reflects the additional design and validation effort beyond core functionality.
Priority: Medium

C-ME 4. *Number of deliverables and milestones within the development timeline.*

Rationale: Ensures that project progress and deliverables are measurable and time-bound.
Priority: Medium

C-ME 5. *Team size and individual role distribution (frontend, backend, model integration, documentation).*

Rationale: Defines the workload balance and collaborative structure of the development process.
Priority: High

23.2 Estimation Approach

C-EA 1. *Each deliverable has been estimated based on the effort required to implement, test, and document it within the academic term.*

Rationale: Provides a structured approach to time allocation aligned with academic deadlines.
Fit Criterion: Estimates are derived from prior experience with similar web-based machine learning projects and adjusted for accessibility integration.
Priority: High

C-EA 2. *Time allocation accounts for model fine-tuning, accessibility testing, and usability evaluation.*

Rationale: Incorporates all major tasks necessary to ensure compliance with WCAG standards and project objectives.
Fit Criterion: Schedule includes model optimization, frontend validation, and pilot user testing within the project timeline.
Priority: Medium

23.3 Cost Breakdown

C-CB 1. *Development Effort*

Rationale: Represents the primary cost driver, measured in person-hours contributed by the student team and supervisor.
Fit Criterion: Based on a team of TEAM_SIZE student developers and one faculty supervisor:

- Initial research, planning, and requirement analysis: HOURS_RESEARCH per team member.
- Model integration and backend implementation: HOURS_BACKEND per team member.
- Frontend development and accessibility compliance: HOURS_FRONTEND per team member.

- Testing, debugging, and refinement: HOURS_TESTING per team member.
- Documentation and presentation preparation: HOURS_DOCS per team member.

Total Estimated Effort: HOURS_TOTAL per team member.
Priority: High

C-CB 2. *Tools and Software*

Rationale: All software components used in the project are open-source or free for academic use.
Fit Criterion: No direct licensing or software procurement costs are incurred.
Priority: High

C-CB 3. *Testing Environment*

Rationale: Defines the testing setup and potential costs associated with user testing incentives.
Fit Criterion: Testing is conducted on McMaster-hosted or open-source platforms. User testing sessions may include incentives between COST_INCENTIVE_MIN–COST_INCENTIVE_MAX.
Priority: Medium

C-CB 4. *Hardware and Infrastructure*

Rationale: Evaluates the physical and computational resources used during development.
Fit Criterion: No additional hardware purchases required beyond student laptops and academic cloud credits.
Priority: Low

23.4 Estimated Cost

C-EC 1. *Total Development Effort*

Rationale: Quantifies total workload for the entire team in measurable units.
Fit Criterion: Approximately HOURS_PROJECT person-hours across team members (TEAM_SIZE $\times$ HOURS_TOTAL).
Priority: High

C-EC 2. *Notional Monetary Cost*

Rationale: Provides an academic-equivalent cost estimate for effort valuation.
Fit Criterion: Assuming an average rate of COST_PER_HOUR, total notional cost is approximately COST_TOTAL.
Priority: Medium

C-EC 3. *Actual Financial Cost*

Rationale: Since the project leverages university and open-source resources, there are no direct expenses.
Fit Criterion: The actual monetary cost is COST_ACTUAL; all expenditure is in research and development effort.
Priority: High

24 User Documentation and Training

24.1 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.

24.2 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.

25 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.
- Allowing users to report issues or feedback anonymously directly through the web tool that can enable user-driven improvements while maintaining privacy.

26 Ideas for Solution

This section discusses potential ways to achieve 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.

1. **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.
2. **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 technical diagrams, paired with examples of descriptive alternative text, allow it to generate accurate and high quality descriptions for new images.

3. **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 — Evaluation Metrics Summary

The following table summarizes the evaluation metrics that will be used to assess the quality and effectiveness of the alternative text generated by the Reading4All system. Each metric includes its scale type, acceptable range, and a brief description of its purpose.

Table 5: Evaluation Metrics Summary

Metric Name	**Scale Type**	**Acceptable Range**	**Summary Description**
Sufficiency of Description	Categorical (1–3)	≥ 3 (Sufficient)	Does the alt text convey enough information to achieve the intended objective?
Length Appropriateness	Categorical (1–3)	≥ 3 (Proper Length)	Is the alt text concise yet complete (not too short or overly verbose)?
Accessibility / Usability	Numerical (0–3)	≥ 2 (Acceptable)	Assistive-technology compatibility and clarity; aligns with WCAG 2.1 Level AA use.
Learning Impact	Numerical (0–3)	≥ 2 (Positive)	Does the alt text support or enhance user understanding in learning contexts?
Qualitative Feedback Notes	Textual	N/A	Free-form comments on clarity, tone, and suggested improvements.

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:

Group Reflection - Reflection

1. **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.**

   Through completing our SRS document, we discovered that we need to learn more about domain specific knowledge and how it translates to software applications. Furthermore, we need to research the optimal length of the generated alternative text and different types of keyboard navigation. We also need to expand our software engineering knowledge to determine the best architecture for our system, while ensuring the given alternative text evaluation metrics are met. Finally, we need to learn more about scripting usability test and ensuring it covers all diverse accessibility requirements.

2. **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?**

Due to the nature of this project, our team decided to share all our findings about the topics listed above, rather than splitting across the team. We believe this will ensure team member has a thorough understanding of every aspect of the project and can contribute in the future.

- **Optimal Length of Generated Alternative Text**: The first approach will be to read the material and research provided by our supervisor, Jing. We will also complete our own research individually and discuss it as a group, ensuring we ask any questions to our supervisor.
- **Types of Keyboard Navigation**: This will involve researching current methods for keyboard navigation and trying it out ourselves. Additionally, we can consult with our supervisor on more resources and peers who may regularly use keyboard navigation.
- **Best Architecture for Our system:** We will first explore the different software architectures and combine them to see of any possible hybrids. While learning about the different architectures, we will try implementing small scale versions. We will also consult Dr Khedri for any additional information and his insights.
- **Scripting Usability Testing:** We will do research on completing user testing with people of diverse accessibility needs. We will take notes while doing so and ensure we do practice testing with our supervisor. This will allow us to learn at low stakes and get feedback.

Fiza Reflection - Reflection

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

Throughout the process, our team maintained organization and had good communication. It was simpler to preserve flow and prevent redundancy because we had a clear framework and consistent formatting from earlier deliverables. We were also able to maintain our efficiency and keep the material consistent with our previous work. In order review sections, define expectations, and make sure the deliverable satisfied all requirements, we also had meetings with our TA and our team.

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

   One difficulty was avoiding redundancy by keeping formatting constant and clearly distinguishing design components from implementation specifics. In order to guarantee accuracy and uniformity throughout all sections, we addressed this by going over the course templates and conducting team editing sessions to make sure the formatting followed a consistent style. We also helped each other out by sharing tips and tricks to improve clarity and presentation.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?**

   Key design and functionality components that were influenced by internal meetings, conversations with our supervisor, pertinent compliance guidelines, and my internship's prior experience working with AI included the accessible interface, the quality of the alt text that was generated, and adherence to AODA and WCAG 2.1 standards.

4. **Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.**

   The Software Requirements course from third year was extremely useful in creating this deliverable. It created a solid foundation for organizing requirement documents, developing precise specifications, and grasping the foundations of software documentation and traceability, all of which significantly aided our work on this project.

Moly Mikhail - Reflection

1. **What went well while writing this deliverable?**
   I believe writing this deliverable many things went well. I really enjoyed getting to think about the different non-functional requirements. I found that have different sections of non-functional requirements encouraged me to think about different aspects of the system and things we will have to keep in mind during development. For example, prior to writing this deliverable, we hadn't considered personalization and

internationalization requirements; however, having to complete that section led us to add important functionality of allowing the user to decide which way to store the alternative text.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   One pain point I experience writing this deliverable dealt with completing the product boundary. Initially, I was confused on The Scope of the Product section and what was expected. To resolve this, I researched the Volere Requirements Specification Template and looked into the section. However, I was still confused and what was expected of the section. Finally, during our meeting with our TA I was able to clarify the expectations for this section and I was able to complete the section.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?**
   I believe many of our non-functional requirements, specifically look and feel requirements, as well as usability and humanity requirements. Through our conversations with our supervisor Jing, we learned a lot of the accessibility requirements for website applications. For example, one specific requirement that was derived from our conversations was that the system cannot use color alone to convey any messages or information. I believe without having this conversation, this is a requirement that would not have been discovered.

4. **Which of the courses you have taken, or are currently taking, will help your team be successful with your capstone project?**
   I believe many courses that I have taken, and some that I'm currently taking will contribute to the success of our capstone project. I completed SFWRENG 4HC3 - Human Computer Interfaces, which has taught me many important design principles, such as Normans Design Principles. Furthermore, completing COMPSCI 4AL3 - Applications of Machine Learning, also will be a lot of help when completing our capstone. This course introduced me to developing machine learning models and will be directly applicable. Finally, taking COMPSCI 3RA3 - Software Requirements and Security Considerations will also help our team be successful.

Casey Francine Bulaclac - Reflection

1. **What went well while writing this deliverable?**
   Having discussed the project thoroughly as a team and with our supervisor helped in writing this deliverable as the team was very knowledgeable about the needs for the project. This deliverable went much smoother than the last due to stronger operational procedures, and better organization in how we structured and completed the SRS. The team communicated well and were clear of the goals for this deliverable.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   One pain point in writing the SRS was figuring out what each of the many sections entailed in the Volere's template. The template is very thorough and needed many details, in which some sections seem to overlap which can be confusing. Another pain point was ensuring traceability between our goals in the project and the requirements. To resolve this, I made sure to ask the TA for feedback and clarification about specific sections. Additionally, communicating with each team member and ensuring our requirements aligned to the goals of the project was very helpful in aiding to ensure traceability.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?**
   Many, if not most, requirements were inspired through speaking with our supervisor, who had the most knowledge and experience with our project's potential users and stakeholders. In this project, it is important to understand our target users as we are designing for accessibility, so it was critical in making our requirements.

4. **Which of the courses you have taken, or are currently taking, will help your team be successful with your capstone project?**
   In this deliverable, the course that was most beneficial was Software Requirements and Security Considerations (SFWRENG 3RA3) as we learned how to create effected SRS documents. A course I've taken that will help thoroughly in ensuring our user interface is accessible is Human Computer Interfaces (SFWRENG 4HC3) as the course taught

us principles of good design. Lastly, another course I took that contribute to the success of our project is Applications of Machine Learning (SFWRENG 4AL3) as this project heavily involves machine learning in generating alternative text.

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.

Dhruv Sardana - Reflection

1. **What went well while writing this deliverable?**
   During the initial phases of our document planning, it helped to segregate the parts equally and fairly which allowed everyone's plate to be full but equally. At the same time, it helped me think critically about the requirements of the system especially WCAG Guidelines that I would have not thought of before. This allows me to have a better system understanding while also understanding the performance, safety, security and users of the system.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   One of the pain points I experienced was the lack of understanding of certain technical terms and concepts that were crucial to the project. I had to research and learn about the concepts of screen readers and accurancy requirements while also understand WCAG level AA guidelines.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?**
   During our meeting with our supervisor, we had intensive discussion about our stakeholders, system requirements, security requirements,

WCAG and the user's needs. This helped us clarify our requirements and make sure that we are on the same page as our supervisor. I would say alot of requirements were inspired by our supervisor and her experience in the field.

4. **Which of the courses you have taken, or are currently taking, will help your team be successful with your capstone project?** I would say Human Computer Interfaces and Databases have been the most helpful courses for this project. While also understanding concepts of writing in 3RA3 and understanding system design and low latency systems.

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