Software Requirements Specification for GestureAI

Team 7Petabytes
Team Members: Arun Soma (Team Lead), Laxminarasimha Soma, Taris Major

May 3, 2025

Contents

1 Revision History

2 Introduction

This document is the Software Requirements Specification (SRS) for the GestureAI project. GestureAI is a system designed to recognize and translate American Sign Language (ASL)

Revision	Date	Author	Description
1.0	2025-03-08	Arun Soma	Initial version of the Software Requirements Specification (SRS) document for GestureAI.
1.1	2025-03-11	Taris Major	3.1(8) Requirement removed.
1.2	2025-03-11	Laxminarasimha Soma	3.3(2(c)) Database changed from SQLite to PostgreSQL. Requirement changed.
1.3	2025-05-01	Arun Soma	Only text modules implemented; audio modules discarded for initial development.

Table 1: Revision History

into text and speech. The system comprises both a mobile application and a hardware prototype featuring a display component to show the translated text.

2.1 Purpose

The purpose of this document is to provide a detailed specification of the functional and non-functional requirements for GestureAI system. It is intended for developers, testers, and stakeholders to ensure a common understanding of the system's objectives and constraints.

2.2 Scope

GestureAI aims to improve communication accessibility for Deaf and hard-of-hearing individuals by:

1. Training an AI model using TensorFlow to recognize ASL gestures.
2. Converting recognized gestures into text and speech in real-time.
3. Prototyping a hardware display (using a Raspberry Pi camera and a Waveshare OLED screen) for showing text output.
4. Supporting both mobile (iOS/Android) and desktop platforms.
5. Implement AI-driven ASL recognition using TensorFlow and OpenCV.
6. Convert ASL gestures into text and speech with a latency below 100ms.
7. Feature an interactive avatar for real-time ASL translation.
8. Ensure high usability and compliance with accessibility standards (WCAG 2.1).

2.3 References

1. IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998.
2. `https://www.w3.org/TR/WCAG21/` (WCAG 2.1)
3. PPT for GestureAI project by Team 7Petabytes.
4. Feasibility Study for GestureAI project by Team 7Petabytes.

2.4 Overview

The remainder of this document is organized as follows:

1. Section 2 provides an overall description of the system.
2. Section 3 details the specific requirements.
3. Appendices include any additional supporting information.

3 Overall Description

3.1 Product Perspective

GestureAI is an AI-powered application designed to recognize and translate ASL into text and speech. It consists of both a mobile application and a hardware prototype with a display for real-time communication accessibility. It is designed to work on modern operating systems (Windows 10 or higher, macOS) and leverages both mobile and desktop platforms.

3.2 Product Functions

The major functions of GestureAI include:

1. Capturing live video input via a camera.
2. Recognizing ASL gestures in real-time using a trained TensorFlow model.
3. Converting recognized gestures into text.
4. Generating speech from the translated text.
5. Displaying the translated text on a hardware component.
6. Managing and storing training data and system logs.

3.3 User Characteristics

The intended users of GestureAI are:

1. Deaf and hard-of-hearing individuals.
2. Communication facilitators (e.g., interpreters, educators).
3. End-users who require real-time translation in public or private settings.

Users are expected to have basic familiarity with mobile or desktop applications.

3.4 Assumptions and Dependencies

1. The system will have reliable internet connectivity for API calls and model updates.
2. Users will have access to mobile devices, meeting the specified hardware and software requirements.
3. Dependencies include Python packages (TensorFlow, OpenCV), React Native libraries, and necessary hardware drivers.

4 Specific Requirements

4.1 Functional Requirements

1. **Live Sign Language Translation(High Priority)**
   (a) The system shall provide real-time sign language translation into text.
   (b) The system shall provide real-time sign language translation into audio.
2. **ASL Support(High Priority)**
   (a) The system shall support American Sign Language (ASL).
   (b) The system shall provide accurate ASL translations for common phrases and expressions.
3. **Cross-Platform Accessibility(High Priority)**
   (a) The system shall be available on mobile platforms (iOS/Android).
   (b) The system shall be available on desktop platforms.
4. **User Account Management(High Priority)**
   (a) Users shall be able to create an account.
   (b) Users shall be able to edit their account details.
5. ~~**AR Glasses Support (Medium Priority)**~~
   (a) The system shall display real-time subtitles or captions on AR glasses.
   (b) The system shall ensure minimal delay in displaying translations on AR glasses.
6. **User Feedback Submission(Medium Priority)**
   (a) Users shall be able to submit feedback regarding system performance.
   (b) Users shall be able to report translation inaccuracies.
7. **Integration with Public Establishments(Medium Priority)**
   (a) The system shall be integrable with hospitals.
   (b) The system shall be integrable with judicial offices and other public establishments.
8. **Animated Avatar for Translation(High Priority)**
   (a) The system shall provide an animated avatar that translates text into sign language.
   (b) The system shall provide an animated avatar that translates audio into sign language.

4.2 Non-Functional Requirements

1. **Low Latency Performance(Medium Priority)**
   (a) The system shall offer real-time translation with minimal delay.

(b) The system shall ensure efficient processing to reduce lag.

2. **Intuitive User Interface(High Priority)**

   (a) The system shall have a user-friendly interface.

   (b) The system shall provide an easy-to-navigate menu structure.

3. **System Performance Monitoring(High Priority)**

   (a) Admins shall be able to monitor system latency and performance.

   (b) Admins shall receive real-time alerts for performance issues.

4. **AI Model Management(High Priority)**

   (a) Admins shall be able to update the AI models for better accuracy.

   (b) Admins shall have access to model performance metrics.

5. **ASL Dataset Customization(Medium Priority)**

   (a) Admins shall be able to update the ASL dataset.

   (b) Admins shall ensure dataset modifications are reflected in real-time.

6. **Future-Proofing and Updates(Medium Priority)**

   (a) The system shall be designed to support future updates.

   (b) The system shall allow for seamless integration of improvements.

7. **User Data Analysis(Low Priority)**

   (a) The system shall analyze user feedback to improve performance.

   (b) The system shall generate reports on user interactions for optimization.

8. **GPU(High Priority)**

   (a) The system shall use NVIDIA RTX 6000 ADA (18,176 CUDA cores, 568 Tensor Cores).

4.3 Constraints

1. **Hardware Constraints:**

   (a) System must use Windows 10 or higher, MacBook M1 or higher.
   (b) Camera must use Raspberry Pi High-Resolution Camera.
   (c) Display must use Waveshare 0.96-inch OLED.
   (d) Server must use DigitalOcean (8GB RAM, 2TB Storage).
   (e) Storage must use a minimum of 4 TB hard-drive space.

2. **Software Constraints:**

(a) Frontend must use React Native (JavaScript, XML).
(b) Backend must use Python.
(c) Database must use PostgreSQL.
(d) Version Control must use Git.
(e) AI Training must use TensorFlow.
(f) Object Detection must use OpenCV.
(g) API routes must use FASTAPI implementation

4.4 Performance Requirements

1. The gesture recognition process must operate in real-time with a latency of less than 100 ms.
2. The system shall maintain an accuracy rate of at least 95% in gesture recognition.
3. The text-to-speech conversion should occur within 200 ms after gesture recognition.

4.5 Other Requirements

1. **Security:** The system must use `bcrypt`
2. **Usability:** The UI must be user-friendly and accessible to users with varying levels of technical expertise.
3. **Maintainability:** The codebase shall be maintained under version control (Git) and follow industry-standard coding practices.

5 Appendices

5.1 Appendix A: Glossary

1. **ASL:** American Sign Language.
2. **AI:** Artificial Intelligence.
3. **AR:** Augmented Reality
4. **GUI:** Graphical User Interface.
5. **OpenCV:** An open-source computer vision library.
6. **PostgreSQL:** A self-contained SQL database engine.
7. **SRS:** Software Requirements Specification.
8. **TensorFlow:** An open-source machine learning framework.