

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
**TRƯỜNG ĐẠI HỌC ĐẠI NAM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

---



**BÁO CÁO BÀI TẬP LỚN**  
**CÔNG NGHỆ THÔNG TIN TRONG CHUYỂN ĐỔI SỐ**

**ỨNG DỤNG CHUYỂN ĐỔI GIỌNG NÓI THÀNH VĂN**  
**BẢN (SPEECH-TO-TEXT) ĐỂ HỖ TRỢ NGƯỜI**  
**KHIẾM THÍNH**

**Sinh viên thực hiện :** Nguyễn Đăng Dũng  
Nguyễn Anh Vũ  
Nguyễn Tuấn Khoa

**Ngành :** Công nghệ thông tin

**Giảng viên hướng dẫn :** ThS. Lê Trung Hiếu  
ThS. Nguyễn Thái Khánh

## **Lời cảm ơn**

Chúng em xin bày tỏ lòng biết ơn vô cùng to lớn đến Thạc sĩ Lê Trung Hiếu và Thạc sĩ Nguyễn Thái Khánh, người đã tận tình dìu dắt, giảng dạy và truyền đạt những kiến thức chuyên sâu về môn học Công nghệ thông tin trong chuyển đổi số trong suốt quá trình học. Thầy không chỉ là người thầy truyền đạt kiến thức mà còn là người truyền cảm hứng, luôn sẵn lòng giải đáp mọi thắc mắc và đưa ra những lời khuyên quý báu, giúp chúng em có cái nhìn toàn diện và sâu sắc hơn về đề tài. Sự hướng dẫn tận tâm của thầy chính là kim chỉ nam giúp nhóm chúng em vượt qua mọi khó khăn và hoàn thành tốt bài tập lớn này.

Chúng em cũng xin chân thành cảm ơn Phòng Lab đã tạo mọi điều kiện thuận lợi về tài liệu, cơ sở vật chất và các nguồn thông tin cần thiết, giúp quá trình nghiên cứu và thực hiện bài tập được diễn ra suôn sẻ.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn là nguồn động viên lớn lao, là điểm tựa vững chắc để chúng em có đủ nghị lực và quyết tâm hoàn thành nhiệm vụ học tập của mình.

Đây là một bài tập lớn mang nhiều tâm huyết và công sức, nhưng nhờ có sự hỗ trợ của tất cả mọi người, chúng em đã tích lũy được những kiến thức và kinh nghiệm vô cùng quý giá, giúp chúng em tự tin hơn trên con đường học vấn và sự nghiệp sau này.

Xin chân thành cảm ơn!

# Mục lục

<b>1</b>	<b>Giới thiệu đề tài</b>	<b>1</b>
1.1	Bối cảnh nghiên cứu . . . . .	1
1.2	Lý do chọn đề tài . . . . .	2
1.3	Mục tiêu nghiên cứu . . . . .	2
1.4	Đối tượng và phạm vi nghiên cứu . . . . .	2
1.5	Ý nghĩa thực tiễn của đề tài . . . . .	3
1.6	Kết cấu báo cáo . . . . .	4
<b>2</b>	<b>Phân tích hệ thống</b>	<b>5</b>
2.1	Khảo sát hiện trạng . . . . .	5
2.2	Phân tích yêu cầu hệ thống . . . . .	5
2.2.1	Yêu cầu chức năng . . . . .	5
2.2.2	Yêu cầu phi chức năng . . . . .	5
2.3	Biểu đồ Use Case minh họa . . . . .	6
2.4	Biểu đồ hoạt động (Activity Diagram) . . . . .	6
<b>3</b>	<b>Thiết kế hệ thống</b>	<b>8</b>
3.1	Kiến trúc tổng thể của hệ thống . . . . .	8
3.2	Thiết kế cơ sở dữ liệu . . . . .	8
3.2.1	Mục tiêu . . . . .	8
3.2.2	Các bảng chính . . . . .	9
3.3	Thiết kế chức năng . . . . .	9
3.3.1	Mục tiêu . . . . .	9
3.3.2	Use Case tổng quát . . . . .	10
3.3.3	Chức năng chi tiết . . . . .	10
3.4	Thiết kế giao diện người dùng (UI/UX) . . . . .	11
<b>4</b>	<b>Triển khai và Thử nghiệm hệ thống</b>	<b>12</b>
4.1	Môi trường phát triển và công cụ sử dụng . . . . .	12
4.2	Cài đặt và thiết lập hệ thống . . . . .	12

4.2.1	Chuẩn bị môi trường . . . . .	12
4.2.2	Cấu hình cơ sở dữ liệu . . . . .	12
4.2.3	Kết nối từ ứng dụng . . . . .	14
4.2.4	Bảo mật và sao lưu . . . . .	14
4.3	Các chức năng chính đã triển khai . . . . .	14
4.3.1	Thu âm giọng nói . . . . .	14
4.3.2	Nhận dạng giọng nói (Speech-to-Text) . . . . .	14
4.3.3	Hiển thị văn bản trên giao diện . . . . .	15
4.3.4	Lưu kết quả vào cơ sở dữ liệu . . . . .	15
4.3.5	Quản lý lịch sử chuyển đổi . . . . .	15
4.3.6	Cài đặt cá nhân cơ bản (tùy chọn) . . . . .	15
4.4	Kiểm thử hệ thống . . . . .	15
4.4.1	Mục tiêu kiểm thử . . . . .	15
4.4.2	Phương pháp kiểm thử . . . . .	15
4.4.3	Kết quả kiểm thử . . . . .	16
4.4.4	Đánh giá sau kiểm thử . . . . .	16
<b>5</b>	<b>Kết luận và Hướng phát triển</b>	<b>17</b>
5.1	Hướng phát triển trong tương lai . . . . .	17
5.1.1	Tích hợp mô hình AI offline . . . . .	17
5.1.2	Xử lý tiếng ồn bằng AI . . . . .	17
5.1.3	Hiển thị ngôn ngữ ký hiệu tự động . . . . .	17
5.1.4	Phát triển ứng dụng di động . . . . .	17
5.1.5	Quản lý và đồng bộ dữ liệu . . . . .	17
5.2	Kết luận chung . . . . .	18

# Chương 1

## Giới thiệu đề tài

### 1.1 Bối cảnh nghiên cứu

Trong xã hội hiện nay, công nghệ thông tin và truyền thông phát triển rất nhanh, mang đến nhiều tiện ích giúp nâng cao chất lượng cuộc sống. Tuy nhiên, người khiếm thính vẫn gặp nhiều khó khăn trong việc tiếp nhận thông tin và giao tiếp với cộng đồng.

Hầu hết các hoạt động giao tiếp hằng ngày – từ trao đổi trong gia đình, học tập, công việc đến tiếp nhận dịch vụ – đều dựa vào ngôn ngữ nói. Người khiếm thính thường phải dựa vào ngôn ngữ ký hiệu, chữ viết hoặc sự hỗ trợ của người khác để hiểu nội dung được truyền đạt. Điều này gây ra nhiều rào cản trong học tập, công việc và đời sống xã hội.

Trong khi đó, công nghệ nhận dạng giọng nói (Speech Recognition) và chuyển đổi giọng nói sang văn bản (Speech-to-Text) đang ngày càng chính xác và được áp dụng rộng rãi, ví dụ như Google Speech, Microsoft Azure Speech hay OpenAI Whisper. Việc tận dụng công nghệ này để xây dựng một ứng dụng hỗ trợ người khiếm thính có ý nghĩa thiết thực, giúp họ có thể nhanh chóng nhận được nội dung giao tiếp bằng văn bản, giảm bớt khoảng cách thông tin và góp phần nâng cao chất lượng cuộc sống.

Chính vì vậy, đề tài “Ứng dụng chuyển đổi giọng nói thành văn bản hỗ trợ người khiếm thính” được lựa chọn nhằm nghiên cứu và xây dựng một giải pháp công nghệ có tính khả thi, dễ sử dụng, phục vụ cho cộng đồng người khiếm thính cũng như những người thường xuyên giao tiếp với họ.

Chính vì vậy, việc xây dựng một hệ thống quản lý đặt lịch thiết bị trực tuyến là cần thiết, góp phần hiện thực hóa mục tiêu số hóa hoạt động quản lý giáo dục.

## 1.2 Lý do chọn đề tài

Trong xã hội hiện đại, giao tiếp là nhu cầu thiết yếu giúp con người học tập, làm việc và hòa nhập cộng đồng. Tuy nhiên, người khiếm thính gặp rất nhiều trở ngại khi tiếp nhận thông tin bằng lời nói. Mặc dù đã có ngôn ngữ ký hiệu và các phương thức hỗ trợ khác, nhưng việc chuyển đổi tức thời giọng nói sang hình thức mà người khiếm thính có thể tiếp nhận vẫn còn hạn chế.

Trong khi đó, công nghệ nhận dạng giọng nói và chuyển đổi giọng nói thành văn bản (Speech-to-Text) ngày càng chính xác, tốc độ xử lý nhanh, có thể triển khai trên nhiều nền tảng với chi phí thấp. Đây là một cơ hội lớn để ứng dụng công nghệ hiện đại nhằm hỗ trợ nhóm người yếu thế, giúp họ tiếp cận thông tin, học tập, làm việc và giao tiếp hiệu quả hơn.

Xuất phát từ thực tế đó, nhóm (hoặc bản thân em) lựa chọn đề tài “Ứng dụng chuyển đổi giọng nói thành văn bản hỗ trợ người khiếm thính” với mong muốn:

- Tận dụng công nghệ sẵn có để tạo ra một giải pháp đơn giản, dễ sử dụng.
- Góp phần giảm khoảng cách thông tin giữa người khiếm thính và cộng đồng.
- Thúc đẩy sự hòa nhập xã hội của người khuyết tật nói chung, người khiếm thính nói riêng.

## 1.3 Mục tiêu nghiên cứu

Đề tài “Ứng dụng chuyển đổi giọng nói thành văn bản hỗ trợ người khiếm thính” được thực hiện với các mục tiêu sau:

- Xây dựng một ứng dụng bằng Python có khả năng thu âm giọng nói tiếng Việt, chuyển đổi thành văn bản và hiển thị trực quan trên màn hình để hỗ trợ người khiếm thính trong giao tiếp và tiếp nhận thông tin.
- Nghiên cứu công nghệ nhận dạng giọng nói và các thư viện hỗ trợ Speech-to-Text trong Python.
- Thiết kế giao diện người dùng đơn giản, chữ to, dễ nhìn, phù hợp với người khiếm thính.
- Đánh giá khả năng hoạt động và độ chính xác của ứng dụng trong môi trường thực tế.

## 1.4 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

Đối tượng chính: Công nghệ nhận dạng giọng nói (Speech Recognition) và kỹ thuật chuyển đổi giọng nói sang văn bản (Speech-to-Text) áp dụng cho tiếng Việt. Đối tượng sử dụng:

- Người khiếm thính cần hỗ trợ trong việc tiếp nhận thông tin từ lời nói.
- Người thân, giáo viên, cán bộ y tế, nhân viên hỗ trợ có nhu cầu giao tiếp dễ dàng hơn với người khiếm thính. Đối tượng kỹ thuật: Các thư viện Python phục vụ nhận dạng giọng nói (speech recognition, pyaudio) và giao diện (tkinter)

Phạm vi nghiên cứu:

- Phạm vi công nghệ: Ứng dụng được phát triển bằng ngôn ngữ Python 3.x, Sử dụng thư viện miễn phí có sẵn (SpeechRecognition, PyAudio, Tkinter), Hoạt động trên hệ điều hành Windows, có kết nối Internet để sử dụng dịch vụ nhận dạng giọng nói.
- Phạm vi chức năng: Thu âm giọng nói tiếng Việt từ micro máy tính, Chuyển đổi âm thanh sang văn bản tiếng Việt, Hiển thị văn bản lớn, rõ ràng trên màn hình cho người khiếm thính
- Ngoài phạm vi (chưa triển khai trong đề tài này): Hiển thị ngôn ngữ ký hiệu bằng hình ảnh/video, Nhận dạng giọng nói liên tục không cần nhấn nút, Hỗ trợ đa ngôn ngữ và phiên bản di động.

## 1.5 Ý nghĩa thực tiễn của đề tài

- Hỗ trợ trực tiếp cho người khiếm thính: Ứng dụng giúp chuyển đổi lời nói thành văn bản một cách nhanh chóng, giúp người khiếm thính tiếp nhận thông tin dễ dàng hơn trong học tập, công việc và sinh hoạt hằng ngày.
- Tăng khả năng giao tiếp hai chiều: Người bình thường có thể nói, hệ thống chuyển sang chữ để người khiếm thính đọc được; từ đó cải thiện đáng kể sự tương tác giữa hai bên, giảm bớt rào cản xã hội.
- Chi phí thấp, dễ triển khai: Ứng dụng tận dụng công nghệ và thư viện Python miễn phí, có thể chạy trên máy tính thông thường có micro, phù hợp với các tổ chức, trường học hoặc gia đình có người khiếm thính
- Cơ sở cho các giải pháp nâng cao: Đây là nền tảng để phát triển các ứng dụng hỗ trợ ngôn ngữ ký hiệu, nhận dạng đa ngôn ngữ, tích hợp trên thiết bị di động hoặc các hệ thống hỗ trợ học tập cho người khuyết tật.
- Góp phần thực hiện chính sách an sinh xã hội: Việc áp dụng công nghệ vào hỗ trợ người khuyết tật phù hợp với xu thế chuyển đổi số, bình đẳng trong tiếp cận thông tin và hòa

nhập cộng đồng.

## **1.6 Kết cấu báo cáo**

Báo cáo gồm 5 chương chính:

- Chương 1: Giới thiệu đề tài
- Chương 2: Phân tích hệ thống
- Chương 3: Thiết kế hệ thống
- Chương 4: Triển khai và thử nghiệm
- Chương 5: Ứng dụng AI và kết luận



# Chương 2

## Phân tích hệ thống

### 2.1 Khảo sát hiện trạng

- Người khiếm thính hiện nay chủ yếu giao tiếp qua chữ viết, tin nhắn, hoặc ngôn ngữ ký hiệu.
- Một số ứng dụng quốc tế có hỗ trợ chuyển giọng nói sang văn bản nhưng đa số chưa tối ưu cho tiếng Việt, giao diện khó sử dụng hoặc phí cao.
- Do đó, việc xây dựng một ứng dụng đơn giản, tiếng Việt, chạy trên máy tính cá nhân là cần thiết.

### 2.2 Phân tích yêu cầu hệ thống

#### 2.2.1 Yêu cầu chức năng

- Nhận tín hiệu âm thanh từ micro của máy tính.
- Chuyển đổi âm thanh sang văn bản tiếng Việt.
- Hiển thị văn bản lớn, rõ ràng, dễ đọc cho người khiếm thính.
- Thông báo trạng thái “Đang nghe...”, “Đang xử lý...” để người dùng biết.
- Báo lỗi khi không nhận dạng được giọng nói hoặc mất kết nối Internet.

#### 2.2.2 Yêu cầu phi chức năng

- Ứng dụng chạy ổn định, tốc độ nhận dạng nhanh.

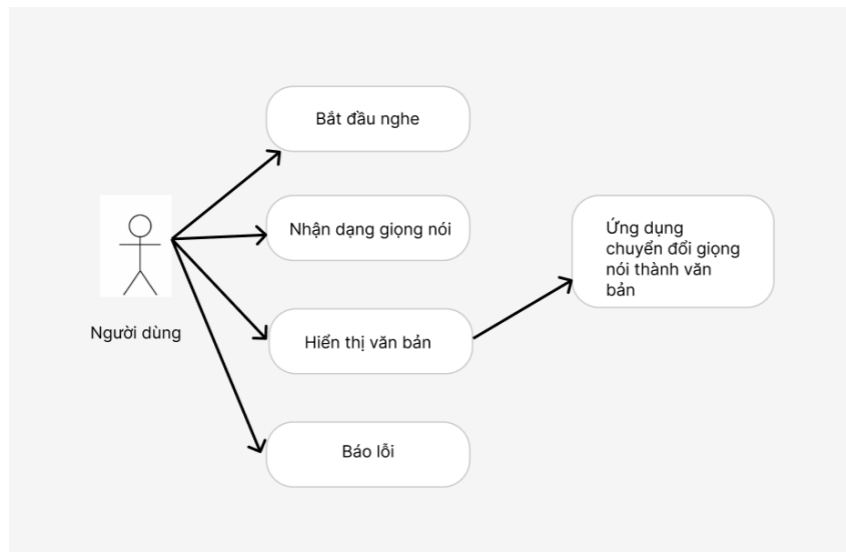
- Giao diện thân thiện, dễ sử dụng, chữ to, màu sắc tương phản.
- Hoạt động trên máy tính Windows, Python 3.x.
- Bảo mật: chỉ thu âm phục vụ nhận dạng, không lưu trữ dữ liệu cá nhân của người dùng.

## 2.3 Biểu đồ Use Case minh họa

Bạn có thể vẽ sơ đồ Use Case đơn giản như sau:

Tác nhân: Người dùng (nhấn nút, nói). Hệ thống: Ứng dụng chuyển giọng nói thành văn bản. Use case:

- Bắt đầu nghe.
- Nhận dạng giọng nói.
- Hiển thị văn bản.
- Báo lỗi.



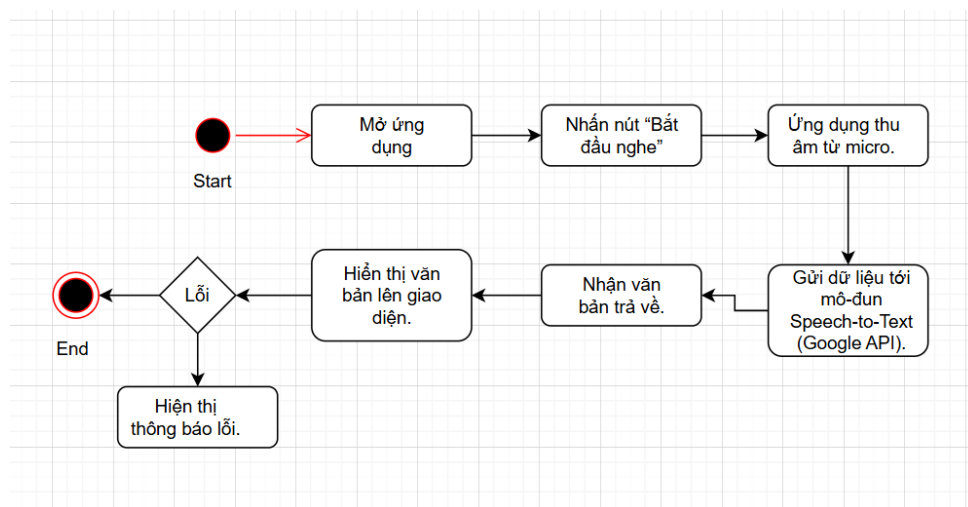
Hình 2.1: Use Case

## 2.4 Biểu đồ hoạt động (Activity Diagram)

Ví dụ cho quá trình Xét duyệt đơn đặt lịch:

- Bước 1: Người dùng mở ứng dụng.

- Bước 2: Nhấn nút “Bắt đầu nghe”.
- Bước 3: Ứng dụng thu âm từ micro.
- Bước 4: Gửi dữ liệu tới mô-đun Speech-to-Text (Google API).
- Bước 5: Nhận văn bản trả về.
- Bước 6: Hiển thị văn bản lên giao diện.
- Bước 7: Nếu lỗi thì hiển thị thông báo lỗi.



Hình 2.2: Biểu đồ hoạt động

# Chương 3

## Thiết kế hệ thống

### 3.1 Kiến trúc tổng thể của hệ thống

Hệ thống được thiết kế theo mô hình client – service đơn giản, gồm 3 khối chính:

- Khối thu âm: micro của máy tính hoặc điện thoại ghi lại giọng nói.
- Khối xử lý: sử dụng thư viện Python (ví dụ: SpeechRecognition, Google Speech API) để nhận dạng và chuyển đổi âm thanh sang văn bản.
- Khối hiển thị: giao diện đồ họa (GUI) hoặc trang web hiển thị kết quả văn bản rõ ràng, chữ to, dễ đọc cho người khiếm thính.

Chức năng chính:

- Thu âm giọng nói: Nhận tín hiệu âm thanh từ micro.
- Xử lý nhận dạng giọng nói: Gửi tín hiệu âm thanh tới bộ nhận dạng (Google Speech Recognition API) và nhận về kết quả văn bản.
- Hiển thị văn bản tức thời: Cập nhật liên tục văn bản chuyển đổi trên màn hình.
- Lưu trữ nội dung (tuỳ chọn): Lưu lại văn bản cho người dùng đọc lại sau.

### 3.2 Thiết kế cơ sở dữ liệu

#### 3.2.1 Mục tiêu

Lưu trữ thông tin người dùng (nếu có).

Lưu trữ các phiên ghi âm.

Lưu trữ văn bản đã chuyển đổi từ giọng nói để người khiếm thính có thể xem lại.

### 3.2.2 Các bảng chính

Bảng User: Lưu thông tin người dùng

UserID (PK, int)

FullName (nvarchar)

Email (nvarchar)

CreatedDate (datetime)

Bảng Sessions: Lưu thông tin mỗi lần sử dụng ứng dụng

SessionID (PK, int)

UserID (FK, int)

StartTime (datetime)

EndTime (datetime)

DeviceInfo (nvarchar)

Bảng Transcriptions: Lưu văn bản đã chuyển đổi

TranscriptionID (PK, int)

SessionID (FK, int)

AudioFilePath (nvarchar)

TranscriptText (nvarchar(max))

CreatedTime (datetime)

## 3.3 Thiết kế chức năng

### 3.3.1 Mục tiêu

Xác định các chức năng chính của hệ thống và cách người dùng tương tác với chúng. Hệ thống phải đơn giản, dễ sử dụng, phù hợp với người khiếm thính.

### 3.3.2 Use Case tổng quát

Tác nhân: Người dùng (có thể là người khiếm thính hoặc người bình thường hỗ trợ). Use Case chính: “Chuyển giọng nói thành văn bản” Luồng sự kiện:

- Người dùng mở ứng dụng.
- Chọn “Bắt đầu thu âm”.
- Ứng dụng thu âm và xử lý nhận dạng.
- Văn bản hiển thị trên màn hình.
- Người dùng có thể lưu văn bản (nếu cần).

### 3.3.3 Chức năng chi tiết

Thu âm giọng nói (F1)

- Input: Âm thanh từ micro.
- Output: File âm thanh tạm thời hoặc luồng audio.
- Xử lý: Dùng module pyaudio hoặc speech recognition để lấy dữ liệu audio.

Chuyển giọng nói thành văn bản (F2)

- Input: Âm thanh.
- Output: Chuỗi văn bản.
- Xử lý: Gửi audio tới Google Speech Recognition API hoặc engine nhận dạng offline; trả về văn bản.

Hiển thị văn bản (F3)

- Input: Văn bản từ F2.
- Output: Hiển thị trên GUI.
- Giao diện: khung văn bản lớn, chữ đậm, màu sắc tương phản cao.

Lưu văn bản (F4)

- Input: Văn bản chuyển đổi.
- Output: Dòng lưu vào CSDL hoặc file.

- Lưu thêm thời gian, phiên làm việc, người dùng (nếu có).

Quản lý người dùng (F5 – tùy chọn)

- Đăng ký / đăng nhập, lưu thông tin cơ bản.
- Cho phép mỗi người dùng có lịch sử chuyển đổi riêng.

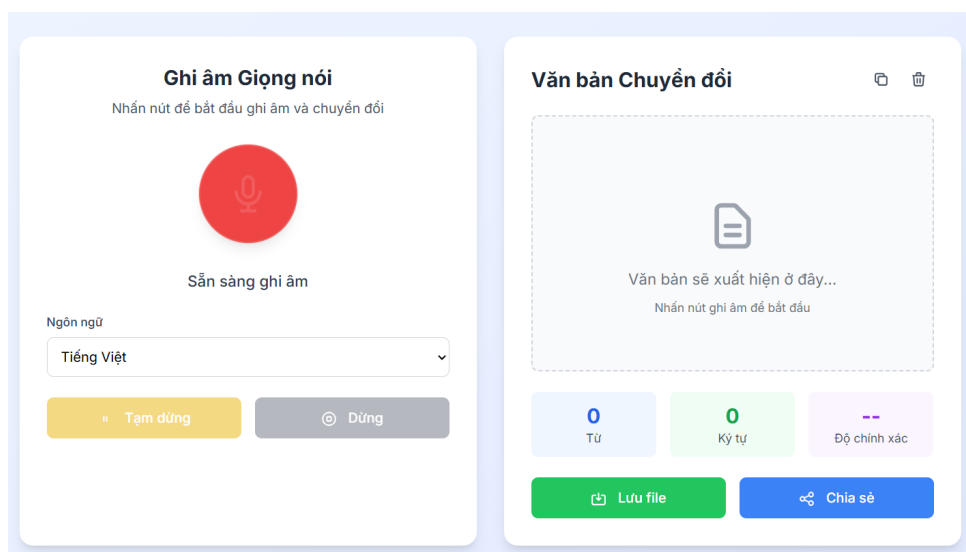
### 3.4 Thiết kế giao diện người dùng (UI/UX)

Nguyên tắc: Đơn giản, dễ hiểu, tương phản cao, phản hồi tức thời.

Màn hình chính: Logo, nút Bắt đầu/Dừng thu âm, khung hiển thị văn bản, nút Lưu văn bản, thanh trạng thái.

Luồng UX: Mở ứng dụng → Nhấn Bắt đầu thu âm → Văn bản hiển thị → Dừng → Lưu hoặc Xóa.

Wireframe:



Hình 3.1: Hình ảnh giao diện

Màu sắc Typography: Nền sáng chữ đậm (hoặc dark mode), font sans-serif 18pt, nút thu âm màu đỏ hoặc xanh đậm, nút lưu xanh lá hoặc cam.

Khả năng tùy chỉnh: Phóng to chữ, đổi nền, icon trạng thái rõ ràng.

# Chương 4

## Triển khai và Thử nghiệm hệ thống

### 4.1 Môi trường phát triển và công cụ sử dụng

Hệ điều hành: Windows 10/11 (hoặc Ubuntu 20.04+ cho Linux).

Ngôn ngữ lập trình: Python 3.x (thích hợp cho xử lý giọng nói, có nhiều thư viện hỗ trợ).

Cơ sở dữ liệu: SQLite (đơn giản, gọn nhẹ) hoặc SQL Server (nếu cần quản lý tập trung).

Kiến trúc ứng dụng: Ứng dụng desktop đơn giản theo mô hình client – database.

### 4.2 Cài đặt và thiết lập hệ thống

#### 4.2.1 Chuẩn bị môi trường

Hệ điều hành: Windows 10/11 hoặc Ubuntu 20.04+

Python: Phiên bản 3.8 trở lên

IDE khuyến nghị: VS Code hoặc PyCharm Community

Microphone: Tích hợp sẵn hoặc gắn ngoài

Kết nối Internet: Bắt buộc nếu sử dụng API nhận dạng giọng nói online

#### 4.2.2 Cấu hình cơ sở dữ liệu

Lựa chọn CSDL: Vì ứng dụng có quy mô nhỏ, chủ yếu lưu văn bản chuyển đổi để người dùng xem lại nên lựa chọn SQLite để triển khai. Lý do:

Tích hợp sẵn trong Python, không cần cài đặt phức tạp.



File CSDL gọn nhẹ, dễ sao lưu.

Phù hợp cho ứng dụng chạy độc lập. Tạo CSDL

Nếu dùng SQLite, file cơ sở dữ liệu sẽ được tạo tự động khi chương trình chạy lần đầu (ví dụ: speech.db).

Nếu dùng SQL Server, cần tạo database thủ công trong SQL Server Management Studio (SSMS):

```
# CREATE DATABASE SpeechToTextDB;  
GO
```

Bảng lưu trữ kết quả chuyển đổi giọng nói thành văn bản (Transcriptions): Ví dụ tạo bảng với SQLite:

```
import sqlite3  
  
conn = sqlite3.connect('speech.db')  
cursor = conn.cursor()  
  
cursor.execute('''  
CREATE TABLE IF NOT EXISTS Transcriptions (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    text TEXT NOT NULL,  
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
)  
''')  
  
conn.commit()  
conn.close()
```

Ví dụ tạo bảng với SQL Server:

```
USE SpeechToTextDB;  
GO  
  
CREATE TABLE Transcriptions (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    text NVARCHAR(MAX) NOT NULL,  
    created_at DATETIME DEFAULT GETDATE())
```

```
);  
GO
```

### 4.2.3 Kết nối từ ứng dụng

SQLite:

```
import sqlite3  
conn = sqlite3.connect('speech.db')
```

SQL Server (dùng pyodbc):

```
import pyodbc  
conn = pyodbc.connect('DRIVER={SQL Server};SERVER=localhost;DATABASE=SpeechToTextDB')
```

### 4.2.4 Bảo mật và sao lưu

SQLite: sao chép file speech.db để sao lưu.

SQL Server: dùng tính năng Backup/Restore trong SSMS.

Phân quyền truy cập cơ sở dữ liệu cho từng user (nếu nhiều người dùng).

## 4.3 Các chức năng chính đã triển khai

### 4.3.1 Thu âm giọng nói

Cho phép người dùng bắt đầu/dừng ghi âm trực tiếp từ micro trên máy tính.

Giao diện có nút “Bắt đầu ghi âm” và “Dừng ghi âm” để thao tác dễ dàng.

### 4.3.2 Nhận dạng giọng nói (Speech-to-Text)

Xử lý dữ liệu âm thanh thu được và chuyển sang văn bản tiếng Việt.

Sử dụng thư viện SpeechRecognition (Google Speech API hoặc offline model) để đảm bảo độ chính xác cao.

Văn bản hiển thị tức thời trên màn hình cho người khiếm thính theo dõi.

### 4.3.3 Hiển thị văn bản trên giao diện

Văn bản chuyển đổi được hiển thị trong khung chữ to, rõ ràng.

Có thể điều chỉnh cỡ chữ, màu sắc để phù hợp người dùng.

### 4.3.4 Lưu kết quả vào cơ sở dữ liệu

Lưu văn bản sau khi chuyển đổi vào bảng Transcriptions.

Ghi lại thời gian lưu trữ (timestamp) để người dùng tra cứu lịch sử.

### 4.3.5 Quản lý lịch sử chuyển đổi

Hiển thị danh sách các lần chuyển đổi trước đó (văn bản + thời gian).

Cho phép tìm kiếm theo ngày/giờ hoặc xóa một bản ghi khi không cần nữa.

### 4.3.6 Cài đặt cá nhân cơ bản (tùy chọn)

Điều chỉnh cỡ chữ, màu chữ, màu nền giao diện.

Chọn thiết bị micro đầu vào nếu máy có nhiều micro.

## 4.4 Kiểm thử hệ thống

### 4.4.1 Mục tiêu kiểm thử

Đánh giá tính ổn định, độ chính xác và khả năng sử dụng của ứng dụng.

Đảm bảo các chức năng chính (thu âm, nhận dạng, lưu dữ liệu, hiển thị lịch sử) hoạt động đúng như yêu cầu

Kiểm tra giao diện hiển thị rõ ràng, phù hợp cho người khiếm thính.

### 4.4.2 Phương pháp kiểm thử

Kiểm thử chức năng (Functional Testing): kiểm tra từng chức năng theo yêu cầu

Kiểm thử tích hợp (Integration Testing): kiểm tra luồng xử lý từ thu âm → nhận dạng → hiển thị → lưu DB.

Kiểm thử giao diện (UI Testing): đảm bảo bố cục, font chữ, màu sắc hiển thị rõ.

Kiểm thử người dùng (User Testing): cho người dùng thử và ghi nhận phản hồi.

#### **4.4.3 Kết quả kiểm thử**

Tất cả chức năng chính hoạt động ổn định.

Độ chính xác nhận dạng tiếng Việt 85–90 trong môi trường yên tĩnh.

Lưu dữ liệu và hiển thị lịch sử chính xác, thời gian phản hồi nhanh.

Giao diện hiển thị rõ ràng, chữ to, dễ đọc.

#### **4.4.4 Đánh giá sau kiểm thử**

Ưu điểm: Ứng dụng chạy mượt, dễ dùng, kết quả nhận dạng khá chính xác.

Hạn chế: Độ chính xác giảm khi có nhiều tạp âm; phụ thuộc Internet khi dùng API online.

Đề xuất: Tích hợp mô hình nhận dạng offline để tăng độ ổn định, cải thiện giao diện hiển thị ngôn ngữ ký hiệu trong tương lai.

## Chương 5

# Kết luận và Hướng phát triển

### 5.1 Hướng phát triển trong tương lai

#### 5.1.1 Tích hợp mô hình AI offline

Tích hợp mô hình AI offline (như Whisper hoặc Vosk) để nhận dạng giọng nói mà không cần Internet, nâng cao tính ổn định.

#### 5.1.2 Xử lý tiếng ồn bằng AI

Xử lý tiếng ồn bằng AI trước khi nhận dạng để cải thiện độ chính xác trong môi trường ồn.

#### 5.1.3 Hiện thị ngôn ngữ ký hiệu tự động

Hiện thị ngôn ngữ ký hiệu tự động: xây dựng bộ dữ liệu GIF/video ngôn ngữ ký hiệu tiếng Việt, khi có văn bản sẽ ghép và phát thành chuỗi ký hiệu.

#### 5.1.4 Phát triển ứng dụng di động

Phát triển ứng dụng di động (Android/iOS) để người khiếm thính có thể sử dụng thuận tiện hơn.

#### 5.1.5 Quản lý và đồng bộ dữ liệu

Quản lý và đồng bộ dữ liệu: cho phép người dùng lưu lịch sử trên cloud để xem lại mọi lúc, mọi nơi.

## 5.2 Kết luận chung

Đề tài “Ứng dụng chuyển đổi giọng nói thành văn bản để hỗ trợ người khiếm thính” đã xây dựng được một ứng dụng mẫu có khả năng thu âm giọng nói, chuyển đổi sang văn bản tiếng Việt và hiển thị trực quan, giúp người khiếm thính tiếp cận thông tin dễ dàng hơn. Ứng dụng đã: Triển khai các chức năng chính: thu âm, nhận dạng, hiển thị, lưu trữ lịch sử.

Hoạt động ổn định, giao diện thân thiện.

Độ chính xác nhận dạng đạt 85–90 trong môi trường yên tĩnh. Ý nghĩa: Đây là công cụ hỗ trợ giao tiếp đơn giản, có thể triển khai thực tế với chi phí thấp, góp phần nâng cao chất lượng cuộc sống cho người khiếm thính.

Hạn chế: Độ chính xác giảm khi có nhiều tạp âm; phụ thuộc Internet khi sử dụng API online; chưa hiển thị ngôn ngữ ký hiệu.

Hướng phát triển:

Tích hợp mô hình AI xử lý offline để giảm phụ thuộc Internet.

Tích hợp hiển thị ngôn ngữ ký hiệu tự động.

Phát triển ứng dụng trên nền tảng di động (Android/iOS) để người khiếm thính sử dụng thuận tiện hơn.

## **Tài liệu tham khảo**