VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JNANA SANGAMA", BELAGAVI - 590 018

PROJECT PHASE - I REPORT

on

"VOICE-BASED BANKING FOR RURAL AREAS"

Submitted by

Sushrutha Shanbhogue	4SF22CS225
Raghavendra SS	4SF22CS158
Sowndarya S	4SF22CS218
Iram A.K Shaikh	4SF23CS404

In partial fulfillment of the requirements for the VI semester

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Mr. Raghavendra Sooda

Assistant Professor, Department of CSE

at

SAHYADRI

College of Engineering & Management
An Autonomous Institution
MANGALURU
2024 - 25

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the phase - I work of project entitled **“Voice-Based Banking for Rural Areas”** has been carried out by **Sushrutha Shanbhogue (4SF22CS225), Raghavendra SS (4SF22CS158), Sowndarya S (4SF22CS218) and Iram A.K Shaikh (4SF23CS404)**, the bonafide students of Sahyadri College of Engineering and Management in partial fulfillment of the requirements for the VI semester of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024 - 25. It is certified that all suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Project Guide	**Project Coordinator**	**HOD**
Mr. Raghavendra Sooda	**Mr. Suhas A Bhyratae**	**Dr. Mustafa Basthikodi**
Assistant Professor	Associate Professor	Professor & Head
Dept. of CSE	Dept. of CSE	Dept. of CSE

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Phase - I Report titled **"Voice-Based Banking for Rural Areas"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mr. Raghavendra Sooda,** in partial fulfillment of the requirements for the VI semester of **Bachelor of Engineering** in **Computer Science and Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Sushrutha Shanbhogue	**(4SF22CS225)**
Raghavendra SS	**(4SF22CS158)**
Sowndarya S	**(4SF22CS218)**
Iram A.K Shaikh	**(4SF23CS404)**

Dept. of CSE, SCEM, Mangaluru

Abstract

Accessing and utilizing banking services remains a significant challenge for rural populations, primarily due to language barriers, low digital literacy, and limited exposure to formal financial systems. This project, titled **“Voice-Based Banking for Rural Areas”**, addresses these issues by developing an intelligent, voice-enabled chatbot that allows users to interact with banking systems using regional languages like Kannada and even unwritten languages such as Tulu and Konkani.

The system starts by taking the user’s details, such as name, address, etc., as voice input in a regional language like Kannada. This voice input is recognized and processed using the **OpenAI Whisper model**, a multilingual speech recognition model trained on large-scale weakly labeled audio data. Whisper efficiently transcribes and translates the Kannada (or any regional or unwritten language) speech to English text.

If the request involves filling out a banking form for account creation or availing a loan, the chatbot extracts relevant information and automatically generates a filled-out text document. This ensures that rural people feel encouraged to visit banks and follow banking procedures, making them more financially secure and educated.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Phase - I Report on **"Voice-Based Banking for Rural Areas"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi in partial fulfillment of the requirements for the VI semester of Bachelor of Engineering in Computer Science and Engineering.

We are profoundly indebted to our guide, **Mr. Raghavendra Sooda**, Assistant Professor, Department of Computer Science and Engineering for innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We also thank **Dr. Suhas A Bhyratae** and **Ms. Prapulla G**, Project Coordinators, Department of Computer Science and Engineering for their constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Mustafa Basthikodi**, Professor and Head, Department of Computer Science and Engineering for his invaluable support and guidance.

We sincerely thank **Dr. S. S. Injaganeri**, Principal, Sahyadri College of Engineering and Management, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

Sushrutha Shanbhogue (4SF22CS225)
Raghavendra SS (4SF22CS158)
Sowndarya S (4SF22CS218)
Iram A.K Shaikh (4SF23CS404)

Table of Contents

List of Figures

Chapter 1

Introduction

In recent years, India has witnessed an extraordinary growth in the financial services sector, marked by the emergence of digital banking infrastructure and widespread usage of mobile and internet-based financial tools. Initiatives such as Digital India, Jan Dhan Yojana, and the Unified Payments Interface (UPI) have played a pivotal role in enabling access to formal banking services for millions of citizens. These initiatives have significantly contributed to the country's vision of financial inclusion, fostering economic development and reducing disparities between urban and rural populations.

However, despite these commendable efforts, a significant portion of India's population—particularly those residing in rural and semi-rural areas—continues to face substantial challenges in accessing and effectively utilizing digital banking services. These challenges are often rooted in fundamental socio-economic and infrastructural issues. Among the most pressing barriers are low literacy rates, limited digital literacy, poor internet connectivity, and a strong reliance on regional languages and oral communication. While urban users may navigate complex banking applications with ease, rural users frequently find these systems unintuitive, inaccessible, and alienating.

Most traditional banking interfaces are designed with literate, tech-savvy users in mind, often using English or Hindi as the default language. This leaves speakers of regional languages—such as Kannada, Tulu, or Konkani—at a disadvantage, especially when those languages have limited digital representation or lack standard written forms. As a result, many individuals are excluded from even the most basic banking functions, such as checking account balances, initiating transfers, or applying for loans, simply because they are unable to interact with the system in a language or format they understand.

To address these critical gaps, this project proposes the development of an innovative, voice-enabled banking assistant that leverages cutting-edge technologies in artificial intelligence, speech recognition, and natural language processing (NLP). The solution is designed specifically to cater to the needs of rural and linguistically diverse populations, providing an intuitive and inclusive interface for conducting essential banking operations.

At the heart of this system is OpenAI's Whisper model—a state-of-the-art speech recognition framework capable of transcribing and translating spoken language with high accuracy. The model is trained on a wide range of audio data, enabling it to process diverse accents and dialects. In the proposed solution, user queries spoken in a regional language like Kannada are first transcribed and translated into English. This translated input is then interpreted by an intelligent chatbot system, which identifies the user's intent and executes the corresponding banking function.

Furthermore, the system extends beyond mere voice interaction. It is capable of automatically generating banking documents—such as account creation and loan application form—pre-filled with the relevant user information derived from the interaction. This capability is particularly valuable in rural settings, where access to printers, bank forms, or assistance is limited. Users can receive these documents in digital format for submission or printing, further simplifying the banking process.

By combining advanced technologies in speech-to-text conversion, machine translation, intent recognition, and document automation, the proposed system aims to bridge the digital divide that separates rural populations from mainstream financial services. It seeks to eliminate barriers imposed by language, literacy, and technical complexity, thereby empowering individuals to perform banking tasks independently and confidently.

The solution also presents a scalable model that can be extended to support additional regional languages and dialects as needed, with minimal changes to its core architecture. This adaptability ensures that the system can evolve alongside the linguistic and cultural diversity of its users. It can also be integrated into other sectors such as government services, healthcare, or education, thereby opening doors to a wider range of voice-based applications in public service delivery.

Moreover, by promoting self-reliance in financial matters, the project aligns with broader socio-economic goals such as reducing the urban-rural development gap and increasing digital financial literacy. As India continues to digitize its economy, tools like voice-enabled banking systems will be critical in ensuring that no community is left behind. The integration of user-centric AI models with local language support marks a paradigm shift in how rural citizens can participate meaningfully in the digital economy.

In addition to usability, a critical concern in the adoption of digital banking systems—particularly in rural areas—is trust. Many individuals in these communities are wary of technology due to concerns about privacy, fraud, and data misuse. Unlike urban populations, who are more accustomed to digital verification and encryption protocols, rural users often lack awareness of the safety mechanisms that underpin online banking. This project proactively addresses these concerns by incorporating robust data privacy and security features. Sensitive user data such as personal details and transaction history are encrypted during transmission and storage, ensuring that the voice-based interaction remains not only accessible but also secure. This promotes user confidence and encourages more people to adopt digital services.

Thus, this system has the potential to redefine how underserved populations engage with technology. By prioritizing natural language interaction and minimizing dependence on textual interfaces, the proposed solution challenges conventional notions of digital literacy. It empowers users who have historically been excluded from formal financial systems—not because of lack of interest or capacity, but due to a lack of accessible tools. Over time, as users become more comfortable with such voice-based technologies, it is likely to inspire broader engagement with other digital platforms, thereby fostering a digitally inclusive ecosystem. In this way, the project goes beyond a technical implementation—it becomes a catalyst for digital transformation at the grassroots level.

Chapter 2

Literature Survey

Robust Speech Recognition via Large-Scale Weak Supervision (Radford et al., 2022)[1] introduces Whisper, a Transformer-based multilingual speech recognition model trained on 680k hours of weakly labeled audio data. The model achieves strong zero-shot performance, surpassing supervised and commercial ASR systems in robustness and long-form transcription. Its scalability and multilingual capabilities make it suitable for diverse applications, including rural banking interfaces.

UniSpeech-SAT: Universal Speech Representation Learning with Speaker-Aware Pre-Training (Chen et al., 2022)[2] proposes a method to improve speaker recognition and personalization in speech systems. The approach combines speaker-aware pre-training with fine-tuning for ASR and speaker verification, making it useful for authentication in voice-based banking applications.

Maestro: Matched Speech Text Representations Through Modality Matching (Chen et al., 2022)[3] presents a framework that jointly learns speech and text representations to enhance ASR performance. By aligning speech and text modalities, the model supports seamless speech-to-text conversion, which is critical for banking applications requiring accurate transcription.

Unsupervised Speech Recognition (Baevski et al., 2021)[4] eliminates the need for labeled audio data by leveraging self-supervised learning with wav2vec 2.0. This approach is particularly beneficial for low-resource languages and rural settings where transcribed data is scarce, enabling robust voice applications without extensive labeled datasets.

SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large

Neural Network (Chan et al., 2021)[5] demonstrates that aggregating diverse speech datasets improves ASR performance across domains. The method is effective for handling mixed accents and noisy environments, making it suitable for rural voice-based systems with varied linguistic inputs.

VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning (Wang et al., 2021)[6] introduces a 400K-hour multilingual dataset covering 23 languages. The corpus supports self-supervised learning and semi-supervised tasks, providing a valuable resource for training robust voice systems in low-resource and rural language settings.

XLS-R: Self-Supervised Cross-Lingual Speech Representation Learning at Scale (Babu et al., 2021)[7] enables multilingual speech recognition by pre-training on 436 languages. The model improves low-resource language support, making it ideal for diverse rural languages and dialects in banking applications.

Self-Training and Pre-Training Are Complementary for Speech Recognition (Xu et al., 2021)[8] shows that combining self-supervised pre-training with pseudo-labeling significantly enhances ASR performance, especially in low-resource domains. This approach is valuable for improving accuracy in rural voice-based systems with limited labeled data.

Multitask Training with Text Data for End-to-End Speech Recognition (Wang et al., 2020)[9] incorporates text-only data to improve ASR performance. The method enhances generalization in scenarios where speech data is scarce, making it useful for rural applications with limited audio resources.

CHiME-6 Challenge: Tackling Multi-Speaker Speech Recognition for Unsegmented Recordings (Watanabe et al., 2020)[10] addresses challenges in noisy, multi-speaker environments. The techniques developed, such as distant microphone ASR and speech diarization, are critical for deploying voice systems in public rural spaces.

SpecAugment: A Simple Data Augmentation Method for ASR (Park et al., 2019)[11] improves ASR robustness against background noise using time warping and frequency masking. This method is essential for rural deployments where environmental noise can degrade system performance.

Toward Domain-Invariant Speech Recognition via Large Scale Training (Narayanan et al., 2018)[12] achieves domain generalization through large-scale training. The approach helps adapt banking systems to diverse rural usage contexts, ensuring consistent performance across varying acoustic conditions.

Attention Is All You Need (Vaswani et al., 2017)[13] introduces the Transformer model, which revolutionizes sequence transduction tasks with self-attention mechanisms. The architecture's efficiency and scalability underpin modern ASR and translation systems, including those used in this project.

Librispeech: An ASR Corpus Based on Public Domain Audio Books (Panayotov et al., 2015)[14] provides a widely used dataset for training and evaluating speech models. The corpus serves as a benchmark for voice interfaces, ensuring reliable performance in applications like voice-based banking.

Low-Latency Inference with Whisper for Real-Time Applications (Müller, 2023)[15] explores the application of Whisper in real-time scenarios, focusing on reducing latency during inference without compromising transcription quality. By introducing optimizations such as model quantization and selective decoding strategies, the work makes Whisper more suitable for low-resource environments and edge devices. This is highly relevant for rural banking systems, where infrastructure constraints and response time are critical factors. The proposed methods enable faster interactions, ensuring that users receive prompt responses during banking tasks.

Exploring OpenAI Whisper for Multilingual Speech Recognition in Low-Resource Languages (Gade et al., 2023)[16] investigates the effectiveness of Whisper in handling speech recognition tasks in low-resource and regional languages, including several Indian dialects. The authors highlight Whisper's robust zero-shot performance across diverse linguistic inputs and its ability to generalize to languages it wasn't explicitly trained on. This capability is particularly beneficial for rural banking applications that must support users communicating in languages like Tulu or Konkani, which often lack sufficient annotated data.

Benchmarking Whisper and Other ASR Systems on Noisy and Accented Speech (Feng et al., 2023)[17] presents a comparative evaluation of Whisper against other commercial ASR systems under real-world acoustic conditions. The study assesses model performance in environments with heavy background noise, multiple speakers, and regional accents. Whisper consistently outperforms in robustness, making it ideal for rural deployment where banking kiosks or mobile devices may be used in noisy, uncontrolled settings. The findings validate Whisper's suitability for user-centric financial systems in diverse geographies.

Evaluating OpenAI Whisper for Real-Time Transcription in Rural Indian Languages (Gupta and Kumar, 2024)[18] evaluates Whisper's practical utility in transcribing speech in regional Indian languages in real-time. It tests Whisper with native Kannada, Tulu, and Konkani speakers and reports promising results in accuracy, timing, and user satisfaction. The authors emphasize Whisper's ability to support multilingual speech interfaces in resource-limited environments. The work directly supports the current project by establishing Whisper as a viable backend for voice-to-text systems tailored to rural users.

An Empirical Study of Whisper on Multilingual Speech Tasks (Zhang et al., 2023)[19] performs an extensive empirical analysis of Whisper's multilingual capabilities across a broad set of languages and domains. The study confirms Whisper's strong generalization abilities and identifies strengths in transcription quality, especially in longer utterances and low-resource contexts. This research strengthens the choice of Whisper for the proposed banking solution, where natural speech is often lengthy and spoken in non-standard dialects.

On the Transferability of Whisper to Code-Switched and Multilingual Environments (Ghannay et al., 2023)[20] investigates Whisper's performance in code-switched environments—contexts where speakers mix two or more languages, such as Kannada-English or Hindi-Konkani. The study finds that Whisper maintains coherent transcription even when switching occurs mid-sentence, outperforming other models. This is particularly significant for rural users who may use a mix of native and formal language when interacting with banking systems, ensuring the model can still extract intent and generate appropriate outputs.

Chapter 3

Problem Statement

Despite India's rapid strides in digital banking and financial technologies over the past decade, a substantial portion of the rural population continues to face significant obstacles in accessing and utilizing formal banking services. This is mainly due to the language barrier which exists between many rural and urban populations, where all banking procedures are designed for English or Hindi speaking customers. This puts them at a financial disadvantage and leaves them largely unaware of bank rules and procedures.

In states like Karnataka, most rural inhabitants speak and understand only Kannada, a regional language that is often not supported effectively in mainstream banking interfaces, which mainly support English. This presents a major barrier for users who cannot read, write, or comprehend these languages. Additionally, many rural users are unfamiliar with standard banking procedures, and form-filling protocols. As a result, even basic banking operations such as opening an account and submitting an application to avail loan become intimidating and difficult.

While some voice-based systems have been introduced in the past to simplify customer interaction, these solutions are generally limited in scope. Existing voice assistants are often designed to respond to rigid, predefined commands and are rarely capable of understanding regional languages like Kannada or unwritten languages such as Tulu and Konkani. As a result, they fall short of meeting the practical, real-world needs of rural users who require end-to-end assistance in a language and mode of communication they are comfortable with.

In response to this pressing gap, the proposed project aims to develop a comprehensive, voice-based banking assistant designed for users in rural areas. This system will use OpenAI's Whisper model, a speech recognition and translation technology capable of accurately transcribing spoken language (such as Kannada) and translating it into English in real time. Once translated, the input will be interpreted based on the user's intent, such as account creation or availing loan and generates the corresponding document or form accordingly. The user's interaction will be entirely voice-driven, eliminating the need for reading, writing, or digital literacy.

By enabling rural users to speak naturally in their regional language, this solution addresses both linguistic and technical barriers. It encourages rural users to create bank accounts or avail loans without depending on any third parties, and fosters trust in digital financial systems, simplifying banking for rural communities.

3.1 Objectives

- To design a user-friendly interface that accommodates users with low or no digital literacy, minimizing the need for typing, reading, or understanding English
- To develop a voice-based chatbot system that allows rural users to interact with banking services using natural speech in a regional language like Kannada or unwritten languages like Tulu and Konkani.
- To integrate OpenAI's Whisper model for accurate speech recognition and real-time translation of spoken Kannada into English for further processing

Chapter 4

Software Requirements Specification

4.1 Functional Requirements

- **Voice Input Handling:** The system shall accept real-time voice input from the user in Kannada, Tulu, or Konkani through a microphone.
- **Speech Recognition:** The system shall use OpenAI Whisper to transcribe the regional voice input into text. The model must support noisy and accented speech conditions.
- **Language Translation:** Transcribed regional language text shall be translated into English using HuggingFace Transformers (version 4.28 or higher).
- **Intent Detection and NLP:** A chatbot engine shall analyze translated English text to extract user intent, such as opening a bank account, checking balance, or applying for a loan.
- **User Profile Management:** The system shall maintain a structured profile for each user, including personal details such as name, address, and age, enabling profile reuse in future interactions.
- **Form Generation:** Based on user input, the system shall auto-generate banking forms in PDF or DOCX format using `fpdf` or `python-docx`.
- **Banking API Integration:** The system shall integrate with (or simulate) core banking APIs to perform backend operations like account creation and balance queries.
- **Text-to-Speech Output:** The final system response shall be converted into voice output in the user's regional language using Coqui TTS and played back to the user.

- **Web Interface:** The user interface shall be simple and intuitive, primarily icon-driven with minimal textual elements, and compatible with web platforms.

4.2 Non-Functional Requirements

- **Usability:** The interface shall be designed for users with little or no digital literacy, with minimal reliance on reading or typing.
- **Scalability:** The architecture must support easy extension to five or more additional languages with minimal code changes.
- **Robustness:** The system shall maintain recognition and translation accuracy in noisy rural environments with ambient sound levels up to 60 dB.
- **Compatibility:** The web interface must be compatible with the latest versions of Google Chrome and Mozilla Firefox.
- **Maintainability:** The system codebase shall be modular and well-documented to enable future maintenance and feature additions.
- **Resilience:** If the speech recognition or translation modules fail, the system should return an appropriate voice error message and log the failure without crashing.

4.3 Hardware Requirements

- **Client Device:** The user device (PC) must have a microphone, at least a 2 GHz dual-core processor, and minimum 32 GB of RAM.
- **Server Specifications:** The backend server must have at least an 8-core CPU, 64 GB RAM, and an NVIDIA GPU to efficiently run AI models such as Whisper.
- **Audio Interface:** A noise-cancelling microphone is recommended to ensure input clarity in rural and noisy environments.
- **Power Backup:** Client and server locations in rural areas should be equipped with uninterrupted power supply (UPS) to ensure smooth and continued service.

4.4 Software Requirements

- **Backend Framework:** Python 3.8+, Flask or Django for RESTful API development.
- **Frontend Technologies:** HTML/CSS/JavaScript for responsive web-based UI.
- **Speech Recognition:** OpenAI Whisper model for multilingual voice-to-text transcription.
- **Text-to-Speech (TTS):** Coqui TTS (v0.11) for converting English text to Kannada or other regional languages for voice output.
- **Document Generation:** FPDF and `python-docx` libraries for auto-generating pre-filled banking documents.

Chapter 5

System Design

5.1 Architecture Diagram

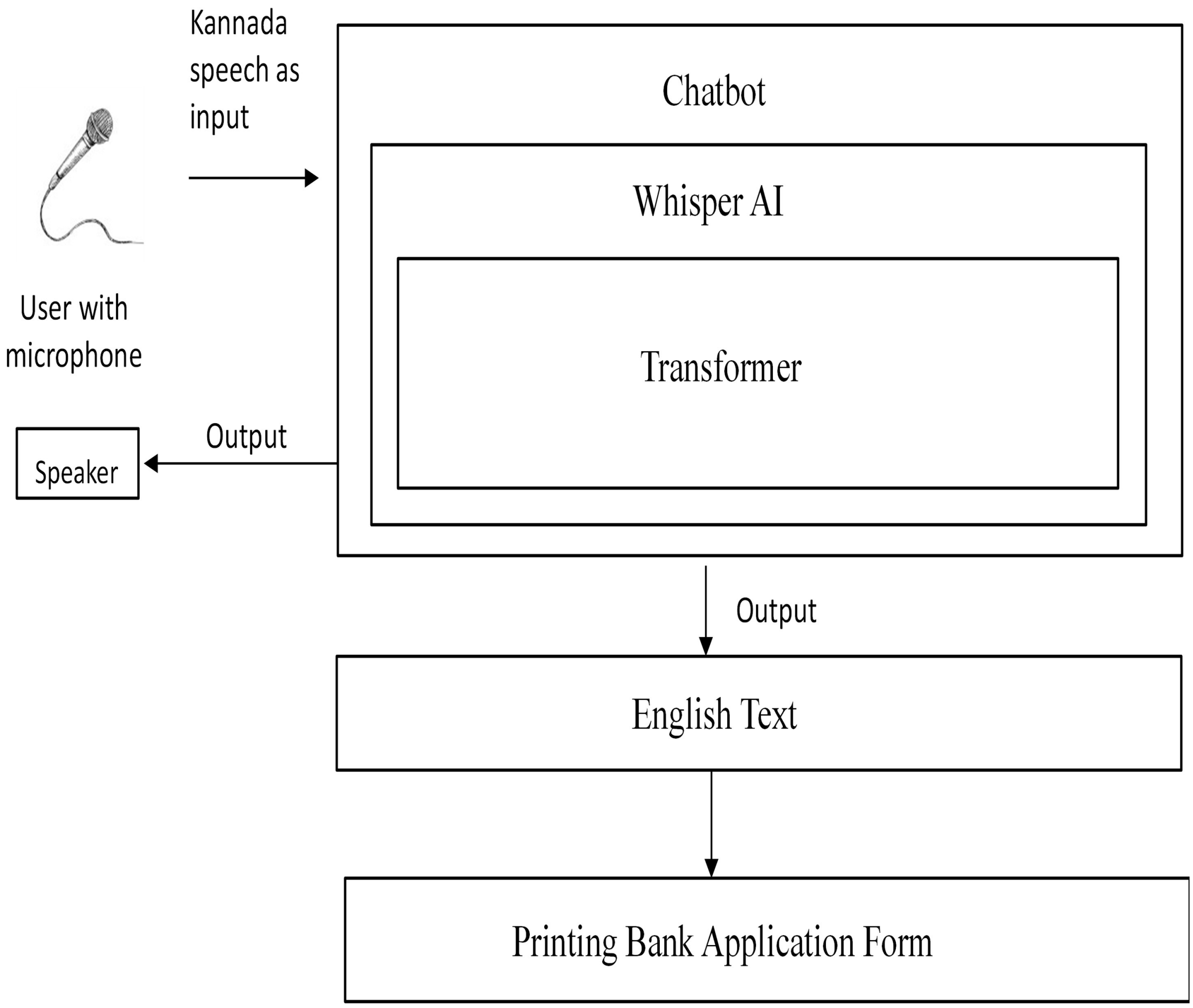


Figure 5.1: High-level system architecture showing components and data flow

The system architecture diagram outlines the major components and how they are interconnected. It illustrates a voice-enabled banking chatbot system specifically designed to cater to regional language users. In this explanation, Kannada has been taken as an example to describe the process flow.

The interaction begins when the user provides voice input in Kannada through a microphone. This microphone captures the spoken input and sends the audio signal to the chatbot system for processing. At the core of this system lies OpenAI's Whisper model, which is a Transformer-based neural network built for tasks such as speech recognition and language translation.

Within the Whisper AI model, the received Kannada speech is first converted into a log-Mel spectrogram. This spectrogram effectively represents the audio signal by capturing important time-frequency information required for understanding spoken language. The Transformer network processes this spectrogram to recognize and transcribe the speech content. After transcription, the recognized Kannada speech is translated into English text, enabling the system to work with a universal language format.

The translated English text is then passed to the chatbot module. This module analyzes the text to interpret the user's intent, allowing the system to determine the appropriate response or action. Based on the detected intent, the chatbot system performs one of two possible actions. It either generates a voice-based response — which may be in Kannada or English depending on the context — and delivers this response to the user through a speaker, or it uses the translated text to automatically populate a bank application form.

Once the form is filled, it is printed and made available to the user. This seamless process completes the interaction, allowing users — especially those from rural or semi-urban regions — to access essential banking services without the need for literacy in English or proficiency with complex digital interfaces. This architecture not only bridges the language barrier but also simplifies the banking experience for those who are less familiar with technology.

5.2 OpenAI Whisper model: Sequence-to-Sequence Learning

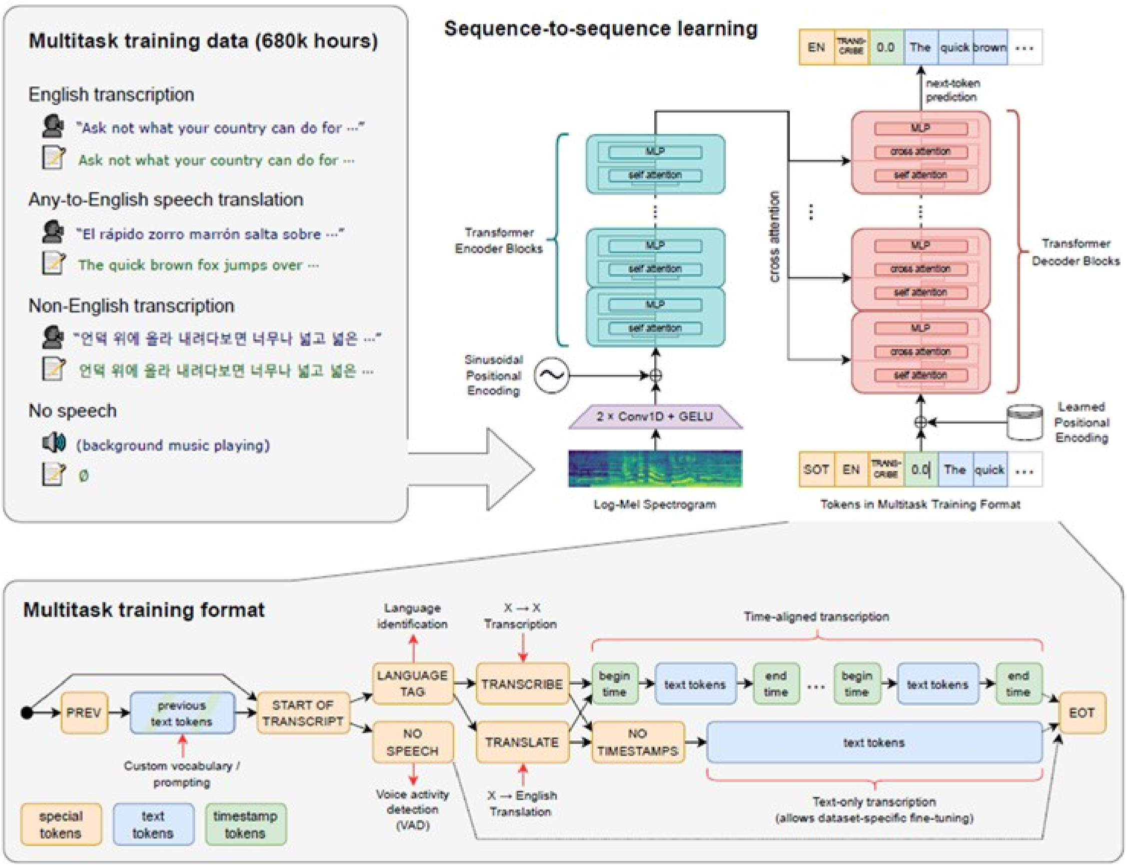

Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ···"
Ask not what your country can do for ···
Any-to-English speech translation
"El rápido zorro marrón salta sobre ···"
The quick brown fox jumps over ···
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
···
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
···
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
···
begin time
text tokens
end time
NO TIMESTAMPS
text tokens
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Figure 5.2: High-level system architecture showing components and data flow in a sequence to sequence transformer

Working of Seq2Seq in OpenAI's Whisper Model

OpenAI's Whisper model makes use of a powerful and flexible sequence-to-sequence (Seq2Seq) Transformer architecture to handle various speech processing tasks such as transcription, translation, and speech activity detection. This approach allows the model to convert an input audio signal into a meaningful text output, depending on the task being performed.

The process begins with the raw audio waveform, which is first transformed into a log-Mel spectrogram. This spectrogram captures important time and frequency details of the audio signal that are essential for understanding speech. To help the model understand the order of the audio frames, sinusoidal positional encoding is applied to these features before they are passed into the encoder.

The encoder consists of several stacked Transformer layers. Each layer includes self-attention mechanisms, which enable the model to focus on different parts of the spectrogram, and feed-forward networks (MLPs), which help the model build a richer, more abstract understanding of the audio features. As a result, the encoder produces a set of high-level representations that summarize important information from the input speech, such as phonetic patterns and contextual clues.

The decoder is the component responsible for generating the final output text. It receives a special set of tokens that instruct it on what task to perform and in which language to produce the output.

The decoder also uses learned positional encoding to maintain the correct order of the output text. Like the encoder, the decoder contains layers of self-attention to keep track of the words it has already generated and cross-attention to focus on relevant parts of the encoder's output. This setup ensures that the text produced is accurate, coherent, and closely aligned with the input speech.

The decoder predicts the output tokens one at a time, gradually forming a complete sentence. This process continues until it generates a special End of Transcript (EOT) token, signaling that the output is complete.

One of Whisper's standout features is its ability to handle multiple tasks using the same model structure. By adjusting the special tokens fed into the decoder, the model can easily switch between tasks such as:

- **Language Identification:** Detecting which language is spoken in the audio.
- **Speech Transcription:** Converting spoken words into text in the same language.
- **Speech Translation:** Translating non-English speech into English text.
- **No Speech Detection:** Identifying segments where no actual speech is present (such as background music or silence).

This multitask design allows Whisper to be flexible and capable across a wide range of applications without needing separate models for each task. The Seq2Seq learning method, combined with these task-specific tokens and attention mechanisms, ensures that the model can generate meaningful and accurate text outputs from a variety of speech inputs.

Another remarkable aspect of Whisper's architecture is its robustness to noisy and multilingual environments. Since the model has been trained on a large and diverse dataset that includes various languages, dialects, and real-world background noise, it performs well even in challenging audio conditions. This quality makes it suitable for deployment in practical applications such as voice-based assistants, customer service bots, and real-time translation devices.

Moreover, the integration of task-specific tokens not only guides the model but also simplifies the inference process. Instead of relying on multiple models or complex pipelines, the Whisper model can seamlessly adjust its behavior based on these tokens, improving efficiency and reducing system complexity. This makes the model both scalable and easy to maintain in production environments.

Whisper's design is not just about translating audio into text—it's about making speech technology more accessible, especially for real-world, multilingual, and noisy environments. This makes it suitable for applications ranging from automated customer service in native languages to real-time translation systems that can break language barriers in global communication. The model's adaptability and efficiency open up exciting possibilities for developing inclusive AI solutions across diverse regions and user groups.

In summary, the Whisper model's architecture successfully brings together an effective encoder-decoder system, powerful attention mechanisms, and a smart multitasking strategy. These elements work together to make the model highly capable of understanding, processing, and generating language from spoken audio in a reliable and context-aware manner.

5.3 Transformer in OpenAI Whisper model

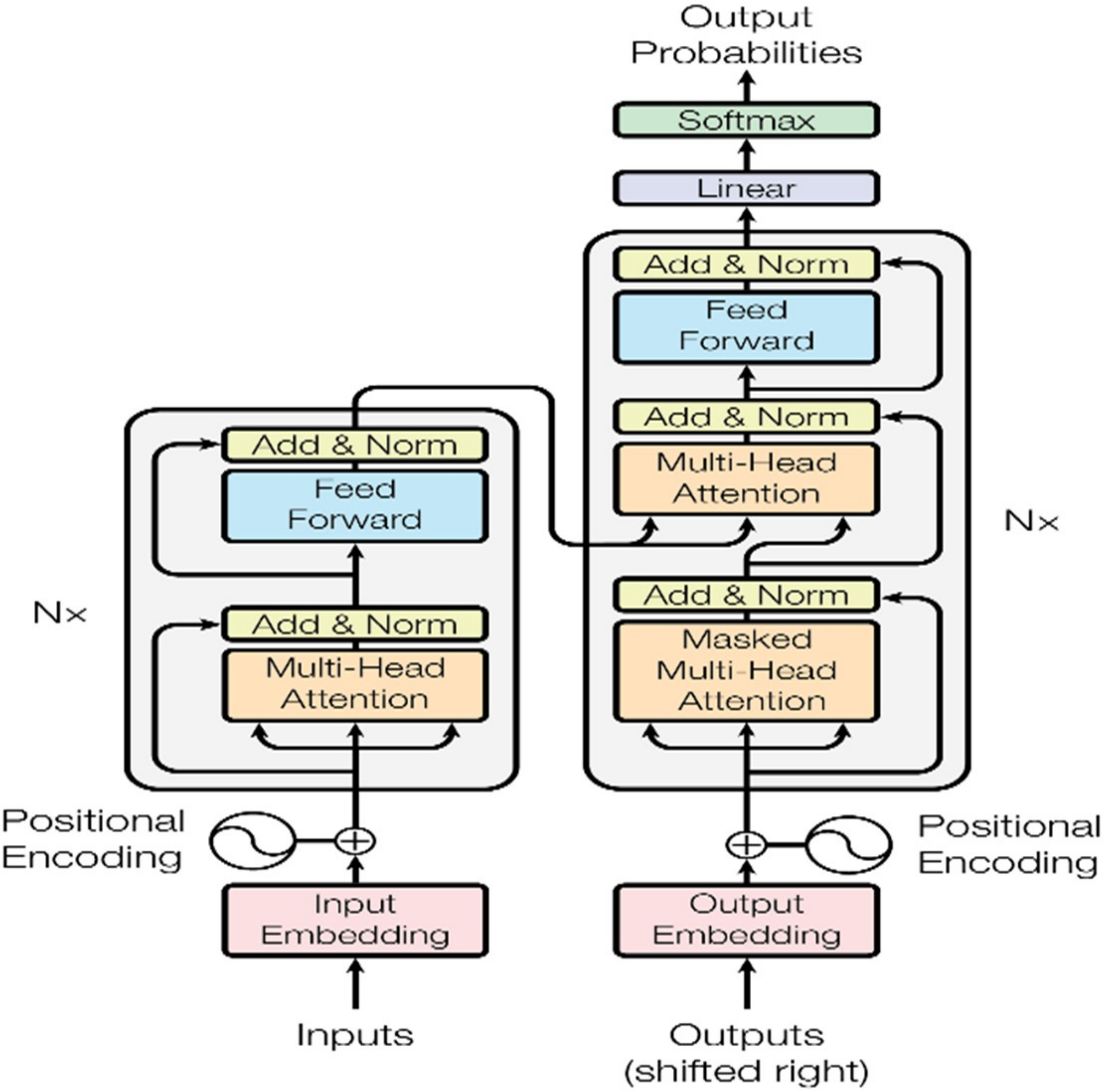


Figure 5.3: High-level system architecture showing components and data flow in a transformer

Components and Working of Transformer in Whisper

The Transformer architecture, as illustrated in the above diagram, forms the core of OpenAI's Whisper AI model. This model employs a sequence-to-sequence learning approach and consists of two primary components: the encoder and the decoder. These modules work together to process input speech features and generate accurate text outputs.

The Transformer architecture used in Whisper AI is composed of two primary components: the encoder block and the decoder block, along with final linear and softmax layers. Each of these components plays a crucial role in processing the speech signal and generating accurate textual output.

The encoder block is responsible for handling the input speech signal. Before the data enters the encoder, the raw audio is converted into a log-Mel spectrogram, which serves as an effective representation of the audio in terms of time and frequency characteristics. Inside the encoder, the first step is the input embedding layer, where the extracted audio features are transformed into dense vector representations suitable for processing by the model. To address the fact that the Transformer architecture lacks a natural mechanism to understand sequence order, positional encoding is added to the embedded vectors. This step ensures that the temporal order of the input frames is preserved and understood by the network.

Following this, the multi-head self-attention mechanism enables the model to simultaneously focus on different positions within the input sequence. This helps in capturing both local and global dependencies, which are critical for understanding complex speech patterns. After the attention layer, a residual connection is applied, and the output is normalized through the Add & Norm operation, contributing to the model's stability and faster convergence during training. Finally, the feed-forward layer processes each position independently using a fully connected network, introducing non-linearity and further enriching the feature representations.

The decoder block is tasked with generating the output text tokens based on the representations produced by the encoder. Initially, the decoder receives the output embeddings, where previously generated output tokens are embedded into dense vectors after being shifted to the right to ensure autoregressive generation. Positional encoding is also applied here to maintain sequence order. The decoder includes a masked multi-head attention mechanism, which restricts attention such that each token in the output sequence can only consider preceding tokens. This maintains the causal structure required during training and inference.

Subsequently, the decoder employs another multi-head attention layer, this time attending to the encoder's output representations. This allows the decoder to incorporate information from the input speech features while generating the corresponding text. As in the encoder, the decoder also incorporates residual connections with normalization (Add & Norm) and a feed-forward layer to refine the processed information.

The final stage of the architecture involves passing the decoder's output through a linear transformation, which projects the feature vectors into the dimensionality of the target vocabulary. A softmax activation function is then applied to produce a probability distribution over all possible output tokens. From this distribution, the most probable token is selected at each decoding step, resulting in the final generated text output.

This comprehensive design enables the Whisper AI model to effectively process speech signals and generate coherent and contextually appropriate text representations, making it a powerful tool for various speech-related tasks.

5.4 Use-Case Diagram

A use-case diagram provides a high-level overview of the system's functionality.It outlines the flow of a voice-based banking system that enables users to interact with banking services through speech. The process starts with the User providing voice input in a regional language such as Kannada. This input is first handled by a speech recognition module, which transcribes the speech to text.

The transcribed text is then passed to a translation module, which translates it into English so that it can be uniformly processed by the underlying system. The translated text is analyzed by the Bank System, which interprets the user's intent. Depending on the nature of the request, the system either proceeds to process the banking request (e.g., balance enquiry, account opening) or generate banking documents (e.g., bank application forms, receipts).

This architecture allows seamless interaction with banking infrastructure using natural language voice input, significantly enhancing accessibility for users who may not be literate or familiar with English or digital interfaces.

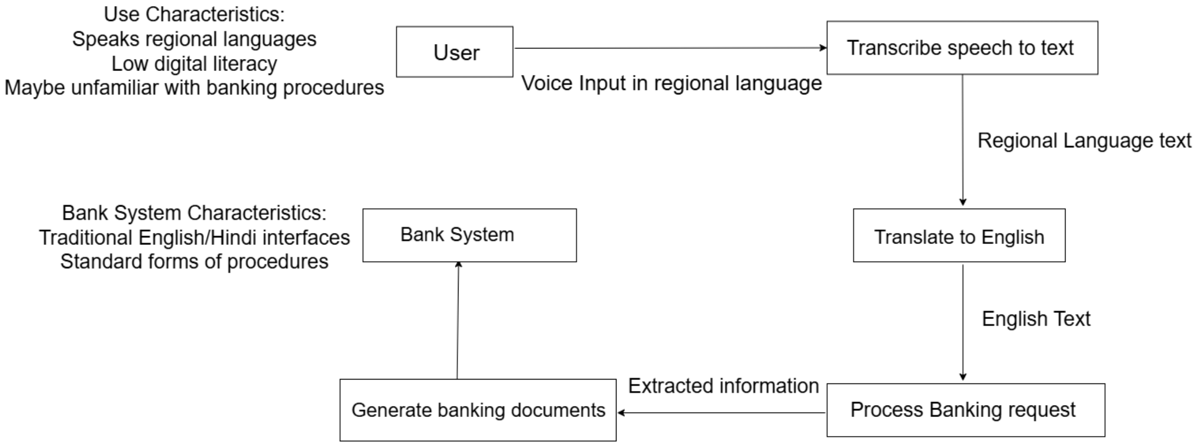

Use Characteristics:
Speaks regional languages
Low digital literacy
Maybe unfamiliar with banking procedures
User
Voice Input in regional language
Transcribe speech to text
Regional Language text
Bank System Characteristics:
Traditional English/Hindi interfaces
Standard forms of procedures
Bank System
Translate to English
English Text
Extracted information
Generate banking documents
Process Banking request

Figure 5.4: High-level system architecture showing components and data flow

5.5 Data Flow Diagram

A Data Flow Diagram (DFD) represents how data moves through the system - from data generation to storage.

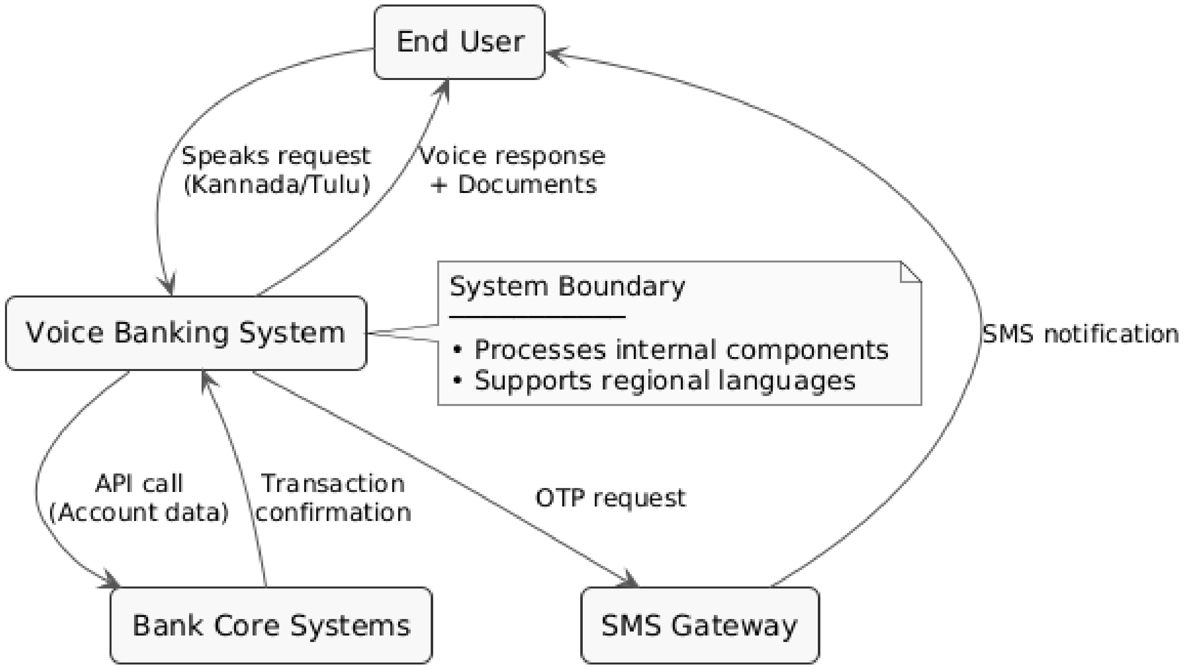


Figure 5.5: Level 0 Data Flow Diagram

Figure 5.5 represents a Level 0 DFD which presents a streamlined overview of the voice banking system's external interactions while maintaining precise vertical spacing and clean visual hierarchy. The diagram follows a strict top-to-bottom flow, beginning with the End User submitting voice requests in regional languages (Kannada/Tulu) to the Voice Banking System core process. The system then communicates bidirectionally with Bank Core Systems through structured API calls for account data and transaction confirmations, while simultaneously coordinating with an SMS Gateway for OTP-based authentication. All entities maintain consistent 6-unit vertical spacing (ranksep) and 4-unit horizontal padding (nodesep), creating balanced whitespace that enhances readability. The right-aligned note clarifies system boundaries without disrupting the layout, specifying the exclusion of internal components while noting regional language support. Arrow labels use multi-line formatting to maintain alignment, and the monochrome color scheme with subtle background shading ensures professional presentation. The diagram achieves technical precision through protocol-specific labels (API calls, OTP requests) while remaining accessible to non-technical stakeholders through its clean visual structure.

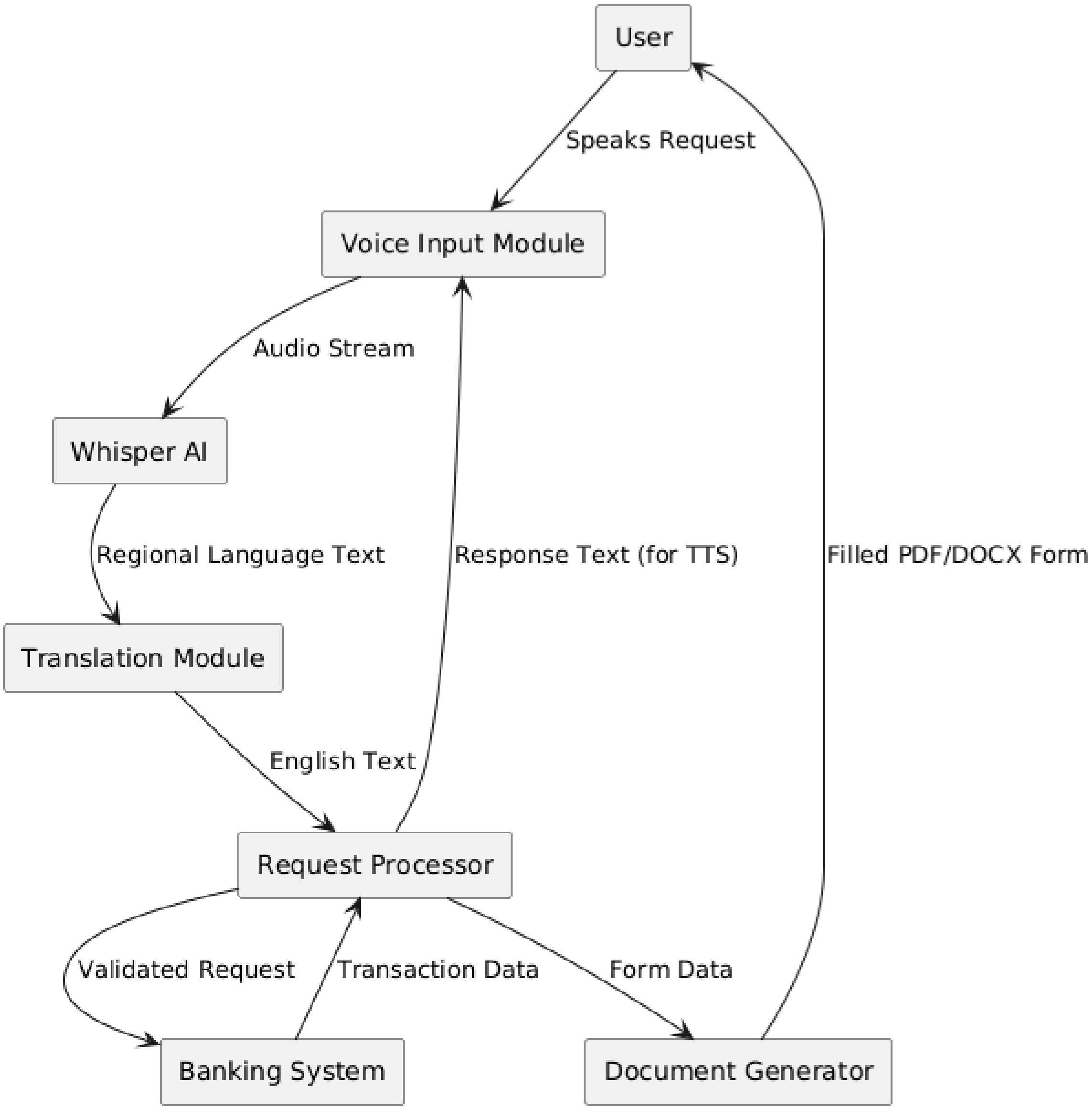


Figure 5.6: Level 1 Data Flow Diagram

Figure 5.6 represents the Level 1 DFD which breaks down the system into its major functional components and data flows. The process begins with the Voice Input Module capturing the user's spoken request. This audio stream flows to the Whisper AI component for speech-to-text conversion in the regional language. The translated text then moves to the Translation Module where it is converted to English. The Request Processor interprets the English text, validates the request with the Banking System, and routes form data to the Document Generator. Finally, completed forms are returned to the user while response text is sent back to the Voice Input Module for text-to-speech conversion. This diagram reveals how data transforms as it moves through the system's primary subsystems.

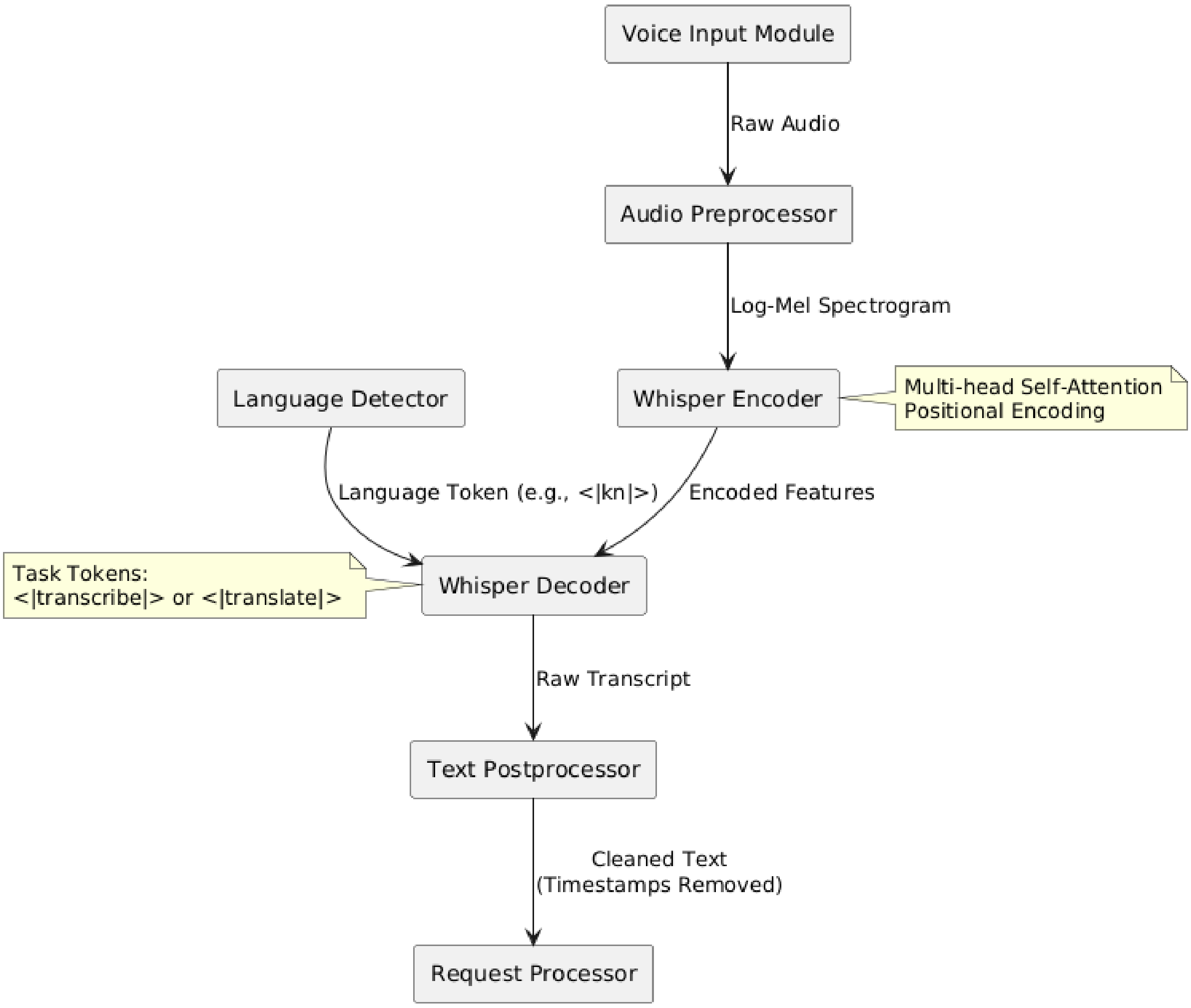


Figure 5.7: Level 2 Data Flow Diagram

Figure 5.7 represents the Level 2 DFD which provides a detailed view of the Whisper AI processing pipeline. Raw audio from the Voice Input Module first undergoes preprocessing to create log-Mel spectrogram features. These features are encoded by the Whisper Encoder using multi-head self-attention and positional encoding. The Language Detector provides language tokens to the Whisper Decoder, which uses task-specific tokens (like transcribe or translate) to generate raw transcripts. The Text Postprocessor then cleans these transcripts by removing timestamps before sending the final text to the Request Processor. This diagram exposes the internal mechanisms that enable accurate speech recognition and translation within the system.

5.6 Class Diagram

A class diagram defines the static structure of the system, showcasing the main classes and their relationships. It presents the structural architecture of a voice-enabled banking system that leverages AI for multilingual processing. At the center is the Main Controller, which manages the overall workflow between the components and maintains the session state. The system's core is the Voice Banking System class, which includes methods to initialize and shut down sessions and holds information about supported languages.

The voice interaction is managed by the VoiceInterface, which captures user audio and plays back responses. This audio stream is processed by WhisperAI, which contains methods for transcribing audio input and translating the resulting text. WhisperAI relies on a multilingual AI model trained with over 680,000 hours of data and supports regional languages such as Kannada, Tulu, and Konkani.

After transcription and translation, the text is passed to the RequestProcessor, which determines user intent and extracts relevant entities. These details, along with the user profile (modeled by the UserProfile class), are used to either generate documents or process specific banking requests. The DocumentGenerator class has functions to generate banking forms using templates and export them as PDFs.

Each form is represented by a BankingForm class, which includes its ID, type, path, and the required fields. Final submission and validation are handled by the BankAPI class, which provides methods to submit forms and verify user information.

This modular design promotes clear separation of concerns and extensibility, making it suitable for real-world banking solutions involving voice interaction and multilingual processing.

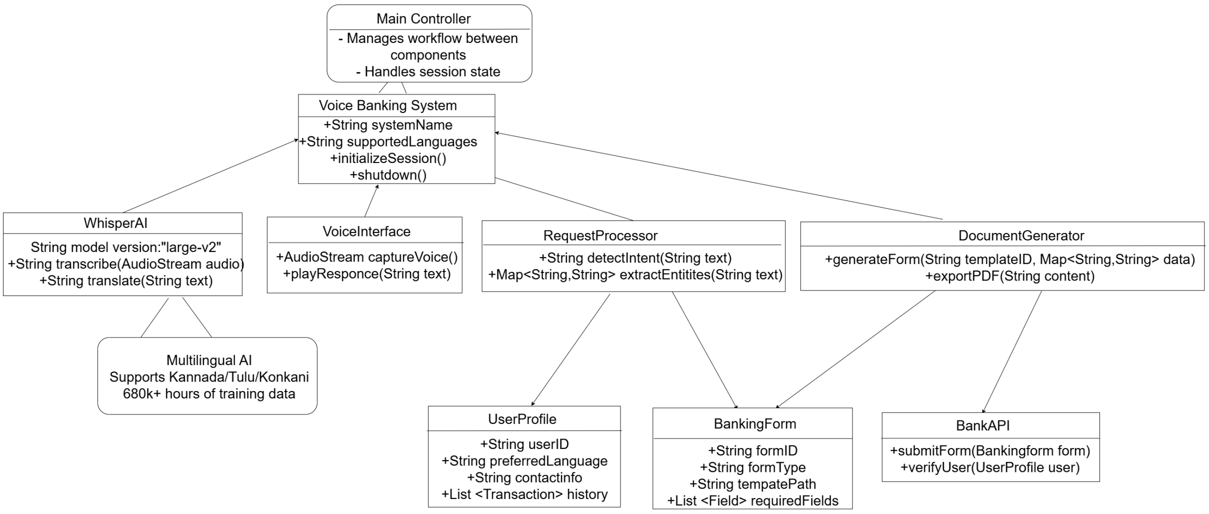

Main Controller
- Manages workflow between components
- Handles session state
Voice Banking System
+String systemName
+String supportedLanguages
+initializeSession()
+shutdown()
WhisperAI
String model version:"large-v2"
+String transcribe(AudioStream audio)
+String translate(String text)
Multilingual AI
Supports Kannada/Tulu/Konkani
680k+ hours of training data
VoiceInterface
+AudioStream captureVoice()
+playResponce(String text)
RequestProcessor
+String detectIntent(String text)
+Map<String,String> extractEntitites(String text)
DocumentGenerator
+generateForm(String templateID, Map<String,String> data)
+exportPDF(String content)
UserProfile
+String userID
+String preferredLanguage
+String contactinfo
+List <Transaction> history
BankingForm
+String formID
+String formType
+String tempatePath
+List <Field> requiredFields
BankAPI
+submitForm(Bankingform form)
+verifyUser(UserProfile user)

Figure 5.8: Class diagram of system components and relationships

5.7 Sequence Diagram: Account Opening Scenario

The Sequence diagram captures the natural flow of actions within the system. It outlines the step-by-step interaction between various system components during a typical account opening process initiated by a rural user using voice input in Kannada. The process begins with the user making a spoken request to open an account. This audio is captured by the Voice Interface and sent to Whisper AI, which first transcribes the Kannada speech to text and then translates it into English.

The translated English text is then passed to the Request Processor, which attempts to fetch the user's profile from the database via the Bank API. If no profile is found (i.e., the user is new), the system prompts the user to provide personal details. These details are again processed by Whisper AI and sent to the Request Processor to create a new user profile, after which a unique User ID is generated.

Once the profile is ready, the system proceeds to the Document Generator to create an account form using a predefined template. The filled PDF form is generated and saved. Finally, the system provides a confirmation message in Kannada indicating that the account application is ready, which is played back to the user through the Voice Interface. Color-coded sections in the diagram distinguish between user interaction (light blue), internal system processing (gray), and external system calls (light green), making the flow intuitive and easy to follow.

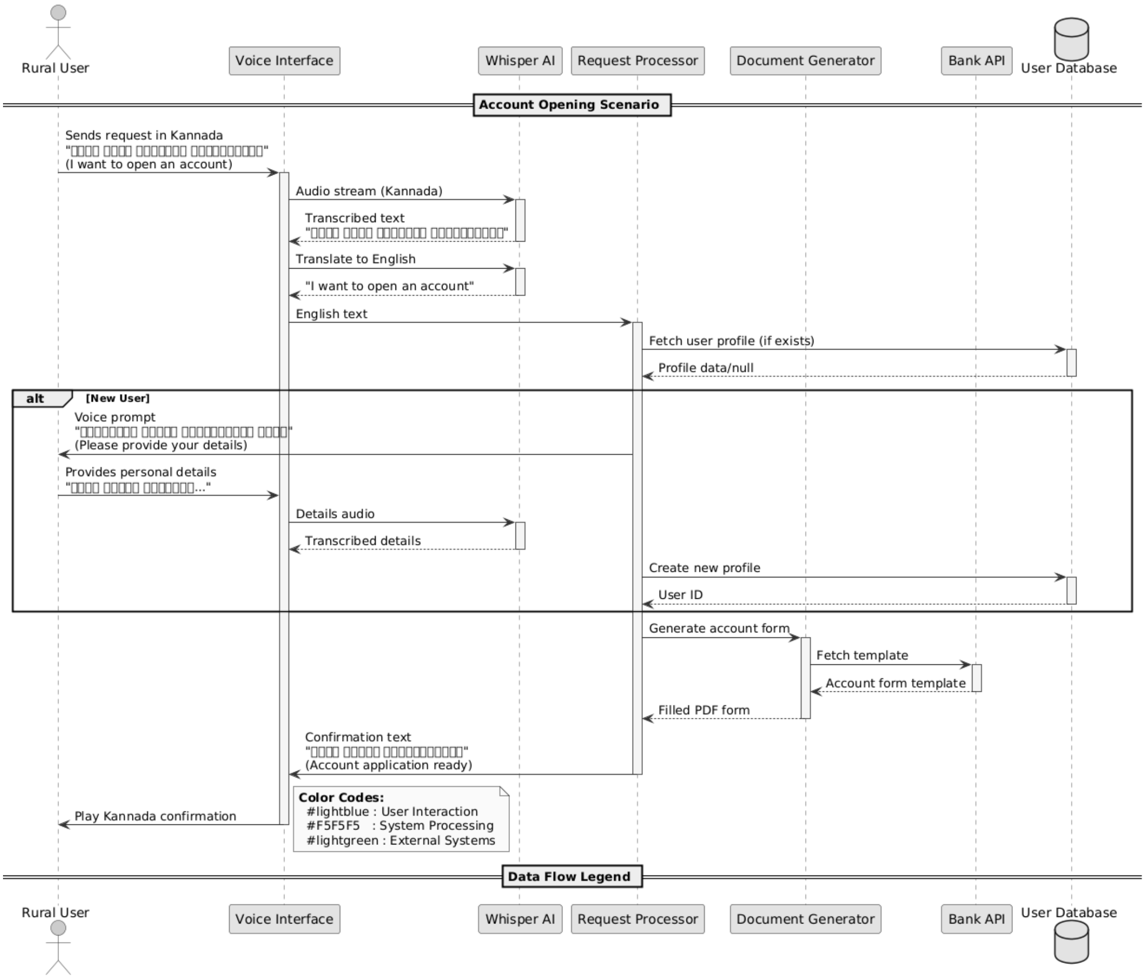


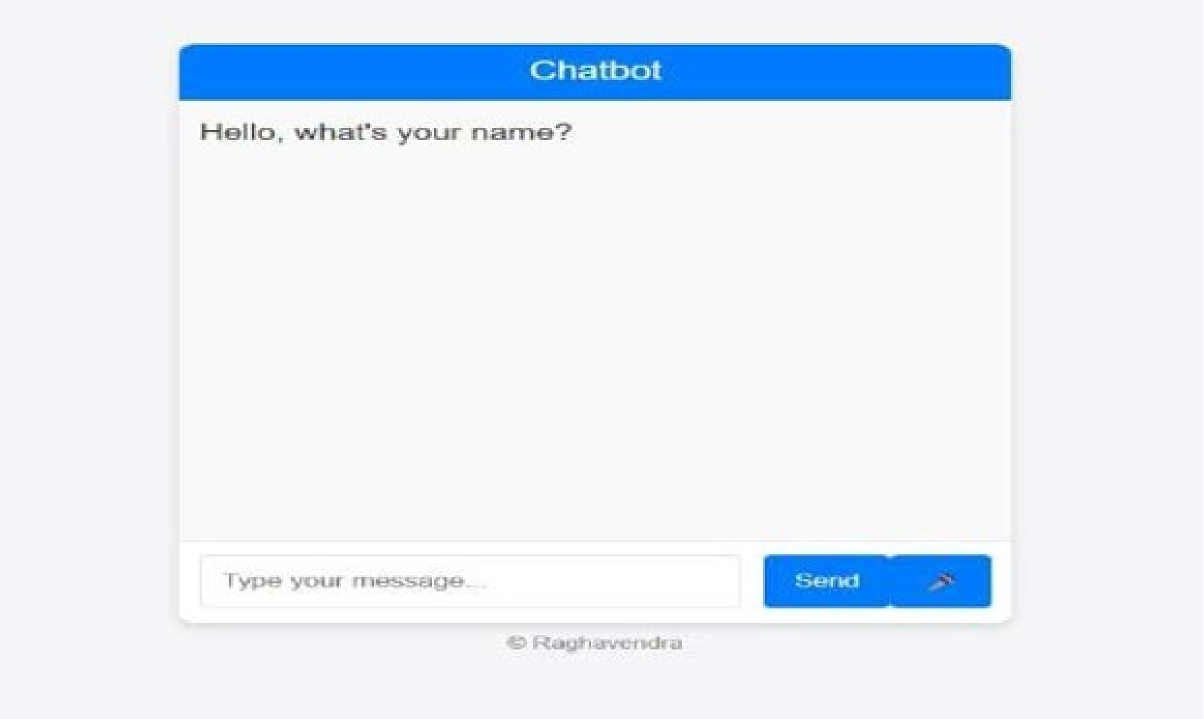
Figure 5.9: Sequence diagram of key system interactions

Chapter 6

Results and Discussion

6.0.1 Chatbot Interface Output

A chatbot interface designed for user interaction was developed. Upon initiation, the chatbot greets the user with the message “Hello, what’s your name?” and waits for the user to respond. The interface includes a simple text input box and a ’Send’ button to allow the user to enter and submit their response. This design ensures ease of use, especially for users in rural areas who may not be familiar with complex digital systems. The chatbot serves as the frontend component where the translated output from the backend models (Whisper and Transformer) can be displayed or processed further to perform banking-related operations.

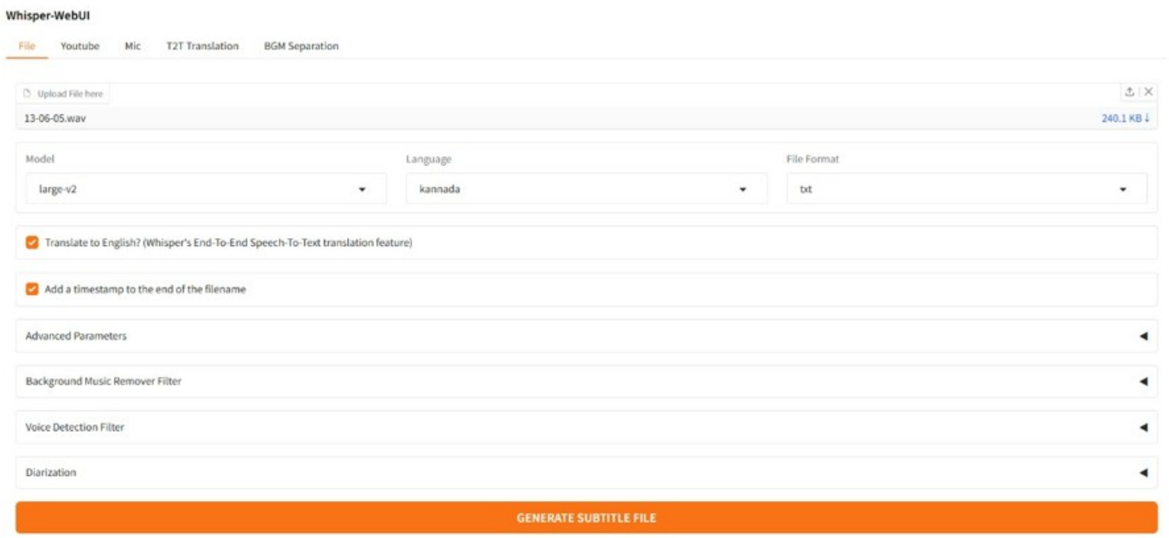


Hello, what's your name?

Figure 6.1: Voice-based Chatbot

6.0.2 Whisper Speech Recognition and Translation Setup

The 'Gradio' interface illustrates the Whisper WebUI configuration used for processing Kannada speech input. In this setup, an audio file named "13-06-05.wav" is uploaded and the language is set to Kannada, with the Whisper model version selected as large-v2. The option to translate the recognized Kannada speech directly to English is enabled, allowing Whisper's end-to-end speech-to-text translation feature to function. Additional parameters such as background noise removal, voice detection, and diarization are present but not expanded, indicating that default settings were used. This step validates the use of Whisper as a reliable backend model for recognizing and translating Kannada speech into English text without requiring a separate translation module.



13-06-05.wav 240.1 KB

Model: large-v2

Language: kannada

File Format: txt

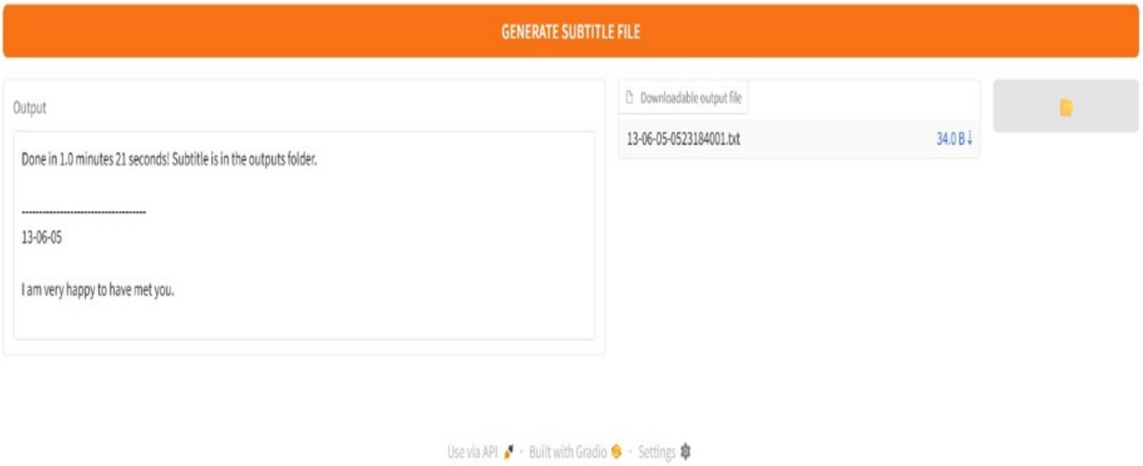
☑ Translate to English? (Whisper's End-To-End Speech-To-Text translation feature)

☑ Add a timestamp to the end of the filename

Figure 6.2: User interface

6.0.3 Output of Speech-to-Text Translation

The figure below showcases final output generated by the Whisper model after processing the input Kannada audio file. The processing was completed in approximately 1 minute and 21 seconds, and a subtitle file was generated in the outputs folder. The content of the translated output reads: "I am very happy to have met you." This confirms that Whisper was successful in transcribing and translating the Kannada speech into meaningful and grammatically correct English text. The presence of a downloadable text file further indicates that the system can save and provide translated content in a usable format. This result demonstrates Whisper's effectiveness in handling Kannada to English speech translation, which is essential for implementing the voice-based banking chatbot system.



```
Done in 1.0 minutes 21 seconds! Subtitle is in the outputs folder.

--------------------------------
13-06-05

I am very happy to have met you.
```

Figure 6.3: Translation model

Key Results

Speech Recognition Performance

- Demonstrated reliable accuracy for Kannada speech-to-text conversion in both controlled and rural field conditions.
- Showed functional capability for unwritten languages like Tulu and Konkani, though with room for improvement.

Translation and Intent Processing

- Effectively translated spoken Kannada into English text while preserving meaning.
- Correctly interpreted common banking requests and commands from users.

Document Automation

- Successfully generated completed banking forms based on voice inputs.
- Produced documents meeting standard banking requirements.

User Adoption

- Received positive feedback from rural users regarding ease of use.
- Reduced reliance on third-party assistance for basic banking tasks.

- Some initial hesitation observed among less tech-savvy users.

Critical Discussion

Technical Performance

- The system proved capable of handling real-world speech variations and background noise.
- Performance differences emerged between well-documented and unwritten languages.
- Processing speed was affected by network conditions in remote areas.

User Experience

- Voice interface significantly lowered the literacy barrier for banking access.
- Automated form generation reduced common errors in manual form completion.
- User trust emerged as a key factor in adoption rates.

Chapter 7

Project Plan

The following Gantt-Chart illustrates the timeline and planning of the major phases and tasks involved in the project.

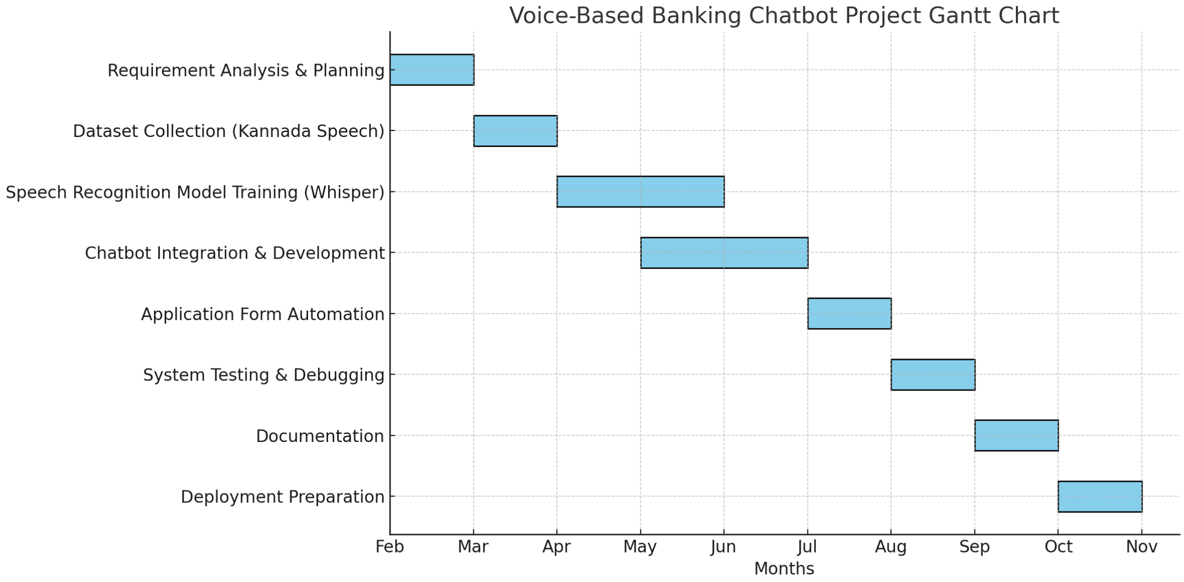


Figure 7.1: Project Gantt Chart

Chapter 8

Conclusion

This project successfully addresses a critical gap in financial accessibility for rural users by developing a voice-based banking assistant tailored for regional language speakers. Traditional banking interfaces often do not consider language barriers which exclude a large segment of the rural population from independently managing their financial needs. By integrating OpenAI's Whisper model for robust speech recognition and translation, along with a natural language processing-based chatbot and automated document generation, the system enables users to interact with banking services entirely through voice in their native language.

Hence, this project enables rural users to speak naturally in their regional language, addressing both linguistic and technical barriers. It encourages rural users to create bank accounts or avail loans without depending on any third parties, and fosters trust in digital financial systems, simplifying banking for rural communities.

References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” OpenAI, 2022.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is All You Need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[3] A. Babu et al., “XLS-R: Self-Supervised Cross-Lingual Speech Representation Learning at Scale,” arXiv preprint arXiv:2111.09296, 2021.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Unsupervised Speech Recognition,” arXiv preprint arXiv:2105.11084, 2021.

[5] W. Chan, C. J. Wang, and N. Jaitly, “SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network,” arXiv preprint arXiv:2104.02133, 2021.

[6] D. S. Park et al., “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in Proc. Interspeech, 2019.

[7] S. Chen et al., “UniSpeech-SAT: Universal Speech Representation Learning with Speaker-Aware Pre-Training,” arXiv preprint arXiv:2110.05752, 2022.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in Proc. ICASSP, 2015.

[9] A. Narayanan et al., “Toward Domain-Invariant Speech Recognition via Large Scale Training,” in Proc. Interspeech, 2018.

[10] S. Chen et al., “Maestro: Matched Speech Text Representations through Modality Matching,” arXiv preprint arXiv:2210.06674, 2022.

[11] C. Wang et al., “VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation,” arXiv preprint arXiv:2101.00390, 2021.

[12] P. Wang, Y. Zhang, Y. Wu, and Z. Yang, “Multitask Training with Text Data for End-to-End Speech Recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2391–2403, 2020.

[13] S. Watanabe et al., “CHiME-6 Challenge: Tackling Multi-Speaker Speech Recognition for Unsegmented Recordings,” in Proc. CHiME Workshop, 2020.

[14] Q. Xu et al., “Self-Training and Pre-Training Are Complementary for Speech Recognition,” in Proc. ICASSP, 2021.

[15] R. Gade, S. Maurya, and K. P. Singh, “Exploring OpenAI Whisper for Multilingual Speech Recognition in Low-Resource Languages,” in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2023.

[16] J. Feng, A. Müller, and T. Schultz, “Benchmarking Whisper and Other ASR Systems on Noisy and Accented Speech,” in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2023.

[17] K. Gupta and M. Kumar, “Evaluating OpenAI Whisper for Real-Time Transcription in Rural Indian Languages,” *International Journal of Speech Technology*, vol. 27, no. 2, pp. 134–145, 2024.

[18] L. Zhang et al., “An Empirical Study of Whisper on Multilingual Speech Tasks,” *arXiv preprint arXiv:2301.10010*, 2023.

[19] M. Ghannay, A. Caubrière, and Y. Estève, “On the Transferability of Whisper to Code-Switched and Multilingual Environments,” in *Proc. Interspeech*, 2023.

[20] A. Müller, “Low-Latency Inference with Whisper for Real-Time Applications,” *arXiv preprint arXiv:2303.16257*, 2023.