

Software Requirements Specification for Software Engineering: subtitle describing software

Team 13, Speech Buddies

Mazen Youssef

Rawan Mahdi

Luna Aljammal

Kelvin Yu

October 10, 2025

Contents

1	Purpose of the Project	vii
1.1	User Business	vii
1.2	Goals of the Project	vii
2	Stakeholders	vii
2.1	Client	vii
2.2	Customer	vii
2.3	Other Stakeholders	vii
2.4	Hands-On Users of the Project	vii
2.5	Personas	vii
2.6	Priorities Assigned to Users	vii
2.7	User Participation	viii
2.8	Maintenance Users and Service Technicians	viii
3	Mandated Constraints	vii
3.1	Solution Constraints	vii
3.2	Implementation Environment of the Current System	vii
3.3	Partner or Collaborative Applications	vii
3.4	Off-the-Shelf Software	vii
3.5	Anticipated Workplace Environment	vii
3.6	Schedule Constraints	vii
3.7	Budget Constraints	vii
3.8	Enterprise Constraints	viii
4	Naming Conventions and Terminology	viii
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project	viii
4.2	Technical Terminology	viii
4.3	Medical Terminology	viii
5	Relevant Facts And Assumptions	ix
5.1	Relevant Facts	ix
5.2	Business Rules	ix
5.3	Assumptions	ix

6	The Scope of the Work	ix
6.1	The Current Situation	ix
6.2	The Context of the Work	ix
6.3	Work Partitioning	ix
6.4	Specifying a Business Use Case (BUC)	ix
7	Business Data Model and Data Dictionary	ix
7.1	Business Data Model	ix
7.2	Data Dictionary	x
8	The Scope of the Product	x
8.1	Product Boundary	x
8.2	Product Use Case Table	x
8.3	Individual Product Use Cases (PUC's)	x
9	Functional Requirements	x
9.1	Functional Requirements	x
10	Look and Feel Requirements	x
10.1	Appearance Requirements	x
10.2	Style Requirements	xi
11	Usability and Humanity Requirements	xi
11.1	Ease of Use Requirements	xi
11.2	Personalization and Internationalization Requirements	xi
11.3	Learning Requirements	xi
11.4	Understandability and Politeness Requirements	xi
11.5	Accessibility Requirements	xi
12	Performance Requirements	xii
12.1	Speed and Latency Requirements	xii
12.2	Safety-Critical Requirements	xii
12.3	Precision or Accuracy Requirements	xii
12.4	Robustness or Fault-Tolerance Requirements	xii
12.5	Capacity Requirements	xii
12.6	Scalability or Extensibility Requirements	xiii
12.7	Longevity Requirements	xiii

13 Operational and Environmental Requirements	xiii
13.1 Expected Physical Environment	xiii
13.2 Wider Environment Requirements	xiii
13.3 Requirements for Interfacing with Adjacent Systems	xiii
13.4 Productization Requirements	xiv
13.5 Release Requirements	xiv
14 Maintainability and Support Requirements	xiv
14.1 Maintenance Requirements	xiv
14.2 Supportability Requirements	xiv
14.3 Adaptability Requirements	xv
15 Security Requirements	xv
15.1 Access Requirements	xv
15.2 Integrity Requirements	xv
15.3 Privacy Requirements	xv
15.4 Audit Requirements	xvi
15.5 Immunity Requirements	xvi
16 Cultural Requirements	xvi
16.1 Cultural Requirements	xvi
17 Compliance Requirements	xvi
17.1 Legal Requirements	xvi
17.2 Standards Compliance Requirements	xvi
18 Open Issues	xvii
19 Off-the-Shelf Solutions	xvii
19.1 Ready-Made Products	xvii
19.2 Reusable Components	xvii
19.3 Products That Can Be Copied	xvii
20 New Problems	xvii
20.1 Effects on the Current Environment	xvii
20.2 Effects on the Installed Systems	xvii
20.3 Potential User Problems	xviii
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	xviii

20.5 Follow-Up Problems	xviii
21 Tasks	xviii
21.1 Project Planning	xviii
21.2 Planning of the Development Phases	xviii
22 Migration to the New Product	xix
22.1 Requirements for Migration to the New Product	xix
22.2 Data That Has to be Modified or Translated for the New System	xix
23 Costs	xix
24 User Documentation and Training	xx
24.1 User Documentation Requirements	xx
24.2 Training Requirements	xx
25 Waiting Room	xx
26 Ideas for Solution	xx
26.1 Browser-Based Extensions	xx
26.2 Personalized ASR Fine-Tuning	xx
26.3 LLM Command Mapping	xx
26.4 Noise Filtering	xxi
26.5 Open-Source Integrations	xxi
26.6 Accessibility Enhancements	xxi
26.7 Data Encryption	xxi

Revision History

Date	Version	Notes
------	---------	-------

1 Purpose of the Project

1.1 User Business

Insert your content here.

1.2 Goals of the Project

Insert your content here.

2 Stakeholders

2.1 Client

Insert your content here.

2.2 Customer

Insert your content here.

2.3 Other Stakeholders

Insert your content here.

2.4 Hands-On Users of the Project

Insert your content here.

2.5 Personas

Insert your content here.

2.6 Priorities Assigned to Users

Insert your content here.

2.7 User Participation

Insert your content here.

2.8 Maintenance Users and Service Technicians

Insert your content here.

3 Mandated Constraints

3.1 Solution Constraints

Insert your content here.

3.2 Implementation Environment of the Current System

Insert your content here.

3.3 Partner or Collaborative Applications

Insert your content here.

3.4 Off-the-Shelf Software

Insert your content here.

3.5 Anticipated Workplace Environment

Insert your content here.

3.6 Schedule Constraints

Insert your content here.

3.7 Budget Constraints

Insert your content here.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

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

Insert your content here.

4.2 Technical Terminology

ASR - Automatic Speech Recognition

TTS - Text To Speech

STT - Speech To Text

4.3 Medical Terminology

Aphasia - A condition that robs you of the ability to communicate. It can affect your ability to speak, write and understand language, both verbal and written. Aphasia usually occurs suddenly after a stroke or a head injury. But it can also come on gradually, as in the case of a brain tumor or a progressive neurological disease.

ALS - Amyotrophic Lateral Sclerosis

Dysarthria - A motor speech disorder that makes it hard to speak. It is caused by damage to the nervous system, which can affect the muscles used for speaking. People with dysarthria may have slurred or slow speech, and they may have difficulty controlling the pitch, volume, and rhythm of their speech.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

Insert your content here.

5.2 Business Rules

Insert your content here.

5.3 Assumptions

Insert your content here.

6 The Scope of the Work

6.1 The Current Situation

Insert your content here.

6.2 The Context of the Work

Insert your content here.

6.3 Work Partitioning

Insert your content here.

6.4 Specifying a Business Use Case (BUC)

Insert your content here.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

Insert your content here.

8.2 Product Use Case Table

Insert your content here.

8.3 Individual Product Use Cases (PUC's)

Insert your content here.

9 Functional Requirements

9.1 Functional Requirements

Insert your content here.

10 Look and Feel Requirements

10.1 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.

10.2 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.

11 Usability and Humanity Requirements

11.1 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.

11.2 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.

11.3 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.

11.4 Understandability and Politeness Requirements

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

11.5 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.

12 Performance Requirements

12.1 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

12.2 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.

12.3 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.

12.4 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.

12.5 Capacity Requirements

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

12.6 Scalability or Extensibility Requirements

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

12.7 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.

13 Operational and Environmental Requirements

13.1 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.

13.2 Wider Environment Requirements

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

13.3 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.

13.4 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.

13.5 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.

14 Maintainability and Support Requirements

14.1 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.

14.2 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.

14.3 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.

15 Security Requirements

15.1 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.

15.2 Integrity Requirements

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

15.3 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.

15.4 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.

15.5 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.

16 Cultural Requirements

16.1 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.

17 Compliance Requirements

17.1 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.

17.2 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.

18 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.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

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

19.2 Reusable Components

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

19.3 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.

20 New Problems

20.1 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.

20.2 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.

20.3 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.

20.4 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.

20.5 Follow-Up Problems

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

21 Tasks

21.1 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.

21.2 Planning of the Development Phases

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

22 Migration to the New Product

22.1 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.

22.2 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.

23 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.

24 User Documentation and Training

24.1 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.)

24.2 Training Requirements

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

25 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.

26 Ideas for Solution

26.1 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).

26.2 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.

26.3 LLM Command Mapping

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

26.4 Noise Filtering

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

26.5 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.

26.6 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.

26.7 Data Encryption

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

Appendix — Reflection

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

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

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

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