

Software Requirements Specification for VoiceBridge: An Accessible Speech-to-Control System

Team 13, Speech Buddies

Mazen Youssef

Rawan Mahdi

Luna Aljammal

Kelvin Yu

December 25, 2025

Contents

1	Purpose of the Project	6
1.1	User Business	6
1.2	Goals of the Project	6
2	Stakeholders	7
2.1	Client	7
2.2	Customer & Hands-On Users of the Project	7
2.3	Other Stakeholders	7
2.4	Personas	7
2.5	Priorities Assigned to Users	7
2.6	User Participation	8
2.7	Maintenance Users and Service Technicians	8
2.8	Society	8
3	Mandated Constraints	9
3.1	Solution Constraints	9
3.2	Implementation Environment of the Current System	9
3.3	Partner or Collaborative Applications	9
3.4	Off-the-Shelf Software	10
3.5	Anticipated Workplace Environment	10
3.6	Schedule Constraints	10
3.7	Budget Constraints	10
3.8	Enterprise Constraints	11
4	Naming Conventions and Terminology	11
5	Relevant Facts And Assumptions	14
5.1	Relevant Facts	15
5.2	Business Rules	16
5.3	Assumptions	17
6	The Scope of the Work	17
6.1	The Current Situation	17
6.2	The Context of the Work	18
6.3	Work Partitioning	18
6.4	Specifying a Business Use Case (BUC)	19
7	Business Data Model and Data Dictionary	21
7.1	Business Data Model	21
7.2	Data Dictionary	23
8	The Scope of the Product	24
8.1	Product Boundary	24
8.2	Product Use Case Table	25
8.2.1	Individual Product Use Cases (PUC's)	25
9	Undesired Event Handling	28

10 Functional Requirements	29
10.1 FR1: Accept Speech Audio via Microphone	29
10.2 FR1: Accept Speech Audio via Microphone	29
10.3 FR2: Convert Impaired Speech to Text with greater than or equal to 80% Accuracy (MVP)	29
10.4 FR3: Display Transcription for Verification	29
10.5 FR4: Map Text to Arbitrary Device Commands	29
10.6 FR5: Execute Commands on the Host Device	30
10.7 FR6: Handle Security Software Interference	30
11 Look and Feel Requirements	30
11.1 Appearance Requirements	30
11.2 Style Requirements	30
12 Usability and Humanity Requirements	30
12.1 Ease of Use Requirements	30
12.2 Personalization and Internationalization Requirements	30
12.3 Learning Requirements	31
12.4 Understandability and Politeness Requirements	31
12.5 Accessibility Requirements	31
13 Performance Requirements	31
13.1 Speed and Latency Requirements	31
13.2 Safety-Critical Requirements	31
13.3 Precision or Accuracy Requirements	31
13.4 Robustness or Fault-Tolerance Requirements	31
13.5 Capacity Requirements	31
13.6 Scalability or Extensibility Requirements	31
13.7 Longevity Requirements	32
14 Operational and Environmental Requirements	32
14.1 Expected Physical Environment	32
14.2 Wider Environment Requirements	32
14.3 Requirements for Interfacing with Adjacent Systems	32
14.4 Productization Requirements	32
14.5 Release Requirements	32
15 Maintainability and Support Requirements	33
15.1 Maintenance Requirements	33
15.2 Supportability Requirements	33
15.3 Adaptability Requirements	33
16 Security Requirements	33
16.1 Access Requirements	33
16.2 Integrity Requirements	33
16.3 Privacy Requirements	33
16.4 Audit Requirements	33
16.5 Immunity Requirements	34
17 Cultural Requirements	34
17.1 Cultural Requirements	34

18 Compliance Requirements	34
18.1 Legal Requirements	34
18.2 Standards Compliance Requirements	34
19 Requirements Likely and Unlikely to Change	34
19.1 Likely to Change	34
19.2 Unlikely to Change	35
20 Open Issues	35
21 Off-the-Shelf Solutions	35
21.1 Ready-Made Products	35
21.2 Reusable Components	35
21.3 Products That Can Be Copied	35
22 New Problems	36
22.1 Effects on the Current Environment	36
22.2 Effects on the Installed Systems	36
22.3 Potential User Problems	36
22.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	36
22.5 Follow-Up Problems	36
23 Tasks	36
23.1 Project Planning	36
23.2 Planning of the Development Phases	36
24 Migration to the New Product	36
24.1 Requirements for Migration to the New Product	36
24.2 Data That Has to be Modified or Translated for the New System	37
25 Costs	37
26 User Documentation and Training	37
26.1 User Documentation Requirements	37
26.2 Training Requirements	37
27 Waiting Room	37
28 Ideas for Solution	37
28.1 Browser-Based Extensions	37
28.2 Personalized ASR Fine-Tuning	37
28.3 LLM Command Mapping	38
28.4 Noise Filtering	38
28.5 Open-Source Integrations	38
28.6 Accessibility Enhancements	38
28.7 Data Encryption	38
29 Requirements Phase-In Plan	38
29.1 Phase Definitions	38
29.2 Requirements Phase-In	38
29.3 Requirement Implementation Timeline	41

30 Formal Specification	41
30.1 System States	41
30.2 Inputs and Transitions	42
30.3 Output Behavior	42
30.4 Mapping to Requirements	42

Revision History

Date		Version	Notes
October 2, 2025		Rawan and Luna	Filled-In Initial Draft of SRS Document
October 10, 2025		Rawan and Luna	Completed SRS Document
October 10, 2025		Mazen and Kelvin	Reviewed and Edited SRS Document
November 3, 2025		Luna	Addressed SRS Glossary TA and Peer Feedback, added customer to glossary for clarity, minor glossary organization improvements
December 19, 2025		Rawan	Addressing peer feedback on caregivers as stakeholders, and addition of C5 and FR6
December 19, 2025		Rawan	Addressing TA feedback: adding section descriptions, figure/table descriptions, and clarification to varying TA questions, removing duplicate constraint (C1), UML description, and other minor fixes

Table 1: Revision History for SRS Document

1 Purpose of the Project

This section describes the overall goals and intended outcomes of the project. It describes the opportunities for developing the project as well as the value it provides for individual users and overall community.

1.1 User Business

Individuals with speech impairments face significant barriers when interacting with digital devices. VoiceBridge addresses this gap by providing an accurate, inclusive, and accessible speech-to-control system that enables users to communicate with their devices using their speech, regardless of clarity of articulation. A detailed description of the VoiceBridge problem statement may be references in the [Problem Statement & Goals document](#).

Building on familiar technology, such as personal computers and mobile devices, may be one of the most cost-effective and easily adoptable approaches for improving an individual with disability’s autonomy and access to the world around them. The rise of Automatic Speech Recognition (ASR) technology and Artificial Intelligence (AI) integrations in the industry provides a novel landscape of opportunities to improve accessibility interfaces. VoiceBridge exploits bleeding-edge technology for a practical and impactful application.

1.2 Goals of the Project

ID	Goal	Description
G1	Accurate Speech Transcription	Reliably convert impaired or slurred speech into text.
G2	Command Mapping	Translate recognized speech into actionable browser commands.
G3	User Independence	Enable users with speech impairments to browse autonomously.
G4	Lightweight & Accessible Design	Keep the system simple, fast, and cost-effective.
G5	Cross-Browser Compatibility	Support major browsers (e.g., Chrome, Edge, Firefox).
G6	Robust Error Handling	Detect and recover gracefully from common failures.
G7	Data Privacy & Security	Protect user data and ensure secure local processing.
G8	Customizable Interface	Allow users to adjust sensitivity, shortcuts, and feedback modes.
G9	Scalable Architecture	Design the system for future integration beyond browsers.

Table 2: Project Goals for VoiceBridge: Identifies and describes the overarching goals of VoiceBridge

2 Stakeholders

This section describes VoiceBridge Stakeholders, assigns priorities to them, and defines their respective goals.

2.1 Client

The primary client for the VoiceBridge project is the organization or individual funding or commissioning the system. The client is primarily concerned with achieving the following goals: **G1**, **G5**, and **G9**, ensuring accurate speech transcription, cross-browser compatibility, and scalable architecture.

2.2 Customer & Hands-On Users of the Project

The primary customers are also the users of the project, them being individuals with speech impairments, who seek independence and autonomy through technology. Their needs directly relate to: **G1**, **G2**, **G3**, and **G8**.

2.3 Other Stakeholders

Secondary stakeholders include experts in linguistics, speech processing, and healthcare domains:

- Relatives and caregivers of the hands on user of the VoiceBridge Platform.
- Speech researchers and linguistics specialists, including the project supervisor, Dr. Christian Brodbeck, who provide insight toward **G1**, **G6**, and **G9**.
- Healthcare professionals and speech therapists who advise on usability and accessibility, contributing to **G3** and **G8**.
- Accessibility advocates and organizations interested in promoting the application, aligned with **G4** and **G5**.
- Software developers who implement and maintain the system, supporting **G9** and **G6**.

Tertiary stakeholders include caregivers and professionals who interact with end users, supporting **G3** and **G7**.

2.4 Personas

Potential end users of VoiceBridge include:

- **Amira**, a 45-year-old with Parkinson's disease, uses the system to log into Gmail and send emails. Her voice is described by speech pathologists as being monopitch and breathy, often making it difficult for her to be understood by the people and technology she interacts with
- **David**, a stroke survivor, uses the system to browse the web and make purchases. His voice is described by pathologists as having slurred and distorted consonants, as well as being monopitch, distorting his speech.

2.5 Priorities Assigned to Users

The highest priorities are assigned to end users with speech impairments, as their experience with the system defines its success. Secondary priorities include caregivers and technical experts who support the end users in using and maintaining the system.

2.6 User Participation

Individuals matching the target user profiles will be recruited for prototype testing and personalization development. Their participation directly supports refining goals **G1**, **G2**, **G3**, and **G8**.

2.7 Maintenance Users and Service Technicians

Maintenance Users:

- **Role:** End-users or caregivers performing basic troubleshooting and initiating support requests.
- **Responsibilities:**
 - Reporting errors or unexpected system behavior (**G6**).
 - Installing application updates (**G9**).
 - Managing user-specific configurations (**G8**).

Service Technicians:

- **Role:** Trained technical staff with deeper access to system logs and back-end services.
- **Responsibilities:**
 - Investigating reported issues (**G6**).
 - Ensuring transcription accuracy (**G1**).
 - Deploying updates and patches (**G9**).
 - Ensuring compatibility with operating systems and accessibility frameworks (**G5**).
 - Performing preventive maintenance, including performance monitoring and optimization (**G1**, **G6**).

2.8 Society

Society is recognized as an essential stakeholder in the VoiceBridge project, given the system’s potential impact on accessibility, inclusion, and digital independence for individuals with speech impairments. The project directly supports broader social goals of equity and participation by enabling users to interact with technology without barriers, contributing to fair access to digital communication and online services.

Health and Safety: VoiceBridge promotes safe technology use by minimizing physical strain associated with traditional input methods, such as typing or clicking, which can be challenging for users with motor limitations. Data privacy and security protocols safeguard user health information and prevent psychological stress caused by data misuse or breaches, aligning with ethical design standards for assistive technology.

Cultural and Ethical Considerations: The system is designed to accommodate diverse speech patterns, dialects, and accents, ensuring that cultural and linguistic differences are respected rather than penalized by the model. VoiceBridge avoids stigmatizing language or error messaging, instead emphasizing supportive feedback to maintain user confidence and dignity.

Social Impact: By reducing communication barriers, VoiceBridge fosters greater inclusion in workplaces, education, and online communities. The system thus contributes to both individual empowerment and collective social benefit.

3 Mandated Constraints

This section outlines the external and internal constraints that the system must comply with. It includes environmental constraints, budgetary limits, and implementation restrictions that may impact design decisions, system behavior, or deployment.

3.1 Solution Constraints

C1 — The product shall accept non-deterministic user input in the form of natural language speech.

Rationale: Individuals with speech impairments may produce varied speech patterns that cannot be handled by rigid or deterministic command structures.

Fit Criterion: The system must be capable of processing and responding to variable natural language inputs without requiring a fixed set of commands.

C2 — The product shall integrate with a browser control application to execute voice-based commands.

Rationale: Browser interaction is a primary accessibility point for most digital services, and browser control is essential for practical use of the system.

Fit Criterion: The system must successfully perform browser actions (e.g., opening tabs, navigating to URLs, scrolling) through the integrated control application.

3.2 Implementation Environment of the Current System

C3 — The product will operate within a consumer computing environment consisting of personal computers equipped with microphones.

Rationale: This setup reflects the most common user hardware configuration, ensuring the solution is accessible without additional devices.

Fit Criterion: All core functionalities must operate correctly on standard personal computers with a functioning audio input device.

C4 — The application must operate within standard browser security models and cannot bypass AV or OS-level protections.

Rationale: The user's antivirus/endpoint protection software may block automated browser actions (e.g., script injection, simulated clicks, extensions).

Fit Criterion: Full functionality depends on the browser and security software allowing the extension/tool to run.

3.3 Partner or Collaborative Applications

C5 — The initial integration target is Browser Use, an open-source browser control and automation application.

Rationale: Leveraging existing open-source browser automation tools accelerates development and reduces implementation complexity.

Fit Criterion: The system must demonstrate the ability to execute at least three browser actions through the integrated partner application during testing.

C6 — Future integrations may include API servers, mobile device agents, and home assistants to extend accessibility and functionality.

Rationale: Ensuring extensibility allows the system to grow and adapt to new platforms or user needs.

Fit Criterion: The system architecture must allow seamless integration with additional partner applications without major redesign.

3.4 Off-the-Shelf Software

C7 — The product will rely on off-the-shelf software components, specifically Browser Use for automation and Project Euphonia for voice recording.

Rationale: Using established open-source components allows focus on core functionality, improves maintainability, and supports experimentation with real-world data.

Fit Criterion: These components must be successfully integrated into the development workflow, and collected recordings must be stored and processed for model training without licensing conflicts.

3.5 Anticipated Workplace Environment

C8 — The product may be used at home, in public spaces, or in clinical settings, each with varying background noise conditions.

Rationale: The product must function effectively across diverse real-world environments to meet accessibility goals.

Fit Criterion: The speech recognition system must maintain acceptable accuracy and responsiveness across all three identified environments during testing.

3.6 Schedule Constraints

C9 — Proof of concept must be completed by November 17, 2025. MVP must be ready within eight months, by May 2026.

Rationale: Deadlines align with capstone project milestones and funding timelines, ensuring timely testing and delivery.

Fit Criterion: All proof of concept requirements must be met by the November deadline, and MVP functionality must be fully operational by May 2026.

3.7 Budget Constraints

C10 — The project must operate within the capstone budget allocated for compute infrastructure and development resources.

Rationale: Budget limitations require prioritizing open-source solutions and efficient resource allocation.

Fit Criterion: Total cost of infrastructure, hosting, and third-party services must not exceed the allocated capstone budget of \$125 CAD, provided by the supervising institution, McMaster University.

3.8 Enterprise Constraints

C11 — The product must comply with all relevant accessibility and privacy regulations, including data usage and user privacy agreements.

Rationale: Compliance protects user rights, upholds institutional standards, and avoids legal or ethical issues.

Fit Criterion: All data collection and processing workflows must undergo compliance review, and accessibility features must align with recognized standards (e.g., WCAG).

4 Naming Conventions and Terminology

Glossary of Terms and Acronyms

General Project Terms

Client Group or individual funding or commissioning the system, typically the organization requesting the software.

Customer An individual or organization that purchases, subscribes to, or deploys the VoiceBridge system for use by end users. Customers may not directly participate in defining requirements but represent those who adopt or distribute the final product.

End User Someone who directly uses the system to assist with speech or browser tasks.

Persona Example character representing a user to demonstrate design needs, such as Amira with Parkinson's or David after a stroke.

Requirement Phase-In The process of introducing requirements in planned stages, from proof-of-concept to full deployment.

Stakeholder Any individual or group involved in or affected by the project, including users, caregivers, technical experts, and funding organizations.

VoiceBridge A software system that enables users with speech impairments to control browsers and devices using their voice, even if their speech is slurred or hard to understand. It is the technology whose requirements are described in this document.

Technical Terminology

API (Application Programming Interface) A set of rules allowing one software program to communicate, exchange services or data with another.

ASR (Automatic Speech Recognition) Technology that listens to speech and automatically converts it into written text.

Browser Agent Software/code that helps automate browser control tasks.

CLI (Command Line Interface) An interaction method using typed text to control a computer, usually for advanced settings.

JSON (JavaScript Object Notation) A simple data format both easy for humans and computers to read and write.

LLM (Large Language Model) Advanced AI designed for understanding and generating human language.

Microphone Input Audio signal collected from a user’s microphone for speech processing.

MVP (Minimum Viable Product) Basic version of software with enough features for user testing before further development.

PoC (Proof of Concept) A small system built to demonstrate that the main idea works in practice.

STT (Speech To Text) Another term for ASR, representing the conversion of speech into text.

Session The time period when a user is actively interacting with the system, such as from login to logout.

TTS (Text To Speech) Technology that transforms written text into spoken words using a computer-generated voice.

UML (Unified Modeling Language) Visual diagram style to represent system structure/workflow, mainly for programmers.

UUID (Universally Unique Identifier) Special ID assigned to sessions or user interactions for clear data tracking.

Medical Terminology

Aphasia Medical condition causing difficulties in communication, affecting speaking, writing, and understanding language.

ALS (Amyotrophic Lateral Sclerosis) Nervous system disease affecting movement, including speech, due to muscle weakness.

Dysarthria Speech disorder resulting in slurred/slow speech because of nervous system damage.

Parkinson’s Disease Progressive neurological disorder affecting movement and speech, often causing tremors and slurred speech.

Monopitch Limited variation in pitch; voice sounding flat, often linked to dysarthria.

Breathy Incomplete vocal fold closure can cause breathiness during speech.

Accessibility and Compliance

Accessibility Creating software everyone can use, including those with disabilities (e.g., clear text, screen reader support).

PIPEDA (Personal Information Protection and Electronic Documents Act) Canadian law regulating personal data collection, use, and protection.

WCAG (Web Content Accessibility Guidelines) Guidelines for making web content usable for people with disabilities; “Level AA” meets many key needs.

System Features and Components

Audit Logging Recording system actions/events to track problems or security issues.

Command Mapping System’s ability to recognize and map transcribed text to actions (e.g., “open Gmail”).

Data Encryption Protecting information by converting it into unreadable code except for authorized parties.

Error Handling Detecting, reporting, and recovering from mistakes or unexpected situations.

Latency Time between user speech input and system response.

Personalization Adjusting system behavior to match each user’s preferences, speech pattern, or needs.

Robustness/Fault-Tolerance Ability to keep working even with errors, noise, or network issues.

Scalability Ability for a system to support more users or new features over time.

Transcription Accuracy How correctly speech is transformed into text; higher accuracy means fewer mistakes.

Browser and Platform Terms

Browser Control Application Tool for users to control browsers with voice commands.

Cross-Browser Compatibility Ability for software to work across different browsers (Chrome, Edge, Firefox, etc.).

Device Command Instruction sent to a computer to perform a task, such as opening applications or typing.

Off-the-Shelf Software Ready-made software/components integrated into the project instead of building from scratch.

Miscellaneous Terms

Data Dictionary Documentation listing information types used by the system, with explanations.

Feedback Message Confirmation or notification shown to users, indicating actions or errors.

Intent The underlying meaning or goal of what the user says, interpreted by the system.

Training Data Audio recordings and transcripts used to improve speech recognition accuracy.

User Profile Data associated with each user, like preferences and speech patterns, for personalized command recognition.

Acronym Reference Table

Table 3 provides a consolidated list of all acronyms used throughout this document.

Acronym	Full Term	Simple Description
ALS	Amyotrophic Lateral Sclerosis	Disease causing muscle movement problems.
API	Application Programming Interface	Rules for different software to work together.
ASR	Automatic Speech Recognition	Converts spoken words into text automatically.
BUC	Business Use Case	Description of a scenario representing how a business or organization will use the system.
CER	Character Error Rate	Metric for evaluating speech recognition accuracy at the character level, useful for fine-grained analysis.
CLI	Command Line Interface	Computer control by typing commands.
FAQ	Frequently Asked Questions	Common questions and answers list.
JSON	JavaScript Object Notation	Easy data format for computers and humans.
LLM	Large Language Model	Advanced AI for understanding language.
MVP	Minimum Viable Product	First usable version with basic features.
PIPEDA	Personal Info Protection Document Act	Canadian privacy protection law.
PoC	Proof of Concept	A small-scale early prototype.
STT	Speech To Text	Converts speech into written text.
TTS	Text To Speech	Converts text to computer-generated spoken words.
UML	Unified Modeling Language	Visual diagrams for system planning.
UUID	Universally Unique Identifier	Special ID for sessions or users.
WCAG	Web Content Accessibility Guidelines	Rules for making web sites accessible.
WER	Word Error Rate	Metric for evaluating the accuracy of speech recognition by comparing transcribed words to the reference text.

Table 3: Acronym Quick Reference: A table defining the acronyms used in the VoiceBridge SRS document.

5 Relevant Facts And Assumptions

This section presents key facts about the users, environment, and operational context, along with the assumptions that guide system design. It includes user-centered facts, business rules, and design assumptions that influence functionality, usability, and system behavior. These elements ensure that the requirements are grounded in real-world conditions and support the intended goals of the system.

5.1 Relevant Facts

Fact ID	Fact	Explanation / Relevance
F1	Users experience speech impairments of varying severity.	VoiceBridge must handle varying speech clarity, from slurred to partially formed words (related to (G1), (G2)).
F2	Users want to communicate and navigate independently.	Motivates autonomy in using the system without reliance on caretakers (supports (G3)).
F3	Users may have limited mobility.	Hands-free operation improves accessibility and inclusion (supports (G3), (G4)).
F4	Users value simplicity and low cognitive load.	Commands should remain intuitive and easy to use (supports (G4)).
F5	Users may have emotional sensitivity around speech difficulty.	Interface should be respectful and encouraging without repeating goals (complements (G6)).
F6	Users expect privacy and dignity.	Data handling should preserve privacy and transparency (supports (G7)).
F7	Users may use different languages or accents.	System must accommodate linguistic diversity without bias (relates to (G1), (G8)).
F8	Users may use assistive tools concurrently.	VoiceBridge should integrate seamlessly with other accessibility tools (supports (G4), (G8)).
F9	Users will vary in technical comfort.	Onboarding should be minimal and low-friction (supports (G4), (G8)).
F10	Users appreciate visual feedback and control.	Feedback builds trust and reduces frustration (complements (G1), (G8)).

Table 4: User-Centered Facts for VoiceBridge: This table describes the facts linked to the users condition, expectations, and values relevant to their interactions with VoiceBridge.

5.2 Business Rules

ID	Business Rule	Rationale
BR1	Users must be able to cancel or stop a command at any time.	Empowers users and prevents frustration. (G3, G4)
BR2	Transcribed text must be displayed for user verification before executing critical commands.	Ensures accuracy, avoids misinterpretation, and maintains user confidence, inline with (G1)
BR3	Browser commands must not execute without user consent for actions with potential data impact (e.g., sending messages, closing tabs).	Protects user privacy and prevents accidental operations. (G2, G7)
BR4	The system should provide immediate visual feedback within 5 seconds of speech input.	Builds trust, transparency, and usability for impaired users. (G6, G1)
BR5	System must handle moderate background noise without significant degradation of performance.	Maintains reliability in real-world environments.(G1, G6)
BR6	Users should be able to configure simple command mappings for personalized tasks.	Supports individual preferences and improves user autonomy. (G2, G3, G8)

Table 5: Business Rules for VoiceBridge: This table defines the business rules identified in order to satisfy respective VoiceBridge goals, along with a rationale linking the relevant goal.

5.3 Assumptions

ID	Assumption	Implication for Design
A1	Users have access to a working microphone and modern browser.	The system assumes functional input hardware and browser APIs for speech capture (see G1 , G5).
A2	Users will tolerate minor transcription errors if quickly correctable.	Fast feedback and correction options are more important than perfect accuracy (see G1 , G6).
A3	Users are willing to train or calibrate the model briefly.	A short setup phase (e.g., sample phrases) can improve recognition quality (see G1 , G8).
A4	Users prefer transparent, explainable behavior.	VoiceBridge should indicate what command is being executed to prevent confusion or mistrust (see G8).
A5	Users may be in noisy or uncontrolled environments.	Noise-robust models and confirmation prompts are required to maintain reliability (see G1 , G6).
A6	Users want emotional ease of use.	Tone and interface language must feel supportive — e.g., “Let’s try again” instead of “Error” (see G6).
A7	Users will likely use the tool for daily web tasks.	The feature set should prioritize essential browser actions (navigation, scrolling, typing, tab control) (see G2 , G3 , G8).

Table 6: Design Assumptions for VoiceBridge: This table defines assumptions imposed on the user’s environment/preferences, and the implications of the assumption on the final design

6 The Scope of the Work

This section defines the boundaries and objectives of the VoiceBridge system. It describes the current limitations in browser interaction for users with speech impairments, the context in which the system will operate, and the partitioning of work into functional components. It also specifies the business use cases that guide system functionality from speech input to executed browser commands.

6.1 The Current Situation

Individuals with speech impairments currently rely on standard speech-to-text or manual input to use browsers and devices. Existing systems struggle with slurred or atypical speech, often requiring caregiver assistance.

Typical workarounds include typing commands, using alternative inputs, or correcting errors, leading to frustration and limited autonomy.

Current workflow: User speaks → Standard recognition → Frequent errors → Manual/caregiver correction → Action executed

VoiceBridge aims to replace this with accurate transcription, command mapping, and immediate feedback, enabling independent browser control.

6.2 The Context of the Work

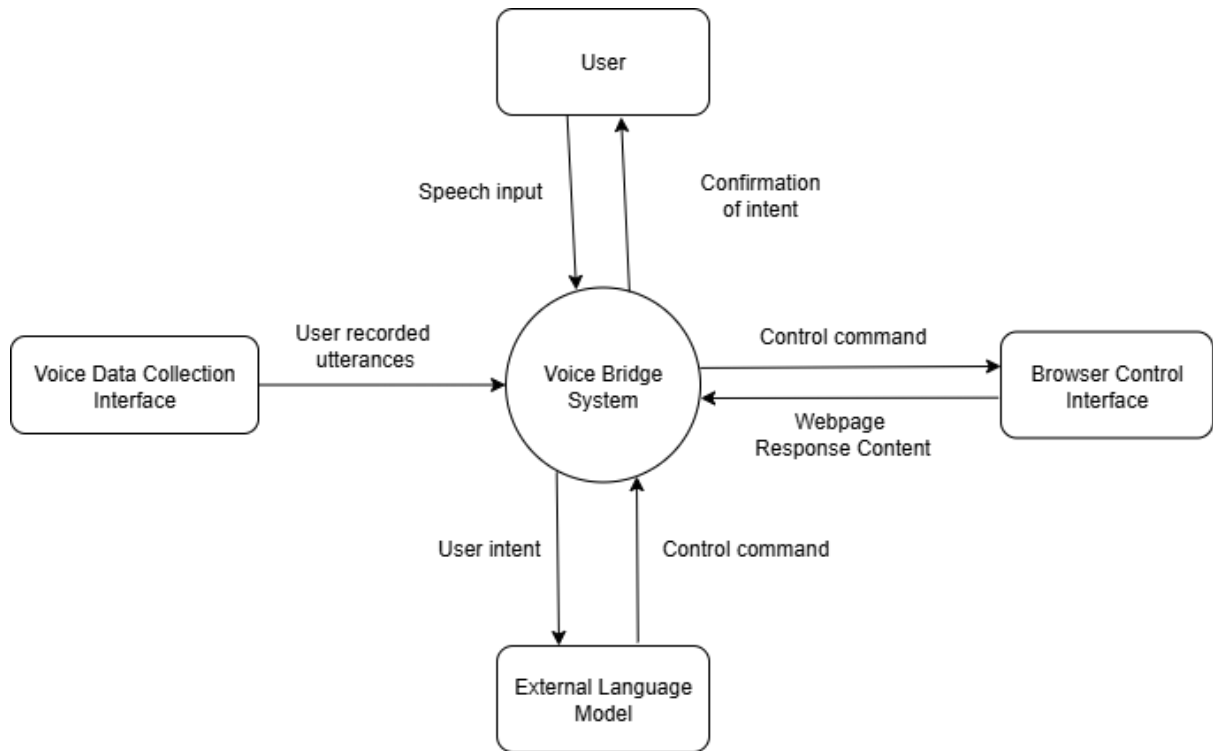


Figure 1: Context Diagram of VoiceBridge: This table defines the interfaces that interact within and external to the VoiceBridge System

6.3 Work Partitioning

Event Name	Input	Output	Brief Summary	BUC	Relevant Data Classes
Input Capture	User speech via microphone	Audio stream to system	Capture speech for processing	speech	Raw audio, timestamp
Speech-to-Text Modelling	Audio stream	Transcribed text	Convert impaired speech to text	impaired speech to text	Audio data, transcription
Intent Confirmation	Transcribed text	User confirmation	Verify intended command	intended command	Text data, confidence score
Command Mapping	Confirmed text	Actionable command	Map text to browser action	text to browser action	Command definitions, user preferences
Browser Interaction Layer	Actionable command	Executed browser action	Perform command in browser	command in browser	Command data, page context, execution status

Table 7: VoiceBridge Events, Inputs, Outputs, and Data Classes

6.4 Specifying a Business Use Case (BUC)

BUC ID	Name	Description
BUC-1	Capture Speech Input	The user initiates a speech session, and the system captures audio from the microphone for ASR processing.
BUC-2	Transcribe Speech to Text	The system converts captured audio into text through feature extraction and ASR model inference.
BUC-3	Interpret Intent	The system parses recognized text to determine the user's intent and prepares a structured intent object.
BUC-4	Execute Browser Command	The structured intent is mapped to a browser action, executed, and confirmed visually or audibly to the user.

Table 8: Business Use Cases (BUCs) for VoiceBridge

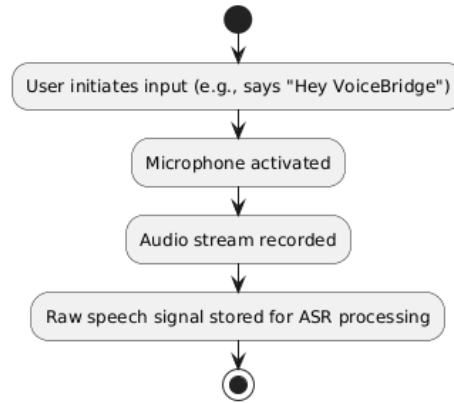


Figure 2: BUC-1: Capture Speech Input

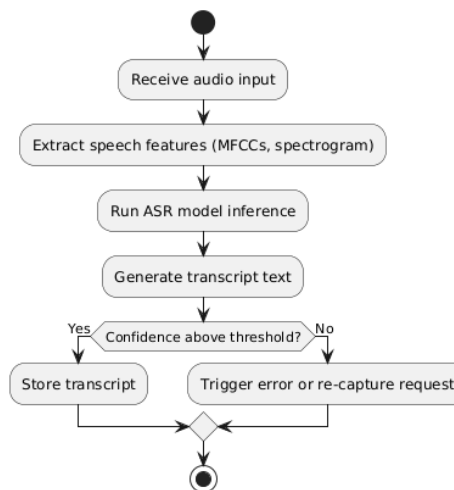


Figure 3: BUC-2: Transcribe Speech to Text

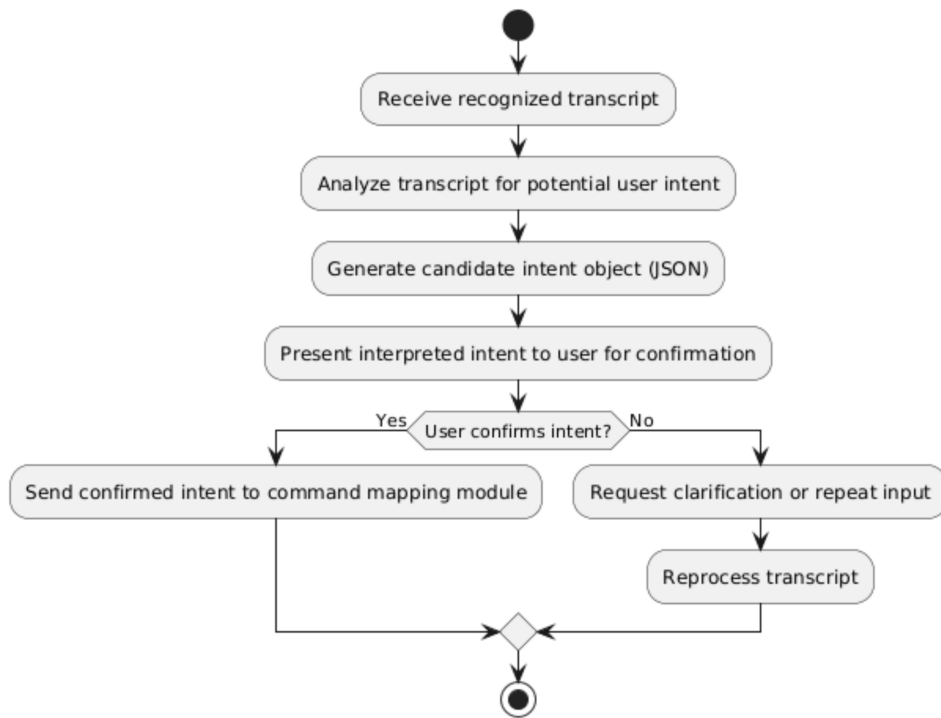


Figure 4: BUC-3: Interpret Intent

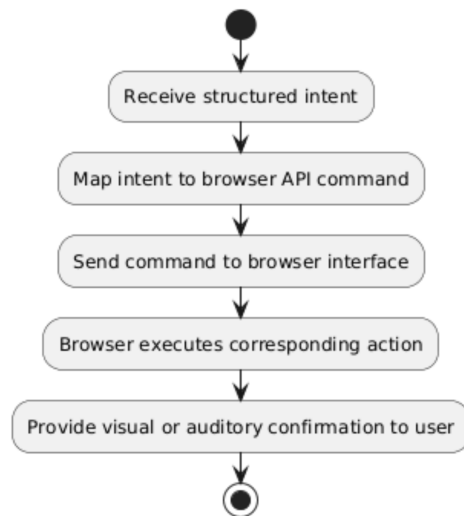


Figure 5: BUC-4: Execute Browser Command

7 Business Data Model and Data Dictionary

7.1 Business Data Model

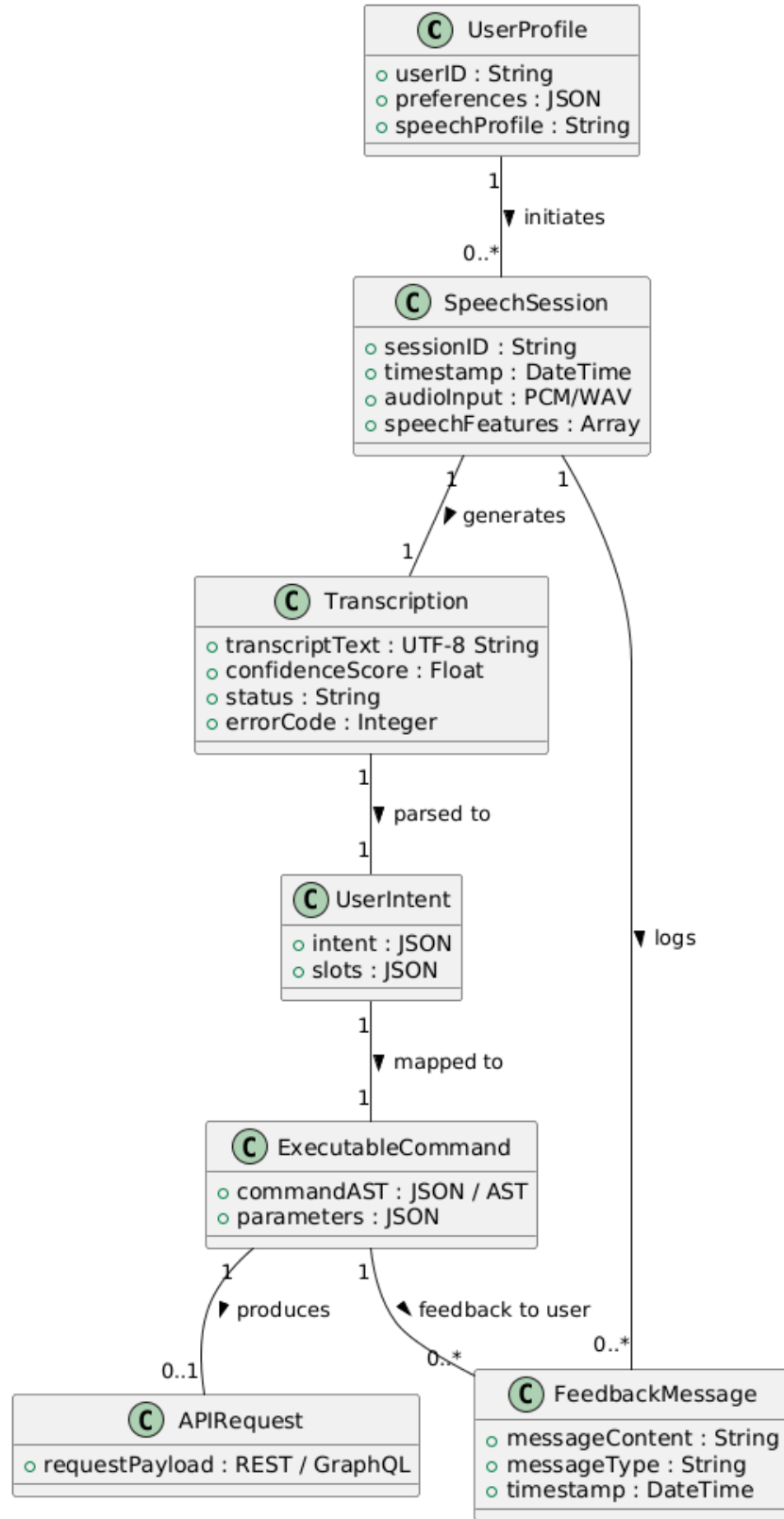


Figure 6: VoiceBridge UML Diagram.

The UML class diagram in [Figure 6](#) models the key data structures and relationships in the VoiceBridge system. It captures how user interactions flow through the system, from speech input to executed browser commands and user feedback.

- **UserProfile:** Stores individual user data, preferences, and speech characteristics.
- **SpeechSession:** Represents a single interaction instance, capturing audio input and derived speech features.
- **Transcription:** Stores the converted text, confidence scores, and status/error codes from speech recognition.
- **UserIntent:** Represents the parsed intent from the transcription, including semantic slots.
- **ExecutableCommand:** Maps the user intent to actionable browser commands with associated parameters.
- **APIRequest:** Represents optional requests generated by commands for external systems.
- **FeedbackMessage:** Logs system responses or messages to the user during the session.

Relationships and Rationale:

- **UserProfile & SpeechSession:** Each user may initiate multiple speech sessions. This models user activity over time and allows the system to track session history for personalization and auditing.
- **SpeechSession & Transcription:** Each session produces a single transcription. This ensures that each audio capture is processed into text for intent interpretation.
- **Transcription & UserIntent:** Each transcription is parsed into a user intent. This relationship models the mandatory step of interpreting what the user intends to do.
- **UserIntent & ExecutableCommand:** Each intent maps to a specific browser command. This ensures that interpreted user intentions are actionable within the system.
- **ExecutableCommand & APIRequest:** Some commands may produce an external API request. This optional relationship models integration with external services when required.
- **SpeechSession & FeedbackMessage:** Sessions may generate multiple feedback messages, such as transcription confirmations or error alerts, improving transparency and user trust.
- **ExecutableCommand & FeedbackMessage:** Executed commands may produce multiple messages, such as success notifications or warnings, keeping users informed about system actions.

This diagram complements the event and BUC tables by showing how data flows between components and how user interactions are represented in the system's data model.

7.2 Data Dictionary

Field	Description	Type	Source	Usage
Audio_Input	Raw speech signal captured from microphone	PCM, WAV, 16 kHz, 16-bit	User microphone	Primary input for ASR processing
Speech_Features	Extracted acoustic features (e.g., MFCCs, spectrogram)	N-dimensional array	Feature extractor module	Internal representation used by ASR engine
Transcript_Text	Recognized text output from speech input	UTF-8 encoded free text	ASR engine	Basis for interpreting user intent
Error_Code	Status or error indicator for system response	Integer	ASR engine / Controller	0 = Success, >0 = specific error type
User_Profile	Speaker-specific data such as ID, preferences, and speech patterns	JSON object	User database	Enables personalized recognition and response
Timestamp	Time marker for recognition or command event	DateTime	System clock	Used for logging, debugging, and tracking
User_Intent	Parsed intent derived from natural language input	JSON (intent and slots)	Interpreter module	Structured meaning representation (e.g., {"intent": "open_file", "file": "report.pdf"})
Exec_Command	Structured command before browser execution	JSON object or Abstract Syntax Tree	Interpreter module	Translates user intent into executable browser actions
API_Request	Outgoing request to external or browser API	REST / GraphQL formatted message	Interpreter module → API translator	Executes command or triggers action in external system
Feedback_Message	User-facing feedback or error output	String / Audio / JSON	VoiceBridge / UI	Confirms action success or requests clarification
Session_ID	Unique identifier for a single user interaction session	UUID / String	System controller	Links data objects across a single speech-to-command event

Table 9: VoiceBridge Data Dictionary

8 The Scope of the Product

8.1 Product Boundary

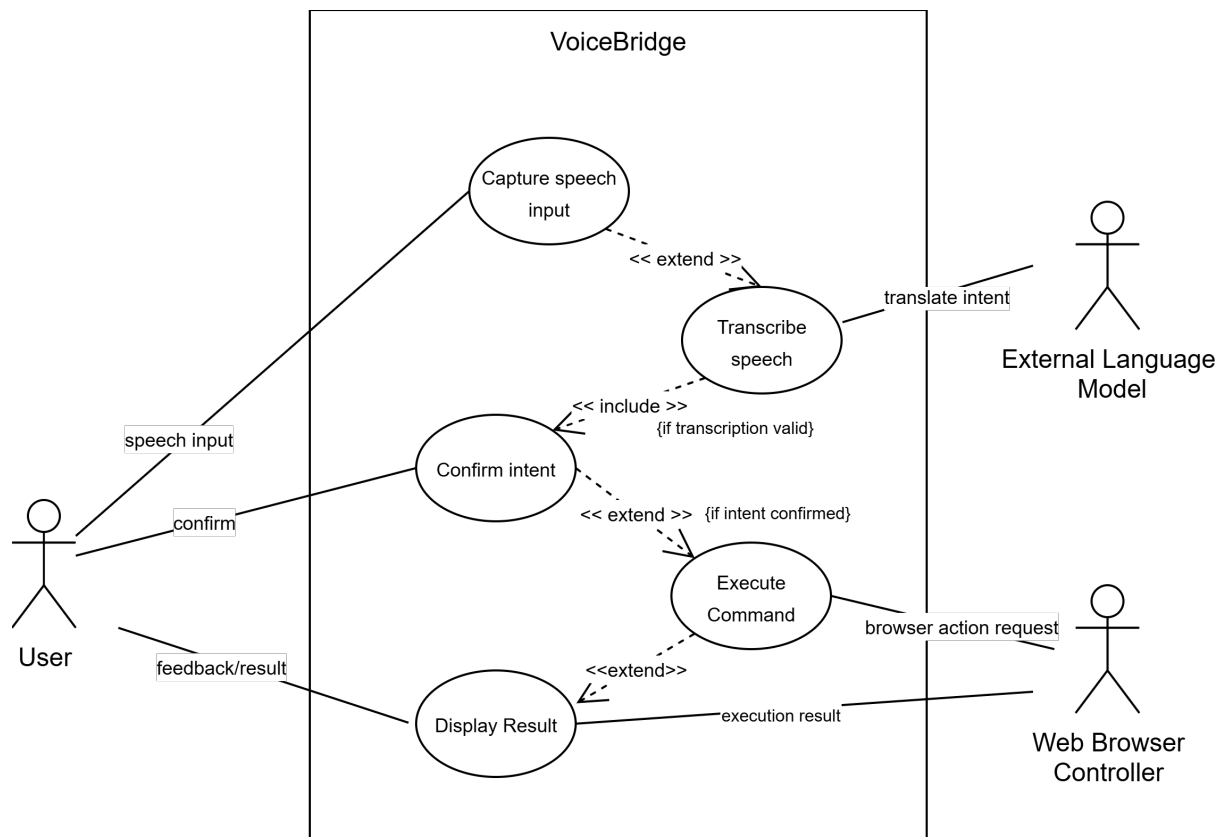


Figure 7: VoiceBridge Use Case Diagram. Each ellipse represents a Product Use Case (PUC) corresponding to a major system function.

8.2 Product Use Case Table

Table 10: Product Use Case (PUC) Table

PUC ID	Event Trigger	Input	Output	Description
PUC-1	User speaks into microphone	Audio signal	Captured speech data	System listens for user speech input.
PUC-2	Speech captured	Speech data	Text transcription	Converts impaired speech to text using trained model.
PUC-3	Transcription complete	Transcribed text	Confirmation prompt / feedback	Seeks user confirmation to ensure correct interpretation.
PUC-4	Intent confirmed	Confirmed command text	Browser command execution	Maps intent to actionable browser function.
PUC-5	Command executed or error	System response	Success or error feedback to user	Provides feedback or prompts retry on failure.

8.2.1 Individual Product Use Cases (PUC's)

This section defines the detailed Product Use Cases (PUCs) for the VoiceBridge system. Each PUC describes the interaction between the user and the system, including purpose, actors, triggers, inputs/outputs, and main scenarios.

PUC-1: Capture User Speech

Primary Actor: User

Trigger: User begins speaking into the microphone.

Precondition: The microphone is connected and permissions are granted.

Input: Audio signal (live speech).

Output: Captured audio data stream.

Postcondition: Audio data is made available for processing by the ASR (Automatic Speech Recognition) module.

Main Scenario:

1. User activates the VoiceBridge interface (e.g., presses a “Listen” button).
2. The system listens through the microphone for input.
3. The system captures the speech signal and stores it temporarily in memory.
4. Captured data is sent to the speech-to-text module for processing.

Alternative Flow:

- If the microphone is unavailable or access is denied, the system displays an error prompt.
- User can retry after adjusting permissions or hardware connection.

PUC-2: Convert Speech to Text

Primary Actor: System (Speech Recognition Module)

Trigger: Audio capture event completed.

Input: Captured speech data.

Output: Transcribed text.

Postcondition: Transcription results are ready for user confirmation.

Main Scenario:

1. System processes the captured audio stream using a trained ASR model.
2. Acoustic and linguistic features are extracted.
3. Speech is converted to text and stored temporarily.
4. Transcribed text is passed to the confirmation display.

Alternative Flow:

- If transcription confidence is below threshold, the system requests a repeat.

PUC-3: Confirm Transcription

Primary Actor: User

Trigger: System displays transcribed text.

Input: Transcribed text.

Output: User confirmation or correction.

Main Scenario:

1. The transcribed text is displayed to the user for verification.
2. User confirms that the text is correct or requests reprocessing.
3. System records the confirmation and proceeds to intent mapping.

Alternative Flow:

- If user rejects the transcription, the system returns to PUC-1 for re-input.

PUC-4: Map Intent to Browser Command

Primary Actor: System (Command Mapping Module)

Trigger: User confirmation received.

Input: Confirmed command text.

Output: Actionable browser command.

Main Scenario:

1. System parses confirmed text for intent (e.g., “open YouTube”).
2. System searches for matching browser or OS command.
3. Mapped command is passed to the execution layer.

Alternative Flow:

- If no matching command is found, the system provides suggestions.

PUC-5: Execute Command and Provide Feedback

Primary Actor: System

Trigger: Actionable command received.

Input: Command representation (API or accessibility call).

Output: Visible browser or OS action, and feedback message.

Main Scenario:

1. System executes the command using the browser's API or accessibility layer.
2. The target application or tab performs the intended action (e.g., opens a webpage).
3. System provides visual and/or auditory feedback to confirm success.

Alternative Flow:

- If the command execution fails, an error message or retry option is displayed.

9 Undesired Event Handling

Event ID	Description	Possible Cause	System Response / Handling Strategy	Severity	User Notification
UE-1	No Speech Detected	User is silent or microphone muted.	System times out after configurable interval (e.g., 5 s) and displays “No input detected. Please speak again.”	Low	Visual and optional audio cue.
UE-2	Unintelligible Speech / Low Confidence Transcription	Background noise, unclear articulation, or poor model match.	VoiceBridge requests repetition and highlights uncertain words. Logs instance for model improvement.	Medium	On-screen prompt: “Sorry, I didn’t catch that.”
UE-3	Misinterpreted Command	Incorrect speech-to-text mapping or ambiguous command phrase.	Requires confirmation before executing potentially disruptive browser actions (e.g., closing tab, submitting form).	High	Confirmation dialog: “Did you mean ‘Close tab’?”
UE-4	Microphone Access Denied	User or OS blocks microphone permission.	System notifies user and provides step-by-step instructions to enable microphone in browser/OS settings.	High	Visual warning banner.
UE-5	Browser Extension / API Failure	Browser crashes, extension disabled, or interface unresponsive.	Detect failure via heartbeat signal. Attempt reconnection; if unsuccessful, disable commands and show recovery notice.	High	“Browser connection lost — attempting to reconnect.”
UE-6	Network Connectivity Loss	Internet connection interrupted.	Offline mode activated: local commands (scroll, zoom, etc.) remain functional; external navigation disabled.	Medium	“Network unavailable — limited mode active.”
UE-7	System Crash / Model Process Failure	Unexpected error during ML inference or memory overflow.	System logs event, restarts inference module, and recovers previous session state when possible.	Critical	“VoiceBridge restarted after an error.”
UE-8	Data Privacy Breach Attempt	Unauthorized access to stored speech logs or transcripts.	Encrypts all local logs; triggers security alert, terminates unauthorized session, and notifies user.	Critical	“Security alert — your data is protected and the session has been closed.”
UE-9	Latency Exceeds Threshold	Inference or browser response exceeds acceptable delay (e.g., ≥ 2 s).	Display progress indicator; optionally degrade to simplified model or local processing fallback.	Low	Spinner with “Processing...” message.
UE-10	User Cancels Ongoing Command	User says “cancel” or “stop.”	Immediately halts current command execution and clears pending queue.	Low	Confirmation sound or text “Cancelled.”

Table 12: VoiceBridge User Event Handling Scenarios

10 Functional Requirements

10.1 FR1: Accept Speech Audio via Microphone

Description: The system must capture live speech input from the user through a standard built-in or external microphone.

Rationale: Without microphone input, the system cannot acquire the user’s speech for transcription. Requiring only standard/built-in microphones keeps the solution accessible and affordable.

Fit Criterion: The system reliably detects and records audio from default OS microphone devices across Windows, macOS, iOS, and Android, with a minimum 16 kHz sampling rate.

10.2 FR1: Accept Speech Audio via Microphone

Description: The system must capture live speech input from the user through a standard built-in or external microphone.

Rationale: Speech audio contains frequency components up to approximately 8 kHz. To capture this audio accurately without distortion, the sampling rate must be at least twice the maximum frequency, resulting in a minimum required rate of 16 kHz. This ensures reliable speech transcription and is supported by standard built-in and external microphone hardware.

Fit Criterion: The system reliably detects and records audio from default OS microphone devices across Windows, macOS, iOS, and Android, with a minimum 16 kHz sampling rate.

10.3 FR2: Convert Impaired Speech to Text with greater than or equal to 80% Accuracy (MVP)

Description: The system must process the captured audio and output a textual representation of the spoken utterance.

Rationale: Accurate transcription is the core functionality that enables communication and command execution. Without acceptable accuracy, the product fails its purpose.

Fit Criterion: In evaluation on a test dataset of impaired speech, transcription accuracy must reach at least 80% Word Error Rate (WER) reduction compared to baseline models, and achieve greater than or equal to 80% accuracy for common commands.

10.4 FR3: Display Transcription for Verification

Description: The transcribed text must be displayed in real time on the user’s device interface.

Rationale: Transparent feedback allows the user to verify correctness, catch errors, and build trust in the system.

Fit Criterion: Every spoken input is displayed within 2 seconds as text on the UI, with at least 95% consistency across trials.

10.5 FR4: Map Text to Arbitrary Device Commands

Description: The system must recognize when transcribed text corresponds to a predefined device action (e.g., “open amazon.ca,” “draft a new email”) and translate it into the appropriate command representation (API, CLI, or accessibility call).

Rationale: Mapping allows the system to extend beyond communication into real device control, empowering user independence.

Fit Criterion: For a test set of 50 predefined commands, the system maps user input to the correct command representation in at least 90% of cases.

10.6 FR5: Execute Commands on the Host Device

Description: The system must execute the mapped commands through the host device's accessibility framework or APIs, resulting in visible user action (e.g., app launch, scrolling, text entry).

Rationale: Without execution, the system remains a transcription tool only. Execution closes the loop between speech input and device interaction.

Fit Criterion: For each correctly recognized command, the intended system action occurs on the device within 2 seconds, with greater than or equal to 95% reliability across test scenarios.

10.7 FR6: Handle Security Software Interference

Description: The system must detect when browser, operating system, or antivirus security controls block or restrict automated actions and provide clear feedback to the user, including the affected command and available fallback options.

Rationale: Security software may prevent automated execution, which could otherwise result in silent failures and unsafe user assumptions about task completion. User awareness preserves safety and usability.

Fit Criterion: In test scenarios where automation is blocked, the system correctly detects the restriction and notifies the user in greater than or equal to 95% of cases within 2 seconds of the failed action.

11 Look and Feel Requirements

11.1 Appearance Requirements

The interface shall have a clean and minimal design to reduce cognitive load. Key elements should be visually distinct, with consistent spacing, and color usage to support quick recognition of actions.

Since it's a browser integration, the user interface should minimally interfere with the visibility of the content on the page. The interface should only capture the user's attention as functionally needed (i.e., listening to user prompts, confirming user intent), but should otherwise blend in with the browser interface.

11.2 Style Requirements

The system shall maintain a professional and neutral visual style suitable for general workplace use. Colors, icons, and fonts should prioritize clarity over branding at this stage. Future iterations may incorporate custom styling or theming.

12 Usability and Humanity Requirements

12.1 Ease of Use Requirements

The system should minimize user effort by providing a simple, intuitive interface. Key actions should be accessible within 3-4 interactions, with clear feedback after each action.

12.2 Personalization and Internationalization Requirements

The system should support basic personalization (e.g., remembering user preferences) and allow easy adaptation for different languages or regions at a later stage. For the PoC, English support is sufficient.

12.3 Learning Requirements

The system should be learnable within 10 minutes without prior training or documentation. Users should be able to complete core tasks on their first attempt through the interfaces navigation tutorial upon first time launch. Additional documentation should be supplemental but not necessary.

12.4 Understandability and Politeness Requirements

The system should use clear, direct, and neutral language in responses. Error messages or clarifications should remain polite and informative.

12.5 Accessibility Requirements

The interface should be navigable using standard assistive tools and offer clear text contrast and legible font sizes. Full accessibility compliance is not required at the PoC stage but should be feasible for future iterations.

13 Performance Requirements

13.1 Speed and Latency Requirements

Under normal operating conditions, latency requirements can be broken down into:

- Speech interpretation: 5 s after end of utterance

- Command generation & execution: 10-15 seconds after end of speech interpretation

13.2 Safety-Critical Requirements

The system shall enforce guardrails to prevent unsafe or unintended actions, requiring validation and user confirmation for potentially disruptive operations, and provide warnings or fail-safes for errors.

13.3 Precision or Accuracy Requirements

ASR accuracy shall be at least 70% in stationary noise conditions. The system shall achieve at least 80% command recognition precision under stationary noise conditions for the PoC.

13.4 Robustness or Fault-Tolerance Requirements

The system shall remain stable under fluctuating network conditions and noisy input. Fallback mechanisms (e.g., retry logic, error messaging) shall ensure graceful issue handling.

13.5 Capacity Requirements

The system shall support at least 20 concurrent users without service degradation if the product is commercialized.

13.6 Scalability or Extensibility Requirements

The system architecture shall allow horizontal scaling to handle increased traffic and modular extensions for new interaction capabilities.

13.7 Longevity Requirements

The system shall be designed to operate reliably over a minimum of 5 years, with maintainable and updatable components to support long-term product evolution.

14 Operational and Environmental Requirements

14.1 Expected Physical Environment

The product shall be operable in a variety of typical office, home, or institutional environments where users perform their daily tasks. The system shall be robust to stationary background noise such as air conditioning, computer fans, and ambient hum. It is not required to reliably handle non-stationary noise, including multiple people speaking or sudden loud interruptions. No modifications to the host operating system, browser, or network configuration shall be required.

14.2 Wider Environment Requirements

The primary interface shall be web-based, accessible via standard web browsers, to maximize user accessibility and support flexible use cases.

14.3 Requirements for Interfacing with Adjacent Systems

1. The system shall integrate with existing browser-based platforms and may interface with external language interpreter modules.
2. Open-source components (e.g., browser interaction agents and libraries) may be incorporated, ensuring compatibility and maintainability.

14.4 Productization Requirements

1. The product shall be deployable for multiple users within an organization, supporting secure user accounts and personalized ASR profiles.
2. The design shall allow packaging and distribution without requiring technical setup by end users. The product shall have straightforward installation or access via pre-configured web access and automatic model initialization.
3. Productization shall include logging suitable for monitoring performance and usage in a hosted environment.
4. The system shall include mechanisms for updates to features with minimal disruption to users.

14.5 Release Requirements

The product shall follow a defined release cycle, providing minor updates quarterly and major updates semi-annually. Each release must maintain backward compatibility with user data, personalization settings, and existing features. Release planning will account for maintenance effort, compute resources, and compliance obligations.

15 Maintainability and Support Requirements

15.1 Maintenance Requirements

Code must be modular, documented, and testable to support scalability, debugging, and future updates. Maintenance must be possible by developers who were not the original authors.

15.2 Supportability Requirements

The product shall provide an accessible Help Page and a Frequently Asked Questions (FAQ) section with clear instructions. It shall be displayed with high-contrast visuals, simple language, and auditory and visual aids.

15.3 Adaptability Requirements

The system must run on common workplace platforms via a web browser, including Windows 10 or later, macOS 12 Monterey or later, and Linux distributions such as Ubuntu 20.04 LTS or later, supporting modern web browsers.

16 Security Requirements

16.1 Access Requirements

Only authorized end users shall be able to access personalized ASR features, voice command execution, and saved transcripts. Access shall be role-based:

- **Primary users (end users):** can access their own data.
- **Secondary users (supervisors/SMEs):** may have read-only access to assist with support or troubleshooting.
- **Tertiary users (caregivers):** may have limited access to assist the end user, can view basic usage history and transcripts but cannot modify settings.

Access to external services (e.g., LLM APIs) shall be rate-limited to ensure system stability, control costs, and prevent abuse.

16.2 Integrity Requirements

The system must provide real-time confirmation and validation of commands before execution to ensure they match user intent.

16.3 Privacy Requirements

The system must gather explicit user consent for storing voice and personal data. All user data used for model improvement must be anonymized. Database and personalized ASR models must maintain integrity through secure, versioned backups. Unauthorized changes or corruption of user data must be prevented.

16.4 Audit Requirements

The system shall maintain secure logs of major actions and commands, including logging events, access to profile and personal data, and command execution failures. Logs shall be protected and retained in a secure database.

16.5 Immunity Requirements

The system must be resilient to accidental misuse. It shall handle noisy input robustly, avoid executing unintended commands, and operate safely within rate limits to prevent resource overload.

17 Cultural Requirements

17.1 Cultural Requirements

The system shall maintain a culturally neutral and respectful tone when prompting users, avoiding slang, bias, and discriminatory language. It must include ethical guardrails to prevent the generation of harmful content or execution of potentially dangerous commands.

It shall support inclusive and accessible design to serve users across diverse cultural backgrounds.

18 Compliance Requirements

18.1 Legal Requirements

The project shall comply with the Personal Information Protection and Electronic Documents Act (PIPEDA) regarding the collection, storage, and handling of personal information.

18.2 Standards Compliance Requirements

The application must comply with the Web Content Accessibility Guidelines (WCAG) 2.0, Level AA guidelines to ensure usability by individuals with disabilities.

19 Requirements Likely and Unlikely to Change

19.1 Likely to Change

- **12.1 Speed & Latency:** The system shall process speech input and return text in near real-time.
Rationale: Performance targets may improve as speech recognition and browser processing capabilities evolve.
- **14.3 Adaptability:** The system shall allow updates for new languages, user preferences, or accessibility features.
Rationale: Future user needs or new assistive technologies may require system modifications.
- **11.2 Personalization & Internationalization:** The system shall support preferred voice, accent handling, and language selection.
Rationale: User preferences and supported languages are likely to expand over time.
- **12.6 Scalability & Extensibility:** The system shall support an increasing number of concurrent users and new modules.
Rationale: User growth or additional features may require changes in infrastructure.
- **13.5 Release Requirements:** Updates shall be deployable on web browsers without downtime.
Rationale: Deployment strategies may change as development tools or environments evolve.

- **26.2 Personalized ASR Fine-Tuning:** The system shall allow individualized model tuning to improve transcription accuracy.
Rationale: Model improvements and user feedback may necessitate adjustments.
- **17.2 Standards Compliance Requirements:** System shall follow relevant industry standards.
Rationale: Standard adherence ensures safety, reliability, and credibility. As the product evolves, new features may require compliance with additional standards.

19.2 Unlikely to Change

- **9.1 Functional Requirements:** The system shall accurately transcribe speech to text for users with speech impairments.
Rationale: Core transcription functionality is foundational and will not change.
- **10.1 Appearance:** Interface shall be clean, minimal, and visually organized to reduce cognitive load.
Rationale: Minimal and clear design is essential for accessibility and usability.
- **10.2 Style:** Interface shall use consistent fonts, spacing, and colors to support clarity.
Rationale: Visual consistency is fundamental for usability.
- **11.1 Ease of Use:** Users shall complete tasks with minimal steps or cognitive effort.
Rationale: Streamlined interaction is a core accessibility requirement.
- **15.3 Privacy Requirements:** User data shall be encrypted in the database and logs.
Rationale: Privacy protection is legally required and critical for trust, this requirement will not change.
- **17.1 Legal Requirements:** System shall comply with relevant regulations (e.g., PIPEDA).
Rationale: Compliance is mandatory for this project, must be complied with at all times.

20 Open Issues

The primary open issue is maintaining high ASR accuracy across the diverse and severe spectrum of dysarthric speech. This requires extensive training and validation data, as well as careful model tuning.

21 Off-the-Shelf Solutions

21.1 Ready-Made Products

The project will evaluate existing, specialized speech recognition models and applications (e.g., Whisper model and Project Euphonia) as performance baselines.

21.2 Reusable Components

Potential reused components could include existing Text-to-Speech (TTS) modules for feedback and LLMs to map user commands into structured actions.

21.3 Products That Can Be Copied

Open-source browser automation agents (e.g., The AI browser agent) may be integrated for command execution via voice input.

22 New Problems

22.1 Effects on the Current Environment

The product shall operate without modifying the user's OS, browser, or network configuration. This separation prevents unintended impact from incorrect commands.

22.2 Effects on the Installed Systems

The product shall not bypass firewalls, alter security settings, access banned sites, install untrusted content, or perform any malware execution.

22.3 Potential User Problems

Due to the non-deterministic nature of dysarthric speech patterns, a user's exact speech type may not be fully captured by the model, leading to higher training overhead for personalization before the system becomes reliably usable for them. Users may experience misinterpretations requiring retries, which can cause frustration.

22.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Variability in dysarthric speech may necessitate frequent retraining of ASR models.

Real-time performance varies between devices, depending on processing power and low-latency operation.

22.5 Follow-Up Problems

The local operation limits the ability to perform remote problem diagnosis if any issues arise.

23 Tasks

23.1 Project Planning

Breakdown of major tasks:

Our main tasks are centered on preparing the system by training and tuning the dysarthric ASR model, integrating the core command interpreter, designing the accessible interface, and conducting rigorous testing.

23.2 Planning of the Development Phases

A detailed schedule of development phases, milestones, and dependencies is outlined in the PoC and Development Plan.

24 Migration to the New Product

24.1 Requirements for Migration to the New Product

The system shall support new users with no prior ASR and interpreter experience through an onboarding process.

The system shall support transitioning users migrating from an existing system by allowing for easy uploading of recordings of their speech training data.

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

All previous transcripts, audio files, and personalization data shall be tied to secure user accounts.

Users shall be able to access their data from any supported device after logging in and authentication.

25 Costs

The primary costs for this product are related to machine learning computation and hosting. Development and training will initially leverage Compute Canada credits provided by the supervising research team, along with Colab Pro (CAD 13.99/month).

No external hosting costs are anticipated for the initial release; however, optional hosting may be required for commercial distribution and deployments. If hosting is included, approximate costs for cloud-based deployment on Google Cloud Platform (GCP) include: training and inference on GPUs (\$50 CAD/month for pilot use), storage (\$3 CAD/month), and minimal networking (\$2–5 CAD/month). Costs scale with the number of users and training frequency. All estimates are approximate and intended for planning purposes.

26 User Documentation and Training

26.1 User Documentation Requirements

The system shall provide clear, accessible documentation, including an instructional guide and FAQ. (As noted in Section 14b, supporting requirements already cover this in detail.)

26.2 Training Requirements

Training shall not require formal instruction; the documentation and interface shall support self-directed onboarding.

27 Waiting Room

The system shall display launch, loading, or processing pages during transitions to provide users with clear feedback and reduce confusion during waiting periods.

28 Ideas for Solution

28.1 Browser-Based Extensions

The product is primarily web-based. Future considerations could include desktop or mobile apps to broaden the scope and accessibility (see Section 13 for environment and interface requirements).

28.2 Personalized ASR Fine-Tuning

Consider incremental model tuning paired with real-time streaming of speech input. This could be an approach to adapt to individual speech patterns while minimizing the training time overhead, related to the adaptability requirements discussed in earlier sections.

28.3 LLM Command Mapping

User intent parsing via context-aware LLMs could improve natural language command interpretation, as mentioned in Section 13.

28.4 Noise Filtering

Beyond stationary noise, adaptive filtering techniques could be adapted to enhance recognition in busier environments. Section 13 already specifies stationary noise handling.

28.5 Open-Source Integrations

Potential use of browser automation or voice control frameworks can be investigated, as mentioned in Section 16 for off-the-shelf solutions.

28.6 Accessibility Enhancements

Additional visual, auditory, or haptic cues for feedback and help features could enhance usability. Section 14 highlights basic supportability requirements that these enhancements would build on.

28.7 Data Encryption

Anonymized logging and privacy approaches can balance model improvement with privacy in mind. Section 15 discusses privacy requirements in detail.

29 Requirements Phase-In Plan

The following plan outlines the order in which the system requirements (9.1–17.2) will be addressed, aligned with project phases. Prioritization ensures that core functional requirements and critical non-functional requirements are validated first, with refinements and extensions added in later phases. Timelines for each phase are detailed in the Development Plan.

29.1 Phase Definitions

- **Proof of Concept (PoC):** Validate feasibility of core functionality and critical performance metrics.
- **Revision 0:** Expand functionality, address usability, accessibility, and integrate initial security/privacy mechanisms.
- **Revision 1:** Final refinement, full productization, compliance, cultural, and maintainability requirements.

29.2 Requirements Phase-In

1. **9.1 Functional Requirements:** Core system behaviors including input processing and basic transcription.
Phase: PoC
Rationale: Core functionality must be demonstrated to validate feasibility.
2. **10.1 Appearance Requirements:** Initial UI layout and interface elements.
Phase: Revision 0
Rationale: Visual interface can be refined after core functions are stable.

3. **10.2 Style Requirements:** Consistent fonts, colors, and spacing.
Phase: Revision 0
Rationale: Enhances readability and user experience once functionality is proven.
4. **11.1 Ease of Use Requirements:** Interface must be intuitive and straightforward.
Phase: Revision 0
Rationale: Important for usability testing, applied after functional PoC is complete.
5. **11.2 Personalization and Internationalization Requirements:** Language/voice adaptations.
Phase: Revision 1
Rationale: Applied later when system is functionally stable.
6. **11.3 Learning Requirements:** System should adapt to repeated user patterns.
Phase: Revision 1
Rationale: Machine learning or adaptation features require validated data pipelines.
7. **11.4 Understandability and Politeness Requirements:** Feedback should be clear and respectful.
Phase: Revision 0
Rationale: Improves user experience during early testing.
8. **11.5 Accessibility Requirements:** Interface supports assistive technologies.
Phase: Revision 0
Rationale: Early implementation ensures PoC testing is inclusive.
9. **12.1 Speed and Latency Requirements:** System responds within acceptable time frames.
Phase: PoC
Rationale: Performance validation is crucial for feasibility.
10. **12.2 Safety-Critical Requirements:** Ensure no harmful outputs or behaviors.
Phase: PoC
Rationale: Early validation prevents critical errors in PoC.
11. **12.3 Precision or Accuracy Requirements:** Transcription must meet minimum WER/CER.
Phase: PoC
Rationale: Accuracy is the central PoC metric.
12. **12.4 Robustness or Fault-Tolerance Requirements:** Handle errors gracefully.
Phase: Revision 0
Rationale: Necessary for reliable system use beyond PoC.
13. **12.5 Capacity Requirements:** Support a small set of concurrent users/data.
Phase: Revision 0
Rationale: Ensure system can handle expected early usage.
14. **12.6 Scalability or Extensibility Requirements:** System should allow future growth.
Phase: Revision 1
Rationale: Added after functional validation for longer-term planning.
15. **12.7 Longevity Requirements:** Components maintainable over time.
Phase: Revision 1
Rationale: Stability and maintainability are late-stage concerns.

16. **13.1 Expected Physical Environment:** System works on typical office/lab computers.
Phase: PoC
Rationale: Early testing on target devices is essential.
17. **13.2 Wider Environment Requirements:** Browser compatibility.
Phase: Revision 0
Rationale: Once core functionality works, broaden environment testing.
18. **13.3 Requirements for Interfacing with Adjacent Systems:** Integration points identified.
Phase: Revision 1
Rationale: External integration is late-phase after internal systems are stable.
19. **13.4 Productization Requirements:** Ready for deployment.
Phase: Revision 1
Rationale: Combines all prior validations for release.
20. **13.5 Release Requirements:** Packaging, distribution readiness.
Phase: Revision 1
Rationale: Ensures final delivery meets standards.
21. **14.1 Maintenance Requirements:** System maintainable.
Phase: Revision 1
Rationale: Implemented once core system is functional.
22. **14.2 Supportability Requirements:** Technical support procedures.
Phase: Revision 1
Rationale: Late-stage requirement after main functionality.
23. **14.3 Adaptability Requirements:** System can be upgraded.
Phase: Revision 1
Rationale: Flexibility comes after base system is validated.
24. **15.1 Access Requirements:** User roles and permissions.
Phase: Revision 0
Rationale: Early implementation secures initial PoC data.
25. **15.2 Integrity Requirements:** Prevent tampering or corruption.
Phase: Revision 0
Rationale: Ensures reliable data handling.
26. **15.3 Privacy Requirements:** Encrypt data in transit and at rest.
Phase: Revision 0
Rationale: Essential for user trust and legal compliance.
27. **15.4 Audit Requirements:** Track critical system events.
Phase: Revision 1
Rationale: Auditing implemented after basic functionality and privacy are validated.
28. **15.5 Immunity Requirements:** Protection from external attacks.
Phase: Revision 1
Rationale: Security hardening is a late-stage activity.
29. **16.1 Cultural Requirements:** Respect cultural norms and sensitivities.
Phase: Revision 1
Rationale: Applied after all functional, UI, and content elements are stable.

30. **17.1 Legal Requirements:** Comply with laws and regulations.
Phase: Revision 1
Rationale: Legal compliance verified after security, privacy, and functionality are validated.
31. **17.2 Standards Compliance Requirements:** Follow industry standards.
Phase: Revision 1
Rationale: Ensures credibility and certification readiness at final stage.

29.3 Requirement Implementation Timeline

The phasing of requirements for VoiceBridge follows a structured, incremental schedule aligned with prototype maturity and evaluation cycles. The table below summarizes the planned order and target completion dates for each phase.

Phase	Target Date	Focus and Included Requirements
Proof of Concept (PoC)	October–November 2025	Implement core functionality: speech input capture, transcription (FR9.1, 12.1–12.3, 13.1), and safety validation (12.2). Establish baseline latency, accuracy, and system feasibility.
Revision 0	December 2025–January 2026	Integrate user interface, accessibility, and initial security layers. Includes appearance (10.1–10.2), usability (11.1, 11.4–11.5), robustness (12.4–12.5), and privacy/integrity controls (15.1–15.3). Conduct early user testing.
Revision 1	February–April 2026	Expand system capability and compliance. Add personalization (11.2–11.3), scalability (12.6–12.7), interoperability (13.3–13.5), supportability (14.1–14.3), advanced security (15.4–15.5), and cultural/legal alignment (16.1, 17.1–17.2). Prepare for deployment.

This phased plan ensures that feasibility and safety are validated early (PoC), usability and inclusivity are developed next (Revision 0), and full scalability, compliance, and maintainability are achieved in the final iteration (Revision 1).

30 Formal Specification

This section provides a simplified formal model of the VoiceBridge system using a state-based representation. The goal is to describe how the system transitions between major states during operation.

30.1 System States

Let the system have the following states:

$$Q = \{Idle, Listening, Processing, Confirming, Executing, Error\}$$

The system starts in the *Idle* state and reacts to user or system inputs.

30.2 Inputs and Transitions

Inputs to the system are represented as:

$$\Sigma = \{Start, AudioInput, Confirm, Reject, Execute, Fail\}$$

The transition function $\delta(q, \sigma)$ describes how the system changes state:

$$\begin{aligned}\delta(Idle, Start) &= Listening \\ \delta(Listening, AudioInput) &= Processing \\ \delta(Processing, Confirm) &= Executing \\ \delta(Processing, Reject) &= Listening \\ \delta(Executing, Execute) &= Idle \\ \delta(Executing, Fail) &= Error\end{aligned}$$

30.3 Output Behavior

Each state produces a specific output:

$$\begin{aligned}Idle &\rightarrow \text{System ready prompt} \\ Listening &\rightarrow \text{Microphone active} \\ Processing &\rightarrow \text{Speech to text conversion} \\ Confirming &\rightarrow \text{User verification message} \\ Executing &\rightarrow \text{Command execution feedback} \\ Error &\rightarrow \text{Error or retry prompt}\end{aligned}$$

30.4 Mapping to Requirements

This formal model represents the main workflow of VoiceBridge and aligns with the following functional requirements:

- FR1: Accept Speech Input ($Idle \rightarrow Listening$)
- FR2: Convert Speech to Text ($Listening \rightarrow Processing$)
- FR3: Confirm Transcription ($Processing \rightarrow Confirming$)
- FR4: Map Intent to Command ($Confirming \rightarrow Executing$)
- FR5: Execute Command ($Executing \rightarrow Idle$)

This simplified formalization shows how user input moves through each stage of the system in a predictable, testable way, ensuring clarity and correctness of behavior.

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

Organizing the SRS using the Volere template went surprisingly smoothly. Breaking down each requirement and writing clear rationales helped us think through the project in a structured way. Separating likely-to-change requirements from unlikely ones also made the document more readable and easier to navigate.

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

Deciding which requirements might evolve over time versus those that would remain stable was a bit challenging. We resolved this by discussing potential future scenarios for the product and weighing Dr. Brodbeck's expert guidance, which made our decisions much more confident.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies?

Our requirements mostly came from our supervisor, Dr. Brodbeck, who is a linguistics expert and acts as our secondary stakeholder. We didn't consult end-users directly, so the requirements reflect expert advice on accessibility, usability, and system functionality.

4. Which of the courses you have taken, or are currently taking, will help your team succeed with this project?

- **SFWRENG 3RA3 – Software Requirements and Security:** Helped with writing clear Functional and Non-Functional Requirements and thinking about security, privacy, and reliability.
- **SFWRENG 4HC3 – Human-Computer Interfaces:** Provided a foundation for usability, stakeholder engagement, and accessibility considerations.
- Other courses in AI and Machine Learning are also valuable for speech recognition and natural language understanding.

5. What knowledge and skills will the team need to successfully complete this project?

We need to improve on:

- Speech and linguistics for accessibility.
- Web development, particularly browser extensions.
- Machine learning for speech recognition.
- Accessibility standards and usability testing.

- Clear documentation and team coordination.

6. **How will each team member acquire these skills or knowledge?**

- **Speech and linguistics:** Weekly guidance from Dr. Brodbeck and reading related literature.
- **Web development/browser extensions:** Hands-on prototyping, online tutorials, and sharing coding tasks among team members.
- **Machine learning:** Experimenting with pre-trained speech models and following online courses.
- **Accessibility standards:** Reviewing WCAG guidelines and testing our prototype with assistive technologies.