B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

Object Oriented Modelling Mini Project Report

Indian Sign Language(ISL) Generator

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Sumit Kumar Chaudhary (1BM22CS296)
Tushar Tyagi (1BM22CS311)
Yash Kumar Sinha (1BM22CS334)
Yashraj Sinha (1BM22CS335)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2024-2025

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Sumit Kumar Chaudhary (1BM22CS296), Tushar Tyagi (1BM22CS311), Yash Kumar Sinha (1BM22CS334), Yashraj Sinha (1BM22CS335), students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this Object Oriented Modelling Mini Project entitled "Indian Sign Language(ISL) Generator" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester Sep 2024- Jan 2025. I also declare that to the best of our knowledge and belief, the OOM mini Project report is not from part of any other report by any other students.

Signature of the Candidate

Sumit Kumar Chaudhary (1BM22CS296)
Tushar Tyagi (1BM22CS311)
Yash Kumar Sinha (1BM22CS334)
Yashraj Sinha (1BM22CS335)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Object Oriented Modelling Mini Project titled “Indian Sign Language(ISL) Generator” has been carried out by Sumit Kumar Chaudhary (1BM22CS296), Tushar Tyagi (1BM22CS311), Yash Kumar Sinha (1BM22CS334), Yashraj Sinha (1BM22CS335) during the academic year 2024-2025.

Signature of the Faculty in Charge

Table of Contents

Chapter 1: Problem Statement

In India, millions of individuals with hearing or speech impairments rely on Indian Sign Language (ISL) as their primary means of communication. However, the language barrier between ISL users and non-sign language users often results in significant challenges in social, educational, and professional interactions.

The objective is to develop an intelligent application capable of converting audio, video, or text inputs into Indian Sign Language gestures (animated or visual) and vice versa. The application should leverage advanced technologies such as natural language processing (NLP), speech recognition, computer vision, and machine learning. It must be accessible, real-time, and user-friendly for seamless communication between ISL users and non-users.

The application aims to bridge the communication gap, promote inclusivity, and empower individuals with disabilities, fostering their integration into mainstream society while promoting ISL literacy.

Chapter 2: Software Requirement Specification

Software Requirements Specification (SRS) Document

AI Tool/Mobile App for Indian Sign Language (ISL) Generation from AudioVisual 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. General 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

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

2.4 Operating Environment

1. Mobile platforms (Android and iOS).

2. Minimum Android version: 9.0 (Pie).
3. Minimum iOS version: 13.0.
4. Requires camera and microphone access for real-time translation.
5. 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

- 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 System Architecture

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

6.2 Data Flow

- Audio Input → Speech Recognition → NLP → Gesture Generation → ISL Output
- 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. Schedule and Budget

8.1 Schedule

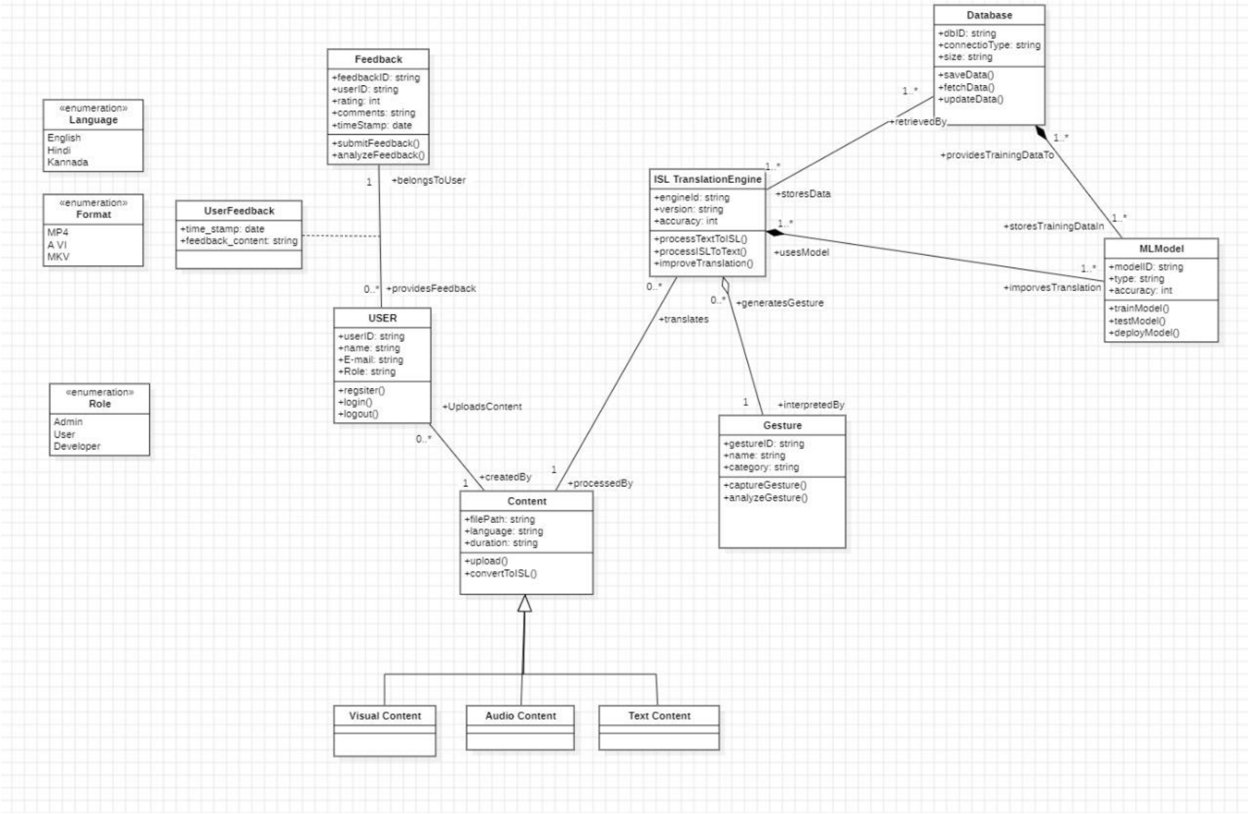
- Project duration: 6-8 months.
- Key phases:
    - Requirements Gathering: 1 month
    - Design & Architecture: 1.5 months
    - Development: 3 months

    - 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

Chapter 3: Class Modeling



Class Explanations:

1. **USER:**

Represents the users of the ISL translation system. A user can upload content, provide feedback on generated gestures, and interact with the system. Attributes such as user roles (e.g., admin, regular user) can be derived from the Role enumeration.

2. **Content:**

Represents the text, audio, or video uploaded by a user to be translated into ISL gestures. It includes properties such as content type or format, derived from the Format enumeration. Each piece of content is associated with exactly one user but can be processed by the ISL Translation Engine.

3. **Feedback:**

Captures user feedback on the generated gestures. Each feedback entry belongs to exactly one user and may include ratings or comments about the accuracy or usability of the gestures.

4. **ISL Translation Engine:**

The core system component responsible for processing content and generating ISL gestures using ML models. It interacts with the Database for fetching rules and mappings and uses the ML Model for accurate translations.

5. **Gesture:**

Represents the ISL gestures generated by the system based on the uploaded content. Each gesture is tied to the ISL Translation Engine that generated it and is stored in the database for future reference or analysis.

6. **MLModel:**

The machine learning model used by the ISL Translation Engine to generate ISL gestures. The model is trained using data stored in the Database and continually updated to improve the accuracy of translations.

7. **Database:**

Serves as the storage system for the ISL system, housing content, ISL mappings, generated gestures, user feedback, and ML training data. It facilitates data retrieval and storage for multiple system components.

8. **UserFeedback:**

Represents detailed feedback entries that can be aggregated under a single Feedback object. These entries might include specific data points such as timestamps, ratings, or suggestions for improvement.

Enumerations:

1. **Language:**

Defines the language(s) supported by the system for content translation (e.g., English, Hindi). This enumeration is used as an attribute in the Content class.

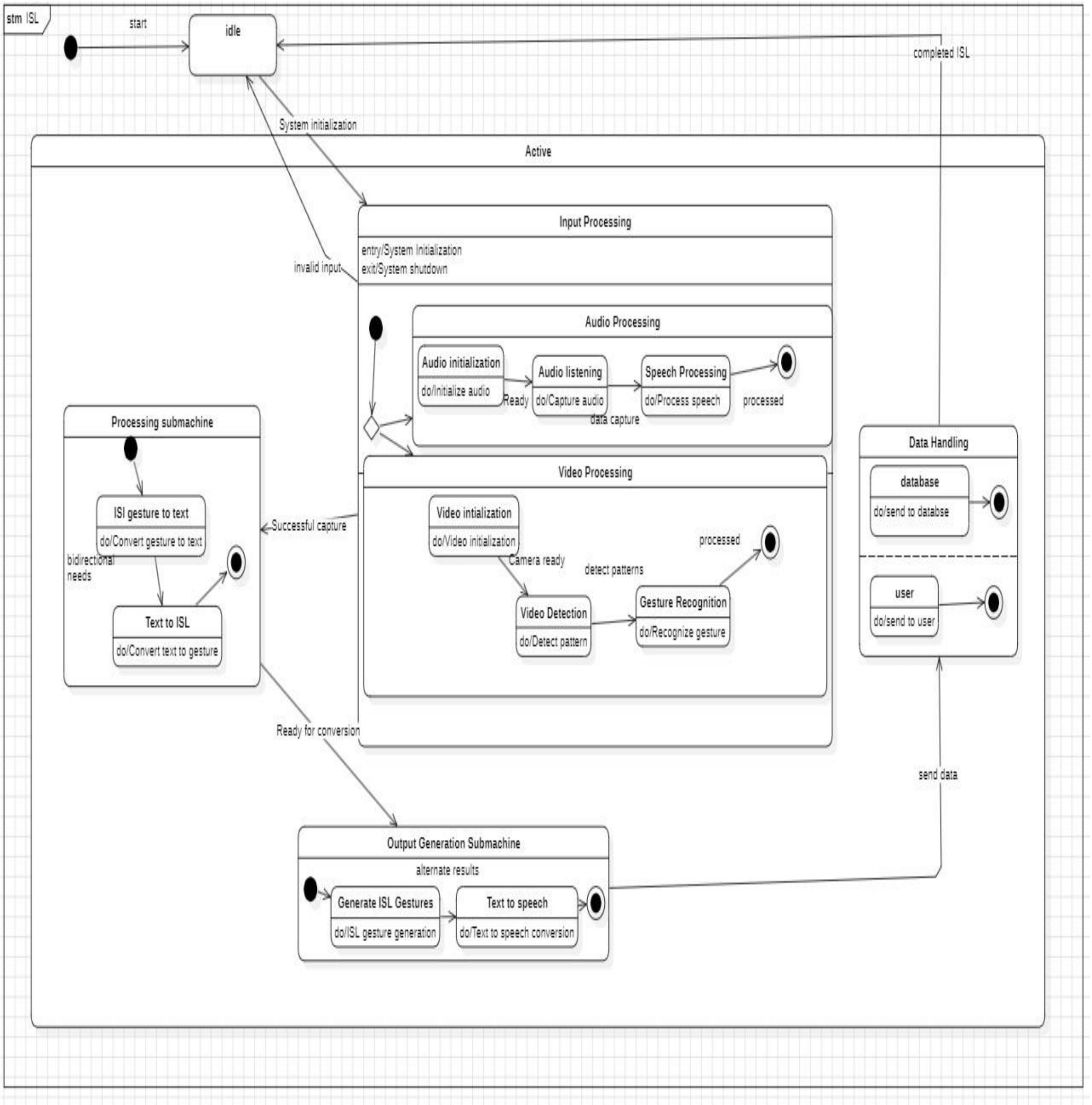
2. **Format:**

Specifies the type or format of the content, such as text, audio, or video. This enumeration is applied to the Content class to differentiate the processing requirements.

3. **Role:**

Represents the different roles a user can have within the system (e.g., admin, regular user). This enumeration is used as an attribute in the USER class to control access and permissions.

Chapter 4: State Modeling



The state diagram you provided outlines the behavior of an application designed to convert **audio, video, or text input into Indian Sign Language (ISL) gestures** and perform the reverse operation. Here’s an explanation of the diagram's key components and the relevance of each state and event:

1. Idle State

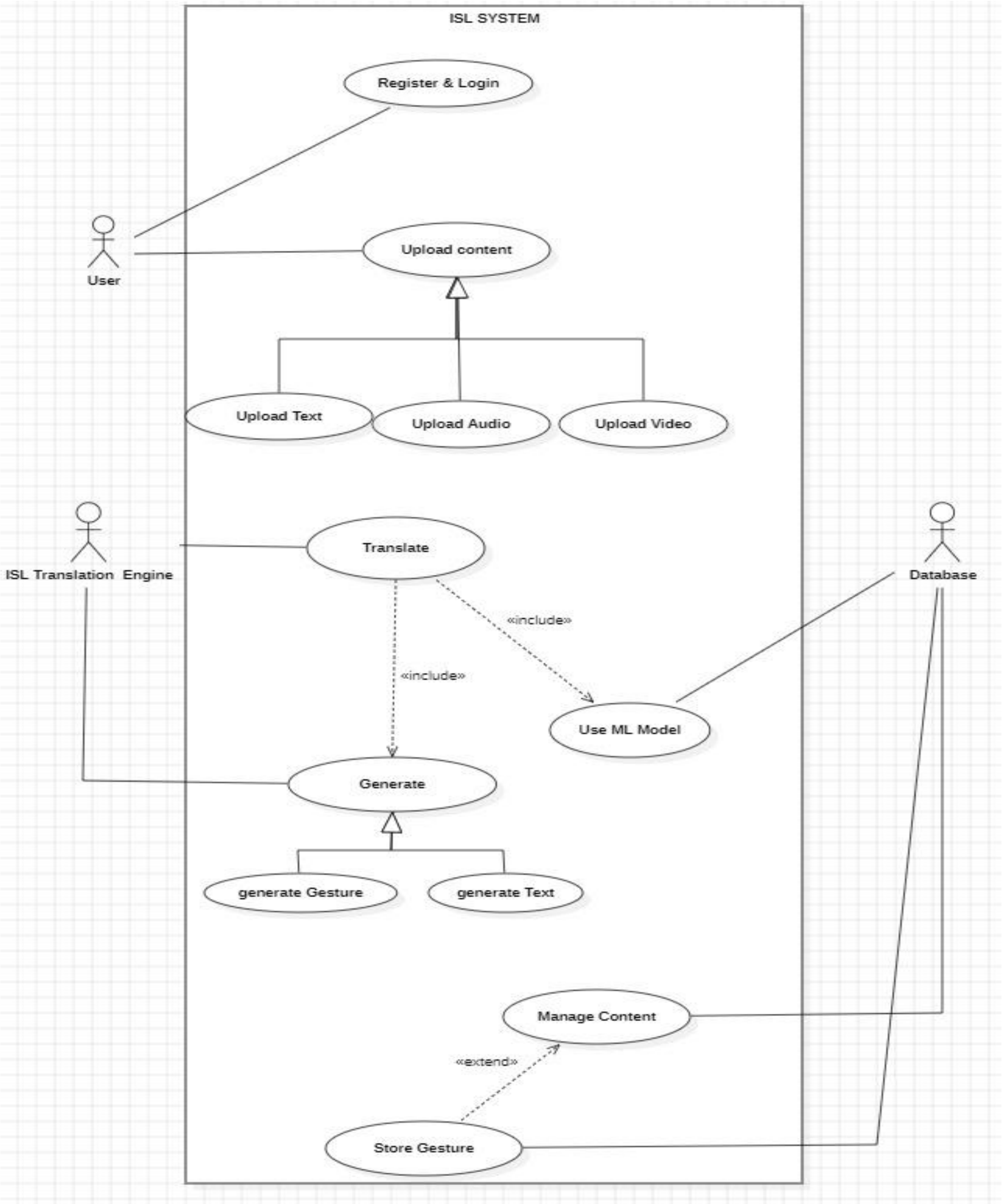
- **Purpose**: This is the default or starting state where the system waits for initialization.
- **Event**: start transitions the system from the idle state to the Active state.
- **Relevance**: It ensures the system remains inactive until explicitly started.

2. Active State

- **Purpose**: Represents the main functional state of the application where all processing takes place.
- **Entry Activity**: entry/System Initialization. The system sets up necessary configurations, such as initializing input modules (audio, video, or text).
- **Exit Activity**: exit/System Shutdown. Ensures a proper cleanup of resources when transitioning out of this state.
- **Relevance**: It acts as a container for all core sub-states related to input processing, output generation, and data handling.

Chapter 5: Interaction Modeling

Use Case Diagram



Usecase Summary:-

Actors:

1. User: Interacts with the system to register, log in, upload content, and manage content.

2. ISL Translation Engine: Translates uploaded content into gestures or text using machine learning.

3. Database: Stores and retrieves content and generated gestures.

Use Cases:

1. Register & Login: Ensures authenticated access to the system.

2. Upload Content: Users can upload text, audio, or video files as input for translation.

3. Translate: Converts uploaded content into sign language. Includes the use of an ML Model for processing the translation.

4. Generate: Outputs the translation as either gestures or text.
   Sub-use cases:
   Generate Gesture
   Generate Text

5. Store Gesture: Saves the generated gestures for future use.

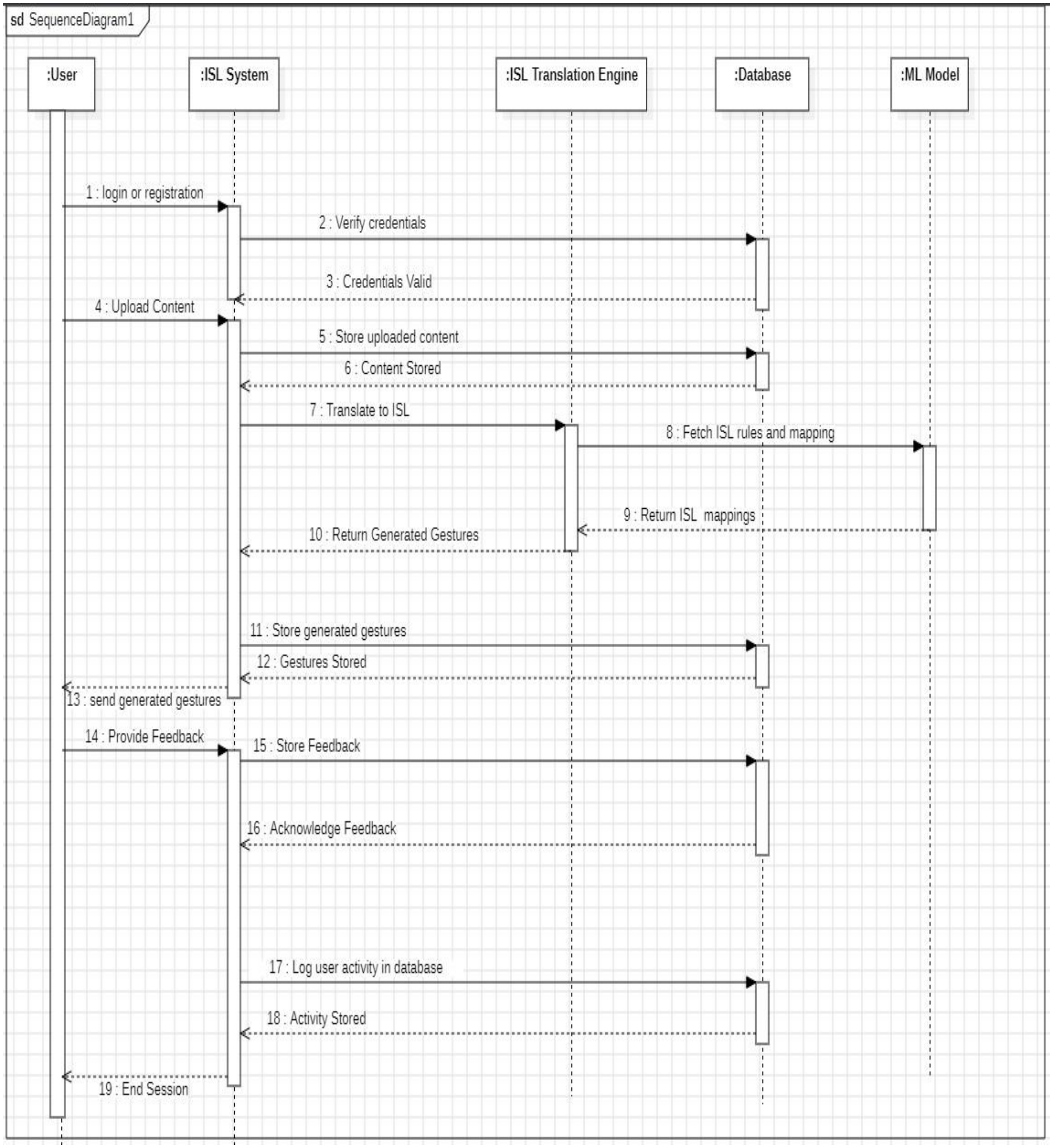
6. Manage Content: Extends functionality for organizing and updating stored content.

Relationships:

1. Include Relationships:
2. Translate includes the functionality of Use ML Model.

3. Extend Relationships:
4. Manage Content extends the Store Gesture functionality.

Sequence Diagram



Explanation of the Sequence Diagram:

User Login or Registration:

The process begins with the User initiating a login or registration request to the ISL System.The ISL System sends a message to Verify credentials. If credentials are valid, the ISL System confirms with a message: Credentials Valid.

User Access and Content Upload:

After successful login or registration, the User is logged in.
The User uploads content (text, audio, or video) to the ISL System for ISL translation.
The ISL System stores the uploaded content in the Database, confirming the operation with Content Stored.

Translation Process:

The ISL System sends the content to the ISL Translation Engine for processing.
The ISL Translation Engine fetches ISL rules and mapping data from the Database using Fetch ISL rules and mapping.
The Database returns the necessary ISL mappings to the ISL Translation Engine.

Generating Gestures:

The ISL Translation Engine uses the fetched rules and mappings, along with the ML Model, to translate the content into ISL gestures.
Once the gestures are generated, the Translation Engine sends the generated gestures back to the ISL System with the message: Return Generated Gestures.

Storing and Sending Gestures:

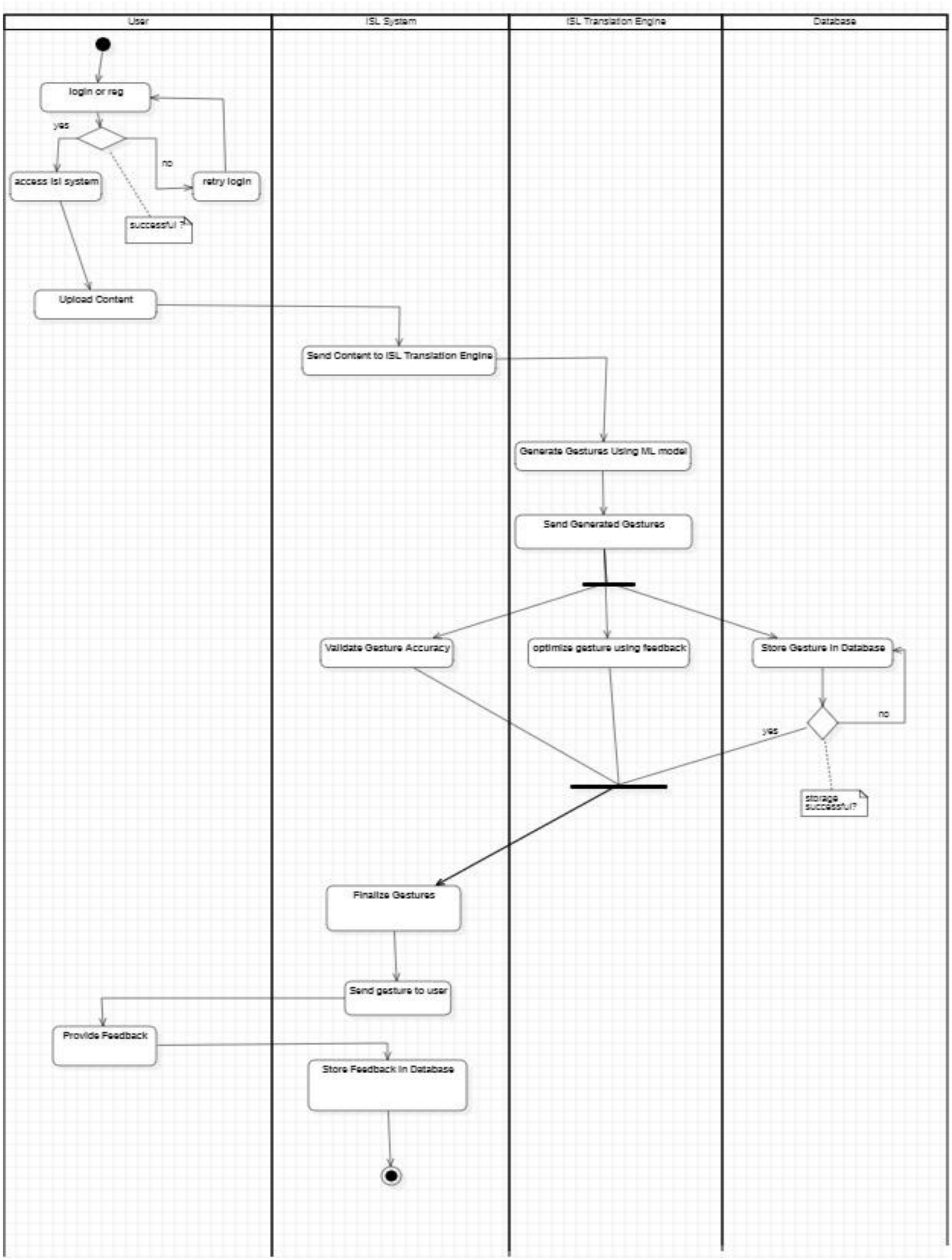
The ISL System stores the generated gestures in the Database and confirms with Gestures Stored.
The system sends the finalized gestures to the User with the message: Send generated gestures.

Feedback Handling:

The User provides feedback on the generated gestures to the ISL System.

The feedback is stored in the Database with the message: Store Feedback, which is acknowledged by Acknowledge Feedback.

Activity Diagram



Explanation of the Activity Diagram

Flow of Control

1. User Interaction:

- The process begins in the User swimlane. The user either logs in or registers in the system.
- If the login attempt fails, control loops back, prompting the user to retry until successful.
- Once logged in, the user uploads content, which could be text, audio, or video, for translation into ISL gestures. Control then moves to the ISL System swimlane.

2. Content Processing in the ISL System:

- The uploaded content transitions to the ISL Translation Engine swimlane for further processing.
- The ISL Translation Engine uses a Machine Learning Model to analyze the content and generate ISL gestures. These steps occur within the Translation Engine swimlane, interacting with the ML Model swimlane for gesture generation.

3. Gesture Generation and Transmission:

- Once gestures are generated, they are sent back to the ISL System swimlane.
- The system initiates validation and optimization processes to ensure that the generated gestures are of high quality.

4. Validation and Optimization:

- Validation: The system compares the generated gestures against predefined benchmarks or standards to assess their accuracy. If the gestures pass validation, they move forward. If they fail, control splits, and the system engages in iterative improvement to refine the gestures.
- Optimization: Failed gestures undergo further refinement, often incorporating user feedback or additional processing until they meet the required accuracy standards.

5. Decision Points:

- The system evaluates whether the gestures meet the validation criteria. This is a decision node in the activity diagram:
  - If gestures pass validation, they are stored in the database directly.
  - If they fail, the iterative refinement process continues.

6. Gesture Finalization and Feedback Collection:

- Once validated and optimized, the finalized gestures are sent to the User swimlane.
- The user evaluates these gestures and provides feedback. Control returns to the ISL System swimlane, where the feedback is stored in the database for further analysis.

7. End Process:
   - The workflow ends after feedback is stored in the database. This feedback loop allows for continuous improvement of the machine learning model and gesture generation process, creating a cycle of refinement and user-centered optimization.

Swimlanes

1. User: Represents all user-related activities, such as logging in, uploading content, and providing feedback.
2. ISL System: Covers system-level activities, such as handling user inputs, managing content, and validating gestures.
3. ISL Translation Engine: Handles the processing of uploaded content and gesture generation.
4. ML Model: Provides the computational framework and algorithms used for gesture generation.
5. Database: Stores content, generated gestures, user feedback, and system logs for analysis and improvement.

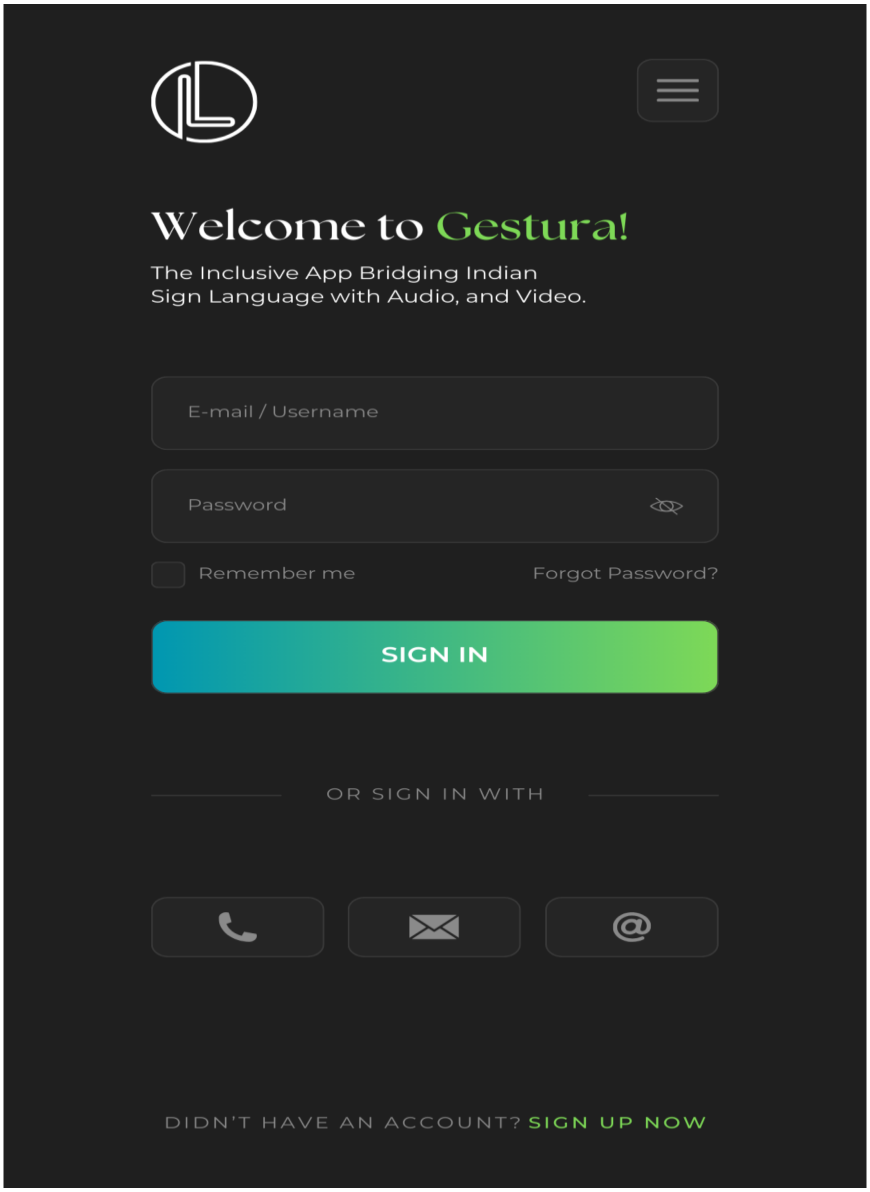
Splitting and Merging of Control

1. Splitting of Control:
   - Occurs after gesture generation, where validation determines if gestures meet the quality criteria.
   - If the gestures pass validation, control splits toward database storage. If not, it moves toward refinement and optimization.

2. Merging of Control:
   - After refinement or validation, all pathways converge to a single point where gestures are either finalized or sent to the user for evaluation.
   - Feedback collection merges the user's evaluation process back into the ISL System for feedback storage.

Chapter 6: UI Design with Screenshots

Sign-in Page:-

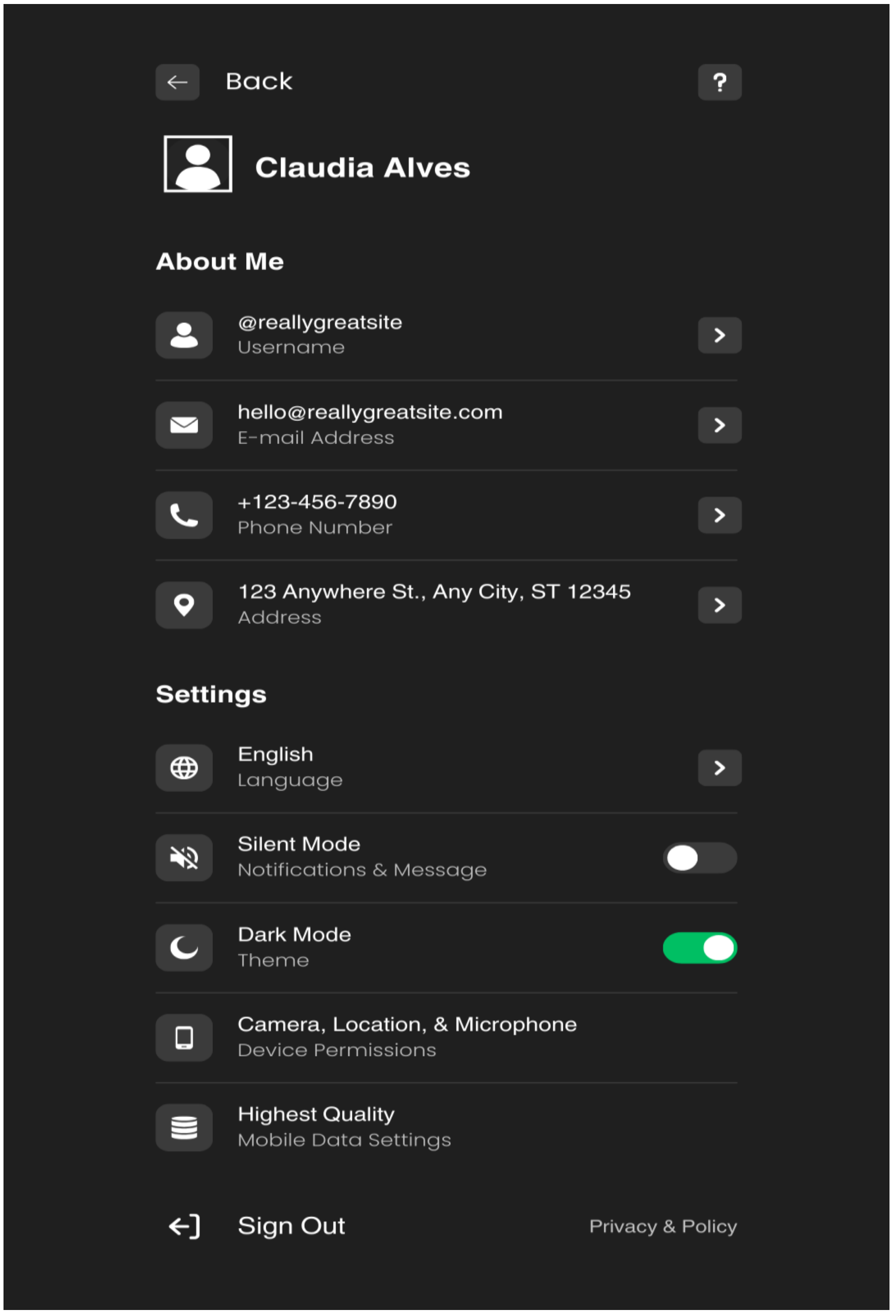


Welcome to Gestura!

The Inclusive App Bridging Indian Sign Language with Audio, and Video.

The Sign-in page allows user to enter their credentials and log in into their account. This page is necessary for user authentication as well as facilitating account creation that helps in storing the history of usage of the user.

Menu Page :-



Back ?

Claudia Alves

About Me

@reallygreatsite
Username

hello@reallygreatsite.com
E-mail Address

+123-456-7890
Phone Number

123 Anywhere St., Any City, ST 12345
Address

Settings

English
Language

Silent Mode
Notifications & Message

Dark Mode
Theme

Camera, Location, & Microphone
Device Permissions

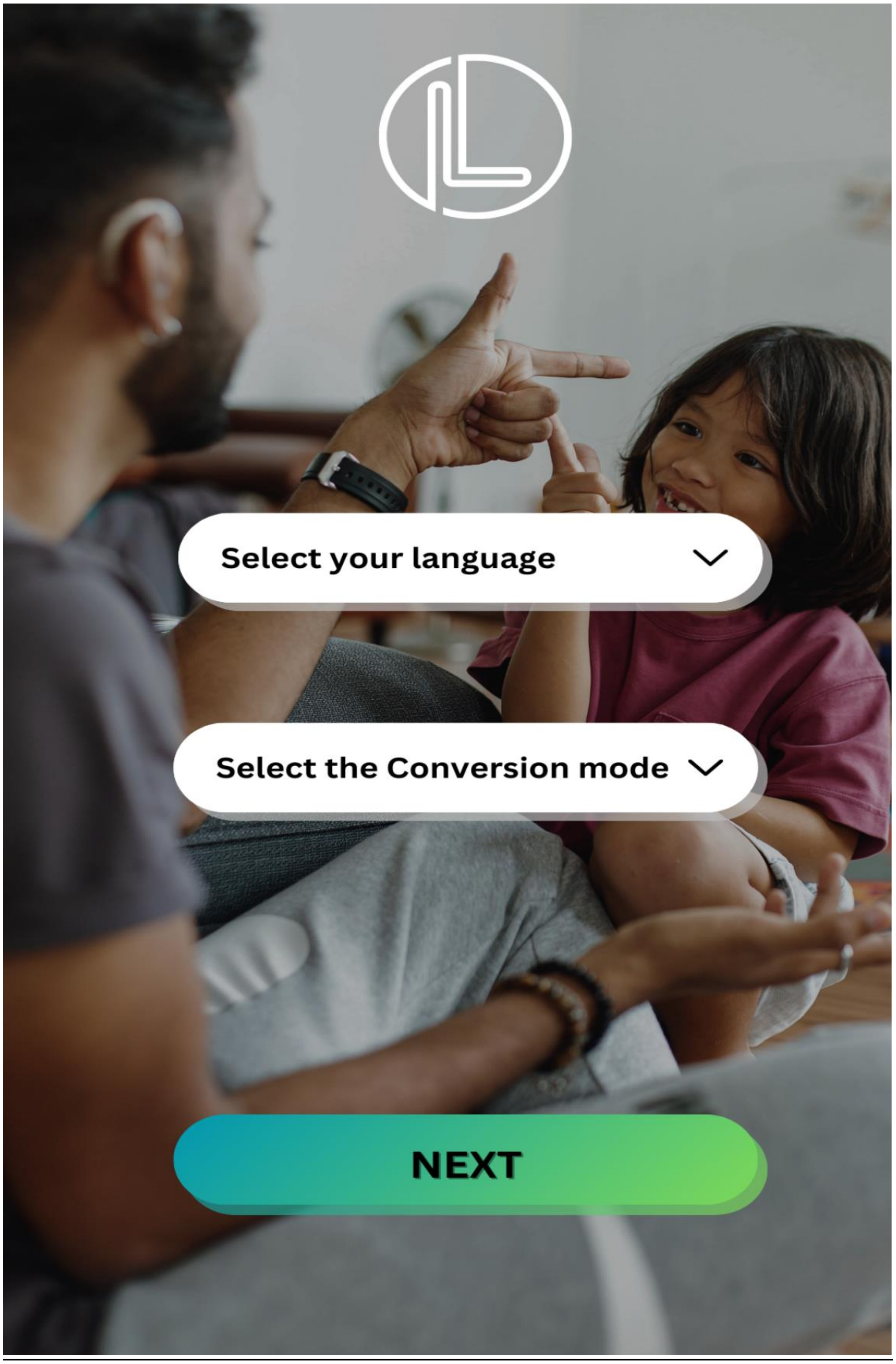
Highest Quality
Mobile Data Settings

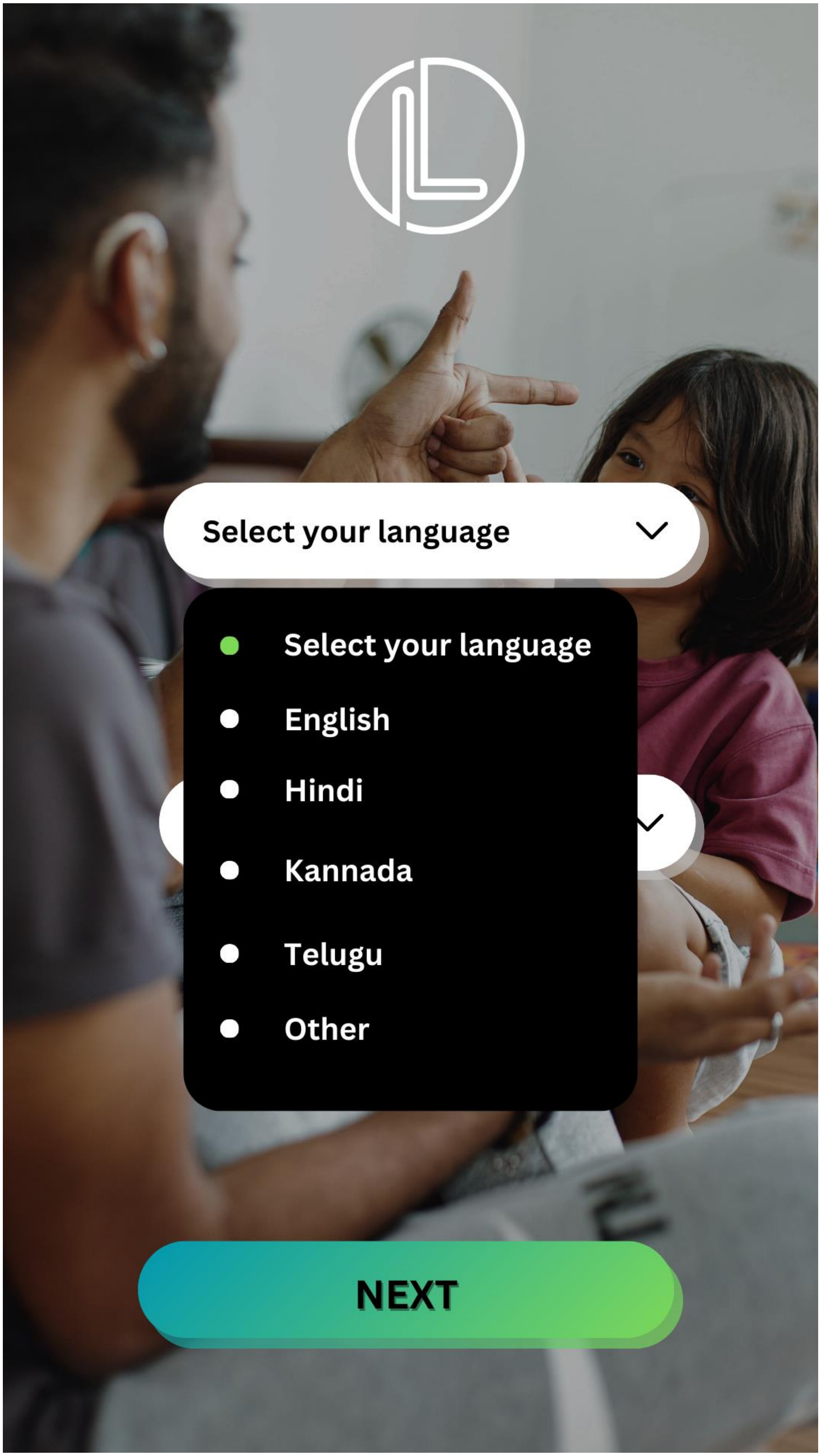
Sign Out

Privacy & Policy

The Menu page page allows user to customize their accounts, manage their credentials as well as make changes to the settings of the application as per their needs.

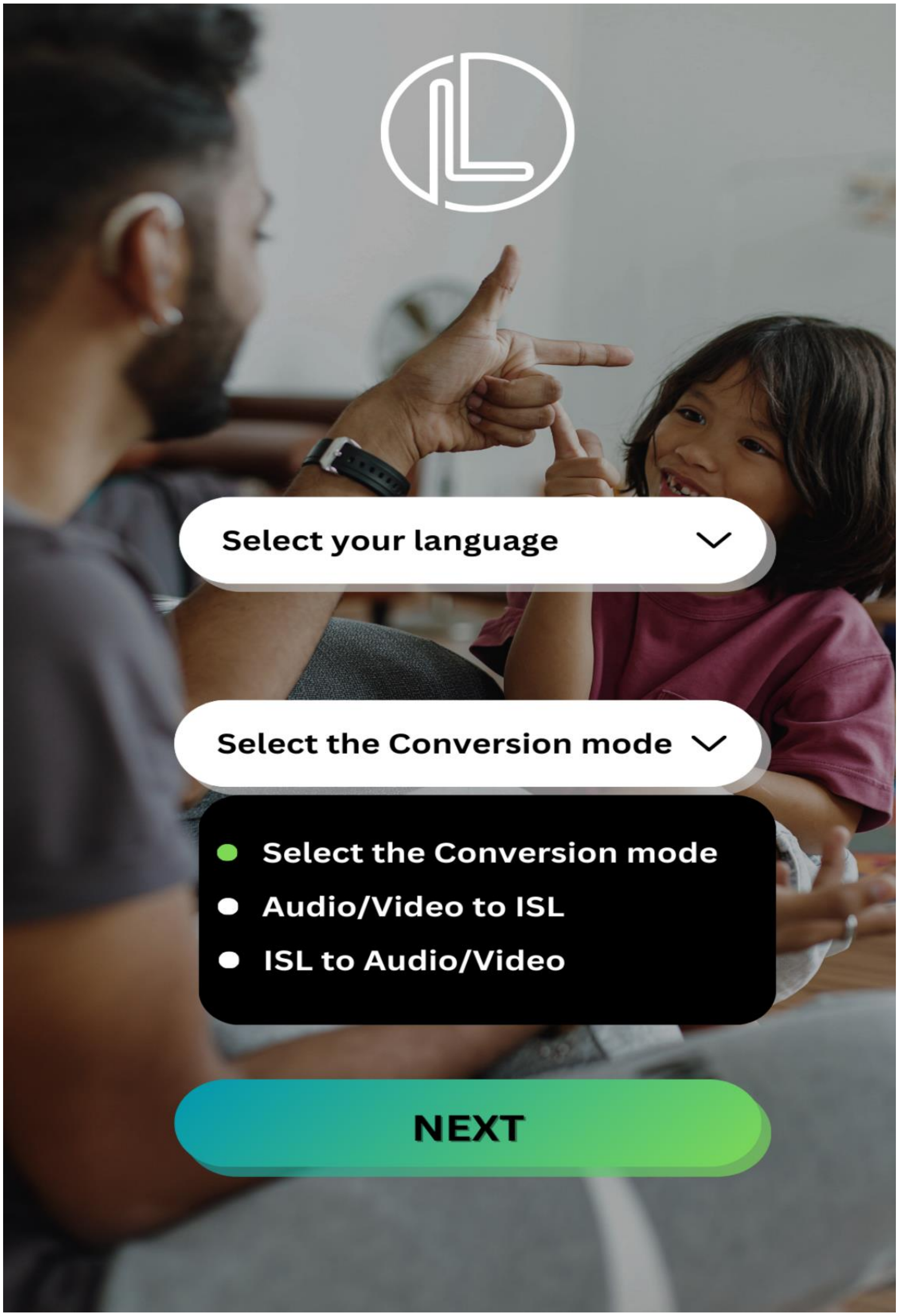
Home Page :-


Select your language
Select the Conversion mode
NEXT



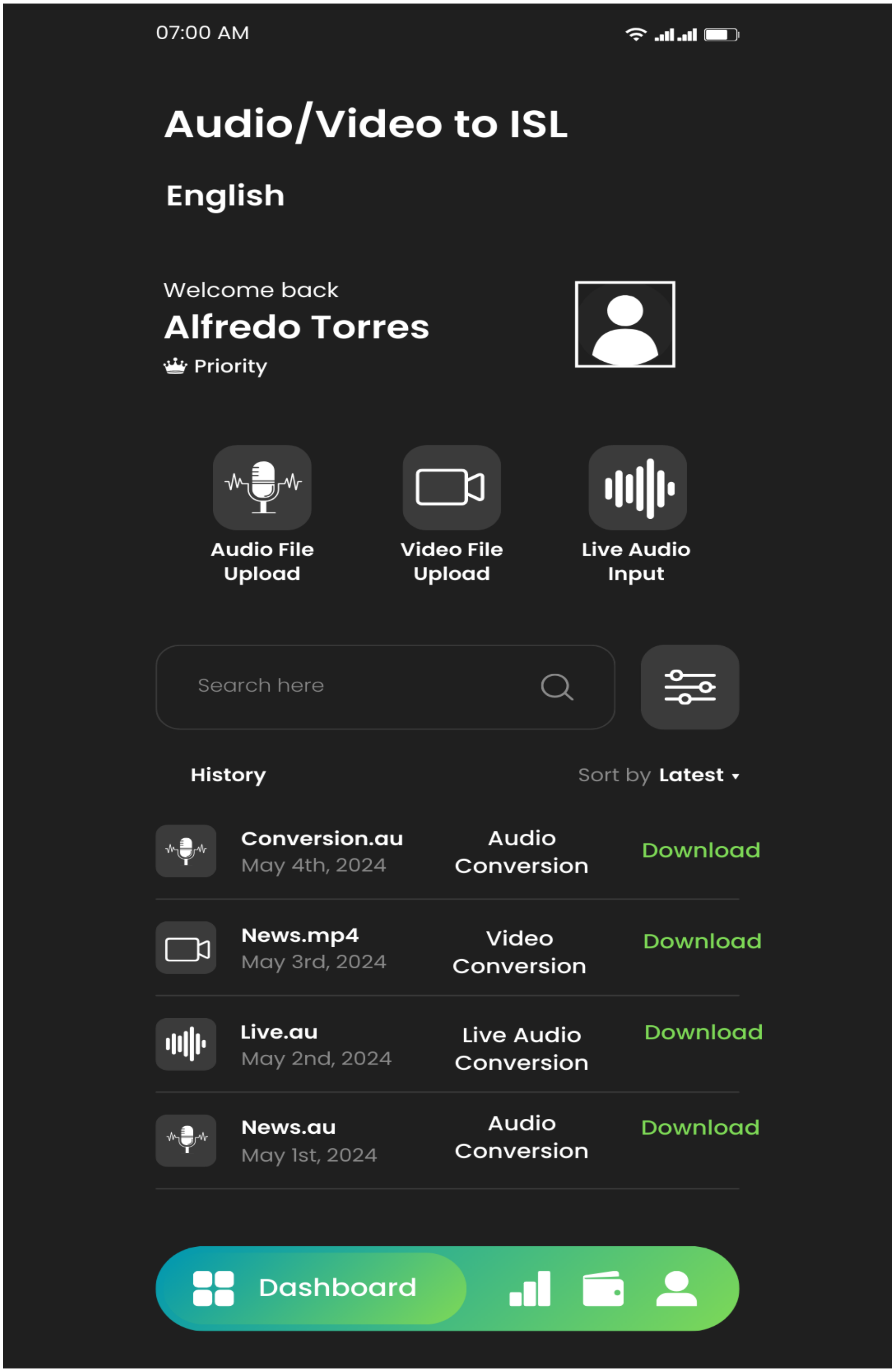
Select your language

- ● Select your language
- ● English
- ● Hindi
- ● Kannada
- ● Telugu
- ● Other



The Home page allows user to set the type of conversion the would like to make, whether from audio/video file to ISL or otherwise. User can also set the language parameter.

Dashboard :-



07:00 AM

Audio/Video to ISL

English

Welcome back

Alfredo Torres

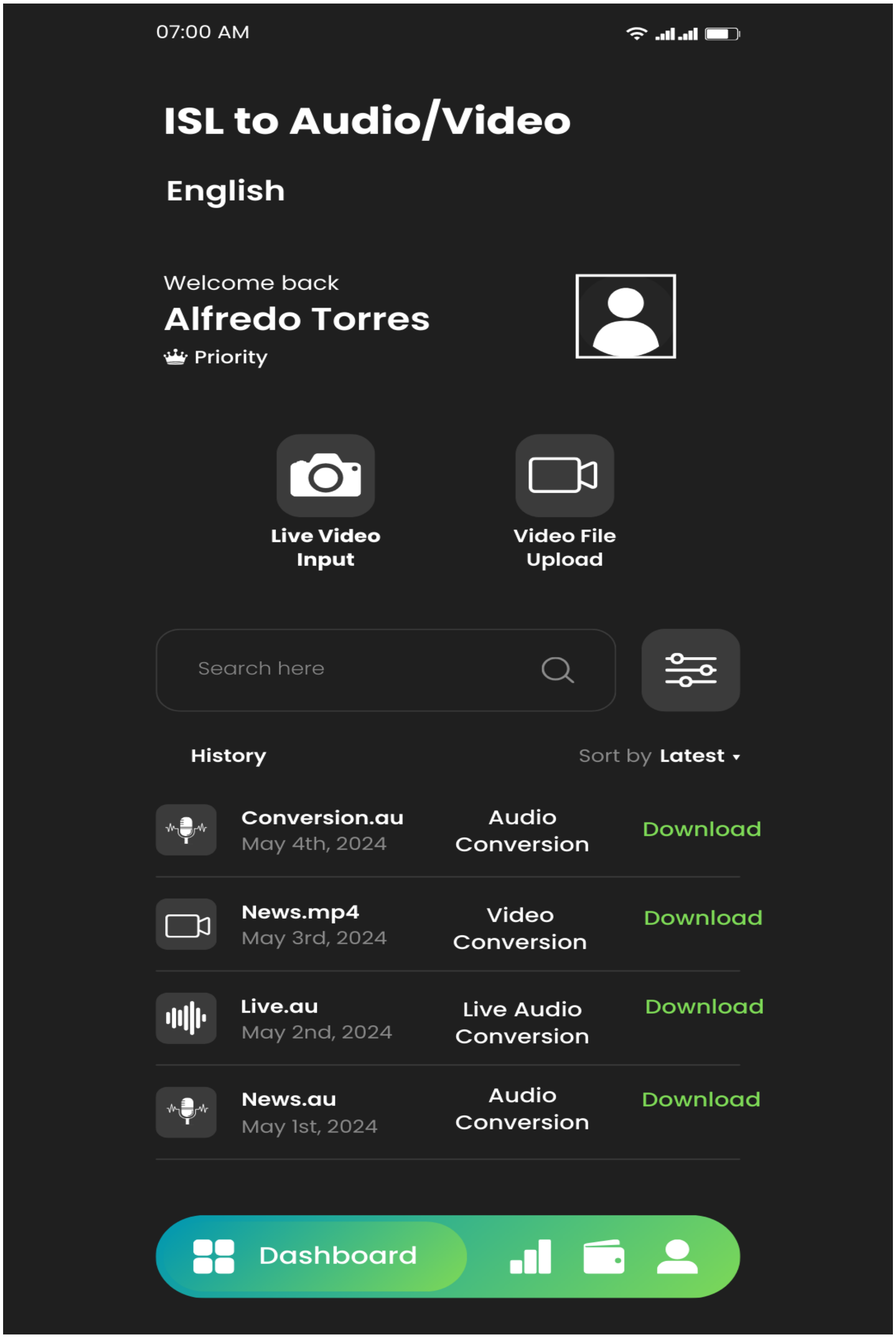
♛ Priority

Search here

History

Sort by **Latest** ▾

🎙	**Conversion.au** May 4th, 2024	Audio Conversion	Download
▭	**News.mp4** May 3rd, 2024	Video Conversion	Download
ılıılı	**Live.au** May 2nd, 2024	Live Audio Conversion	Download
🎙	**News.au** May 1st, 2024	Audio Conversion	Download



ISL to Audio/Video

English

Welcome back

Alfredo Torres

Priority

History

Sort by Latest

Conversion.au May 4th, 2024	Audio Conversion	Download
News.mp4 May 3rd, 2024	Video Conversion	Download
Live.au May 2nd, 2024	Live Audio Conversion	Download
News.au May 1st, 2024	Audio Conversion	Download

The Dashboard is where user can upload the content, based on the type of conversion they chose in the home page. Dashboard also contains the history of conversions that the user has done to allow user to revisit them easily when needed.