

Software Requirements Specification (SRS) Document

AI Tool/Mobile App for Indian Sign Language (ISL) Generation from Audio-Visual Content in English/Hindi and Vice-Versa

1. Introduction

1.1 Purpose

The purpose of this document is to provide the detailed requirements for developing an AI tool or mobile application capable of converting audio-visual content in English or Hindi into Indian Sign Language (ISL) content and vice-versa. The application aims to bridge the communication gap for the deaf and hard-of-hearing community.

1.2 Scope

The AI tool will be used to:

- Convert spoken English/Hindi audio or video into ISL videos.
- Convert ISL video into text/audio in English or Hindi.
- Enable bi-directional communication between hearing individuals and the deaf or hard-of-hearing community.
- Operate on mobile platforms for easy access and portability.

1.3 Definitions, Acronyms, and Abbreviations

- **ISL:** Indian Sign Language
 - **AI:** Artificial Intelligence
 - **ASR:** Automatic Speech Recognition
 - **NLP:** Natural Language Processing
-

2. Overall Description

2.1 Product Perspective

The system will use AI technologies such as ASR for converting spoken language into text, NLP for language understanding, and computer vision for recognizing sign language gestures. It will serve as a communication aid for deaf users by translating audio or video into ISL and translating ISL back into spoken or textual content.

2.2 Product Features

1. **Audio-to-ISL Conversion:**
 - Translate spoken English/Hindi into ISL gestures.
 - Generate ISL videos from text/audio input.

2. **Video-to-Text/Audio Conversion:**
 - Recognize ISL gestures from video input and convert them into corresponding text or speech.
3. **Bi-directional Translation:**
 - Facilitate real-time conversations between ISL users and non-sign language speakers.
4. **Customizable Language Preferences:**
 - Support for regional variations in ISL.
 - Language selection (Hindi/English) for text/audio output.
5. **Offline Mode:**
 - Option to use basic features offline for areas with limited internet connectivity.

2.3 User Classes and Characteristics

- **Deaf/Hard-of-Hearing Users:**
Primary users who need ISL for communication.
- **Hearing Users:**
Individuals who need to understand ISL or communicate with ISL users.
- **Interpreters and Educators:**
Professionals who can use the app for teaching and translation purposes.

2.4 Operating Environment

- Mobile platforms (Android and iOS).
 - Minimum Android version: 9.0 (Pie)
 - Minimum iOS version: 13.0
 - Requires camera and microphone access for real-time translation.
 - Internet connectivity for advanced AI-based features.
-

3. Functional Requirements

3.1 Audio to ISL Video Translation

- **Input:** Audio in English or Hindi.
- **Process:**
 - Speech recognition converts audio to text.
 - Text is processed using NLP to understand context.
 - ISL gestures are generated based on recognized speech and context.
- **Output:** ISL video (3D model or human-like avatar performing signs).

3.2 ISL Video to Text/Audio Translation

- **Input:** Video of a user performing ISL.
- **Process:**
 - AI-based gesture recognition converts sign language to text.
 - Optional NLP for grammatical corrections and context analysis.
- **Output:** Text or synthesized audio in English or Hindi.

3.3 Text to ISL Video Translation

- **Input:** Typed or pasted text in English/Hindi.
- **Process:** Text is processed, and corresponding ISL gestures are generated.
- **Output:** ISL video (animated avatar).

3.4 Real-time Conversations

- **Input:** Audio or video from both parties.
- **Process:**
 - Audio is converted into ISL for deaf users.
 - ISL is converted into audio/text for hearing users.
- **Output:** Synchronized real-time translation for both users.

3.5 User Customization

- Allow users to adjust the speed of ISL gestures.
 - Users can select specific regional dialects of ISL.
 - Option to replay translations for clarity.
-

4. Interface Requirements

4.1 Hardware Interfaces

- Device camera for gesture recognition.
- Microphone for audio input.

4.2 Software Interfaces

- Integration with NLP libraries, speech-to-text APIs (e.g., Google Speech API), and gesture recognition models.
 - Cloud-based AI models for improved accuracy (optional).
-

5. Performance Requirements

5.1 Performance Requirements

- Real-time translation with a delay of no more than 2-3 seconds.
- Video processing for ISL gesture recognition should be smooth and fast, even on lower-end devices.

5.2 Reliability

- The app should have high accuracy in translation (>90%).
- It should function offline for basic operations like text-to-ISL translation.

5.3 Usability

- The interface should be intuitive and easy to use for both hearing and non-hearing individuals.
- Instructions should be clear and available in both text and video format.

5.4 Security

- Ensure privacy of user data, especially video inputs that might contain personal information.
- Encrypted storage of sensitive information.

5.5 Compatibility

- Compatible with different mobile screen sizes.
- Ensure cross-platform functionality between Android and iOS.

6. Design Constraints

6.1 Modular Architecture

The system will be divided into key modules:

1. **ASR/NLP Module:** Handles audio recognition and language processing.
2. **Gesture Recognition Module:** Detects and interprets ISL signs.
3. **Sign Generation Module:** Converts audio or text into animated ISL.
4. **User Interface Module:** Provides the interface for interaction and control.

6.2 Data Flow

1. **Audio Input** → **Speech Recognition** → **NLP** → **Gesture Generation** → **ISL Output**
2. **ISL Input** → **Gesture Recognition** → **Text Generation** → **Audio/Text Output**

7. Non-Functional Attributes

7.1 Training and Support

- Provide tutorial videos for users on how to use the app.
- In-app customer support for troubleshooting.

7.2 Updates and Maintenance

- Regular updates to improve AI translation accuracy.

- Support for new versions of ISL and regional dialects.
-

8. 8. Schedule and Budget

8.1 Schedule

- Project duration: 6-8 months.
- Key phases:
 - Requirements Gathering: 1 month
 - Design & Architecture: 1.5 months
 - Development: 3 months
 - Testing & Debugging: 1.5 months
 - Deployment: 1 month

8.2 Budget

- Development Team: \$50,000
- Software & Tools: \$10,000
- Testing: \$8,000
- Maintenance: \$7,000
- Total Estimated Budget: \$75,000