CHAPTER 5

SYSTEM REQUIREMENTS

- User Characteristics
- Functional Requirement
- Non-Functional Requirement
- Hardware and Software Requirement

5.1 USER CHARACTERISTICS

When designing a Sign Language to Text Conversion system, it is crucial to consider the characteristics of the target audience. The success of the system will largely depend on how well it meets the needs of its users, particularly individuals who rely on sign language for communication.

The primary users of the Sign Language to Text Conversion system are individuals who are deaf or hard of hearing and use sign language as their primary mode of communication. These users need a solution that allows them to communicate easily with individuals who do not understand sign language. The system should be user-friendly and accessible, catering to a wide range of sign language users with varying levels of technical literacy.

In addition to direct users, the system may also be used by hearing individuals who need to communicate with those who use sign language, such as teachers, healthcare providers, and service personnel. These users are looking for an efficient way to understand sign language gestures and improve communication in educational, healthcare, or service environments.

The system may also appeal to researchers and developers in the fields of linguistics, accessibility, and assistive technology, as they seek tools for studying or improving communication for the deaf and hard-of-hearing community. For these users, accuracy and the ability to handle complex gestures in a variety of sign languages are key factors.

The users of this system value real-time translation, accuracy, and efficiency. Many may be busy individuals who need a solution that works reliably in everyday interactions, whether in personal, educational, or professional settings. For tourists or visitors in new regions, the system could also assist with communication across different sign languages.

It’s essential to consider that users may have different expectations and needs based on their demographics, location, and cultural background. Similarly, users may expect different levels of accuracy and real-time responsiveness based on the context in which they use the system (e.g., casual vs. professional settings).

In summary, the primary users of the Sign Language to Text Conversion system are individuals who rely on sign language and those who interact with them. These users value convenience, real-time translation, accuracy, and accessibility. Therefore, it is critical to consider the diverse user characteristics during the design and development phases to ensure the system effectively meets the needs and preferences of its target audience.

5.2 FUNCTIONAL REQUIREMENTS

The functional requirements for the Sign Language to Text conversion project define the system's essential behaviors, actions, and functions that it must perform to meet its goals. These requirements focus on how the system interacts with users and other systems to produce the desired outcome (i.e., converting sign language gestures into text). Here's a breakdown of the key functional requirements:

1. **Gesture Recognition**
   - Capture Hand Movements: The system must be able to capture hand gestures made by the user in real-time using a camera or motion-sensing technology.
   - Identify and Classify Gestures: The system must recognize specific hand gestures (i.e., signs) and classify them into corresponding sign language symbols. It should support a predefined set of sign language gestures.

2. **Text Conversion**
   - Map Gestures to Text: Once a gesture is recognized, the system must convert it into the corresponding text in the target language (e.g., English).
   - Display the Text: The converted text should be displayed in a text box or output area that the user can easily view and read.
   - Real-Time Text Output: The system should provide real-time conversion and display of text as the gestures are performed, without significant delays.

3. **System Integration**
   - Camera or Sensor Integration: The system should be able to access and use the device’s camera or an external sensor for gesture recognition.

4. **Performance Requirements**
   - Real-Time Processing: The system must process gestures and convert them to text within an acceptable response time (e.g., less than 1 second delay).
   - Low Latency: Gesture recognition and conversion should occur with minimal delay to ensure a fluid user experience.

5. **Security and Privacy**
    - Access Control: If the system supports multiple users, there should be proper authentication and authorization mechanisms to ensure data privacy for individual users.
    - Legal Compliance: The system must comply with any relevant local or international standards, such as data protection laws (e.g., GDPR) to ensure that user data is handled responsibly and ethically.
    - Industry Standards: The system should follow industry best practices for machine learning, AI ethics, and software development, especially in sensitive areas such as accessibility.

5.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements (NFRs) describe the system's performance characteristics, usability, security, and other aspects that are not related to specific functionalities but are critical for overall user satisfaction and system reliability. Here are the key non-functional requirements for the Sign Language to Text conversion project:

1. **Scalability**
   - Handling Increased Users: The system must scale to accommodate an increasing number of users if deployed on a server or cloud platform, without a decline in performance.

2. **Availability**
   - System Uptime: The system should have an availability of 99.9%, ensuring minimal downtime. If used in critical applications, such as for communication for deaf users, uptime is crucial.
   - Failure Recovery: In case of system failures, it must recover gracefully, allowing users to resume their activities without data loss or re-input of previous gestures.

3. **Usability**
   - Ease of Use: The user interface must be simple and intuitive, allowing users with no technical background to interact with the system efficiently.
   - Learnability: The system should be easy to learn, with minimal training or instructions required to understand how to start gesture recognition and view converted text.
   - Accessibility: The interface should follow accessibility guidelines (e.g., WCAG) to ensure that it’s usable by people with various disabilities. This may include support for screen readers, high-contrast modes, and large font sizes.

4. **Reliability**
    - Accuracy of Recognition: The system should maintain a gesture recognition accuracy rate of at least 95%, ensuring that most recognized gestures are correctly converted to text.
    - Robustness: The system must remain stable under various conditions, such as changes in lighting for gesture capture or variations in gesture execution by different users.
    - Minimal Dependencies: The system should have minimal external dependencies, making it easier to install and run on different environments without requiring extensive configuration.

5. **Maintainability**
    - Modular Code Structure: The system must be developed in a modular manner to facilitate easy updates, bug fixes, and the addition of new features without impacting existing functionality.
    - Documentation: Comprehensive documentation should be provided for developers and users to assist with system maintenance, updates, and troubleshooting.
    - Testability: The system should be designed to support automated testing, allowing for regular and reliable tests to be conducted during the development and deployment phases.

6. **Extensibility**
    - Addition of New Features: The system architecture should allow for the addition of new features, such as support for more languages, without significant refactoring.
    - Plug-and-Play Models: New machine learning models or gesture libraries should be easily integrable into the system, allowing updates in recognition algorithms without major system changes.

7. **Interoperability**
    - API Availability: Provide an API (Application Programming Interface) that allows other applications to interact with the system, enabling the retrieval of gesture recognition data or text output.

5.4 HARDWARE AND SOFTWARE REQUIREMENT

The hardware requirements for the Sign Language to Text conversion project depend on the system's complexity, the type of devices used for gesture capture, and the processing power needed for real-time gesture recognition. Below are the typical hardware components needed for both the development and deployment environments:

5.4.1 Hardware Requirements

1. **Camera or Motion Sensor**
   - High-Resolution Camera: A camera with a minimum resolution of 720p (preferably 1080p or higher) to accurately capture hand gestures in real-time. The camera should have good frame rates (at least 30 FPS) to handle dynamic gestures.
   - Example: Integrated laptop/desktop camera, external USB webcams,

2. **Processing Unit (CPU)**
   - Processor: Intel Core i5 (or AMD equivalent) or higher.
   - Clock Speed: 2.5 GHz or higher.
   - The CPU should be capable of handling real-time video processing and machine learning tasks related to gesture recognition without lag.

3. **Graphics Processing Unit (GPU)**
   - If the project uses advanced machine learning models or neural networks for gesture recognition, a dedicated GPU will significantly improve performance.
   - GPU: Nvidia GTX 1050 or equivalent.

4. **Memory (RAM)**
   - 8 GB of RAM.

5. **Storage**
   - Storage: 256 GB SSD (Solid State Drive).
   - SSDs are recommended for faster data access, especially when handling large datasets for machine learning, image processing, or saving gesture recordings for analysis.

6. **Power Supply**
    - For Desktop Systems: A stable power supply unit (PSU) rated for at least 500W to support the CPU, GPU, and other peripherals.

7. **Network and Connectivity**
    - Internet Access: Required for downloading model updates, libraries, and deploying cloud-based features, if applicable.
    - Wi-Fi/Bluetooth: If the system integrates with other devices (e.g., external sensors), ensure the machine has stable Wi-Fi or Bluetooth connectivity.

5.4.2 Software Requirements

1. **Operating System**
    - Windows 10/11 or higher, macOS 10.14 (Mojave) or higher, or Linux (Ubuntu
    - 18.04 LTS or higher).

2. **Development Tools**
    - Visual Studio Code (for cross-platform development and web apps).

3. **Programming Languages**
    - Python: Widely used for machine learning, gesture recognition, and image processing.
    - Libraries such as OpenCV, TensorFlow, and PyTorch work well in Python.

4. **Libraries and Frameworks**
    - Gesture Recognition and Image Processing:
    - OpenCV (Open Source Computer Vision Library): Essential for capturing and processing images or video streams from the camera.
    - MediaPipe: A powerful library from Google for real-time hand and body gesture tracking.

5. **Machine Learning Frameworks:**
    - TensorFlow or Keras: For training and deploying gesture recognition models.
    - Scikit-learn: For data preprocessing, classification, and model evaluation (if using traditional machine learning techniques).

6. **Machine Learning Development Tools**
    - Google Colab: A cloud-based notebook environment that provides free GPU and TPU access for training machine learning models.