

**SIGN LANGUAGE INTERPRETER APP**

•

**By**

**COLLINS KEFA - 664965**

A Final Project submitted to the School of Science and Technology in Partial Fulfilment of the requirements for the degree of Bachelor of Science in Applied Computer Technology.

**(FALLSEMESTER 2024)** **USIU-AFRICA)**

# ACKNOWLEDGEMENTS

So far, I would like to express my sincere appreciation to my project supervisor, **Dr. Paul Okanda,** for his invaluable guidance, feedback, and unwavering support in shaping this project.

Furthermore I would also like to acknowledge my friends who listened to me whine about my life constantly and for supporting me throughout my project.

# DECLARATION OF ORIGINAL WORK

"I attest that all of the content in this report is original to me and does not include any unattributed or unreferenced material. I also guarantee that the aforementioned claim holds true for the execution of the project and all related paperwork. With respect to the electronically submitted version of this work, I agree to its electronic copying and storage for evaluation purposes, including the use of plagiarism detection software by the Department to guarantee the accuracy of the work that is evaluated. I consent to having my name clearly listed as the author and for my dissertation to be made available to the public."

Sign………………………….. Date:……………………………..

# ABSTRACT

The project aims to develop a comprehensive and user-friendly web-based application designed to bridge the communication gap between sign language users and non-signers. Leveraging cutting-edge technologies like computer vision and machine learning, the system will accurately interpret hand gestures and convert them into spoken language in real time. By employing advanced gesture recognition algorithms, the app will ensure precise and natural speech output, facilitating smoother conversations for users who rely on sign language as their primary mode of communication.

This project not only focuses on enhancing accessibility for the hearing and speech-impaired but also strives to foster inclusivity by enabling seamless interaction between signers and non-signers. The system's core objective is to provide real-time, reliable, and accessible communication for sign language users, regardless of their surroundings, be it in personal conversations or professional settings. Additionally, by incorporating machine learning, the system can continually improve its accuracy and adaptability, offering a personalized and intuitive experience for each user over time.

With its potential applications ranging from everyday conversations to educational and professional environments, this project aims to make a significant impact on the lives of millions of sign language users, empowering them with a tool that reduces communication barriers and promotes inclusivity.

In addition to its core functionality, the system will also offer customizable features such as language preferences, voice modulation, and speed of speech output to cater to individual user needs. This flexibility will allow users to tailor the experience to their specific communication styles, ensuring a more natural and comfortable interaction. By integrating user feedback and continuous learning, the app will evolve to provide an even more accurate and responsive platform for effective communication between signers and non-signers.

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENTS i](#_Toc182566011)

[DECLARATION OF ORIGINAL WORK ii](#_Toc182566012)

[ABSTRACT iii](#_Toc182566013)

[1.INTRODUCTION 1](#_Toc182566014)

[2. LITERATURE REVIEW 4](#_Toc182566015)

[2.1 Existing Systems 4](#_Toc182566016)

[2.1.2 Sign All 5](#_Toc182566017)

[2.1.3 Kenyan Sign Language (KSL) Initiatives 7](#_Toc182566018)

[2.2 Strengths and Weaknesses 8](#_Toc182566019)

[3. Aims and Objectives 9](#_Toc182566020)

[3.1 General Aims and Objectives 9](#_Toc182566021)

[3.2 Specific Aims and Objectives 10](#_Toc182566022)

[4. PROPOSED PROJECT 11](#_Toc182566023)

[4.1 Phases of the project 11](#_Toc182566024)

[4.2 Program of work 13](#_Toc182566025)

[4.4 Requirements 16](#_Toc182566026)

[5. SYSTEM ANALYSIS AND DESIGN 17](#_Toc182566027)

[5.1 Introduction 17](#_Toc182566028)

[5.2 System Models 17](#_Toc182566029)

[UML Class Diagram 17](#_Toc182566030)

[UML Use Case Diagram 18](#_Toc182566031)

[18](#_Toc182566032)

[Entity Relationship Diagram 19](#_Toc182566033)

[Class Diagram 20](#_Toc182566034)

[Network Diagram 21](#_Toc182566035)

[Dataflow diagrams 22](#_Toc182566036)

[System Architecture 23](#_Toc182566037)

[Keywords (essential concepts) 24](#_Toc182566038)

**LIST OF TABLES & FIGURES**

Figure 1- Screenshot of Springer Link home page 9

[Figure 2- Real time translation illustration (source: www.nature.com)](about:blank) 10

[Figure 3- Gantt chart 21](#_heading=h.vx1227)

[Figure 4- UML Class Diagram](about:blank) 24

[Figure 5- UML Use Case Diagram](about:blank) 25

[Figure 6- Entity Relationship Diagram 26](#_heading=h.111kx3o)

[Figure 7- Class Diagram 27](#_heading=h.206ipza)

[Figure 8- Network Diagram 29](#_heading=h.2zbgiuw)

[Figure 9- Lvl 0Data Flow Diagram 30](#_heading=h.3ygebqi)

[Figure 10- LVL 1 DFD 30](#_heading=h.2dlolyb)

[Figure 11- LVL 2 DFD 31](#_heading=h.sqyw64)

[Table 1 - Strengths & Weaknesses table 13](#_heading=h.lnxbz9)

# 1.INTRODUCTION

For many individuals who are hearing or speech impaired, sign language serves as a crucial means of communication, allowing them to express thoughts, emotions, and needs with clarity and efficiency. Sign language, with its unique grammar and structure, enables fluid conversation between those who use it as their primary language. However, despite its significance, a considerable communication gap remains between sign language users and the majority of the population, which often does not understand sign language. This divide frequently results in miscommunication, frustration, and a sense of isolation for those who depend on sign language to communicate.

One of the primary challenges faced by sign language users is the lack of universal understanding of sign language among non-signers. While interpreters play a critical role in bridging this gap, they are not always accessible, either due to geographic limitations, availability, or financial constraints. In many day-to-day situations—whether at a doctor's office, in a classroom, or during casual conversations—sign language users are left without immediate support, resulting in limited interaction or complete communication breakdowns. This issue underscores the urgent need for more inclusive solutions that empower signers to communicate effectively in diverse settings without relying on human interpreters.

Recognizing this gap, the project at hand seeks to develop a web-based application that will leverage machine learning and computer vision to translate sign language into spoken language in real time. By employing advanced machine learning models trained on extensive sign language datasets, the system will be able to recognize and interpret the hand gestures and motions used in sign language, transforming them into spoken words or sentences that can be easily understood by non-signers. This approach offers the potential to significantly reduce communication barriers, enabling smoother interactions between sign language users and the broader community.

The core technology driving this system involves training computer vision models to accurately identify and interpret sign language gestures. This process involves feeding the models vast amounts of labeled sign language data, allowing them to learn the intricacies of hand movements, finger positioning, and facial expressions that are essential to sign language. With this training, the system will be able to recognize and translate the most common signs used in everyday communication. Over time, the accuracy of the system will improve as it is exposed to more data and more diverse gestures, ensuring that it becomes a reliable tool for real-time communication.

This project is designed with accessibility and ease of use in mind, making it a web-based platform that can be accessed from various devices, including computers, smartphones, and tablets. This ensures that users can have immediate access to the translation service wherever they are, without the need for specialized equipment. Furthermore, the system will be optimized for real-time performance, ensuring that the translation from sign language to spoken language happens instantaneously, minimizing any delays that could disrupt communication.

One of the key innovations of this project is its scalability. While the initial phase of the project will focus on translating a core set of commonly used signs, there is significant potential for growth. In future updates, the system will be expanded to support multiple sign languages, starting with American Sign Language (ASL) and eventually encompassing other widely used sign languages such as British Sign Language (BSL), Auslan (Australian Sign Language), and others. This will make the application applicable to a broader global audience, ensuring inclusivity for sign language users from various regions.

The application’s potential impact goes beyond everyday casual conversations. It has the capacity to improve accessibility in a wide range of environments, including educational settings, healthcare facilities, and professional workplaces. For instance, students who use sign language as their primary means of communication could benefit from the app’s ability to facilitate discussions with peers and instructors who may not be fluent in sign language. Similarly, in healthcare environments, where clear and accurate communication is essential, the app could help patients and medical professionals communicate more effectively, reducing the risk of miscommunication and enhancing the overall quality of care. In professional settings, employees who rely on sign language could participate more fully in meetings, discussions, and collaborations with colleagues, fostering a more inclusive work environment.

The broader social impact of this project is its potential to increase awareness and understanding of sign language and the needs of the Deaf and hard-of-hearing communities. By making sign language translation more accessible and commonplace, the project could help to normalize interactions between signers and non-signers, reducing the stigma that sometimes surrounds disabilities and communication differences. Moreover, the use of cutting-edge technology in this application underscores the importance of integrating inclusivity into the development of new technological solutions.

Beyond immediate communication needs, the project also seeks to contribute to the growing field of assistive technology by setting a precedent for the integration of artificial intelligence (AI) and machine learning in improving accessibility for individuals with disabilities. As AI continues to advance, the potential for creating even more sophisticated tools for sign language interpretation will only grow, opening the door to innovations such as automatic facial expression recognition, dialect-specific sign language variations, and personalized learning systems that adapt to the unique signing style of individual users.

In conclusion, this project represents a forward-thinking solution to the long-standing communication challenges faced by sign language users. By combining computer vision, machine learning, and a user-centric design approach, the web-based application will offer real-time translation of sign language gestures into spoken language, breaking down barriers and fostering more inclusive communication. The system’s scalability, adaptability, and focus on accessibility ensure that it will be a valuable tool for individuals across a wide range of social, educational, and professional contexts. Ultimately, this project aims to create a future where communication is seamless for all, regardless of the language they speak or sign, empowering millions of sign language users to engage fully in the world around them.

# 2. LITERATURE REVIEW

The following is a review of other existing systems.

## **2.1 Existing Systems**

#### **2.1.1** Springer Link (ASL) Recognition System

The recognition and translation of American Sign Language (ASL) have been a primary focus of SLR research. For instance, (Camgöz, 2018)developed a neural sign language translation system that uses Convolutional Neural Networks (CNNs) for feature extraction and a sequence-to-sequence model with attention mechanisms for translating sign language into spoken language. Their approach demonstrated significant improvements in translation accuracy, showing the potential of machine learning models for real-time ASL translation.

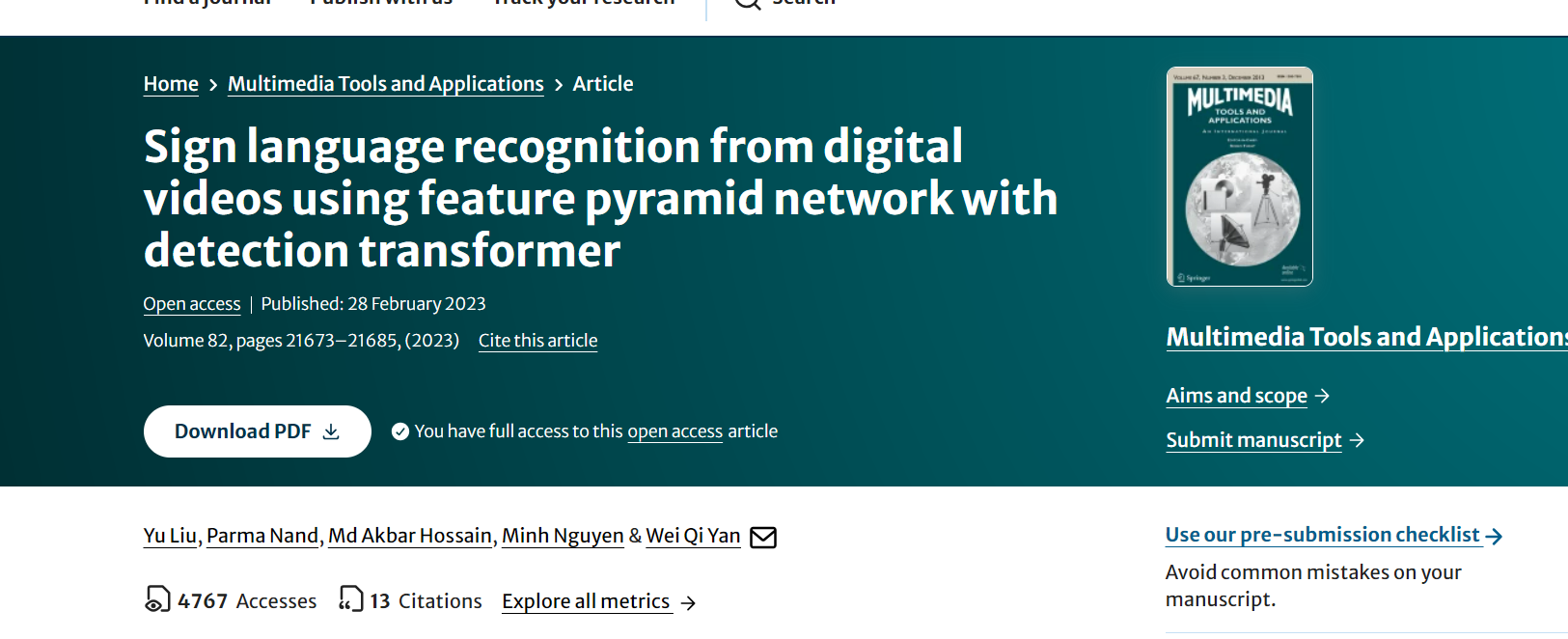


Figure - Screenshot of Springer Link home page(source: <https://link.springer.com>)

Figure one above shows a screenshot of the model that was made to interprete sign language from videos and images. It is a complete documentation on how the system was built from gathering data to producing output.

In developing a sign language interpreter application using machine learning, several key techniques and approaches have been explored in the literature. For instance, convolutional neural networks (CNNs) have been widely employed for sign language recognition due to their capacity to identify patterns in visual data such as hand movements and gestures. Elham and Hurroo's research emphasizes the use of CNNs combined with computer vision for accurate sign language recognition, which is highly relevant to building robust and scalable models (Camgöz, 2018).

Another significant trend is the use of deep learning models, such as those explored by Al Hammadi et al., who developed efficient hand gesture representation models using deep learning techniques (Association, 2022). These models improve recognition accuracy while minimizing computational complexity, making them suitable for real-time applications like a sign language interpreter.

In addition to CNNs, other techniques such as fuzzy decision trees and recurrent neural networks (RNNs) have been utilized for continuous sign language recognition. These methods address the challenges of recognizing signs in real-time and complex sentence structures.

A comprehensive approach is crucial, as it should involve both hand gesture detection and context-based understanding of signs to translate not just isolated words but full sentences. The combination of these technologies can lead to an inclusive and highly functional system capable of facilitating communication for the deaf and hard-of-hearing community in various settings.

In summary, by integrating machine learning techniques like CNNs, RNNs, and fuzzy logic into a sign language interpreter, your project can take advantage of existing advances to build an application that is both effective and accessible across different sign languages.

### **2.1.2 Sign All**

One of the most advanced sign language translation systems on the market today is SignAll, which was created by a Hungarian business. Using a combination of computer vision and artificial intelligence, this system translates American Sign Language (ASL) into text in real-time, marking a significant advancement in the field of sign language interpretation.



Figure 2- American Sign Language (ASL)(source: <https://signall.com>

*Figure - Real time translation illustration (source: www.nature.com)*

Figure 2 above illustrates how American Sign Language (ASL) may be translated into text in real-time. It shows a user making ASL movements, and a computer interprets those signals into English text on a screen.

The SignAll system uses depth sensors in conjunction with a multi-camera configuration to record the finer points of sign language communication. This hardware setup makes it possible to thoroughly analyze not only hand movements but also body posture and facial expressions, all of which are essential elements of sign language. Unlike many earlier translation attempts that depend only on 2D (two- dimensional) video input, the system can accurately comprehend signs in three-dimensional space thanks to the use of depth sensors.

One of Sign All's most remarkable characteristics is its continuous signing translation capability, which goes beyond basic word-for-word translation to accurately analyze entire sentences and express their meaning. Advanced machine learning algorithms, trained on large datasets of ASL signals and their meanings, are responsible for this capability.  
Nevertheless, there are benefits and drawbacks to the system's dependence on a unique hardware configuration. Although it enables very accurate and thorough sign capture, it also limits the system's accessibility and portability. SignAll is better suitable for stationary sites like offices, schools, or customer service centers due to the requirement for a regulated environment with precise camera and sensor configurations than for personal, mobile use.

.

### **2.1.3** Kenyan Sign Language (KSL) Initiatives

Kenyan Sign Language (KSL) is a critical communication tool for many individuals who are deaf or hard of hearing in Kenya. However, reliance on human interpreters for interactions involving KSL presents a significant challenge. Human interpreters are often unavailable or insufficient in number, which creates considerable barriers for KSL users in accessing essential services, including education, healthcare, employment, and social engagement. These barriers often result in missed opportunities, miscommunication, and social isolation for those who rely on KSL as their primary form of communication

In the context of education, the lack of accessible interpretation services has a direct impact on students who are deaf, limiting their participation and success in both basic and higher education. Similarly, in employment settings, the absence of interpreters or sign language-friendly environments hinders the inclusion of qualified individuals with hearing impairments. This exclusion is not only a loss for the individuals affected but also for the country’s workforce as a whole, which misses out on the talents and skills of a substantial portion of its population.

Recognizing these challenges, the Kenyan government has taken steps to promote inclusivity in the workforce through programs like Kazi Mtaani. Launched in response to youth unemployment, Kazi Mtaani aims to provide job opportunities while also addressing social inequalities. In its broader inclusivity goals, the program has recognized the importance of making workplaces more accessible to individuals with disabilities, including those who use KSL. This initiative reflects a growing awareness in Kenya of the need to ensure that employment opportunities are equally accessible to all, regardless of physical or communication barriers.

In 2022, the National Employment Authority (NEA) emphasized the need to increase inclusivity in workspaces by facilitating better access for individuals with disabilities. For those who are hearing-impaired, this includes efforts to provide sign language interpretation services and promote environments that cater to KSL users. Such initiatives mark an important shift towards recognizing the importance of communication access for the deaf community, allowing for fuller participation in the workforce, as well as in social and civic life.

## **2.2 Strengths and Weaknesses**

Table - Strengths & Weaknesses table

|  |  |  |
| --- | --- | --- |
| Website/Application | SignAll | Springer Link |
| **Strengths** |  |  |
| Real-time use | High-accuracy ASL translation for real-time interactions. | Real-time gesture identification for instructional applications. |
| Accuracy | Combines several cameras to increase precision. | Only needs a webcam, which makes it incredibly accessible. |
| Application | Utilized for communication in public and corporate settings. | Web-based and available everywhere for learning sign language. |
| **Weaknesses** |  |  |
| Hardware | Requires specialized equipment (multiple cameras). | Less accurate due to reliance on a single webcam. |
| Scope | Limited to ASL, no global language support. | Educational focus, not for real-time conversational use. |
| Cost | Not cost effective for large-scale use. | Free and accessible via browser. |

Table 1 above compares Sign All and a Springer Link application. Sign All offers high-accuracy, real-time ASL translation using multiple cameras, ideal for corporate communication, but it is expensive and requires specialized hardware. In contrast, the Springer Link app is free, web-based, and more accessible, using a single webcam for sign language learning. However, it’s less accurate and focuses on educational use rather than real-time conversations.

**Issues to be solved in weaknesses**

1. The sign language interpreter bridges communication gaps with high accuracy and less hardware.

2. It will ensure the deaf individuals can train their language.

3. They support equal opportunities and inclusion as it will be free to use.

# **3. Aims and Objectives**

The primary aim of this project was to serve as an innovative aid platform that empowers the hearing and speech impaired to feel fit in the society.

## **3.1 General Aims and Objectives**

The primary aim of this project is to develop a backend system for a Python-based sign language interpreter app, utilizing machine learning models to enable real-time sign language translation. The system will focus on accurately recognizing sign language gestures and converting them into spoken or written text. This project addresses the critical communication barriers faced by individuals who rely on sign language, particularly in educational, employment, and social settings.

This other objective is to facilitate a thorough literature survey of existing sign language interpreter applications, machine learning techniques for gesture recognition, and available datasets for training models which will help determine the appropriate machine learning algorithms, libraries, and tools necessary for successful backend development.

Furthermore this project aims to develop a robust backend in Python, focusing on model training, API development, and data handling, while implementing real-time gesture recognition using pre-trained models and fine-tuning them for enhanced accuracy. Additionally, the system aims to establish a scalable architecture capable of handling continuous data input and processing for fast and reliable translation of sign language into text or speech.

Thorough testing will guarantee the Real-Time Sign Language Interpreter System's efficacy. The accuracy and responsiveness of the RNN-based language conversion and the CNN-powered gesture recognition component will be put through a rigorous testing process. Verifying usability and pinpointing areas for enhancement through user testing with hearing and deaf people will guarantee the technology functions seamlessly in real-world situations.

## **3.2 Specific Aims and Objectives**

The specific aims and objectives of this project is to:

* Enable Fast Data Processing and High Accuracy for high accuracy for smooth communication.
* Improve the user experience and incorporate Text-to-Speech (TTS) technology to produce clear, natural spoken output.
* To test and validate equipment and improve system accessibility such as webcams.

# **4. PROPOSED PROJECT**

This chapter generally outlines the phases the project will undergo.The proposed project involves developing a robust backend system that can support real-time sign language interpretation via Python-based machine learning models. The backend will process input from the user (video or image of sign language gestures), run the input through a machine learning model to recognize the gesture, and then output the corresponding text or speech. This will significantly reduce communication barriers for sign language users in various environments.

## **4.1 Phases of the project**

The following is a breakdown on how the project will be undertaken.

**Phase One**: Research and Requirements Gathering

In the first phase, the focus will be on thoroughly understanding the technological requirements necessary for building the sign language interpreter system. This research will begin by identifying existing sign language datasets, which are crucial for training machine learning models. A careful review will be conducted to assess their suitability, and any gaps in these datasets will be identified for potential future data collection. Following this, the selection of appropriate machine learning algorithms is a key task. Techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) will be evaluated for their performance in gesture recognition, with the goal of selecting the most effective algorithms for accurately interpreting sign language gestures. Additionally, the system architecture will be defined, focusing on the design of APIs, data storage solutions, and seamless communication with the frontend of the application. The outcome of this phase will be a detailed requirements specification document that outlines the datasets, machine learning models, and overall system architecture.

**Phase Two**: Software Design

In the second phase, the backend system will be designed and implemented. The core component of this phase involves model training , where Python libraries such as TensorFlow or PyTorch will be employed to train machine learning models on sign language datasets. The training process will aim to fine-tune these models to achieve high levels of accuracy in recognizing gestures, accounting for variations in users and environmental factors such as lighting. Additionally, API development will be undertaken to facilitate communication between the backend and the user interface. These RESTful APIs will process video or image inputs, run the gesture recognition algorithms, and generate text or speech output. Another important aspect of this phase is database design, where a reliable storage solution (e.g., PostgreSQL or Firebase) will be set up to manage user data, logs, and training information. By the end of this phase, the backend system, including its APIs, model training functionality, and data management, will be fully developed and operational.

**Phase Three:** Testing and Documentation

The final phase focuses on ensuring the functionality, security, and scalability of the backend system through rigorous testing and detailed documentation. Unit and integration tests will be performed to verify that each component, including the API, machine learning model, and data handling, operates correctly and in harmony with the others. Furthermore, performance testing will be conducted to ensure the system can efficiently process multiple requests simultaneously without significant delays, ensuring the application remains responsive under varying loads. User acceptance testing will also be a priority during this phase, gathering feedback from end-users to assess usability and performance in real-world conditions. Based on this feedback, the system will be refined and optimized. Upon completion of this phase, the backend will be fully tested, validated, and ready for deployment, along with comprehensive documentation to support future maintenance and updates.

**Project Write-Up**

The project write-up documented the entire process, encompassing the research, design, implementation, testing, and feedback stages. Each phase was meticulously detailed, outlining the objectives, methodologies, and outcomes. The write-up included the technical specifications, design decisions, implementation challenges, and testing results. It also documented the feedback received during the first project presentation and the subsequent adjustments made to the platform. This comprehensive documentation served as a valuable resource for understanding the project's evolution and the rationale behind key decisions.

**Final Project Presentation and Submission of Report**

The final project presentation showcased the completed platform, highlighting its success, challenges, and overall outcomes. The presentation demonstrated how the feedback from the first project presentation was incorporated to enhance the platform. The final version of the project was evaluated based on its functionality, user experience, and technical robustness. The accompanying final report summarized the project’s achievements, evaluated its impact, and provided recommendations for future improvements. The report included an analysis of the project's strengths and areas for further development, offering insights for future iterations and potential enhancements. This comprehensive presentation and report marked the culmination of the project, demonstrating its value and potential in the field of career development.

## **4.2 Program of work**

This will talk about the steps the project will undergo before success

**4.2.1 Phase One: Research and Requirements Gathering**

* **Objective**: Define technical and functional requirements for the project.
* **Tasks**:
  + Identify datasets and machine learning models.
  + Define system architecture.
* **Timeline**: 1 week.

**Phase Two: System Design and Implementation**

* **Objective**: Build and implement the backend system.
* **Tasks**:
  + Train machine learning models using Python.
  + Develop and deploy APIs.
  + Ensure secure data storage.
* **Timeline**: 4 weeks.

**Phase Three: Testing, Evaluation and Documentation**

* **Objective**: Validate the system’s functionality and performance.
* **Tasks**:
  + Conduct unit, integration, and user acceptance testing.
* **Timeline**: 2 weeks.

#### **4.3 Final Project Presentation and Submission of Report Tasks:**

* Present the final version of the project to the lecturer.
* Outline the project’s success, challenges, and overall outcomes.
* Submit the final report that summarizes the project’s achievements, evaluates its impact, and provides recommendations for future improvements.

**Deliverables:**

* Presentation Slides

Final Project Report Submission

**Timeline:** 1 week

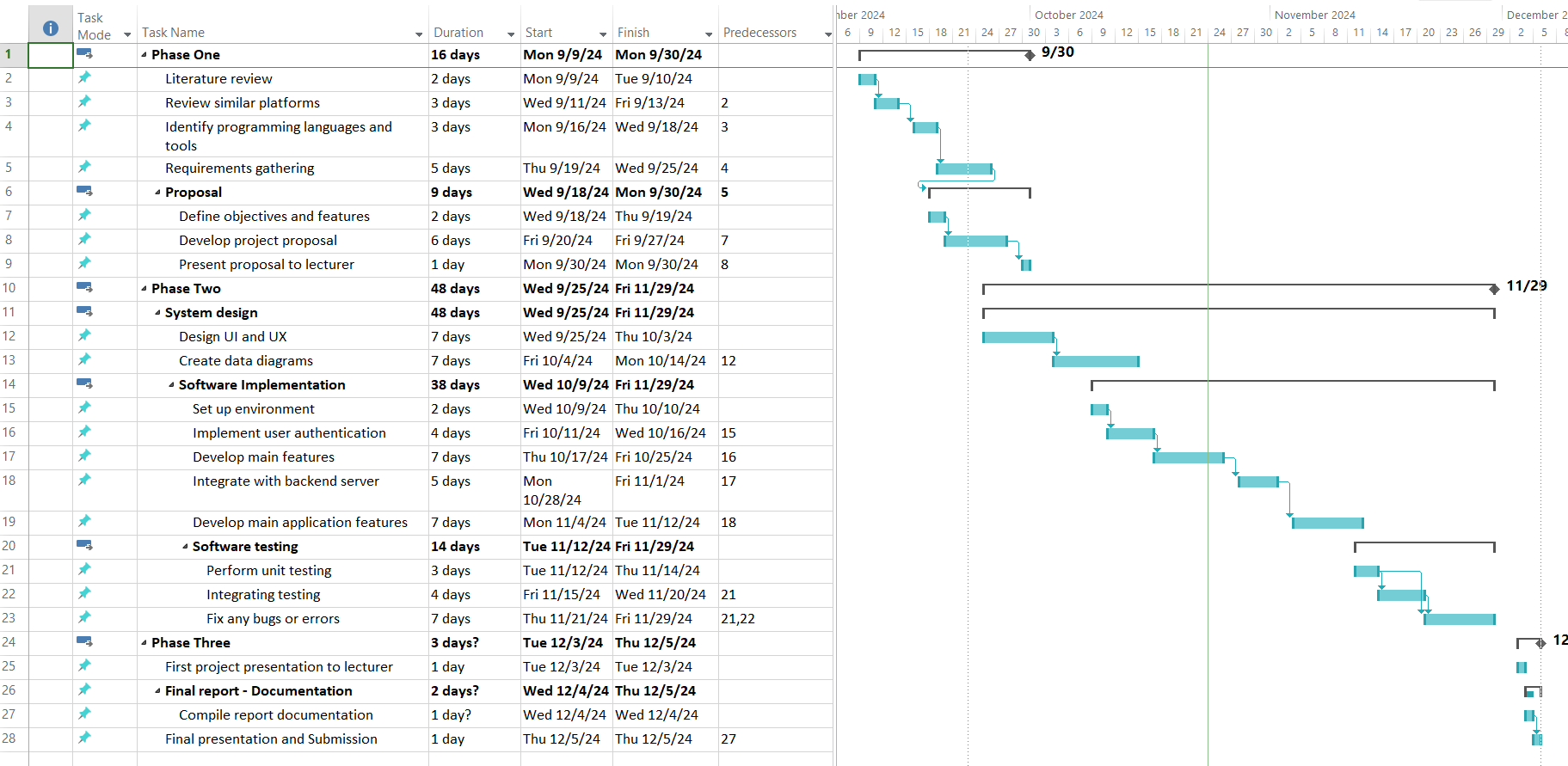


Figure 3- Gantt chart

**4.3 Gantt Chart**

Figure 3 above illustrate a Gantt chart for the career development web application project. The Gantt chart visually represents the project's timeline, spanning from August 2024, to December 8, 2024, and outlines the major phases and tasks involved. The chart indicates that each task has been assigned a specific duration, start date, and dependencies, ensuring a structured and sequential progression through the project phases. Each phase is marked by critical milestones that collectively aim to achieve the successful completion of the career development web application by the project's end date.

## **4.4 Requirements**

Below are the functional and non-functional requirements of this project:

#### **4.4.1 Functional Requirements**

These are the core features and behaviors that your app should have:

1. **Sign Language Recognition**:

The app will be able to recognize and interpret sign language gestures (e.g., American Sign Language, British Sign Language, etc.) using a camera or external device.

Provide real-time translation of sign language gestures to text or speech.

1. **Real-Time Communication**:

For live communication, the app should allow users to engage in real-time conversations with sign language interpreters or other users.

1. **Storage and Retrieval**:

The app must allow users to save their sign language interpretations or chat history for later reference.

#### **4.4.2 Non- Functional requirements**

**1. Performance:**

* The app should provide near-instant feedback when interpreting gestures or speech to minimize delays.
* The translation engine must be fast and responsive, especially in live communication or video calls.
  1. **Scalability**:
* The app should be able to handle a large number of concurrent users, especially if it includes real-time interpreter services or community-based features.
  1. **Response Time**:
* User actions, such as pressing buttons, submitting inputs, or initiating translations, should result in a response time under 2-3 seconds.

# 5. SYSTEM ANALYSIS AND DESIGN

This chapter mainly focuses on the designs and architecture of the web application.

## 5.1 Introduction

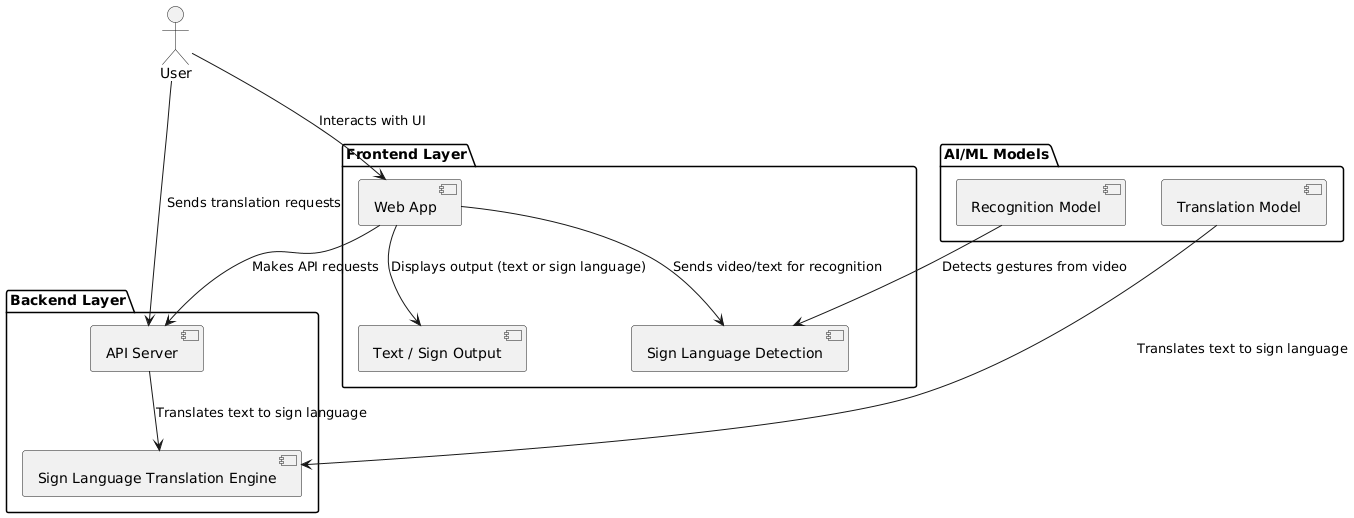
In the subsequent sections of this document, I delved into the intricacies of system analysis and design methodologies, exploring key aspects such as use case diagrams, class diagrams, data flow diagrams, and more. Throughout this system analysis and design journey, my focus remained rooted in understanding the unique requirements, models, and preferences of users for the system.

## 5.2 System Models

This section focuses on the use case diagrams and data flows.

### 5.2.1 System Architecture

This is where the system architecture for the sign language interpreter web app is explained



System architecture

The front-end components are highlighted in this system architecture diagram, which shows the architecture of the real-time sign language interpreter:

Presentation Layer:

* Web Browser: The client-side environment where the application runs.
* Back-end Application: Python, JavaScript.

Business Logic Layer:

* Tensor flow: Handles component-specific logic and state.
* API Integration: Interfaces with backend services and external APIs.

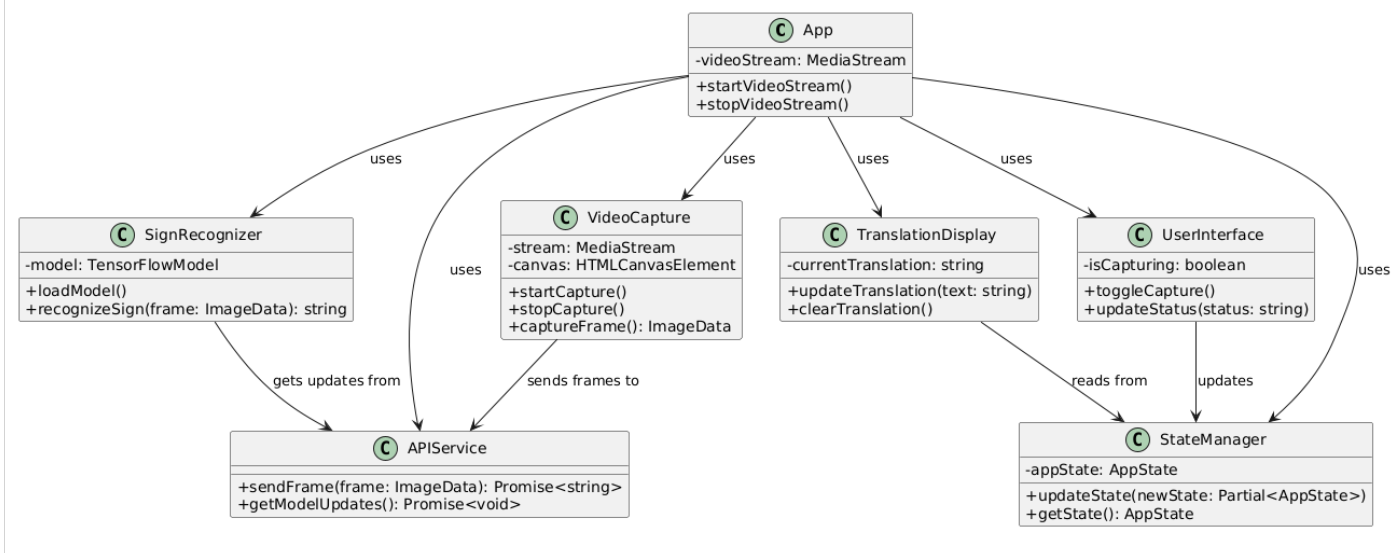
Data Layer:

* Firebase: Provides real-time database, authentication, and cloud functions.

This architecture makes sure that the real-time sign language translator has a modular, scalable, and maintainable structure, which promotes a smooth user experience and effective data handling.

5.2.2 Use Case Diagram

### UML Class Diagram

The figure below represents the architecture of a real-time sign language interpretation system is depicted in this UML diagram, along with the relationships between its many components. The primary entry point, which controls the video stream and provides general functionality,

*Figure - UML Class Diagram*

Figure 5

### UML Use Case Diagram

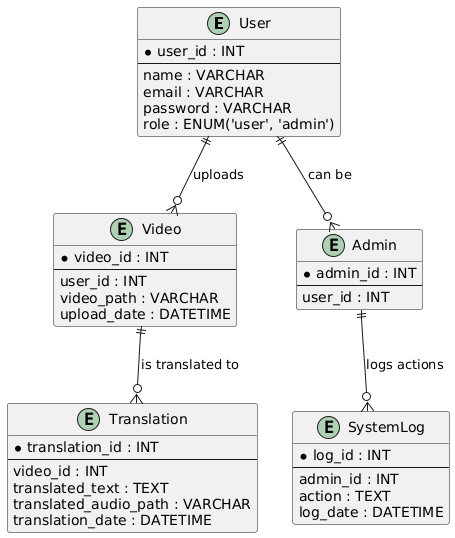
### 

Figure

*Figure - UML Use Case Diagram*

The use case diagram above depicts the real-time sign language interpreter system's core functionalities. It shows two actors: User and Admin. The User interacts with five main use cases: starting/stopping video stream, capturing sign language, toggling capture, and viewing translation. Internal procedures like sign recognition and translation presentation are triggered by these activities. The model for recognizing signs can be updated by the admin. All activities are coordinated by a common use case called "Manage System State". The flow diagram effectively summarizes the main functions and interactions of the program by showing how user inputs are processed by the system and turned into output.

### Entity Relationship Diagram

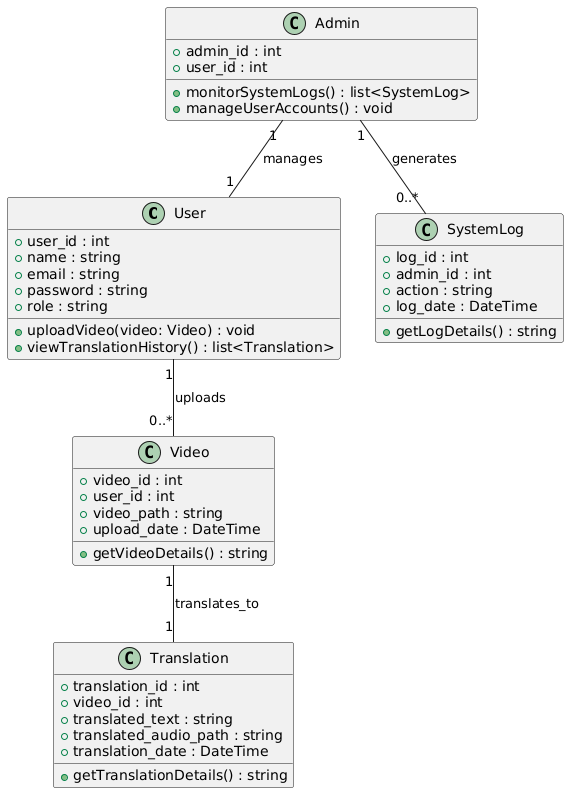


Figure

*Entity Relationship Diagram*

The Entity-Relationship Diagram (ERD), which illustrates the connections between important entities, provides an overview of the architecture of a real-time sign language interpretation system. The User object is associated with Session, which records the user's activities with start and finish times, and has characteristics such user ID, username, and email. Translations, which are produced during each session and comprise both the source and translated text, are produced. The Model item contains the Sign entity, which represents recognized motions, along with details about the various machine-learning models that are used to recognize gestures, including version and latest updated time. Translations are made using the Model, which is essential for understanding gestures. Overall, the relationship between users, sessions, translations, signals, and models within the system is evident from this ERD.

### Class Diagram

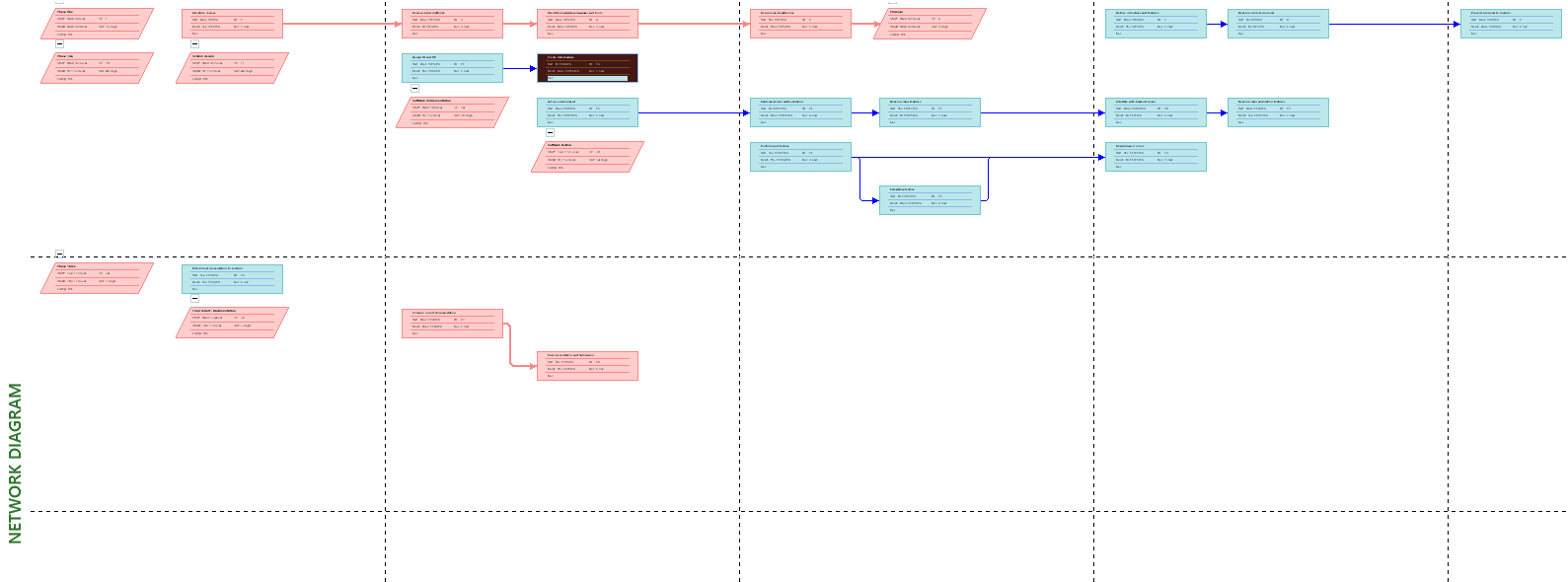


Figure

*Class Diagram*

The class diagram illustrates the connections between important parts and provides an overview of the architecture of a real-time sign language interpretation system. Details about each user are stored in the User class, and each user is associated with several Session instances that log session durations. Original and translated material are produced as Translations during each session. The Model class contains data about the recognition model, such as updates and version, whereas the Sign class represents gesture data and meaning. The model illustrates how the various parts of the system interact and is utilized in the translation process.

### Network Diagram

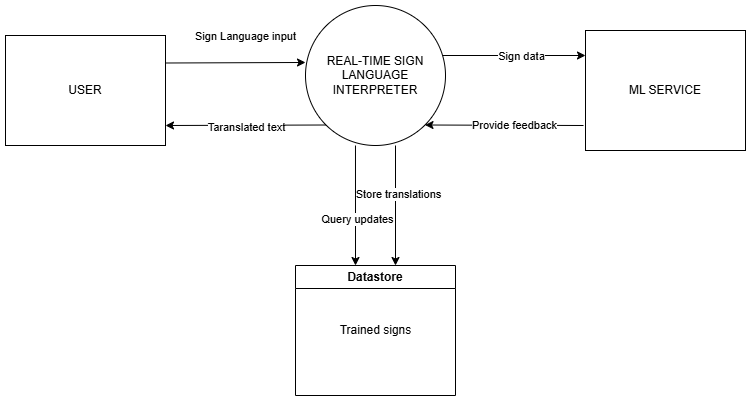


*Figure 8- Network Diagram*

Figure 8 above shows a real-time sign language systems main elements and processes involved in real-time translation are shown in this network diagram.

### Dataflow diagrams

Level 0 DFD



Figure

*Lvl 0Data Flow Diagram*

This Data Flow Diagram demonstrates Level 0 Data Flow Diagram for the Real-time Sign Language Interpreter system provides an overview of its core functionality.

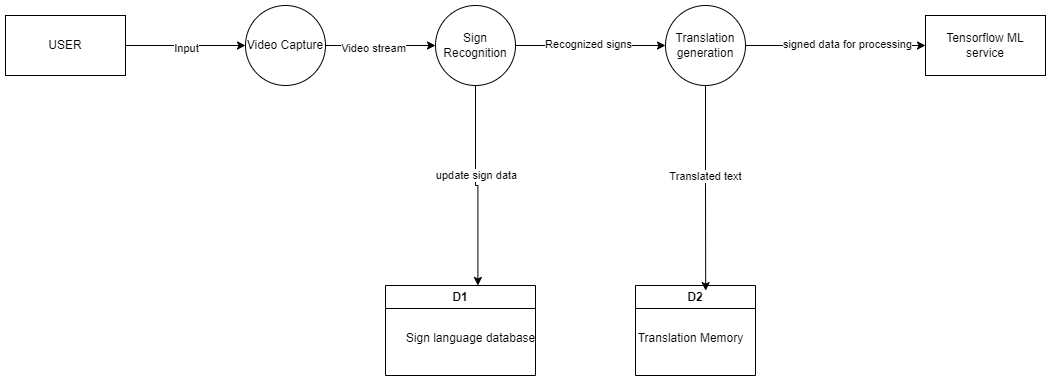
Level 1 DFD

Figure 10

Figure 10 above shows the primary procedure of the Real-time Sign Language Interpreter system is divided into three interrelated parts by the Level 1 DFD

Level 2 DFD

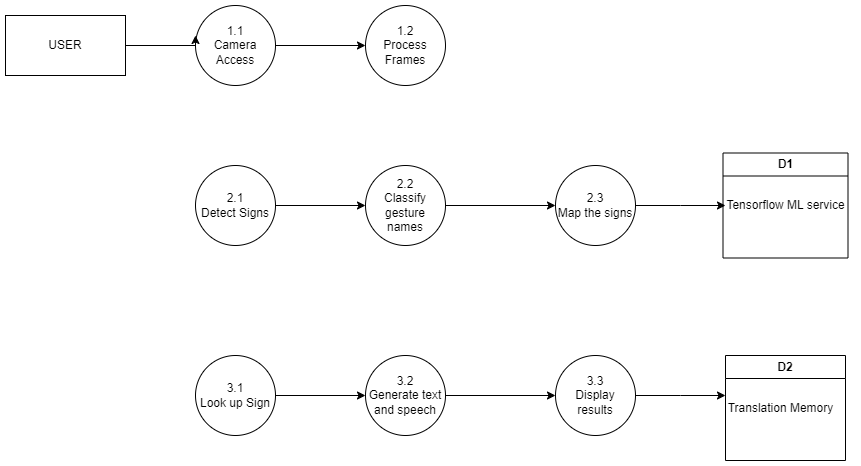


Figure  LVL 2 DFD

Figure 11 above breaks down processes into detailed steps: video input is captured, hands are detected and gestures classified, and signs are translated into text. Each step connects to data stores and external entities for accurate sign language interpretation.

# **Keywords (essential concepts)**

Real-time translation

Sign language

Machine learning

Computer vision

Convolutional Neural Networks (CNN)

Recurrent Neural Networks (RNN)

Natural language processing (NLP)

Gesture recognition

Speech synthesis

Accessibility

Inclusivity

Web-based system

# 6. IMPLEMENTATION

References

Bragg, D. K. (2019). Sign Language Recognition, Generation, and Translation.

Camgöz, N. C. (2018). In *Neural Sign Language Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7784-7793).

Chai. (2013). Automatic Face and Gesture Recognition. *Automatic Face and Gesture Recognition*, (pp. 1-6).

National Employment Authority. (2022). Retrieved from neaims.go.ke: https://neaims.go.ke/

World Health Organization. (n.d.). Retrieved from https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss