**The Islamia University of Bahawalpur**

**Department of Artificial Intelligence**

****

**SOFTWARE DESIGN DESCRIPTION**

**(SDD DOCUMENT)**

**for**

**Real Time Translator**

***By***

**Student Name**

**GULRAIZ SHAKOOR**

**Roll No**

**F20BARIN1E02049**

**Session Spring/Fall 2020 – 2024**

***Supervisor***

**Supervisor Name**

**Supervisor Designation**

***Bachelor of Science in Artificial Intelligence***

**Table of Contents**

[**Introduction 1**](#_Toc121305837)

[**Design methodology and software process model 1**](#_Toc121305838)

[**System overview 1**](#_Toc121305839)

[**Design models [along with descriptions] 1**](#_Toc121305840)

[**Data design 2**](#_Toc121305841)

[**Algorithm & Implementation 2**](#_Toc121305842)

[**Software requirements traceability matrix 2**](#_Toc121305843)

[**Human interface design 3**](#_Toc121305844)

[**Appendix I 3**](#_Toc121305845)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature

a

### 1 Introduction

The Real-Time [Translator](http://en.wiktionary.org/wiki/translator) project [aims](http://en.wiktionary.org/wiki/aims) to [facilitate](http://en.wiktionary.org/wiki/facilitate) [seamless](http://en.wiktionary.org/wiki/seamless) communication across different languages in a real-time manner. The project has [progressed](http://en.wiktionary.org/wiki/progressed) through several key [modules](http://en.wiktionary.org/wiki/modules) , each [contributing](http://en.wiktionary.org/wiki/contributing) to the overall [functionality](http://en.wiktionary.org/wiki/functionality) of the system. Here's a brief [overview](http://en.wiktionary.org/wiki/overview) of the project scope and its covered [modules](http://en.wiktionary.org/wiki/modules) : Language Support [Module](http://en.wiktionary.org/wiki/module) , Speech Recognition [Module](http://en.wiktionary.org/wiki/module) ,Translation [Module](http://en.wiktionary.org/wiki/module) , Text-to-Speech [Module](http://en.wiktionary.org/wiki/module) ,User [Interface](http://en.wiktionary.org/wiki/interface) [Module](http://en.wiktionary.org/wiki/module) ,Dynamic Language [Configuration](http://en.wiktionary.org/wiki/configuration) , Continuous Listening [Module](http://en.wiktionary.org/wiki/module) ,Error Handling [Module](http://en.wiktionary.org/wiki/module) ,Technological [Innovation](http://en.wiktionary.org/wiki/innovation) ,Global Collaboration and Cross-Cultural Interaction,

**2 Design**[**methodology**](http://en.wiktionary.org/wiki/methodology)**and software process model :**

**Object-**[**Oriented**](http://en.wiktionary.org/wiki/oriented)**Design (**[**OOD**](http://en.wiktionary.org/wiki/ood)**):**[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)Explanation: The Real-Time [Translator](http://en.wiktionary.org/wiki/translator) application [employs](http://en.wiktionary.org/wiki/employs) an object- [oriented](http://en.wiktionary.org/wiki/oriented) design [methodology](http://en.wiktionary.org/wiki/methodology) . The system is structured around [objects](http://en.wiktionary.org/wiki/objects) , which represent real-world [entities](http://en.wiktionary.org/wiki/entities) and [encapsulate](http://en.wiktionary.org/wiki/encapsulate) both data and the [methods](http://en.wiktionary.org/wiki/methods) that operate on the data.  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Justification:**  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)[**Modularity**](http://en.wiktionary.org/wiki/modularity)**:**[OOD](http://en.wiktionary.org/wiki/ood) [promotes](http://en.wiktionary.org/wiki/promotes) [modularity](http://en.wiktionary.org/wiki/modularity) , making it [easier](http://en.wiktionary.org/wiki/easier) to manage, scale, and maintain the application. Each component, such as language translation, speech recognition, and user [interface](http://en.wiktionary.org/wiki/interface) , can be [encapsulated](http://en.wiktionary.org/wiki/encapsulated) within separate [objects](http://en.wiktionary.org/wiki/objects) .  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)[**Reusability**](http://en.wiktionary.org/wiki/reusability)**:** Object- [oriented](http://en.wiktionary.org/wiki/oriented) design [facilitates](http://en.wiktionary.org/wiki/facilitates) code [reuse](http://en.wiktionary.org/wiki/reuse) . Common [functionalities](http://en.wiktionary.org/wiki/functionalities) , like language translation and [audio](http://en.wiktionary.org/wiki/audio) processing, can be [encapsulated](http://en.wiktionary.org/wiki/encapsulated) within [classes](http://en.wiktionary.org/wiki/classes) , [promoting](http://en.wiktionary.org/wiki/promoting) [reusability](http://en.wiktionary.org/wiki/reusability) across the application.  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Ease of Understanding:** [OOD](http://en.wiktionary.org/wiki/ood) [enhances](http://en.wiktionary.org/wiki/enhances) code [readability](http://en.wiktionary.org/wiki/readability) and [comprehension](http://en.wiktionary.org/wiki/comprehension) by [modeling](http://en.wiktionary.org/wiki/modeling) the application based on real-world [entities](http://en.wiktionary.org/wiki/entities) and their [interactions](http://en.wiktionary.org/wiki/interactions) .  
**Software Process Model:**  
[Iterative](http://en.wiktionary.org/wiki/iterative) and [Incremental](http://en.wiktionary.org/wiki/incremental) Development ( [IID](http://en.wiktionary.org/wiki/iid) ):  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Explanation**: The development of the Real-Time [Translator](http://en.wiktionary.org/wiki/translator) application [follows](http://en.wiktionary.org/wiki/follows) an [iterative](http://en.wiktionary.org/wiki/iterative) and [incremental](http://en.wiktionary.org/wiki/incremental) approach. The system [evolves](http://en.wiktionary.org/wiki/evolves) through multiple [iterations](http://en.wiktionary.org/wiki/iterations) , each building upon the previous one, and new [features](http://en.wiktionary.org/wiki/features) are added [incrementally](http://en.wiktionary.org/wiki/incrementally) .  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Justification:**  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)[**Adaptability**](http://en.wiktionary.org/wiki/adaptability)**:** [IID](http://en.wiktionary.org/wiki/iid) [allows](http://en.wiktionary.org/wiki/allows) for flexible [adaptation](http://en.wiktionary.org/wiki/adaptation) to [changing](http://en.wiktionary.org/wiki/changing) [requirements](http://en.wiktionary.org/wiki/requirements) . As the application [involves](http://en.wiktionary.org/wiki/involves) complex [functionalities](http://en.wiktionary.org/wiki/functionalities) like speech recognition and translation, an [iterative](http://en.wiktionary.org/wiki/iterative) approach [enables](http://en.wiktionary.org/wiki/enables) [refining](http://en.wiktionary.org/wiki/refining) and improving these [components](http://en.wiktionary.org/wiki/components) over time.  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Early and Continuous User**[**Feedback**](http://en.wiktionary.org/wiki/feedback)**:** [Incremental](http://en.wiktionary.org/wiki/incremental) development [allows](http://en.wiktionary.org/wiki/allows) for the early delivery of essential [features](http://en.wiktionary.org/wiki/features) . This [enables](http://en.wiktionary.org/wiki/enables) [users](http://en.wiktionary.org/wiki/users) to provide [feedback](http://en.wiktionary.org/wiki/feedback) , [ensuring](http://en.wiktionary.org/wiki/ensuring) that the application [aligns](http://en.wiktionary.org/wiki/aligns) with their [expectations](http://en.wiktionary.org/wiki/expectations) and [requirements](http://en.wiktionary.org/wiki/requirements) .  
[**•**](http://en.wiktionary.org/wiki/%E2%80%A2)**Risk**[**Mitigation**](http://en.wiktionary.org/wiki/mitigation) : [Addressing](http://en.wiktionary.org/wiki/addressing) high-risk [components](http://en.wiktionary.org/wiki/components) early in the development process [minimizes](http://en.wiktionary.org/wiki/minimizes) [uncertainties](http://en.wiktionary.org/wiki/uncertainties) . For example, [refining](http://en.wiktionary.org/wiki/refining) the accuracy of speech recognition or [optimizing](http://en.wiktionary.org/wiki/optimizing) translation [algorithms](http://en.wiktionary.org/wiki/algorithms) can be [tackled](http://en.wiktionary.org/wiki/tackled) [incrementally](http://en.wiktionary.org/wiki/incrementally) .

**Development Process:**

1. **Requirements Analysis:**

• Identify and analyze the requirements of the Real-Time Translator application, considering features like real-time speech recognition, language translation, and user-friendly interfaces.

2.**Design:**

Apply object-oriented design principles to create modular and reusable classes. Design the architecture to support real-time interactions and intuitive user interfaces.

3.**Implementation (Iteration 1):**

Develop the initial version of the application, focusing on core functionalities like speech recognition and basic language translation.

4.**Testing (Iteration 1):**

Conduct testing to ensure the reliability and correctness of the core features. Address any identified issues.

5.**Implementation (Iteration 2):**

Enhance the application by adding more languages, improving translation accuracy, and refining the user interface.

6.**Testing (Iteration 2):**

Perform additional testing to validate the new features and improvements. Conduct user acceptance testing to gather feedback.

7.**Repeat:**

Continue the iterative process, addressing user feedback, adding new features, and refining existing functionalities based on evolving requirements. This iterative and incremental development process, combined with object-oriented design, enables the Real-Time Translator application to evolve, adapt, and improve over time, meeting user expectations and accommodating changes in technology and user needs.

### 3 System overview

### Functionality:

### The Real-Time Translator is an application designed to facilitate instant translation of spoken language, providing users with the ability to communicate seamlessly across different languages. The primary functionalities include:

### Speech Recognition:

### • Utilizes speech recognition algorithms to convert spoken language into text.

### Language Translation:

### • Employs translation algorithms to convert the recognized text from one language to another in real-time.

### Text-to-Speech:

### • Converts the translated text into audible speech, enabling users to hear the translated content.

### User Interface:

### • Provides a user-friendly interface for language selection, input, and output.

### Modularity

### • Organizes functionalities into modular components, enhancing maintainability and scalability.

### Context:

### In a globalized world with diverse linguistic backgrounds, effective communication is often hindered by language barriers. The Real-Time Translator addresses this challenge by offering a practical and efficient solution for instant language translation during verbal communication.

### Key Contextual Elements:

### • Multilingual Communication: Facilitates communication between speakers of different languages in real-time.

### • Cross-Cultural Interaction:

### Supports cross-cultural interactions in various scenarios, such as travel, business, and social events.

### • Enhanced Accessibility:

### Provides an inclusive tool for individuals with different language proficiencies to communicate effectively.

### Design:

### Object-Oriented Design:

### • Organizes the application into modular objects representing core functionalities like speech recognition, translation, and user interface.

### • Promotes reusability, maintainability, and scalability.

### Iterative and Incremental Development:

### • Adopts an iterative approach for continuous improvement and adaptation to user feedback.

### • Allows for the incorporation of new languages, enhancements to translation accuracy, and user interface refinements over successive iterations.

### User-Friendly Interface:

### • Incorporates an intuitive interface for easy language selection, audio input, and clear presentation of translated output.

### Background Information:

### • The project originates from the need for a practical, real-time language translation tool that transcends traditional language barriers.

### • Built to address the challenges faced by individuals navigating linguistic diversity in various contexts.

The Real-Time Translator project aims to foster seamless communication across languages, promoting inclusivity, and breaking down language barriers in an interconnected world.

**1.1**[**Architectural**](http://en.wiktionary.org/wiki/architectural)**design**  
To achieve the complete [functionality](http://en.wiktionary.org/wiki/functionality) of the Real-Time [Translator](http://en.wiktionary.org/wiki/translator) system, a [modular](http://en.wiktionary.org/wiki/modular) program structure is designed, [emphasizing](http://en.wiktionary.org/wiki/emphasizing) [modularity](http://en.wiktionary.org/wiki/modularity) , [reusability](http://en.wiktionary.org/wiki/reusability) , and [maintainability](http://en.wiktionary.org/wiki/maintainability) . The major [subsystems](http://en.wiktionary.org/wiki/subsystems) include Speech Recognition, Language Translation, and Text-to-Speech. Here's a high-level [overview](http://en.wiktionary.org/wiki/overview) of the system's [modular](http://en.wiktionary.org/wiki/modular) structure:  
[**Modules**](http://en.wiktionary.org/wiki/modules)**:**  
**1. Speech Recognition**[**Module**](http://en.wiktionary.org/wiki/module)**:**  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) Responsible for [converting](http://en.wiktionary.org/wiki/converting) spoken language into text.  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Utilizes](http://en.wiktionary.org/wiki/utilizes) speech recognition [algorithms](http://en.wiktionary.org/wiki/algorithms) and libraries.  
**1. Language Translation**[**Module**](http://en.wiktionary.org/wiki/module)**:**  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Performs](http://en.wiktionary.org/wiki/performs) real-time translation of the [recognized](http://en.wiktionary.org/wiki/recognized) text.  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Utilizes](http://en.wiktionary.org/wiki/utilizes) language translation [algorithms](http://en.wiktionary.org/wiki/algorithms) and external translation [services](http://en.wiktionary.org/wiki/services) .  
**2. Text-to-Speech**[**Module**](http://en.wiktionary.org/wiki/module)**:**  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Converts](http://en.wiktionary.org/wiki/converts) [translated](http://en.wiktionary.org/wiki/translated) text into [audible](http://en.wiktionary.org/wiki/audible) speech.  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Utilizes](http://en.wiktionary.org/wiki/utilizes) text-to-speech synthesis [algorithms](http://en.wiktionary.org/wiki/algorithms) and libraries.  
**3. User**[**Interface**](http://en.wiktionary.org/wiki/interface)[**Module**](http://en.wiktionary.org/wiki/module)**:**  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Manages](http://en.wiktionary.org/wiki/manages) the interaction between the user and the system.  
[•](http://en.wiktionary.org/wiki/%E2%80%A2) [Handles](http://en.wiktionary.org/wiki/handles) language selection, input, and output presentation.  
[**Relationships**](http://en.wiktionary.org/wiki/relationships)**:**  
+-------------------------+  
[|](http://en.wiktionary.org/wiki/%7C) User [Interface](http://en.wiktionary.org/wiki/interface) [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) ------------------------- [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - Input Language [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - Output Language [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - Interaction Logic [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
v v  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) Speech Recognition [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) ------------------------- [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - [Audio](http://en.wiktionary.org/wiki/audio) Input [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - [Recognized](http://en.wiktionary.org/wiki/recognized) Text [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
v v  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) Language Translation [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) ------------------------- [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - [Translated](http://en.wiktionary.org/wiki/translated) Text [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
v v  
+----+--------------+-----+  
[|](http://en.wiktionary.org/wiki/%7C) Text-to-Speech [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) ------------------------- [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) - [Audio](http://en.wiktionary.org/wiki/audio) Output [|](http://en.wiktionary.org/wiki/%7C)  
[|](http://en.wiktionary.org/wiki/%7C) [|](http://en.wiktionary.org/wiki/%7C)  
+-------------------------+

##### **3.2 Process flow/Representation**

Provide a representation of the flow of **MAJOR Processes** of your system in the form of an activity diagram.

### • Speech Recognition ⟷ Language Translation:

### • The Speech Recognition module provides recognized text to the Language Translation module.

### • Language Translation processes the text and returns the translated content.

### • Language Translation ⟷ Text-to-Speech:

### • The Language Translation module provides translated text to the Text-to-Speech module.

### • Text-to-Speech converts the text into audible speech.

### • User Interface ⟷ Speech Recognition:

### • The User Interface module initiates speech recognition based on user input.

### • Receives the recognized text for further processing.

### • User Interface ⟷ Language Translation:

### • Manages user selections for input and output languages.

### • Communicates with the Language Translation module to obtain translations.

### • User Interface ⟷ Text-to-Speech:

### • Coordinates with the Text-to-Speech module to play the translated speech.

### • Presents the translated output to the user.

### 4 Design models [along with descriptions

### Design models provide a visual representation of the system's structure, behavior, and interactions. Here's how the mentioned design models can be applied to a real-time translator project:

1. **Class Diagram:**

• Description: Illustrates the classes and their relationships in the system. It includes attributes and methods of each class.

• Application: Useful for an object-oriented approach. Shows entities like SpeechTranslator, Audio Input, Translator, etc. and their relationships.

1. **Sequence Diagram:**

**•** Description: Demonstrates the interactions between different components or objects in a sequential manner over time.

• Application: Helps visualize the flow of actions during a specific operation, such as the translation process from audio input to output.

1. **State Transition Diagram:**

**•** Description: Represents the states an object goes through in response to events. Defines the conditions for transitioning between states.

• Application: Useful for modeling the different states of the translation process, such as listening, processing, and speaking.

1. **Data Flow Diagram (DFD):**

**•** Description: Illustrates the flow of data between processes, data stores, and external entities.

• Application: Helpful for representing the flow of audio data, translation results, and interactions between system components.

1. **Schematic Diagram (Hardware projects):**

**•** Description: Provides a graphical representation of the system's hardware components and their connections.

• Application: Relevant for hardware aspects of the project, illustrating the connections between microphones, processors, and output devices.

1. **Timing Diagram (Hardware projects):**

• Description: Shows the timing relationships between different components in a system.

• Application: Useful for representing the timing of audio input, processing, and output in a real-time translator hardware system.

**In addition:**

**• System Block Diagram: •**

Description: Provides an overview of the major components of the system and their connections.

• Application: Useful for understanding the high-level architecture of the real-time translator system.

**• Flow Charts:**

**• Description:**

Diagrams that represent the sequence of operations in a process.

**• Application:**

Can be used to illustrate the flow of control in different parts of the real-time translation process. Ensure that the choice of each model is justified based on the nature of the project. For instance, class diagrams are suitable for an object-oriented approach, while DFDs are more aligned with procedural design. Each diagram should be clear, well-labeled, and aligned with the overall design of the real-time translator.

### 5 Data design

### In the context of a real-time translator system, the information domain involves the audio input, language selection, translation process, and audio output. This information domain is transformed into data structures to enable efficient storage, processing, and organization of the major entities within the system. Let's break down how this transformation occurs:

### Audio Input:

### • Data Structure: Raw audio data, typically represented as a digital signal. • Storage: Audio data can be stored in memory or temporary buffers.

### • Processing:

### The system captures audio input using a microphone, and the digital signal is processed for recognition.

### Language Selection:

### • Data Structure: Language codes or identifiers.

### • Storage: Stored in variables or configuration settings.

### • Processing: User input or system settings determine the source and target languages for translation.

### Translation Process:

### • Data Structure:

### Text data for recognized speech, translated text.

### • Storage: Recognized speech is stored as text; translated text is stored temporarily.

### • Processing:

### The system employs natural language processing (NLP) techniques to recognize and translate speech. This involves the use of algorithms and data structures to analyze and understand the linguistic content.

Top of Form

List any databases or data storage items.

**1• Processing:**

Facilitates smooth transitions between different stages of the translation process, allowing data to be efficiently passed between components.

1. **Translation History (Optional):**

**3 • Data Structure:** Database or file storing previous translations.

**4 • Storage:** Persistent storage for historical data.

**5 • Processing:** Allows users to access and review past translations.

This transformation from the information domain to data structures involves careful consideration of data representation, storage mechanisms, and the algorithms used for processing. It ensures that the system can handle real-time translation efficiently, providing accurate and timely results. The choice of appropriate data structures contributes to the overall effectiveness and responsiveness of the real-time translator.

##### **5.1 Data dictionary**

##### A data dictionary provides a detailed description of the data used in a system, including its structure, meaning, relationships, and attributes. In the context of a real-time translator system, here's an example of a simplified data dictionary:

##### **Audio Input Data:**

##### • **Description:** Raw audio data captured from the user's speech.

##### • **Attributes**:

##### • **Format:** Digital audio signal

##### • **Source:** Microphone

##### • **Storage:** Temporary buffer

##### **2. Language Selection Data:**

##### • **Description**: Information related to the selected input and output languages.

##### • **Attributes**:

##### • **Input Language:** Chosen by the user

##### • **Output Language:** Chosen by the user

##### • **Format:** Language codes or identifiers

##### • **Storage:** Configuration settings

##### **3. Translation Data:**

##### • **Description:** Text data representing recognized speech and its translation.

##### • **Attributes**:

##### • **Recognized Text:** Converted from audio input

##### • **Translated Text:** Result of language translation

##### • **Format:** String

##### • **Storage:** Temporary variables

##### **Audio Output Data:**

##### • **Description:** Processed audio data for playback.

##### • **Attributes:**

##### • **Format:** Digital audio signal

##### • **Destination:** Speakers or headphones

##### • **Storage:** Temporary buffer

##### **System Entities Data:**

##### **• Description:** Objects representing key system components.

##### **• Attributes:**

##### • SpeechTranslator:

##### Main application instance

##### • Audio Input:

##### Component handling audio input

##### • Translator:

##### Component performing language translation

##### • Format: Objects/classes

##### • Storage: Instances in memory

##### **Temporary Storage Data:**

##### • Description:

##### Temporary buffers or queues for intermediate data.

##### • Attributes:

##### • Format: Buffers, queues

##### • Usage: Storing intermediate results during processing

##### • Storage: In-memory structures

##### **Translation History Data (Optional):**

##### • Description:

##### Historical data of previous translations.

##### • Attributes:

##### • Records: Previous translations with timestamps

##### • Format: Database or file storage

##### • Storage: Persistent storage

##### This data dictionary provides a high-level overview of the types of data used in the real-time translator system, their attributes, and their storage characteristics. It serves as a reference for developers, helping ensure a common understanding of data elements across the project team and facilitating effective communication during the design and implementation phases.

### 6 Algorithm & Implementation

These pseudo-code snippets provide a high-level summary of the algorithms for key functions in the real-time translator system, including language setup, translation, text-to-speech, and the main program loop. The actual implementation details would involve specific libraries and APIs based on the chosen programming language.Top of Form

### 7 Software requirements traceability matrix

Certainly! Below is an example table summarizing how each software requirement has been met in the document:  
**Table 7 Requirements Traceability Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **Ref. Item** | **Design Component** | **Component Items** |
| FR01 | Class Diagram | Class Name | Function Name(s) |

**Table 8 Requirements Traceability Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| OR | | | |
| FR01 | DFD | Diagram Number/Level | Function Name(s) |

### Human interface design

### From the user's perspective, the real-time translation system provides a simple and intuitive interface to facilitate seamless language translation using speech. The key features and user interaction are as follows:

### Language Selection:

### • The user starts by selecting the input language (the language spoken) and the output language (the desired translated language).

### The system provides a list of supported languages for the user to choose from.

### Speech Input:

### • The user speaks into the microphone, providing the system with the speech input in the selected input language. The system utilizes speech recognition to convert the spoken words into text.

### Translation:

### • The recognized text is then translated in real-time from the input language to the specified output language using the Google Translate API.

### Output Display:

### • The translated text is displayed on the user interface, allowing the user to view the real-time translation results.

### Text-to-Speech Output:

### • Simultaneously, the translated text is converted into speech using the gTTS (Google Text-to-Speech) library. The system plays the translated audio, allowing the user to hear the translated content.

### User Feedback:

### • Throughout the process, the user receives feedback on the system's actions. Messages such as "Listening..." during speech input, "Processing..." during translation, and "Text to Speech in [Language]..." during audio output keep the user informed about the system's progress.

### Error Handling:

### • Robust error handling is in place to manage potential issues, and informative error messages are displayed if errors occur during speech recognition, translation, or text-to-speech processes.

### User-Friendly Interface:

### • The interface is designed to be user-friendly, with clear options for language selection and intuitive feedback messages. Buttons for language choices and a responsive system contribute to a positive user experience.

### Multiple Language Support:

### • The system supports multiple input and output languages, providing flexibility for users to communicate in different languages.

By following these steps, the user can effectively utilize the system for real-time language translation, both in written and spoken forms. The combination of speech recognition, translation, and text-to-speech functionalities enhances the user's ability to communicate across language barriers. The feedback messages contribute to a transparent and interactive user experience.

8.1 Screen images

Display screenshots showing the interface from the user‟s perspective. These can be hand- drawn, or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

* 1. Screen objects and actions

In the context of a real-time translation system, the user interacts with the application through a graphical user interface (GUI). The screen objects and associated actions facilitate a smooth and user-friendly experience. Here's a discussion of common screen objects and the actions associated with them:

1. Language Selection Dropdowns:

• Objects: Dropdown menus for selecting the input and output languages.

• Actions: Users can click on the dropdowns to choose the source (spoken) and target (translated) languages from the provided list.

1. Speech Input Display:

• Objects: Visual representation or text area displaying the speech input as it is being recognized.

• Actions: Provides real-time feedback to the user, showing the system's interpretation of the spoken words.

1. Translation Display:

• Objects: Text area or display field showing the translated text.

• Actions: Updates in real-time as the system processes the speech input and translates it into the selected output language.

1. Audio Player Controls:

• Objects: Play, pause, and volume controls for the translated audio.

• Actions: Allows the user to control the playback of the translated audio, including starting, pausing, and adjusting the volume.

1. Feedback Messages:

• Objects: Dynamic text areas or pop-up messages.

• Actions: Display messages such as "Listening...", "Processing...", and "Text to Speech in [Language]..." to inform the user about the system's current state.

1. Error Messages:

• Objects: Popup or inline messages in case of errors.

• Actions: Provides informative messages if issues occur during speech recognition, translation, or audio playback.

1. Language Options:

• Objects: Buttons or icons for language options.

• Actions: Allow users to change language preferences or access language settings for input and output.

1. Translation Button:

• Objects: Button triggering the translation process.

• Actions: Initiates the real-time translation when clicked after the user has selected input and output languages.

1. Clear/Reset Button:

• Objects: Button for clearing input and resetting the system.

• Actions: Clears the speech input, translated text, and resets the system for a new translation.

1. Microphone Icon:

• Objects: Visual representation of a microphone.

• Actions: Indicates when the system is actively listening for speech input.

These screen objects and associated actions collectively contribute to a comprehensive and interactive user interface. The design focuses on clarity, responsiveness, and real-time updates to enhance the overall user experience while using the real-time translation system.

### Appendix I

* + - How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: <http://agilemodeling.com/artifacts/>
    - How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link: <http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf>
    - Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:
      * <http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm>
      * Software Engineering –A Practitioners approach by Roger Pressman
    - Architecture diagram: For guidance please follow the instructions mentioned in the link and book:
      * Ian Sommerville – Software Engineering 9th Edition– Chapter 6