Software Requirements Specification

for

Real Time Voice Translation

Prepared By: -
Sajal Shrivastava [22100BTCSE11746]
Sarthak Patel [22100BTCSE11757]
Shivkant Kurmi [22100BTCSE11768]
Shivam Dhangar [22100BTCSE11770]

Bachelor of Technology
Computer Science Engineering

Shri Vaishnav Vidyapeeth Vishwavidyalaya
Shri Vaishnav Institute of Information Technology

Guide: -
Ms. Varsha Choudhary

Version: 1.0

Date: [13/09/2025]
Location: Indore

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements for the **Real-Time Multilingual Voice and Sign Language Translation System**. The system facilitates communication across multiple Indian and global languages by integrating **speech-to-text, neural machine translation, text-to-speech, and sign language recognition**. This SRS ensures that developers, testers, researchers, and end users have a clear understanding of the system's features, constraints, and objectives.

1.2 Document Conventions

- **Font:** Times New Roman, Size 12
- **Headings:** Bold, Title Case
- **Subsections:** Numbered as per IEEE SRS standard (e.g., 1.1, 1.2)
- **References:** Numbered [1], [2] inline citations

1.3 Intended Audience and Reading Suggestions

- **Developers:** To implement translation, STT, TTS, and gesture recognition modules.
- **Testers:** To validate real-time performance and accuracy.
- **End Users:** To use the app for multilingual communication and accessibility.
- **Researchers:** To explore extensions in NLP, speech processing, and vision.

1.4 Product Scope

The system provides:

- Real-time **voice-to-voice translation** between Indian and global languages.
- **Sign language recognition** using deep learning and computer vision.
- **Dual modes**: Online (Google APIs) and Offline (Vosk, MarianMT, Coqui TTS).
- **Device support**: microphones, earbuds, cameras.
- **Applications**: education, healthcare, tourism, business, and accessibility.

By combining open-source tools and AI models, the solution addresses India's linguistic diversity and accessibility needs.

1.5 References

- Vosk Speech Recognition Toolkit – https://alphacephei.com/vosk/
- Hugging Face MarianMT Models – https://huggingface.co/models
- OpenCV Documentation – https://opencv.org/
- Google gTTS API – https://pypi.org/project/gTTS/

2. Overall Description

2.1 Product Perspective

This system is an independent, **real-time AI-based translation platform** that integrates speech recognition, neural machine translation, and gesture recognition. Unlike existing tools, it supports both **voice and sign inputs**, as well as online/offline operation.

2.2 Product Functions

- Record speech and convert it into text.
- Translate text into target language.
- Convert translated text into voice.
- Capture sign language via camera and convert into text + voice.
- Provide user-friendly interface for device and language selection.

2.3 User Classes and Characteristics

- General Users: Need real-time translation for daily communication.
- Students: For multilingual learning and inclusive classrooms.
- Healthcare Professionals: To interact with patients across language barriers.
- Differently-abled Users: To communicate through sign language detection.

2.4 Operating Environment

- Hardware: Standard laptop/desktop with microphone, camera, and speakers.
- OS: Windows/Linux/macOS.
- Libraries: Python, Streamlit, OpenCV, Hugging Face Transformers, Vosk.
- GPU support for efficient offline translation and gesture recognition.

2.5 Design and Implementation Constraints

- Limited sign language datasets for Indian gestures.
- Real-time processing speed dependency on hardware.

- Internet required for online mode.

2.6 Considerations; Design Conventions or Programming Standards

- Programming Language: Python 3.x, due to its extensive support for NLP, deep learning, and audio/video processing libraries.
- Coding Standards: PEP 8 (Python Enhancement Proposal) will be followed for code readability, naming conventions, and documentation.
- Version Control: Git will be used for source code management, ensuring proper collaboration, version history, and rollback capability.
- Dependency Management: Virtual environments (venv/conda) will be used to isolate and manage project dependencies consistently.
- Modular Design: The system will follow a modular architecture, separating concerns into speech recognition, translation, TTS, and gesture recognition modules.
- Documentation: Inline code comments and Markdown-based project documentation (README, API usage, system guide) will be provided.
- Error Handling: Exception handling will be implemented for robust real-time performance in both online and offline modes.
- Security & Privacy: No user data (audio/text) will be stored persistently; temporary files will be auto-deleted post-processing.
- Maintainability: Since the system may later be maintained by the customer's organization, clean code structure, API abstraction, and modularity will reduce technical complexity.

2.7 User Documentation

- User Manual (PDF/Docx):
  Provides step-by-step instructions for installation, configuration, device selection (microphone, camera, earbuds), and usage of online/offline modes.
- Quick Start Guide (One-Page PDF):
  A concise guide covering the essential steps for starting the application, selecting languages, and performing real-time translations.
- Online Help (In-App Documentation):
  Tooltips and descriptive texts embedded in the Streamlit web interface to guide users in recording, translating, and managing files.
- Video Tutorial (MP4/YouTube Link):
  A short demo video illustrating how to use the app for voice translation and sign language detection.
- Technical Documentation (Markdown/HTML):
  Includes API references, system architecture details, model setup (Vosk, MarianMT, TTS), and developer notes for future maintenance.

- FAQ and Troubleshooting Guide (PDF/HTML):
  Covers common user issues such as microphone not working, camera setup errors, or offline models not being detected.

2.8 Assumptions and Dependencies

The following assumptions and dependencies are identified for the development and deployment of the **Real-Time Multilingual Voice and Sign Language Translation System**:

- **Assumptions**
  - Users will have access to devices with a **microphone, camera, and speakers/earbuds** for proper functioning.
  - A **stable internet connection** will be available for online mode (Google APIs for STT, translation, and TTS).
  - Offline models such as **Vosk (for STT), MarianMT (for translation), and Coqui TTS (for text-to-speech)** can be pre-installed on the user's system if internet is unavailable.
  - Users will have **basic computer literacy** to interact with the Streamlit-based interface.
  - The hardware used (e.g., laptops with Intel i5/RTX GPU or equivalent) will be sufficient to handle real-time processing requirements.

- **Dependencies**
  - **Third-party libraries and frameworks** such as SpeechRecognition, Vosk, Hugging Face Transformers, gTTS, Coqui TTS, and OpenCV are essential for system functionality. Any major changes, version incompatibility, or discontinued support for these libraries could affect the project.
  - **Python runtime and environment dependencies** (virtual environment, package management with pip/conda).
  - **External APIs** (Google Speech Recognition and Google Translate API) are required for online mode. The project's performance and accuracy in online mode depend on the availability and policies of these APIs.
  - **Dataset availability** for Indian Sign Language (ISL) gestures is crucial for training and improving gesture recognition accuracy.
  - **Operating system support**: Windows, Linux, and macOS must provide compatible drivers for camera, microphone, and GPU acceleration.
  - Future maintainability may depend on the **customer organization's ability** to manage dependencies, update libraries, and retrain/fine-tune models as required.

3. External Interface Requirements

3.1 User Interfaces

- Streamlit-based web UI.
- Dropdowns for selecting languages and input devices.
- Buttons for recording, translating, and deleting audio.

3.2 Hardware Interfaces

- Microphone/Earbuds: Speech input.
- Camera: Sign gesture input.
- Speakers/Headphones: Audio output.

3.3 Software Interfaces

- SpeechRecognition & Vosk: Speech-to-text.
- Google Translate API & MarianMT: Translation.
- gTTS & Coqui TTS: Text-to-speech.
- OpenCV: Gesture detection.

3.4 Communications Interfaces

- Internet connection for online mode (Google APIs).
- Local processing for offline mode (Vosk, MarianMT, TTS).

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Voice Translation\

- Input: User's speech.
- Processing: STT → Translation → TTS.
- Output: Translated voice + text display.

4.2 Sign Language Detection

- Input: Camera captures gesture.
- Processing: Gesture recognition → Text mapping.
- Output: Text + synthesized voice.

4.3 Online Mode

- Uses Google APIs for high-accuracy STT, translation, and TTS.

4.4 Offline Mode

- Uses Vosk, MarianMT, and Coqui TTS for low-resource environments.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Real-time translation (<2 seconds latency).
- Accuracy >85% for speech and >80% for sign gestures.

5.2 Safety Requirements

- No harmful operations; safe for general users.

5.3 Security Requirements

- User data (audio/text) not stored permanently.
- Offline mode ensures privacy.

5.4 Software Quality

- Usability: Easy-to-use Streamlit UI.
- Reliability: Stable offline mode.
- Portability: Runs on Windows/Linux/macOS.
- Scalability: Additional languages/models can be added.

6. Other Requirements

- Dataset preparation for Indian Sign Language.
- Fine-tuning MarianMT for Indian languages.
- Optimizing real-time inference on GPU-enabled devices.

Appendix A: Glossary

- STT (Speech-to-Text): The process of converting spoken words into written text using AI-based models.
- TTS (Text-to-Speech): The process of converting written text into audible speech.
- NLP (Natural Language Processing): A field of AI that enables machines to understand, interpret, and generate human language.
- Vosk: An open-source offline speech recognition toolkit.
- MarianMT: A neural machine translation framework developed by Microsoft and available via Hugging Face.
- gTTS (Google Text-to-Speech): A Python library for interfacing with Google's text-to-speech API.
- Coqui TTS: An open-source text-to-speech system based on deep learning.
- OpenCV: Open-source library for computer vision tasks such as image and gesture recognition.
- ISL (Indian Sign Language): The standardized form of sign language used in India.
- Streamlit: An open-source framework for building interactive data-driven web applications in Python.

Appendix B: Analysis Models

1. Use Case Diagram – Illustrates the interaction between users, the system, and external devices (microphone, camera, speakers).

2. Sequence Diagram – Shows the flow of messages between components (STT, Translator, TTS, Output) during translation.

3. System Architecture Diagram – Block diagram depicting the flow from speech/sign input to translated text/voice output.

4. Data Flow (DFD – Level 0):

    - Input: Voice or Gesture.
    - Process: STT → Translation → TTS OR Gesture Recognition → Text → TTS.
    - Output: Translated text and/or synthesized voice.

Appendix C: To Be Determined List

- Dataset Availability: A large-scale, high-quality dataset for Indian Sign Language gestures is required for accurate recognition.
- Model Fine-Tuning: Need to determine the scope and compute resources for fine-tuning MarianMT on Indian languages.
- Performance Benchmarks: Final latency and accuracy thresholds for offline mode need to be tested and validated.
- Cross-Platform Support: Mobile app deployment feasibility (Android/iOS) to be evaluated.
- Integration with Video Conferencing Tools: Potential expansion for real-time multilingual meetings is under consideration.
- Licensing Issues: Verification of open-source licenses for Vosk, MarianMT, and Coqui TTS before deployment.