**Video Translation Software**

**ABSTRACT**

This project endeavors to develop a sophisticated software solution aimed at facilitating the translation of videos from English to various languages spoken in India, encompassing diverse religious and cultural backgrounds. The primary objective is to bridge linguistic gaps and promote inclusivity by making video content accessible and comprehensible to a wider audience.

The software employs state-of-the-art machine translation techniques and deep learning algorithms to ensure accurate and contextually relevant translations. Leveraging the advancements in natural language processing, the system aims to provide high-quality translations that capture the nuances of different languages, including those associated with various religions prevalent in India.

Key features of the software include user-friendly interfaces, efficient video processing capabilities, and support for multiple Indian languages. The development process involves the integration of cutting-edge technologies such as neural machine translation and robust language models.

The anticipated impact of this project is significant, contributing to the democratization of information and fostering cross-cultural understanding. By enabling the translation of videos into languages associated with different religions, the software aspires to promote cultural harmony and facilitate the sharing of diverse perspectives.

**Keywords:** Video Translation, Machine Translation, Deep Learning, Natural Language Processing, Cross-Cultural Communication.

**1. INTRODUCTION**

**1.1 OBJECTIVE OF PROJECT:**

The project aims to develop user-friendly software for translating videos from English to various languages associated with different religions in India. The primary objectives include implementing advanced machine translation techniques, ensuring precise translations, supporting multiple video formats, and promoting cross-cultural communication. The software emphasizes linguistic diversity, cultural understanding, and the democratization of information through cutting-edge technologies and user-friendly interfaces.

**1.2 PROBLEM STATEMENT:**

The project addresses the challenge of bridging linguistic and cultural gaps in video content by developing software for accurate translation from English to various languages representing different religions in India. The problem lies in the limited availability of tools catering to diverse linguistic needs, hindering the accessibility and understanding of content across different communities. This project aims to overcome these barriers and provide a solution that promotes inclusivity and cultural sensitivity in video communication.

**1.3 MOTIVATION:**

* **Multilingual Accessibility:** Enable individuals from diverse linguistic backgrounds in India and other regions to access video content in their native languages, fostering inclusivity.
* **Cultural Bridging:** Address the challenge of linguistic and cultural barriers by providing a tool that facilitates the translation of videos into languages associated with different religions and cultures.
* **Community Empowerment:** Empower communities by enabling them to engage with content that reflects their cultural and linguistic identities, promoting a sense of belonging and representation.
* **Enhancing Communication:** Contribute to improved communication and understanding among people of various linguistic and religious backgrounds, fostering mutual respect and harmony.
* **Promoting Inclusivity:** Tackle the issue of language-related exclusion in media by creating a software solution that promotes inclusivity and equal access to information and entertainment.

**1.4 SCOPE:**

The scope of the project is extensive, covering various aspects of linguistic diversity, cultural representation, and accessibility to video content. The key components of the project's scope include:

* **Language Coverage:** The software aims to support translation into languages associated with different religions in India, ensuring broad coverage to accommodate diverse linguistic preferences.
* **User-Friendly Interface:** The development of a user-friendly interface is a crucial aspect of the project's scope. The software should be intuitive and easily navigable, allowing users to seamlessly interact with the budding process.
* **Translation Accuracy:** The project places importance on the accuracy of translation. Efforts will be directed towards leveraging advanced language processing techniques to ensure the faithful translation of video content.
* **Cultural Sensitivity:** Cultural nuances and sensitivities will be considered during the translation process to maintain the integrity and appropriateness of the content across different linguistic and cultural contexts.
* **Scalability**: The software will be designed with scalability in mind, allowing for the addition of new languages and features in future updates. This ensures that the tool remains relevant and adaptable to evolving linguistic landscapes.
* **Community Feedback Integration:** The project scope involves mechanisms for gathering user feedback to continually improve the software. This iterative process aims to address user needs and preferences, enhancing the overall user experience.
* **Resource Efficiency:** Efficient resource utilization, both in terms of computational resources and translation processing time, is a critical consideration. The software should deliver timely translations without excessive resource demands.
* **Security and Privacy:** The scope extends to implementing robust security measures to safeguard user data and maintain privacy during the translation process. Compliance with data protection regulations will be prioritized.
* **Educational Outreach:** As part of the broader scope, there may be opportunities for educational outreach programs to raise awareness about the software's capabilities and promote its usage among different communities.
* **Integration with Existing Platforms:** Exploring possibilities for integration with existing video-sharing platforms or applications is within the project scope. This could enhance the software's reach and impact.

The comprehensive scope of the project aims to create a versatile and impactful software solution that goes beyond mere translation, addressing the multifaceted challenges associated with linguistic diversity, cultural representation, and accessibility in the realm of video content.

**1.5 PROJECT INTRODUCTION:**

The advent of globalization and digital connectivity has led to an unprecedented influx of content consumption across diverse linguistic and cultural landscapes. However, this surge in digital content is not always accessible or relatable to individuals whose primary language and cultural context differ from the content's origin. In this context, the project emerges as a response to the growing need for a sophisticated software solution that seamlessly translates and buds videos from English to various Indian languages, with a specific emphasis on accommodating religious nuances.

Understanding the multifaceted nature of linguistic diversity in India, the software goes beyond mere translation. It employs advanced budding algorithms to ensure a smooth and contextually appropriate transition between segments of the video content. Moreover, the software recognizes the importance of religious sensitivity in language translation, particularly when dealing with content related to diverse religions. As such, it incorporates mechanisms to handle religious terminologies with respect and accuracy, enhancing the overall quality of the translated content.

One of the distinguishing features of the software is its user-centric approach. It provides users with the flexibility to customize the budding process based on their preferences, allowing for a personalized and culturally sensitive viewing experience. By placing control in the hands of the users, the software aims to cater to individual preferences and ensure that the translated content aligns with their cultural and religious backgrounds.

In essence, this project is not merely a technological solution but a cultural bridge that facilitates a deeper understanding and appreciation of content across linguistic and religious boundaries. As we delve into the project's development and functionalities, it becomes evident that it addresses a significant gap in the digital content landscape, contributing to cultural inclusivity and technological innovation.

**2. LITERATURE SURVEY**

**2.1 Related work:**

**[1] Felix Stahlberg:**

The landscape of machine translation (MT) has undergone a revolutionary transformation in recent years, marked by a decisive shift from traditional Statistical MT to the ascendancy of Neural Machine Translation (NMT). For decades, Statistical MT relied on count-based models, but the advent of NMT introduced a singular neural network approach to translation. This work delves into the evolutionary trajectory of modern NMT architectures, tracing their roots to the foundational concepts of word and sentence embeddings. Additionally, we explore earlier instances of the encoder-decoder network family, laying the groundwork for the subsequent rise of NMT.

**Summary:**

In summary, this exploration navigates the historical progression of machine translation methodologies, focusing on the pivotal transition from Statistical MT to the predominant era of Neural Machine Translation. Emphasizing the role of neural networks, particularly encoder-decoder architectures, we uncover the foundations that paved the way for modern NMT. The narrative extends to recent trends, providing a comprehensive overview of the current state of the field and its ongoing evolution.

**[2] Markus Freitag, Orhan Firat:**

This paper addresses the prevalent English-centric focus in Multilingual Neural Machine Translation (MNMT) models, emphasizing the underutilization of direct data between non-English language pairs. The authors explore the concept of multi-way aligned examples within widely used bilingual corpora, presenting a strategy to enrich English-centric parallel datasets with complete connectivity. The resulting model, named complete Multilingual Neural Machine Translation (cMNMT), is introduced to establish connections between all source and target languages.

**Summary:**

The study investigates the incorporation of multi-way aligned examples to expand English-centric corpora into a complete graph, connecting every language pair. The authors propose a novel training data sampling strategy conditioned on the target language, leading to competitive translation quality for all language pairs in cMNMT. The paper also explores the impact of multi-way aligned data size, transfer learning capabilities, and the ease of adding new languages to MNMT.

**[3] Alexandre Berard, Olivier Pietquin, Christophe Servan, Laurent Besacier:**

This paper pioneers an end-to-end approach for speech-to-text translation, eliminating the reliance on source language transcription during both learning and decoding phases. The proposed model showcases promising outcomes on a limited synthetic French-English corpus. This departure from the necessity of source language transcription holds significant implications for data collection methodologies in speech translation, particularly in under-resourced scenarios. The conventional practice of obtaining speech transcripts, often requiring detailed transcription guides, may be circumvented by successful end-to-end speech-to-text translation models.

**Summary:**

The research challenges the conventional paradigm by introducing a model that directly translates speech to text without the need for source language transcription. The approach's success is exemplified through positive results on a synthetic French-English corpus. The paper discusses the potential transformation in data collection strategies, envisioning scenarios where bilingual speakers can produce speech in the source language based on target language text utterances. This innovative methodology extends applicability to unwritten source languages.

**[4] Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, Rishemjit Kaur:**

Neural Machine Translation (NMT) has evolved significantly over the past decade, gaining widespread adoption since the early 2000s and reaching a mature phase. Despite its dominance in machine translation, its efficacy diminishes when applied to low-resource language pairs, primarily due to the scarcity of extensive parallel corpora. The spotlight is now on implementing NMT techniques tailored for low-resource languages, sparking notable research in this domain. This article conducts a comprehensive survey of advancements in Low-Resource Language NMT (LRL-NMT), coupled with a quantitative analysis to discern prevalent techniques. The aim is to furnish guidelines for selecting suitable NMT techniques in diverse LRL data settings based on empirical findings. The article also offers a panoramic overview of the LRL-NMT research landscape and suggests avenues for furthering research efforts.

**Summary:**

This article delves into the intricacies of Neural Machine Translation (NMT), emphasizing its transformative journey and widespread adoption. Despite its success, the challenges in applying NMT to low-resource language pairs prompt a detailed exploration of research developments in Low-Resource Language NMT (LRL-NMT). The survey encompasses a quantitative analysis of prevailing techniques, intending to guide the selection of NMT approaches tailored for specific LRL data settings. The holistic view provided encompasses the current state of LRL-NMT research and extends recommendations to catalyze future research endeavors.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING METHOD**

The conventional approach to video translation involved manual processes, including hiring translators, synchronizing translations with video content, and addressing cultural nuances. This method faced challenges such as high costs, time consumption, and scalability issues, hindering timely and efficient translation.

**Limitations:**

* **Manual Translation:** Labor-intensive and time-consuming manual translation processes.
* **Synchronization Challenges**: Difficulty in achieving perfect synchronization of translated text with video scenes.
* **Cultural Adaptation:** Requirement for careful consideration of cultural nuances during translation.
* **Limited Scalability:** Inability to easily scale for multiple languages or a large volume of content.
* **High Costs:** Financial burden associated with human translation services.
* **Time-Consuming:** Delays in the delivery of translated content affecting timely releases.
* **Quality Control:** Challenges in maintaining consistent translation quality across diverse contexts.

**Proposed Improvements:**

The proposed software aims to revolutionize video translation by introducing automation, scalability, real-time capabilities, and a user-friendly interface. This innovative approach addresses existing limitations, making video translation more efficient, accessible, and adaptable to diverse linguistic and cultural contexts.

**3.2 DISADVANTAGES**

* **Manual Feature Engineering:** Traditional methods often require the manual extraction and selection of audio features, which can be time-consuming and may not capture all relevant information in complex audio data.
* **Limited Adaptability:** These methods may have limited adaptability to varying audio patterns and may struggle to generalize well to diverse sound environments.
* **Labeling and Data Requirements:** Traditional approaches heavily rely on labeled datasets for supervised learning, which can be expensive and time-intensive to create, especially for large-scale applications.
* **Difficulty with Noisy Data:** Handling noisy or unstructured audio data is challenging for traditional methods, as they may not effectively filter out irrelevant information or cope with background noise.
* **Suboptimal Performance:** In comparison to deep learning models like CNN, MobileNet, and ResNet, traditional approaches may achieve suboptimal performance, particularly when dealing with large and complex audio datasets.

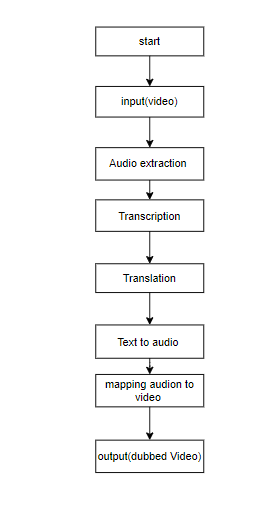
**3.3 PROPOSED SYSTEM**

The software aims to revolutionize video translation from English to various Indian languages. Key features include advanced language processing for accuracy, a user-friendly interface, support for multiple languages, real-time translation, adaptive language models, platform compatibility, and robust privacy measures. The agile development approach ensures continuous improvement and user satisfaction.

**3.4 ADVANTAGES:**

* **Accurate Language Processing:** The software utilizes advanced language processing algorithms, ensuring precise and contextually relevant translations.
* **User-Friendly Interface:** A seamless and intuitive interface allows users to navigate and use the software effortlessly, promoting accessibility.
* **Multi-Language Support:** The software supports translation into various Indian languages, facilitating broader audience reach and cultural inclusivity.
* **Real-Time Translation**: Real-time translation capabilities enable users to experience immediate language conversion during video playback.
* **Adaptive Language Models:** The software employs adaptive language models, continuously learning and improving translation accuracy over time.
* **Platform Compatibility:** Designed to be compatible with multiple platforms, users can access the software on diverse devices, enhancing convenience.
* **Robust Privacy Measures:** Stringent privacy measures are implemented to safeguard user data, ensuring a secure and confidential translation experience.
* **Agile Development Approach:** The software follows an agile development methodology, allowing for continuous updates, enhancements, and responsiveness to user feedback.

**3.5 PROJECT FLOW**



**4. HARDWARE & SOFTWARE REQUIREMENTS**

**SOFTWARE REQUIREMENS**

Operating System : Windows 7/8/10

Server side Script : HTML, CSS, Bootstrap & JS

Programming Language : Python

Libraries : Flask, Pandas, Librosa,, GTTS, moviepy, speech\_recognition. googletrans

IDE/Workbench : VSCode

Technology : Python 3.6+

Server Deployment : Xampp Server

Database : MySQL

**HARDWARE REQUIREMENTS**

Processor - I3/Intel Processor

RAM - 8GB (min)

Hard Disk - 128 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - Any

1. **Methodology:**

**1. Data Collection:**

* **Diverse Dataset:** Gather a comprehensive dataset comprising videos representing different content genres, linguistic styles, and cultural contexts.
* **Audio Variation:** Include videos with diverse audio characteristics, such as accents, intonations, and expressions, to capture the richness of language.

**2. Audio Extraction:**

* **Tools and Techniques:** Utilize audio extraction tools to separate the audio streams from the collected videos.
* **Quality Check:** Ensure the extracted audio maintains high quality and fidelity.

**3. Audio Transcription:**

* **Automatic Speech Recognition (ASR):** Employ ASR techniques to transcribe the extracted audio into textual format.
* **Model Consideration:** Choose between pre-trained ASR models or consider training custom models based on the dataset characteristics.

**4. Text Language Detection:**

**Algorithm Implementation:** Implement language detection algorithms to identify the source language (English) of the transcribed text.

**Religious Context:** Develop language detection mechanisms sensitive to religious terms and expressions.

**5. Text Translation:**

* **Machine Translation**: Use machine translation algorithms to translate the transcribed English text into the desired Indian languages.
* **Religious Terminology:** Implement specialized translation modules for accurate rendering of religious terminology.

**6. Text-to-Speech (TTS):**

* **TTS Synthesis:** Convert the translated text into speech using Text-to-Speech synthesis.
* **Customization:** Customize the TTS system to reflect appropriate intonations, expressions, and cultural nuances, especially in the context of religious content.

**7. Mapping of Audio to Video:**

* **Integration:** Integrate the translated audio back into the original video content.
* **Synchronization:** Apply mapping techniques to synchronize specific audio segments with corresponding video scenes, ensuring coherence and context.

**8. Software Implementation:**

* **User Interface:** Develop a user-friendly interface that guides users through the entire budding process.
* **Customization Features:** Implement features allowing users to specify language preferences, select translation nuances, and adjust text-to-speech characteristics.

**9. Quality Assurance:**

* **Testing:** Conduct rigorous testing to ensure accuracy in audio extraction, transcription, translation, and synchronization.
* **Linguistic Nuances:** Address potential issues related to linguistic nuances, religious terminology, and the coherence of audio-visual elements.

**10. User-Centric Customization:**

* **Fine-Tuning Options:** Empower users with the ability to fine-tune the budding process based on their linguistic, cultural, and religious preferences.
* **Iterative Feedback:** Gather user feedback on the translated content and implement iterative improvements to enhance user satisfaction.

**6. SYSTEM DESIGN**

**6.1 Introduction of Input Design:**

In the context of the video budding software, input design plays a crucial role in capturing and processing the raw data, which, in this case, involves linguistic and audio-visual elements. Consideration must be given to the following aspects:

**6.1.1 Input Devices:**

* Identify input devices such as audio extraction tools and language detection algorithms.
* Ensure compatibility with diverse video formats and linguistic variations.

**Quality of System Input:**

* Prioritize the quality of audio extraction to maintain the fidelity of religious expressions.
* Implement language detection algorithms sensitive to religious terms.

**Input Forms and Screens:**

* Design user-friendly interfaces for uploading videos and specifying language preferences.
* Incorporate customization options for users to define translation nuances and cultural preferences.

**Basic Design Principles:**

* Focus on straightforward and easy-to-fill input forms.
* Ensure consistency, simplicity, and attention to users' cultural and religious contexts.

**Objectives for Input Design:**

The objectives of input design for the video budding software are as follows:

**Data Entry and Input Procedures:**

* Design intuitive data entry procedures for linguistic and audio-visual elements.
* Ensure effective capturing of religious nuances in the input.

**Reducing Input Volume:**

* Implement efficient methods to reduce input volume without compromising on content richness.

**Source Documents and Data Capture:**

* Design source documents and methods for capturing linguistic and cultural nuances.

**Input Data Records and Validation:**

* Develop input data records considering diverse linguistic and religious expressions.
* Implement validation checks for accurate language identification.

**6.1.2 Output Design:**

The output design is paramount for delivering the translated and culturally nuanced videos. It involves identifying output types, ensuring controls, and presenting the information in an accessible format.

**Objectives of Output Design:**

* The objectives of output design for the video budding software are:

**Purposeful Output:**

* Develop an output design that serves the purpose of linguistic and cultural translation effectively.

**User Requirements:**

* Tailor output design to meet end users' requirements, considering diverse linguistic and religious contexts.

**Appropriate Quantity:**

* Deliver the right quantity of output videos with accurate linguistic and cultural representation.

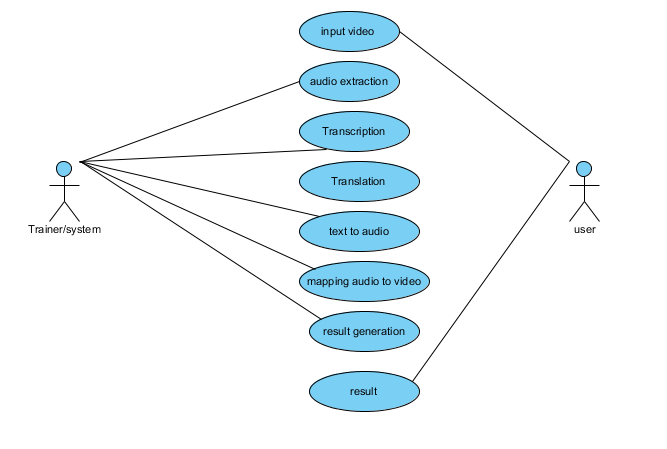
**Formatted and Timely Output:**

* Present output videos in the appropriate format, reflecting religious and cultural sensitivities.
* Ensure timely availability of output for effective decision-making.

**6.2 UML Diagrams:**

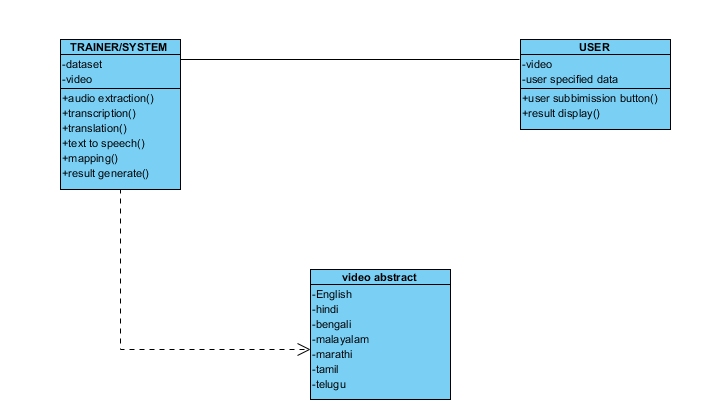
**6.2.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



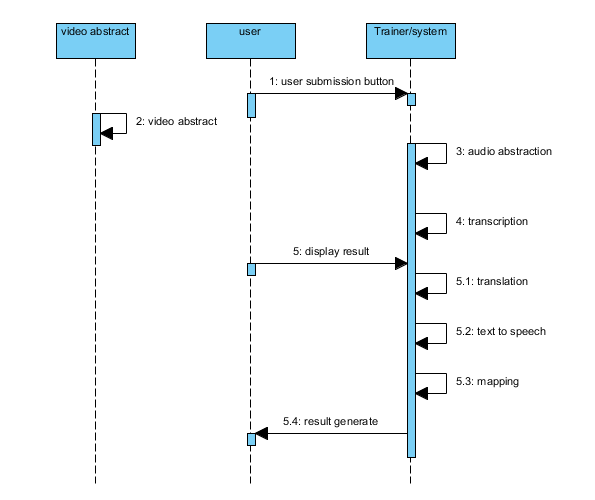
**6.2.2 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



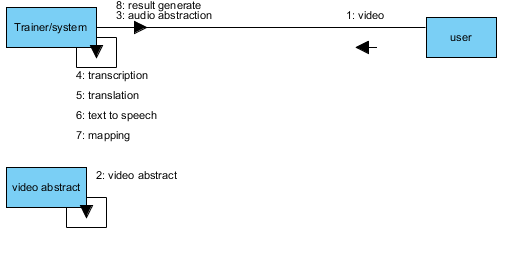
**6.2.3 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



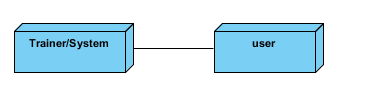
**6.2.4 Collaboration Diagram:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



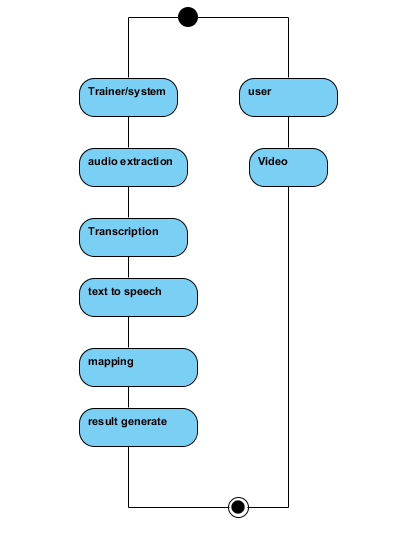
**6.2.5 DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



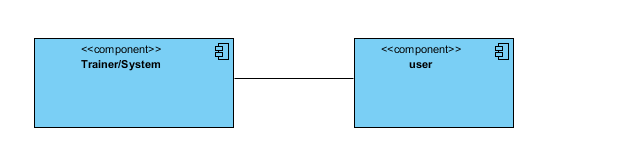
**6.2.6 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**6.2.7 Component diagram**:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by

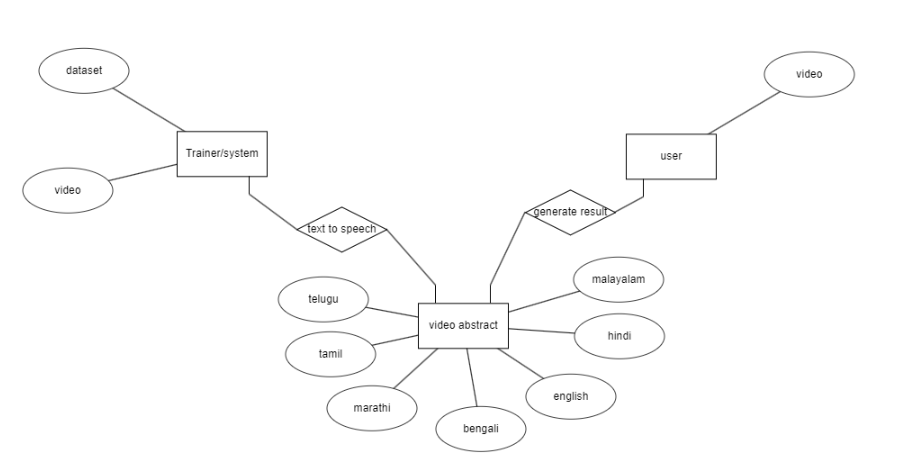


**6.2.8 ER DIAGRAM**

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

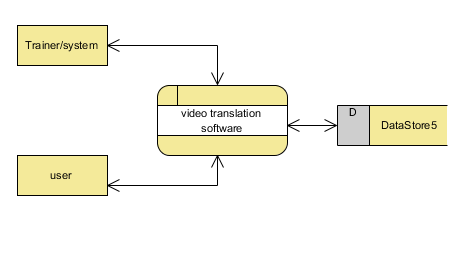
In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.



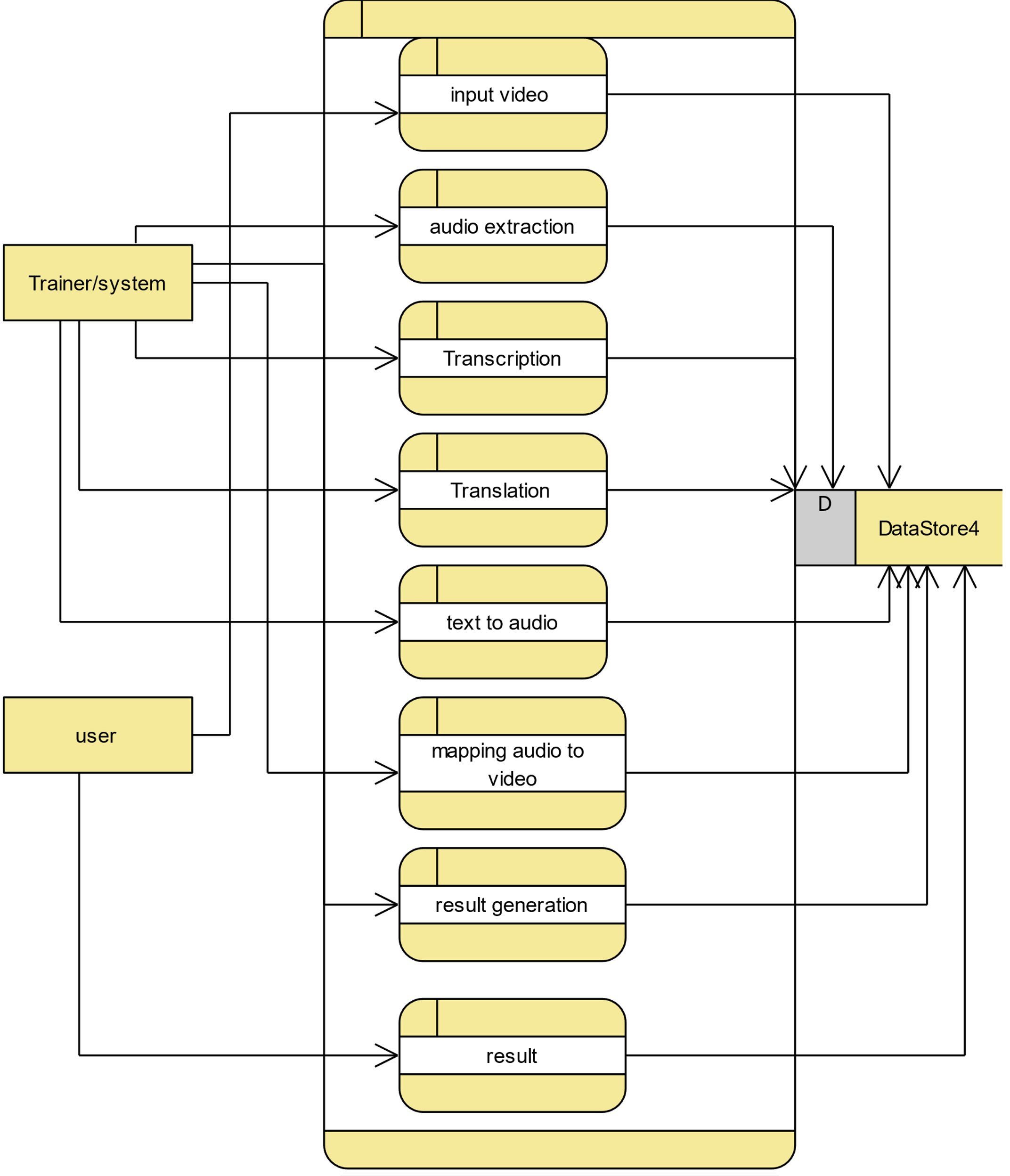
**6.3 DFD DIAGRAM**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

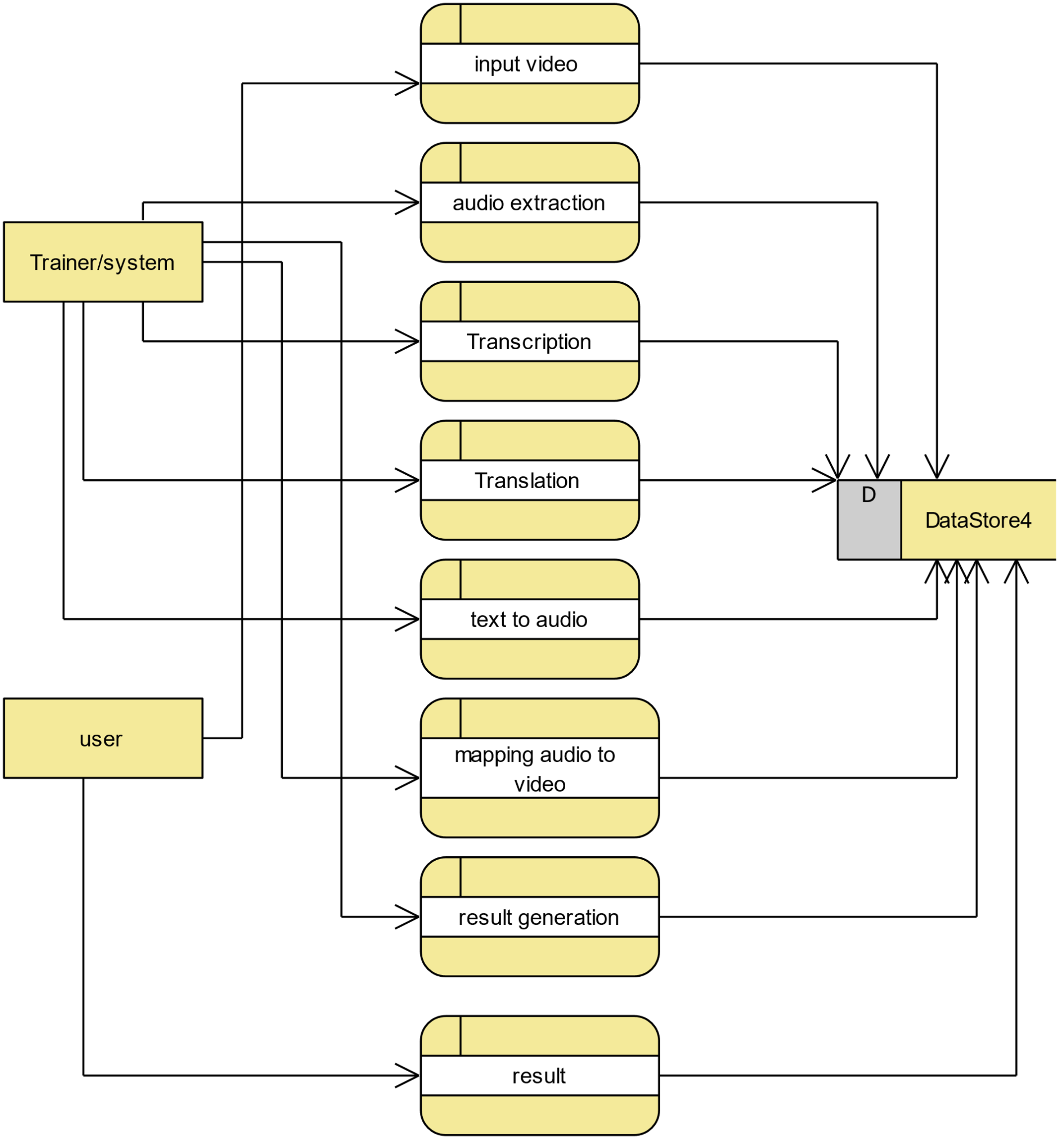
**Context Diagram:**



**DFD Level-1 Diagram:**



**DFD Level-2 Diagram:**



**7. IMPLEMENTATION AND RESULTS**

**1. System:**

**1.1 Data Collection:**

- Gather videos in English and corresponding scripts.

- Organize and store the data for further processing.

**1.2 Preprocessing:**

- Clean and preprocess the video and script data.

- Handle missing information and ensure data quality.

**1.3 Translation Integration:**

- Implement translation services for converting English scripts to Indian languages.

- Integrate translation outputs with corresponding video segments.

**1.4 Video Budding:**

- Develop algorithms for intelligently segmenting and budding videos.

- Ensure smooth transitions and coherence in the budding process.

**1.5 Quality Assurance:**

- Implement checks to maintain video and audio quality during the budding process.

- Address any issues related to translation artifacts or distortions.

**1.6 User Feedback Mechanism:**

- Integrate a feedback system for users to provide input on translation accuracy and video quality.

- Utilize feedback to enhance the translation and budding algorithms.

**2. User:**

**2.1 Video Upload:**

- Provide a user-friendly interface for uploading English videos.

- Ensure compatibility with various video formats.

**2.2 Language Selection:**

- Allow users to choose the desired Indian language for translation.

- Provide options for multiple language selections if needed.

**2.3 Translation Preview:**

- Offer users a preview of the translated scripts before finalizing the budding process.

- Ensure user satisfaction with the translation outputs.

**2.4 Budding Customization:**

- Allow users to customize the budding process based on preferences.

- Options may include choosing specific translation styles or segment durations.

**2.5 Output Access:**

- Enable users to access and download the translated and budded videos.

- Ensure a seamless viewing experience with appropriate playback controls.

**2.6 User Analytics:**

- Implement analytics to gather user preferences and usage patterns.

- Utilize analytics for continuous improvement and feature enhancements.

This modular structure ensures a systematic and user-centric approach to the video budding software, addressing both technical and user experience aspects.

**8. SYSTEM STUDY AND TESTING**

**8.1 Feasibility Study**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* Economical feasibility
* Technical feasibility
* Social feasibility

**Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**8.2 Types of Tests**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Functional testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**11. REFERENCES**

[1] Felix Stahlberg (2020): Neural Machine Translation: A Review. Journal of Artificial Intelligence Research (JAIR)

[2] Markus Freitag, Orhan Firat (2020): Complete Multilingual Neural Machine Translation. arXiv:2010.10239

[3] Alexandre Berard, Olivier Pietquin, Christophe Servan, Laurent Besacier (2016): Listen and Translate A Proof of Concept for End-to-End Speech-to-Text Translation. arXiv:1612.01744

[4] Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, Rishemjit Kaur (2023): Neural Machine Translation for Low-resource Languages: A Survey. arXiv