Project Report for EquaSpace

April 5, 2025

Service Oriented Systems

Course Instructor: Dr. Abhishek Srivastava

Indian Institute of Technology Indore

Group 8

Team Members:

Abhinav Kumar (210001002)

Anjani Kumar (210001004)

Jilagam Dinesh Venkar Kumar (210001026)

Hrishesh Sharma (210003037)

Mrityunjay Dwivedi (2430201004)

Contents

1 Introduction

In today's digital age, video calling has become a fundamental mode of communication, yet many existing platforms fall short when it comes to accessibility for people with special needs. To bridge this gap, we have developed a video calling application specifically designed to support individuals with special needs. The application includes sign language translation for users who cannot speak, real-time audio-to-caption conversion for those who cannot hear, screen-sharing explanation using generative AI for visually impaired users, and a unique feature that converts chat messages into vibration-based Morse code to assist users who cannot see. By integrating these innovative features into a single platform, our goal is to create a more inclusive communication experience and ensure that everyone, regardless of ability, can participate equally in digital conversations.

2 Features

Our video calling application integrates several accessibility-focused features designed to support users with hearing or visual impairments. Each feature is built with the goal of enhancing inclusivity, communication clarity, and independence for people with special needs. To ensure maximum usability, all core features are both **icon-activated** and **voice-activated**, allowing users—especially those who are blind or visually impaired—to easily access functionalities without relying on visual cues.

2.1 Real-time Video and Audio Conferencing

Description: Enables high-quality, low-latency video and audio communication using advanced codecs and adaptive streaming. Ensures smooth, uninterrupted conversations even in poor network conditions.
Impact: Enhances virtual collaboration by providing seamless communication, reducing disruptions in meetings, and improving user experience across varying bandwidths.

2.2 Sign Language Translation

Description: The application uses computer vision to recognize and translate sign language gestures into readable text in real time during a video call.
Impact: This feature bridges the communication gap between sign language

users and non-signers, enabling more natural and efficient conversations without the need for an interpreter.

2.3 Audio-to-Caption Conversion

Description: Real-time speech-to-text transcription converts spoken dialogue into on-screen captions.
Impact: This provides immediate access to spoken content for deaf and hard-of-hearing users, making conversations more accessible and reducing communication barriers.

2.4 Screen Sharing Explanation Using Generative AI

Description: When a user shares their screen, the application uses generative AI to interpret and explain the visual content in descriptive language.
Impact: This helps visually impaired users understand shared content such as documents, presentations, or interfaces, enabling them to actively follow and contribute to discussions.

2.5 Chat-to-Vibration-Based Morse Code

Description: Incoming chat messages are translated into Morse code and communicated through device vibrations.
Impact: This feature supports users who are blind or unable to read screen text, allowing them to receive messages through tactile feedback and stay engaged in the conversation.

3 Technical Implementation

3.1 Real-time Video and Audio Conferencing

Overview: This feature enables high-quality, low-latency video and audio communication by leveraging advanced codecs and adaptive streaming techniques. It ensures smooth and uninterrupted conversations even under challenging network conditions.

Implementation Details:

1. The platform captures real-time audio and video streams from participants.

2. Advanced codecs compress the media while maintaining quality.

3. Adaptive streaming algorithms adjust bitrate and resolution dynamically based on network bandwidth.

4. Error correction and jitter buffers are employed to minimize delays and packet loss.

Technologies Used:

- Communication Protocols: WebRTC, RTP
- Programming Languages: JavaScript

3.2 Sign Language to Text Conversion

Overview: This feature enables real-time translation of hand gestures into English text using deep learning and hand tracking. It also includes auto-correction and text-to-speech functionality.
Implementation Details:

1. The feature continuously captures frames from the webcam.

2. Hand gestures are detected and tracked using the HandDetector module from cvzone.

3. The hand region is isolated and passed to a trained CNN model which classifies the gesture into corresponding alphabet characters.

4. The system maintains a sentence history and uses the pyenchant spellchecker to suggest corrections for incomplete or misspelled words.

5. The output is displayed as live text and can also be converted to speech implemented using a text-to-speech (TTS) engine (pyttsx3).

3.3 Real-Time Speech-to-Text Transcription

Overview: This feature enables real-time transcription of spoken audio into text using the Whisper model. It allows users—especially those with hearing impairments—to read spoken content in a continuously updating text format.
Implementation Details:

1. The application records microphone audio in short, fixed-size chunks (100ms) using PyAudio.

2. The audio is buffered for a short duration (e.g., 5 seconds), allowing contextual inference without significant delay.

3. When enough audio is collected, it is passed to OpenAI's Whisper model for transcription in a background thread to ensure smooth, non-blocking performance.

4. The process continues in real time until the user manually stops the recording.

Technologies Used:

- Speech Recognition Model: Faster-Whisper, a fast CTranslate2-based version of OpenAI's Whisper

- Audio Input: PyAudio (microphone capture)

- Concurrency: Python threading module for parallel transcription

3.4 Screen Sharing Explanation Using Generative AI

Overview: This feature provides visually impaired users with a detailed description of shared screen content by utilizing a state-of-the-art vision-language model.

Implementation Details:

1. The application captures a screenshot of the shared screen.

2. The image is sent via an API request to the **LLaMA-3.2-90B-Vision-Preview** model hosted by Grok.

3. The model processes the image and generates a textual description of the content.

4. The generated response is either:

   - Displayed as text for sighted users.
   - Converted into speech using a text-to-speech (TTS) system for blind users.

Technologies Used:

- Vision-Language Model: LLaMA-3.2-90B-Vision-Preview (via Grok API)

- Screenshot Capture: Python-based screen capture libraries

3.5 Chat-to-Vibration-Based Morse Code

Overview: This feature converts incoming chat messages into Morse code and communicates them via system vibrations, allowing blind users to perceive text through tactile feedback.
Implementation Details:

1. The application captures incoming text messages from the chat.
2. The text is converted into Morse code using a predefined dictionary mapping characters to their Morse representations (dots and dashes).
3. The system triggers device vibrations corresponding to the Morse code pattern:
    - Short vibration for a dot (`.`)
    - Long vibration for a dash (`-`)
4. Python is used to handle text-to-Morse translation and system API calls to control vibrations.

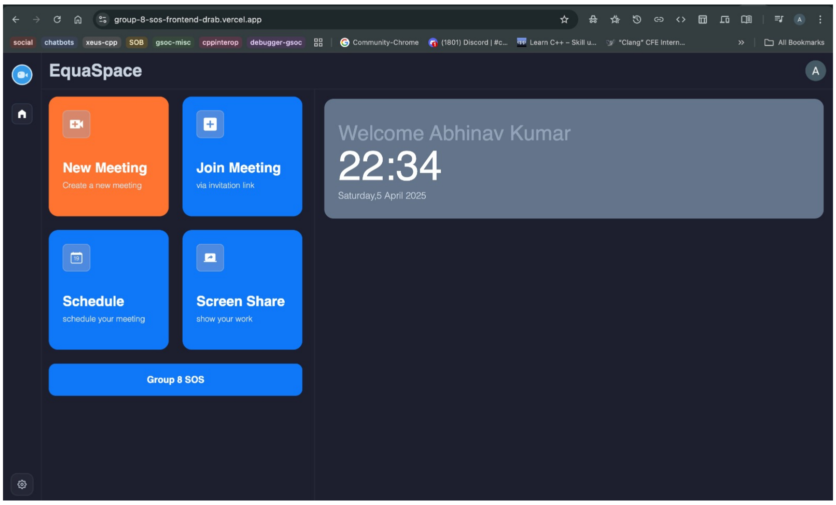
Technologies Used:

- Programming Language: Python, Bash
- Morse Code Encoding: Dictionary-based mapping

4 Application Features Demo

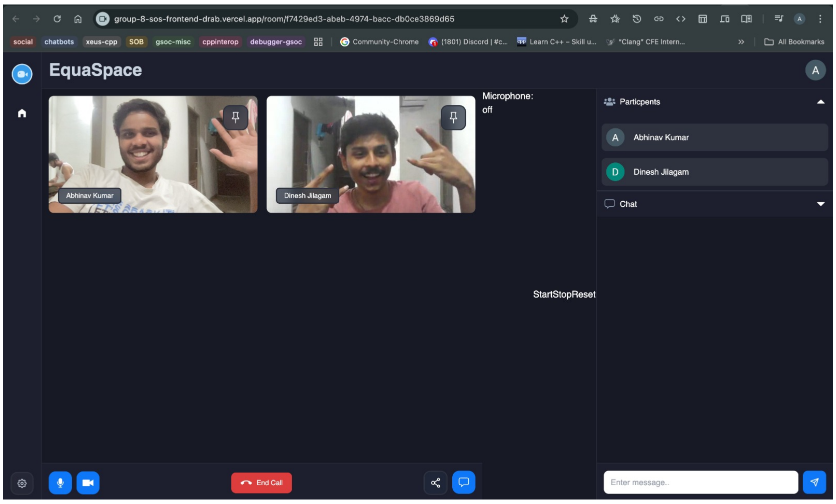
4.1 Homepage

It showcases image of the homepage with features like New Meeting, Join Meeting, Schedule and Screen Share.



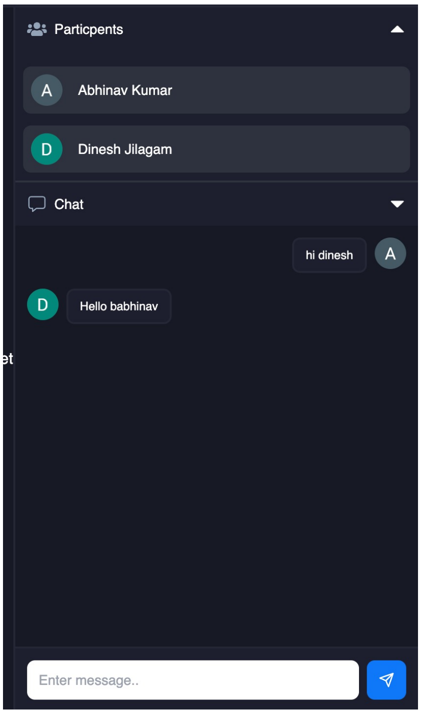
4.2 Room

This image showcases two participants engaged in a video call with options for chat, participant management, and call controls.



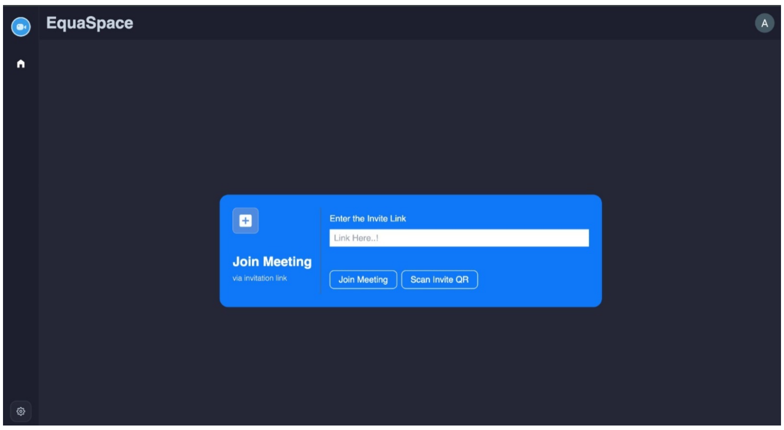
4.3 Chat Feature

Users in same room can chat with each other. It is based on websockets in backend-server.



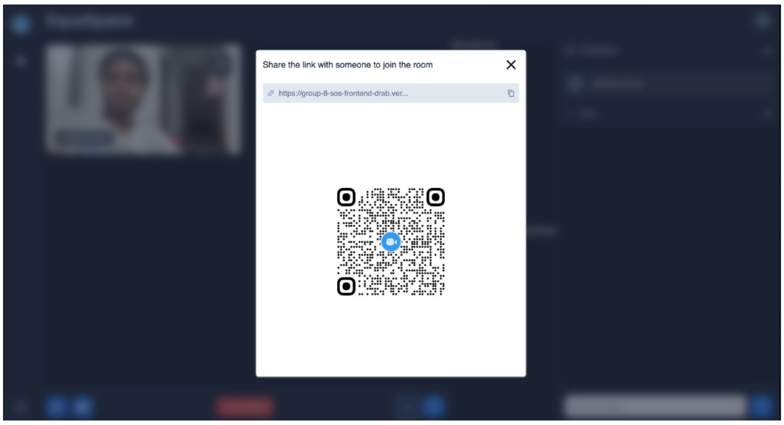
4.4 Join Feature

Users can join via pasting the Shared-URL or just by scanning QR code.



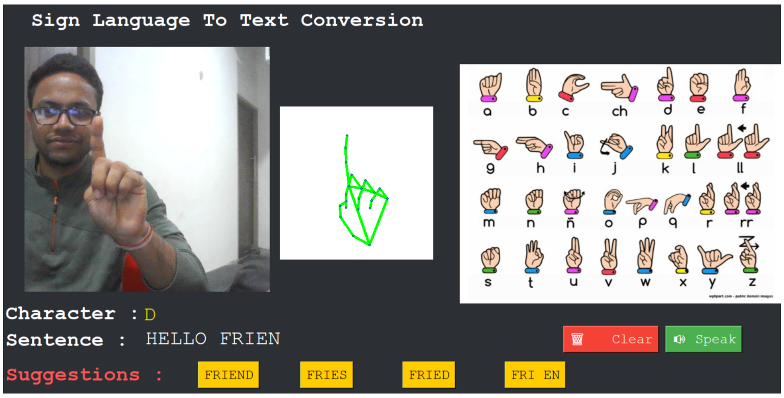
4.5 Invite Feature

Customers can invite others via URL or by sharing QR-code.



4.6 Sign Language

A user is performing a sign gesture that is detected and converted into text.



5 Github Repository

The complete source code and project files can be found at : Github Link

6 Future Scope

While the current implementation of our video calling application significantly enhances accessibility for users with special needs, there is considerable potential for further development and refinement. Future improvements can focus on both expanding functionality and increasing the system's adaptability to diverse user requirements. Below are several key areas for future work:

- **Multilingual Sign Language Support:** Currently, the system may support only a specific sign language (e.g., ASL). Future versions can include multilingual support for various regional sign languages, such as BSL, ISL, or LSF, to broaden its global applicability.
- **Context-Aware Screen Explanation:** The generative AI module for screen explanation can be improved to become more context-sensitive by adapting descriptions based on the type of content being shared (e.g., documents vs. videos vs. code).
- **Haptic Feedback Customization:** The Morse code chat system can be extended with customizable vibration patterns or Braille display support to better suit different user preferences and comfort levels.
- **Emotion and Tone Detection:** Integrating emotional tone analysis from voice or facial expressions could help convey the sentiment of the speaker to users who rely on text or haptic communication.

By exploring these future directions, the application can continue to evolve into a fully inclusive, intelligent communication platform that addresses the diverse needs of all users.

7 Conclusion

In this project, we have developed an innovative video calling application designed to enhance accessibility for individuals with special needs. By integrating features such as sign language translation, real-time audio-to-caption conversion, generative AI-based screen explanation, and chat-to-vibration Morse code conversion, the application ensures a more inclusive communication experience.

While the current implementation successfully addresses key accessibility barriers, there is significant scope for further improvements. Enhancing sign language recognition, supporting multiple languages, refining generative AI explanations, and integrating cross-platform compatibility are future directions that can elevate the application's impact.

By prioritizing inclusivity and leveraging state-of-the-art AI technologies, this application contributes to bridging the digital divide for people with disabilities. As accessibility continues to gain importance in technology, this project serves as a foundation for further research and development in creating universally accessible communication platforms.