

Software Engineering Department  
ORT Braude College

Capstone Project Phase A – 61998

**24-1-R-2**

****

Supervisor:

Alex Keselman

[Alex@vir-tec.net](mailto:Alex@vir-tec.net)

GitHub: [ShayZak1/SonicCipher (github.com)](https://github.com/ShayZak1/SonicCipher)

Authors:

Shay Zak 312330749

[Shay.zak@e.braude.ac.il](mailto:Shay.zak@e.braude.ac.il)

Michael Ioffe 315388900

[Michael.Ioffe@e.braude.ac.il](mailto:Michael.Ioffe@e.braude.ac.il)

Table of Contents

[1. Abstract 3](#_Toc167377801)

[2. Introduction 3](#_Toc167377802)

[3. Background & Related Work 4](#_Toc167377803)

[3.1 API 4](#_Toc167377804)

[3.1.1 Google Voice To Text API 5](#_Toc167377805)

[3.1.2 OpenAI API 5](#_Toc167377806)

[3.1.3 Google Translate API 6](#_Toc167377807)

[3.1.4 Google Tank Management API 6](#_Toc167377808)

[3.2 Security 6](#_Toc167377809)

[3.2.1 Peer-To-Peer 6](#_Toc167377810)

[3.2.2 Asymmetric encryption 7](#_Toc167377811)

[3.3 Development Tools 7](#_Toc167377812)

[3.3.1 Preact 7](#_Toc167377813)

[3.3.2 HTML 8](#_Toc167377814)

[3.3.3 CSS 8](#_Toc167377815)

[3.3.5 Node.JS 8](#_Toc167377816)

[3.3.5 Preact Vs React 9](#_Toc167377817)

[3.3.6 Tailwind CSS 9](#_Toc167377818)

[4. Expected Achievements 10](#_Toc167377819)

[4.1 Success Criteria 10](#_Toc167377820)

[4.2 Unique Features 10](#_Toc167377821)

[5. Engineering Process 11](#_Toc167377822)

[5.1 process 11](#_Toc167377823)

[5.2 Website Page 12](#_Toc167377824)

[5.3 Voice Input Capture 13](#_Toc167377825)

[5.3.1 Google Speech-To-Text 13](#_Toc167377826)

[5.3.2 Text Translation via Google Translate API for Basic Translation 14](#_Toc167377827)

[5.3.3 Text translation via OpenAI API 14](#_Toc167377828)

[5.3.4 Read the Translation via Google Text-To-Speech API 14](#_Toc167377829)

[5.3.5 Bidirectional Translation Feature with Reply Functionality 15](#_Toc167377830)

[6. Product 15](#_Toc167377831)

[6.1 Use Case 15](#_Toc167377832)

[6.1 Requirements 16](#_Toc167377833)

[6.2 User Interface 17](#_Toc167377834)

[7. Evaluation/Verification Plan 19](#_Toc167377835)

[8. References 20](#_Toc167377836)

# Abstract

In today's connected world, overcoming language barriers is crucial for enhancing communication and fostering understanding among diverse populations. Our capstone project at ORT Braude College introduces an innovative web application aimed at bridging these barriers through cutting-edge voice technologies. Utilizing Google's Speech-to-Text and Translate APIs alongside OpenAI's ChatGPT, our application offers real-time voice-to-text conversion and translation across multiple languages in two distinct modes: basic translation for quick understanding and advanced translation for context-sensitive, nuanced communication. This dual approach allows users to choose the level of translation that best suits their needs, enhancing flexibility and user experience. The application stands out for its ability to deliver seamless and accurate communication solutions that are accessible on both desktop and mobile platforms. It is designed to improve user satisfaction by providing intuitive and responsive interactions, ensuring that users can easily engage in cross language related conversations without the need for technical expertise. Additionally, the application incorporates essential security measures to ensure that user interactions remain private and protected, supporting a safe communication environment. This project aims to enhance global communication, making it simpler and more inclusive, thereby transforming the world into a more connected and accessible place.

# Introduction

The start of globalization has made it necessary to develop technologies that helps

Others around the world to surpass the language related barriers, helping others around the world to talk with each other in easier way. Our project introduces an innovative web application that works both for desktop and smartphone users, which aims to close the gap between languages through the power of voice. This application will stand at the junction of advanced natural language processing (NLP) with encryption technologies,

Utilizing the Google translate API and ChatGPT API to record the spoken language and convert it into written text and afterwards translate it into the user’s preferred language with outstanding accuracy.in this era where our world is connected more than ever,

the need for secure and efficient communication between languages is now more important than ever. Our project addresses this demand by enabling users to easily communicate in their origin language, while guaranteeing that their messages are understood by the application accurately and translated accurately to the language of the recipient. More than only translation, our application integrates Peer-to-peer and asymmetric encryption, which promises that the exchange of the communication during the translations and meetings not only flawless but also secure, protecting the privacy of the communication from potential eavesdropping. Created with a strong emphasis on user satisfaction, our application is addressing this need by enabling users to effortlessly communicate in their native language, whether you are on your phone or sitting at a computer. with this app, we strive to improve and promote a more inclusive digital world.

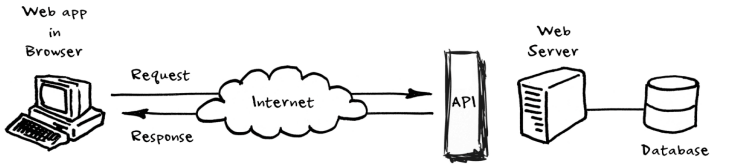
**2.1** motivation

This project is motivated from recognizing the limitations of the current translation tools, we use the ChatGPT API which understands in advanced way the language that the user seeks to translate his input or translate by Google translate API that stores a huge language library. This combination of the APIs aims to close the gap between the language barriers more effectively and ensure that the communication will stay private. By using these technologies together, we are assisting the users to have a clearer conversation with different languages. Furthermore, many popular translation tools like Google Translate, iTranslate, Microsoft Translate are well known and commonly used across the world for quick text and speech translation across many languages. However those tools makes mistakes when it comes to fully understanding the variations of the given language. And by understanding this problem This is where our project comes in, which will use ChatGPT with the advanced understanding of languages or with Google Translate API in order to translate our recorded text in basic way.

# Background & Related Work

In our increasingly globalized world, the need to overcome language barriers has never been more critical. Voice-based communication tools offer a promising solution, allowing people to interact naturally in their native languages while relying on technology to bridge the gap. However, ensuring these communications are secure, especially when sensitive information is shared, is equally important. This literature survey explores the current landscape of voice recognition, real-time translation technologies, and secure communication methods.

## 3.1 **API**

API (Application Programming Interface) is a set Procedures for building and interacting with software applications. It enables different software programs to interact with each other, allowing them to share data and features without the need to know how they're implemented with each other. In web-based systems and development, APIs are crucial for enabling fusion and communication between different software applications and services over the web. They allow web applications to receive data from servers without needing to reload the webpage and more. API’s enables many functions like payment processing and data analyzation in web applications which makes them more interactive with the user

### 3.1.1 Google Voice To Text API

Googles Voice To Text API, is part of Google Clouds Speed To Text service which represents a complex and sophisticated solutions for making deep learning techniques

Much better in transforming spoken language into text. Voice to Text APIs works by capturing audio input which afterwards is handled by deep learning algorithms which transform the speech into text. It is designed to recognize over 120 languages. This API works in many different environments, such as noisy backgrounds by using advanced noise cancellation techniques. It has many other features Such as supporting real time transcription, distinguish between multiple speakers and more. The API also is customizable, enabling users to train the model on domain-specific vocabulary for improved accuracy. This precision make it an invaluable tool for developers looking to incorporate voice recognition into their applications.



### 3.1.2 OpenAI API

OpenAI API is a new solution from OpenAI. Its designed to bring sophisticated natural language processing capabilities into many applications. the API enables the integration of conversational AI, making machines understand and generate human-like text responses. its built on the latest versions of the GPT models, which are known for the ability to understand context, making relevant text, and even being a sensible activity.

this API supporting many various of languages and can handle with some conversational scenarios. From simple interactions to complex dialogue management, the GPT allows real-time conversations based in real-time texts and asynchronous across various domains. It is especially adapting at understanding nuances in language, thanks to its extensive training in many various internet texts, what makes it appropriate to the relevant applications which is required for the interactions between human and computer.

Developers can fit the API to their specific needs with matched prompts so they can guide responses to particular tones or content styles. This matching, combined with its ability to improve responses for a long time, and making the ChatGPT to a valuable tool for businesses and developers which are strive to improve their user experience with conversational AI. Whether if its for customer support or for entertainment , the ChatGPT API provides us strong framework for AI communication into apps, websites and etc.

### 3.1.3 Google Translate API

Google Translate API is a tool from Google. It helps computers change text from one language to another. This API lets apps understand and translate text in many languages. It uses smart tech to do translations that make sense. This API can be used in different apps and websites. It's good for when you need to show text in many languages. You can use it for simple words or for big paragraphs. It makes talking or sharing information with people who speak different languages easier. People who make apps or websites can use this API to help their users read or talk in the language they know best. It's helpful for businesses or anyone who wants to reach more people in their own language. Google Translate API is easy to fit into your app or website. You can start using it to translate text right away. It's a powerful tool for anyone who needs to communicate across different languages.

### 3.1.4 Google Tank Management API

The Google Tank Management API, a component of Google Cloud’s infrastructure services, is designed to streamline the management of large-scale data storage systems, or "tanks." This API enables precise control and maintenance of data tanks, utilizing deep learning to optimize resource allocation and predict maintenance needs, thus reducing downtime. It performs reliably under various operational conditions, offering real-time monitoring that provides insights into system health and resource usage. This feature is crucial for maintaining the efficiency and stability of data operations across diverse environments. By simplifying data management challenges, the Google Tank Management API allows businesses to enhance operational efficiency and focus on growth, making it a valuable tool for developers in data-intensive industries.

## 3.2 Security

### 3.2.1 Peer-To-Peer

Our project Applies Peer-to-Peer (P2P) technology to break down language barriers by employing a distributed approach, real time communication. Different from other traditional models, P2P allows direct data exchange among users, enhancing and improving the privacy and reducing latency of the process. This technology enhances and makes efficient voice to text conversion and translation without relying on central servers, making our system scalable and durable. We will use WebRTC for peer connections, ensuring seamless interaction. Security is addressed through encryption and peer authentication. P2P's integration underscores our commitment to providing a secure, efficient platform for overcoming language obstacles, embodying our vision for a more inclusive digital world. P2P technology will be implemented in our project when we develop our live video website conversations which will allow others to translate their speech during live conversation.

### 3.2.2 Asymmetric encryption

Asymmetric encryption is a way of scrambling messages so that only the person meant to read them can understand. Each user has a public key and a private key. Unlike a simple lock and key system where one key opens one lock, this method uses two keys for each person. This method ensures that only the intended recipient, who has the correct private key, can decrypt the message. Anyone can use this key to encrypt a message for you, but once it's locked, only you have the private key to decrypt and read it. This ensures that even if someone intercepts the message, they can't read it, only the intended person can. The process of key generation involves complex mathematics, basically depends on problems considered computationally infeasible, such as the factorization of large numbers or the computation of discrete logarithms. This is what makes the keys secure and keeps the messages safe. In our project, asymmetric encryption plays a crucial role in securing communication between users speaking different languages. When you send a message, it gets locked with the receiver's public key. This is not only secures the message from unauthorized access but also verifies the sender's identity through digital signatures, improving trust and privacy in our communication app.. This way, we're not just keeping the messages safe from snooping, we're also making sure that the message really comes from you, building trust and keeping things private in our app for chatting across languages. Asymmetric encryption will be implemented in our project when we develop our live video website conversations which will allow others to translate their speech during live conversation.

## 3.3 Development Tools

### 3.3.1 Preact

Preact is a lightweight JavaScript library that mirrors the capabilities of React but at a fraction of the size, making it ideal for projects where performance and payload are critical. It employs the same component-based architecture as React, allowing developers to manage complex UIs efficiently. With its compatibility with React, projects can seamlessly transition to Preact using the preact-compat add-on, facilitating easy migration without significant codebase changes. Preact's small footprint (around 3KB) ensures faster load times, enhancing performance especially in progressive web applications and other high-demand environments. It maintains high compatibility with React’s ecosystem, providing developers a robust alternative that doesn't compromise on the ability to deliver dynamic and responsive interfaces. By adopting Preact, developers leverage its fast rendering capabilities and efficient update handling through a virtual DOM, ensuring a smooth and performant user experience in web applications.

### 3.3.2 HTML

HTML (HyperText Markup Language) is the foundational building block of the web, providing the structural framework for web pages. It enables the creation of structured documents by defining elements such as headers, paragraphs, links, and other content units. HTML is essential for web development as it dictates the organization and presentation of content on the Internet. Its simplicity and versatility allow developers to easily integrate multimedia resources, form elements, and other interactive features, making it indispensable in creating user-friendly web applications. By leveraging HTML, developers ensure that content is accessible and functional across various devices and web browsers, enhancing the reach and effectiveness of web applications.

### 3.3.3 CSS

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of documents written in HTML or XML. CSS empowers web developers to control layout, colors, fonts, and the overall visual aspect of web pages, enabling consistent styling across multiple pages of a website. The separation of content (HTML) from design (CSS) simplifies site maintenance and ensures greater flexibility and control in the presentation of web pages. By using CSS, developers can create responsive designs that adapt to different screen sizes and resolutions, providing an optimal viewing experience. CSS's capabilities contribute significantly to the aesthetic appeal and user experience of web applications, playing a crucial role in user engagement and retention.

### 3.3.5 Node.JS

Is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting and running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games). The Node.js distributed development project was previously governed by the Node.js Foundation and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

### 3.3.5 Preact Vs React

Preact and React are both popular JavaScript libraries for creating user interfaces, but they serve slightly different purposes due to their inherent characteristics: Preact is a streamlined version of React, designed to be small and efficient. It retains a similar API to React, which allows developers who are experienced with React to adapt easily to Preact. The main goal of Preact is to offer a lighter and faster alternative for projects where size and performance are critical, such as mobile applications, single-page applications, and progressive web apps. On the other hand, React is more robust and supports a broader range of features, making it suitable for developing complex, large-scale applications, including those for enterprises. One of the criticisms of React is its relatively large size and slower initial load times, which stem from its comprehensive feature set and the use of a virtual DOM. This overhead can affect performance, particularly on devices with limited resources. These performance concerns with React have led to the development of alternatives like Preact, which strips away some of the less commonly used features and optimizations inherent in React to provide a nimbler, more resource-efficient library. For projects requiring fast performance and efficient resource utilization, Preact is often the preferred choice over React.

### 3.3.6 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that allows for rapid development by applying pre-defined classes directly in HTML. This approach reduces the need for custom CSS, promoting efficiency and maintainability. Tailwind's customization options enable us to create a unique and responsive design tailored to our project's needs. Its utility classes simplify responsive design, ensuring our application looks great on all devices. Additionally, Tailwind's focus on reusability and component-based styling aligns with our goal of building a consistent and visually appealing user interface for our language translation web application.

# Expected Achievements

Our project aims to create a friendly and efficient web application which is designed to surpass language barriers through innovative voice to text translation and after wards voice to voice translation technologies which will be highly accurate, secure and effective by leveraging the power of multiple AI APIS and combining them together in order to receive the best translated output that will ensure high accuracy in diverse environments and across various dialects and accents

## 4.1 Success Criteria

* Achieving a high level of accuracy in voice recognition and translation across a broad spectrum of languages, with minimal delays in processing, thereby enhancing user satisfaction.
* Successfully implementing encryption and security measures that protect users live conversations, meeting global privacy standards and regulations.
* The ability of the application to scale seamlessly with increased usage while maintaining high performance, demonstrating the robustness of its architecture.
* Broad language support and accessible design that cater to a diverse user base, promoting global communication without barriers.

## 4.2 Unique Features

* Adaptive Voice recognition
* Contextual translation accuracy
* Real time language switching
* End to End encryption.

# Engineering Process

This section elaborates the challenges we had faced and are expecting to encounter, as well techniques we had employed during the engineering process of the product.

## 5.1 process

In the previous phase of our project, aimed at Helping different languages work together and communicate through advanced voice translation technology, we started exploring and trying to understand the structure of our project and understanding of the foundational architecture required to realize our vision. The project was segmented into distinct learning (phase a) and development phases (phase b), each with specific objectives and methodologies, to ensure a comprehensive understanding and application of the necessary technologies, including the Google Translate API and ChatGPT API and Google Voice To Text API.

1. **learning the tools for our project/learning for research**

Parallel to our technical preparations, we engaged in an extensive research phase aimed at grounding our project in the latest advancements and best practices within the fields of speech recognition and language translation.

This included: A thorough investigation into existing speech recognition systems, their basic models, and commercial products leveraging similar technologies.

- An academic review of literature in the speech recognition domain, enhancing our understanding of key concepts and metrics such as Word Error Rate (WER), similarity indices, and data indexing strategies. The formulation of hypotheses on how to extend the capabilities of existing APIs to achieve greater accuracy and functionality in our application.

1. **challenges we face**

A critical question guided our investigative efforts: How can we augment the baseline functionality of Google's speech to text API to deliver enhanced accuracy and comprehensive functionality? Through our research, we identified potential methods to increase precision, including the implementation of similarity indices for improved result validation. Furthermore, we explored innovative approaches to enable full application functionality through voice commands, enriching the user experience and expanding the applicability of our service.

As we transition into the next stage of development, the project demands an expansion of our technical skillset, including advanced web development, infrastructure management,

and programming languages not previously covered in our academic coursework. This requirement underscores the project's role as a catalyst for professional growth and

learning.

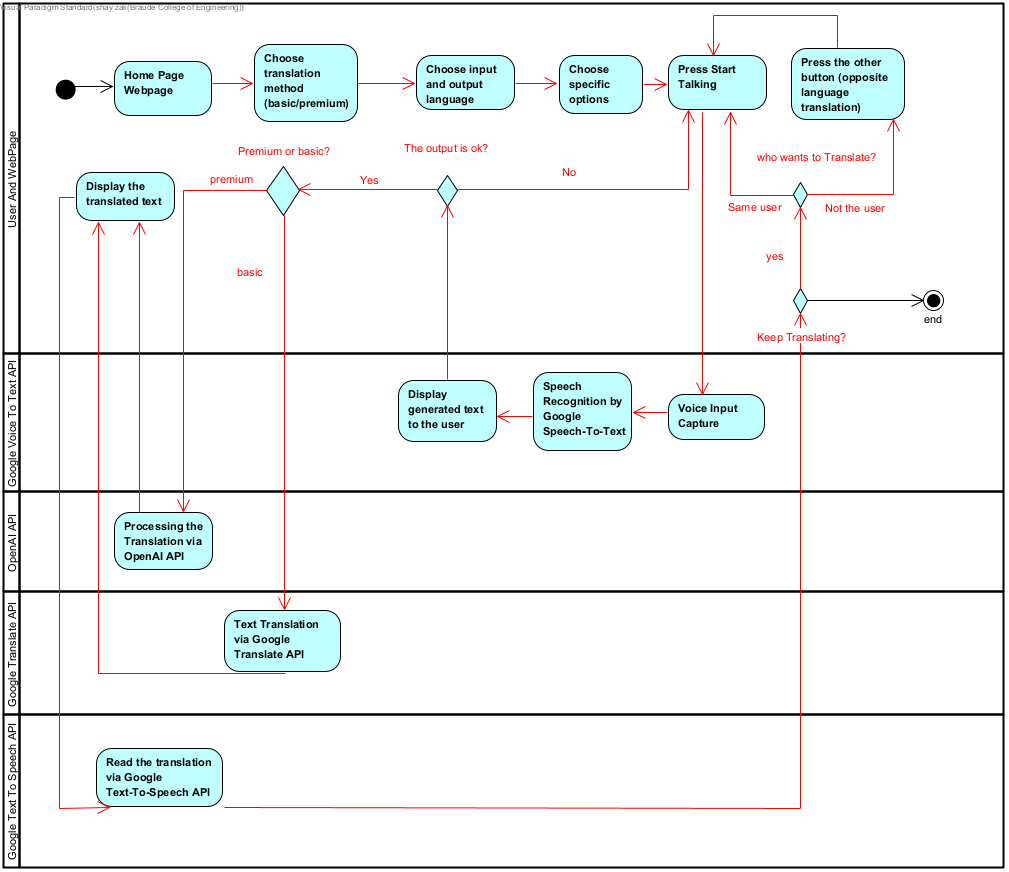


Figure 1: System Architecture

## 5.2 Website Page

Our website acts as the primary interface for our innovative language translation application, designed to be straightforward and accessible to users of all backgrounds, regardless of their technical expertise. The goal is to develop a user interface that is intuitive and simple to navigate. This approach involves the strategic placement of visual elements to create an appealing interface that does not overwhelm users. We ensure that the design remains consistent throughout the user's journey, from languages selection and specific options to the final translation output.

In building our website, we leverage robust web development frameworks and libraries such as Bootstrap and Tailwind and React\Preact. Bootstrap use will be to utilized to ensure that the website is responsive and functional across various devices and screen sizes, providing a consistent experience whether on a desktop, tablet, or mobile device. Preact's component-based architecture enables us to efficiently manage the state and dynamics of the website, facilitating real-time interactions such as language selection and the display of translated text.

## 5.3 Voice Input Capture

When a user visits the website, they are presented with a simple and intuitive interface. To start the voice input process, the user clicks on a "Start Recording" button. This action prompts the user for permission to access their microphone, which is a standard security feature in browsers to protect privacy. Before starting the voice input, users select their preferred input and output languages from the provided options. This ensures that the speech recognition and translation will be done in the most efficient way.Afterwards, users will have the opportunity to choose additional settings that could affect the voice capture, such as a specific dialect or accent within the selected language for better recognition accuracy, gender of the user and to who the translation is meant to. After setting their preferences, users initiate the voice input capture by pressing the "Start Talking" button. This action activates the microphone, and the website starts recording the user's speech.

### 5.3.1 Google Speech-To-Text

After The website captures the spoken words in real-time, using the device's microphone. This stage involves converting the analog sound waves into digital signals that can be processed by the speech recognition API. The digital audio data is sent to Google's Speech-To-Text service (API). Here, the audio is analyzed, and advanced algorithms transcribe the spoken language into written text. The recognition process is designed to accommodate the chosen input language and specific user settings. After the speech is recognized and converted into text that will be transferred to Google Translate API. this transcribed text is immediately displayed on the website for the user. This provides an opportunity for the user to confirm the accuracy of the captured speech.

### 5.3.2 Text Translation via Google Translate API for Basic Translation

The Google Translate API serves as a fundamental component in our application by providing quick and efficient text translations. After the user's speech is converted to text via Google's Speech-To-Text API, this text is then fed into the Google Translate API. This API offers a direct translation method, focusing on word-for-word accuracy across a broad range of languages. It's particularly useful for users who seek rapid and straightforward translation without the nuanced adjustments provided by the OpenAI API.

### 5.3.3 Text translation via OpenAI API

The OpenAI API enhances our application by providing contextual translations based on user specifications. After converting speech to text using Google's Speech-To-Text API, the text is processed by the OpenAI API, which uses advanced models to interpret and translate it according to the context, tone, and user-defined settings such as formality level and intended audience. This contrasts with the Google Translate API, which offers more straightforward, but less context-aware translations. The OpenAI API's ability to understand nuanced language makes it ideal for users needing precise and culturally relevant translations. This approach ensures that our application meets the diverse needs of our users, providing them with accurate and relevant translations across various languages and contexts.

### 5.3.4 Read the Translation via Google Text-To-Speech API

Once the users spoken language is captured and translated by Google Translate API and OpenAI API the final translated text will be displayed in the wished translation language.

But the main function of our application is to read aloud the translated generated text to the other person which the translated text is meant to. Afterwards employment of the Google Text-To-Speech (TTS) API to vocalize the translated text will be applied. This feature is particularly useful for users who are learning a new language, are visually impaired, or prefer auditory learning. After the text is ready to be played as voice playback, the user can initiate the voice playback by clicking a 'Listen' button or Icon.

The user has control over the playback and can pause, stop, or replay the audio as needed to fully understand the content. To ensure quality and user satisfaction, there might be a feature that allows users to provide feedback on the clarity and usefulness of the audio playback and the translation in order to study and improve our accuracy.

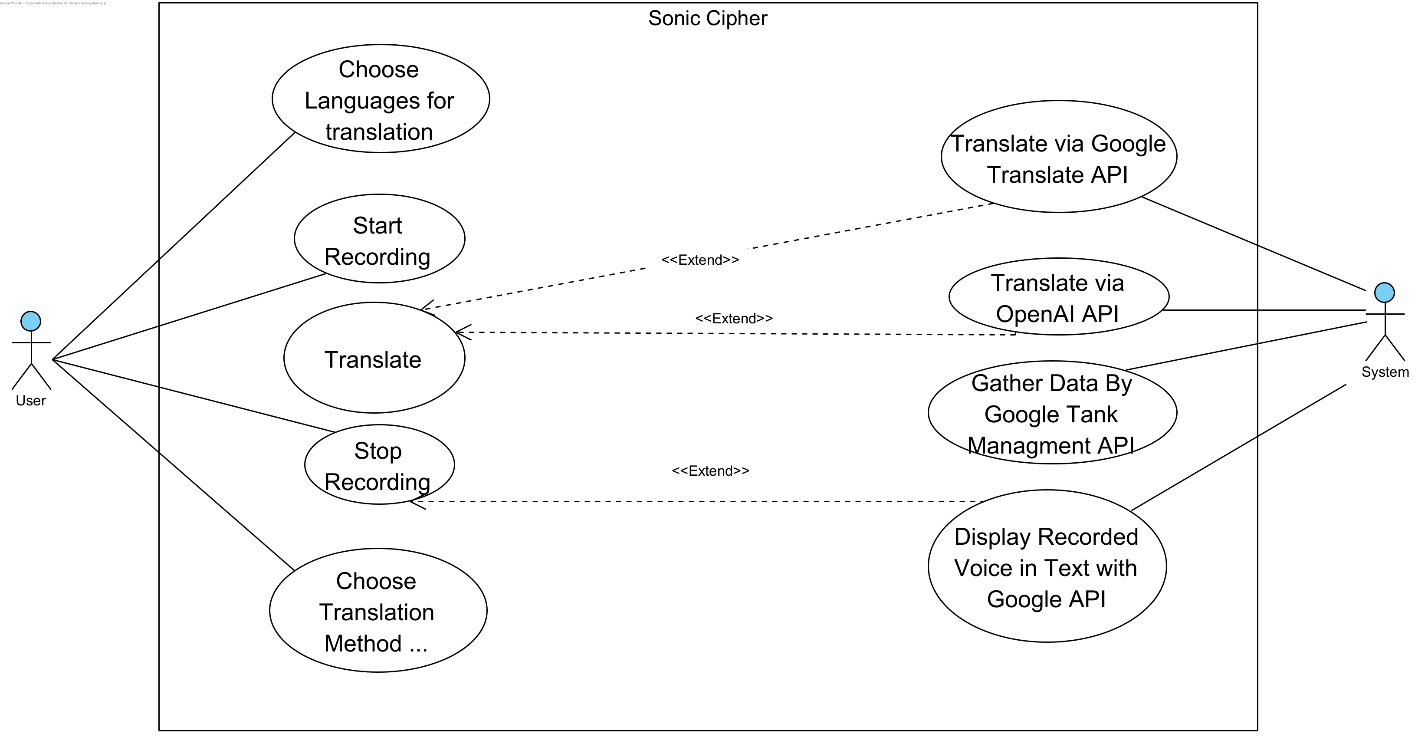
### 5.3.5 Bidirectional Translation Feature with Reply Functionality

To augment the utility of our translation platform and foster two-way communication, we are introducing an innovative "Reply" feature. This functionality is designed to allow a listener, who has just heard the translated text, to respond in their own language, with the reply then being translated back into the original user's language.

The 'Reply' button facilitates a dialogue, allowing the second user to instantly engage in the conversation by recording their voice message. Once the second user has finished speaking, their message is processed through the same robust path captured, transcribed, and translated back into the original user's language, mirroring the initial user's experience but in reverse. This button is prominently displayed and becomes active after the first translation is read aloud, inviting the listener to participate in the exchange.

# Product

## 6.1 Use Case

****

## 6.1 Requirements

**Functional requirements**

|  |  |
| --- | --- |
| 1 | The system facilitates real-time voice-to-text translation. |
| 2 | The system supports a comprehensive list of languages for input and output. |
| 3 | The system integrates with the Google Speech-to-Text API for transcription. |
| 4 | The system integrates with the Google Translate API for language translation. |
| 5 | The system integrates with the Google Text-to-Speech API for auditory output of translated text. |
| 6 | The system provides an interactive 'Reply' feature for two-way translation in a conversation. |
| 7 | The system allows users to review and edit transcriptions and translations before finalizing. |
| 8 | The system includes content filtering to prevent the translation of inappropriate content. |
| 9 | The system offers customization options for audio output, such as speed and voice selection. |

**Non-Functional Requirements**

|  |  |
| --- | --- |
| 1 | The system ensures user privacy and data protection in accordance with global standards. |
| 2 | The system delivers a user-friendly and intuitive interface for ease of navigation and use. |
| 3 | The system maintains high availability with minimal downtime for a reliable user experience. |
| 4 | The system is accessible and fully functional across various devices and web browsers. |
| 5 | The system achieves rapid processing and display of translations for efficient communication. |
| 6 | The system produces high-quality translations with a focus on accuracy and contextual relevance. |
| 6 | The system's design is scalable to accommodate a growing number of users and increased data volume. |

## 6.2 User Interface

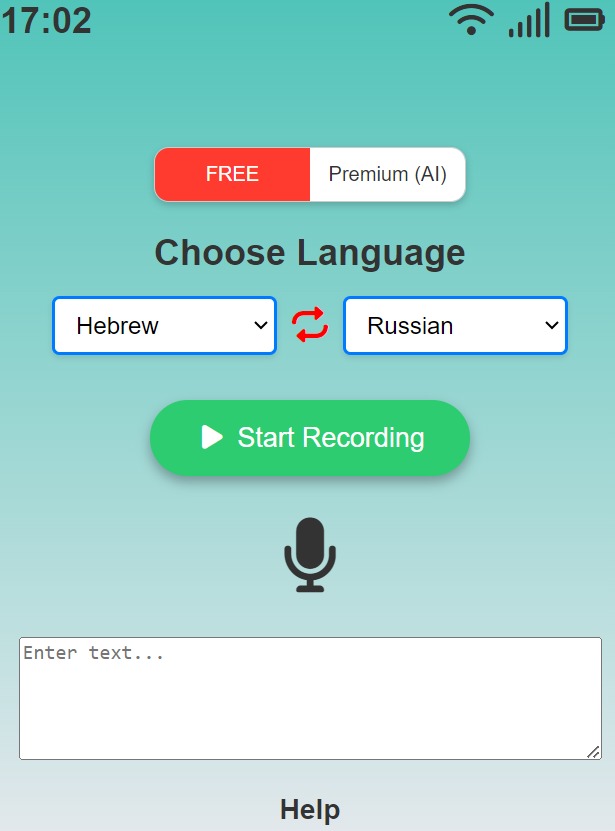
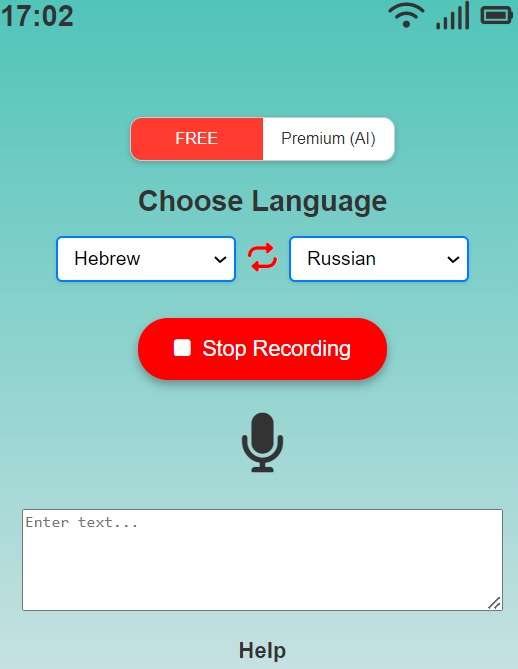
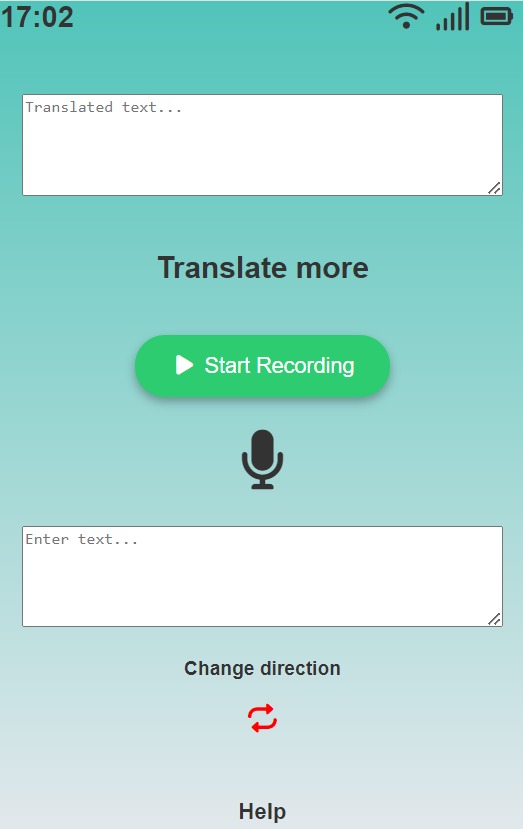


Figure 4: GUI Free translation recording

Figure 3: GUI Free translation

A screenshot of a computer

Description automatically generated

Figure 6: Post Translation screen

Figure 5: GUI Premium(AI) translation

**Explanation about the User Interface screens:**

1. (Figure 3) the main page of our site.

We can see in the top middle of the page an option (FREE, PREMIUM(AI)) which allows to user To choose which method he wants to use to translate his captured voice text. Below it we can see the drops boxes which will allow the to pick the languages he wants to make the translation.

Bellow it we have the button “Start Recording” which will be changed after the user pressed it. Right after the user presses the button the voice capture process is activated. And below in the text box we will see the generated text which is captured by Google Translate API.

1. (Figure 4) this is the same page just like in Figure 3, the only difference between them is the Record Button which here is red, and his label is “Stop Recording” which allows the user to stop the recording process.
2. (Figure 5) this page shows us the pre translation page for the Premium (AI) option. in this section the user can translate text using AI (OpenAI API) to translate. Therefore, the user will have an another text box which will allow the user to enter specifications Which will allow to OpenAI API to generate the translation in much more precise way and with the user wished way to translate or pronounce the text.
3. (Figure 6) we can see the translation page which will show us the translated output in the upper textbox, in case the user wants to keep translating in the same language only different voice input, he can press Start recording and the same process will begin. in case the other side (which the original text is translated for) wants to reply and he stands next to user which translated first. He can press the Change Direct button in order to change the language way of the translation (for example if the main user translated the text from Hebrew to English, the change direction button will translate from now own the text from English to Hebrew) .

# Evaluation/Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Tested Function | Tested Module | Testing Plan | Expected Result |
| 1 | Initiating voice recording | Voice Input Capture | User initiates the voice recording by clicking "Start Recording". | System begins recording and displays a visual cue. |
| 2 | Stopping voice recording | Voice Input Capture | User clicks "Stop Recording". | System stops recording and saves the audio file. |
| 3 | Voice to text conversion | Google Speech-To-Text API | Recorded audio is sent to the API for transcription. | System displays the transcribed text on the screen. |
| 4 | Text translation | Google Translate API | Transcribed text is sent to the API for translation. | Translated text is displayed in the user-selected language. |
| 5 | Speech command execution | System Commands | User gives a voice command to navigate the app. | App navigates according to the command if recognized. |
| 6 | Language selection | User Interface | User selects a language from the dropdown menu. | Selected language is set for voice input and output. |
| 7 | Real-time translation | Translation System | User speaks into the microphone; system translates in real-time. | System displays and voices the translation instantly. |
| 8 | Playback of translated text | Google Text-To-Speech API | System reads back the translated text using TTS. | Audio playback of translated text is clear and accurate. |
| 9 | Bidirectional translation with reply | Two-way Translation | Second user responds using the "Reply" feature. | System translates and displays the second user's response. |
| 10 | User feedback submission | Feedback System | User submits feedback through the app interface. | Feedback is recorded for analysis and future improvements. |
| 11 | System performance under load | System Stability | Multiple users access the app simultaneously. | App remains stable and responsive without crashes or lag. |
| 12 | Error handling for unsupported languages | Error Management | User selects a language not supported by the APIs. | System displays an error message and suggests alternatives. |
| 13 | Continuous voice input without interruption | Voice Input Capture | User speaks for an extended period without breaks. | System captures entire input without cutting off. |

# References

*[1] Google Developers. (2022). "Speech-to-Text: Automatic Speech Recognition." Google Cloud. https://cloud.google.com/speech-to-text*

*[2] OpenAI. (2023). "OpenAI API." https://beta.openai.com/docs/*

*[3] Google Developers. (2022). "Translate API: Integrating real-time translation into your apps." Google Cloud. https://cloud.google.com/translate*

*[4] Mozilla Foundation. (2023). "HTML: Hypertext Markup Language." MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTML*

*[5] Mozilla Foundation. (2023). "CSS: Cascading Style Sheets." MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/CSS*

*[6] Preact. (2023). "Preact: Fast 3kB alternative to React with the same modern API." https://preactjs.com/*

*[7] WebRTC. (2021). "WebRTC: Real-time communication for the open web." https://webrtc.org/*

*[8] IEEE. (2018). "Explore the Principles of Asymmetric Encryption." IEEE Xplore, March 2018. https://ieeexplore.ieee.org/document/8374916*