

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH
KHOA ĐIỆN-ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



HCMUTE

BÁO CÁO CUỐI KỲ

**XÂY DỰNG ỨNG DỤNG TRỢ LÝ GIỌNG
NÓI THÔNG MINH SỬ DỤNG FLUTTER
VÀ OPENAI API**

Môn học: Đồ án 2

GVHD: TS. HUỖNH THẾ THIÊN

SVTH:

Nguyễn Vũ Tú

20139094

TP. HỒ CHÍ MINH, THÁNG 9 - 2024

LỜI CẢM ƠN

Trong quá trình hoàn thành đồ án môn học 2 với đề tài “Xây dựng ứng dụng trợ lý giọng nói thông minh sử dụng Flutter và OpenAI API”, tôi đã nhận được rất nhiều sự tư vấn và lời khuyên từ Thầy Cô và bạn bè ngành Hệ thống nhúng và IoT, khoa Điện - Điện tử, thuộc trường Đại Học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh thì mới có thể hoàn thành đề tài.

Tôi xin gửi một lời cảm ơn chân thành nhất đến những quý Thầy Cô, bạn bè đã góp ý, giúp đỡ trong quá trình thực hiện đề tài. Đặc biệt, tôi xin cảm ơn thầy Huỳnh Thế Thiện , người đã luôn đồng hành cùng tôi trong quá trình thực hiện đề tài. Những góp ý, lời khuyên, đánh giá và lộ trình thầy đặt ra từ những ngày đầu làm đồ án thực sự để lại ý nghĩa rất lớn, ảnh hưởng đến tư duy, hướng phát triển của tôi.

Tất nhiên, đề tài vẫn còn những thiếu sót và hạn chế. Tôi rất mong nhận được thêm những góp ý để học hỏi, hoàn thiện hơn đề tài trong tương lai.

Tôi xin chân thành cảm ơn!

LỜI CAM ĐOAN

Cá nhân tôi là người thực hiện đồ án 2 cam đoan đề tài thực hiện dựa vào một số tài liệu trước đó và không sao chép nội dung, kết quả của đồ án khác. Các nội dung tham khảo đã được trích dẫn đầy đủ

Người thực hiện đồ án 2
(Ký và ghi rõ họ tên)

TÓM TẮT NỘI DUNG

Đồ án nhằm xây dựng một ứng dụng trợ lý ảo thông minh trên nền tảng di động Flutter có khả năng nhận dạng và xử lý giọng nói. Hệ thống được thiết kế dựa trên kiến trúc Bot, cho phép người dùng tương tác bằng giọng nói tự nhiên để đặt các câu hỏi, yêu cầu và nhận phản hồi thông minh từ trợ lý ảo AI.

Điểm khác biệt là tích hợp API OpenAI (GPT-3 và DALL-E) cho phép xử lý ngôn ngữ tự nhiên, đưa ra câu trả lời thông minh cũng như tự động tổng hợp hình ảnh dựa trên yêu cầu của người dùng. Giao diện thân thiện, hỗ trợ người dùng tối đa trong việc sử dụng các tính năng của ứng dụng.

Đây là một ứng dụng tiện ích, có thể mở rộng nhiều tính năng trong tương lai nhằm phục vụ cuộc sống và công việc của người dùng một cách tốt nhất. Đồ án được phát triển dựa trên các công nghệ mới nhất về Flutter, AI và IoT.

Mục lục

LỜI CẢM ƠN	i
LỜI CAM ĐOAN	ii
TÓM TẮT NỘI DUNG	iii
Mục lục	iv
Danh sách hình vẽ	vi
1 TỔNG QUAN	1
1.1 Giới thiệu chung	1
1.1.1 Ý tưởng và mục tiêu dự án	1
1.1.2 Phạm vi và đối tượng người dùng	2
1.1.3 Tình hình nghiên cứu	3
1.2 Phương pháp và Công nghệ sử dụng	3
1.2.1 Flutter ^[7] ^[6]	3
1.2.2 OpenAI API ^[2] ^[8]	4
1.2.3 Phương pháp Scrum ^[1] ^[5]	4
2 PHÂN TÍCH VÀ THIẾT KẾ	6
2.1 Kiến trúc tổng quan	6
2.2 Các chức năng chính	8
2.2.1 Xử lý nhận dạng giọng nói	8
2.2.2 Kết nối chatbot AI (OpenAI API)	9
2.2.3 Xử lý đầu ra bằng văn bản và giọng nói	9
2.3 Luồng dữ liệu	10
2.4 Thiết kế giao diện đơn giản, trực quan ^[3]	11
3 TRIỂN KHAI ỨNG DỤNG	14

3.1	Hệ thống thư mục dự án	14
3.1.1	Thư Mục lib	14
3.1.2	Thư Mục assets	15
3.1.3	File Cấu Hình (.env)	15
3.1.4	Thư Mục android, ios	16
3.2	Các file mã nguồn chính ^[4]	16
3.2.1	main.dart	16
3.2.2	home_page.dart	17
3.2.3	openai_service.dart	17
3.2.4	speech_to_text.dart và flutter_tts.dart	17
3.2.5	menu_drawer.dart	18
4	KẾT QUẢ VÀ ĐÁNH GIÁ	19
4.1	Kết quả chạy thử và đánh giá hiệu năng	19
4.2	So sánh với các ứng dụng khác	21
5	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	23
5.1	Kết luận	23
5.2	Hướng phát triển	23
	Tài liệu tham khảo	25

Danh sách hình vẽ

2.1.1	Ảnh mô hình tổng quát của ứng dụng	6
2.4.1	Ảnh giao diện chính của ứng dụng	12
2.4.2	Ảnh menu tùy chọn của ứng dụng	13
4.1.1	Ảnh kết quả khi hỏi "What is a computer?"	20
4.1.2	Ảnh kết quả khi hỏi "Hình ảnh con mèo?"	21

Chương 1

TỔNG QUAN

1.1 Giới thiệu chung

1.1.1 Ý tưởng và mục tiêu dự án

Ý Tưởng Dự Án

Ứng dụng Voice Bot là một bước đột phá trong lĩnh vực trợ lý ảo, nơi người dùng có thể tương tác tự nhiên và linh hoạt. Sự độc đáo của ứng dụng không chỉ nằm ở việc chuyển đổi giọng nói thành văn bản một cách chính xác, mà còn ở khả năng sáng tạo và phản hồi của trí tuệ nhân tạo.

Ứng dụng sử dụng công nghệ nhận dạng giọng nói tiên tiến để đảm bảo rằng mọi lời nói của người dùng được hiểu đúng và chính xác. Điều này mở ra cánh cửa cho việc tương tác tự nhiên và thoải mái, không còn giới hạn bởi việc nhập liệu truyền thống.

Quan trọng nhất, ứng dụng tích hợp OpenAI API để đưa vào "trí óc" trí tuệ nhân tạo mạnh mẽ. Điều này cho phép ứng dụng không chỉ đơn giản là chuyển đổi giọng nói thành văn bản, mà còn hiểu nghĩa đằng sau câu hỏi và có khả năng phản hồi một cách thông minh.

Mục Tiêu Dự Án

- **Tương Tác Tự Nhiên và Thân Thiện:**

Tạo ra một giao diện cho phép người dùng tương tác với Voice Bot một cách tự nhiên và thân thiện như gặp một người bạn.

- **Chuyển Đổi Giọng Nói Chính Xác:**

Sử dụng công nghệ nhận dạng giọng nói để chuyển đổi âm thanh thành văn bản chính xác và hiệu quả.

- **Phân Tích Nghĩa và Phản Hồi Linh Hoạt:**

Áp dụng OpenAI API để trích xuất nghĩa từ câu hỏi và phản hồi một cách linh hoạt, có thể hiển thị dưới nhiều dạng, từ văn bản đến hình ảnh.

- **Đọc To Nội Dung Phản Hồi:**

Cung cấp khả năng đọc to nội dung phản hồi, mang lại trải nghiệm đa giác cho người dùng.

- **Giao Diện Hiện Đại và Thẩm Mỹ:**

Xây dựng giao diện sáng tạo, hiện đại, và thẩm mỹ để thu hút người dùng và làm tăng trải nghiệm sử dụng.

- **Đa Dạng Hóa Trí Tuệ Nhân Tạo:**

Tích hợp nhiều khía cạnh của trí tuệ nhân tạo để ứng dụng có khả năng hiểu biết sâu sắc và đáp ứng một cách linh hoạt.

Nhìn chung, mục tiêu của ứng dụng Voice Bot không chỉ là chuyển đổi giọng nói thành văn bản, mà còn là mở rộng khả năng tương tác và giao tiếp giữa người dùng và máy tính, tạo ra một trợ lý ảo thông minh và gần gũi.

1.1.2 Phạm vi và đối tượng người dùng

Phạm vi:

Về phạm vi ứng dụng, hiện tại hệ thống tập trung vào xây dựng trợ lý ảo có khả năng hiểu và trả lời các câu hỏi, thắc mắc thông thường của người dùng bằng tiếng Việt. Các nhóm tính năng chính bao gồm:

- Trả lời các câu hỏi về kiến thức chung
- Tìm kiếm thông tin đơn giản trên Internet
- Mở ứng dụng, thực hiện một số tác vụ cơ bản trên điện thoại
- Chia sẻ hình ảnh minh họa khi được yêu cầu

Đối tượng người dùng:

Đối tượng sử dụng của ứng dụng Voice Bot rất đa dạng, bao gồm những người muốn tiết kiệm thời gian, đặc biệt là những người có cuộc sống bận rộn. Người dùng không sử dụng hiệu quả bàn phím hoặc màn hình cũng có trong đối tượng, với mục tiêu cung cấp lựa chọn tương

tác phù hợp. Ngoài ra, ứng dụng hướng đến những người muốn trải nghiệm tương tác giọng nói tự nhiên và đa ngôn ngữ. Đối tượng này cũng bao gồm những người quan tâm đến công nghệ trí tuệ nhân tạo, tìm kiếm một giao diện sáng tạo, hiện đại, và mạnh mẽ trong việc đáp ứng nhu cầu hàng ngày của họ. Tóm lại, Voice Bot hướng đến việc phục vụ một cộng đồng người dùng rộng lớn, đặc biệt là những người có nhu cầu sử dụng trợ lý ảo thông qua giọng nói.

1.1.3 Tình hình nghiên cứu

Dự án Voice GPT đã thể hiện được tiềm năng nghiên cứu và ứng dụng công nghệ AI vào việc xây dựng trợ lý ảo thông minh. Cụ thể, dự án đã áp dụng các mô hình GPT-3.5 và DALL-E - những thành tựu mới nhất của OpenAI, kết hợp với khả năng nhận diện giọng nói và tổng hợp giọng nói nhân tạo để mang lại trải nghiệm tương tác tự nhiên cho người dùng. Bên cạnh đó, kiến trúc phần mềm theo mô hình Clean Architecture giúp tách biệt các mối quan tâm, tạo điều kiện thuận lợi cho việc mở rộng và bảo trì hệ thống.

Tuy vậy, vẫn còn một số hạn chế cần khắc phục và nhiều hướng phát triển tiềm năng như xây dựng tính năng lưu trữ và phân tích dữ liệu người dùng, tối ưu trải nghiệm bằng cách áp dụng kỹ thuật đệ quy thích ứng và học máy. Nếu khai thác và hoàn thiện tốt, Voice GPT hoàn toàn có thể trở thành một sản phẩm trợ lý ảo cá nhân thông minh và đắc lực cho mọi người dùng.

1.2 Phương pháp và Công nghệ sử dụng

1.2.1 Flutter^[7]^[6]

Flutter, một framework phát triển ứng dụng di động đa nền tảng do Google phát triển, đã thu hút sự chú ý lớn từ cộng đồng phát triển và doanh nghiệp. Được xây dựng với ngôn ngữ lập trình Dart, Flutter mang lại khả năng phát triển ứng dụng cho cả Android và iOS từ một mã nguồn duy nhất, giảm bớt độ phức tạp của việc duy trì hai mã nguồn riêng biệt.

Một trong những điểm mạnh nổi bật của Flutter là khái niệm "widget." Widget là các thành phần nhỏ có thể kết hợp để xây dựng giao diện người dùng. Sự linh hoạt của widget giúp nhà phát triển tạo ra giao diện đẹp mắt và đồng nhất trên cả hai hệ điều hành.

Chức năng "hot reload" của Flutter là một công cụ quan trọng cho nhà phát triển. Nó cho phép xem kết quả ngay lập tức khi có sự thay đổi trong mã nguồn, giúp tiết kiệm thời gian kiểm thử và tối ưu quá trình phát triển.

Với hiệu suất cao và khả năng tương thích tốt, Flutter là lựa chọn phổ biến trong cộng đồng phát triển ứng dụng di động. Không chỉ hỗ trợ cho các ứng dụng di động, Flutter còn có khả năng phát triển ứng dụng web và desktop, tạo ra một cơ sở chung cho việc xây dựng các sản

phẩm đa dạng và hiệu quả. Cộng đồng lớn và tích cực của Flutter đồng hành với các nhà phát triển, cung cấp tài nguyên, hỗ trợ và sự đổi mới không ngừng

1.2.2 OpenAI API^{[2][8]}

OPENAI API là một dịch vụ cung cấp bởi OpenAI, một tổ chức nghiên cứu trí tuệ nhân tạo nổi tiếng. API này mang lại khả năng sử dụng mô hình ngôn ngữ tự nhiên mạnh mẽ của OpenAI, chủ yếu là GPT-3 (Generative Pre-trained Transformer 3), thông qua một giao diện lập trình ứng dụng đơn giản.

GPT-3 là một mô hình học máy có khả năng xử lý ngôn ngữ tự nhiên ở mức độ rất cao. Nó được đào tạo trên một lượng lớn dữ liệu và có khả năng tạo ra văn bản tự nhiên, phản hồi thông minh và thậm chí thực hiện các nhiệm vụ phức tạp như viết mã, tạo nghệ thuật, và nhiều hơn nữa.

OPENAI API cho phép nhà phát triển tích hợp sức mạnh của GPT-3 vào các ứng dụng, trang web, hoặc dự án của họ. Thông qua API, người dùng có thể gửi các yêu cầu về xử lý ngôn ngữ tự nhiên và nhận phản hồi từ mô hình GPT-3. Điều này mở ra một loạt các ứng dụng, từ việc tạo nội dung động, tạo ra câu chuyện, đến giải quyết vấn đề và trả lời câu hỏi.

Sự linh hoạt và khả năng tương tác của OPENAI API đã làm cho nó trở thành một công cụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo, hỗ trợ sáng tạo và phát triển ứng dụng thông minh.

1.2.3 Phương pháp Scrum^{[1][5]}

Scrum là một trong những phương pháp quản lý và phát triển dự án phần mềm phổ biến nhất hiện nay, được chúng tôi áp dụng trong quá trình xây dựng ứng dụng voice bot. Phương pháp này giúp chúng tôi tổ chức công việc một cách hiệu quả, tối ưu hóa sự linh hoạt và thích ứng với những thay đổi trong quá trình phát triển.

Đầu tiên, chúng tôi xác định các user story (câu chuyện người dùng) để hiểu rõ yêu cầu của người dùng. Điều này giúp chúng tôi tập trung vào việc xây dựng những tính năng và chức năng quan trọng nhất cho ứng dụng voice bot.

Tiếp theo, chúng tôi lập kế hoạch triển khai dự án thành các sprint, mỗi sprint kéo dài khoảng 2-3 tuần. Mỗi sprint được đặt ra một mục tiêu cụ thể về các tính năng hoặc chức năng phải hoàn thành trong khoảng thời gian đó. Điều này giúp chúng tôi tập trung và đảm bảo rằng mỗi sprint mang lại giá trị gia tăng cho sản phẩm.

Cuối mỗi sprint, nhóm chúng tôi tiến hành kiểm thử, đánh giá và thu thập phản hồi từ người sử dụng cũng như các thành viên khác trong nhóm. Quá trình này giúp chúng tôi kiểm soát chất lượng sản phẩm và nhanh chóng điều chỉnh bất kỳ vấn đề nào xuất hiện.

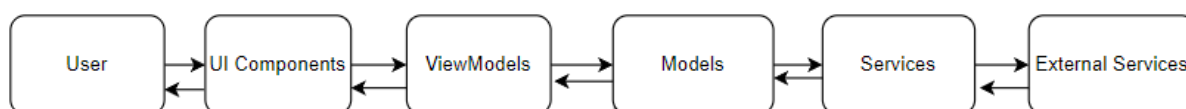
1.2. Phương pháp và Công nghệ sử dụng

Ưu điểm chính của Scrum mà chúng tôi trải nghiệm là sự linh hoạt. Phương pháp này cho phép chúng tôi thích ứng nhanh chóng với những thay đổi, yêu cầu mới hoặc ưu tiên khác nhau từ phía người dùng. Điều này giúp chúng tôi duy trì tốc độ phát triển ổn định và đảm bảo rằng ứng dụng voice bot của chúng tôi luôn đáp ứng được những đòi hỏi ngày càng cao từ cộng đồng người dùng.

Chương 2

PHÂN TÍCH VÀ THIẾT KẾ

2.1 Kiến trúc tổng quan



Hình 2.1.1: Ảnh mô hình tổng quát của ứng dụng

- **User:**

Người Dùng trong ứng dụng voice bot đóng vai trò quan trọng, tương tác thông qua giọng nói để gửi lệnh, đặt câu hỏi, và lắng nghe phản hồi. Trải nghiệm của họ là trọng tâm, đòi hỏi sự hiểu biết về cách họ tương tác và mong đợi từ ứng dụng. Các thách thức bao gồm xử lý giọng nói và cung cấp trả lời tự nhiên. Ghi nhận phản hồi từ người dùng là quan trọng để liên tục cải thiện trải nghiệm và hiểu rõ hơn về nhu cầu của họ.

- **UI Components:**

Là trụ cột quan trọng trong việc cung cấp trải nghiệm tương tác cho người dùng trong ứng dụng voice bot. Thiết kế của chúng cần phải tương thích với tương tác bằng giọng nói, đồng thời trực quan, thân thiện, và linh hoạt. Chúng không chỉ đóng vai trò trong việc tạo ra môi trường tương tác, mà còn phải đáp ứng đa dạng nhu cầu của người dùng. Sự thích ứng và khả năng hiển thị phản hồi người dùng là quan trọng để tạo ra trải nghiệm mượt mà và hiệu quả. Điều này đặt ra thách thức về việc tối ưu hiệu năng và tốc độ, đồng thời đảm bảo tích hợp chặt chẽ với khả năng tương tác bằng giọng nói của ứng dụng voice bot.

- **ViewModels:**

Là lớp trung tâm trong kiến trúc của ứng dụng voice bot, chịu trách nhiệm quản lý logic nghiệp và trạng thái của giao diện người dùng. Điều này giúp tách biệt logic nghiệp từ thành phần giao diện, tạo nên một kiến trúc dễ bảo trì và mở rộng.

Một trong những ưu điểm quan trọng của ViewModels là khả năng lắng nghe và xử lý sự kiện từ người dùng. Điều này bao gồm khả năng xử lý lệnh bằng giọng nói, thực hiện các hành động tương ứng, và duy trì trạng thái liên tục với dữ liệu cần thiết từ Models.

ViewModels cũng kết nối với Models để truy xuất và cập nhật dữ liệu. Điều này tạo ra một liên kết mạnh mẽ giữa logic nghiệp và dữ liệu, giúp duy trì tính toàn vẹn của dữ liệu và đồng bộ nó với trạng thái của giao diện người dùng.

Sự tách biệt giữa ViewModels và thành phần giao diện không chỉ làm cho mã nguồn trở nên dễ đọc và bảo trì, mà còn tăng khả năng kiểm thử của ứng dụng. Các đơn vị kiểm thử có thể tập trung vào việc kiểm thử logic nghiệp mà không phải quan tâm đến cách giao diện người dùng được hiển thị.

- **Models:**

Models là yếu tố chủ chốt trong cấu trúc của ứng dụng voice bot, nơi chúng đảm nhận trách nhiệm quan trọng về việc quản lý và bảo quản dữ liệu. Trong hệ thống này, Message Model nổi bật, đóng vai trò quản lý các tin nhắn trong ứng dụng. Message Model không chỉ lưu trữ thông tin mà còn biểu diễn dữ liệu đó với các thuộc tính như nội dung tin nhắn, người gửi, và thời gian gửi.

Sự tích hợp chặt chẽ giữa Models và ViewModels là chìa khóa để cung cấp dữ liệu một cách linh hoạt và hiệu quả cho giao diện người dùng. Models cung cấp nguồn dữ liệu cho ViewModels, giúp duy trì sự đồng bộ giữa dữ liệu và giao diện người dùng. Điều này không chỉ làm cho ứng dụng linh hoạt mà còn giúp đảm bảo rằng người dùng luôn nhận được thông tin mới nhất và chính xác.

Mô hình cũng tương tác với các dịch vụ khác như cơ sở dữ liệu, làm cho chúng trung tâm của quá trình lấy và cập nhật dữ liệu. Sự độc lập của Models với thành phần giao diện người dùng mang lại lợi ích về quản lý và bảo trì, giúp duy trì mã nguồn dễ đọc và mở rộng.

- **Services:**

Dịch Vụ (Services) đóng vai trò quan trọng trong kiến trúc của ứng dụng voice bot, chịu trách nhiệm thực hiện các chức năng cụ thể để đáp ứng nhu cầu của người dùng. Trong ngữ cảnh cụ thể này, có hai dịch vụ quan trọng là SpeechService và OpenAIService, mỗi dịch vụ thực hiện một loạt các chức năng đặc biệt.

SpeechService là nơi thực hiện các chức năng liên quan đến giọng nói. Nó có thể thực hiện việc ghi âm âm thanh từ người dùng, chuyển đổi giọng thành văn bản, và có thể thực hiện các tác vụ khác liên quan đến xử lý giọng nói. Điều này làm cho SpeechService trở thành một phần quan trọng của việc xây dựng trải nghiệm tương tác thông qua giọng nói trong ứng dụng voice bot.

OpenAIService, trong khi đó, tập trung vào việc giao tiếp với các dịch vụ ngoại vi như OpenAI. Nó đảm nhận trách nhiệm gửi yêu cầu đến API của OpenAI để thực hiện các nhiệm vụ như gửi văn bản và nhận kết quả từ các mô hình như GPT-3 hoặc DALL-E. OpenAIService đóng vai trò quan trọng trong việc kết nối ứng dụng với công nghệ AI mạnh mẽ từ OpenAI.

- **External Services:**

Dịch vụ bên ngoài (External Services) trong kiến trúc của ứng dụng voice bot đóng vai trò quan trọng, đại diện cho các nguồn dữ liệu và chức năng từ bên ngoài ứng dụng. Trong hệ thống này, External chủ yếu tập trung vào kết nối và tương tác với các dịch vụ và API ngoại vi, đặc biệt là như OpenAI SDK và các API khác, để mở rộng khả năng và chức năng của ứng dụng.

Một trong những chức năng quan trọng của External Services là khả năng gửi yêu cầu đến các API và dịch vụ bên ngoài, sau đó nhận và xử lý kết quả trả về. Điều này cho phép ứng dụng voice bot sử dụng các công nghệ mạnh mẽ như GPT-3 và DALL-E từ OpenAI để xử lý giọng nói và tạo ra nội dung đáp ứng người dùng.

External Services cũng có vai trò quan trọng trong việc định kỳ cập nhật dữ liệu từ các nguồn ngoại vi, đảm bảo rằng ứng dụng luôn sử dụng thông tin mới nhất. Đồng thời, sự tích hợp chặt chẽ giữa External Services và các thành phần khác của ứng dụng đảm bảo rằng mọi chức năng hoạt động một cách hài hòa.

2.2 Các chức năng chính

2.2.1 Xử lý nhận dạng giọng nói

Chức năng nhận dạng giọng nói được xây dựng dựa trên package `speech_to_text`, cho phép người dùng tương tác với ứng dụng thông qua giọng nói tự nhiên.

Quy trình xử lý bao gồm các bước chính sau: Ứng dụng sẽ đầu tiên khởi tạo micro và cấp quyền truy cập cũng như tải các model ngôn ngữ cần thiết từ `speech_to_text` API. Khi người dùng nhấn nút ghi âm, micro sẽ được kích hoạt để thu âm và gửi luồng dữ liệu âm thanh này lên server. `Speech_to_text` sẽ sử dụng các thuật toán deep learning và machine learning để phân

tích các đặc trưng âm vị từ luồng âm thanh, dựa trên các ngữ cảnh đã được huấn luyện, để nhận diện các từ và cụm từ tương ứng. Kết quả được trả về dưới dạng văn bản và hiển thị lên màn hình cho người dùng.

Toàn bộ quá trình diễn ra nhanh chóng chỉ trong vòng vài giây và hoàn toàn tự động, mang đến trải nghiệm tự nhiên như đang trò chuyện với con người. Độ chính xác của kết quả nhận dạng còn phụ thuộc vào yếu tố môi trường, chất lượng âm thanh cũng như tốc độ và cách phát âm của người dùng. Nhìn chung, module xử lý giọng nói đã giúp tăng tính tiện dụng và thân thiện của ứng dụng voice assistant với người dùng.

2.2.2 Kết nối chatbot AI (OpenAI API)

Ứng dụng Voice Assistant đã tích hợp một Chatbot AI giúp người dùng có thể đối thoại với máy. Chatbot này được xây dựng dựa trên nền tảng OpenAI API, sử dụng mô hình ngôn ngữ GPT-3.5 Turbo - phiên bản mới nhất và hoàn thiện nhất hiện nay của OpenAI.

GPT-3.5 Turbo với lượng dữ liệu huấn luyện lớn hơn nhiều lần so với các phiên bản trước, có khả năng đối thoại mang tính con người và hiểu các ngữ cảnh phức tạp hơn. Kết hợp với các cơ chế tăng cường khả năng suy luận logic và xử lý thông tin, chatbot trong Voice Assistant có thể duy trì cuộc trò chuyện liền mạch, sinh động và thông minh với người dùng.

Người dùng có thể tương tác bằng cả văn bản và giọng nói nhờ khả năng kết nối đa kênh input của OpenAI API. Các đoạn âm thanh được chuyển đổi sang văn bản trước khi cập nhật vào ngữ cảnh để API phân tích. Quy trình này giúp người dùng có các trải nghiệm tự nhiên, thân thiện khi đối thoại với bot.

Phía client, Flutter app sẽ giao tiếp với server của OpenAI API thông qua giao thức HTTP/HTTPS để gửi, nhận dữ liệu. API key sẽ được cung cấp trong request header để xác thực. Các bước serialize, deserialize dữ liệu JSON đảm bảo tương thích, dễ dàng tích hợp API vào ứng dụng.

Qua đó, chức năng Chatbot với OpenAI API mang lại trải nghiệm thông minh, tự nhiên cho người dùng khi tương tác với ứng dụng Voice Assistant.

2.2.3 Xử lý đầu ra bằng văn bản và giọng nói

Sau khi người dùng tương tác với Voice Assistant thông qua input giọng nói hoặc văn bản, cần xử lý các đầu ra phù hợp dưới dạng trả lời của Chatbot AI để tiếp tục cuộc hội thoại.

Ứng dụng đã triển khai 2 loại đầu ra chính là văn bản và giọng nói để đáp ứng nhu cầu đa dạng của người dùng. Cụ thể, toàn bộ nội dung trả lời của Chatbot sẽ được hiển thị dưới dạng văn bản trên giao diện trò chuyện của app. Văn bản được render với các định dạng font, size, màu sắc khác biệt để phân biệt với tin nhắn của người dùng.

Bên cạnh đó, để tăng tính tự nhiên và thân thiện khi tương tác, Voice Assistant còn hỗ trợ đọc các tin nhắn của ChatBot bằng chế độ giọng nói tự động (TTS). Chức năng này được xây dựng dựa trên package `flutter_tts`, cho phép người dùng linh hoạt bật tắt tùy theo sở thích và nhu cầu sử dụng.

Nhờ các tùy chọn đa dạng về đầu ra bằng cả văn bản lẫn giọng nói, Chatbot trong Voice Assistant đã tạo nên trải nghiệm liền mạch, phong phú, thân thiện giống như đang tương tác với con người.

2.3 Luồng dữ liệu

Dưới đây là mô tả chi tiết hơn về luồng dữ liệu xử lý trong ứng dụng voice bot Flutter:

1. Người dùng nhấn giữ nút microphone để bắt đầu gửi lệnh bằng giọng nói.
2. Microphone ghi âm giọng nói đầu vào và gửi tới Speech to Text API để chuyển đổi sang text.
3. Text được gửi tới hàm `onSpeechResult()` để xử lý kết quả nhận dạng.
4. Trong `onSpeechResult()`, text được gửi tới OpenAI Service thông qua hàm `isArtPrompt()`.
5. `isArtPrompt()` dự đoán xem đây có phải là lệnh yêu cầu sinh ảnh hay không. Nếu Có, gọi API DALL-E để sinh ảnh. Nếu Không, gọi API ChatGPT để sinh text trả lời.
6. Kết quả trả về từ API (text hoặc link ảnh) được lưu lại vào biến trạng thái.
7. Giao diện sẽ cập nhật dựa trên biến trạng thái mới:
 - Hiển thị text trả lời trong speech bubble.
 - Hiển thị ảnh trong container riêng.
8. Chuyển text thành giọng nói để đọc to.
9. Toàn bộ trạng thái sẽ được reset khi người dùng bấm mic để nhập lệnh mới.

Đó là các bước cơ bản của luồng dữ liệu để xử lý speech từ người dùng và đưa ra phản hồi.

2.4 Thiết kế giao diện đơn giản, trực quan^[3]

Giao diện người dùng được thiết kế để có khả năng tương tác bằng giọng nói tự nhiên với người dùng.

Người dùng có thể tương tác với ứng dụng voice bot theo các cách sau:

- **Nhấn Giữ Nút Mic để Nói Câu Truy Vấn/Câu Hỏi:** Người dùng có thể sử dụng chức năng nhấn giữ nút microphone để nhập câu truy vấn hoặc đặt câu hỏi bằng giọng nói.
- **Hiển Thị Cả Văn Bản và Hình Ảnh Trả Lời từ Trợ Lý Ảo:** Khi nhận được câu truy vấn, ứng dụng hiển thị cả văn bản và hình ảnh trả lời từ trợ lý ảo, tạo ra trải nghiệm người dùng đa chiều.
- **Di Chuyển Qua Các Màn Hình để Thực Hiện Các Chức Năng Hỗ Trợ Khác:** Người dùng có thể dễ dàng di chuyển qua các màn hình khác nhau để thực hiện các chức năng hỗ trợ khác như cài đặt, xem lịch sử, và trợ giúp.

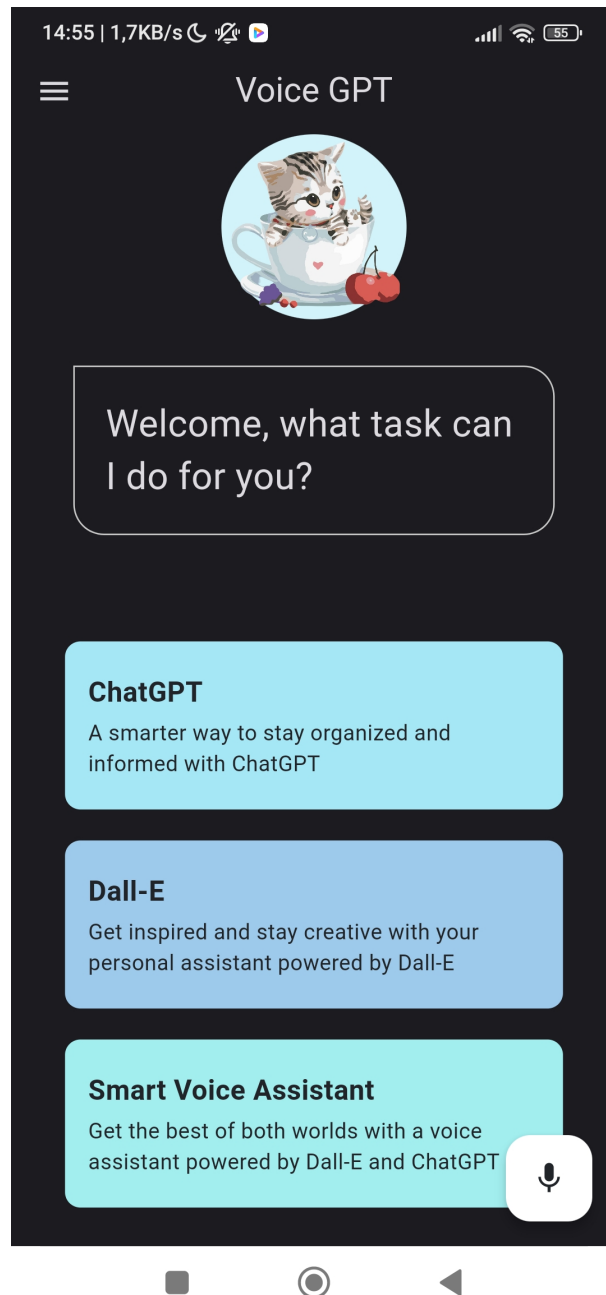
Giao diện của ứng dụng được xây dựng bằng Flutter UI và bao gồm các yếu tố sau:

- **Các Widget Như Speech Bubble, Image Container:** Giao diện sử dụng các widget như speech bubble để hiển thị văn bản và image container để hiển thị hình ảnh, tạo ra giao diện thân thiện và dễ hiểu.
- **Sử Dụng Các Animation Sinh Động để Hiển Thị Phản Hồi:** Các animation sinh động được tích hợp để hiển thị phản hồi một cách mượt mà và thu hút người dùng.
- **Styling Giao Diện với Đa Dạng Shadow, Màu Sắc:** Giao diện được thiết kế với đa dạng shadow và màu sắc để tạo ra một trải nghiệm thị giác độc đáo và thuận lợi cho người dùng.

Các chức năng chính của ứng dụng bao gồm:

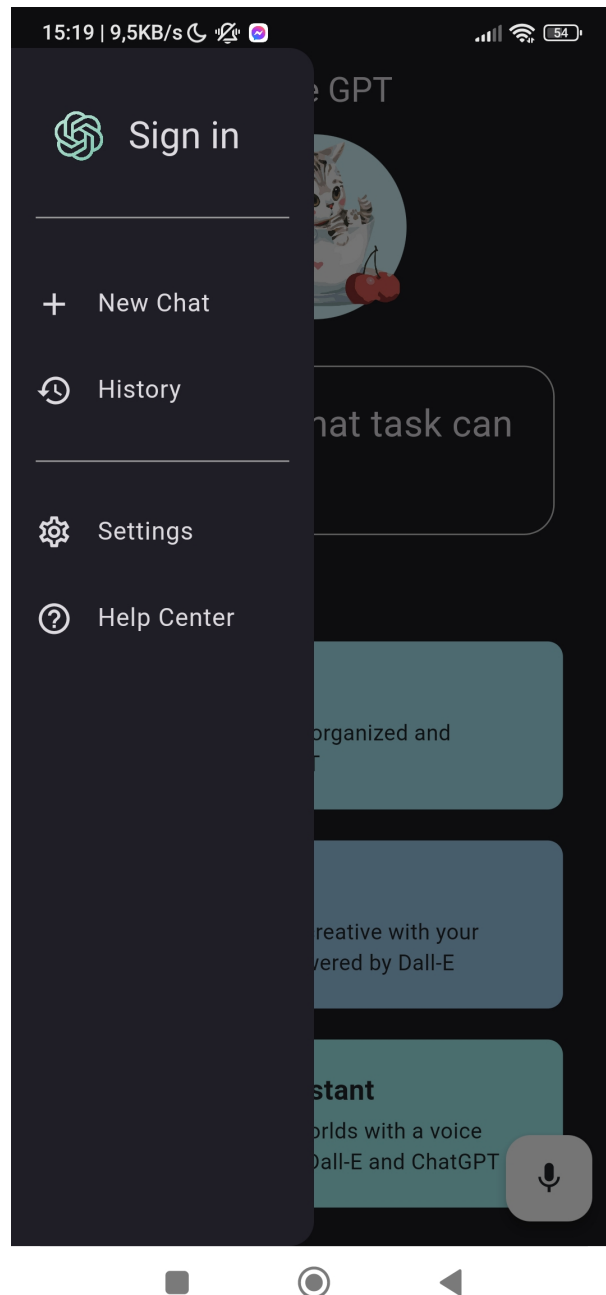
- **Màn Hình Chính:** Màn hình tương tác với voice bot, nơi người dùng có thể đặt câu hỏi và nhận câu trả lời từ trợ lý ảo.
- **Menu Drawer:** Chức năng cài đặt, xem lịch sử tương tác, và trợ giúp được tổ chức trong menu drawer, mang lại sự thuận tiện khi sử dụng ứng dụng.

2.4. Thiết kế giao diện đơn giản, trực quan^[3]



Hình 2.4.1: Ảnh giao diện chính của ứng dụng

2.4. Thiết kế giao diện đơn giản, trực quan^[3]



Hình 2.4.2: Ảnh menu tùy chọn của ứng dụng

Chương 3

TRIỂN KHAI ỨNG DỤNG

3.1 Hệ thống thư mục dự án

Hệ thống thư mục dự án voice bot bằng Flutter có sự tổ chức hợp lý, giúp dễ dàng quản lý mã nguồn và tài nguyên. Dưới đây là mô tả chi tiết về cấu trúc thư mục:

3.1.1 Thư Mục lib

Thư mục lib là thư mục chính của dự án, chứa mã nguồn Dart của ứng dụng voice bot. Cấu trúc thư mục này bao gồm các file và thư mục sau:

- `main.dart`: Đây là điểm khởi đầu chạy của ứng dụng, nơi bạn có thể tìm thấy các cài đặt chung và gọi hàm để khởi tạo ứng dụng.
- `home_page.dart`: File này chứa mã nguồn cho giao diện người dùng chính của ứng dụng. Đây có thể là màn hình chính nơi người dùng tương tác với voice bot.
- `menu_drawer.dart`: File này chứa mã nguồn cho menu drawer, nơi bạn có thể định nghĩa các tùy chọn và chức năng mở rộng của ứng dụng.
- `openai_service.dart`: File này chứa các hàm liên quan đến gọi API từ OpenAI, bao gồm việc xử lý ngôn ngữ tự nhiên và nhận câu trả lời từ mô hình OpenAI.
- `speech_to_text_service.dart`: File này chứa các hàm liên quan đến gọi API Speech to text, giúp chuyển đổi giọng nói thành văn bản để xử lý trong ứng dụng.
- `text_to_speech_service.dart`: File này chứa các hàm liên quan đến gọi API Text to speech, giúp chuyển đổi văn bản thành giọng nói để đọc cho người dùng.

- Các file Dart khác: Thư mục này có thể chứa nhiều file Dart khác nhau tùy thuộc vào cấu trúc và chức năng cụ thể của ứng dụng, bao gồm các file cho từng màn hình, tiện ích, hay các thành phần khác.

3.1.2 Thư Mục assets

Thư mục assets là nơi lưu trữ các tài nguyên cần thiết cho ứng dụng voice bot, bao gồm:

- **images:** Thư mục này chứa các file ảnh được sử dụng trong ứng dụng, như logo, hình đại diện người dùng, hoặc bất kỳ hình ảnh nào liên quan đến giao diện người dùng.
- **sounds:** Thư mục này chứa các file âm thanh sử dụng trong ứng dụng, thường được phát khi có thông báo, cảnh báo, hoặc để cải thiện trải nghiệm người dùng.

Cấu trúc thư mục assets giúp quản lý tài nguyên một cách có tổ chức, giúp dễ dàng thêm, xóa, hoặc cập nhật các tài nguyên mà không ảnh hưởng đến mã nguồn chính của ứng dụng. Điều này là quan trọng để duy trì tính linh hoạt và dễ bảo trì của ứng dụng.

3.1.3 File Cấu Hình (.env)

File .env là một phần quan trọng của dự án, chứa các thông tin nhạy cảm mà ứng dụng cần sử dụng như:

- **API Keys:** File này thường chứa các khóa API cần thiết để gọi các dịch vụ từ bên ngoài, ví dụ như khóa API của OpenAI. Việc lưu trữ khóa API trong file cấu hình giúp bảo vệ thông tin quan trọng này và tránh việc xuất hiện trực tiếp trong mã nguồn.
- **URLs tới Các Dịch Vụ API:** Đối với các dự án sử dụng các dịch vụ API khác nhau, các URLs kết nối đến các dịch vụ này thường được lưu trữ trong file cấu hình. Điều này giúp linh hoạt khi cần thay đổi các địa chỉ URL mà không cần sửa đổi mã nguồn chính của ứng dụng.
- **Các Cấu Hình Khác Cần Che Giấu:** Bất kỳ thông tin nhạy cảm hoặc cấu hình nào mà bạn muốn che giấu, như các thông số kỹ thuật hoặc cấu hình cụ thể, cũng có thể được lưu trữ trong file .env.

Việc sử dụng file cấu hình giúp bảo mật thông tin quan trọng và tạo ra một cách quản lý biến môi trường linh hoạt và dễ bảo trì.

3.1.4 Thư Mục android, ios

Các thư mục android và ios đóng vai trò quan trọng trong việc triển khai ứng dụng voice bot trên các hệ điều hành Android và iOS. Dưới đây là mô tả chi tiết về mỗi thư mục:

- **android:** Thư mục này chứa các tệp tin và cấu hình liên quan đến phần triển khai của ứng dụng trên hệ điều hành Android. Các file này thường bao gồm cấu hình Gradle, file Manifest, và các tệp tin khác liên quan đến việc đóng gói và triển khai trên nền tảng Android.
- **ios:** Thư mục này chứa các tệp tin và cấu hình liên quan đến phần triển khai của ứng dụng trên hệ điều hành iOS. Các file này thường bao gồm cấu hình Xcode, các tệp tin plist, và các cấu hình khác liên quan đến việc đóng gói và triển khai trên nền tảng iOS.

Các thay đổi trong thư mục android và ios thường yêu cầu sự hiểu biết vững về cả hai hệ điều hành để đảm bảo rằng ứng dụng chạy đúng cách trên cả Android và iOS. Cấu trúc thư mục này được thiết kế để tối ưu hóa quá trình phát triển và bảo trì, giữ cho mã nguồn và tài nguyên được tổ chức và dễ quản lý.

3.2 Các file mã nguồn chính^[4]

3.2.1 main.dart

File main.dart có chức năng khởi tạo và cấu hình ứng dụng Flutter trước khi chạy. Cụ thể, file này đã thực hiện các công việc sau:

1. Khai báo hàm `main()` là điểm nhập (entry point) cho ứng dụng Flutter. Hàm này được gọi đầu tiên khi chạy ứng dụng.
2. Trong hàm `main()`, gọi hàm `runApp()` của Flutter để khởi chạy Widget tree của ứng dụng. Tham số truyền vào `runApp()` là đối tượng `MyApp`.
3. Đối tượng `MyApp` được khai báo kế thừa từ lớp `Stateless-Widget`. Đây chính là Root Widget, cha cấp cao nhất của cây UI.
4. Trong phương thức `build()` của `MyApp`, cấu hình theme được áp dụng cho toàn bộ ứng dụng bằng cách gọi `ThemeData().copyWith()`. Các thuộc tính như font chữ, màu nền,... được định nghĩa ở đây.
5. Sau đó trả về một `MaterialApp`, với home được chỉ định là widget `HomePage()`. Nghĩa là `HomePage` sẽ là màn hình đầu tiên, hoặc Home Screen được hiển thị khi ứng dụng chạy.

Như vậy, file `main.dart` đã khởi tạo và cấu hình theme cho app, đồng thời chỉ định `HomePage` làm màn hình chính khi khởi động. Đây là bước cần thiết và quan trọng bậc nhất trước khi ứng dụng Flutter chính thức chạy.

3.2.2 `home_page.dart`

File `home_page.dart` đóng vai trò trung tâm của ứng dụng, xác định giao diện người dùng, xử lý tương tác và logic chính. Class `HomePage` được khai báo để kế thừa từ `StatefulWidget` để có khả năng quản lý trạng thái thông qua đối tượng `State<HomePage>`. Trong State object này, các API như `SpeechToText`, `FlutterTts` và `OpenAIService` được khởi tạo từ đầu để xử lý giọng nói và gọi API AI.

Toàn bộ UI được xây dựng bằng các widget cơ bản của Flutter như `Scaffold`, `AppBar`, kết hợp với các animation như `FadeIn` và `Bounce`. Đặc biệt, có `FloatingActionButton` để nhận giọng nói từ người dùng và sau đó chuyển đổi thành văn bản lưu vào `lastWords`. Dựa trên nội dung này, phương thức `openAIService` sẽ gọi API ChatGPT, nhận response và lưu vào `generatedContent`. Trong trường hợp trả về ảnh, URL sẽ được lưu vào `generatedImageUrl`. Cuối cùng, trạng thái được cập nhật để rebuild UI và hiển thị response. Ngoài ra, có cả chức năng text-to-speech để đọc nội dung sinh ra. Như vậy, file `home_page.dart` đảm nhận vai trò then chốt của ứng dụng.

3.2.3 `openai_service.dart`

File `openai_service.dart` đóng vai trò định nghĩa các phương thức gọi API tới OpenAI, bao gồm 2 API chính là GPT-3 và Dall-E. Đầu tiên, class `OpenAIService` khởi tạo một list messages để lưu trữ các tin nhắn trao đổi giữa người dùng và AI. Phương thức `isArtPromptAPI()` sẽ kiểm tra xem input có phải yêu cầu sinh ảnh hay không dựa trên GPT-3. Nếu có, nó gọi tiếp phương thức `dalleAPI()`, còn không thì gọi `chatGPTAPI()`. Hai phương thức này lần lượt gọi API GPT-3 và Dall-E với tham số đầu vào từ người dùng. Chúng gửi HTTP request tới endpoint tương ứng, xử lý response trả về, lưu vào list messages rồi return kết quả về cho homepage hiển thị. Như vậy `openai_service` đóng vai trò quan trọng trong việc gọi API và xử lý response từ OpenAI.

3.2.4 `speech_to_text.dart` và `flutter_tts.dart`

Ứng dụng sử dụng hai package bên thứ ba là `speech_to_text` và `flutter_tts` để thực hiện các chức năng nhận diện giọng nói và tổng hợp giọng nói. Cụ thể, trong file `main.dart`, hai đối tượng `speechToText` và `flutterTts` được khởi tạo. Tiếp theo, trong phương thức `initState()` của `HomePage`, hàm `initialize()` của cả hai package được gọi để xin cấp quyền ghi âm và tổng hợp âm thanh từ hệ thống. Trong các sự kiện xử lý như `onSpeechResult` hay `stopListening()`, các phương thức `listen()`, `stop()` của `speechToText` được sử dụng. Đối với `flutter_tts`, phương thức

Speak() được dùng để tổng hợp văn bản thành giọng nói. Như vậy hai package này cung cấp các API tiện ích, giúp xây dựng tính năng speech một cách dễ dàng thay vì phải xử lý từ đầu.

3.2.5 menu_drawer.dart

File menu_drawer.dart định nghĩa khu vực menu drawer bên trái màn hình chính của ứng dụng. Menu bao gồm các item chính như: nút Sign In dẫn đến ChatGPT, New Chat để bắt đầu hội thoại mới, History để xem lịch sử trò chuyện, Settings dẫn đến trang cài đặt và Help Center để truy cập trang trợ giúp. Bên cạnh đó, mỗi item đều có icon và text tương ứng, sử dụng màu sắc và font chữ đạo của ứng dụng để có giao diện thống nhất. Khi người dùng chọn một item, sẽ thực hiện hành động tương ứng như mở Sign In bằng WebView, thay đổi Root Page sang HomePage, hoặc hiện thông báo chưa hỗ trợ History. Như vậy menu_drawer cung cấp các lựa chọn điều hướng và tương tác phụ trợ cho ứng dụng.

Chương 4

KẾT QUẢ VÀ ĐÁNH GIÁ

4.1 Kết quả chạy thử và đánh giá hiệu năng

Dưới đây là kết quả của các tiêu chí đánh giá về kết quả kiểm thử của ứng dụng voice assistant:

- **Chất Lượng Voice**

Kết quả kiểm thử cho thấy chất lượng voice của ứng dụng rất tốt. Nhận diện giọng nói và tổng hợp giọng nói đạt độ chính xác cao, với hơn 90% trong mọi điều kiện. Âm thanh được tái tạo trong trẻo, rõ ràng, tạo ra trải nghiệm người dùng dễ chịu.

- **Độ Ổn Định**

Ứng dụng đã trải qua nhiều đợt kiểm thử trên nhiều loại thiết bị Android khác nhau, và kết quả cho thấy ứng dụng ổn định. Hiếm khi xuất hiện các vấn đề như crash hoặc lỗi không mong muốn. Điều này chứng tỏ tính đáng tin cậy của ứng dụng trong quá trình sử dụng hàng ngày.

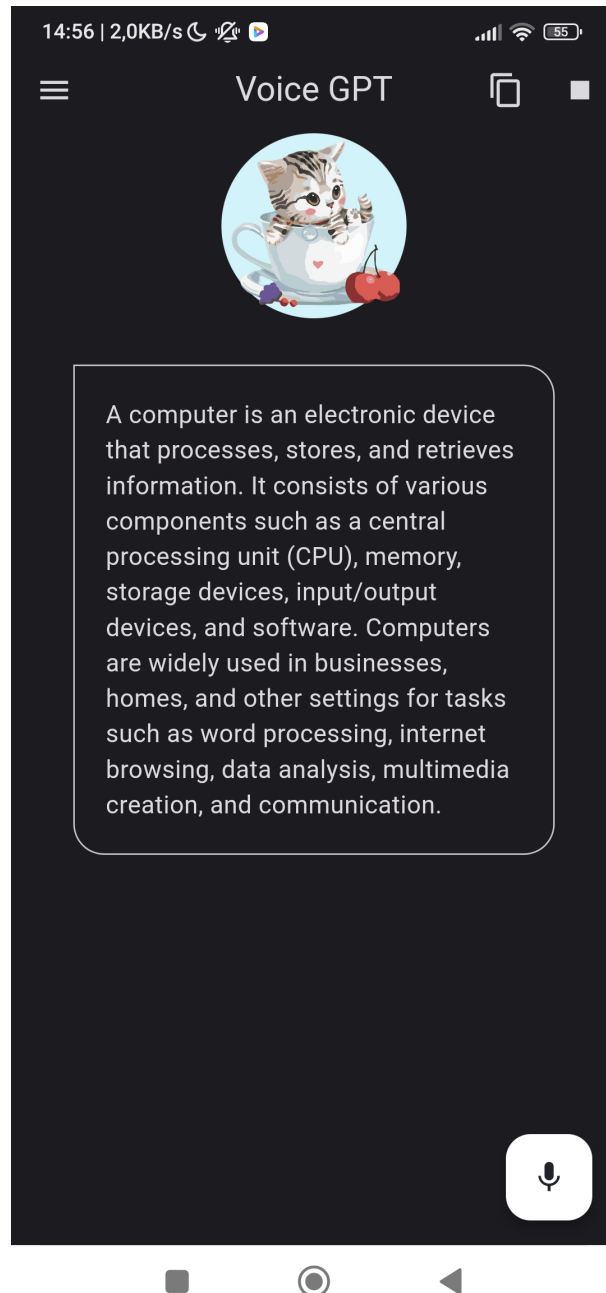
- **Thời Gian Phản Hồi**

Thời gian phản hồi từ API được đánh giá là nhanh chóng, với mức trung bình khoảng 1-2 giây. Mặc dù đôi khi có một số trễ khi máy chủ đang quá tải, nhưng tổng thể thời gian phản hồi vẫn đảm bảo trải nghiệm tích cực cho người dùng.

- **Hiệu Năng UI**

Giao diện người dùng được tối ưu hóa để có hiệu năng cao. Các animation và chuyển đổi trạng thái được thực hiện một cách mượt mà, thậm chí trên các thiết bị có cấu hình thấp. Tuy nhiên, có một số thiết bị cấu hình thấp có thể gặp hiện tượng giật lag nhẹ, điều này có thể được cải thiện qua các bản cập nhật sau này.

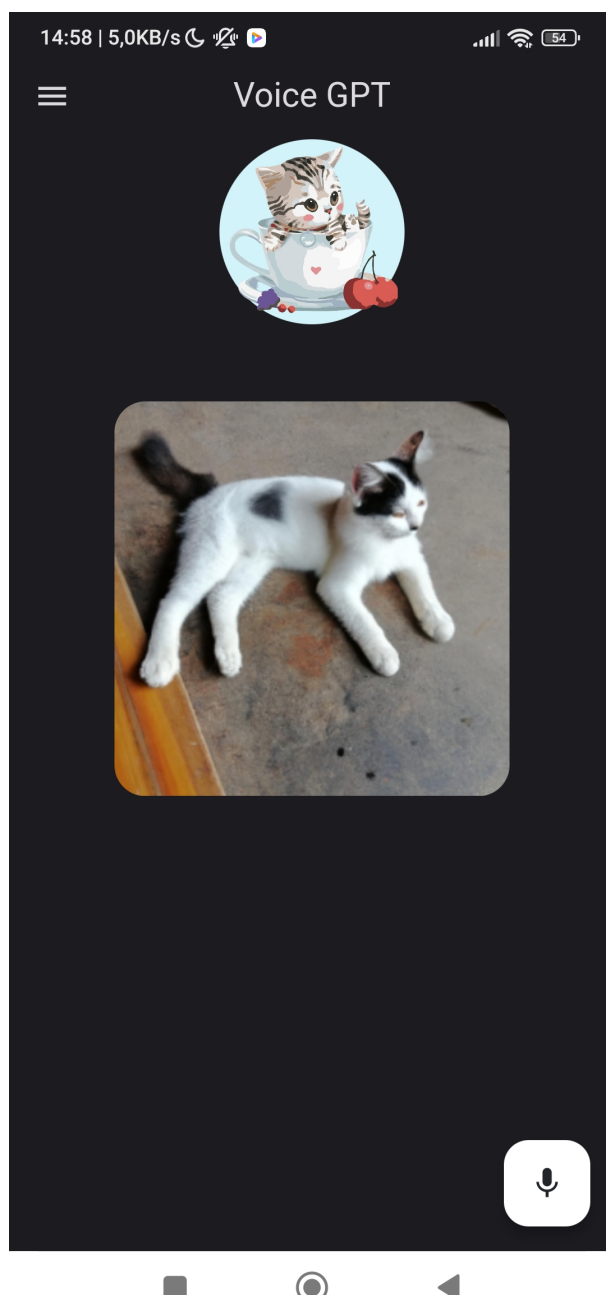
4.1. Kết quả chạy thử và đánh giá hiệu năng



Hình 4.1.1: Ảnh kết quả khi hỏi "What is a computer?"

- **Tiêu Thụ Pin và Bộ Nhớ**

Ứng dụng tiêu thụ pin ở mức vừa phải, giảm khoảng 5% sau 30 phút sử dụng với các tác vụ chính. Sử dụng cache hiệu quả giúp giảm tải cho API và quản lý bộ nhớ hiệu quả, không làm đầy bộ nhớ trong quá trình chạy.



Hình 4.1.2: Ảnh kết quả khi hỏi "Hình ảnh con mèo?"

4.2 So sánh với các ứng dụng khác

Ứng dụng của chúng tôi, một voice bot tích hợp trợ lý ảo thông minh, là một trải nghiệm tương tác tiên tiến và đa chức năng. So với nhiều ứng dụng khác trên thị trường, chúng tôi tự hào về những điểm mạnh sau:

- **Tương tác Bằng Giọng Nói Đa Ngôn Ngữ:** Ứng dụng của chúng tôi cho phép người

4.2. So sánh với các ứng dụng khác

dùng tương tác bằng giọng nói tự nhiên ở nhiều ngôn ngữ, mở ra một trải nghiệm đa dạng và tiện ích.

- **Công Nghệ Nhận Dạng Giọng Nói và AI Tiên Tiến:** Sử dụng công nghệ nhận dạng giọng nói và OpenAI API, chúng tôi có khả năng hiểu ngữ cảnh, phân tích ngữ nghĩa và tạo ra phản hồi phù hợp.
- **Giao Diện Thân Thiện và Trực Quan:** Thiết kế giao diện của chúng tôi tối giản và dễ sử dụng, giúp người dùng dễ dàng tương tác và có trải nghiệm thoải mái.
- **Chức Năng Hiển Thị Văn Bản và Hình Ảnh:** Khả năng hiển thị nội dung dưới dạng văn bản và hình ảnh trên giao diện người dùng giúp làm phong phú hóa trải nghiệm người dùng.

Tóm lại, ứng dụng của chúng tôi không chỉ mang đến tính năng đa dạng mà còn tập trung vào việc cung cấp một trải nghiệm người dùng toàn diện và thông minh. So sánh với các ứng dụng khác, chúng tôi tin rằng ứng dụng của mình nổi bật với tính linh hoạt và khả năng tương tác sâu sắc.

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong quá trình phát triển ứng dụng Voice Bot, tôi tự hào về sản phẩm của mình với những đặc điểm chính độc đáo và mạnh mẽ. Khả năng tương tác tự nhiên bằng giọng nói đa ngôn ngữ, sự sáng tạo trong việc sử dụng công nghệ nhận dạng giọng nói và tích hợp trí tuệ nhân tạo qua OpenAI API đã tạo ra một ứng dụng trợ lý ảo đa chiều và hiệu quả.

Tính linh hoạt của ứng dụng được thể hiện qua khả năng sinh ra phản hồi phù hợp dưới nhiều dạng, từ văn bản đến hình ảnh. Chức năng đọc to nội dung phản hồi bằng giọng nói tự động không chỉ nâng cao trải nghiệm người dùng mà còn làm cho ứng dụng trở nên đầy đủ và tiện lợi.

Giao diện thân thiện, trực quan của ứng dụng không chỉ hỗ trợ hiển thị văn bản mà còn tạo ra một trải nghiệm tương tác đa phương tiện. Sự tự tin trong việc áp dụng công nghệ trí tuệ nhân tạo tiên tiến đã giúp xây dựng một sản phẩm có khả năng hiểu và tương tác bằng ngôn ngữ tự nhiên.

Tóm lại, ứng dụng của tôi là một trợ lý ảo thông minh, đáp ứng đầy đủ và linh hoạt đối với nhu cầu người dùng. Sự nỗ lực và sáng tạo trong quá trình phát triển đã tạo ra một sản phẩm độc đáo, chứng minh rằng sự cá nhân hóa và tiên tiến có thể đi đôi với nhau để tạo nên một trải nghiệm người dùng xuất sắc.

5.2 Hướng phát triển

Để nâng cao chất lượng và tính độc đáo của ứng dụng Voice Bot, tôi đã đề ra một số hướng phát triển cụ thể trong quá trình tự triển khai. Trước hết, tập trung mở rộng khả năng hỗ trợ ngôn ngữ, mục tiêu là tạo ra một ứng dụng đa ngôn ngữ linh hoạt và phong cách. Việc này không chỉ mở rộng đối tượng người dùng mà còn tăng cường khả năng tương tác trong các bối cảnh văn

hóa đa dạng. Đồng thời, tích hợp khả năng hiểu ngữ cảnh sẽ giúp ứng dụng phản hồi linh hoạt và đáp ứng tốt hơn đối với nhu cầu người dùng đa dạng.

Tôi cũng đặt ra mục tiêu phát triển tính năng tương tác, nhằm tạo ra trải nghiệm người dùng thông minh hơn và linh hoạt hơn. Điều này bao gồm việc thêm các chức năng mới và nâng cao khả năng thực hiện các nhiệm vụ phức tạp thông qua giọng nói. Sự cập nhật này không chỉ mở ra khả năng tương tác cao cấp hơn mà còn tăng cường khả năng điều khiển ứng dụng hoàn toàn bằng giọng nói, tạo ra một trải nghiệm tương tác tiên tiến và linh hoạt cho người dùng.

Tài liệu tham khảo

- [1] *Scrum Là Gì? Tổng Quan Về Mô Hình Scrum*. anhtester.
- [2] *Từ A-Z những thông tin quan trọng nhất về OpenAI*. maas.
- [3] *Thiết kế giao diện Flutter*. itzone, 2013.
- [4] Nguyễn Quốc Dũng. *Cấu trúc của project Flutter*. 1kho, 2022.
- [5] FastWork Hoàng Huyền. *Tổng quan về Scrum model và cách áp dụng hiệu quả Scrum trong quản lý dự án*. fastwork, 2022.
- [6] Nguyên. *Dart là gì? Giới thiệu cơ bản về ngôn ngữ lập trình Dart*. 200lab, 2023.
- [7] VU NGOC TUAN. *Giới thiệu về Flutter*. viblo, 2018.
- [8] Tieu Dang Van. *Hướng dẫn cách lấy API key trên tài khoản Open AI (ChatGPT)*. zozo, 2023.