

Capstone Project Phase A

**SignQuest**

Project No. 24-1-D-7

**Supervisor: Dr. Reuven Cohen**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Email** |
| Guy Shlomo Banbo | 316139310 | guybanbo@gmail.com |
| Elior Malik | 204361505 | elior676@gmail.com |

[Link to Github](https://github.com/guybanbo/Teaching-the-American-Sign-Language-using-an-interactive-Video-System)

**Contents**

[**Abstract 4**](#_5blhe4s9m6zv)

[**1. Introduction 5**](#_3j27qof670io)

[**2. Literature review 7**](#_3htrx82sboj5)

[**2.2. Existing Tools 7**](#_qxq3fqp5qzi0)

[**2.3 Approaches for sign language recognition 8**](#_tlxsfdoshqod)

[**2.4 ASL Datasets 11**](#_p7klvkgtxnd3)

[**3. Expected Achievements 12**](#_i3qd3e6cg07z)

[**3.1 Creating Teaching Method 12**](#_jppcwtkr2uwb)

[**3.2 Creating a Model 13**](#_zfjc1xic3s41)

[**3.3 Creating dataset for American Sign Language 13**](#_c0s3f43ur33b)

[**4. Engineering Process 14**](#_hzjyry9uvsg7)

[**4.1 Process 14**](#_s9h2u2dfr3v6)

[**4.2 Product 19**](#_8qv7uv1pjhab)

[**5. Evaluation / Verification Plan 24**](#_xid3wr8erp5d)

[**5.1 Evaluation 24**](#_ebwaq13m763y)

[**5.2 System Requirements 25**](#_ee0gzw3cmei1)

[**5.3 Testing 26**](#_vk08loq5u4hb)

[**6. References 27**](#_lu2u5xfuc1an)

# *Abstract*

The ability to communicate effectively is a fundamental human need, and for the deaf and hard-of-hearing community, sign language serves as their main way of communication. However, learning sign language can be challenging, particularly for those without access to qualified instructors or other signers. Our project deals with the development of SignQuest, an application designed to assist in sign language learning through live sign video recognition using machine learning techniques.  
  
SignQuest consists of three main parts, The first part is the teaching system that will teach the ASL in the basic level to enable day to day communication, Each lesson will include exercises with live sign language recognition.

The second part of the system is a dataset, which will include videos of ASL word gestures taken from different sources of ASL datasets.  
The dataset will be used to train the machine learning model for sign language recognition. In addition, the videos will be used in each lesson session to demonstrate the gesture to the user.  
  
The third part of the system will be the recognition model that will be trained with the video dataset in order to accurately recognize the ASL gestures.

The model will utilize the MediaPipe framework for extracting hand, pose and face landmarks, and the machine learning model will be the neural network LSTM (Long short-term memory) for capturing the sequential and contextual nature of the sign language gestures.

This project aims to provide an accessible and effective solution for sign language learning, bridging the communication gap for the deaf and hard-of-hearing community.

# Introduction

Over 5% of the world’s population – or 430 million people – have disabling hearing loss. It is estimated that by 2050 over 700 million people – or one of every ten people – will have disabling hearing loss.

‘Disabling’ hearing loss refers to hearing loss greater than 35 decibels (dB) in the better hearing ear. Nearly 80% of people with disabling hearing loss live in low- and middle-income countries.

A person who is not able to hear as well as someone with normal hearing – hearing thresholds of 20 dB or greater in both ears – is said to have hearing loss. Hearing loss may be mild, moderate, severe, or profound. It can affect one ear or both ears and leads to difficulty in hearing conversational speech or loud sounds.

'Deaf' people mostly have profound hearing loss, which implies very little or no hearing.   
In the United States alone, 500,000 hearing-impaired and deaf people use American sign language (ASL). These are only 1% of the hearing-impaired and deaf population in the United States.

The availability and quality of educational resources are limited, the problem we are trying to address is the lack of using sign language among those who are hearing-impaired.In addition it is targeted to the ones that wish to learn sign language in order to communicate with deaf people. Therefore, we’ll create a solution that will enable a more accessible and interactive way to learn sign language.

Today, there are some solutions for learning the American sign language, one of them is **StartASL**, which provides online and offline courses for learning the American sign language.  
This service doesn’t give immediate feedback for the gestures. For this purpose there is a need for hiring a private instructor that provides feedback with high cost.  
  
Our solution is to create an application that provides an easy and intuitive way to learn the American sign language.  
Starting with the basic words of the language and step by step moving to more complex words and sentences.  
In the studying process, a video with instructions on how to perform a hand gesture will be displayed on the screen, and then the user will use a simple web camera and try to mimic it, if a gesture was performed correctly by the user a feedback will be shown.

The solution we provide will help to solve the problem in the following ways:

1. No special hardware such as Kinect, sensory gloves or VR glasses etc are needed, so it will be accessible to everyone.
2. The application provides immediate feedback for the performed gesture, therefore no human instructor needed.
3. In contrast to face-to-face studying, The application is available to everyone 24/7 and they can choose when to use it.

The stakeholders are people who have hearing disabilities and also people who often interact with those people, including family, teachers, classmates, service providers and others.  
  
The project book is divided into 5 chapters.  
Chapter 1 is the introduction of the project that is written above.  
Chapter 2 includes a literature review of the existing technologies and tools related to ASL teaching and recognition, and also relevant datasets of ASL words.  
Chapter 3 describes the goals of the project, including the creation of an effective ASL teaching method, a recognition model and a dataset that supports this goal.  
In this chapter we’ll also define the criterias for evaluating the success of the application.

Chapter 4 consists of two parts - process and product:

The processpart includes the motivation of the project, the stages of the engineering development process we completed so far. It also includes the technologies we chose for the project, including tools for creating a sign language recognition model and the development tools. then it describes the future stages of the development process, the constraints and the challenges in the process.

The product partincludes the details of the system architecture and the workflow, and the use cases of the system, described using UML diagrams. In addition, we present a prototype of the user interface.

Chapter 5 includes the evaluation methods we will use for the evaluation of the success criteria of the project, the requirements of the system and a testing plan.

# Literature review

* 1. In the field of sign language recognition and teaching, many applications have been developed utilizing machine learning algorithms to recognize sign language across multiple languages. Nevertheless, there is a lack of applications that enable users to learn sign language through real-time hand gesture recognition from a live video feed.  
     Existing tools are divided to few categories:   
     **Teaching games** - games that are teaching ASL, they are both using gesture recognition.  
     **Translation tools** - tools for translating from text to sign language and vice versa.  
     **Video Lessons -** applications that offer video lessons for teaching the ASL.

## Existing Tools

* + 1. **CopyCat [**[**1**](#bn2pgmjwjl5i)**] -** CopyCat is a game that helps deaf children acquire language skills while they play the game. The game uses American Sign Language (ASL) as the primary mode of communication between the child and the computer. Previously, CopyCat relied on limited hardware options like custom gloves for sign verification, but modern 4K cameras and pose estimators present new opportunities.  
       Later, an improvement was proposed to CopyCat by using Kinect camera which enables using the game without wearing gloves.
    2. **StartASL -** This web application offers free basic courses, as well as paid courses of video lessons for teaching the ASL
    3. **Sign.mt [**[**7**](#u3hi5nb22qa8)**] -** This web application enables users to translate text from various languages into sign languages and vice versa. Additionally, it provides the functionality to create a visual avatar that mimics the gestures of the translated word.
    4. **JengASL [**[**8**](#36b9l2owvwz)**] -** JengASL is a gamified approach to sign language learning in virtual reality. It uses 3D hand models, gesture recognition, and interactive gameplay to teach American Sign Language. The game is based on the classic Jenga game.
    5. **Lingvano -** This mobile application teaches the American sign languages by video lessons that are made by deaf teachers. The app also offers an ASL dictionary for looking for a specific sign or sentence.

## 2.3 Approaches for sign language recognition

There are several approaches and tools for sign language recognition using machine learning techniques. A popular method uses convolutional neural networks (CNNs) model to analyze visual input like video frames or images for classifying the hand gestures and body postures that comprise sign language. Long short-term memory (LSTM) networks are a type of recurrent neural network techniques that are capable of capturing temporal dependencies and modeling the sequential nature of sign language. Additionally, frameworks like Google's MediaPipe provide pre-built models and tools that enable sign language recognition by providing real-time tracking of body landmarks, hand keypoints, and other visual cues relevant to sign motions and hand shapes.   
 **2.3.1 A Convolutional Neural Network (CNN) [**[**3**](#sfsyhdumgok5)**] -** is a type of deep learning algorithm specifically designed for image recognition and processing tasks. It excels at analyzing and understanding visual data, making it particularly well-suited for computer vision applications.

Convolutional Neural Networks (CNNs) used in advancing sign language recognition by enabling accurate and robust recognition of hand gestures and other sign language component

CNNs models are trained using large datasets of labeled images, allowing them to learn to recognize patterns associated with specific objects or classes. Once trained, a CNN can be used for various tasks, including image classification, object detection, and image segmentation. Their success extends beyond computer vision, finding applications in fields like self-driving cars, medical imaging, and security systems.

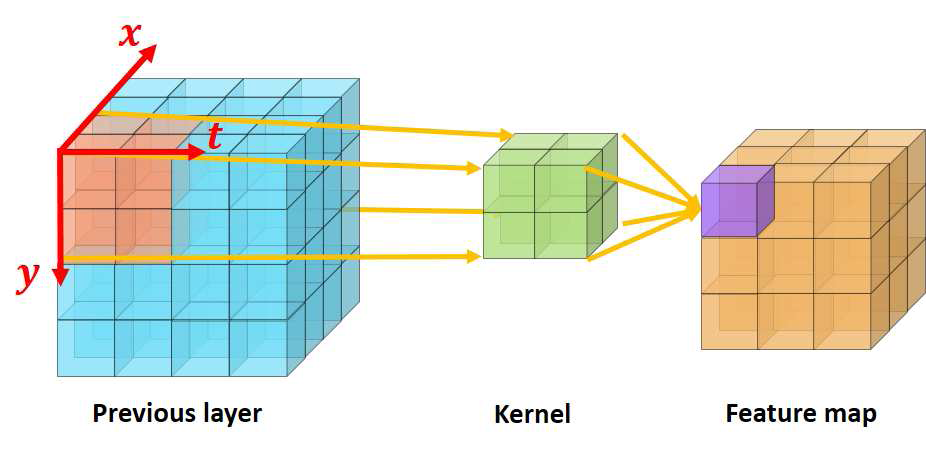
In the article [[3](#sfsyhdumgok5)] an American Sign Language (ASL) fingerspelling(gestures of letters) translator that is based on a CNN is proposed.

**2.3.2 3D CNN-**

A 3D Convolutional Neural Network (3D CNN) is a type of deep learning architecture that extends the traditional 2D CNN to work with three-dimensional data. It is particularly useful for tasks involving volumetric data, such as video analysis, medical image analysis (e.g., CT scans, MRI), and 3D object recognition or classification.

In a 3D CNN, the convolutional filters and pooling operations are performed across three dimensions (height, width, and depth or time) instead of just two dimensions (height and width) as in a 2D CNN. The 3D filters are essentially 3D kernels that slide across the volumetric input data, capturing spatial and temporal information simultaneously.

In the article [[4](#pc8szcjj0g9l)] the authors developed a 3D CNN model for sign language recognition. The model learns and extracts both spatial and temporal features by performing 3D convolutions. To boost the performance, multi-channels of video streams, including color information, depth clue, and body joint positions, are used as input to the 3D CNN in order to integrate color, depth and trajectory information.

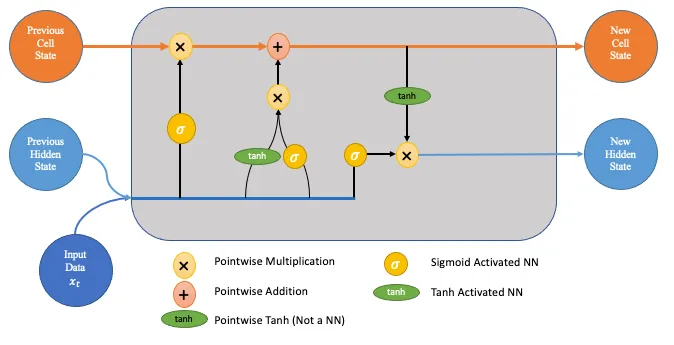
  
*figure 1: 3D convolution*

**2.3.3 LSTM(RNN)** [[9](#h9ot5gek6z7b)] - A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning. The algorithms are incorporated into popular applications such as Siri, voice search, and Google Translate.

The downside of RNN is that it can only remember the immediate past input. It can’t use inputs from several previous sequences to improve its prediction.

In order to solve this problem, a variant of RNN called Long short-term memory (LSTM) that enables the model to expand its memory capacity to accommodate a longer timeline can be used.

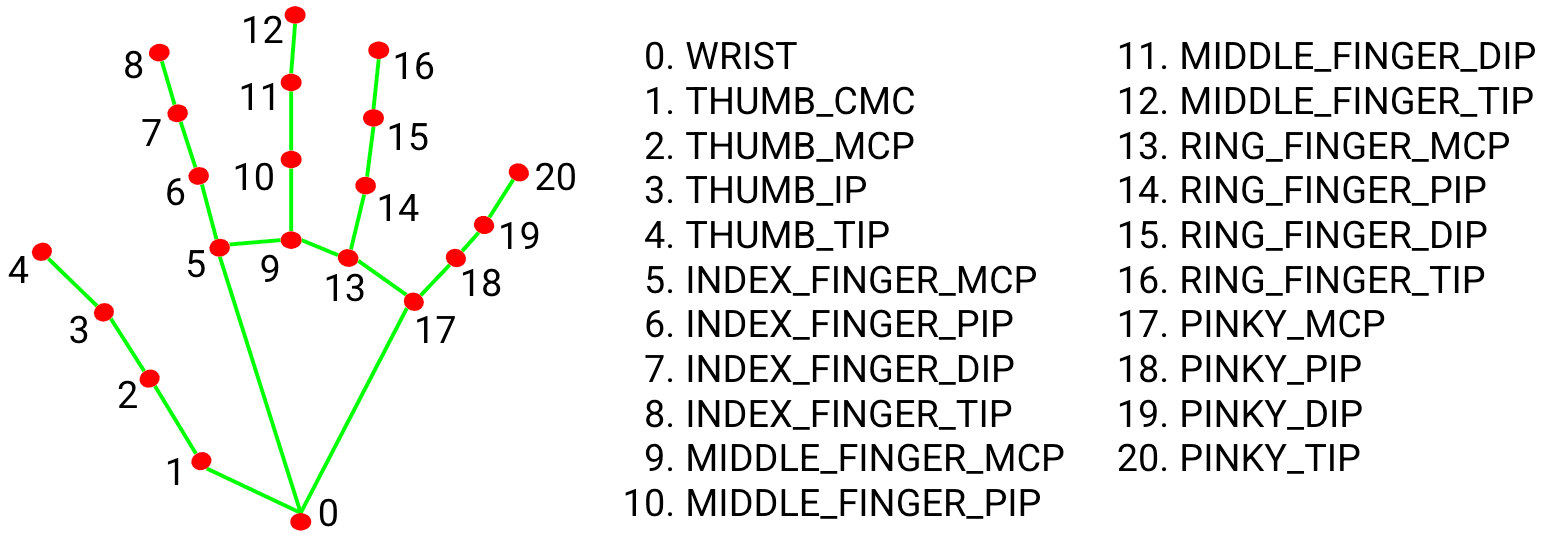
LSTM networks add a special memory block called *cells* in the hidden layer.   
Each cell is controlled by an input gate, output gate, and forget gate, which enables the layer to remember helpful information.  
  
In the article [[9](#h9ot5gek6z7b)] a system for American Sign Language (ASL) alphabet recognition that integrates LSTM and MediaPipe was proposed. The system was able to recognize the letters with accuracy of 99%. The authors concluded that the work can be further extended for word recognition.

  
*figure 2: LSTM Architecture*

**2.3.4 MediaPipe [**[**6**](#huz9y96qf1ph)**]** - MediaPipe is an open-source framework developed by Google that facilitates the creation of real-time multimedia applications. It provides a suite of pre-built machine learning models and tools for various tasks. MediaPipe Holistic which is a solution under the Mediapipe framework includes pose estimation, face detection, hand tracking, and more.  
MediaPipe simplifies complex computer vision tasks by using pre-trained models to analyze images or videos in real-time. It identifies key points such as body joints, facial landmarks, and hand keypoints, enabling applications like sign language recognition, augmented reality effects, and more.

The MediaPipe holistic provides the following landmarks:

* Hand - The hand landmarks consist of 21 3D coordinates representing the positions of key points on the hand, such as the fingertips, knuckles, and wrist. These landmarks can be used to track hand gestures, hand poses, and interactions with virtual objects.
* Pose - The pose landmarks consist of 33 3D coordinates representing the positions of key points on the human body, including shoulders, elbows, wrists, hips, knees, and ankles. These landmarks can be used for pose estimation, tracking body movements, and analyzing body gestures.
* Facial - The facial landmarks consist of 468 3D coordinates representing the positions of key points on the face, such as the eyes, eyebrows, nose, mouth, and jawline. These landmarks can be used for facial recognition, facial expression analysis, and facial motion tracking, among other applications.

**

*figure 3: Hand Landmarks representation used in Mediapipe*

## 2.4 ASL Datasets

There are a few datasets of videos of ASL gestures, the largest of them being WLASL.  
  
**2.4.1 WLASL [**[**5**](#n8ov7tpgh5wu)**]** (Word-Level American Sign Language) is a large video dataset specifically designed for American Sign Language (ASL) recognition at the word level. It contains 2,000 common different words in ASL, with each video corresponding to a specific ASL word. Researchers and developers can use WLASL to train and evaluate models for accurate word-level sign language recognition  
  
**2.4.2 How2Sign[**[**2**](#92kxu77q60ty)**] -** a dataset of more than 80 hours of sign language videos and corresponding modalities, including speech, English transcripts, and depth. It is a multimodal and multiview continuous American Sign Language (ASL) dataset 1. The dataset also includes a three-hour subset that was recorded in the Panoptic studio, enabling detailed 3D pose estimation.  
  
**2.4.3 MS-ASL -** This real-life large-scale sign language dataset comprising over 25,000 annotated videos and evaluated with state-of-the-art methods from sign and related action recognition can help researchers build machine learning based models to help advance the sign language recognition community.

**2.4.4 ASL-LEX** - a database of lexical and phonological properties of American Sign Language signs. It was first released in 2016 with nearly 1,000 signs. ASL-LEX was updated in Fall 2020 with greatly expanded information and an increased size of 2,723 signs.

**2.4.5 SignASL.org** - The largest collection of video signs online. The collection contains videos from multiple sources and Currently there is just over 40,000 videos listed on this site, Signed by more than 150 different signers of different ages, ethnicities and genders.

# Expected Achievements

The main goal of this project is to develop a user-friendly application that accurately recognizes American Sign Language (ASL). By leveraging machine learning algorithms and a comprehensive dataset of ASL videos, the application will provide an effective and intuitive platform for both deaf and hearing individuals to learn and communicate using ASL. The application will focus on teaching core vocabulary and phrases essential for day-to-day communication, enabling users to achieve a functional level of ASL proficiency through interactive exercises and feedback. The success criteria include the ability for users to learn and memorize ASL words. Another criteria is achieving a user satisfaction rate of at least 85% regarding the overall teaching approach and effectiveness.

## 3.1 Creating Teaching Method

The developed app will have a user-friendly interface that everyone can easily use.   
The app will be able to teach both deaf and hearing people different ASL words and sentences in the level of day to day communication.  
The teaching process will be divided into units and lessons.

The app will include different types of exercises:

1. In each lesson the system will show a video of the learned gesture(s) and the student will be asked to repeat the gesture. After correctly signing the gesture the system will show the user a feedback message and he will continue to the next exercises in the lesson.
2. The system will show a video of a gesture and the student will be asked to choose the correct answer from multiple possible choices.
3. The system will show a video of a gesture and the student will be asked to type the corresponding word.
4. The system will show a sentence in English with a missing word and the student will be asked to guess and sign the word.

The different types of exercises are based on memory recognition and recall to ensure an effective learning process.  
At the end of each lesson, a score for the lesson will be calculated based on the exercises that were completed successfully.

The criteria for success is the ability of a user to learn and memorize words and it will be evaluated based on the lesson's score.

In addition, there will be an option to send feedback on the overall teaching level. The success criteria for the system is targeted to at least 85% satisfaction.

## 3.2 Creating a Model

Our goal is to create a machine learning model that will accurately recognize the ASL hand gestures.   
To achieve this goal, we will use a neural network algorithm together with a diverse dataset. The algorithm's capacity to identify complex patterns makes it well-suited for recognizing ASL gestures.  
The trained model will recognize a live gesture in a few seconds with accuracy of at least 77% [[1](#bn2pgmjwjl5i)].

## 3.3 Creating dataset for American Sign Language

Currently, there are multiple ASL datasets, but they have some issues:

* One of the dataset has quite a large amount of words but on the other hand it has few videos of each word.
* Another dataset that has missing videos links of words
* There also old datasets that have a low quality of videos

Our goal is to create a dataset of words that relate to day-to-day speech to enable an easy way to communicate with hearing impaired people.  
The dataset will include videos of ASL word hand gestures from different sources of existing ASL dataset, in order to have sufficient examples to train our model.  
Examples of words and phrases we will include in our dataset are: “hello”, “goodbye”, “eat”, “how are you?”, “ thank you”, “deaf”, “what?”, “where”, “when”, “learn” and more.  
We will collect videos mainly from 2 sources:

1. WLASL
2. SignASL

Since SignASL itself collects videos from multiple sources and because WLASL is the largest world level ASL dataset, they can help us create a diverse and big enough dataset to properly train our model.

# Engineering Process

## 4.1 Process

**4.1.1 Motivation**

The main objective of our project is to bridge the communication gap for hearing-impaired people by creating an app for teaching everyone ASL. Imagine a world where they can express themselves directly, without relying on intermediaries. Our system aims to empower hearing-impaired people to communicate effortlessly with the rest of the world using sign language. We’ll try to achieve it by making it easier for everyone to learn the american sign language, deaf and non-deaf alike.

**4.1.2 Stages of the engineering development process**

* Gathering information about hearing problems and deafness
* Searching and reading articles in the fields of ASL video recognition and ASL teaching.
* Gathering information about existing technologies in the field of learning ASL
* Looking for datasets of ASL gestures videos
* Choosing technologies and tools for the implementation of the system
* Creating screens prototype and UML diagrams that describe the design and functionality of the system

**4.1.3 Chosen technologies**After searching technologies for our sign language teaching application, we chose a few technologies for creating the recognition model and developing the application.  
For the sign language recognition model we chose to use MediaPipe that will help to extract the hand,pose and facial landmarks. In addition we chose LSTM which is a neural networks model that captures the sequential and contextual nature of sign language gestures.  
For the application development we chose to use Python as the programming language of the application, along with Tkinter which is a GUI framework and MongoDB for storing information.

**MediaPipe** – We’ll use the MediaPipe framework because it enables the extraction of the hand, pose and face landmarks from a video. Those landmarks are essential for tracking and recognition of an ASL gesture.  
MediaPipe can extract the landmarks accurately even in cases of weakened hardware devices and lower quality cameras.

**LSTM** – Sign language involves dynamic gestures and sequences of movements.

LSTM [[9](#h9ot5gek6z7b)] is designed to handle sequential data, making it well-suited for capturing the gestures in sign language with sequence of movements that comprise the whole gesture.

The information stored by the LSTM model is essential for creating an   
accurate model for recognizing ASL words and even sentences.

**MongoDB**- We chose to store our dataset of ASL words and matching video links and the users’ information in the MongoDB database.

We chose MongoDB for a couple of reasons:

1. The MongoDB database has a flexible structure that enables it to make changes easily in the future and to scale the app.
2. MongoDB can be easily accessed via python and there is no need for a middleware.

**Tkinter** - We will use Tkinter to design the app’s user interface.

Tkinter is the most used environment for GUI generation in Python.

It is simple and easy to use, and enables you to create a GUI relatively fast.

**4.1.4 Future stages of the development process**

**Phase 1**: **Creating a dataset -** collecting videos from multiple sources and building a dataset of common ASL words. For each word in the dataset we will include a sufficient amount of high quality videos from a few different people. **Phase 2:**  **Model Training** -

MediaPipe will be used to extract hand, pose and face landmarks from video frames. These landmarks are essentially coordinates that represent the position of various points on the body including the hands, such as the tip of the thumb or the base of the pinky finger.

Once the landmarks are extracted, they will be fed into a LSTM network to train it.

**Phase 3**: **Gesture Recognition** - We will use the MediaPipe framework to extract the landmarks of the hands, pose and face from the live video. The landmarks will be fed into the model that will predict which word is being signed by the user’s movement and if it fits to the word that is being taught to the user.

**Phase 4: Developing the ASL learning app** - in this phase we will create the asl learning app. The phase will include creating the different screens and levels of learning and integrating the asl video recognition model in the learning process, enabling immediate feedback to the students.

**4.1.5 Constraints**  
In phase one of the development process we create a dataset of videos of asl signs by gathering videos from multiple datasets. In most of the videos the signer is standing. The consequence of this fact is that using the app while sitting may result in difficulty of the recognition model to correctly predict the user’s hand gestures.

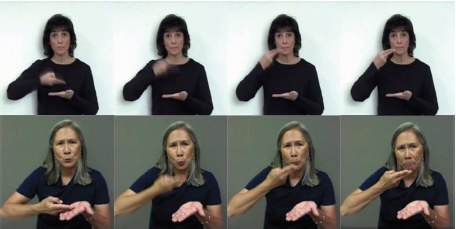
Another constraint is the distance of the user from the camera. The angle of view may vary between different cameras, which can result in some areas relevant for the recognition to be missing from the frame. The user will have to locate himself in some distance from the screen in order to get an accurate recognition.

Because of the constraints mentioned above we prefer to use an external camera over using a built-in one. In addition, we’ll add an instruction that will be shown to the user the first time he uses the app that explains how to correctly position the camera and himself to get good results.  
  
**4.1.6 Development process challenges**

**Hardware limitation -** In the development of our sign language recognition model, we will utilize MediaPipe to extract landmarks and estimate the pose and hand movements of the signer. Additionally, to effectively recognize sequential hand gestures, we will integrate LSTM, a type of RNN specifically designed for handling sequential data.  
To successfully train this model, it is essential to have a powerful GPU and CPU. One possible solution is to leverage cloud-based services like Google Colab, AWS EC2, or Microsoft, as they provide access to robust CPUs and GPUs. By utilizing these services, we can ensure the efficient execution of our model.

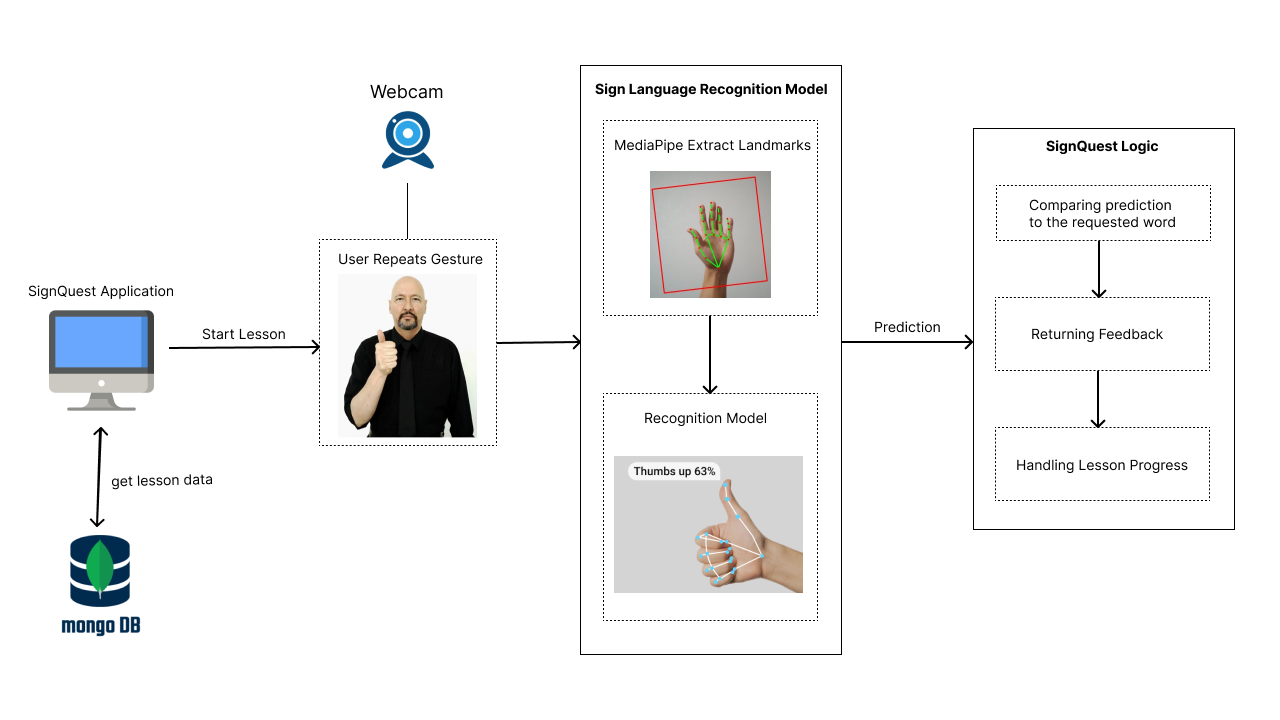
**Dataset Collecting -** There are several ASL datasets, none of them provide an adequate number of videos for each word, which is essential for optimal model training.  
To address this issue, our approach is to merge various ASL datasets in order to create a diverse dataset. Additionally, we will have to select certain words that are relevant to our teaching program.

**Creating Teaching Method -** Finding an effective teaching method to aid users in remembering sign language poses a significant challenge. In addition to recognizing the hand gestures performed by the user, there are several alternative approaches worth exploring. For instance, presenting the user with videos of gestures and right after that asking them to identify the corresponding word (through open-ended or multiple-choice questions). Other options are completing a sentence by performing a gesture, or even utilizing memory games.

**Gestures with multiple meanings-** According to article [[5](#n8ov7tpgh5wu)] some ASL gestures have more than one meaning, which may pose a challenge to the development process and the creation of the lessons.  
To solve this problem, we will display all the meanings of the gesture to the user and we will save the list of meanings of each word with the matching gesture video on our DB.

*figure 4: The same sign represents different words “Rice” (top) and “soup” (bottom).*

## 4.2 Product

**4.2.1 System Architecture**

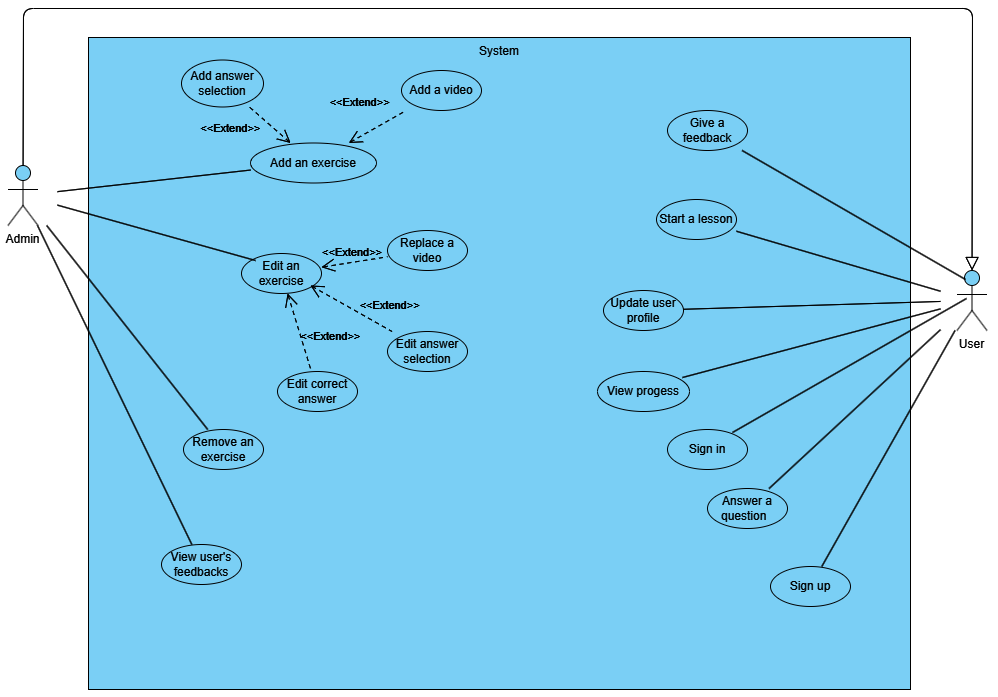
*figure 5: System Architecture*

First, the SignQuest application gets the lessons data from the mongoDB database.

At the start of a lesson the user is asked to repeat an ASL gesture that is shown in an instruction video. The sign language recognition model uses MediaPipe to extract the landmarks and predict the performed user’s gesture.

Then, the prediction is compared to the requested word. In case of a match, a feedback is returned to the user and the lesson continues.

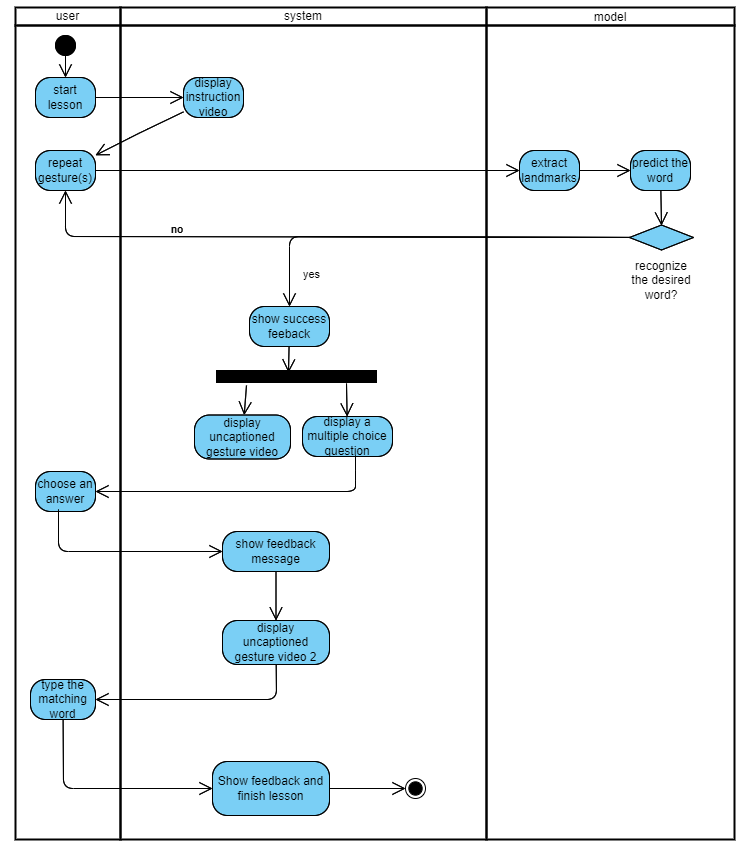
**4.2.2 Use Case Diagram**

*figure 6: Use case diagram*

**Add an exercise** - adding a question (exercise) that is part of a lesson. Some of the exercises include a video or multiple-choice answer.  
  
**View User’s Feedbacks** - the admin can view users’ feedbacks on the overall users’ experience

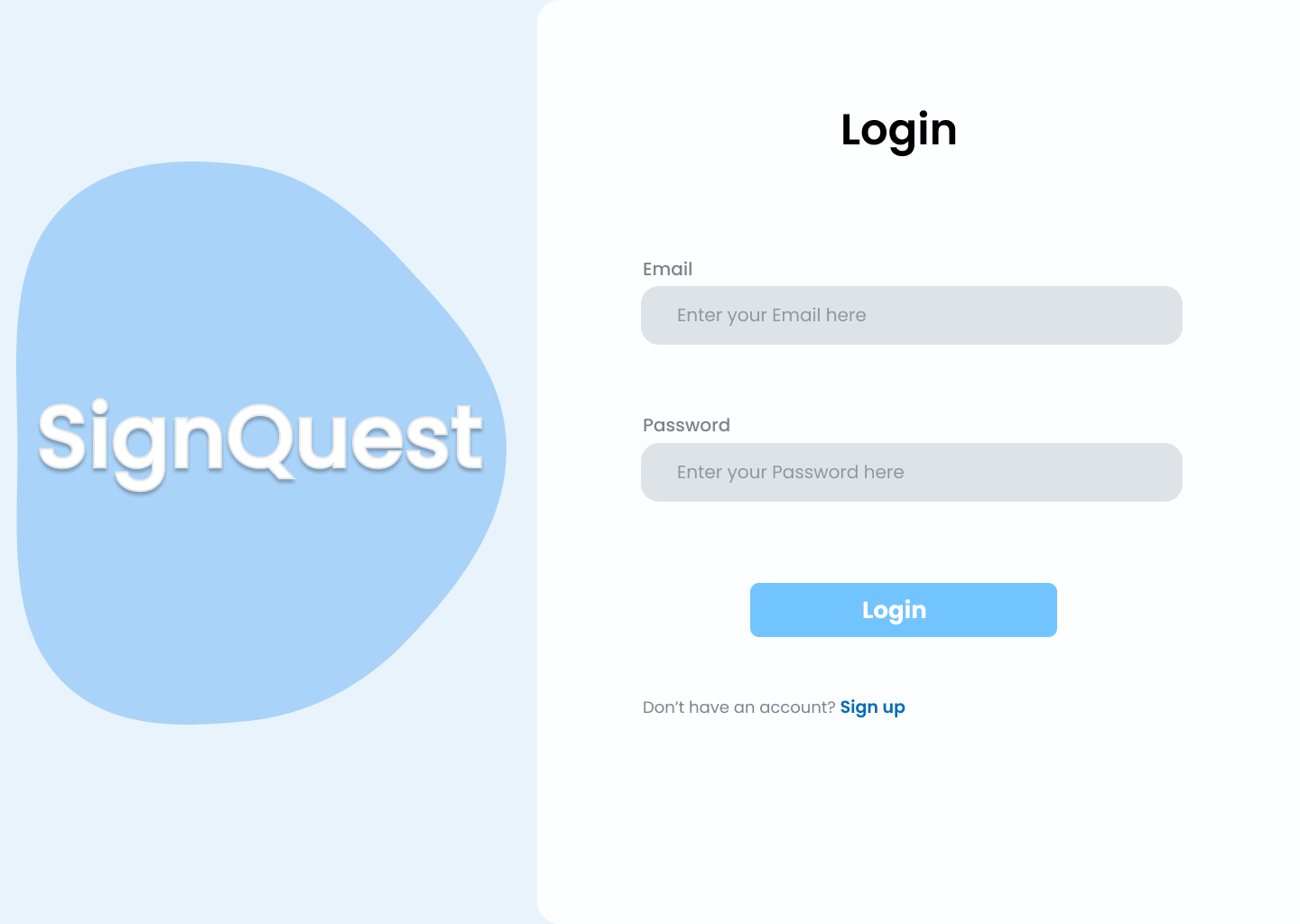
**Start a lesson** - At the beginning of each lesson an instruction video will appear that will include a new gesture performed by a signer  
  
**View Progress -** Each user will have a progress that will show him the percentage of completed lessons ,new learned words and the scores of completed lessons.

**4.2.3 Activity Diagram**

*figure 7: activity diagram*

The diagram describes a typical lesson process.  
The process starts by clicking start lesson, then the system displays an instruction video on how to perform the gesture.  
Using the user’s webcam, he performs the gesture and the system extracts his landmarks and passes it to the model that predicts the word, on success the user moves to the next question until the lesson is over.  
  
  
**4.2.4 User Interface**

**4.2.4.1 login screen**

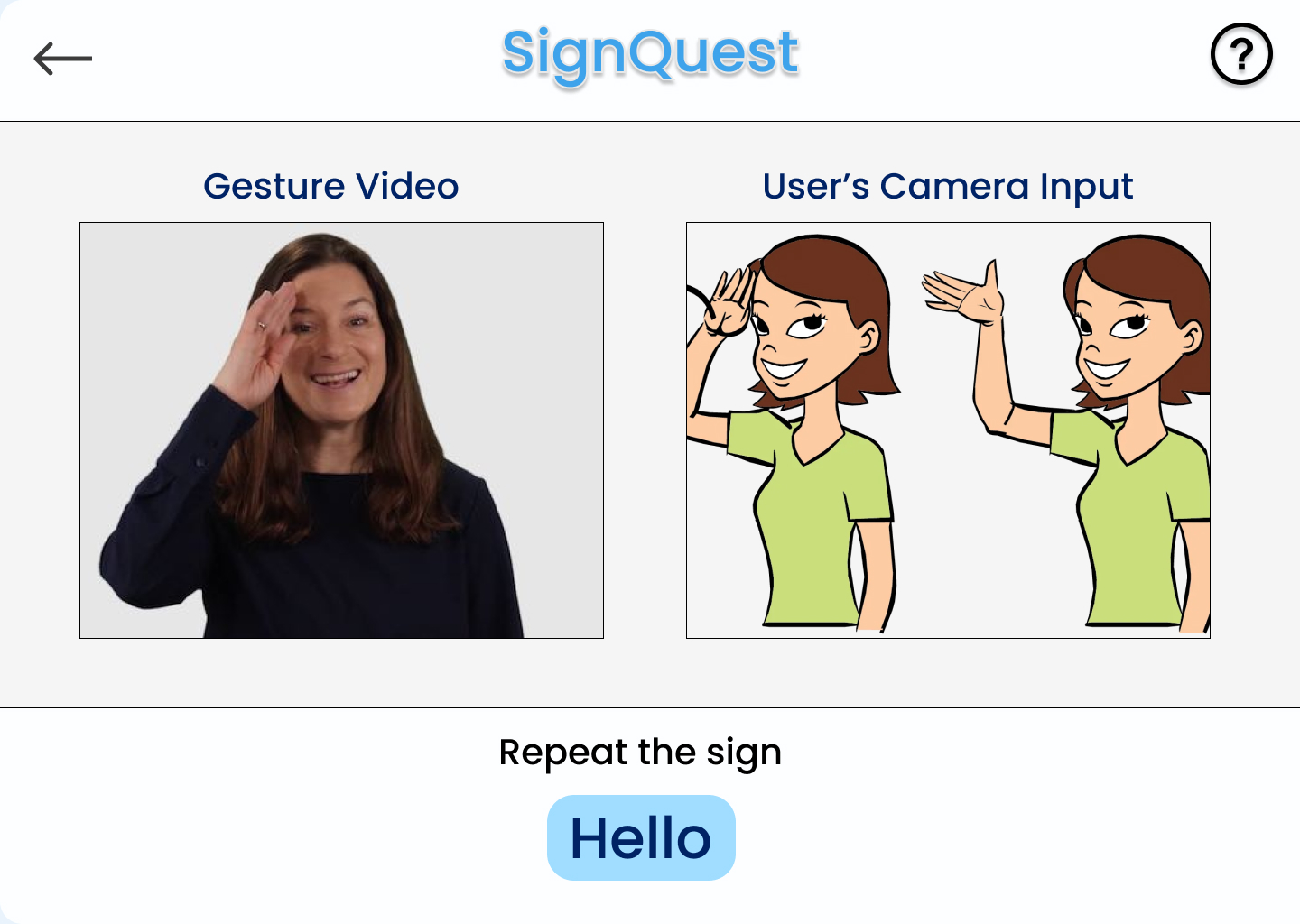
*****figure 8: login screen*

**4.2.4.2 Homepage**

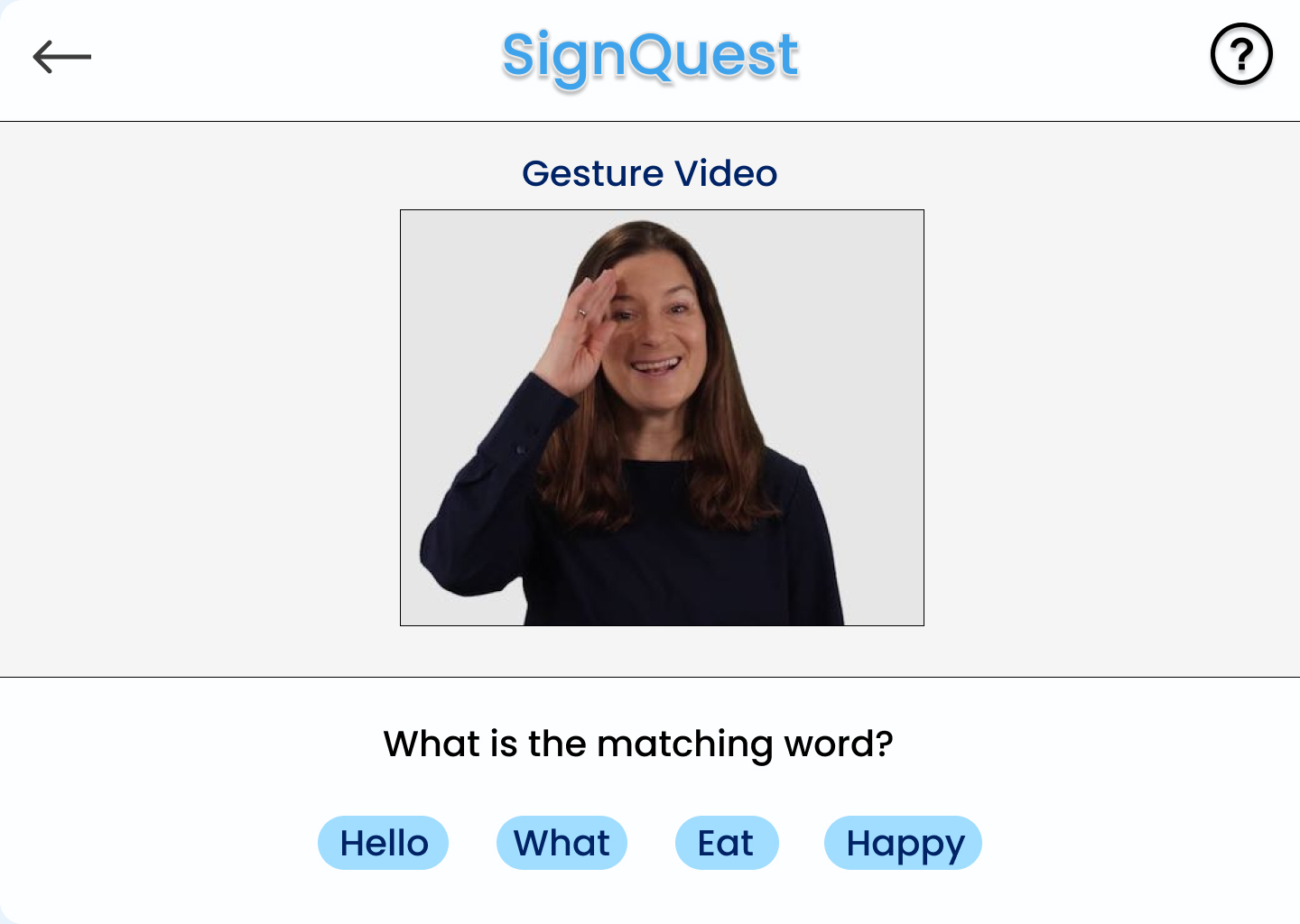
*figure 9: homepage*

After Logging in, The user is redirected to his Homepage, where he can choose a lesson to start.

**4.2.4.3 Exercise - Repeat a Gesture**

  
*figure 10: Repeat a gesture exercise*

The user is shown a video of an ASL word gesture and is asked to repeat it in front of the camera.  
**4.2.4.4 Exercise- Choose The Correct Word**

*  
figure 11: choose correct word exercise*The user is shown a video of an ASL gesture and is asked to choose the matching word.

# 5. Evaluation / Verification Plan

## 5.1 Evaluation

Our main goal is to create an app that will be able to teach sign language for anyone who wants to, whether they're deaf or not.  
  
The evaluation will consist of a few parts.  
In the first part, we will evaluate our model based on a few criterias:

* the model’s accuracy of correctly identifying an ASL gesture .
* The speed of execution in real-time.

Our goal in this part is to train an accurate model that will detect and classify a live gesture in a few seconds. The dataset will be split to train and test and our goal is accuracy of at least 77% [[1](#bn2pgmjwjl5i)].   
  
In the second part, we will evaluate the app’s ability to teach ASL based on 2 criterias:

* the ability of the user to learn ASL words- we’ll evaluate it based on the score the users get in the different lessons. The goal is to achieve an average lesson score of 75%.
* percentage of user satisfaction- our goal is to achieve at least 85% satisfaction rate based on the average of the user’s feedbacks rating
* user reviews- in addition to average user rating, we will address the reviews in the user’s feedback and make improvements if needed

Our goal in this part is to create an educational app that teaches the ASL efficiently and with a user-friendly interface.

## 5.2 System Requirements

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| **1** | The system enables the user signing up with name,email and password |
| **2** | The system enables the user signing in with email and password |
| **3** | The system enables the user to update name and password |
| **4** | The system enables the user to start a lesson |
| **5** | The system enables the user answering a question by typing, choosing an answer or signing in front of a webcam |
| **6** | The system enables the user to view his progress, which will include the percentage of completed units, lessons,words and the scores of completed lessons. |
| **7** | The system will allow the user watching an instruction video that will be displayed in the beginning of a lesson |
| **8** | The system enables the user to give a feedback for the entire app experience |
| **9** | The system enables the admin adding/editing an exercise, stored information is type, the correct answer, video link and list of possible answers |
| **10** | The system enables the admin viewing the users’ feedbacks that includes email, rating and review |
| **11** | The system enables recognizing sign language gestures in real time through web camera |

## 5.3 Testing

|  |  |  |
| --- | --- | --- |
| **ID** | **Action** | **Expected result** |
| **1** | Signing up | 1. Success message 2. New user created in the database. 3. page redirects to login |
| **2** | Signing in | 1. Success message 2. File with user credentials created 3. page redirects to homepage |
| **3** | User updating his profile | User profile is updated with the new information. |
| **4** | User viewing homepage | 1. showing all available lessons and units 2. showing toolbar |
| **5** | User answering a question | 1. giving a feedback according to the answer 2. moving to the next question |
| **6** | User viewing his progress | Correctly shows the user's progress: completed units,lessons ,words and scores of completed lessons. |
| **7** | User watching an instruction video | Correctly shows instruction video corresponding to the word |
| **8** | User submit a feedback | 1. Success message 2. New feedback is added to the DB with a rating and review |
| **9** | Admin adding an exercise | 1. Success message 2. New exercise is added to the DB with a type, correct answer, video link and list of possible answers |
| **10** | Admin editing an exercise | 1. Success message 2. the new exercise information updated in the DB with a type, correct answer, video link and list of possible answers |
| **11** | Admin viewing user’s feedback | All user’s feedback will be shown, with review, rating and date. |
| **12** | User performs sign language gestures | 1. accurately recognizing users’ performed gesture 2. showing feedback to the user |

# 6. References

1. Bansal, D., Ravi, P., So, M., Agrawal, P., Chadha, I., Murugappan, G., & Duke, C. (2021). CopyCat: Using Sign Language Recognition to Help Deaf Children Acquire Language Skills. <https://doi.org/10.1145/3411763.3451523>
2. Duarte, A., Palaskar, S., Ventura, L., Ghadiyaram, D., DeHaan, K., Metze, F., Torres, J., & Giró-I-Nieto, X. (2020). How2Sign: a large-scale multimodal dataset for continuous American sign language. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2008.08143>
3. Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. Convolutional Neural Networks for Visual Recognition, 2(225-232), 8.‏
4. Huang, J., Zhou, W., Li, H., & Li, W. (2015, June). Sign language recognition using 3d convolutional neural networks. In *2015 IEEE international conference on multimedia and expo (ICME)* (pp. 1-6). IEEE.‏
5. Li, D., Opazo, C. R., Yu, X., & Li, H. (2019). Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1910.11006>
6. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019, June). Mediapipe: A framework for perceiving and processing reality. In *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)* (Vol. 2019).‏
7. Moryossef, A. (2023). Sign.MT: Real-Time Multilingual sign language translation application. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2310.05064>
8. Shaw, A., Wünsche, B. C., Mariono, K., Ranveer, A., Xiao, M., Hajika, R., & Liu, Y. (2023). JengASL: A Gamified approach to sign language learning in VR. *Journal of WSCG*, *31*(1–2), 34–42. <https://doi.org/10.24132/jwscg.2023.4>
9. Sundar, B., & Bagyammal, T. (2022). American sign language recognition for alphabets using MediaPipe and LSTM. *Procedia Computer Science*, *215*, 642-651.‏