

A PROJECT REPORT ON

“Transformative Perspectives:AI-Enabled Dubbing Application for Multilanguages”

A Project report submitted in partial fulfillment of the requirement
for the award of degree of
BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

By

S.YASMIN (21AM1A0566)

S. JUFI (21AM1A0515)

K.BHUPATI (21AM1A0505)

A.SUPRIYA (21AM1A0549)

Under the Guidance of

Mr.V.Jeevan Kumar, M. Tech,
Assistant professor , Department of CSE



DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING

SVR ENGINEERING COLLEGE(AUTONOMOUS)

AYYALURU METTA, NANDYAL, KURNOOL (DIST.) – 518503.

(Approved by AICTE, New Delhi & Affiliated to JNTUA,
Anantapuram) ECE & CSE Dept are accredited by NBA.

ACADEMIC YEAR: 2021-2025



SVR ENGINEERING COLLEGE

(AUTONOMOUS)



Approved by AICTE: New Delhi, 2(F) & 12(B) Recognition by UGC Act.1956
Affiliated to JNTUA, Ananthapuramu
Accredited by NAAC & Accredited by NBA, ISO: 9001: 2015 Certified Institution
Ayyalur Metta, NANDYAL Dist. - 518 502, A.P.

ESTD : 2007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE VISION, MISSION AND CORE VALUES

VISION

To produce Competent Engineers with “A strong basics of Engineering Knowledge, A Mindset of Lifelong Learning and the necessary Complementary Skills needed to be Successful Professionals”.

MISSION

- To Impart Affordable and Quality Education to meet the needs of Society.
- To provide Ambient Teaching - Learning Environment, Adequate Infrastructure and Resources.
- To Empower the students through Complementary Skills such as Decision-making, Interpersonal, Ability to apply knowledge across disciplines, Ability to work effortlessly with others.

CORE VALUES

- Commitment to Continual Improvement
- Enhancement of Student Learning through Innovative Instructional Methods
- Collaboration with Industry and other Institutions for Mutual Benefits Pursuit of Excellence in all Activities.



SVR ENGINEERING COLLEGE

(AUTONOMOUS)

Approved by AICTE: New Delhi, 2(F) & 12(B) Recognition by UGC Act.1956
Affiliated to JNTUA, Ananthapuramu
Accredited by NAAC & Accredited by NBA, ISO: 9001: 2015 Certified Institution
Ayyalur Metta, NANDYAL Dist. - 518 502, A.P.



ESTD : 2007

DEPARTMENT VISION AND MISSION

VISION

To cultivate globally competent engineers and ethical leaders in Computer Science, driven by research-based innovation to meet the evolving demands of industry and society.

MISSION

- To establish a robust foundation in Computer Science, emphasizing software development and real-world applications.
- To propel innovation through strategic collaborations with industry, academia, and research organizations, leveraging a dynamic and state-of-the-art infrastructure.
- To cultivate professionalism, ethical integrity, and leadership, equipping students for lifelong learning and career success.

PROGRAM OUTCOMES (PO's):

PO Number	Gratitude Attributes	PROGRAM OUTCOME STATEMENTS
1	Engineering Knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex Engineering problems.
2	Problem Analysis	Identify, formulate, review research literature, and analyse Complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/Development of Solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	Conduct Investigations of Complex Problems	Ability to review research literature, use research methods to execute project and synthesize the problem to provide valid conclusions.
5	Modern Tool Usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6	The Engineer and Society	Apply reasoning informed by the contextual Knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	Environment and Sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11	Project Management and Finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	Life- long Learning	Recognize the need for, and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change..

Program Specific Outcomes (PSOs)

1. Develop innovative software solutions addressing industry and societal needs, advancing technological progress.
2. Utilize advanced tools and collaborate with industry, academia, and research to drive innovation.
3. Demonstrate professionalism, ethical leadership, and technical expertise for lifelong learning and societal contributions.

TITLE	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
Fake Profile Identification In Social Network Using Machine Learning And Nlp														

Signature of the guide

A PROJECT REPORT ON
**“TRANSFORMATIVE PERSPECTIVES:AI ENABLED
DUBBING APPLICATION FOR MULTILANGUAGES”**

Submitted in partial fulfillment of requirement For the award of degree of
BACHELOR OF TECHNOLOGY

IN
COMPUTER SCIENCE & ENGINEERING

By

S.YASMIN	21AM1A0566
S.JUFI	21AM1A0515
K.BHUPATHI	21AM1A0505
A.SUPRIYA	21AM1A0549

Under the Esteemed guidance of
Mr. V.Jeevan Kumar, M.Tech.

Assistant Professor



SVR ENGINEERING COLLEGE
(Autonomous)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(APPROVED BY A.I.C.T.E., NEW DELHI AFFILIATED TO J.N.T.UNIVERSITY, ANANTHAPURAMU)
ESTD -2007

AYYALURU (V), NANDYAL – 518503, NANDYAL Dist. A.P.,

Batch: 2021-2025

SVR ENGINEERING COLLEGE

(Autonomous)

(APPROVED BY A.I.C.T.E., NEW DELHI AFFILIATED TO J.N.T.UNIVERSITY, ANANTHAPURAMU)

NANDYAL Dist -518503, A.P

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certified that the project entitled, **“Transformative Perspectives: AI Enabled Dubbing Application for Multilanguages”** is being submitted by **S.YASMIN, S.JUFI, K.BHUPATHI, A.SUPRIYA** bearing RollNo **21AM1A0566, 21AM1A0515, 21AM1A0505, 21AM1A0549** in a partial fulfillment of the requirement for the award of the degree of **BACHALOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING, SVR ENGINEERING COLLEGE** is record of bonafide work carried out by them during the year 2024-2025 under our guidance and supervision.

Project Guide:

Mr. Jeevan Kumar, M. Tech
Assistant professor, Department of CSE

Head of the Department

M.NAGA MALLIKARJUNA REDDY, M. Tech
Associate professor, Department of CSE&HOD

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I wish to regard a deep sense of gratitude to my guide **Mr.V.Jeevan Kumar M.Tech**, Assistant Professor, **Department of Computer Science & Engineering, SVR ENGINEERING COLLEGE** who has guided and motivated with scholarly advice, meticulous care and deep interest throughout the project duration

I wish to acknowledge our sincere gratitude to **Mr.M.NAGA.MALLIKARJUNA REDDY M.Tech**, CSE Associate Professor, **Head of the Department of CSE, SVR ENGINEERING COLLEGE**, Nandyal for permitting us to do the project and giving all sorts of encouragement and scholarly guidance throughout the project work.

I express my sincere thanks and gratitude to **Dr. P. MALLIKARJUNA REDDY** Professor & Principal, **SVR ENGINEERING COLLEGE, Nandyal**, for having provided with all the facilities of the department to complete this dissertation work successfully.

We are greatly indebted to **Mr.S.DINESH REDDY, M.D & Mr.S.VENKATARAMI REDDY** garu, Chairman of the college for giving us the opportunity and moral support throughout the course.

Finally, I wish to express deep sense of gratitude to my family, friends and our college staff members and laboratory of Computer Science Department who helped us in completing this project work successfully.

S.YASMIN	21AM1A0566
S.JUFI	21AM1A0515
K.BHUPATHI	21AM1A0505
A.SUPRIYA	21AM1A0549

ABSTRACT:-

Dubbing serves as a critical element in rendering content accessible to global audiences, facilitating intercultural adaptation, multilingual marketing endeavors, and upholding the integrity of original intentions. Its essence lies in the substitution of the original soundtrack with translated or modified versions, thereby ensuring comprehension and resonance across diverse linguistic landscapes. The dubbing process encompasses several fundamental steps and considerations. Foremost among these is linguistic precision, which is indispensable for effectively conveying the intended message. Translators are tasked with capturing nuances, cultural references, and colloquialisms while synchronizing dialogue with lip movements to maintain authenticity. Furthermore, voice casting assumes a crucial role in achieving fidelity to the original work, necessitating the involvement of skilled individuals capable of embodying characters across cultural boundaries. From this perspective, video/audio dubbing emerges as an indispensable phase in the broader process of information dissemination, substituting translations for the original track to enable global accessibility. This paper underscores the imperative of preserving authenticity by delineating the foundational phases and factors inherent in global content localization and dissemination efforts, thereby ensuring the retention of the intended message across linguistic and cultural frontiers.

CONTENTS

Chapter 1: Introduction

1.1 About The Project	1
1.2 Existing System	2
1.2.1 Disadvantages	3
1.3 Proposed System	3
1.3.1 Advantages	3

Chapter 2: Literature Survey 4-6

Chapter 3: System Requirements Specifications 7-8

3.1 Requirement Analysis	8
3.2 Hardware Requirements	9
3.3 software Requirements	9

Chapter 4: System Study 10

4.1 Feasibility Analysis	10
4.2 Economical Feasibility	10
4.3 Technical Feasibility	10
4.4 Social feasibility	11

Chapter 5: System Design 12

5.1 Architecture	12
5.2 Uml Diagrams	13
5.2.1 Uml Diagrams Introduction	13-16
5.2.2 Class Diagram	17
5.2.3 Use Case Diagram	18
5.2.4 Sequence Diagram	19
5.3 Activity Diagrams	19-20

5.3.1 Package Diagram	21
5.3.2 Profile Diagram	22
5.4 Algorithms	
5.4.1 movie.py	23-24
5.4.2 Speech Recognition	24
5.4.3 Gtts(Google Text-to-Speech)/GoogleTrans	24
5.4.4 pyttsx3 Voice Synthesis	24-25
5.4.5 pyaudio Audio Mixing	25
5.4.6 Google Web Speech API	25
Chapter 6: System Environment	25
6.1 Python	25
6.1.1 History Of Python	25
6.1.2 Python Features	26
6.1.3 Installation Of Python	27-34
6.2 Django	35
6.2.1 Django History	35
6.2.2 Bundled Application	35
6.2.3 Extensibility	35
6.2.4 Server Arrangements	36
6.2.5 Features Components	36-38
6.2.6 Model Layer	38
6.2.7 View Layer	38
6.2.8 Template Layer	39
6.2.9 Security	39
6.2.9 Common Web Application Tools	40
Chapter 7:Implementation	46
7.1 Modules And Description	46
7.1.1 Service Provider	46

7.1.2 Remote User	46
Chapter 8: Sample Code	47-71
Chapter 9: System Testing	56
9.1 Types Of Testing	56
9.1.1 Unit Testing	56
9.1.2 Integration Testing	56
9.1.3 Acceptance Testing	57
9.2 Testing Methodologies	57
9.2.1 Unit Testing	57
9.2.2 Integration Testing	57
9.2.3 User Acceptance Testing	58
9.2.4 Output Testing	59
9.2.5 Validation Testing	59
Chapter 10: Screenshots	
Chapter 11: Future Enhancement	
Chapter 12: Conclusion	
Chapter 13: Referenses	

LIST OF FIGURES

FIGURE TITLE	PAGENO
1. System Architecture	12
2. Remote User	13
3. Service Provider	14
4. Usecase Diagram	16
5. Class Diagram	17
6. Sequence Diagram	18
7. Data Flow Diagram	19

LIST OF SCREENSHOTS

SCREENSHOT TITLE	PAGENO
1. Home Page	56
2. Remote User Registration	56
3. Remote User Login	57
4. Remote User Details	57
5. Predict Profile Identification Status	58
6. Service Provider Login	58
7. Service Provider Interface	59
8. Fake Profile Dataset Trained And Tested Results	59
9. View Trained And Tested Accuracy In Bar Chart	60
10. View Trained And Tested Results In Line Chart	60
11. View Trained And Tested Results In Bar Chart	61
12. View Online Profile Judgement Ratio	61
13. View Prediction Of Online Judgement Fake Profile	62
14. Download Predicted Datasets	62
15. View User Identity Ratio Results In Line Chart	63
16. View User Identity Ratio Results In Bar Chart	63

CHAPTER 1

INTRODUCTION

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

1.INTRODUCTION

1.1 About the project:

Dubbing serves as a critical element in rendering content accessible to global audiences, facilitating intercultural adaptation, multilingual marketing endeavors, and upholding the integrity of original intentions. Its essence lies in the substitution of the original soundtrack with translated or modified versions, thereby ensuring comprehension and resonance across diverse linguistic landscapes. The dubbing process encompasses several fundamental steps and considerations. Foremost among these is linguistic precision, which is indispensable for effectively conveying the intended message. Translators are tasked with capturing nuances, cultural references, and colloquialisms while synchronizing dialogue with lip movements to maintain authenticity. Furthermore, voice casting assumes a crucial role in achieving fidelity to the original work, necessitating the involvement of skilled individuals capable of embodying characters across cultural boundaries.

From this perspective, video/audio dubbing emerges as an indispensable phase in the broader process of information dissemination, substituting translations for the original track to enable global accessibility. However, the traditional process is fraught with challenges, including high costs, prolonged production times, and dependency on human expertise. As technology advances, artificial intelligence (AI) emerges as a transformative solution to these challenges. AI-enabled dubbing software promises to revolutionize multilingual content localization by automating key elements of the dubbing workflow while preserving the essence of the original content.

This paper explores the evolution and potential of AI-enabled dubbing solutions in the context of multilingual content localization. It delves into the challenges associated with traditional dubbing practices, reviews relevant works and existing systems, and outlines the significance of adopting AI in this domain. By addressing the intricacies of linguistic and cultural adaptation, the paper underscores the transformative role of AI in facilitating efficient, high-quality, and scalable dubbing solutions.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

India is a profoundly multi-lingual nation, boasting a rich tapestry of living languages, including 18 major ones that are constitutionally recognized. In our increasingly interconnected global society, effective communication fosters cross-cultural understanding. Imagine reaching millions beyond your own mother tongue, expanding your business or sharing ideas with an entirely new audience. Translation, here, is in most demand, may it be audio/video format. Dating back, traditional translation and caption generation methodologies were time consuming and involved skilled bunch of human translators. But with the raise of technological advancements, AI, ML to NLP and Cloud service approaches have been contemplated since late 20s. Any verbal, non- verbal form of information requires a sophisticated procedural alignment comprising various techniques; namely (a) ASR -Automatic Speech Recognition, (b) NLP- Natural language Processing, (c) MT-Machine Translation etc., for extraction and revival of information, further aiding audio/video translation, facilitating seamless understanding and connection among diverse communities. The sole objective of the paper is to enable an ordered conversion of English language video/audio into Indian regional language of choice with a supportive high quality, systematic voice modulation, lip synchronization and accurate translation of speech in real time. Ensures an intuitive interface with customizable preferences for generation of localized video content.

1.2 EXISTING SYSTEM

The existing system for dubbing software in multilingual content localization typically relies on manual processes involving teams of translators, voice actors, and sound engineers. These professionals work collaboratively to translate and record the dialogue in the target language, matching the timing and tone of the original content. The audio is then carefully synchronized with the visual elements, ensuring that lip movements align with the dubbed speech. While effective, this process is time-consuming, costly, and often lacks scalability, especially when dealing with large volumes of content across multiple languages. Additionally, the quality of the final product is highly dependent on the skill and coordination of the involved professionals, making it challenging to maintain consistency across different projects. As the demand for multilingual content grows, there is a pressing need for more efficient, automated solutions that can deliver high-quality dubbing while reducing the time and resources required.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

1.2.1 DISADVANTAGES:

- Time-Consuming Process
- High Costs - Inconsistent Quality
- Dependency on Human Resources

1.3 PROPOSED SYSTEM

The proposed system leverages AI technologies to address the limitations of traditional dubbing workflows. By integrating advanced NLP models, speech synthesis, and lip-sync algorithms, the system automates key aspects of the dubbing process. AI-powered translation tools ensure linguistic accuracy and cultural relevance, while TTS and voice cloning technologies replicate natural speech patterns and emotional tones. Deep learning-based lip-sync technology enables precise synchronization between audio and video, enhancing the authenticity of the localized content. This approach not only reduces production time and costs but also ensures scalability and consistency across projects. The proposed system represents a significant step toward achieving fully automated, high-quality dubbing solutions for multilingual content localization.

1.3.1. ADVANTAGES:

1. Enhanced Global Reach
2. AI-Powered Accuracy & Context Awareness
3. Realistic Voice Synthesis
4. Time and Cost Efficiency
5. Personalization and Inclusivity
6. Scalability
7. Educational & Social Impact

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SERVEY

The literature on AI-enabled dubbing spans several interconnected domains, including NLP, speech synthesis, and content localization. A key focus has been on developing neural networks capable of handling the intricacies of language translation. Transformer models, such as BERT and GPT, have been widely studied for their ability to understand context and generate high-quality translations. These models form the backbone of many AI dubbing systems, enabling accurate and contextually relevant language adaptations.

Speech synthesis technologies have also garnered significant attention in the literature. Researchers have explored the use of text-to-speech (TTS) systems that can generate natural-sounding voices in multiple languages. Advances in voice cloning have enabled the replication of specific voice characteristics, allowing AI systems to mimic the original actors' voices while adapting to the target language. This capability is particularly valuable for maintaining consistency and character identity in dubbed content.

Lip-syncing is another critical area of research. Studies have investigated the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to analyze video frames and generate synchronized speech. These approaches aim to achieve real-time synchronization, reducing the manual effort required for audio-visual alignment. The combination of lip-sync technology with speech synthesis has been shown to enhance the overall quality of AI-dubbed content.

The role of cultural adaptation in dubbing has also been extensively examined. Researchers have highlighted the need for AI systems to account for cultural nuances, idiomatic expressions, and regional variations in language use. Machine learning models have been trained on diverse datasets to improve their ability to handle such variations, ensuring that the localized content resonates with target audiences.

Furthermore, ethical considerations in AI-enabled dubbing have been a recurring theme in the literature. Issues such as voice cloning consent, data privacy, and the potential loss of human jobs have been discussed. These concerns underline the need for responsible AI development and deployment in the dubbing industry.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Lastly, user experience and audience reception have been the subject of empirical studies. Researchers have conducted surveys and experiments to assess viewers' perceptions of AI-dubbed content. Findings indicate that while AI-enabled dubbing is generally well-received, audiences often prefer content that retains emotional authenticity and cultural relevance, highlighting areas for further improvement.

CHAPTER 3

SYSTEM ENVIRONMENT

3. SYSTEM ENVIRONMENT

3.1 REQUIREMENT ANALYSIS:

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

Functional Requirements:

- Graphical User interface with the User.

3.2 Software Requirements:

For developing the application the following are the Software Requirements:

1. Python
2. Django

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

3.3 Hardware Requirements:

Processor	: Intel Core i3 or higher – 10th Gen or higher
RAM	: 8 GB (minimum), 16 GB recommended
Storage	: 256 GB SSD (or higher)
Keyboard	: Standard wired or wireless keyboard
Mouse	: Multi-button mouse (wired/wireless)
Monitor	: 1080p Full HD (or higher)

3.4 Software Requirements:

Operating system	: Windows 11.
Coding Language	: Python.
Front-End	: Python.
Back-End	: Django.
Designing	: HTML, CSS, JavaScript.

CHAPTER 4

SYSTEM STUDY

4. SYSTEM ANALYSIS

4.1 FEASIBILITY STUDY:

The feasibility of the project is analyzed 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**

4.2 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.

4.3 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.

4.4 SOCIAL FEASIBILITY:

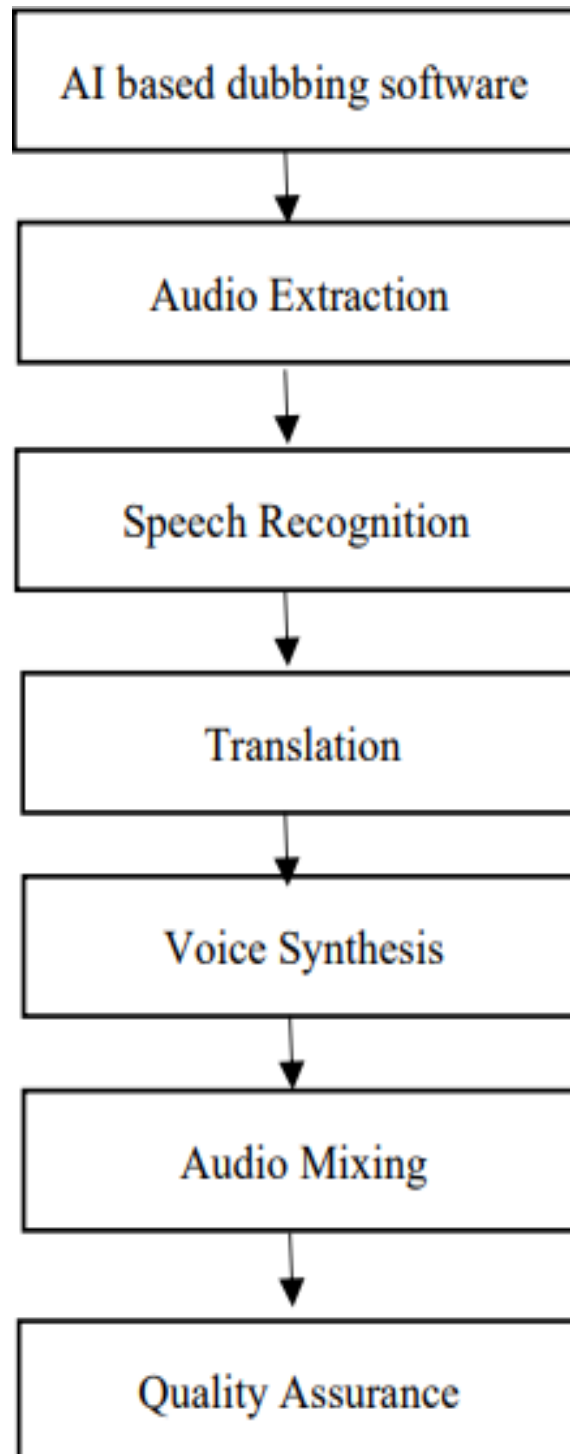
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends 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.

CHAPTER 5

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 ARCHITECTURE:



5.2 UML Diagrams:

5.2.1 UML Diagrams Introduction:-

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

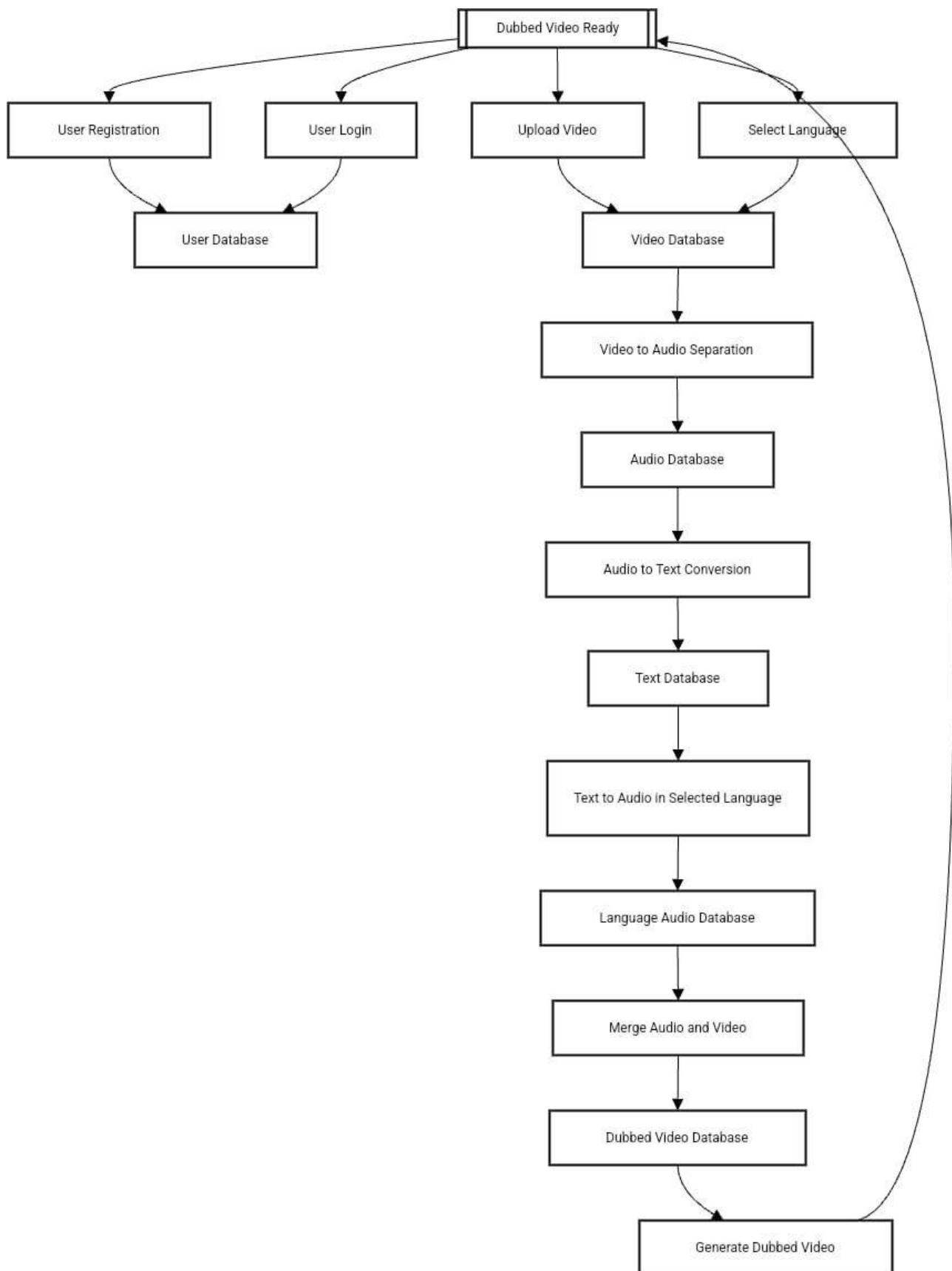
The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

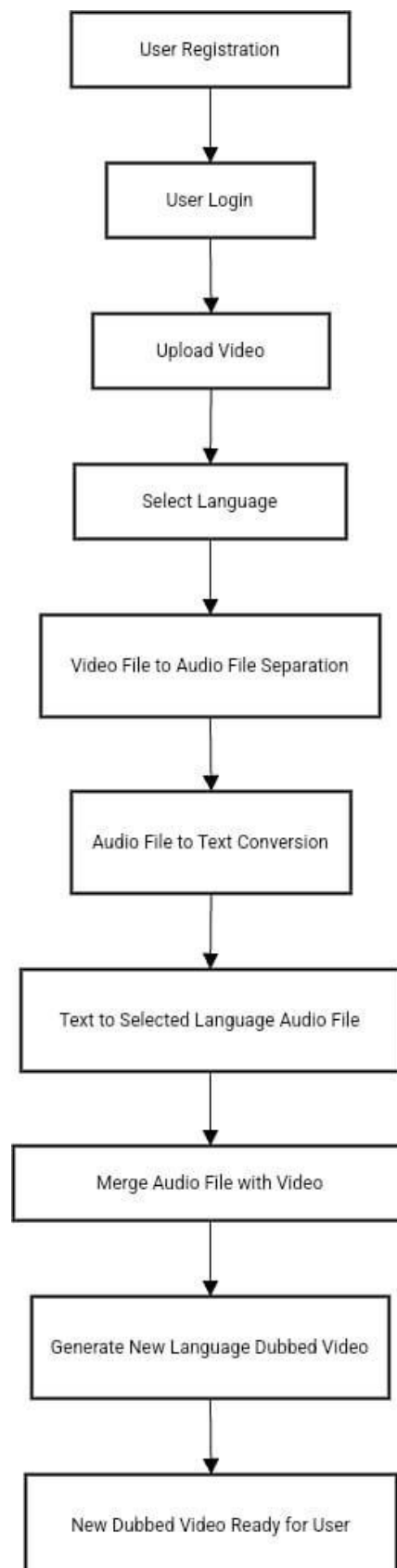
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

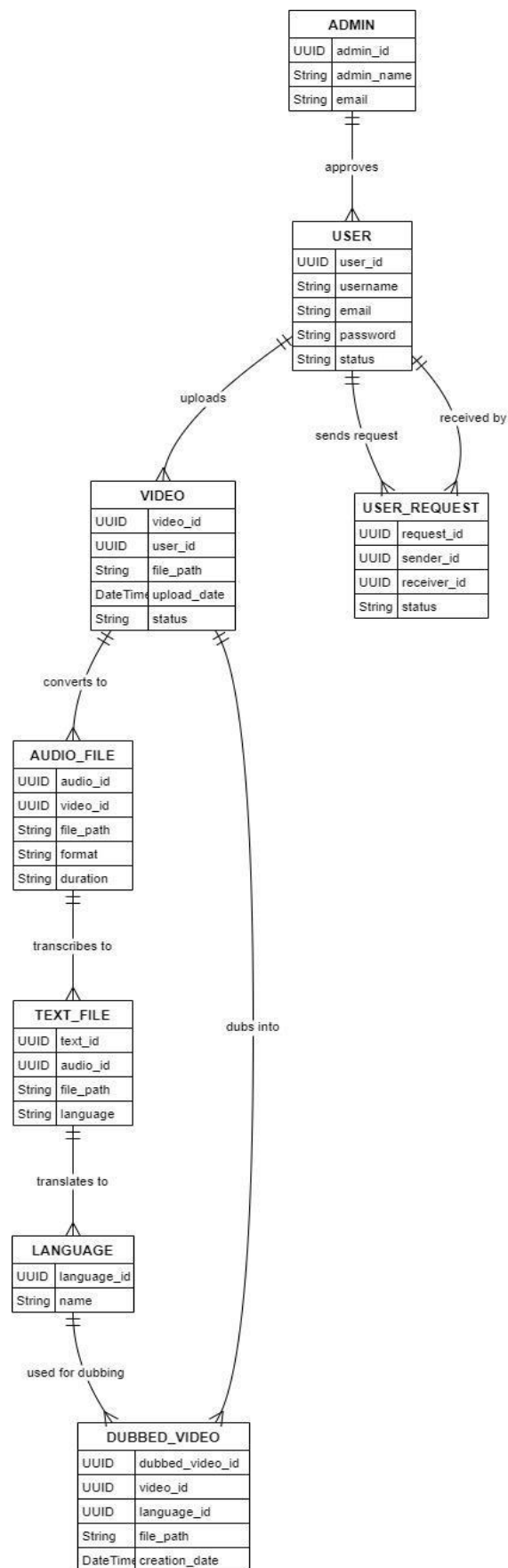
TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



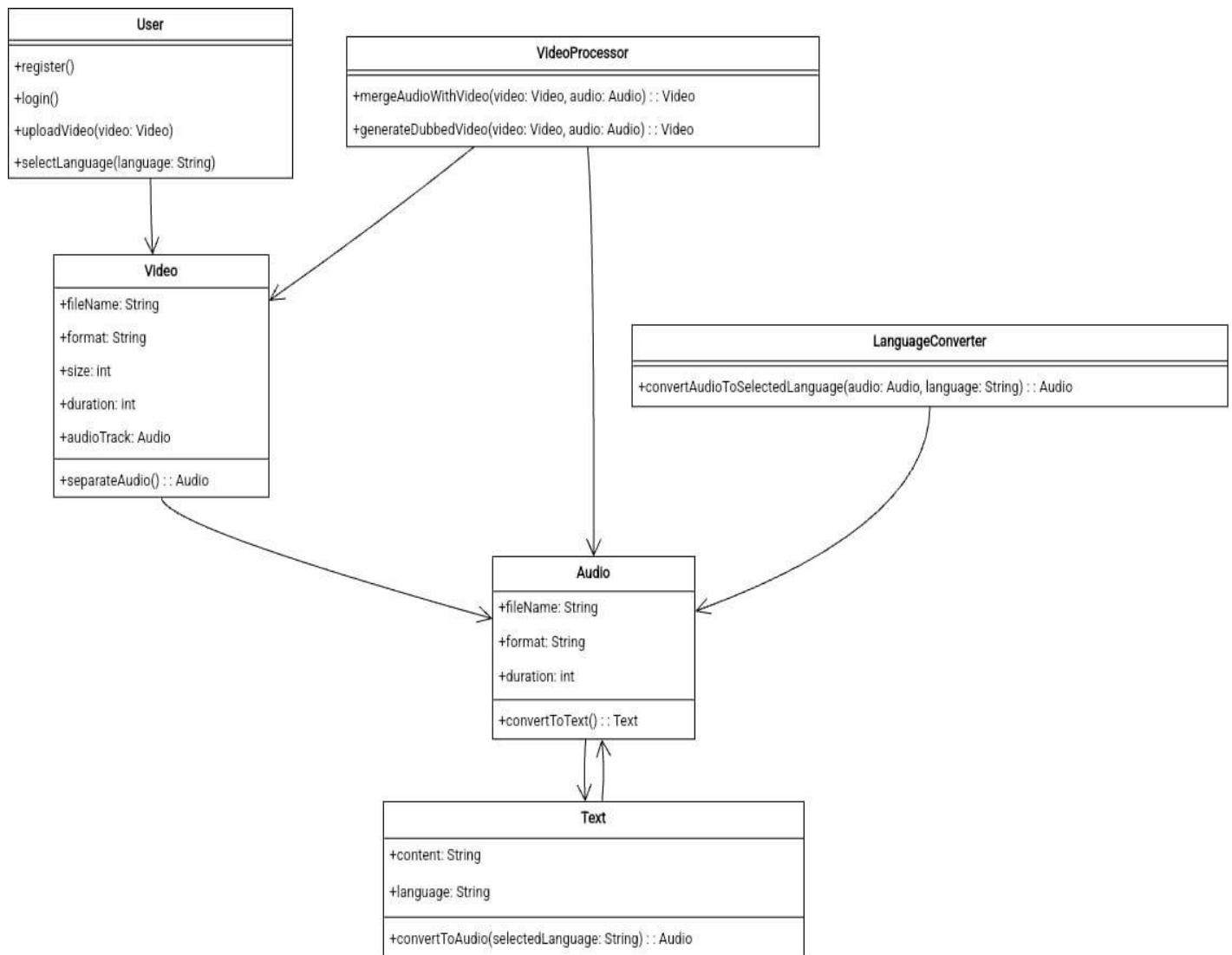
TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

CLASS DIAGRAM

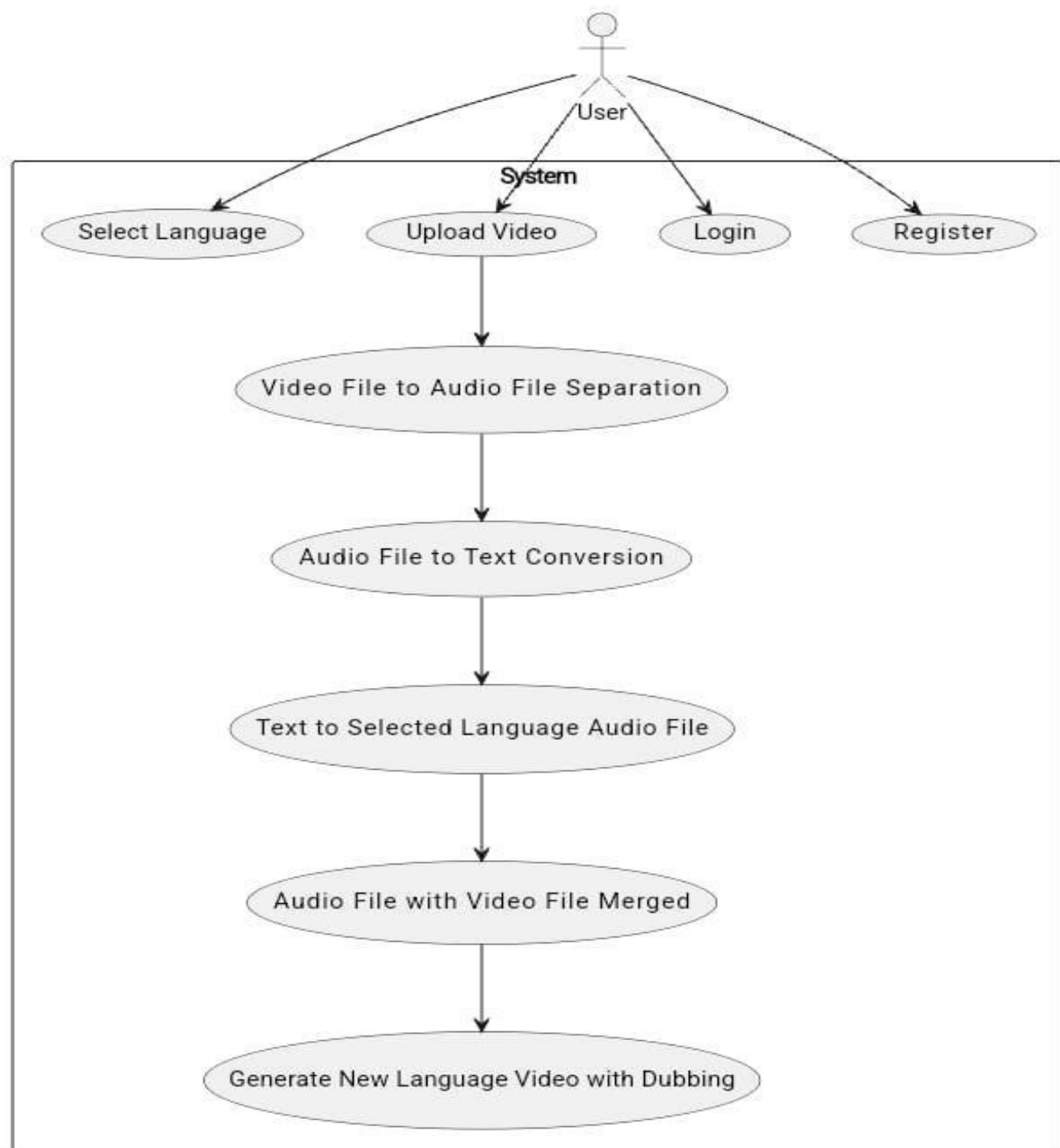
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains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



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

5.2.2 USE CASE DIAGRAM:

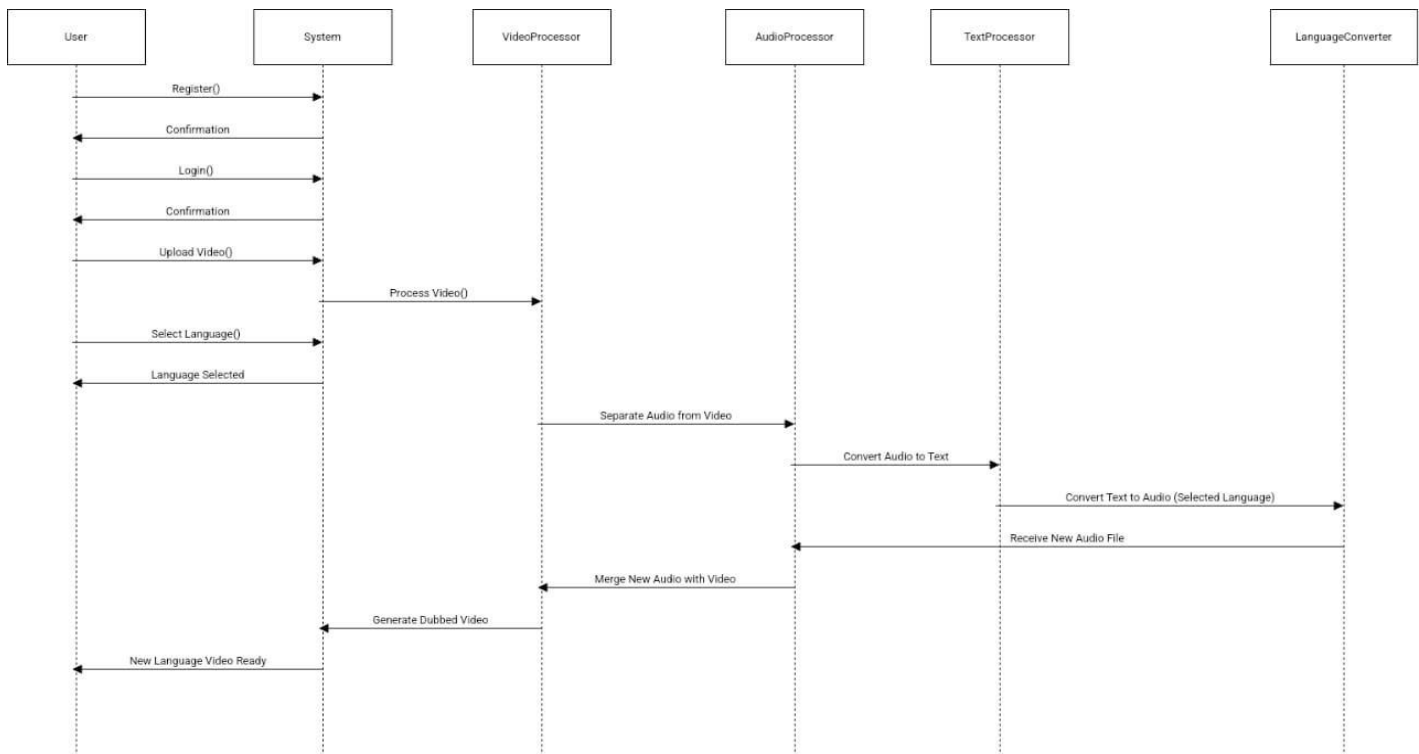
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

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.

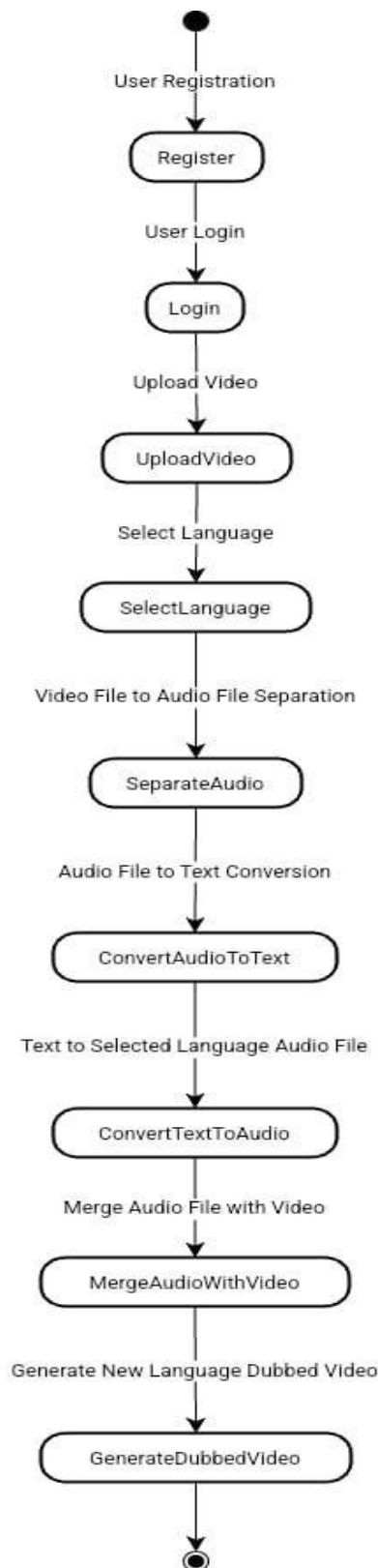


Activity diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions[1] with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.[2][3] Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.[citation needed]Activity diagrams are graphical representations of workflows of stepwise activities and actions[1] with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.[2][3] Although activity diagrams primarily show the overall flow of

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

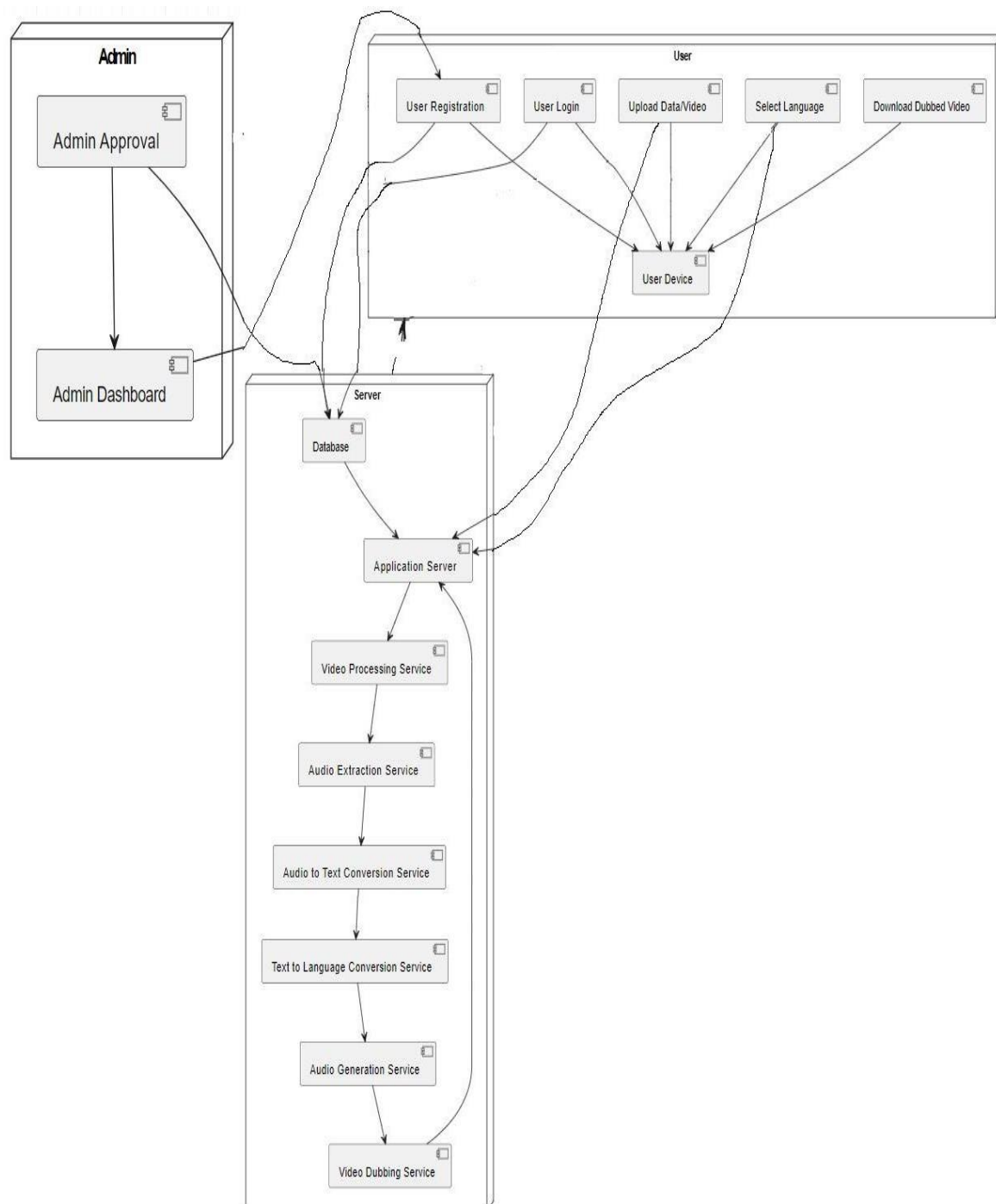
control, they can also include elements showing the flow of data between activities through one or more data stores.[citation needed]



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Package Diagram

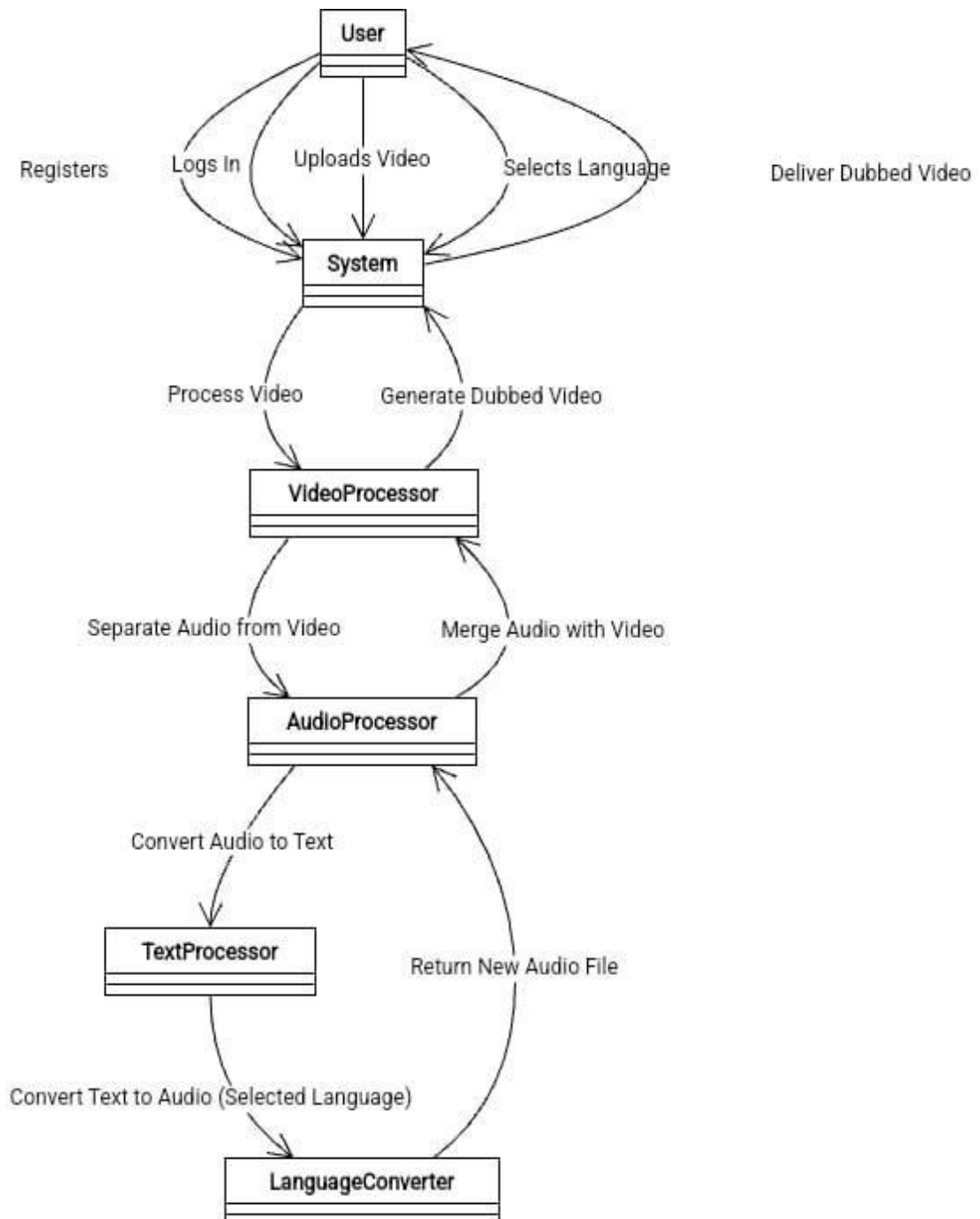
Package diagram is UML structure diagram which shows structure of the designed system at the level of packages. The following elements are typically drawn in a package diagram: package, packageable element, dependency, element import, package import, package merge.



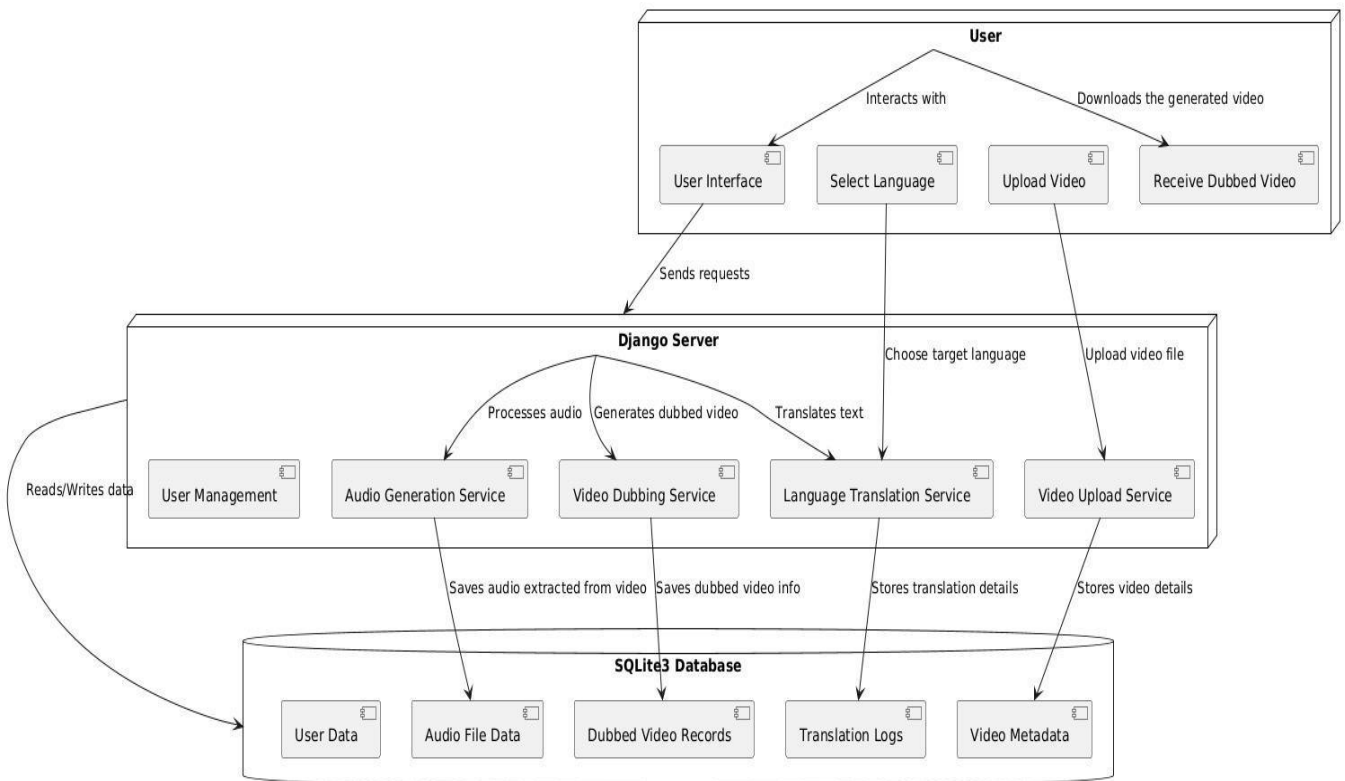
TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Profile Diagram

A Profile diagram is any diagram created in a «profile» Package. Profiles provide a means of extending the UML. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components



TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



5.3 Modules Description:

5.3.1 A. movie.py

An audio extraction tool is software application, specifically to separate or extract audio from various file types, including video files and audio-visual media. These tools enable users to isolate and save the audio component in a standalone format, typically in formats like MP3, WAV, or other common audio file types. The dubbed audio is reviewed and evaluated for accuracy, lip syncing, and overall quality. Any necessary adjustments or corrections are made to ensure a high-quality dubbing output..

Step 1. If all the objects in S belong to the same class, for example C_i , the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O_1, O_2, \dots, O_n . Each object in S has one outcome for T so the test partitions S into subsets S_1, S_2, \dots, S_n where each object in S_i has outcome O_i for T . T becomes the root of the decision tree and for each outcome O_i we build a subsidiary decision tree by invoking the same procedure recursively on the set S_i .

5.3.2 B. Speech recognition

Speech recognition, sometimes referred to as voice recognition or automatic speech recognition (ASR), is a technique that transcribes spoken words into written language. Accurately identifying and transcription of human speech is the main objective of speech recognition systems. Speech recognition systems translate spoken words into written text by analyzing audio data, finding pattern in spoken languages, and applying machine learning techniques and algorithms.

5.3.3 C. gTTS (Google Text-to-Speech)/GoogleTrans)

GoogleTrans can be employed to translate the original script or dialogue from one language to another. The translated text can be processed to ensure accuracy and naturalness in the target language. Once the translated text is available, gTTS can be utilized to generate synthetic speech for the translated content. Users can specify parameters such as voice type, speed, and intonation to match the desired characteristics for the dubbed content. The generated speech can then be synchronized with the corresponding video or audio segments to create the dubbed version of the content.

5.3.4 D. pyttsx3 Voice Synthesis:

The translated text is converted into spoken words in the target language using voice synthesis technology. This step generates audio output which sounds like a native speaker of the target language.

5.3.5 E. pyaudio Audio Mixing:

The software combines the original audio with the synthesized voice in the target language, adjusting the timing and volume levels to create a seamless dubbed audio track.

5.3.6 F. Google Web Speech API:

The Google Web Speech API, now part of the broader Web Speech API, enables developers to integrate speech recognition capabilities into web applications. It allows users to interact with websites using voice commands and speech-to-text transcription. Developers can access this feature through JavaScript APIs, facilitating voice-driven user experiences across various platforms and devices[8].

CHAPTER 6

SYSTEM ENVIRONMENT

6. SYSTEM ENVIRONMENT

6.1 Python (programming language):

Python is a **high-level, interpreted, interactive** and **object-oriented scripting language**. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

6.1.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

6.1.2 PythonFeatures

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

6.1.3 installation of python:-

Download a Python installation kit from the web

Visit the web page <https://www.python.org/downloads> .

Near the top of the page, it will say “Download the latest version of Python for Windows.” Just

under that, there will be two bright yellow buttons. One is for a version of Python 2, and will say

something like “Download Python 2.7.14.” Do not use that. Instead, find the one labeled

“Download Python 3.6.4” (or something similar), which is for Python 3.

Click that button, and you will be prompted to download the file python-3.6.4.exe to your Desktop

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet [here](#).The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>



Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.








TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	 Download	Release Notes
Python 3.6.9	July 2, 2019	 Download	Release Notes
Python 3.7.3	March 25, 2019	 Download	Release Notes
Python 3.4.10	March 18, 2019	 Download	Release Notes
Python 3.5.7	March 18, 2019	 Download	Release Notes
Python 2.7.16	March 4, 2019	 Download	Release Notes
Python 3.7.2	Dec. 24, 2018	 Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

TRANSFORMATIVE PERSPECTIVES: AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPU
Clipped source tarball	Source release		68111671e5b2db4ae77b9ab01b7079be	23017663	5xG
XZ compressed source tarball	Source release		d33e4aae6097051c3eca45ee3604803	17131432	5xG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6a28b4fa7563daf1a442cbafce08e6	34898416	5xG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	3dd905c38217e457738f5eaa936b2a3f	28882845	5xG
Windows help file	Windows		d63999573a2c5882ac58ade6b47c02	8131761	5xG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64/x64	9b09c3cf5d5e0b9afce2154a40728a2	7504391	5xG
Windows x86-64 executable installer	Windows	for AMD64/EM64/x64	a702b4b0a076d9dc05c3a583e563400	26883988	5xG
Windows x86-64 web-based installer	Windows	for AMD64/EM64/x64	29c31c908bdc73a8fe51a3b031b4b22	1362904	5xG
Windows x86 embeddable zip file	Windows		9fab18d178a1879fda9412257413bd8	6741628	5xG
Windows x86 executable installer	Windows		33c00c2942a54446a3d9d5147e394789	25663848	5xG
Windows x86 web-based installer	Windows		1b670cf5d117df82c30983ea371d87c	1324608	5xG

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

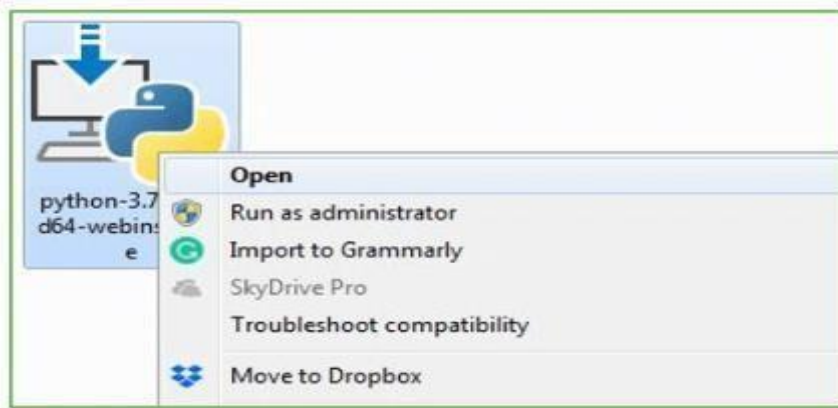
Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



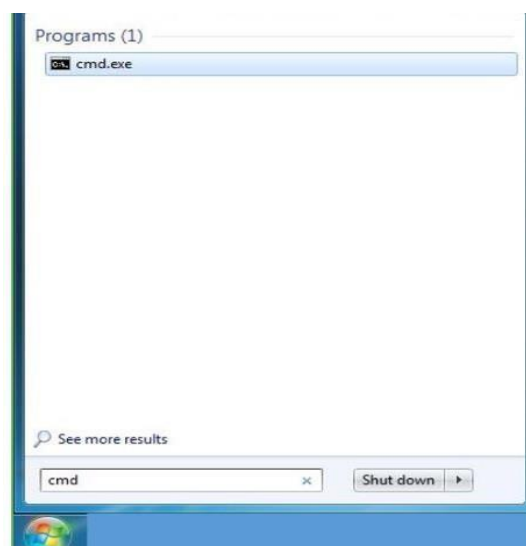
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

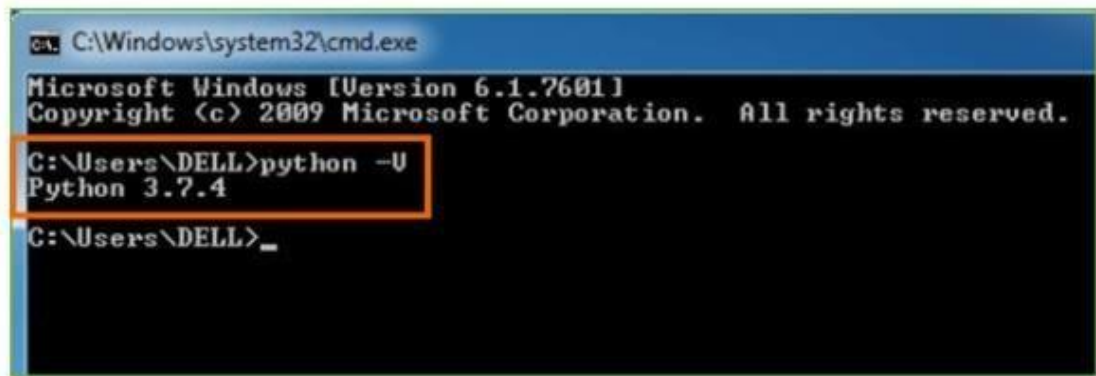
Step 2: In the Windows Run Command, type “cmd”



TRANSFORMATIVE PERSPECTIVES: AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python -V and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

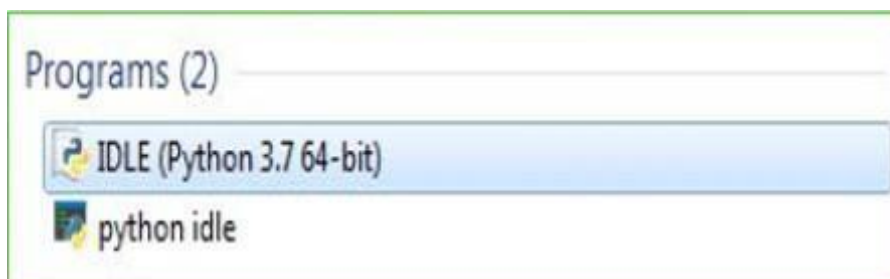
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

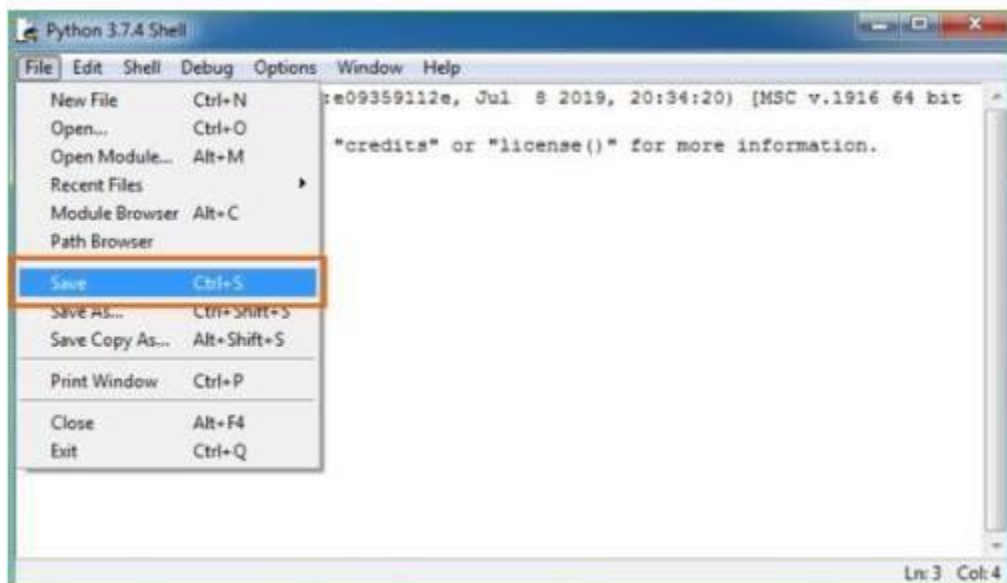
Step 2: In the Windows Run command, type “python idle”



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

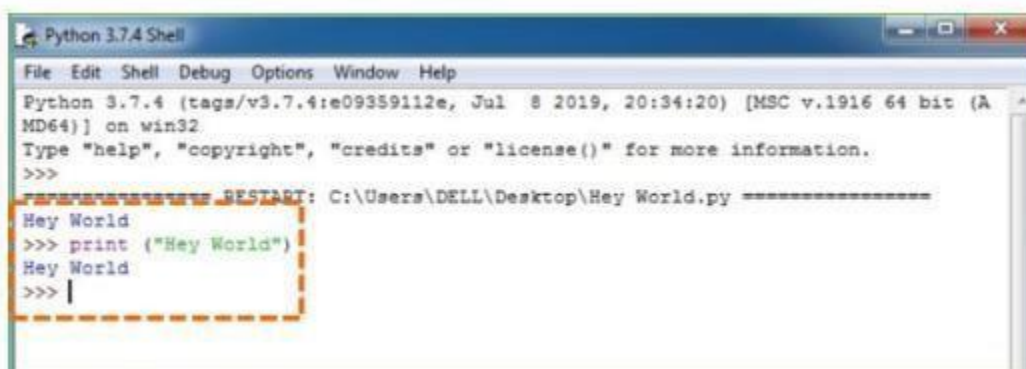
TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print (“Hey World”) and Press



TRANSFORMATIVE PERSPECTIVES: AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

6.2.1 Digo History:

naming of the framework.

In June of 2008, it was announced that a newly established organisation called the Django Software Foundation (dsf) will be responsible for the ongoing maintenance of django in the years to come.

6.2.2 Bundled Application

- The most frontend transmission further packets a variety of implementations inside its "contrib" pkg, incl:
 - An internet key scheme.
 - The energetic regulatory ui (user.
 - Tools as a generating rss and atom syndication needs to feed.
- A "sites" form that enables each laravel configuration versus run different sites on the internet, each of which has their very own applications.
- Tools such as generating google redirect.
- Built-in countermeasures for cross-site proposal fake, cross-site coding, sql inoculation, password cracking and side simple web targets, much of one another decided to turn on to by fallback.
- A blueprint regarding creating gis applications.

6.2.3 Extensibility:

Backend's setup system would allow 3rd person protocol to still be jammed into the a project status, if something that chooses to follow it and reusing app conventions. More often than 2 % of the population packages are readily accessible to increase a framework's earliest actions, proposing solutions of between problems the unique instrument didn't counteract: enrollment, scour, api provision as well as utilization, cms, and so on.

This syntax highlighting is, however, remedied besides inbuilt components' dependency relationships. While backend worldview suggests external entity, this same framework filtration but instead labels surmise another motor execution, but both this same permissioning or manager grouped applications which require the use

of internal ORM. That neither of these filtration systems as well as grouped applications were also compulsory by law complete operated some one Godot venture, and although reusing Android generally rely over one another, empowering development companies to maintain to use the formal layer so that you can advantage entirely from software ecosystem.

6.2.4 Server Arrangements:

Godot can be run in a variety of contexts, including associative operations with Apache, Nginx with WSGI, Gunicorn, or Cherokee with Flup (a NumPy module). The ability to set up a FastCGI server is another requirement for frontend, which simplifies its use in front of any web server that supports FastCGI (including Lighttpd and Hiawatha).

Web servers that adhere to the WSGI standard may also be used. PostgreSQL, MySQL, MariaDB, and SQLite are all database backends that work with Django. Additionally, Oracle is supported. Django-mssql allows access to Microsoft SQL Server on Windows-based platforms. Similarly, IBM DB2, SQL Anywhere, and Firebird all have external backends. Django's DjangoNonRel fork caters to non-relational DBMSs like MongoDB and Google App Engine's data store.

Django can be used with the Python programming language on any Java EE application server, including GlassFish and JBoss. In this case, JDBC drivers for database connectivity cannot be used without installing DjangoJython. This package can also provide the functionality necessary to compile Django into a deployment-ready WAR file.

One of the available frameworks, Django version 1.x.x, has built-in support in Google App Engine.

6.2.5 Features Components

Despite the fact that it uses its own terminology, such as calling the objects that are capable of creating HTTP replies "views," the core Django framework may be seen as an example of an MVC design. It is made up of an object-relational mapper (ORM) that acts as a mediator between data models (specified as Python classes) and a relational database ("model"), a

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

system for processing HTTP requests that is equipped with a web templating system ("view"), and a controller that is based on regular expressions and handles url dispatching ("controller") respectively.

In addition, the core framework provides a web server that is both lightweight and independent, which can be used for both development and testing.

a system for serialising and validating forms that can convert between html forms and values that are appropriate for storing in a database.

a method of creating programmes called templates that takes ideas from object-oriented programming and applies them in the form of inheritance.

a caching framework that allows the user to choose from a number of different cache techniques.

support for middleware classes that may carry out individualised functions and intervene at different points of the request processing process.

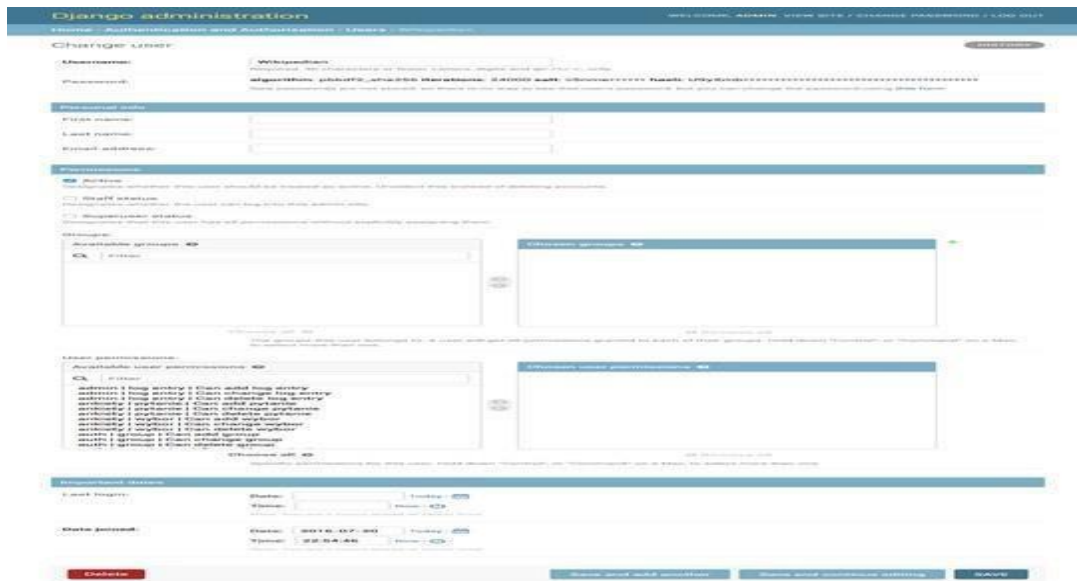
an application's own internal dispatcher system, which enables different parts of the programme to communicate with one another about events via pre-defined signals.

a mechanism for internationalisation, which includes translations of django's own components into a number of other languages.

a serialisation system that is able to build and read representations of django model instances in xml and/or json.

a method for enhancing the functionalities provided by the template engine. Provides an interface to the built-in unit test framework that is included with python.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES



Django Application

6.2.6 Model Layer:

- backend would provide some kind layer mainly (the —models) as a formulating as well as tricking the information anyway someone web server.
- learn more about this and following table:
- models.
- querysets.
- model incidences.
- migrations.
- advanced database works.
- legacy databases.

6.2.7 View Layer:

The "views" concept is one that is available in the Django framework, and it is used to encapsulate the logic that is accountable for the processing of a user's request and for returning the response. Through the following links, you may learn all you need to know about views:

- The fundamentals of urlconf files.
- In-built views; request-and-response objects; template response objects.

- The uploading of files.
- Views that are based on classes.
- Producing csv files and producing pdf files.
- The term "middleware"

6.2.8Template Layer:

- The template layer offers a syntax that is favourable to designers and can be used to display the information that will be shown to the user. Discover how programmers might enhance this syntax and how designers can utilise it in their work by reading the following.
- The fundamentals.
- A language overview, built-in tags and filters, and humanization options are provided for designers.
- For computer programmers, there is a template application programming interface, custom tags and filters, and a custom template backend.

Forms:

laravel provides a sophisticated system that facilitates its establishment anyway aspects or the deceit sure registration information.

- form api | built-in fields | built-in plug - ins.
- forms such as models | integrating media | form sets | customizing validation

6.2.8 Security:

- Confidentiality is a topic of utmost importance there in the construction of apps; nonetheless, Laravel provides numerous safety equipment instead of frameworks:
- Security overview
- Concerns with django's security that have been reported
- Protection against clickjacking
- Protection against fraud in cross-venue applications
- Cryptographic signing
- Security software for middleware

Internationalization and localization

You may design apps for different languages and areas of the globe with the help of django's powerful internationalisation and localization framework:

Performance and optimization

There are a range of strategies and technologies that may assist make your code operating more effectively - quicker, and using fewer system resources.

Geographic framework

The goal of the geodjango project is to develop a geographic web framework of the highest calibre. Its purpose is to make it as simple as possible to construct gis web applications and to capitalise on the potential of data that is spatially enabled..

6.2.9 Common Web Application Tools:

Django offers multiple tools commonly needed in the development of Web applications:

- Authentication
- Caching
- Logging
- Sending emails
- Syndication feeds (RSS/Atom)

CHAPTER 7

IMPLEMENTATION

7 IMPLEMENTATION

7.2 MODULES AND DESCRIPTION

7.2.6 Service Provider

In this module, the service provider has to login by using valid user name and password. after login successful he can do some operations such as login, train & test user profile data sets, view user profile trained and tested accuracy in bar chart, view user profile trained and tested accuracy results, view all profile identity prediction, find and view profile identity prediction ratio, view user profile identity ratio results, download predicted data sets, view all remote users

View And Authorize Users

In this module, the admin can view the list of users who all registered. in this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

7.2.7 Remote User

In this module, there are n numbers of users are present. user should register before doing any operations. once user registers, their details will be stored to the database. after registration successful, he has to login by using authorized user name and password. once login is successful user will do some operations like register and login, predict profile identification status, view your profile.

CHAPTER 8

SAMPLE CODE

```
class UserTestForm(forms.ModelForm):
    Age = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    Gender = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    Country = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    state = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    family_history = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    work_interfere = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    no_employees = forms.CharField(widget=forms.TextInput, required=True, max_length=100)

    remote_work = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    tech_company = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    benefits = forms.CharField(widget=forms.TextInput, required=True, max_length=100)
    CharField(widget=forms.TextInput, required=True, max_length=100)

class Meta():
    model = UserTestModel
    fields = '__all__'

care_options = forms
```


#css code

```
/* appling google font over the website */
body{
    font-family: 'Josefin Sans', sans-serif;
    font-weight: 400;
}
/* styling menu icon */
.nav-menu{
    border: none;
    background-color: #fff;
    font-size: 1.5rem;
    color: #fca61f;
}
/* styling navbar links */
.navbar-nav a{
    color: black;
    position: relative;
}
.navbar-nav a:hover{
    color: #fca61f;
}
.navbar-nav a:after {
    background: none repeat scroll 0 0 transparent;
    bottom: 0;
    content: "";
    display: block;
    height: 2px;
    left: 50%;
    top: 35px;
    position: absolute;
    background: #fca61f;
    transition: width 0.3s ease 0s, left 0.3s ease 0s;
```

```
width: 0;
}
.navbar-nav a:hover:after {
width: 100%;
left: 0;
}
/* styling contact btn */
.c-btn{
border: none;
background: #fca61f;
color: #fff;
font-size: 1.2rem;
}
/* styling home section starts */
.home{
background: linear-gradient(33deg, rgba(242,242,255,1) 0%, rgba(235,249,255,1) 100%);
padding: 30px 0px 10px 0px;
}
.home h2{
font-size: 3rem;
line-height: 4rem;
}
.c-orange{
color: #6f34fe;
font-weight: 700;
font-size: 4rem;
}
.home p{
width: fit-content;
margin-top: 1rem;
font-size: 1.2rem;
}
.h-btn:hover{
```

```
background: #6f34fe;
color: #fff;
transform: translateY(-20px);
transition-duration: 1s;
}
.link{
  text-decoration: none;
  font-size: 1.2rem;
  color: #fca61f;
  position: relative;
}
.link::after{
  content: "";
  position: absolute;
  width: 100%;
  transform: scaleX(0);
  height: 2px;
  bottom: 0;
  left: 0;
  top: 20px;
  background-color: #6f34fe;
  transform-origin: bottom right;
  transition: transform 0.25s ease-out;
}
.link:hover{
  color: #6f34fe;
}
.link:hover::after{
  transform: scaleX(1);
  transform-origin: bottom left;
}
/* adding animation to home section image */
.home img{
```

```
animation: floating 4s ease-in-out infinite;
}
```

```
@keyframes floating {
  0% {
    transform: translateX(0);
  }
  50% {
    transform: translateX(-24px);
  }
  100% {
    transform: translateX(0);
  }
}
```

```
/* expertise section starts */
```

```
.heading small{
  color: #6f34fe;
  font-size: 1.5rem;
  font-weight: 600;
}
.heading h3{
  margin-top: 0.5rem;
  color: #3f396d;
  font-size: 2.5rem;
  font-weight: 700;
}
.expertise h4{
  color: #3f396d;
  font-weight: 700;
}
.expertise a{
```

```
  text-decoration: none;
```

```
color: #fca61f;
}
.expertise a:hover{
color: #6f34fe;
}
.expertise .service-card{
text-align: center;
padding: 15px 15px;
border-radius: 0.7rem;
}

/* skill section starts */
.skill{
background-color: rgb(234, 234, 249);
}
.progress-card{
padding: 12% 10% 10% 15%;
background-color: #fff;
text-align: center;
border-radius: 10%;
}
.progress-card:hover{
transform: scale(1.1,1.1);
transition-duration: 0.6s;
}
.circular-progress{
position: relative;
height: 130px;
width: 130px;
border-radius: 50%;
background: conic-gradient(#7d2ae8 3.6deg, #ededed 0deg);
display: flex;
align-items: center;
```

```
justify-content: center;
}
.circular-progress::before{
  content: "";
  position: absolute;
  height: 120px;
  width: 120px;
  border-radius: 50%;
  background-color: #fff;
}
.progress-value{
  position: relative;
  font-size: 1.5rem;
  font-weight: 600;
}
.html-progress{
  color: #fca61f;
}
.javascript-progress{
  color: #6f34fe;
}
.php-progress{
  color: #20c997;
}
.reactjs-progress{
  color: #3f396d;
}
.text{
  font-size: 1.2rem;
  font-weight: 500;
}
.heading p{
  font-size: 1.3rem;
```

```
}

/* portfolio section starts */
#myBtnContainer{
    margin-bottom: 45px;
}
#myBtnContainer button{
    border: none;
    color: black;
    background: transparent;
    font-size: 20px;
    font-weight: 600;
    cursor: pointer;
    display: inline-block;
    margin: 0 25px;
}
#myBtnContainer button:hover{
    color: #fca61f;
}

#myBtnContainer button:focus{
    outline: none;
}
.post .content h4{
    font-size: 1rem;
    font-weight: 700;
}
.post .card{
    border: none;
}

/* testimonial section starts */
.testimonial img:hover{
```

```
transform: scale(1.2,1.2);
transition-duration: 1s;
}
.testimonial h4{
    color: #3f396d;
}
.testimonial i{
    font-size: 2.9rem;
    color: #fca61f;
}
.testimonial .nextprev-btn{
    border: none;
    background-color: #fff;
}
.prev:hover,.prev-btn:hover{
    color:#6f34fe;
    transform: translateX(-10px);
    transition-duration: 1s;
}
.next:hover,.next-btn:hover{
    color:#6f34fe;
    transform: translateX(10px);
    transition-duration: 1s;
}

/* blog section starts */
.blog{
    background-color: #f2f2ff;
}
.blog .blogpost .card{
    border: none;
    border-radius: 5%;
```



```
}  
.nav-bg{  
    background-color: #fff;  
}  
  
.blog .blogpost small{  
    color: #6f34fe;  
    font-size: 1.1rem;  
}  
.blog .blogpost h4,.blogpost a{  
    color: #3f396d;  
    font-weight: 700;  
    margin-top: 1rem;  
    text-decoration: none;  
}  
.blogpost a:hover{  
    color: #fca61f;  
}  
.blog .blogpost .read-more-btn{  
    text-decoration: none;  
    color: #fca61f;  
    font-size: 1.1rem;  
}  
.blog .blogpost .read-more-btn:hover{  
    color: #6f34fe;  
}  
/* styling modal group */  
.btn-c{  
    margin-left: 94%;  
    background-color: transparent;  
    border: none;  
    color: #fff;  
    font-size: 2rem;
```

```
margin-top: -3%;
}
.btn-c i{
    color: #fca61f;

#user registration model

from django.db import models

# Create your models here.
class UserRegistrationModel(models.Model):
    name = models.CharField(max_length=100)
    loginid = models.CharField(unique=True, max_length=100)
    password = models.CharField(max_length=100)
    mobile = models.CharField(unique=True, max_length=100)
    email = models.CharField(unique=True, max_length=100)
    locality = models.CharField(max_length=100)
    address = models.CharField(max_length=1000)
    city = models.CharField(max_length=100)
    state = models.CharField(max_length=100)
    status = models.CharField(max_length=100)

    def __str__(self):
        return self.loginid

    class Meta:
        db_table = 'UserRegistrations'

# Create your models here.
class UserTestModel(models.Model):
    Age = models.CharField(max_length=100)
    Gender = models.CharField(max_length=100)
    Country = models.CharField(max_length=100)
    state = models.CharField( max_length=100)
    family_history = models.CharField( max_length=100)
```

```
work_interfere = models.CharField(max_length=100)
no_employees = models.CharField(max_length=100)
remote_work = models.CharField(max_length=100)
tech_company = models.CharField(max_length=100)
benefits = models.CharField(max_length=100)
care_options = models.CharField(max_length=100)
loginid = models.CharField(unique=True, max_length=100)

def __str__(self):
    return self.loginid

class Meta:
    db_table = 'UserTest'

class MHVideosModel(models.Model):
    name = models.CharField(max_length=255)
    gender = models.CharField(max_length=255)
    videopath = models.FileField(upload_to='videos/')
    convertvideopath = models.FileField(upload_to='videos/')
    language = models.CharField(max_length=255)

    def __str__(self):
        return self.name

class Meta:
    db_table = 'MHVideos'
```

```
import numpy
import os
import csv
import sqlite3
import pandas as pd
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

def send_email(sender_email, sender_password, recipient_email, subject, body):
    # Create a multipart message and set headers
    message = MIMEMultipart()
    message['From'] = sender_email
    message['To'] = recipient_email
    message['Subject'] = subject

    # Add body to email
    message.attach(MIMEText(body, 'html'))

    # Log in to SMTP server and send email
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
        server.login(sender_email, sender_password)
        server.sendmail(sender_email, recipient_email, message.as_string())

# sender_email = 'sudhatestmail@gmail.com'
# sender_password = 'nhbjssyxvxbendzz'
# recipient_email = 'sudhareddykota23@gmail.com'

# subject = "AI-Enabled Dubbing Software from OTP ",Mainot
```

#views of the document

```
from django.shortcuts import render, HttpResponseRedirect
from .forms import UserRegistrationForm,UserTestForm
from .models import UserRegistrationModel,UserTestModel
from django.contrib import messages
from django.core.files.storage import FileSystemStorage
from subprocess import Popen, PIPE
import subprocess
import moviepy.editor as mp
import speech_recognition as sr
from gtts import gTTS
from googletrans import Translator
import os
import math
import moviepy.editor as mp
import speech_recognition as sr
from gtts import gTTS
from googletrans import Translator
import os
import math
import pyttsx3
import os
import uuid
from moviepy.editor import VideoFileClip, AudioFileClip
import speech_recognition as sr
from .models import MHVideosModel
import asyncio
import edge_tts
import random

import numpy
import os
```

```
import csv
import sqlite3
import pandas as pd
import smtplib

from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

def send_email(sender_email, sender_password, recipient_email, subject, body):
    message = MIMEMultipart()
    message['From'] = sender_email
    message['To'] = recipient_email
    message['Subject'] = subject
    message.attach(MIMEText(body, 'html'))
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
        server.login(sender_email, sender_password)
        server.sendmail(sender_email, recipient_email, message.as_string())

def UserRegisterActions(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            # print('Data is Valid')
            # form.save()
            request.session['user_data'] = form.cleaned_data
            OTP = random.randint(40000, 90000)
            request.session['otp'] = OTP
            print(OTP)
            print("maiotp: ",OTP,".")
```

```
sender_email = 'sudhatestmail@gmail.com'
sender_password = 'nhbjssyxvxbendzz'
recipient_email = form.cleaned_data['email']
subject = f"Your OTP for AI-Enabled Dubbing Software is {OTP}"
body = f"""
<html>
<body>
    <p>Your One-Time Password (OTP) for registration is <b>{OTP}</b>.</p>
    <p>Auto-generated by the AI-Enabled Dubbing Software.</p>
</body>
</html>
"""
send_email(sender_email, sender_password, recipient_email, subject, body)

return render(request, 'UserOTP.html')
else:
    messages.success(request, 'Email or Mobile Already Existed')
    print("Invalid form")
else:
    form = UserRegistrationForm()
return render(request, 'UserRegistrations.html', {'form': form})
```

```
def UserOTPCheck(request):
    if request.method == "POST":
        logotp = request.POST.get('otp')
        userotp = request.session['otp']
        print("logotp: ",logotp, ".")
        print("useotp: ",userotp, ".")

        print("logotp (raw):", logotp)
        print("userotp (raw):", userotp)
```

```
if userotp is None:
    messages.error(request, 'OTP session expired. Please register again.')
    form = UserRegistrationForm()
    return render(request, 'UserRegistrations.html', {'form': form})

try:
    logotp = int(logotp) # Convert logotp to integer
    userotp = int(userotp)
except ValueError:
    messages.error(request, 'Invalid OTP format. Please enter numbers only.')
    return render(request, 'UserOTP.html')

print("logotp (int):", logotp)
print("userotp (int):", userotp)

if logotp == userotp:
    user_data = request.session.get('user_data')
    if user_data:
        UserRegistrationModel.objects.create(**user_data)
        del request.session['user_data']
        del request.session['otp']
        messages.success(request, 'You have been successfully registered')
        form = UserRegistrationForm()
        return render(request, 'UserRegistrations.html', {'form': form})
    else:
        messages.success(request, 'Invalid OTP')
        form = UserRegistrationForm()
        return render(request, 'UserRegistrations.html', {'form': form})
```

```
def UserLoginCheck(request):
```

```
    if request.method == "POST":
```

```
        loginid = request.POST.get('loginname')
```



```
pswd = request.POST.get('pswd')
print("Login ID = ", loginid, ' Password = ', pswd)
try:
    check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)
    status = check.status
    print('Status is = ', status)
    if status == "activated":
        request.session['id'] = check.id
        request.session['loggeduser'] = check.name
        request.session['loginid'] = loginid
        request.session['email'] = check.email
        print("User id At", check.id, status)
        return render(request, 'users/UserHome.html', {})
    else:
        messages.success(request, 'Your Account Not at activated')
        return render(request, 'UserLogin.html')
except Exception as e:
    print('Exception is ', str(e))
    pass
    messages.success(request, 'Invalid Login id and password')
return render(request, 'UserLogin.html', {})

def UserHome(request):
    return render(request, 'users/UserHome.html', {})

def Generatedubbingvio(request):
    return render(request, 'users/Gendubbingvio.html', {})

def Generateddubbingviodssss(request):
    username = request.session['loginid']
    print("username ::",username)
    data = MHVideosModel.objects.get(name=username)
```

```
print("data ::",data)
return render(request, 'users/ViewVideos.html',{'data':data})

def Generateddubbingvios(request):
    username = request.session.get('loginid', None)
    if not username:
        return HttpResponseRedirect("User not logged in", status=403)
    print("username ::", username)
    data = MHVideosModel.objects.filter(name=username)
    print("data ::", data)
    return render(request, 'users/ViewVideos.html', {'data': data})

def extract_audio_from_video(video_path, audio_output_path):
    try:
        video = VideoFileClip(video_path)
        video.audio.write_audiofile(audio_output_path)
    except Exception as e:
        print(f"Error extracting audio: {e}")

def audio_to_text(audio_path, chunk_duration=60):
    recognizer = sr.Recognizer()
    try:
        with sr.AudioFile(audio_path) as source:
            full_audio = recognizer.record(source)
            total_duration = source.DURATION

        transcription = ""
        num_chunks = math.ceil(total_duration / chunk_duration)

        for i in range(num_chunks):
            with sr.AudioFile(audio_path) as source:
                audio = recognizer.record(source, duration=chunk_duration, offset=i * chunk_duration)
            try:
```

```
        chunk_text = recognizer.recognize_google(audio)
        transcription += chunk_text + " "
    except sr.UnknownValueError:
        continue
    except sr.RequestError as e:
        print(f"Google Speech Recognition error: {e}")
        break

    return transcription
except Exception as e:
    print(f"Error in audio transcription: {e}")
    return ""

def translate_text(text, target_lang):
    try:
        translator = Translator()
        return translator.translate(text, dest=target_lang).text
    except Exception as e:
        print(f"Error translating text: {e}")
        return ""

def text_to_speech_pytsx3(text, lang, gender, output_audio_path):
    engine = pytsx3.init()
    voices = engine.getProperty('voices')
    # selected_voice = [v for v in voices if gender.lower() in v.name.lower()]

    # engine.setProperty('voice', selected_voice[0].id if selected_voice else voices[0].id)
    if gender == "male":
        engine.setProperty('voice', voices[0].id)
    elif gender == "female":
        engine.setProperty('voice', voices[1].id)
    engine.setProperty('rate', 125)
```

```
engine.runAndWait()
print(f"Generated speech saved to {output_audio_path}")
```

```
def text_to_speech(text, lang, gender, output_path):
    try:
        tts = gTTS(text=text, lang=lang)
        tts.save(output_path)
    except Exception as e:
        print(f"Error generating speech: {e}")
```

```
def text_to_speech_asyncio(text, lang, gender, output_audio_path):
    VOICES = ['en-AU-NatashaNeural', 'en-AU-WilliamNeural', 'en-CA-ClaraNeural', 'en-CA-
    LiamNeural', 'en-GB-LibbyNeural']
```

```
if gender == "male":
    VOICE = VOICES[3]
    async def amain() -> None:
        communicate = edge_tts.Communicate(text, VOICE)
        await communicate.save(output_audio_path)
    loop = asyncio.get_event_loop_policy().get_event_loop()

    try:
        loop.run_until_complete(amain())
    finally:
        loop.close()
```

```
elif gender == "female":
    VOICE = VOICES[4]
```

```
    communicate = edge_tts.Communicate(text, VOICE)
    await communicate.save(output_audio_path)

loop = asyncio.get_event_loop_policy().get_event_loop()

try:
    loop.run_until_complete(again())
finally:
    loop.close()

def merge_audio_with_video(video_path, audio_path, output_path):
    try:
        video = VideoFileClip(video_path)
        audio = AudioFileClip(audio_path)
        video = video.set_audio(audio)
        video.write_videofile(output_path, audio_codec="aac")
    except Exception as e:
        print(f"Error merging audio with video: {e}")

def TestGeneratedubbingvio(request):
    if request.method == 'POST':
        language = request.POST.get('language')
        gender = "female"
        video_file = request.FILES.get('videopath')

        if not video_file or not language or not gender:
            messages.error(request, "Please provide all required inputs!")
            return render(request, 'Gendubbingvio.html')

    try:
```

```
fs = FileSystemStorage(location='media/')
saved_video_path = fs.save(video_file.name, video_file)
video_path = os.path.join('media/', saved_video_path)
unames = request.session['loginid']
random_number = random.randint(40000, 90000)

audio_file = "extracted_audio.wav"
dubbed_audio_file = f"dubbed_audio_{language}_{unames}_{random_number}.mp3"
final_video_file = f"final_video_{language}_{unames}_{random_number}.mp4"

audio_path = os.path.join('media/', audio_file)
dubbed_audio_path = os.path.join('media/', dubbed_audio_file)
final_video_path = os.path.join('media/', final_video_file)

extract_audio_from_video(video_path, audio_path)
transcription = audio_to_text(audio_path)

if transcription:
    translated_text = translate_text(transcription, language)
    text_to_speech(translated_text, language, gender, dubbed_audio_path)
    # text_to_speech_pytsx3(translated_text, language, gender, dubbed_audio_path)

merge_audio_with_video(video_path, dubbed_audio_path, final_video_path)
username = request.session['loginid']

# Save to DB
video_entry = MHVideosModel(
    name=username,
    gender=gender,
    videopath=video_path,
    convertvideopath=final_video_path, 1
```

CHAPTER 9

SCREENSHOTS

8 SCREENSHORTS

9.1. Home page:

The screenshot shows the home page of the application. At the top, there is a navigation bar with the logo "AI-Enabled Dubbing for Multilingual Content" on the left and three links: "Home", "Generate Dubbing", and "Generated Dubbing videos" on the right. Below the navigation bar, there is a large heading "Generate Dubbing Video Form". Underneath the heading, there is a form with a success message "Dubbing process completed successfully!". The form has two input fields: "video file" with a "Choose File" button and "No file chosen" text, and "language" with a dropdown menu showing "select language". A "Submit" button is at the bottom of the form.

9.2. Remote User Registration:

The screenshot shows the "User Register Form" page. At the top, there is a navigation bar with the logo "AI-Enabled Dubbing for Multilingual Content" on the left and three links: "Home", "Users", and "Admin" on the right. Below the navigation bar, there is a large heading "User Register Form". Underneath the heading, there is a form with several input fields: "User Name" (Yasmin), "Login ID" (Yasmin), "Password" (masked with asterisks), "Mobile" (9876543212), "email" (yasminshaik.tech@gmail), "Locality" (nan), and "Address" (nan). The form is titled "User Register Form".

9.3. Remote User OTP Generation:

User OTP Form

9.4. Admin Login Form:

Admin Login Form

Yasmin

Admin

Please Check Your Login Details

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

9.5 Multilingual Content Identification Status:

AI-Enabled Dubbing for Multilingual Content

[Home](#)[Users](#)[logout](#)

Registered users and Activate them

S.No	Name	Login ID	Mobile	Email	Locality	Status	Activate
1	Nokia	Nokia	9123456789	nokia@gmail.com	Madhapur	activated	Activated
2	Sudharshan	Sudharshan	8671974718	sudharshan.kota@gmail.com	Madhapur	activated	Activated
3	Sudha	Sudha	7411974718	sudha.kota23@gmail.com	Madhapur	activated	Activated

9.6 User Login: Form

AI-Enabled Dubbing for Multilingual Content

[Home](#)[Users](#)[Admin](#)[Registrations](#)

User Login Form

Login

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

9.6 Result Of Uploaded Videos:

Uploaded Videos

Results table

S.No	User Name	Gender	Uploed Video	convert Video	language
------	-----------	--------	--------------	---------------	----------

9.7 Generated Dubbing Video Form:

Generate Dubbing Video Form

Dubbing process completed successfully!

video file

Choose File

No file chosen

language



select language

Submit

9.8 Result of Genarated Dubbing Video:

Uploaded Videos

Results table

S.No	User Name	Gender	Uploed Video	convert Video	language
1	Yasmin	female			hi

**TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING
APPLICATION FOR MULTILANGUAGES**

**CHAPTER 10
SYSTEM TESTING**

9 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 test. Each test type addresses a specific testing requirement.

9.2 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 satisfactory, 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.

TRANSFORMATIVE PERSPECTIVES: AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

Functional test

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.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing 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.

9.2.6 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

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

Features to be tested

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

9.2.7 Integration Testing

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.

9.2.8 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.

9.3 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

9.3.6 Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing paths are tested for the expected results. All error handling paths are also tested.

9.3.7 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of highorder tests are conducted. The main objective in this testing process is to take unit tested

modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

The low-level modules are combined into clusters into clusters that perform a specific Software sub-function. A driver (i.e.) the control program for testing is written to coordinate test case input and output. The cluster is tested. Drivers are removed and clusters are combined moving upward in the program structure. The bottom up approaches tests each module individually and then each module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

9.3.8 User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

9.3.9 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

9.3.10 Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces an output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files

and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules.

aUsing the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules.

This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

TRANSFORMATIVE PERSPECTIVES:AI ENABLED DUBBING APPLICATION FOR MULTILANGUAGES

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

CHAPTER 11

FUTURE ENHACEMENT

10 FUTURE ENHANCEMENT

Develop AI-powered systems capable of real-time dubbing in multiple languages, preserving the speaker's original tone, emotion, and lip synchronization for a seamless multilingual experience.

In the future, AI machines can be enhanced to perform real-time dubbing across multiple languages. This means AI would not only translate spoken content but also recreate the speaker's voice, tone, and emotions accurately in any target language, making communication truly global and seamless. With advancements in artificial intelligence, a major future enhancement could be AI-driven multilingual dubbing. Instead of simply translating text, AI systems would analyze the speaker's voice, emotional tone, and speaking style, and then generate speech in multiple languages while preserving these unique characteristics. This would allow movies, educational videos, meetings, and live broadcasts to be dubbed instantly and naturally into various languages, eliminating language barriers. By combining technologies like deep learning, speech synthesis, natural language processing, and real-time voice cloning, AI could create a highly immersive and authentic multilingual communication experience, reshaping global media and cross-cultural interactions.

CHAPTER 12

CONCLUSION

11 CONCLUSION

The project "**AI Machine Enabled Dubbing in Multilingual**" successfully demonstrates the potential of artificial intelligence in transforming the dubbing process across multiple languages. By integrating advanced speech synthesis, voice cloning, and language translation technologies, the system enables seamless and realistic dubbing of audio-visual content while preserving the speaker's tone and emotion. This approach significantly reduces the time, cost, and human effort traditionally required in multilingual dubbing, making high-quality content localization more accessible. The project also highlights the ability of AI to adapt to various languages and dialects, ensuring cultural relevance and engagement for diverse audiences.

In conclusion, AI-powered dubbing not only enhances global communication but also opens up new opportunities for content creators, educators, and media companies to reach broader audiences in their native languages.

CHAPTER 13

REFERENCES

12 REFERENCES

[1] **VALL-E: Neural Codec Language Models for Contextual Speech Synthesis**

Microsoft Research, 2023

<https://arxiv.org/abs/2301.02111>

- Introduces a model that can synthesize speech with emotional and speaker consistency from just a few seconds of audio.

[2] **Translatotron 2: Robust Direct Speech-to-Speech Translation**

Google Research, 2022

<https://arxiv.org/abs/2210.03012>

- Discusses end-to-end speech-to-speech translation, a core technology for AI dubbing.

[3] **Meta Voicebox: Text-to-Speech and Speech-to-Speech AI Model**

Meta AI, 2023

<https://ai.facebook.com/blog/voicebox-meta-generative-ai-model-speech/>

- Details an advanced speech model capable of multilingual voice synthesis.

[4] **OpenAI Whisper: Multilingual Speech Recognition System**

OpenAI, 2022

<https://openai.com/research/whisper>

- A powerful ASR (Automatic Speech Recognition) model that supports multiple languages.

[5] **Descript's Overdub Tool**

<https://www.descript.com/overdub>

- A commercial tool that uses AI voice cloning to enable easy dubbing and voice editing.

[6] **Deepdub: AI Dubbing for Film and TV**

<https://www.deepdub.ai/>

- A startup that provides AI-powered multilingual dubbing for entertainment content.

[7] **Respeecher: Voice Cloning for Dubbing and Localization**

<https://www.respeecher.com/>

Research Papers and Articles

- Uses AI to create high-quality voice clones for dubbing and voiceovers in different languages.
- "Automated Dubbing: Synthesizing Speech Aligned to Lips"

Authors: Prajwal K. R., Rudrabha Mukhopadhyay, Abhishek Jha, C.V. Jawahar

<https://arxiv.org/abs/1906.04160>

Discusses a pipeline that includes lip-syncing and voice synthesis for automated multilingual dubbing.

- **"Multilingual and Zero-shot Speech Synthesis Using a Unified Acoustic Model"**

Google Research

<https://arxiv.org/abs/2005.00901>

Proposes a multilingual TTS system that can synthesize speech in multiple languages with a single model.

- **"Text-to-Speech Synthesis with Transformer Network"**

Authors: Naihan Li, Shujie Liu, Yanqing Liu, et al.

<https://arxiv.org/abs/1809.08895>

Describes an end-to-end speech synthesis system using transformers, important for voice cloning