International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 15, Issue 5 [May 2026] PP: 65-73

Developing a speech-to-text application on the Android platform

Tran Thi Thanh, Nguyen Thi Huong
Department of Computer engineering, Faculty of Electronic Engineering, Thai Nguyen University of Technology, Thai Nguyen, 250000, Vietnam
Corresponding author: Nguyen Thi Huong

Abstract
In this article, a speech-to-text system is developed and implemented on the Android platform to provide an efficient and reliable solution for converting spoken language into written text. The proposed system offers several significant advantages, particularly in terms of system architecture, database organization, and operational efficiency. One of the key strengths of the system lies in its well-structured and highly optimized database design, where the database tables are logically organized to ensure consistency, scalability, and efficient data handling. This coherent database structure enables the system to manage large volumes of audio recordings and converted text data effectively while minimizing storage redundancy and improving retrieval speed.In addition, the system is designed with an advanced data management approach that enhances overall performance during data processing and access operations. The application supports the storage of both audio input files and their corresponding transcribed text outputs, allowing users to review, reuse, and manage historical conversion records conveniently. By optimizing the storage and retrieval mechanisms, the system achieves faster processing times and improved responsiveness, which contributes to a smoother user experience.
Keywords*: Speech-to-text; application, Android platform; speech record; language translation*

Date of Submission: 05-05-2026 Date of acceptance: 16-05-2026

I. INTRODUCTION

In recent years, Automatic Speech Recognition (ASR) technology has made tremendous strides, with widespread applications in many fields such as virtual assistants, conference recording, and disability support [1]. In particular, the emergence of the Whisper model (developed by OpenAI) has created a major turning point. With its advanced Transformer architecture and training on hundreds of thousands of hours of multilingual data, Whisper demonstrates excellent noise filtering and recognition capabilities in various complex audio environments without prior adjustment (zero-shot) [2]. However, despite its overall power, the original Whisper model still reveals many limitations when processing the Vietnamese language [3]. Vietnamese is a monosyllabic, highly analytical language, particularly complex with its six tones, diverse dialects (Northern, Central, and Southern), and a large number of homonyms. Due to the limited amount of Vietnamese data in Whisper's training set, the model frequently encounters serious errors such as: incorrect tone recognition (completely altering the meaning of sentences), hallucinations (meaningless word repetition), and failure to recognize specialized terminology or proper nouns. This significantly reduces accuracy (high Word Error Rate - WER) and hinders the model's practical application in Vietnam [4]. Recognizing this technological gap, researching intervention and optimization techniques to enhance Whisper's performance specifically for the Vietnamese language is a pressing need. Instead of building an ASR model from scratch at enormous cost, the optimal approach is to leverage Whisper's unique audio capabilities and combine them with specialized natural language processing (NLP) techniques [5]. Hence, a small-scale database system was developed to support speech-to-text conversion, focusing on basic functions such as conversion, data storage, and providing a user interface. Use SQLite and supporting tools to design and implement the database. SQLite is used to design and manage the system's database, from storing user information, audio files, and converted text, to querying data and generating statistical reports. Use Java to build the application interface and connect to the database. The presented work focuses on applying and evaluating methods such as prompting, fuzzy matching, and post-processing language model integration to minimize tone errors and standardize output text formatting, thereby creating a robust and highly applicable Vietnamese ASR system [6].

II. MATERIAL AND METHODS

2.1. Input information of the system

The Input information is what users provide or do on the website interface, in whichaudio data is the most important input information. Users can provide audio information through:

- Direct recording via browser: Users speak into the microphone and the web browser records the sound (via the browser's API such as Web Audio API or MediaStream Recording API). Audio data is transmitted in real time or in small segments to the server.

- Upload audio/Video file: Users select and upload an audio file (e.g., .wav, .mp3, .m4a, .flac) or video file (e.g., .mp4, .avi - the website will only retrieve the audio).

For configuration options, users can select or enter:

- Language: Select the language of the input audio if the system supports multiple languages.

- Subject Domain (if applicable): Select a relevant subject to improve accuracy (e.g., medical, legal).

- Other Settings: Enable/disable profanity filtering, select desired output type (e.g., whether or not to include timestamps).

- User Information: Includes user information (username, password, email, etc.), user permissions, and conversion history.

2.2. Output information of the system

Based on the input audio data and the user's specified language, the system will output an accurate text file to provide the best, fastest, and most precise service.

- Display format: This can be a simple block of text or a more aesthetically pleasing format (with punctuation, capitalization).

- Metadata Display:

- Display timestamps for each word or sentence.
- Mark different speakers (Speaker Diarization) if the backend supports it.
- Display Confidence Score (e.g., blur out less reliable words).
- Display a list of alternative results (N-best list) if available.

- Processing and response status: The website needs to inform the user what the system is doing:

+ "Listening"
+ "Uploading file"
+ "Processing audio"
+ "Complete!"
+ Display a progress bar.
+ Error messages: "Microphone is not found", "File format is not supported", "Processing failed, please try again".

The flowchart of the input/output information of a speech-to-text conversion system is presented in Figure 1

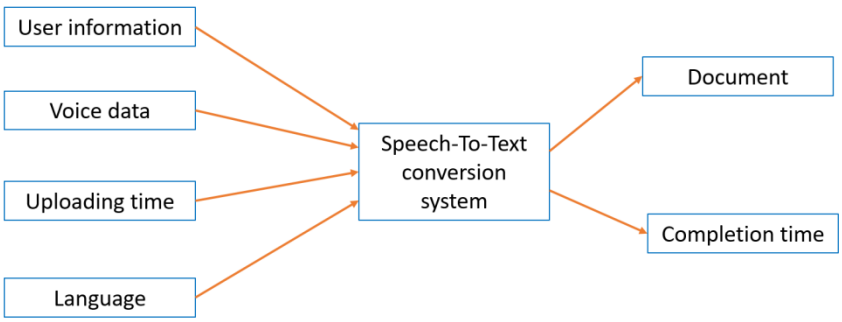


Figure1. Input and output information of the Speech-To-Text conversion system

2.3System structure diagram

A Speech-To-Text (STT) conversion system is designed to transform spoken language into readable text through several processing stages. The system begins with audio input sources such as microphones, phone calls, or uploaded audio files. The audio is first captured by the audio capture module, which converts the incoming sound into a digital signal suitable for processing. The structure diagram of the proposed system is illustrated in Figure 2.

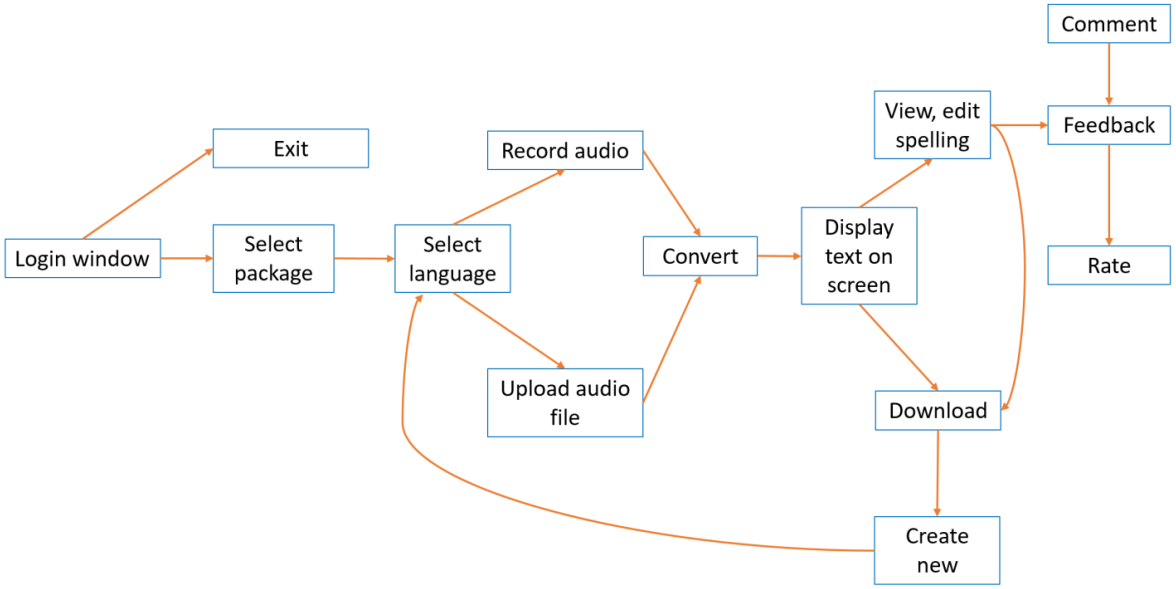


Figure2. Structure diagram of the system

2.4 Database design

To address issues such as data redundancy, inconsistencies, duplication, and ambiguity, it is necessary to check, review, and filter the data before including it in the table. This process is called normalization and is performed in three steps: 1st Normal form (1NF), 2nd Normal form (2NF), and 3rd Normal form (3NF).

- **1st Normal form (1NF)**: A relation is considered to be in first normal form if all attributes are singular, meaning there is no set of identical attributes (repeated attributes). According to the definition of functional dependencies, if there exists a set of repeated attributes, then at any given time, for every key value, there cannot be a unique value for each of the other attributes in the table. Therefore, returning to first normal form means eliminating the group of repeated attributes.

- 2nd Normal form (2NF):A relation is said to be in second normal form (2NF) if it is in first normal form (1NF) and all functional dependencies between the key and attributes are elementary, meaning that every attribute must be functionally dependent on the entire key, not just a part of it. Therefore, to bring a relation to 2NF, all partial functional dependencies on the key must be removed. Any table (entity) with only one attribute as the key is considered to be in 2NF.

- 3rd Normal form (3NF):A relation is said to be in 3NF if it is in 2NF and the functional dependencies between the key and other attributes are direct, or in other words, each attribute is not functionally dependent on any attribute in the relation other than the key.

III. RESULTS AND DISCUSSION

Android Studio is the official Integrated Development Environment (IDE) for Android application development. It is developed by Google and provides developers with powerful tools for designing, coding, testing, and debugging Android applications efficiently. Android Studio supports programming languages such as Java and Kotlin and includes features like a code editor, emulator, Gradle build system, and intelligent code completion.

With its user-friendly interface and advanced development tools, Android Studio helps developers create high-quality mobile applications for smartphones, tablets, and other Android devices. It also supports modern Android technologies such as Jetpack Compose, Firebase integration, and real-time performance analysis, making it one of the most widely used platforms for Android app development. Some key features include

- Gradle-based application build tools (instead of Maven).
- Fast debugging and error correction, Android-oriented.
- Convenient drag-and-drop screen editing tools.
- Integrated wizards to help developers create applications from pre-made templates.
- Google Cloud Platform integration, easily integrating with Google Cloud Messaging and Google App Engine.

3.1Main library used in the program

The main libraries used in the program are shown in below

- **faster-whisper:**
  - + An optimized version of Whisper, rewritten based on C++/CUDA.
  - + Helps increase inference speed and reduce resource usage.
  - + Suitable for deployment in real-world environments (server, API).
- **FFmpeg:**
  - + Powerful multimedia processing toolset.
  - + Supports audio format conversion (mp3, wav, flac, etc.).
  - + Allows audio normalization (sample rate, mono, bitrate).
- **Librosa:**
  - + A Python library specializing in audio signal processing and analysis.
  - + Supports feature extraction such as MFCC and Spectrogram.
  - + Used in audio research and data processing.
- **Pydub:**
  - + Simple and easy-to-use audio processing library.
  - + Supports: Cutting, merging audio, and adjusting volume.
  - + Often used in conjunction with FFmpeg.
- **Express.js:**
  - + A popular Node.js backend framework.
  - + Used to build REST APIs.
  - + Lightweight, easily extensible, and suitable for web systems.

3.2 Program installation

The system receives input data in the form of voice audio signals, which can come from:
+ Audio files: wav, .mp3, .flac, .m4a
+ Direct recording from a microphone
The output is text converted from speech. Figure 3 presents the Backend settings screen for the program, and Figure 4 shows the Frontend setup screen for the program.

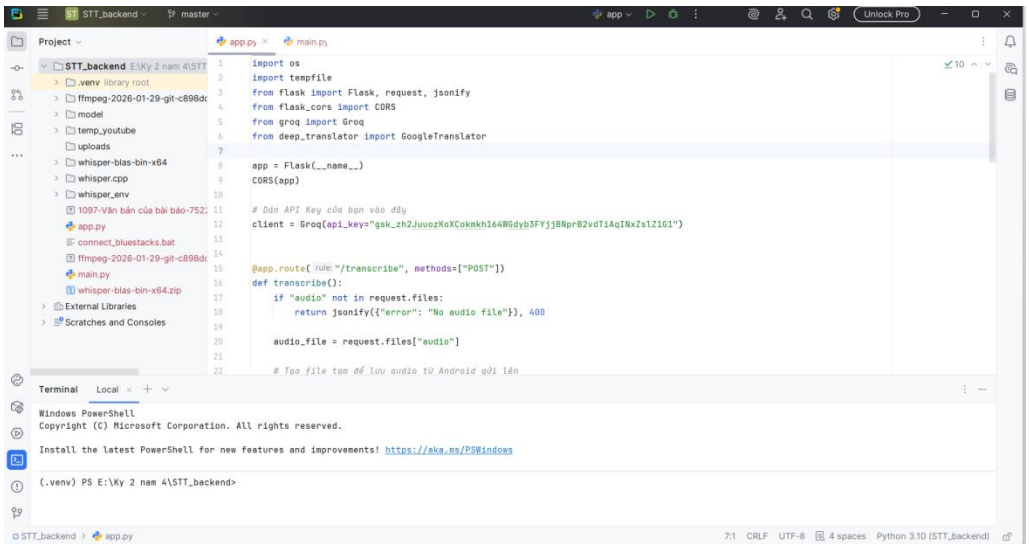


Figure3.*Backend settings screen for the program*

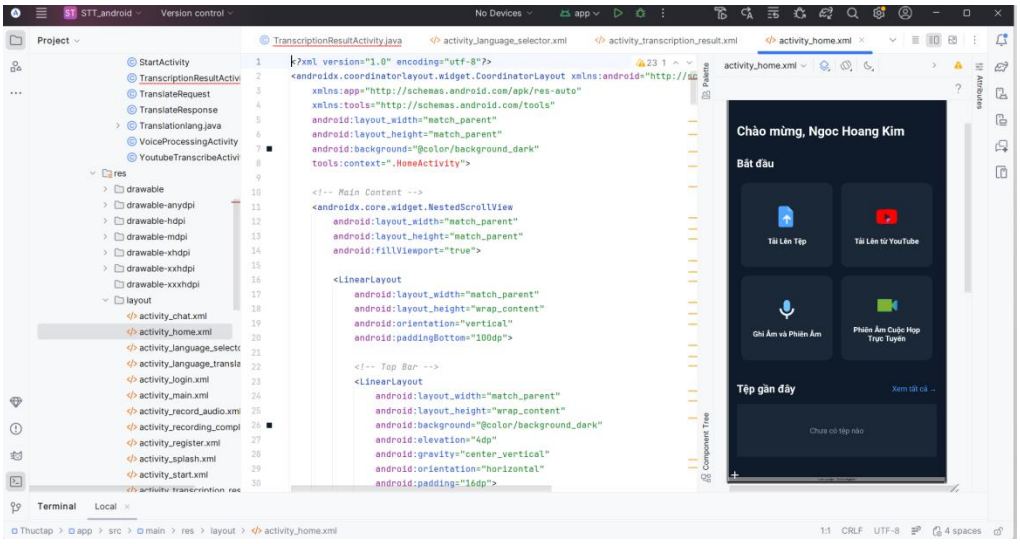
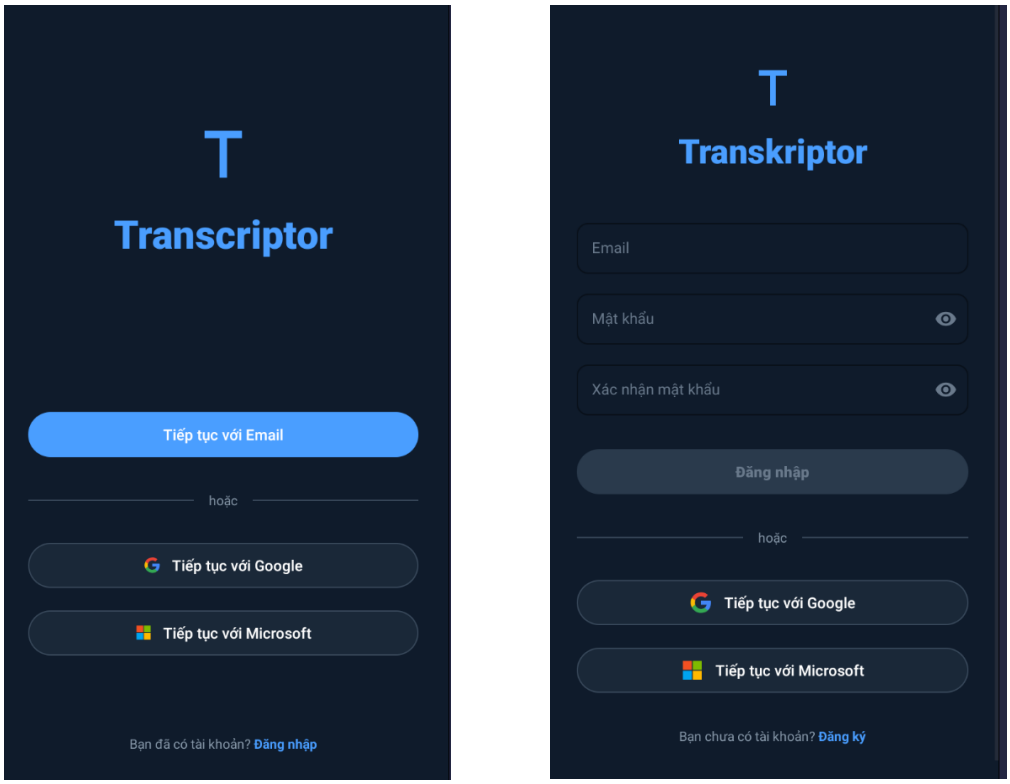


Figure 4.*Frontend setup screen for the program*

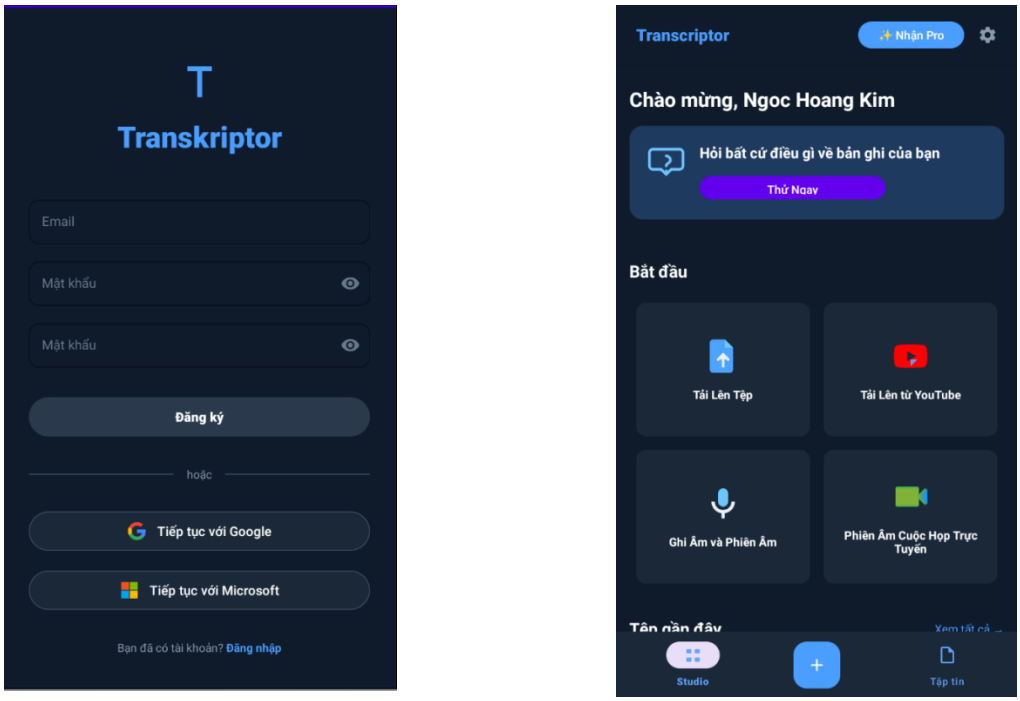
3.3 Program execution results

The following figures illustrate the results of the program execution.Figure 5 presents the startup and login interfaces. Figure 6 shows the register and home screen interfaces in Vietnamese. Figure 7 shows the functions of the proposed application including file upload and voice recording functions.Figure 8 shows how the videos obtained from YouTube are put in the applicationand the language is selected. The speech-to-text result is shown in Figure 9, and it can be also translated into the other language (Figure 10). Additionally, Artificial Intelligence (AI) chat function is also integrated in the proposed program (Figure 11).



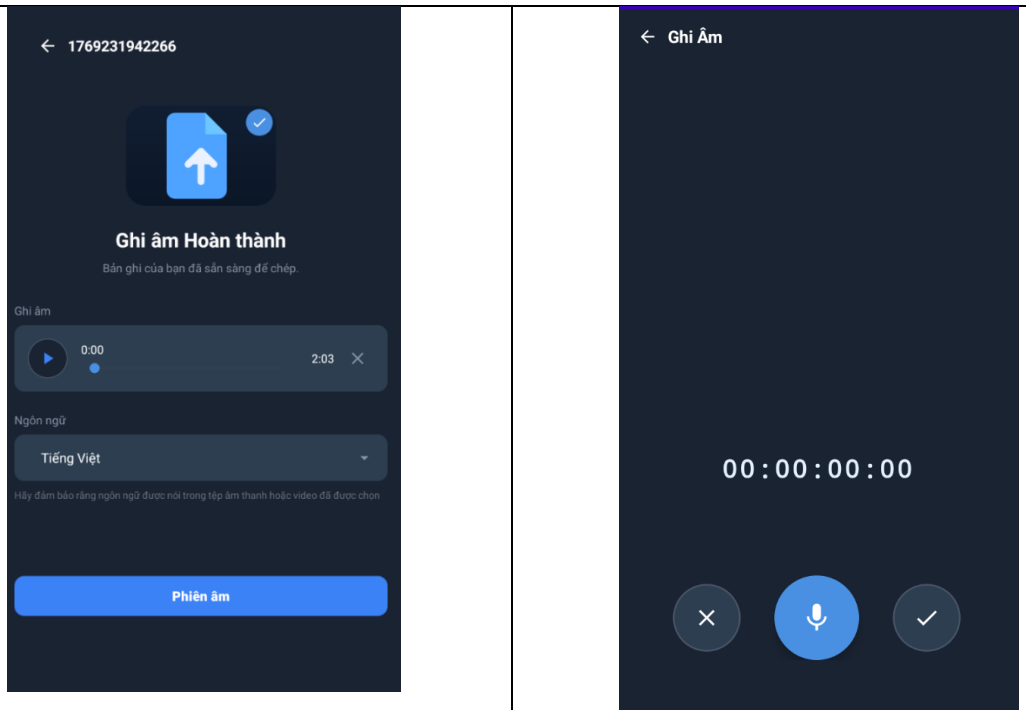
(a) (b)

Figure 5. Application interface: (a) Startup interface, (b) Login interface(in Vietnamese)



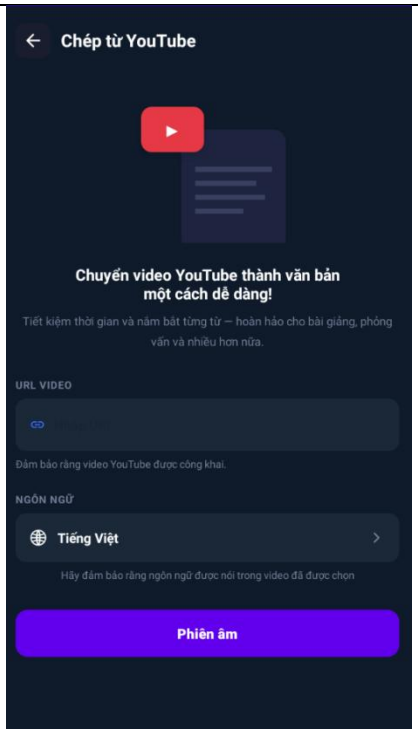
(a) (b)

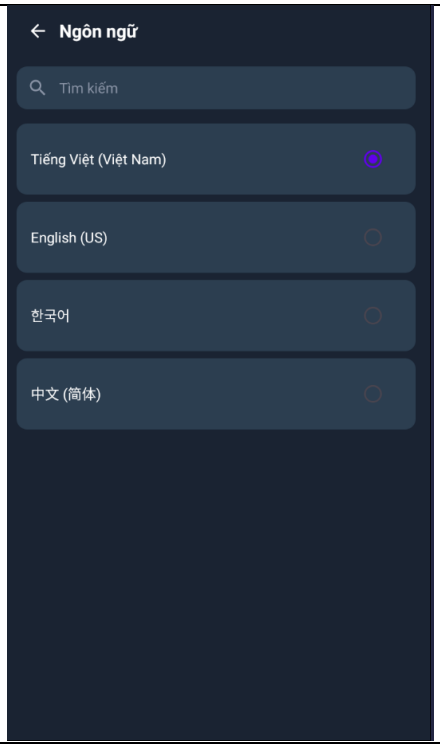
Figure 6.(a) Register interface, (b) Home screen (in Vietnamese)



(a) (b)

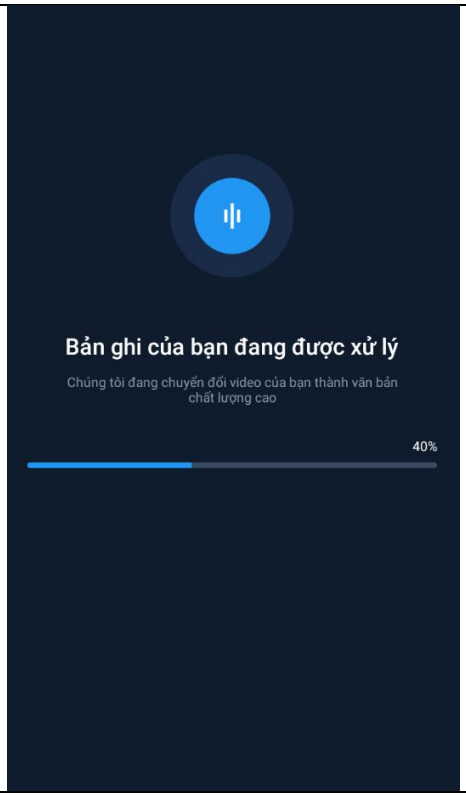
Figure 7.(a) File upload function, (b) Voice recording function (in Vietnamese)

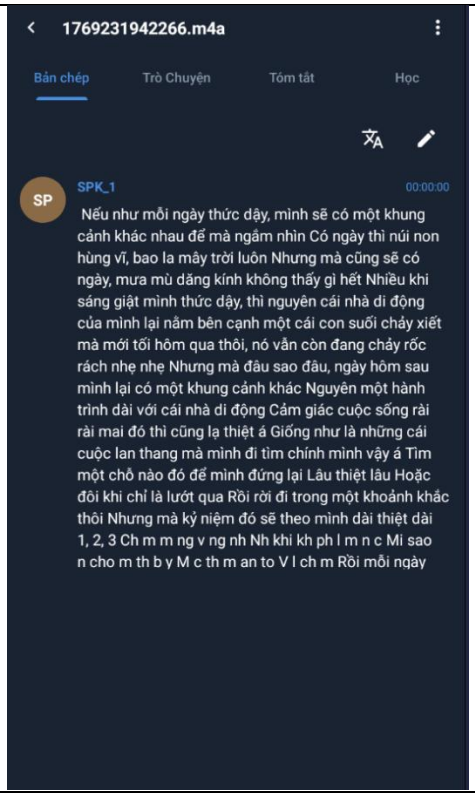




(a) (b)

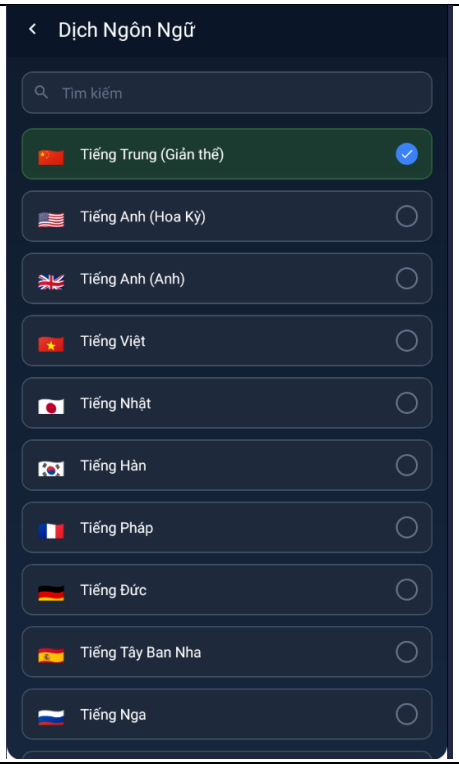
Figure 8.(a) Videos obtained from YouTube, (b) Language selection (in Vietnamese)



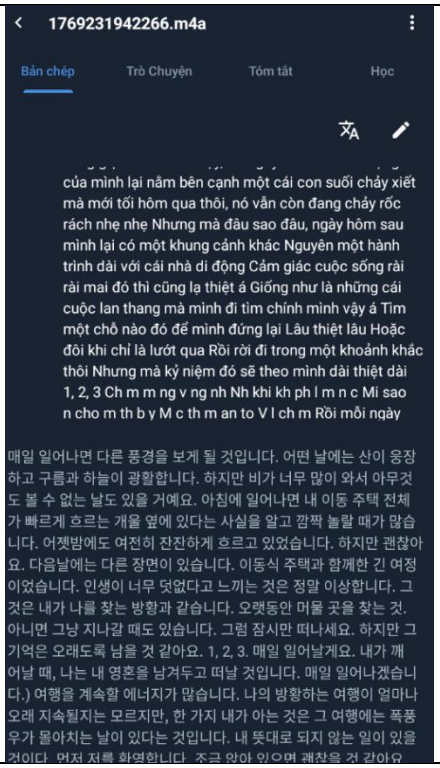


(a) (b)

Figure 9.(a) The record file being processed, (b) Conversion results (in Vietnamese)



(a)



(b)

Figure 10.(a) Translation language selection function, (b) Translation results (in Vietnamese)

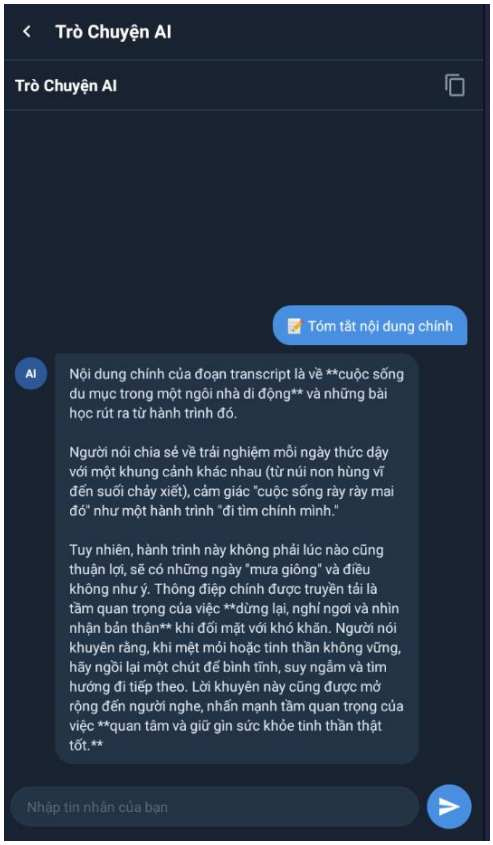


Figure 11.AI chat function(in Vietnamese)

IV. CONCLUSION

In this article, a speech-to-text system is built on the Android platform. The proposed system has several notable advantages, including clearly designed, coherent, and highly optimized database tables. The system design is built with an efficient database structure and data management approach. It supports storing audio and text data after conversion, optimizing data access and processing. The system also provides detailed reports on the accuracy of the conversion process. The system's management is user-friendly, allowing users to perform necessary operations such as file loading, text checking, and adjustments as needed. While significant progress has been made in developing databases for speech-to-text conversion systems, several shortcomings remain. For example, the system design analysis lacks depth, and the program functions are still simplistic. The system's accuracy is heavily dependent on the quality of the input audio; noise, regional accents, or overlapping speech significantly reduce accuracy. Contextual processing capabilities are also limited in long, complex conversations, particularly those involving slang or informal language. Furthermore, the system has limitations in terms of input audio time and file size, making it inconvenient to process long files, which requires significant computing resources and increases costs.

In future work,the system needs to improve accuracy, expand contextual understanding capabilities, increase flexibility in processing time and capacity, optimize resources and reduce costs, enhance security and privacy protection, and support multiple languages and voice-specification.

Acknowledgments

The work presented in this paper is supported by Thai Nguyen University of Technology, Thai Nguyen University, Vietnam.

References

[1] Maryam Asadolahzade Kermanshahi, Ahmad Akbari Azirani, Babak Nasersharif, Seyed Jahanshah Kabudian. Data augmentation techniques for Automatic Speech Recognition: Taxonomy, method analysis, challenges, and future research directions, Computers and Electrical Engineering, 130, 2026, 110851, https://doi.org/10.1016/j.compeleceng.2025.110851.

[2] Sandra Anna Just, Brita Elvevåg, Shrankhla Pandey, Ivan Nenchev, Anna-Lena Bröcker, Christiane Montag, Sarah E Morgan. Moving beyond word error rate to evaluate automatic speech recognition in clinical samples: Lessons from research into schizophrenia-spectrum disorders, Psychiatry Research, 352, 2025, 116690, https://doi.org/10.1016/j.psychres.2025.116690.

[3] Elsayed Issa, Mahmoud Ali, Kevin Hirschi. Measuring linguistic bias in ASR: Whisper large-v3 on non-native speech versus human perception, Procedia Computer Science, 275, 2026, 692-699, https://doi.org/10.1016/j.procs.2026.01.080.

[4] Hassan AlMashhadani, Alaa A. Alsaffar, Hillal Ali AlMaqbali, Ahmed Al-Amayreh. End-to-End Conformer Neural Networks for Multilingual Automatic Speech Recognition and Understanding, Procedia Computer Science, 275, 2026, 541-549, https://doi.org/10.1016/j.procs.2026.01.063.

[5] Ayden M. Cauchi, Jaina Negandhi, Sharon L. Cushing, Karen A. Gordon. Automatic speech recognition technology to evaluate an audiometric word recognition test: A preliminary investigation, Speech Communication, 173, 2025, 103270, https://doi.org/10.1016/j.specom.2025.103270.

[6] Ander González-Docasal, Juan Camilo Vásquez-Correa, Haritz Arzelus, Aitor Álvarez, Santiago A. Moreno-Acevedo. Do modern speech LLMs and re-scoring techniques improve bilingual ASR performance for Basque and Spanish in domain-specific contexts?, Computer Speech & Language, 99, 2026, 101905, https://doi.org/10.1016/j.csl.2025.101905.