Artificial Intelligence Department

Faculty of Computers and Artificial Intelligence

Cairo University

Sign Language Translator

Under Supervision of:

Dr. Amin Allam

Prepared by:

Reem Ahmed Khalil 20201075

Academic Year

2023/2024

Table of Contents

List of Illustrations

Abstract

Sign language translators play a crucial role in enhancing communication accessibility for individuals who use sign languages. However, these applications face several challenges in meeting the expectations and needs of their users.

One primary challenge revolves around ensuring fair and equal accessibility for deaf individuals in today's programs and services. Additionally, users find it quite hard to find one translator that translates both ways with accurate and fast translations.

Sign language translator solves those problems by providing a user friendly interface that translates between sign language and text/voice bi-directionally, which makes the translation process easy and fast both ways. Additionally, Sign language translator includes a ChatGPT that is designed especially for deaf users to access its services with their own native language to make them feel more included.

Introduction

Problem

In a world where communication is predominantly verbal, deaf individuals face barriers in interacting seamlessly with the hearing population, and vice versa. This project addresses this challenge by introducing a revolutionary communication platform designed to foster inclusivity. The main goal is to make communication more fluid and natural between deaf and hearing individuals. By seamlessly integrating text, spoken language, and sign language, this technology fosters an environment where conversations flow effortlessly, creating a more connected and inclusive world for everyone.

The incorporation of ChatGPT to facilitate communication in sign language serves a profound purpose. By allowing deaf individuals to interact with ChatGPT using their native sign language and receiving responses in the same medium, we aim to convey a powerful message of inclusivity and recognition.

This approach not only breaks down communication barriers but also reinforces the idea that every individual, regardless of their hearing abilities, is valued and considered in the realm of technology. Beyond empowering the deaf community, it's about making technology a tool that enhances communication, ensuring that it is not just accessible but genuinely responsive to the diverse needs of its users.

Objective

By the end of this project a sign language translation program would be designed and developed based on computer vision, NLP and machine learning models to address the current challenges facing users such as the difficult communication with sign language speakers and sign language speakers not being included in today's services.

The platform will be designed with a well-structured interface to allow users to easily navigate between different services such as bi-directional sign language translations and communication with sign language chatGPT. By addressing these main points, the platform aims to improve the overall communication experience for users and become a competitive player in the industry.

Solution

The solution to the communication barriers between deaf and hearing individuals lies in the creation of an innovative communication platform. This platform seamlessly integrates text, spoken language, and sign language to make interactions more fluid and natural. By providing a user-friendly interface that accommodates multiple communication modes, the technology ensures that conversations can effortlessly flow between deaf and hearing users.

the integration of ChatGPT, a sophisticated language model, to enable communication in sign language. Deaf users can express themselves using their native sign language, and ChatGPT responds in kind.

Additionally, the platform's adaptability allows hearing users to engage in conversations through text or spoken language, promoting a two-way communication channel that caters to the diverse needs of both communities.

This comprehensive solution is designed not just to address a problem but to transform the way we perceive and experience communication, fostering a world where everyone can connect, communicate, and be heard.

Scope

The scope of this project is to create a Sign Language Translator (SLT) application, enabling bidirectional translation between sign language and written/spoken language. The system will also create a ChatGPT interface which will be designed to recognize the sign language queries and respond with an avatar using sign language.

The development of the system will be thoroughly documented, starting with an introduction to the stakeholders and user analysis. This will be followed by a description of the system analysis and design process, including the identification of functional and non-functional requirements and the creation of design diagrams. The tools used in the development process will also be outlined, and detailed explanations of the main features of the platform and the models used to implement them will also be provided.

Methodology

For the development of a Sign Language Translator (SLT), an iterative and agile methodology is suitable. Given the dynamic nature of language, the need for constant user feedback, and the potential for evolving requirements, an agile approach allows for flexibility and responsiveness throughout the development process.

1- Requirements Gathering: Initial requirements are gathered, but the agile process allows for ongoing refinement and addition of requirements as the project progresses.
2- Analysis and Design: These phases are concurrent and iterative. As new requirements emerge or existing ones evolve, the design is adjusted accordingly, allowing for continuous improvement.
3- Implementation: Incremental development of the SLT occurs, with regular delivery of functional components. This allows stakeholders to see tangible progress and provide feedback early in the process.
4- Testing: Continuous testing is integrated throughout the development cycle, ensuring that each increment is thoroughly tested. This agile approach promotes early detection and resolution of issues.
5- Feedback and Iteration: Regular reviews and feedback sessions with users and stakeholders drive iterative improvements. Changes can be made swiftly, ensuring that the SLT aligns closely with user needs.
6- Maintenance: Ongoing maintenance and enhancements can be seamlessly integrated into the development process, addressing bugs and incorporating user-driven modifications without disrupting the overall project timeline.

Competitors

App	Free	Fast	Multi-language	Avatar	Text to sign	Sign to text	Sentences	chatGPT
Pocket Sign	✓	✓	×	×	✓	×	×	×
Sign Translate	✓	✓	✓	×	✓	×	✓	×
Mimix	✓	✓	×	✓	✓	×	✓	×
Prodeaf	✓	✓	✓	✓	✓	×	✓	×
Hand Talk	✓	✓	✓	✓	✓	×	✓	×
Motionsavy	×	✓	×	×	✓	✓	✓	×
Signfy (ours)	✓	✓	×	✓	✓	✓	✓	✓

Table 1 : competitors

User Centered Analysis

Functional Requirements

1- The customer creates an account by entering his Name, Password, Email, Phone, and Default Location.
2- The user will be able to translate his sign language to text/voice by entering his sign language gestures as a live video and getting a text which can be heard as a voice.
3- The user will be able to translate his text/voice to sign language by entering his text/voice and getting an avatar the translated as sign language
4- The user will be able to communicate with a ChatGPT interface that understands and responds to inquiries made in sign language.
5- The user have the ability to provide feedback on translations for continuous improvement.
6- The user will be able to access a history of the translations he did.
7- The user will be able to reply the translation that he received.

Non Functional Requirements

1- Usability: we aim to have a user-friendly system with easy-to-use features that facilitate the communication between Deaf and speaking user and using ChatGPT by Deaf user. To ensure a maximum usability we are carrying on a solid development process from wire framing to the final deliverable.
2- Compatibility: our system will be provided as an application.
3- Reliability: the system should be reliable enough to do all its functions smoothly and operate in a defined environment. Overall user experience would be consistent and predictable.
4- Performance: our system is descent response time despite the highly complex tasks it does.

Project Stakeholder

1- Stakeholders are all individuals or groups who have an active stake in the project and can potentially impact, either positively or negatively, its development during its lifecycle.

2- **Non-Verbal Individuals**: People who can't speak due to various reasons, such as speech impediments, medical conditions, or other communication challenges. They are direct beneficiaries of the program, relying on it to express themselves effectively using sign language.

3- **Verbal Individuals**: Individuals who can speak but may not have proficiency in sign language. They use the program to translate their spoken words into sign language, facilitating communication with non-verbal individuals. This group includes family members, friends, and professionals interacting with the non-verbal community.

4- **Developers**: The technical team responsible for developing and maintaining the software are vital stakeholders. Their expertise ensures the program's functionality, security, and continuous improvement to meet the evolving needs of users.

Time plan

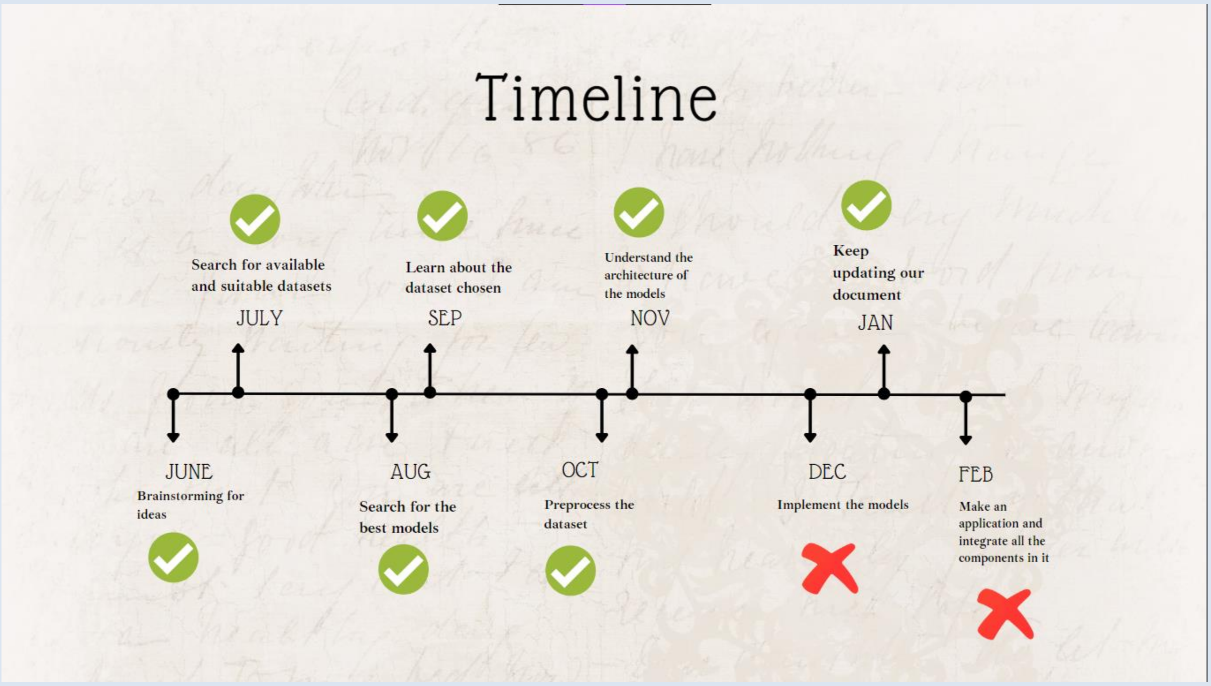

Figure 1 : timeline

System Analysis & Design

Use case diagram

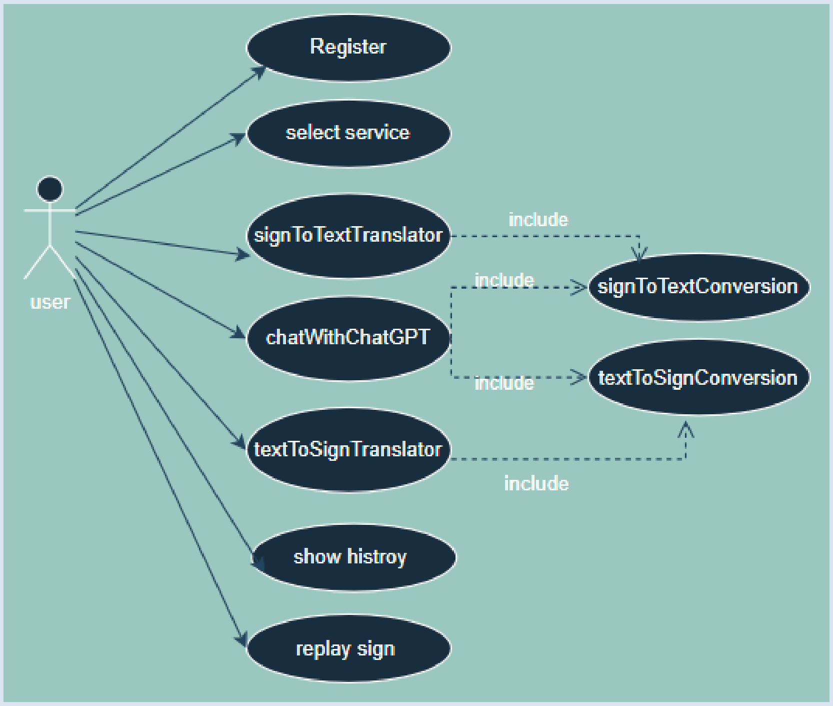

Figure 2 : use case diagram

User Stories & Prototype

➢ User Story #1

User Story ID	**US #1**
User Story Name	Registration / login
Actors	User
Description	**As** a user **I like** to be able to have an account or to reset my password so that my data will be saved in system
Acceptance Criteria	**Given** I'm logged-out and I'm on the Sign-In page When I fill in the "Email" and "Password" fields with my authentication credentials and I click the Sign-In button in case I already have an account or I register instead. Then the system signs me in. Given I can't remember my password

	When I try to log in **Then** the system will send a link with a new one to reset the password and save it

Normal Scenario

Actor Action	System Response
1-User Enters his email and Password. **2-Click log in**	
	3- System Verifies user data 4- System redirects to home page of user
5-Click Sign up	
	6-System displays a form for the user to fill
7- User enters [Email, Password]	
	8-System verifies the input data 9- System redirects to home page of user
10- Clicks Forgot your password	
	11- System displays a form for the user to enter his email.
12- User enters his email and Click Submit	
	13- System sends a link to reset his password.
14- User enters his new password that was sent to him and click Submit	
	15- System verifies the new password 16- System redirects to home page of user

Exceptional Scenario

Actor Action	System Response
1-User Enters name and Password. **2- Clicks log in**	

	3- Email is invalid 4- System rejects signing in and displays an error message for a wrong email “this email does not exist, you have to sign up”
5-Clicks Sign up	
	6-System displays a form for the user to fill
7- User enters [Email, Password]	
	8- Email is invalid 9- System rejects registration and displays an error message for invalid email.
10- User re-enters an email	
	11- System rejects registration and displays an error message for missing password
12- User enters a password	
	13- System creates a new account for the user

Exceptional Scenario

Actor Action	System Response
1-User Enters name and Password. **2- Clicks Sign in**	
	3- Password is invalid 4- System rejects signing in and displays an error message for a wrong password
5-Clicks forgot password	
	6-System displays a form for the user to fill
7- User enters [Email]	
	8- System sends a form in his email to renew his password
9- User renew his password	

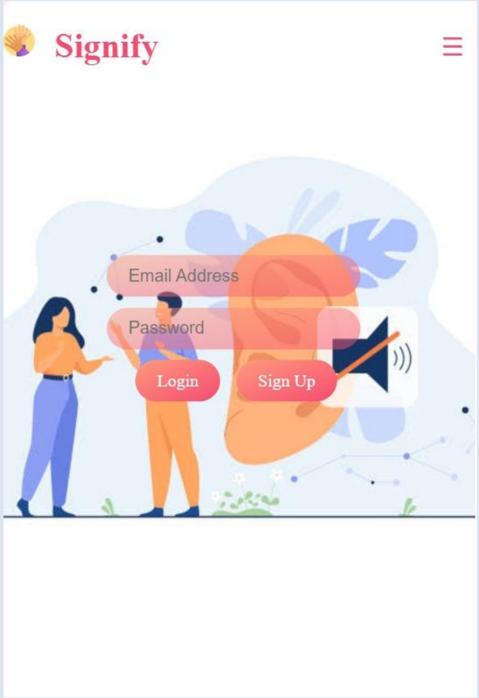

Figure 3 : registration interface

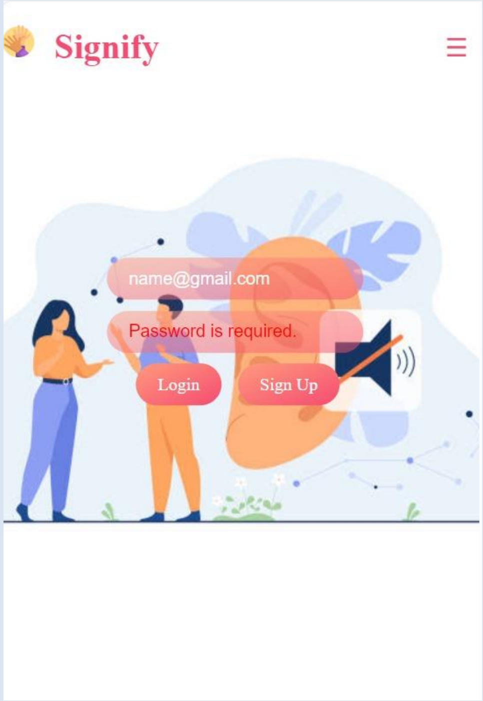

Figure 4 : missing field interface

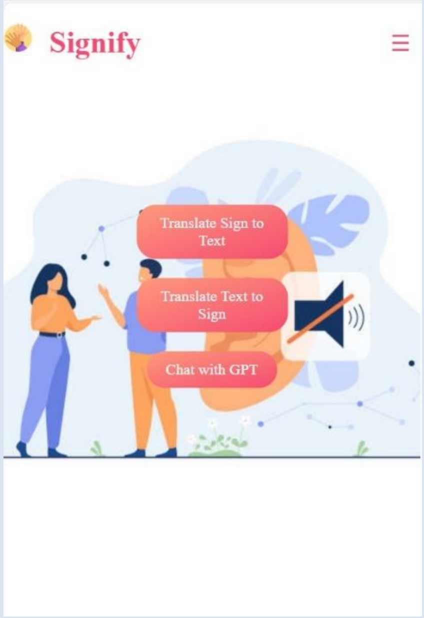

Figure 5 : choose your service interface

➢ User Story #2:

User Story ID	US #2
User Story Name	Sign Language to text and voice Translation
Actors	User
Description	As a user, **I** want to utilize the Sign Language Translator (SLT) for accurate recognition from sign language to written/spoken language
Acceptance Criteria	**Given I** want my signs to be translated **When I** choose the service of sign to text translation **Then** the system accurately translates and displays the corresponding sign language to text or voice

Normal Scenario

Actor Action	System Response
1-user log in/sign up	
	2-system shows three options for user one for translating sign language to text/voice and one for translating text/voice to sign language and one for chatting with chatGPT
3-user chooses the service of translation **4-user inputs his sign language gestures**	
	5-system translates the signs into text with an option to hear it as voice
6-user press the voice option	
	7-system plays the voice translation

Exceptional Scenario

Actor Action	System Response
1-user registers	

	2-system shows three options for user one for translating sign language to text/voice and one for translating text/voice to sign language and one for using the chat bot
3-user chooses the service of translation from sign to text/voice **4-user inputs his sign language gestures**	
	5-system tells the user that there's no person detected
6-user adjust his position **7- user input his sign gestures**	
	8-system translates the signs into text with an option to hear it as voice
9-user press the voice option	
	10-system plays the voice translation

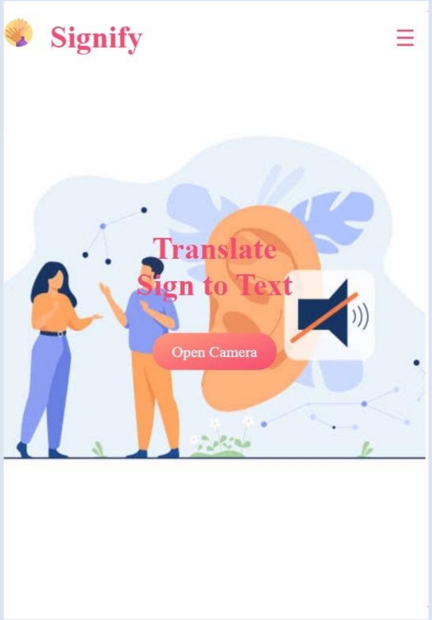

Figure 7 : sign to text translation interface

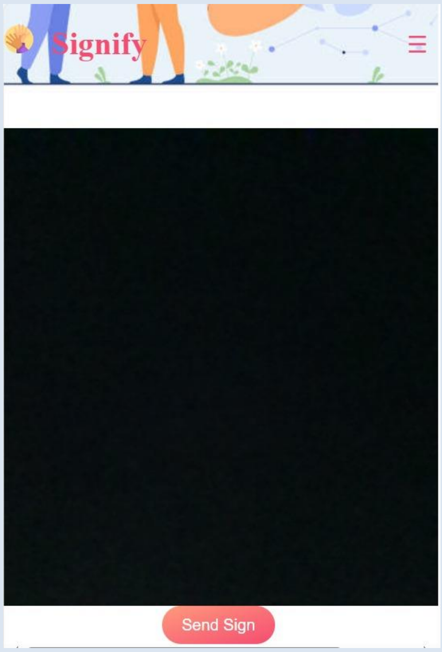

Figure 8 : open camera interface

➢ User Story #3:

User Story ID	US #3
User Story Name	Chatting with Sign Language ChatGPT

Actors	User
Description	As a user, **I** want to engage in a chat with the Sign Language ChatGPT, using sign language for inquiries and receiving responses in sign language.
Acceptance Criteria	**Given** I want to chat with chatGPT **When** I choose the chat service with the Sign Language ChatGPT **Then** the system provides an interface for inputting sign language gestures and outputs responses for my questions in sign language**.**

Normal Scenario

Actor Action	System Response
1-user registers	
	2-system shows three options for user one for translating sign language to text/voice and one for translating text/voice to sign language and one for using the chat bot
3-user chooses the service of chatGPT **4-user inputs his sign language gestures**	
	5-system respond to the user question with an avatar doing sign language
6-user chooses a new answer	
	7-system gives a new answer

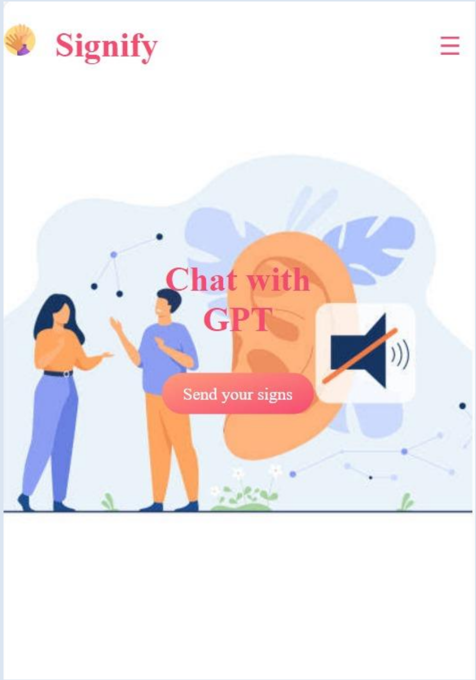

Figure 9 : chatGPT interface

➢ User Story #4:

User Story ID	US #4
User Story Name	text or voice to sign language Translation
Actors	Non-deaf person
Description	As a user, **I** want to utilize the Sign Language Translator (SLT) for accurate translation from written/spoken to sign language
Acceptance Criteria	**Given** I want my text or voice to be translated **When I** choose the service of translation **Then** the system accurately translates and displays the corresponding sign to the text or voice

Normal Scenario

Actor Action	System Response

1-user registers	
	2-system shows three options for user one for translating sign language to text/voice and one for translating text/voice to sign language and one for using the chat bot
3-user chooses the service of translation of voice or text to sign language	
	4-system has an input window with an option of making a voice input
5-user inputs his text or voice	
	6-system translates the voice or text to sign language and display the translation by an avatar

Exceptional Scenario

Actor Action	System Response
1-user registers	
	2-system shows three options for user one for translating sign language to text/voice and one for translating text/voice to sign language and one for using the chat bot
3-user chooses the service of translation voice to text to sign language	
	4-system has an input window with an option of making a voice input
5-user inputs his text or voice	
	6-system shows the user that the input is not valid and tell user to try again
7-user inputs his text or voice again	
	8-system translates the voice or text to sign language and display the translation as an avatar

Figure 10 : text to sign translation interface

Entity Relationship Diagram (ERD)

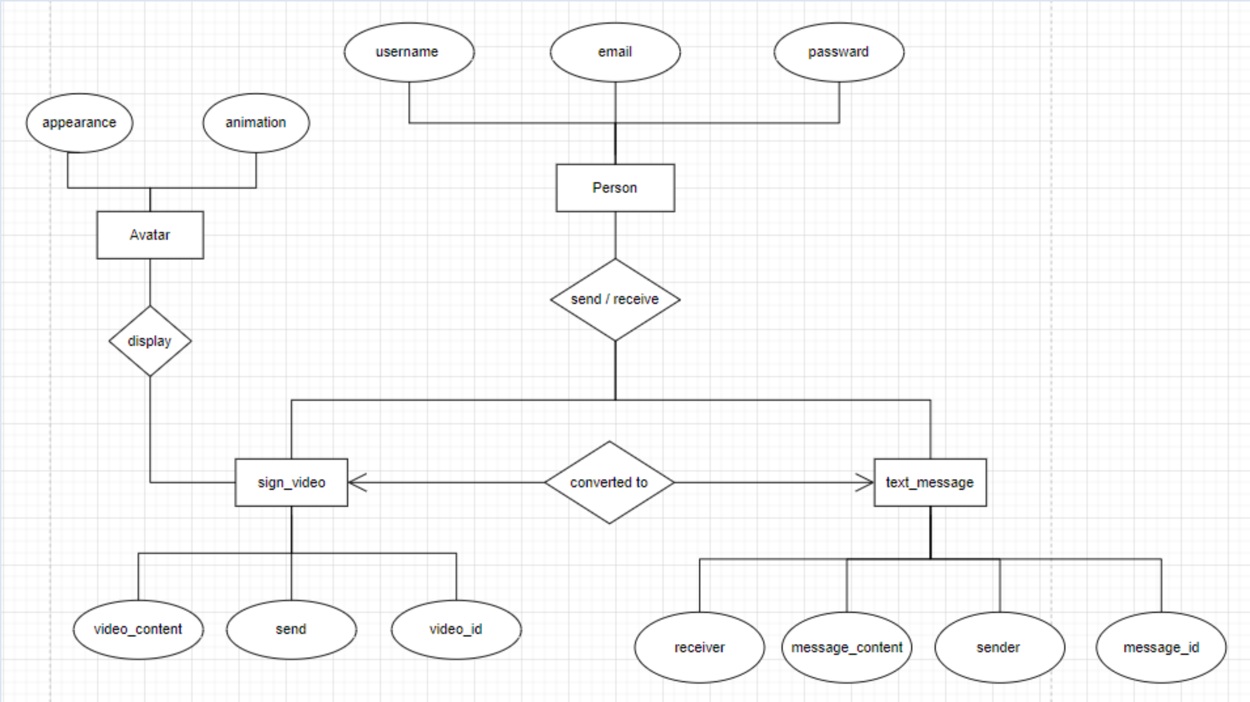

Figure 11 : ERD

Class Diagram

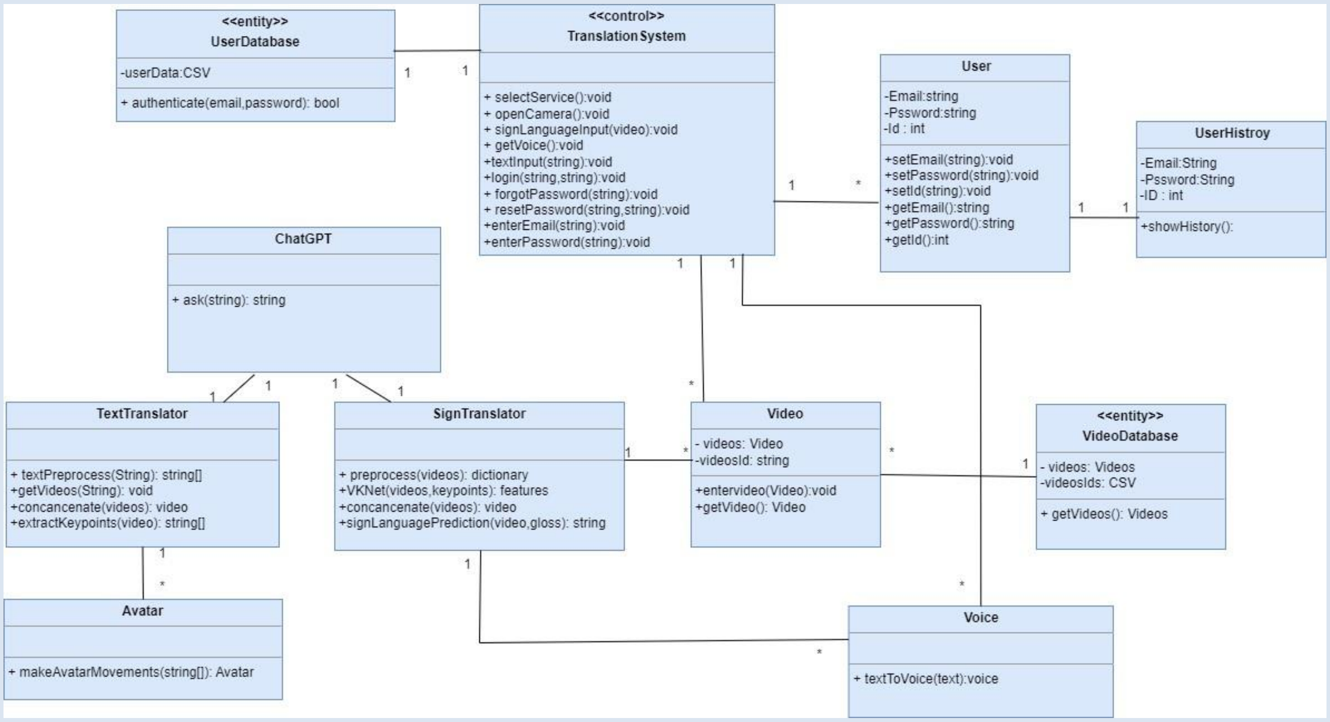

Figure 12 : class diagram

Sequence Diagram

1- Registeration

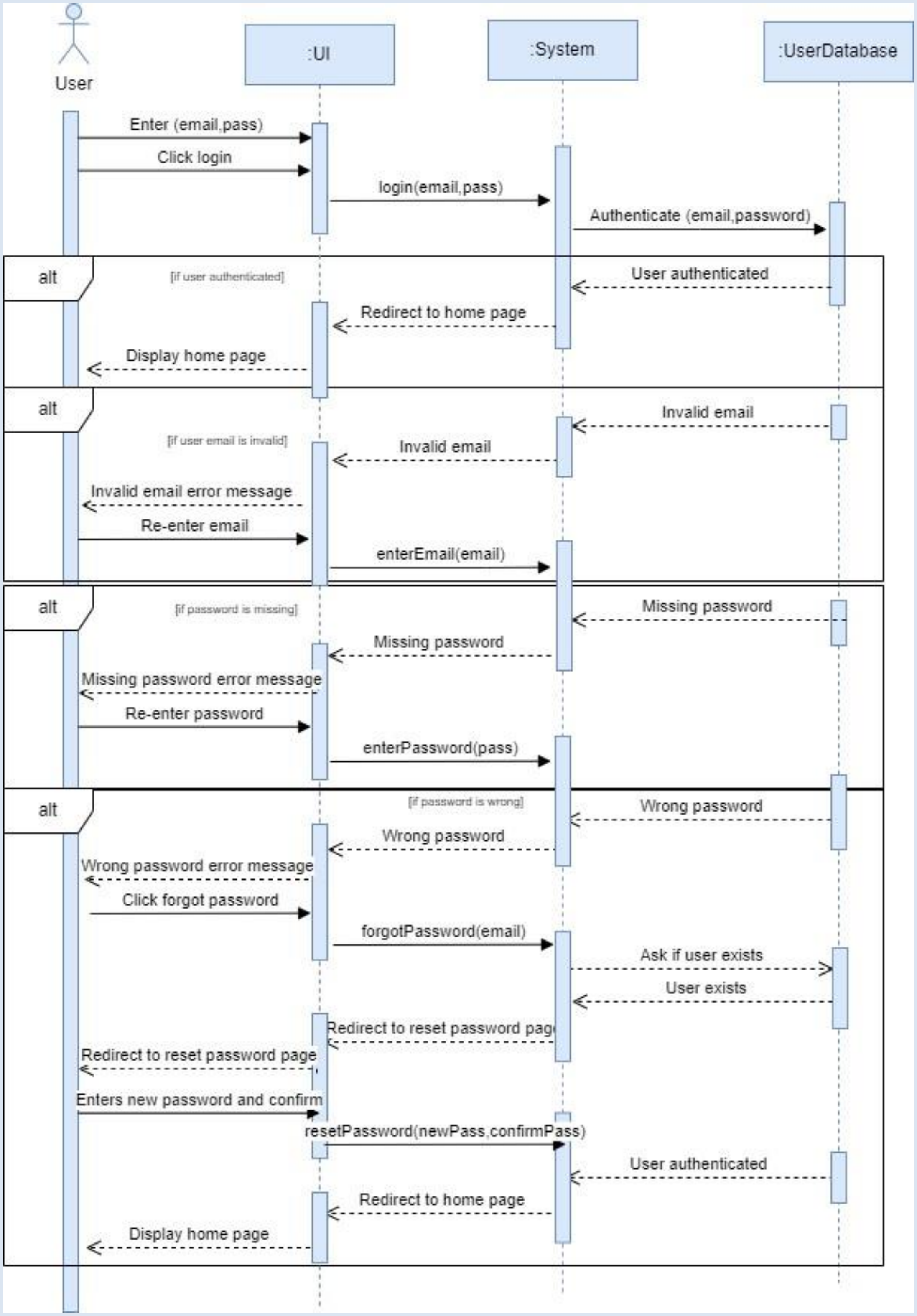

Figure 13 : register sequence diagram

2- Sign to text

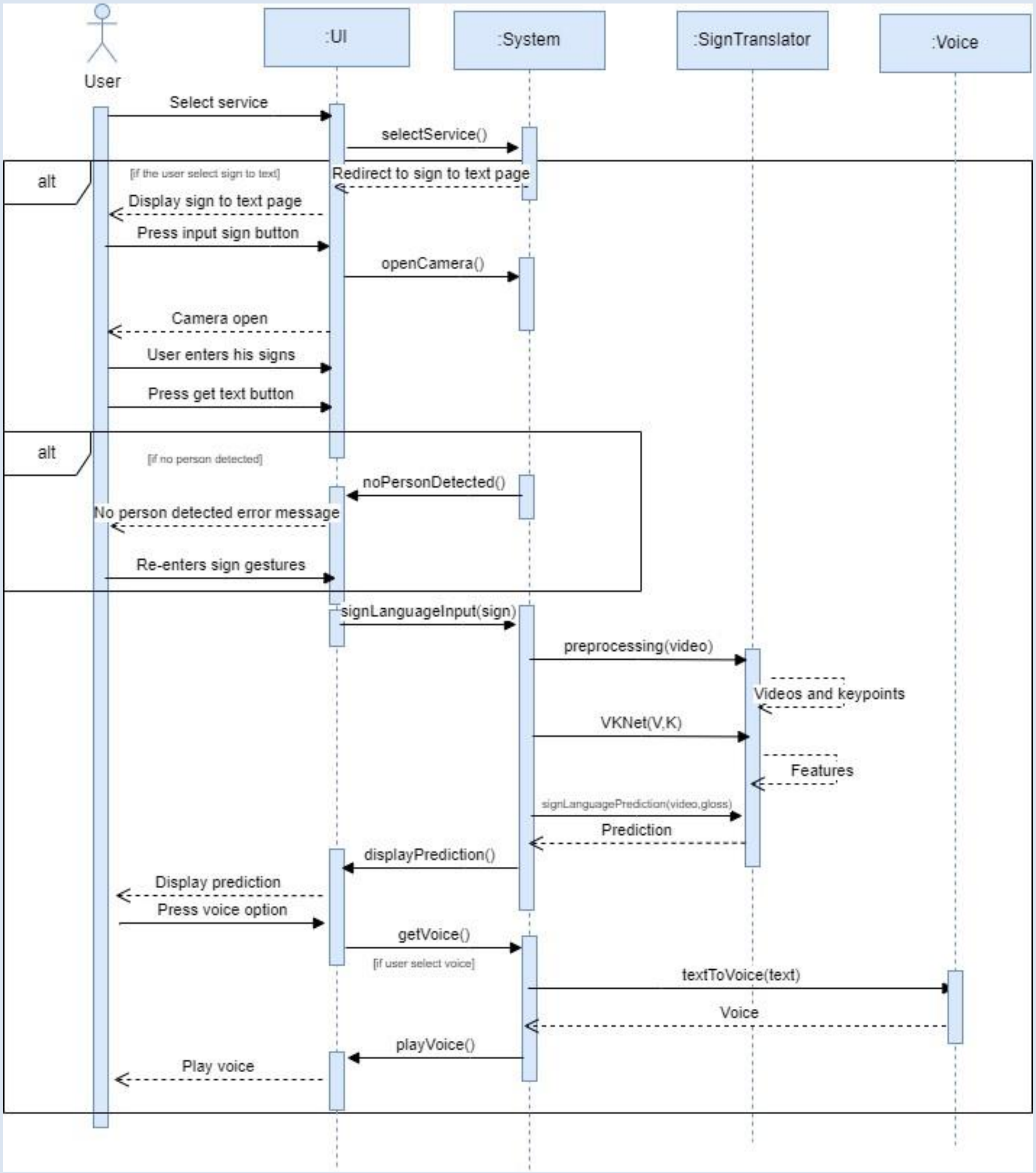

Figure 14 : sign to text sequence

3- Text to sign

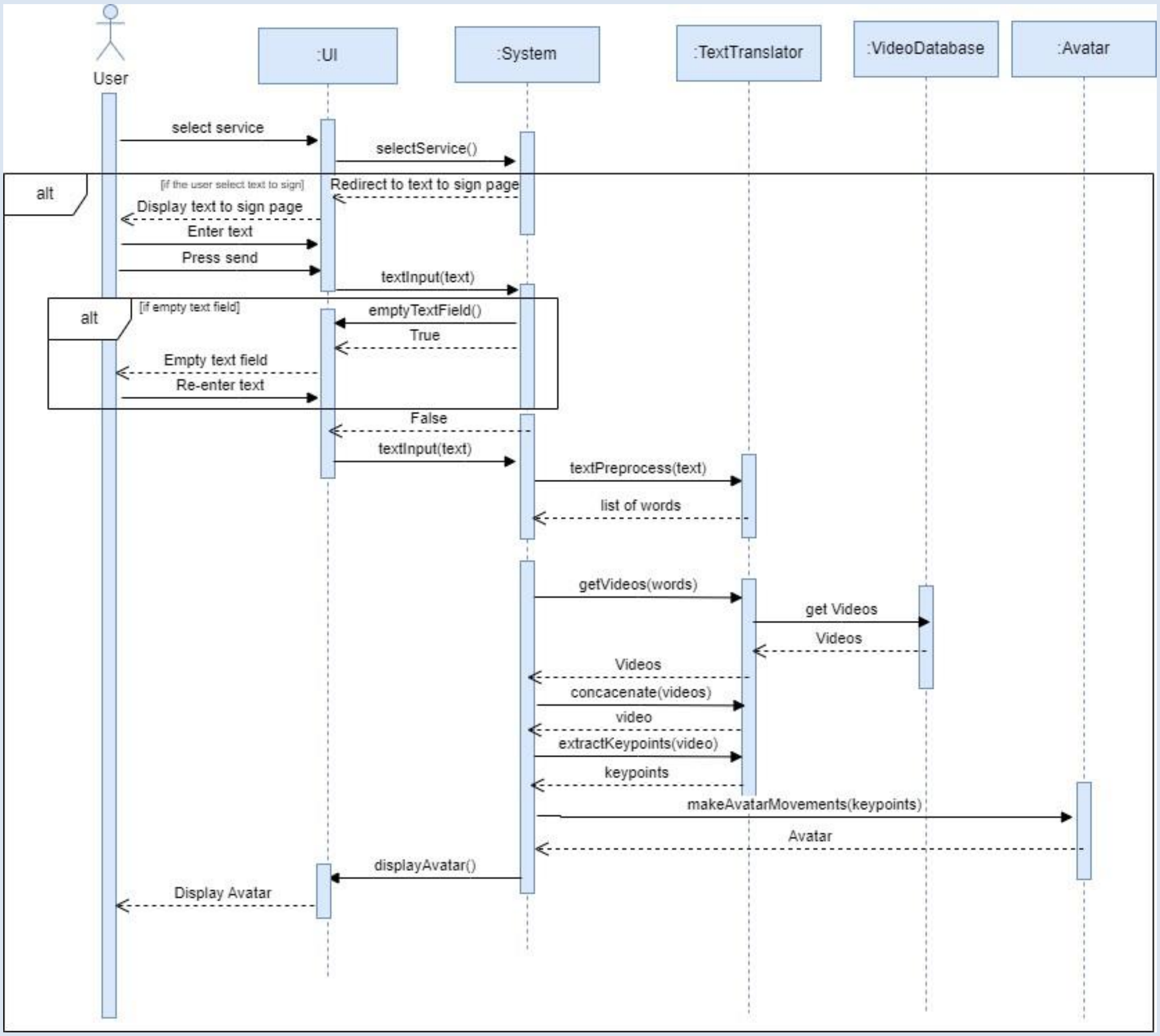

Figure 15 : text to sign sequence

4- Sign chatGPT

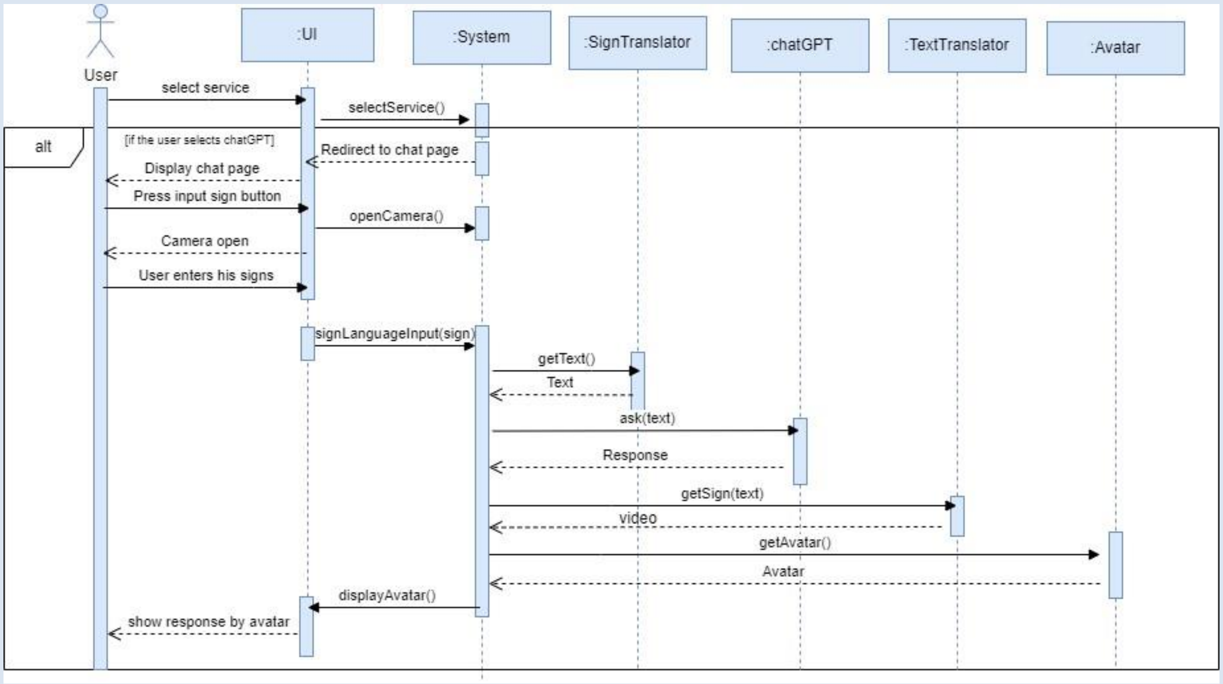

Figure 16 : sign chatGPT sequence

System Architecture

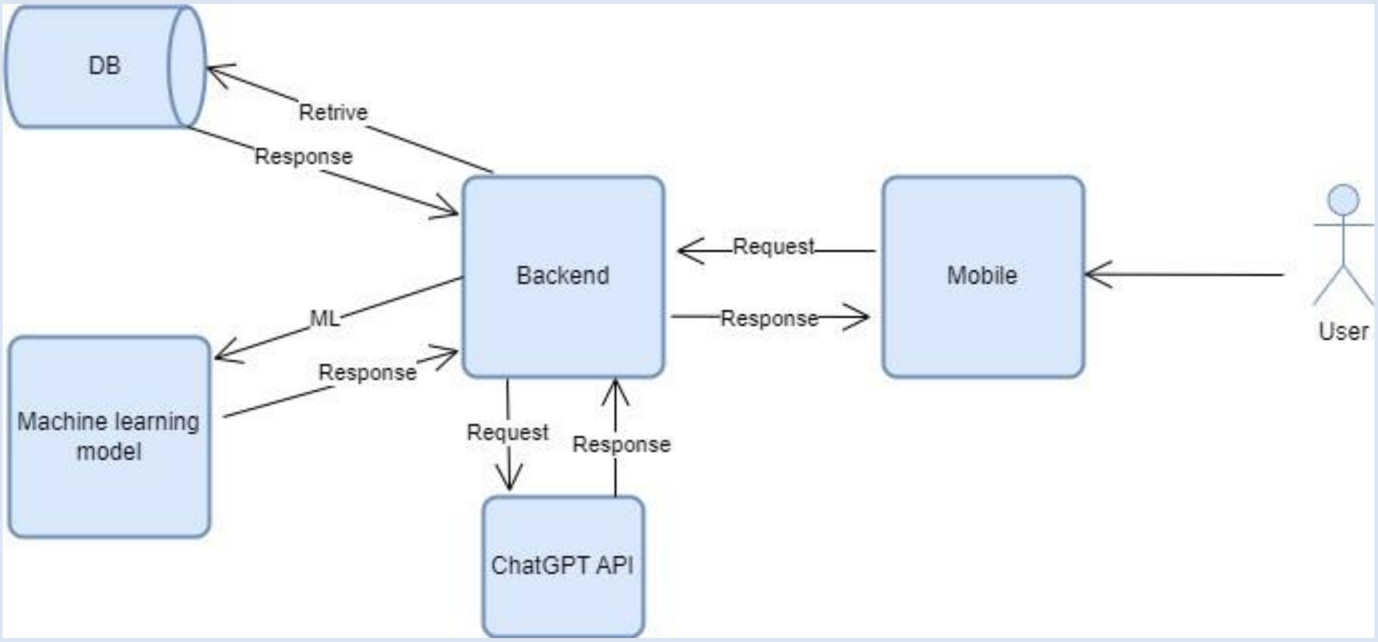

Figure 17 : system architecture

Implementation

Dataset

DATASET	# SIGNERS	# VIDEOS	# GLOSSES
WLASL-100	97	2,038	100
WLASL-300	109	5,117	300
WLASL-1000	116	13,168	1,000
WLASL-2000	**119**	**21,083**	**2,000**

Table 2 : WLASL dataset description

WLASL is an ASL dataset with a vocabulary of 2,000 words and 21,083 samples. It releases WLASL100 and WLASL300 as its subsets. WLASL is collected from Web videos and bring new challenges due to unconstrained reallife recording conditions.
We utilize the WLASL2000 dataset for both our text-to-sign and sign-to-text translators. However, it's important to note that while the dataset comprises 21k entries, only 12k of them are currently available for use.

Techniques

Text to sign model

A majority of prior works on Spoken2Sign translation focused on expressing translation outcomes through keypoints. However, the keypoint representations often pose interpretable challenges for signers. With the evolution of generative models, several studies have employed these keypoints to animate sign images, subsequently creating a sign video. However, the 2D video format is prone to blurriness and visual distortions. In this work, we introduce an innovative method for Spoken2Sign translation by utilizing a 3D avatar to represent the translation results. In contrast to earlier attempts that utilized generative models ,our method prioritizes enhancing understandability and incorporates a 3D human pose prior with a special emphasis on signing poses, allowing for multi-view representations of the translation results

Our Spoken2Sign translation pipeline primarily consists of three components: 1) a Text2Gloss translator that translates the input text into a gloss sequence; 2) a sign connector, which stitches two adjacent signs together; and 3) an avatar producer, which produces the final animated sign avatar.

1- Text2Gloss Translator:

In the text gloss generator, we split the input sentence into words, do some preprocessing to these words such as removing punctuations and correcting grammatical errors, then we check if a word doesn’t exist in the dataset we split it into letters, and then for each word (gloss) we get its corresponding video from the dataset.

2- Sign connector:

In the sign connector, we take the keypoints coming from the translator and stitch them together into one.

3- Keypoints extractor:

In the keypoints extractor, we take the video coming from the connector and extract the keypoints from it and save in JSON file

4- Avatar producer:

In the avatar producer, we utilize the single key points file coming from the extractor to generate the movements for the avatar.

Sign to text model

The framework used is NLA-SLR and it mainly consists of three parts: 1) data pre-processing which generates video-keypoint pairs as network inputs; 2) a video-keypoint network (VKNet) which takes video-keypoint pairs of various temporal receptive fields as inputs for vision feature extraction; 3) a head network containing a language-aware label smoothing branch and an inter-modality mixup branch.

1- Data pre-processing

- Input videos V are cropped and represented as a tensor with dimensions T * H * W * C where number of frames T = 64, spatial resolution H = W = 224, number of channels C = 3
- Keypoints in these videos are estimated using HRNet trained on COCO dataset resulting 63 keypoint for each video 42 out of them are for hands.
- Those keypoints are represented in a heatmap with dimensions H * W * K where H = W = 112, K = 63. Elements on the heatmaps are calculated using a Gussian function. We get those heatmaps for all videos then we stack them together in K with dimensions T * H * W * K.
- Original 64-frame video (V64) paired with its corresponding keypoint heatmap sequence (K64) are cropped into (V32) and (K32) then both pairs are fed into VKNet leading to more informative and discriminative features.

2- VKNet

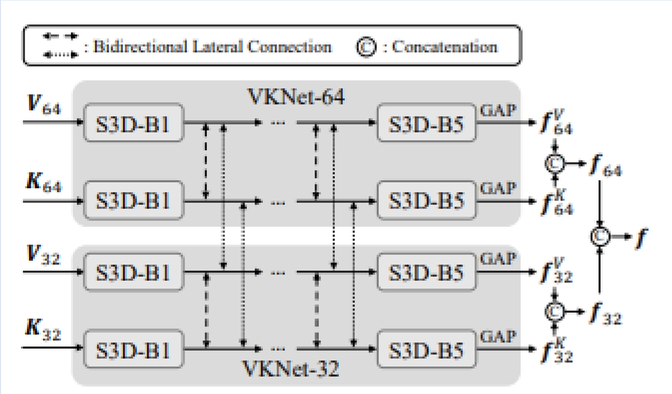

Figure 18 : VKNet architecture

- VKNet is composed of two sub-networks, namely VKNet-32 and VKNet-64, which take (V32, K32) and (V64, K64) as inputs, respectively
- Both have the same architecture which consists of a video encoder and a keypoints encoder, since we use heatmaps for keypoints representation, we use a CNN for extracting the feature out of it.
- The CNN used is S3D due to its excellent accuracy-speed trade-off
- The features coming from both VK-32 and VK-64 are concatenated resulting the final features F.

3- Head network

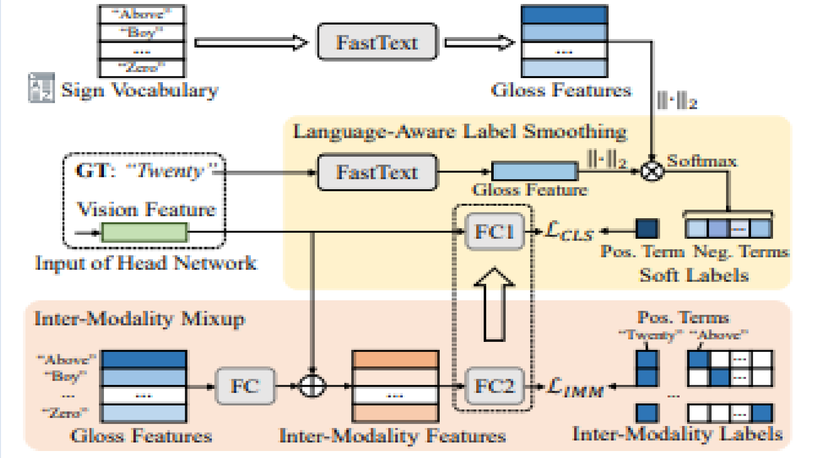

Figure 19 : head network architecture

- The head network consists of two parts a language aware smoothing and an intermodility mixup
- The language aware smoothing solves the problem of two signs that look the same and have similar semantic meanings by getting the

glosses features and computing similarities between those glosses then include them in the label calculation

- The inter modility mixup solves the problem of two signs that look the same and have different semantic meanings by adding our original features to glosses features to maximize signs separability in the latent space.

Method	**WLASL2000**				**WLASL1000**				**WLASL300**				**WLASL100**			
	P-I		P-C		P-I		P-C		P-I		P-C		P-I		P-C	
	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5
Pose-GRU									33	64			46	76		
Pose-TGCN	23	51							38	67			55	78		
GCN-BERT									42	71			60	83		
OpenHands	30															
PSLR									42	71			60	83		
I3D	32	57			47	76			56	79			65	84		
ST- GCN	34	66	32	65					44	73	45	73	50	79	51	79
Fusion-3	38	67			56	79			68	83			75	86		
BSL	46	79	44	78												
HMA	51	86	48	85												
TCK+									68	89	68	89	77	91	77	91
BEST	54	88	52	87					75	92	76	93	81	94	81	94
SIGNBERT+	54	87	52	86					74	91	75	91	82	94	83	95
SAM-SLR	58	91	55	90												
SAM-SLR V2	59	91	56	90												
NLA-SLR	**61**	**91**	**58**	**90**	75	94	75	94	86	97	87	97	92	96	93	97

Table 3 : comparison with previous works on WLASL

ChatGPT

1- We employ sign-to-text model to take the input sign and translate it into text.
2- We send this text to chatGPT to generate a response.
3- We employ text-to-sign model to translate the text coming from chatGPT into sign language represented by an avatar.

References

- Hand Talk App
- ASL Sign Language - start learning ASL now with our video lessons (pocketsign.org)
- Sign Translate
- Mimix Speech to sign language website
- ProDeaf Translator - Microsoft Apps
- Hand Talk: your website accessible in ASL
- MotionSavvy UNI: 1st sign language to voice system | Indiegogo
- WACV 2020 Open Access Repository (thecvf.com)
- [2401.04730] A Simple Baseline for Spoken Language to Sign Language Translation with 3D Avatars (arxiv.org)
- CVPR 2023 Open Access Repository (thecvf.com)
- WACV 2020 Open Access Repository (thecvf.com)
- Pose-Based Sign Language Recognition Using GCN and BERT (thecvf.com)
- [2110.05877] OpenHands: Making Sign Language Recognition Accessible with Pose-based Pretrained Models across Languages (arxiv.org)
- WACV 2021 Open Access Repository (thecvf.com)
- CVPR 2017 Open Access Repository (thecvf.com)
- Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition | Proceedings of the AAAI Conference on Artificial Intelligence
- WACV 2021 Open Access Repository (thecvf.com)
- BSL-1K: Scaling Up Co-articulated Sign Language Recognition Using Mouthing Cues | SpringerLink
- Hand-Model-Aware Sign Language Recognition | Proceedings of the AAAI Conference on Artificial Intelligence
- CVPR 2020 Open Access Repository (thecvf.com)
- [2302.05075] BEST: BERT Pre-Training for Sign Language Recognition with Coupling Tokenization (arxiv.org)
- ICCV 2021 Open Access Repository (thecvf.com)
- CVPR 2021 Open Access Repository (thecvf.com)
- [2110.06161] Sign Language Recognition via Skeleton-Aware Multi-Model Ensemble (arxiv.org)