

ĐỒ ÁN CHUYÊN NGÀNH
ỨNG DỤNG HỌC TIẾNG ANH

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: **ThS. Nguyễn Huy Cường**

Lớp: **21DTHE1**

Sinh viên thực hiện	MSSV
Nguyễn Nhật Huy	2180607567
Nguyễn Xuân Bắc	2180600127
Nguyễn Ngô Trúc Chi	2180607325
Trần Bá Dũng	2180608485

LỜI CẢM ƠN

Lời đầu tiên, cho nhóm chúng em xin gửi lời cảm ơn chân thành đến quý thầy, cô giảng viên Trường Đại học Công Nghệ Thành Phố Hồ Chí Minh và quý thầy cô khoa Công Nghệ Thông Tin đã giúp cho nhóm chúng em có những kiến thức để thực hiện đồ án này.

Đặc biệt, nhóm chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Giảng viên – ThS. Nguyễn Huy Cường, người đã hướng dẫn cho chúng em trong suốt thời gian làm đề tài. Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp cho nhóm chúng em hoàn thành tốt báo cáo môn học của mình. Một lần nữa chúng em xin chân thành cảm ơn thầy và chúc thầy dồi dào sức khỏe.

Trong thời gian một học kỳ thực hiện đề tài này, nhóm chúng em đã vận dụng những kiến thức nền tảng và kết hợp những kiến thức mới từ thầy cô, bạn bè cũng như nhiều nguồn tài liệu tham khảo.

Từ đó, nhóm chúng em đã vận dụng những gì mà chúng em đã thu thập được để hoàn thành tốt đề tài mà chúng em đã chọn một cách tốt nhất. Tuy nhiên, trong quá trình áp dụng những kiến thức đó thì một phần chuyên môn chúng em vẫn còn hạn chế và còn thiếu nhiều kinh nghiệm nên nội dung của báo cáo không tránh khỏi những thiếu sót, chúng em rất mong nhận được sự góp ý, chỉ bảo thêm của quý thầy cô nhằm cải thiện những kiến thức của mình để nhóm chúng em có thể dùng thực hiện tiếp các đề tài khác trong tương lai cũng như là trong việc học tập và làm việc sau này. Một lần nữa xin gửi đến thầy cô, bạn bè lời cảm ơn chân thành và tốt đẹp nhất.

MỤC LỤC

Ứng dụng học tiếng Anh

LỜI CẢM ƠN	1
MỤC LỤC	2
DANH MỤC HÌNH ẢNH	4
CHƯƠNG 1. TỔNG QUAN	6
1.1. Mục tiêu và lý do chọn đề tài.....	6
1.1.1. Mục tiêu	6
1.1.2. Lý do chọn đề tài	6
1.2. Nghiên cứu khảo sát	7
1.2.1. Khảo sát thực trạng hiện tại	7
1.2.2. Khảo sát các ứng dụng tương tự	8
1.3. Nhiệm vụ đồ án.....	9
1.3.1. Mục tiêu đồ án	9
1.3.2. Sơ lược các tính năng.....	9
1.4. Cấu trúc đồ án	10
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	11
2.1. Cơ sở phát triển ứng dụng di động	11
2.1.1. Khái quát.....	11
2.1.2. Phân loại ứng dụng di động	11
2.2. Cơ sở lý thuyết.....	13
2.2.1. Flutter Framework	13
2.2.2. Quản lý trạng thái	15
2.2.3. Kiến trúc sạch (Clean Architecture)	17
2.2.4. Firebase.....	18
2.2.5. OpenAI API	19
CHƯƠNG 3. TRIỂN KHAI ỨNG DỤNG.....	20
3.1. Lựa chọn công nghệ.....	20
3.2. Thiết kế cơ sở dữ liệu	21
3.2.1. Cơ sở dữ liệu cục bộ (SQLite)	21
3.2.2. Cơ sở dữ liệu đám mây (Cloud Firestore)	23
3.3. Kiến trúc và luồng dữ liệu	24
3.3.1. Lớp Data	25
3.3.2. Lớp Domain	26
3.3.3. Lớp Presentation	28

3.4. Biểu đồ mô tả hệ thống	29
3.4.1. Sequence Diagram	29
3.4.2. Usecase Diagram	30
CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM.....	31
4.1. Các chức năng của người dùng.....	31
4.1.1. Các cuộc hội thoại theo chủ đề	31
4.1.2. Các bài học tiếng Anh.....	33
4.1.3. Các chế độ nhỏ.....	35
4.1.4. Hoạt động tài khoản.....	40
4.1.5. Cài đặt tính năng hệ thống	41
4.1.6. Trang chủ	42
4.1.7. Tính năng xác thực người dùng	43
4.2. Các chức năng của Admin	46
4.2.1. Quản lý danh sách người dùng	46
4.2.2. Quản lý danh sách các câu thành ngữ.....	46
4.2.3. Quản lý các từ vựng.....	47
4.2.4. Quản lý các bài học.....	47
4.2.5. Quản lý danh sách các mẫu câu	48
4.2.6. Quản lý loại mẫu câu thường dùng.....	48
4.2.7. Quản lý danh sách âm IPA	49
4.2.8. Quản lý danh sách thì thông dụng	49
4.3. Giao diện chào mừng hệ thống	50
CHƯƠNG 5. KẾT LUẬN	51
5.1. Kết luận.....	51
5.2. Hướng phát triển tương lai.....	51
TÀI LIỆU THAM KHẢO	52
Tiếng Việt.....	52

DANH MỤC HÌNH ẢNH

Hình 1.2-1. Các ứng dụng tiếng anh tương tự.....	8
Hình 2.2.1. Tổng quan kiến trúc flutter (flutter.dev)	13
Hình 2.2.2. Một cây widget đơn giản. (flutter.dev).....	15
Hình 2.2.3-1. Minh họa kiến trúc sạch (flutter.dev)	17
Hình 2.2.3-2. Áp dụng kiến trúc sạch vào đồ án	18
Hình 3.2.1. Cơ sở dữ liệu cục bộ (SQLite)	22
Hình 3.2.2. Cơ sở dữ liệu đám mây (Cloud Firestore)	23
Hình 3.3.1. Kiến trúc và luồng dữ liệu	24
Hình 3.4.1. Sequence Diagram.....	29
Hình 3.4.2. Usecase Diagram	30
Hình 4.1. Thanh công cụ dưới	30
Hình 4.3. Giao diện hệ thống	48
Hình 4.1.1-1. Chủ đề cuộc hội thoại lớn và chủ đề con	32
Hình 4.1.1-2. Màn hình luyện tập cuộc hội thoại.....	33
Hình 4.1.2-1. Các bài học lớn và bài học con	34
Hình 4.1.3-1. Bảng IPA chi tiết và cách phát âm	35
Hình 4.1.3-2. Trắc nghiệm Quiz	36
Hình 4.1.3-3. Flashcard	37
Hình 4.1.3-4. Luyện bài thi đọc hiểu	38
Hình 4.1.3-5. Tìm kiếm thông tin chi tiết từ vựng	39
Hình 4.1.4-1. Hoạt động tài khoản	40
Hình 4.1.5-1. Cài đặt tính năng hệ thống.....	41
Hình 4.1.5-2. Chuyển đổi song ngữ	41
Hình 4.1.6-1. Trang chủ	42
Hình 4.1.7-1. Tính năng đăng ký	43
Hình 4.1.7-2. Tính năng đăng nhập.....	44
Hình 4.1.7-3. Tính năng quên mật khẩu	45
Hình 4.2.1. Quản lý danh sách người dùng	46
Hình 4.2.2. Quản lý danh sách các câu thành ngữ.....	46
Hình 4.2.3. Quản lý các từ vựng	47
Hình 4.2.4. Quản lý các bài học	47

Hình 4.2.5. Quản lý các mẫu câu	48
Hình 4.2.6. Quản lý loại mẫu câu thường dùng	48
Hình 4.2.7. Quản lý danh sách âm IPA.....	49
Hình 4.2.8. Quản lý danh sách thì thông dụng.....	49

CHƯƠNG 1. TỔNG QUAN

1.1. Mục tiêu và lý do chọn đề tài

1.1.1. Mục tiêu

Mục tiêu chính của đề án là phát triển một ứng dụng di động có khả năng cung cấp tài liệu, bài tập và hoạt động rèn luyện tiếng Anh cho người dùng trên hệ điều hành Android.

Ứng dụng sẽ giúp người dùng cải thiện kỹ năng ngôn ngữ, đặc biệt là các kỹ năng cơ bản: nghe, nói, phản xạ,... thông qua các bài học tương tác, ghi âm và phản hồi tức thì.

Chủ đề đề án nhằm hướng tới việc giúp người dùng cải thiện và tự tin hơn trong việc sử dụng tiếng Anh, đáp ứng nhu cầu ngày càng tăng của mọi người trong việc học lẫn công việc sử dụng tiếng Anh trong cuộc sống hàng ngày.

1.1.2. Lý do chọn đề tài

Tiếng Anh hiện nay không chỉ là một công cụ giao tiếp quốc tế mà còn đóng vai trò rất quan trọng trong nhiều khía cạnh của cuộc sống. Việc nắm vững tiếng Anh đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của mọi người, ảnh hưởng đến việc tìm kiếm công việc, giao tiếp với người nước ngoài, du lịch, nghiên cứu, và cả giải trí. Việc có khả năng sử dụng tiếng Anh một cách thành thạo có thể mở ra nhiều cơ hội mới và cung cấp sự tự tin.

Mặc dù nhu cầu học tiếng Anh ngày càng tăng, có một số thách thức về việc tiếp cận tài liệu học tập chất lượng và phương pháp học hiệu quả, gây ra nhiều khó khăn đặc biệt đối với người học. Vì vậy, việc phát triển một ứng dụng học tiếng Anh chất lượng và hiệu quả có tầm quan trọng đặc biệt trong xã hội hiện nay.

Hi vọng rằng thông qua việc nghiên cứu và phát triển ứng dụng, đề án này sẽ đóng góp vào việc cung cấp một công cụ học tập tiện lợi và hiệu quả cho những người mong muốn cải thiện kỹ năng tiếng Anh của họ.

1.2. Nghiên cứu khảo sát

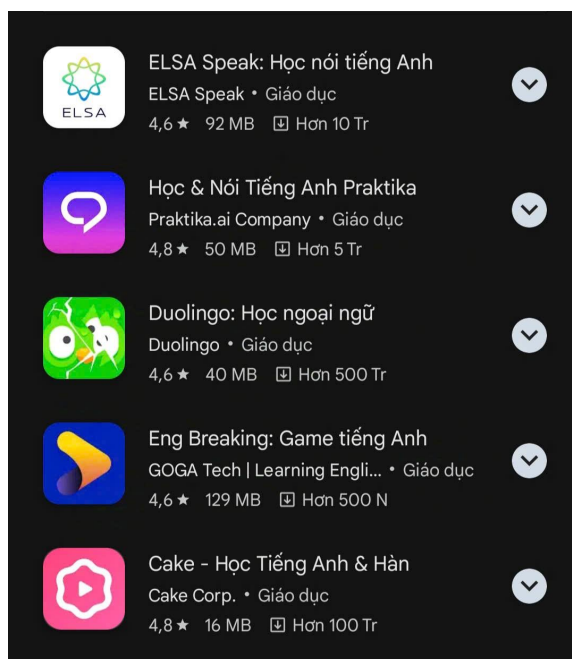
1.2.1. Khảo sát thực trạng hiện tại

Thị trường ứng dụng học Tiếng Anh tại Việt Nam đang chứng kiến sự cạnh tranh ngày càng khốc liệt. Người dùng ngày càng đòi hỏi những ứng dụng không chỉ cung cấp độ chính xác của từ vựng và bài học mà bên cạnh đó đòi hỏi còn phải có giao diện một cách thân thiện cùng với trải nghiệm người dùng mượt mà và kết hợp tính năng cá nhân hóa cao. Tuy nhiên trong những cơ hội các ứng dụng đã và đang đối mặt với nhiều thách thức như việc đảm bảo tính chính xác về mặt nội dung và xây dựng một cộng đồng gắn kết thân thiện trong học tập và đặc biệt quan trọng là việc giữ chân người dùng trong một thời đại mà sự đa dạng trong việc lựa chọn ứng dụng ngày nay.

Để giải quyết những hạn chế hiện nay, nhóm chúng em quyết định tạo và phát triển một ứng dụng học Tiếng Anh thông minh là điều hiển nhiên. Ứng dụng này không chỉ giúp tối ưu hiệu quả việc học tập lẫn việc giảng dạy mà còn tạo ra một môi trường tương tác, cá nhân hóa cho người dùng. Bằng cách tích hợp trí tuệ nhân tạo và dữ liệu lớn, hệ thống có thể tự động hóa nhiều quy trình, cung cấp hứng gợi ý học tập phù hợp và theo dõi tiến độ học tập cá nhân một cách hiệu quả.

1.2.2. Khảo sát các ứng dụng tương tự

Tại Việt Nam, Hiện tại các ứng dụng học Tiếng Anh đã được phát triển và trở nên phổ biến bao gồm như Duolingo, Elsa Speak, và Cake. Những ứng dụng này đang có tính năng từ học từ vựng, ngữ pháp đến luyện phát âm, giao tiếp và nghe hiểu. Sự tiện lợi và tính năng tương tác cao giúp người dùng dễ dàng tiếp cận.



Hình 1.2-1. Các ứng dụng tiếng anh tương tự

Bên cạnh đó còn hỗ trợ qua các bài học giao tiếp thực tế, video và bài tập có tính ứng dụng thực tế, giúp người học nâng cao khả năng sử dụng Tiếng Anh trong môi trường học tập, làm việc và giao tiếp hàng ngày. Với AI trong “Elsa Speak” khi người dùng sử dụng có thể phân tích và đưa phản hồi chi tiết về phát âm, từ đó giúp họ cải thiện kỹ năng nói.

Qua đó các ứng dụng tương tự có các chương trình học theo từng cấp độ, từ cơ bản đến nâng cao và đa dạng trong cách giảng dạy của từng giáo viên và để hỗ trợ người dùng xây dựng lộ trình học tập phù hợp với mục tiêu cần đạt được của mình trong khoảng thời gian nào đó. Tính năng theo dõi tiến độ học tập và hệ thống điểm thưởng cũng tạo động lực cho người học để cố gắng hàng ngày.

Tuy nhiên, vẫn còn một số thách thức khi sử dụng các ứng dụng học tiếng Anh, như việc thiếu sự tương tác trực tiếp với giáo viên và việc cần có kết nối internet ổn định để sử dụng. Nhưng nhờ vào sự phát triển không ngừng của công nghệ, các ứng dụng ngày càng cải tiến để mang đến những trải nghiệm học tập tốt nhất cho người dùng tại Việt Nam.

1.3. Nhiệm vụ đồ án

Từ các vấn đề trên, xây dựng ứng dụng học Tiếng Anh yêu cầu kỹ thuật, chuyên nghiệp hơn và giải quyết các vấn đề khuyết điểm của mô hình tương đương.

1.3.1. Mục tiêu đồ án

Mục tiêu chính của đồ án là phát triển một ứng dụng di động có khả năng cung cấp tài liệu, bài tập và hoạt động rèn luyện tiếng Anh cho người dùng trên cả hệ điều hành.

Ứng dụng sẽ giúp người dùng cải thiện kỹ năng ngôn ngữ, đặc biệt là các kỹ năng giao tiếp: nghe, nói, phản xạ,... thông qua các bài học tương tác, ghi âm và phản hồi tức thì.

Chủ đề đồ án nhằm hướng tới việc giúp người dùng cải thiện và tự tin hơn trong việc sử dụng tiếng Anh, đáp ứng nhu cầu ngày càng tăng của mọi người trong việc học và sử dụng tiếng Anh trong cuộc sống hàng ngày, công việc và du lịch.

1.3.2. Sơ lược các tính năng

Các tính năng chính mà ứng dụng sẽ cung cấp bao gồm:

- Tài liệu học tập đa dạng - thực tế: Cung cấp các tài liệu học tập về từ vựng, ngữ pháp và kỹ năng giao tiếp tiếng Anh.
- Bài tập tương tác: Cho phép người dùng tham gia vào các bài tập tương tác để cải thiện kỹ năng ngôn ngữ.
- Khả năng ghi âm và phản hồi tức thì: Cho phép người dùng ghi âm bản thân dựa trên các câu mẫu có sẵn, sau đó nhận phản hồi chi tiết về cách phát âm và giao tiếp.
- Thiết kế giao diện và trải nghiệm người dùng: Đảm bảo rằng ứng dụng có giao diện dễ sử dụng và trải nghiệm người dùng thú vị.
- Tích hợp các dịch vụ học tập và giải trí: Kết hợp các công cụ học tập và giải trí giúp người dùng mở rộng vốn từ vựng và hiểu biết tiếng Anh một cách thú vị.
- Và nhiều tính năng khác.

1.4. Cấu trúc đồ án

Cấu trúc của đồ án gồm 5 chương như sau:

- Chương 1: Tổng quan:

Chương đầu tiên giới thiệu mục tiêu, lý do chọn đề tài, nghiên cứu các ứng dụng học tiếng Anh trên thị trường và cấu trúc tổng quan của đồ án.

- Chương 2: Cơ sở lý thuyết:

Chương này tập trung vào việc trình bày các nền tảng lý thuyết và kiến thức liên quan đến việc phát triển ứng dụng học tiếng Anh. Nó bao gồm các nguyên lý về việc phát triển ứng dụng di động, Flutter framework và các dịch vụ bên thứ ba được tích hợp trong ứng dụng.

- Chương 3: Triển khai ứng dụng học tiếng Anh:

Chương này trình bày quá trình thực hiện phát triển ứng dụng, bao gồm các bước từ lựa chọn công nghệ, kiến trúc hệ thống, phát triển tính năng.

- Chương 4: Kết quả thực nghiệm:

Chương này tập trung vào các kết quả từ các thử nghiệm, kết quả và đánh giá tổng thể về dự án.

- Chương 5: Kết luận:

Chương cuối cùng tổng hợp kết quả và đánh giá, rút ra các kết luận quan trọng. Chương này cũng sẽ trình bày kế hoạch triển khai của ứng dụng trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Cơ sở phát triển ứng dụng di động

2.1.1. Khái quát

Trong thời đại số hóa, ứng dụng di động đã và đang tiếp tục trở thành một phần quan trọng của cuộc sống hàng ngày của mọi người. Việc phát triển ứng dụng di động đòi hỏi kiến thức sâu rộng về nhiều khía cạnh, bao gồm lập trình, thiết kế giao diện, quản lý dự án và nắm vững các nền tảng phát triển ứng dụng di động.

2.1.2. Phân loại ứng dụng di động

Dựa vào cách được phát triển và hoạt động, ứng dụng di động được chia thành 2 loại chính: Ứng dụng gốc và Ứng dụng di động đa nền tảng.

- Ứng dụng gốc:
 - Là ứng dụng được viết chuyên biệt cho một loại nền tảng như Android, iOS, hay Windows bằng các ngôn ngữ tương ứng của mỗi nền tảng đó. Ví dụ: Swift với iOS và Java còn Kotlin với Android.
 - Mỗi ứng dụng gốc chỉ chạy được trên một nền tảng và không thể mang sang các nền tảng khác. Ví dụ: Ứng dụng cho iOS sẽ không thể chạy được trên các máy có hệ điều hành Android.
 - Điểm mạnh:
 - + Hiệu năng cao: Ứng dụng Native thường có hiệu suất tốt nhất vì chúng được tối ưu hóa cho hệ điều hành cụ thể.
 - + Truy cập đầy đủ vào tính năng của thiết bị: Có khả năng tận dụng toàn bộ tính năng của điện thoại như máy ảnh, cảm biến, GPS, ...
 - + UI/UX tùy chỉnh: Cho phép thiết kế giao diện và trải nghiệm người dùng tùy chỉnh theo ý muốn.
 - Điểm yếu:
 - + Chi phí và thời gian: Phát triển cho cả hai nền tảng (iOS và Android) tốn nhiều thời gian và công sức.
 - + Khó duy trì: Cần duy trì hai mã nguồn riêng biệt cho từng nền tảng.

- Ứng dụng di động đa nền tảng (Cross-Platform Mobile Apps): Đây là loại ứng dụng có thể chạy trên cả hai hệ điều hành iOS và Android từ một mã nguồn duy nhất.
 - Có hai cách chính để phát triển ứng dụng đa nền tảng:
 - + Ứng dụng di động hybrid: Sử dụng các công nghệ web như JavaScript, HTML, CSS để xây dựng ứng dụng, sau đó sử dụng một trình duyệt web nhúng để hiển thị ứng dụng trên cả nhiều nền tảng. Ví dụ về framework cho loại này là React Native.
 - + Ứng dụng di động đa nền tảng thực sự (True Cross-Platform Apps): Sử dụng các công cụ và ngôn ngữ đặc biệt cho phát triển đa nền tảng, như Flutter (sử dụng Dart) và Xamarin (sử dụng C#). Loại này tạo ra các giao diện người dùng native thay vì dựa vào trình duyệt web nhúng.
 - Điểm mạnh:
 - + Tính đa nền tảng: Chia sẻ mã nguồn một lần và sử dụng trên cả hai nền tảng (iOS và Android).
 - + Tiết kiệm thời gian và nguồn lực: Phát triển nhanh hơn và tiết kiệm chi phí so với phát triển native cho cả hai nền tảng.
 - Điểm yếu:
 - + Hiệu năng thấp hơn: Ứng dụng có thể có hiệu năng kém hơn so với Native Apps.
 - + Có thể không hỗ trợ mọi tính năng native: Một số tính năng đặc biệt của từng nền tảng có thể không được hỗ trợ hoặc cần sử dụng Plugins bên ngoài.

Đồ án chuyên ngành này sử dụng Flutter Framework để phát triển ứng dụng đa nền tảng. Chi tiết về Flutter Framework sẽ được trình bày ở phần tiếp theo.

2.2. Cơ sở lý thuyết

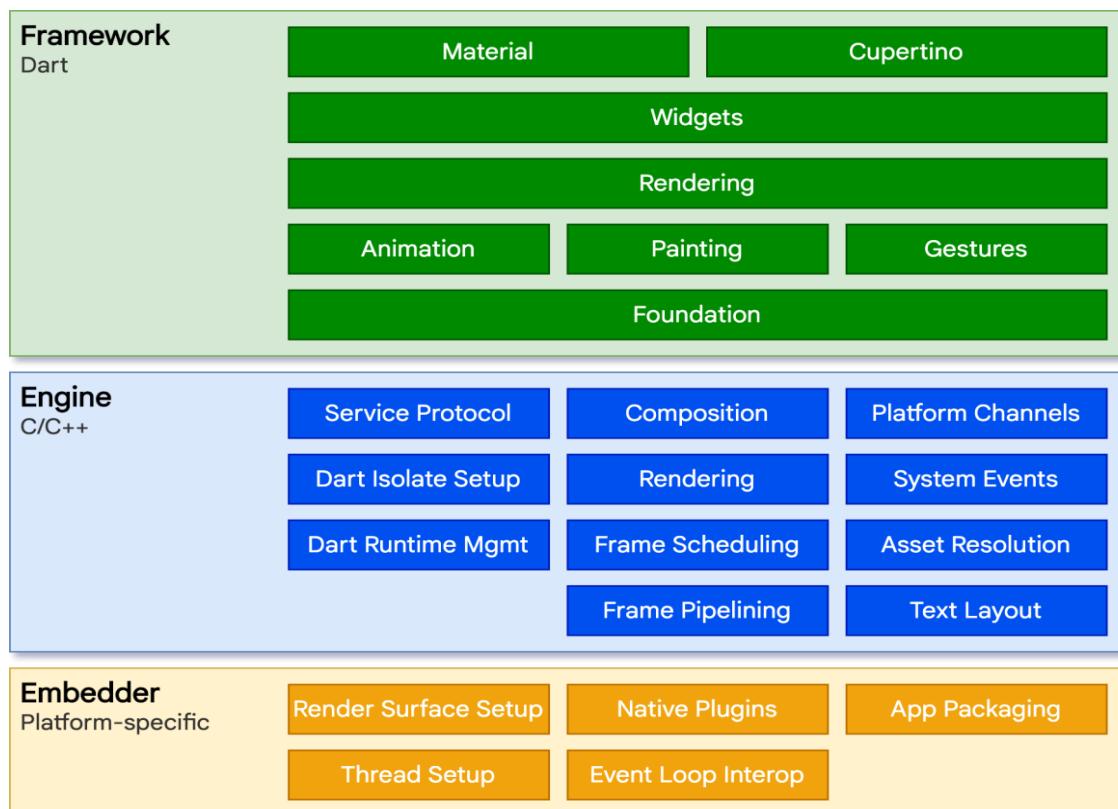
2.2.1. Flutter Framework

Flutter là một framework phát triển ứng dụng di động đa nền tảng mã nguồn mở mạnh mẽ được phát triển bởi Google, Flutter đặc biệt phù hợp cho việc xây dựng các ứng dụng di động đẹp và hiệu quả với hiệu năng cao trên nhiều hệ điều hành, đặc biệt là Android và iOS.

Dart là ngôn ngữ lập trình được sử dụng trong việc phát triển ứng dụng Flutter. Dart được phát triển bởi Google và nổi tiếng với việc biên dịch mã nguồn thành mã máy trước khi chạy (AOT compilation).

Điều này giúp cải thiện hiệu năng so với một số ngôn ngữ khác thường sử dụng biên dịch Just-In-Time (JIT).

2.2.1.1. Kiến trúc tổng quan



Hình 2.2.1. Tổng quan kiến trúc flutter (flutter.dev)

Kiến trúc Flutter (Flutter Architecture) Đây là mô hình cấu trúc chính của Flutter, bao gồm các phần chính: Flutter Framework, Flutter Engine, và Flutter Embedded

- **Lớp Framework:**
 - Cung cấp một API mức độ cao để xây dựng ứng dụng chất lượng.
Ví dụ: Các thành phần giao diện, kiểm tra va chạm, phát hiện cử chỉ, khả năng truy cập, đầu vào văn bản.
 - Tạo cấu trúc cây Widget của ứng dụng.
- **Lớp Engine:**
 - Chịu trách nhiệm kết xuất hình ảnh cho ứng dụng.
 - Cung cấp triển khai mức độ thấp cho các API lõi của Flutter.
Ví dụ: Đồ họa, định dạng văn bản, Dart runtime.
 - Hiện thị chức năng của mình cho lớp Framework qua dart:ui API.
- **Lớp Embedder:**
 - Tương tác với hệ điều hành cơ bản để truy cập các dịch vụ như bề mặt hiển thị, khả năng truy cập, đầu vào.
 - Quản lý vòng lặp sự kiện.
 - Hiện thị platform-specific API để tích hợp Embedder vào các ứng dụng.

2.2.1.2. Widget và Cây Widget

Widget là thành phần cơ bản để xây dựng giao diện người dùng trong Flutter. Một widget có thể là một phần giao diện nhỏ, như một nút bấm hoặc một văn bản hoặc có thể là một thành phần giao diện lớn hơn như một màn hình hoặc một ứng dụng hoàn chỉnh. Flutter chia widget thành hai loại chính:

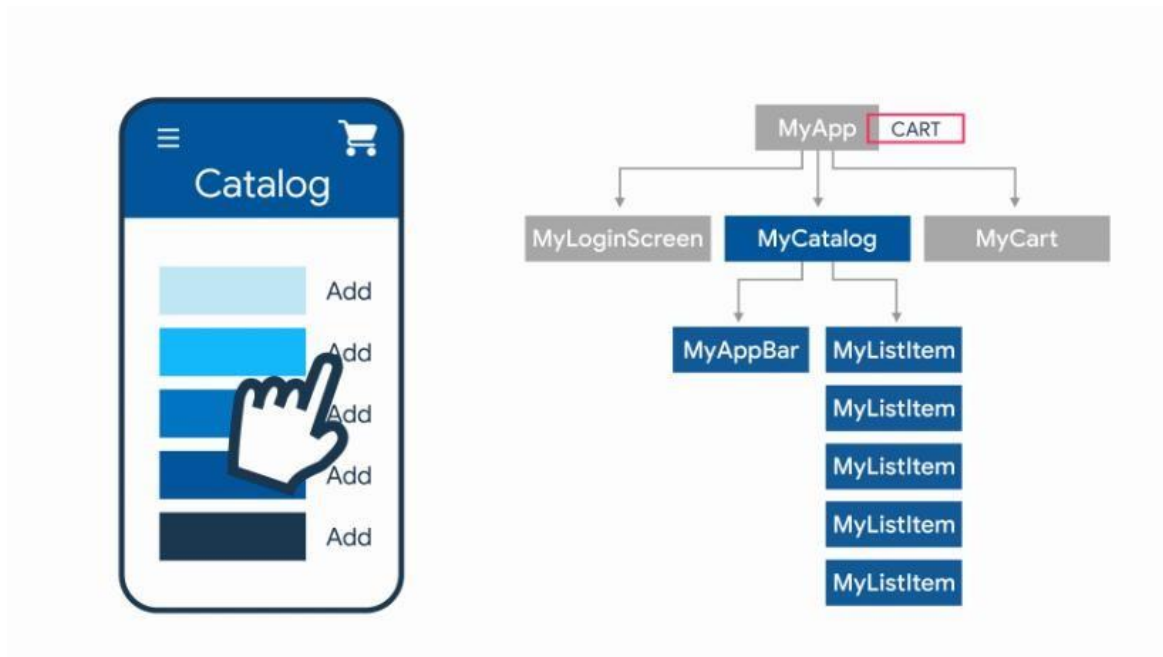
- **Stateless Widget:** là loại không thay đổi trạng thái sau khi được tạo. Chúng thường được sử dụng để hiển thị thông tin tĩnh hoặc không cần tương tác người dùng.

Ví dụ: Một Stateless Widget có thể hiển thị văn bản thông tin cố định.

- **Stateful Widget:** là loại có khả năng thay đổi trạng thái sau mỗi lần cập nhật. Chúng được sử dụng cho các thành phần có tính tương tác.

Ví dụ: Một nút bấm có thể thay đổi trạng thái sau mỗi lần nhấn.

Cây Widget là cấu trúc phân cấp chứa tất cả các widget trong ứng dụng được bắt đầu từ một widget gốc (root widget) và mở rộng xuống các widget con. Mỗi widget chịu trách nhiệm hiển thị một phần giao diện và có thể chứa các widget con khác. Khi trạng thái thay đổi, Flutter cập nhật giao diện người dùng dựa trên cây widget, giúp ứng dụng phản hồi linh hoạt và hiệu quả trước các thay đổi và tương tác của người dùng.



Hình 2.2.2. Một cây widget đơn giản. (flutter.dev)

2.2.2. Quản lý trạng thái

Quản lý trạng thái (State Management) là một phần quan trọng trong phát triển ứng dụng di động Flutter. Trạng thái (state) đại diện cho dữ liệu mà ứng dụng cần theo dõi và cập nhật để giao diện người dùng phản ánh đúng tình trạng hiện tại. Flutter cung cấp nhiều cách quản lý trạng thái, từ phương pháp đơn giản như `setState()` đến các thư viện phức tạp như Provider, Riverpod, Redux, và Bloc.

$$\text{UI} = f(\text{state})$$

The layout on the screen Your build methods The application state

Biểu thức " $\text{UI} = f(\text{state})$ " cho thấy giao diện người dùng phụ thuộc vào trạng thái. Khi trạng thái thay đổi thì giao diện cũng sẽ tự động cập nhật.

Trong đề án này thư viện Riverpod được chọn để quản lý trạng thái. Bởi vì nó cung cấp các tính năng mạnh như không cần BuildContext, tích hợp linh hoạt và tối ưu hóa tái tạo widget.

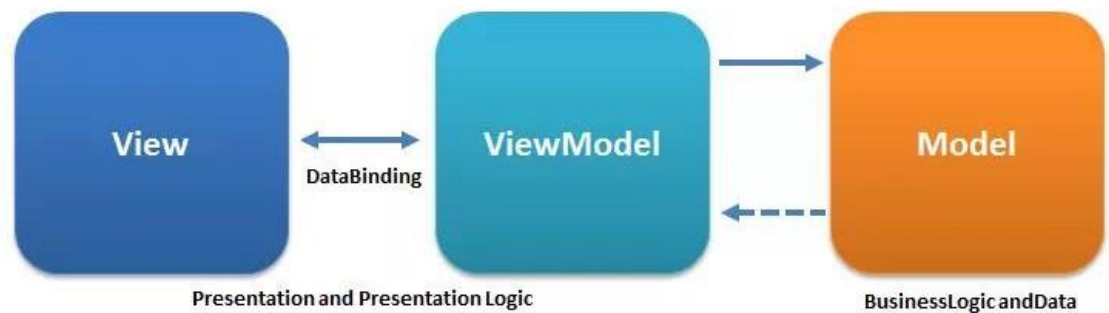
2.2.2.1. Riverpod là gì ?

Riverpod là một thư viện quản lý trạng thái phát triển bởi Remi Rousselet, giúp quản lý trạng thái hiệu quả cho ứng dụng Flutter. Một số ưu điểm của Riverpod:

- Không cần BuildContext: Tránh việc phải truyền context qua nhiều lớp widget.
- Thiết lập độc lập: Các đối tượng Provider có thể được thiết lập và tích hợp linh hoạt.
- Tối ưu hóa tái tạo: Riverpod sử dụng cơ chế AutoDispose, tái tạo widget chỉ khi cần thiết, giúp tăng hiệu suất.

2.2.2.2. Mô hình MVVM là gì ?

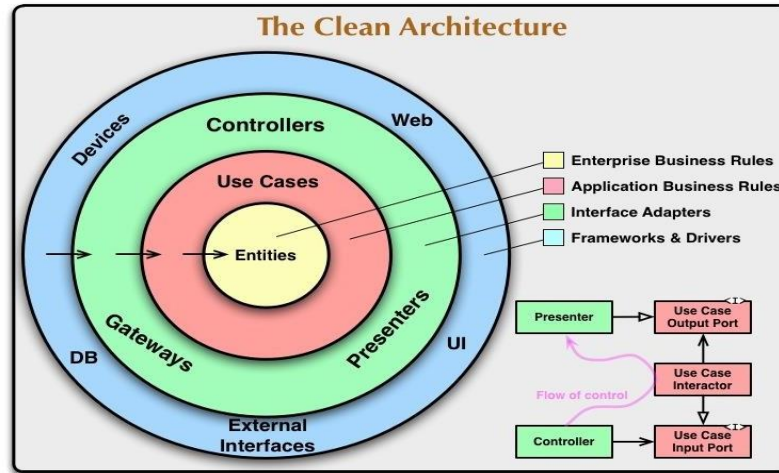
MVVM (Model-View-ViewModel) là một mô hình giúp tách biệt dữ liệu, giao diện và logic. Các thành phần chính:



- View: Giao diện người dùng, hiển thị thông tin.
- Model: Dữ liệu và logic liên quan đến dữ liệu.
- ViewModel: Cầu nối giữa Model-View, xử lý logic và cung cấp dữ liệu cho View.

2.2.3. Kiến trúc sạch (Clean Architecture)

Kiến trúc sạch (Clean Architecture) là mô hình phát triển phần mềm phân chia ứng dụng thành các lớp độc lập, giúp dễ bảo trì, kiểm tra và mở rộng. Kiến trúc này tập trung vào việc tách biệt các thành phần và giảm phụ thuộc vào framework.



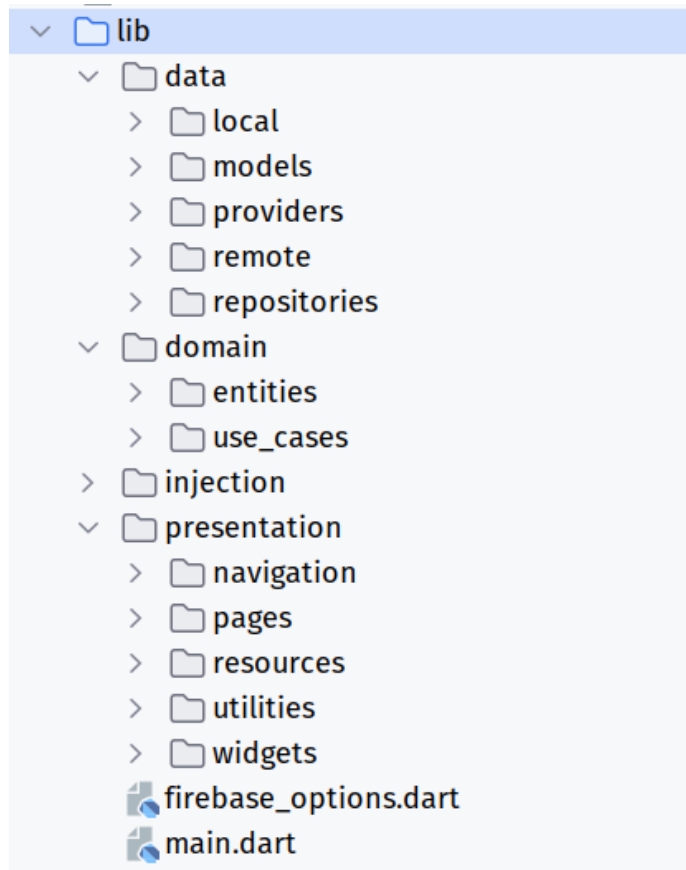
Hình 2.2.3-1. Minh họa kiến trúc sạch (flutter.dev)

Trong Flutter, kiến trúc sạch kết hợp giữa nguyên lý "độc lập" và cách Flutter xây dựng ứng dụng, giúp tái sử dụng, kiểm tra và mở rộng dễ dàng. Các thành phần chính trong kiến trúc sạch:

- Entities (Thực thể): Định nghĩa dữ liệu và logic kinh doanh, không phụ thuộc vào framework.
- Use Cases (Trường hợp sử dụng): Triển khai logic kinh doanh và tương tác với các thực thể.
- Repositories (Kho dữ liệu): Định nghĩa giao diện truy cập dữ liệu từ các nguồn khác nhau.

Các lớp trong kiến trúc sạch của Flutter:

- Presentation Layer: Xây dựng giao diện người dùng, tương tác với Use Cases, không chứa logic kinh doanh.
- Data Layer: Triển khai Repository để truy xuất và xử lý dữ liệu.
- Domain Layer: Chứa thực thể, Use Cases và giao diện Repository, triển khai logic kinh doanh và các đối tượng cốt lõi.



Hình 2.2.3-2. Áp dụng kiến trúc sạch vào đồ án

2.2.4. Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web cung cấp một loạt các dịch vụ và công cụ, bao gồm các API đơn giản và mạnh mẽ mà không cần backend hay server, giúp các nhà phát triển xây dựng ứng dụng một cách nhanh chóng, hiệu quả và dễ dàng quản lý.

Một số thành phần của Firebase được sử dụng trong đồ án:

- Cloud Firestore:
 - Cloud Firestore là một cơ sở dữ liệu NoSQL dựa trên tài liệu được cung cấp bởi Firebase, giúp bạn lưu trữ và truy xuất dữ liệu của ứng dụng một cách dễ dàng và hiệu quả.
 - Firestore sử dụng mô hình dữ liệu tài liệu dựa trên các tài liệu và bộ sưu tập (collections). Firestore hỗ trợ trường dữ liệu đa dạng.

- **Firebase Authentication:**
 - Firebase Authentication là một dịch vụ quản lý xác thực người dùng mạnh mẽ và đa dạng của Firebase.
 - Firebase Authentication cho phép người dùng đăng nhập vào ứng dụng của bạn bằng nhiều phương thức khác nhau, ví dụ như: qua email và mật khẩu, qua số điện thoại, qua các tài khoản mạng xã hội, xác thực đám mây.

2.2.5. OpenAI API

ChatGPT là một mô hình ngôn ngữ mạnh mẽ được phát triển bởi OpenAI, một công ty nổi tiếng trong lĩnh vực trí tuệ nhân tạo. Sử dụng kiến trúc GPT-4o, ChatGPT có khả năng xử lý và tạo ra văn bản với độ tự nhiên cao, giúp tương tác giữa người với máy trở nên linh hoạt và hiệu quả.

OpenAI API là một dịch vụ của OpenAI cho phép các nhà phát triển tích hợp mô hình ChatGPT vào ứng dụng và dịch vụ của họ. Bằng cách sử dụng OpenAI API, bạn có thể tích hợp khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ của ChatGPT vào ứng dụng của mình, từ chatbot đến các ứng dụng thông minh khác. Điều này mang lại trải nghiệm tương tác người-máy đa dạng và tiên tiến cho người dùng.

CHƯƠNG 3. TRIỂN KHAI ỨNG DỤNG

3.1. Lựa chọn công nghệ

Có rất nhiều lựa chọn tốt để làm nền tảng phát triển cho ứng dụng di động, và Flutter là một trong số đó. Sau đây là một số lý do để chọn Flutter cho đồ án:

- Phát triển đa nền tảng: Với mục tiêu cung cấp ứng dụng học tiếng Anh cho cả người dùng iOS và Android, việc sử dụng Flutter cho phép phát triển đa nền tảng một cách hiệu quả. Mã nguồn duy nhất có thể được chia sẻ giữa cả hai nền tảng, giúp tiết kiệm thời gian và nguồn lực phát triển.
- Tích hợp dễ dàng: Flutter có tích hợp dễ dàng với các dịch vụ của Google Cloud, Firebase, Microsoft Azure,... và các dịch vụ học tập, từ điển, giúp dễ dàng trong quá trình phát triển.
- Sự hỗ trợ từ cộng đồng: Flutter có một cộng đồng lớn và nhiệt huyết, giúp dễ dàng tìm kiếm hỗ trợ, tài liệu, và giải pháp cho các vấn đề phát triển.

Một số phần mềm công cụ được sử dụng trong quá trình phát triển:

- IDE: Android Studio, Xcode
- Postman: Dùng để kiểm tra các API.
- Git (GitHub, SourceTree...): Quản lý phiên bản mã nguồn.
- Thiết kế giao diện: Figma, Adobe Photoshop.
- Thiết kế, quản lý cơ sở dữ liệu: dbdiagram.io, Firebase.
- Thiết bị thật Android (Oppo) và Máy ảo (Android Emulator, iOS Simulator),...

3.2. Thiết kế cơ sở dữ liệu

3.2.1. Cơ sở dữ liệu cục bộ (SQLite)

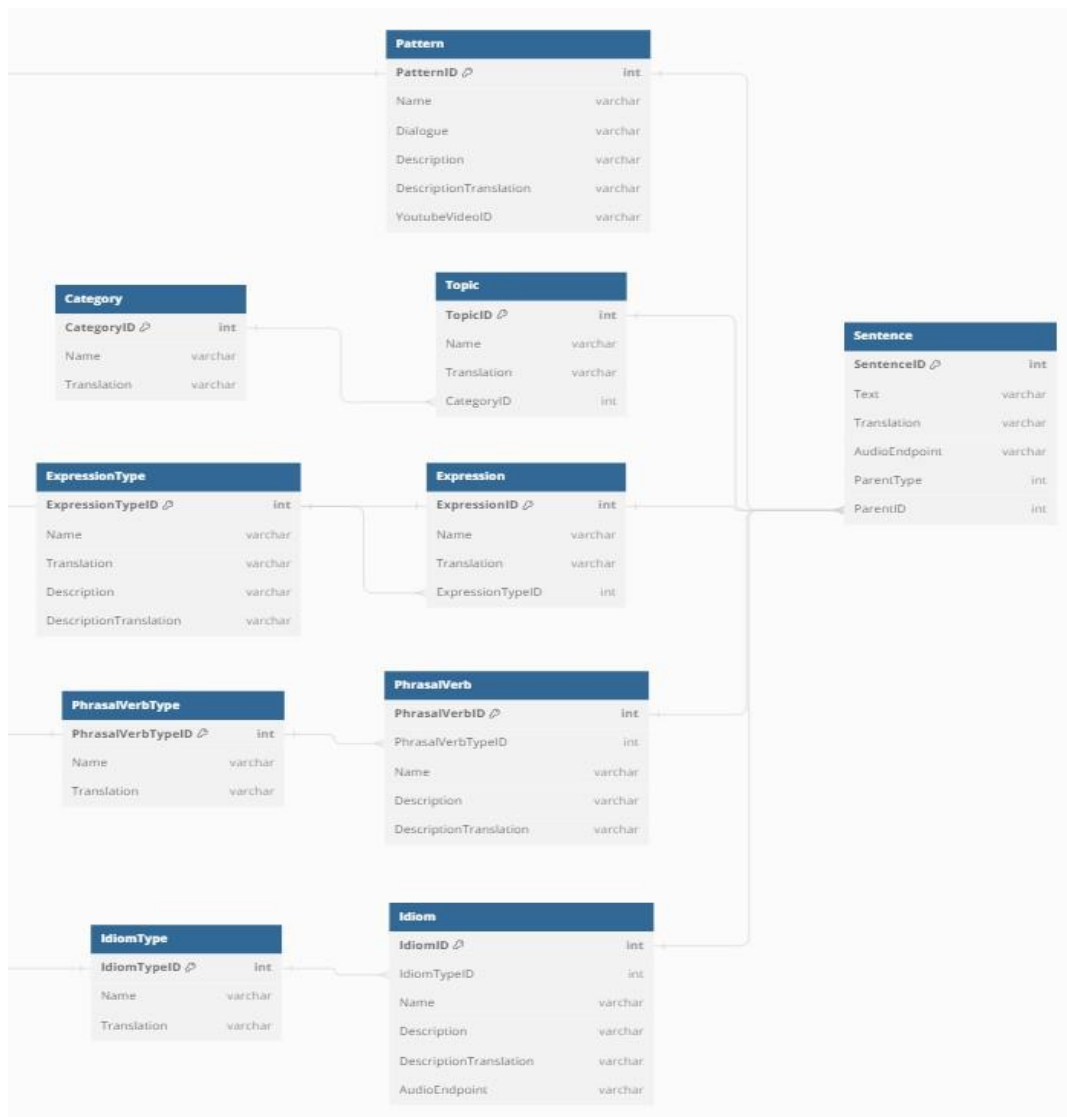
SQLite là hệ thống quản lý cơ sở dữ liệu (DBMS hay là Database Management System) được tích hợp sẵn trong Flutter, cho phép lưu trữ dữ liệu cục bộ, giúp ứng dụng hoạt động nhanh chóng và ngoại tuyến. Điều này rất quan trọng cho các ứng dụng cần hoạt động ổn định mà không phụ thuộc vào mạng.

Dữ liệu các bài học tiếng Anh được thu thập từ “*basicenglishspeaking.com*” và bao gồm các mục như:

- 12 thì trong tiếng Anh
- 102 thành ngữ với ví dụ
- 75 đoạn hội thoại theo chủ đề
- 181 cụm động từ và câu ví dụ
- 100 cụm từ và mẫu câu thông dụng
- 50 cụm từ và câu sử dụng hàng ngày

Dữ liệu sẽ được tổ chức và thiết kế thành các bảng trong cơ sở dữ liệu. Vì dữ liệu bài học không thay đổi thường xuyên, việc lưu trữ cục bộ giúp tiết kiệm dữ liệu và cải thiện hiệu suất ứng dụng, chỉ cập nhật khi có phiên bản mới.

Dưới đây là ảnh Cơ sở dữ liệu cục bộ của đồ án.

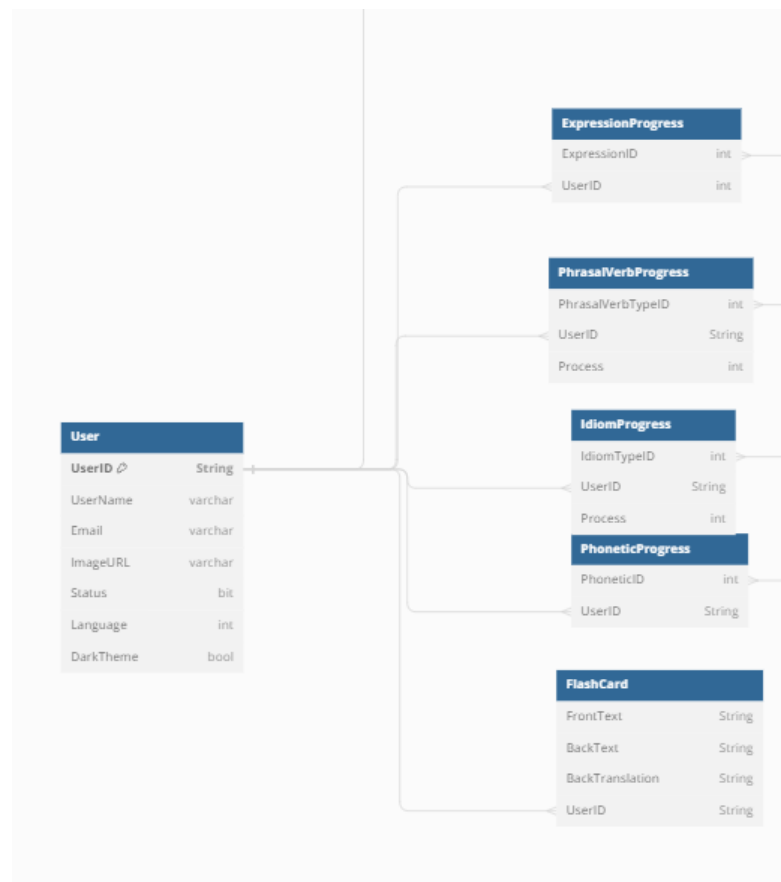


Hình 3.2.1. Cơ sở dữ liệu cục bộ (SQLite)

3.2.2. Cơ sở dữ liệu đám mây (Cloud Firestore)

Cloud Firestore là dịch vụ cơ sở dữ liệu NoSQL của Google Cloud Platform được tích hợp với Flutter và cho phép đồng bộ cơ sở dữ liệu thời gian thực. Nó giúp ứng dụng nhận cập nhật ngay khi dữ liệu thay đổi, phù hợp với các tính năng như trò chơi đa người, trò chuyện và đồng bộ dữ liệu nhanh.

Trong đồ án này sử dụng Cloud Firestore lưu trữ thông tin tiến trình học, danh sách flashcard, tài khoản người dùng, tin nhắn với chatbot, và danh sách video dạy tiếng Anh. Dữ liệu sẽ không bị mất khi người dùng gỡ cài đặt ứng dụng và dễ cập nhật.



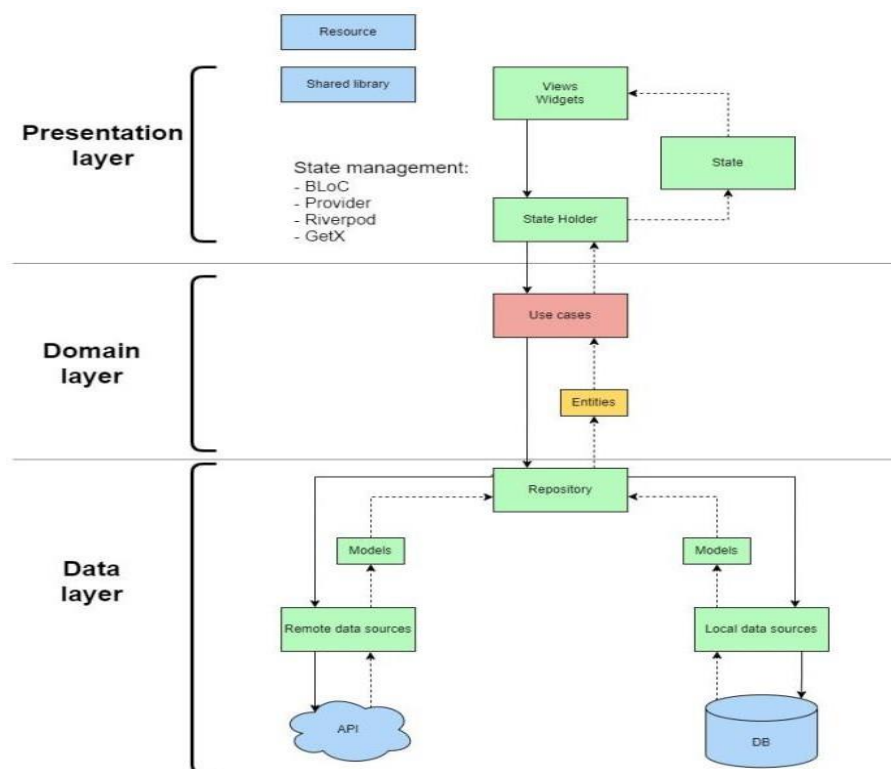
Hình 3.2.2. Cơ sở dữ liệu đám mây (Cloud Firestore)

3.3. Kiến trúc và luồng dữ liệu

Như đã trình bày trong chương 2, ứng dụng trong đồ án này đã áp dụng Clean Architecture gồm 3 lớp chính là Data, Domain Và Presentation.

Các phần sau đây sẽ trình bày chi tiết các thành phần của mỗi lớp, cách dữ liệu được xử lý, và luồng dữ liệu di chuyển qua các lớp(cách các lớp giao tiếp với nhau) trong ứng dụng của đồ án này.

Hình 3.3.1. Kiến trúc và luồng dữ liệu



3.3.1. Lớp Data

3.3.1.1. Nguồn dữ liệu (Data source)

Nguồn dữ liệu được chia thành 2 loại:

- Remote Data Source là nơi ứng dụng tương tác với dữ liệu từ các nguồn bên ngoài, các dịch vụ như Firestore hay Youtube API. Đây là nơi xử lý việc truy vấn dữ liệu từ các nguồn này và chuyển đổi nó thành định dạng phù hợp cho ứng dụng.
- Local Data Source là nơi ứng dụng tương tác với dữ liệu cục bộ, bao gồm SQLite và Shared Preferences.

Lớp chứa các phương thức giao tiếp với Azure Speech to Text API:

```
@RestApi(  
    baseUrl: 'https://southeastasia.stt.speech.microsoft.com'  
    '/speech/recognition/conversation/cognitiveservices/v1/')  
abstract class AzureSpeechClient {  
    factory AzureSpeechClient(Dio dio, {String baseUrl}) =  
    _AzureSpeechClient;  
  
    @POST('')  
    Future<AzureSpeechResponse> getPronunciationAssessment(  
        @Header('Pronunciation-Assessment') String  
        pronunciationAssessment, @Body() Uint8List body,  
    );  
}
```

3.3.1.2. Repository

Repository là nơi điều phối giữa các nguồn dữ liệu khác nhau và cung cấp một giao diện thống nhất cho tầng Domain để tương tác với dữ liệu. Điều này cho phép tầng Domain làm việc với dữ liệu mà không quan tâm đến nguồn dữ liệu cụ thể (Local hay Remote).

Một lớp repository chứa các phương thức xử lý về Speech to Text:

```
class SpeechToTextRepository {
    final AzureSpeechClient _azureSpeechClient;

    SpeechToTextRepository(this._azureSpeechClient);

    Future<SpeechSentence>
        getPronunciationAssessment( String
            pronunciationAssessment,
            Uint8List body,
        ) async {
        final response = await
            _azureSpeechClient.getPronunciationAssessment(
                pronunciationAssessment,
                body,
            );
        return response.nBest.first;
    }
}
```

3.3.2. Lớp Domain

Lớp domain là trung tâm của kiến trúc Clean Architecture, nơi định nghĩa các quy tắc kinh doanh và luật logic cốt lõi của ứng dụng. Lớp domain gồm 2 thành phần chính là các entity và các use case

3.3.2.1. Entities

Entities là một phần của lớp domain và thường được sử dụng để biểu diễn các thực thể quan trọng trong ứng dụng. Đây có thể là các đối tượng hoặc lớp đại diện cho các khái niệm quan trọng mà ứng dụng của bạn đang làm việc.

Dưới đây là 1 lớp định nghĩa entity Word với nhiều thuộc tính như wordID, word, ...

```
class Word {
    final int
    wordID; final
    int phoneticID;
    final String
    word;
    final String
    translation; final
    String pronunciation;

    final Map<String, int> phoneticComponents;

    Word({
        required
        this.wordID,
        required
        this.phoneticID,
        required this.word,
        required
        this.translation,
        required this.pronunciation,
        required
        this.phoneticComponents,
    });
}
```

3.3.2.2. Use Cases

Use Cases là tầng logic của ứng dụng và chứa các trường hợp sử dụng cụ thể cho việc xử lý kịp thời và thao tác với Entities. Mỗi Use Case thực hiện một nhiệm vụ cụ thể và biểu diễn một chức năng hoặc tính năng cụ thể của ứng dụng. Use Cases không chứa logic giao diện người dùng hoặc logic truy cập dữ liệu cụ thể, chúng chỉ xác định cách sử dụng Entities để thực hiện một nhiệm vụ cụ thể.

Dưới đây là một lớp định nghĩa Use case đăng nhập bằng tài khoản Google:

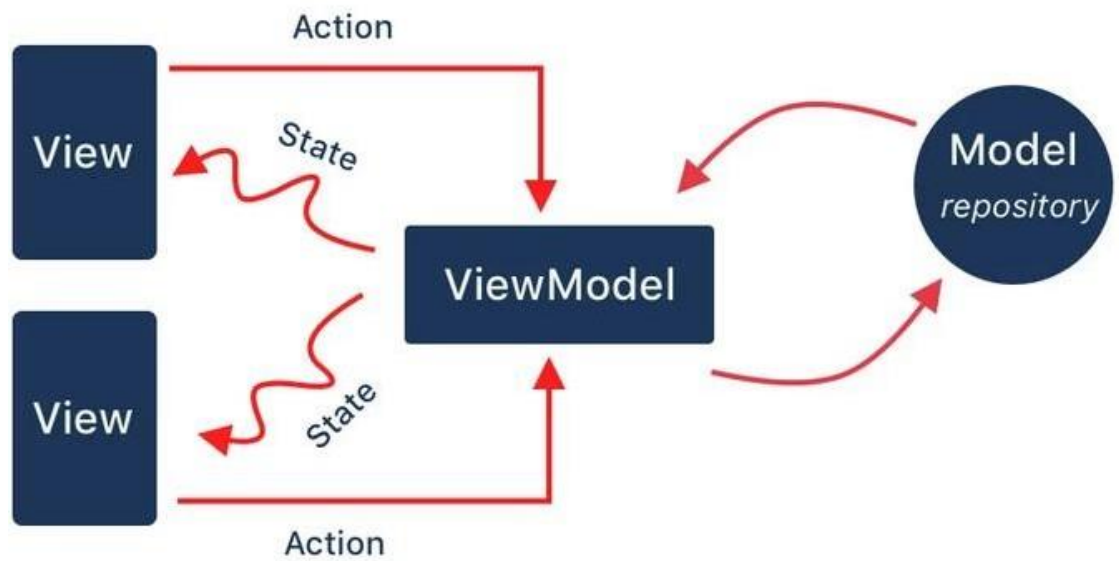
```
class SignInWithGoogleUseCase implements
FutureOutputUseCase<UserCredential> {
    SignInWithGoogleUseCase
    (); @override
    Future<UserCredential> run() async {
        return await
    injector.get<AuthenticationRepository>().signInWithGoogle();
    }
}
```

3.3.3. Lớp Presentation

Lớp Presentation đóng vai trò quan trọng trong việc hiển thị và quản lý giao diện người dùng. Là nơi triển khai logic giao diện, tương tác với người dùng và đảm bảo rằng dữ liệu từ lớp Domain được hiển thị một cách thích hợp.

- Quản lý trạng thái màn hình

Mô hình MVVM kết hợp với Riverpod (2 khái niệm này đã được trình bày trong phần “Quản lý trạng thái” ở chương 2) được sử dụng để quản lý trạng thái cho mỗi màn hình.



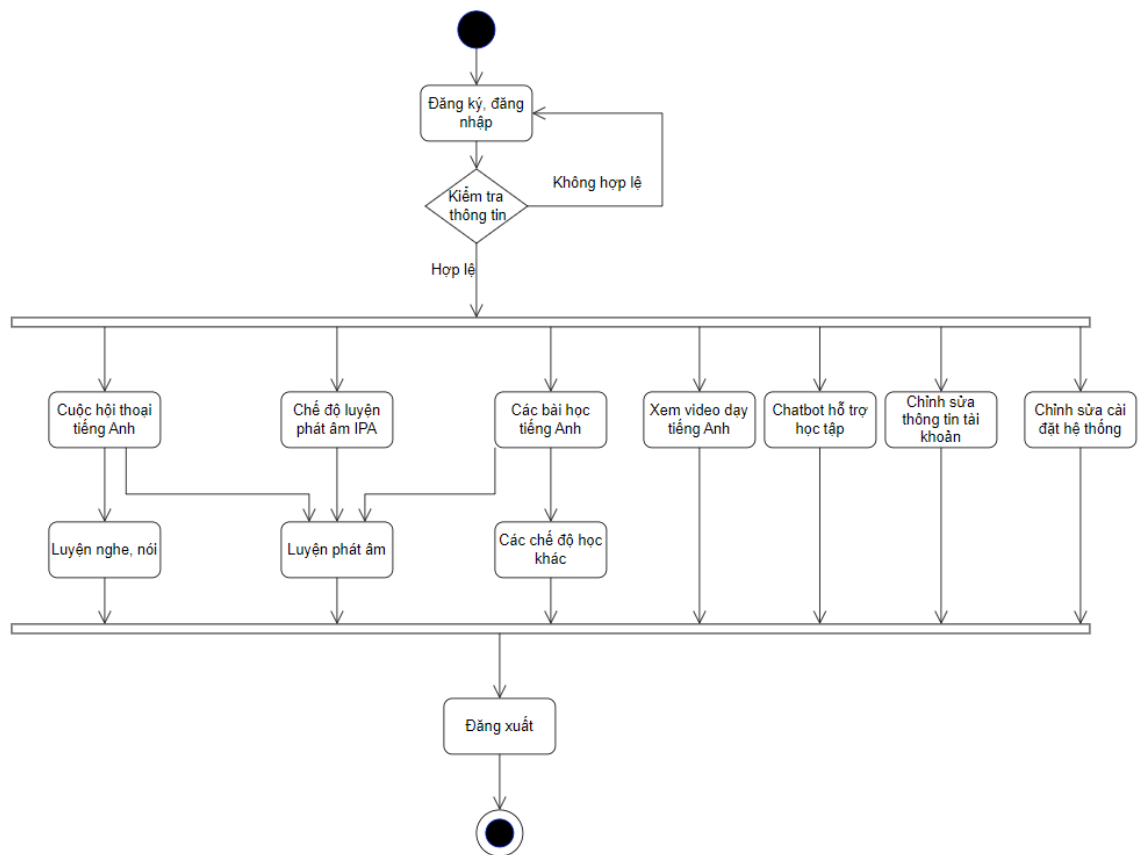
Mỗi màn hình trong ứng dụng sẽ được chia thành 3 tệp: view, view model và state. Tệp view chứa giao diện, tệp state chứa các trạng thái của màn hình như: tình trạng tải, danh sách đối tượng cần hiển thị, ... và tệp view model chứa các phương thức để làm thay đổi state.

Ví dụ: Khi người dùng bấm nút Bắt đầu tải (view), hàm trong viewmodel được gọi, cập nhật lại state. View lắng nghe state, khi state bị thay đổi (từ chưa tải sang bắt đầu tải), view hiển thị loading.

3.4. Biểu đồ mô tả hệ thống

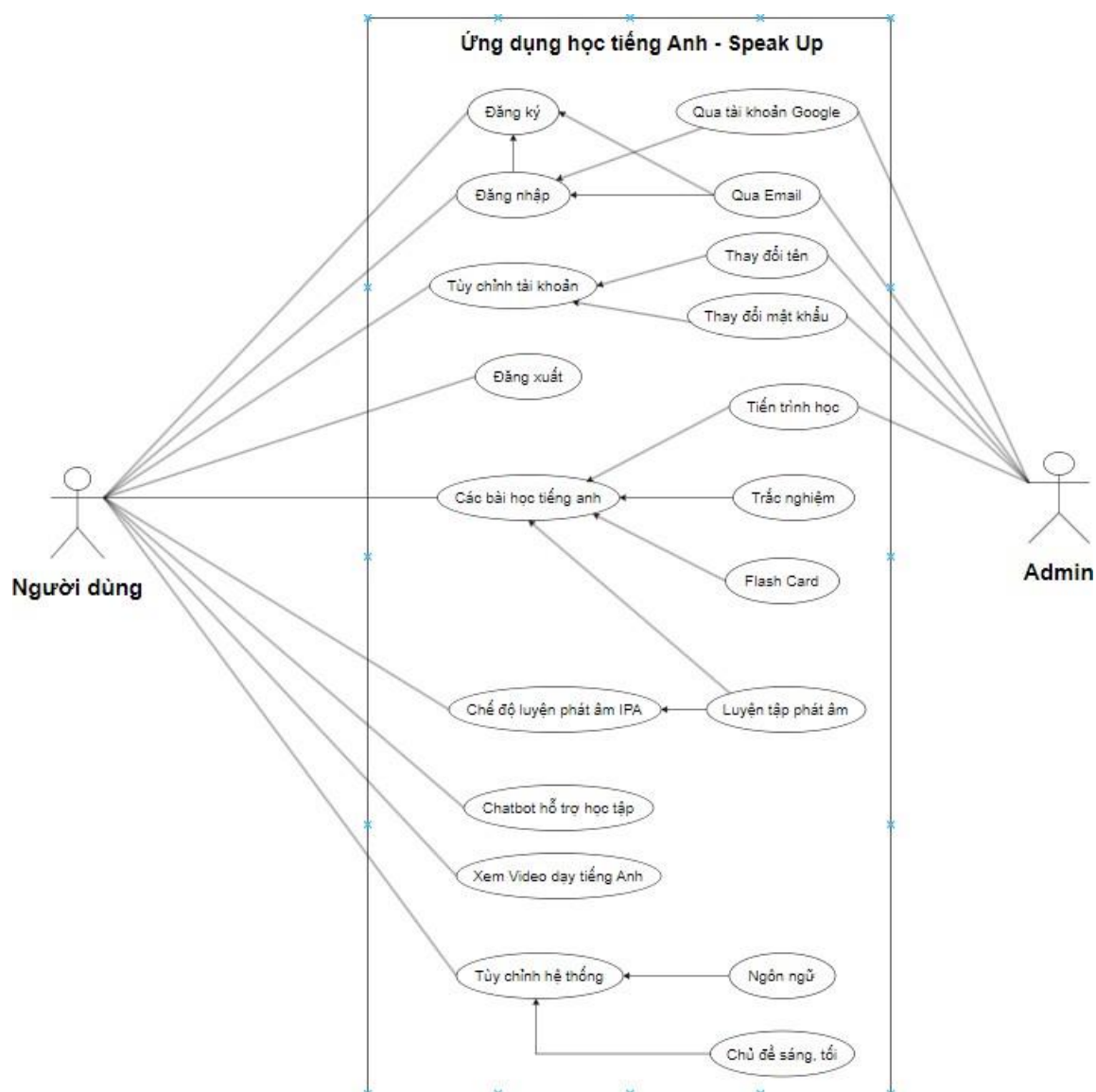
Dưới đây là biểu đồ mô tả của hệ thống

3.4.1. Sequence Diagram



Hình 3.4.1. Sequence Diagram

3.4.2. Usecase Diagram



Hình 3.4.2. Usecase Diagram

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

Phần này trình bày về các chức năng của người dùng và admin đã được tích hợp trong ứng dụng. Chi tiết từng chức năng sẽ được trình bày ở các mục con.

Về cơ bản, ứng dụng gồm 3 phần chức năng chính:

- Các tính năng tài khoản
- Các tính năng học tập
- Các tính năng hệ thống

4.1. Các chức năng của người dùng

Từng tính năng nhỏ của người dùng được nằm trong phần “thanh công cụ dưới” trong ứng dụng. Chi tiết từng tính năng nhỏ sẽ được trình bày ở các mục con.



Hình 4.1. Thanh công cụ dưới

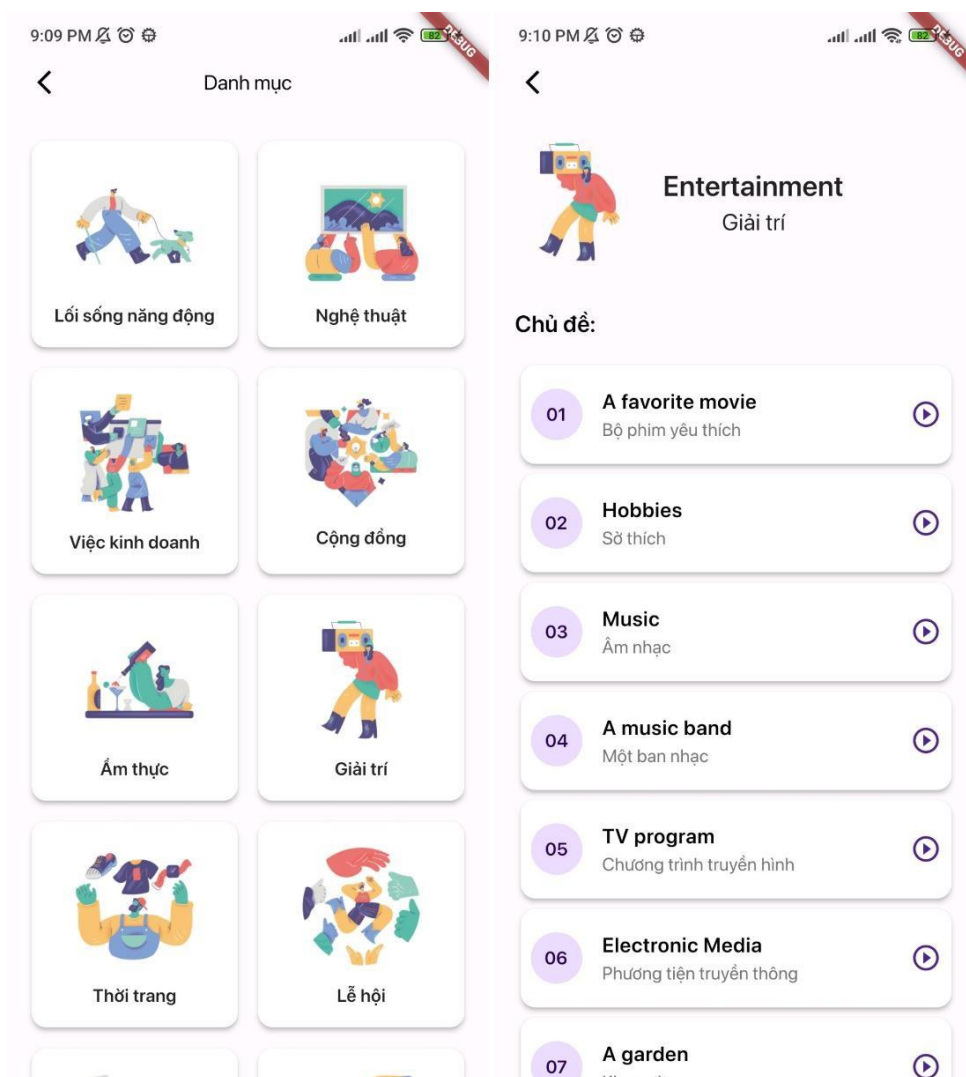
Thanh công cụ bao gồm:

- [1]. Trang chủ
- [2]. Lộ trình học
- [3]. Flashcard & Quiz
- [4]. Học từ vựng
- [5]. Hoạt động tài khoản
- [6]. Chat with AI Assistant

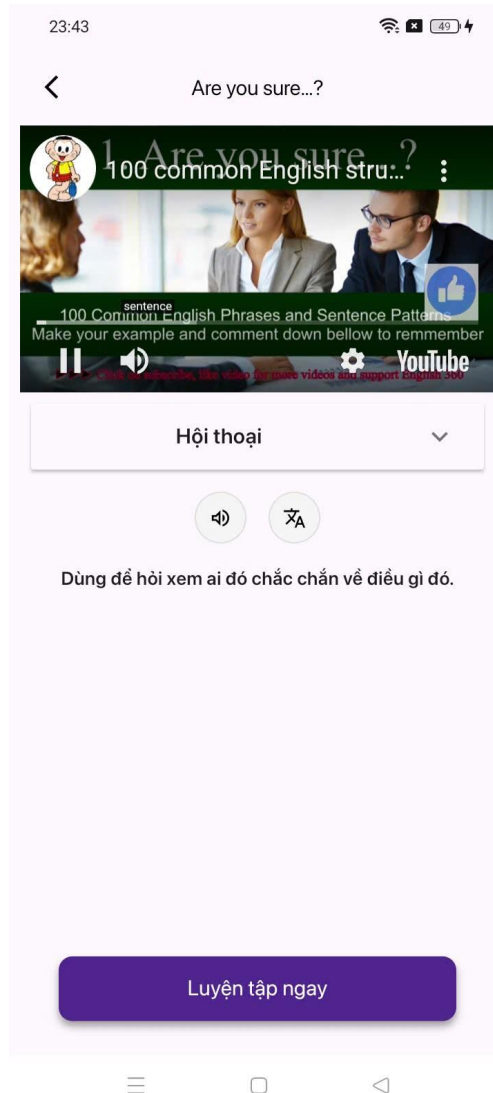
4.1.1. Các cuộc hội thoại theo chủ đề

Tính năng "Cuộc hội thoại theo chủ đề" cho phép người dùng tham gia vào các cuộc hội thoại theo từng chủ đề. Ứng dụng chứa rất nhiều cuộc hội thoại với sự đa dạng chủ đề trong đời sống.

Mỗi chủ đề lớn lại bao gồm nhiều chủ đề con. Các chủ đề này đều là các chủ đề quen thuộc trong cuộc sống, từ đó ứng dụng mang tính thực tiễn cao, giúp người dùng có thể áp dụng vào cuộc sống thường ngày.



Hình 4.1.1-1. Chủ đề cuộc hội thoại lớn và chủ đề con

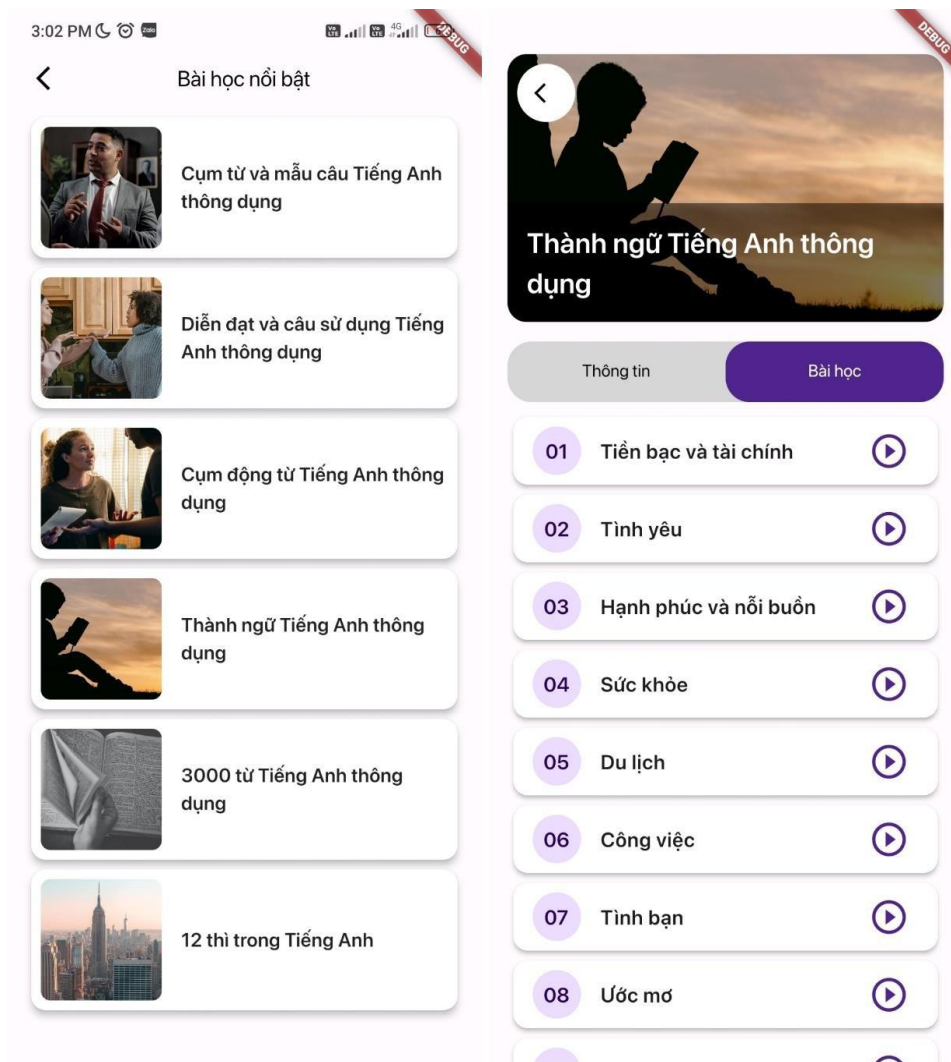


Hình 4.1.1-2. Màn hình luyện tập cuộc hội thoại

4.1.2. Các bài học tiếng Anh

Để giúp người dùng phát triển kiến thức nền tảng trong tiếng Anh, ứng dụng đồ án cung cấp các bài học đa dạng và chất lượng như : 12 thì cơ bản tiếng Anh, 3000 từ thông dụng, mẫu câu, cụm từ, thành ngữ,... thường được sử dụng trong giao tiếp.

Mỗi bài học trong ứng dụng được chia thành nhiều bài học con được phân theo loại hoặc chủ đề. Điều này giúp người dùng nắm được kiến thức cơ bản đến phức tạp hơn và tìm hiểu về các chủ đề họ quan tâm.



Hình 4.1.2-1. Các bài học lớn và bài học con

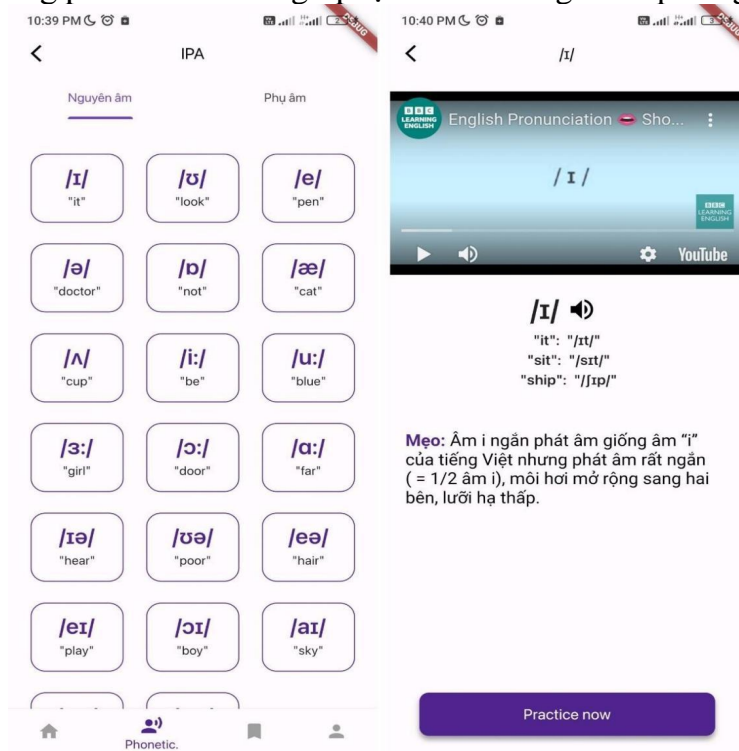
4.1.3. Các chế độ nhỏ

4.1.3.1. Luyện Phát Âm

Tính năng "Luyện Phát Âm" trong ứng dụng là một phần quan trọng của quá trình học tiếng Anh. Đây là nơi mà người dùng có cơ hội rèn luyện và cải thiện khả năng phát âm của họ thông qua việc lặp lại hàng nghìn câu Tiếng Anh.

Bên cạnh đó, ứng dụng còn cung cấp một chế độ luyện phát âm chuyên sâu, theo từng âm trong 44 IPA(International Phonetic Alphabet) đi kèm hướng dẫn chi tiết phát âm của từng âm, giúp người dùng dễ dàng luyện tập

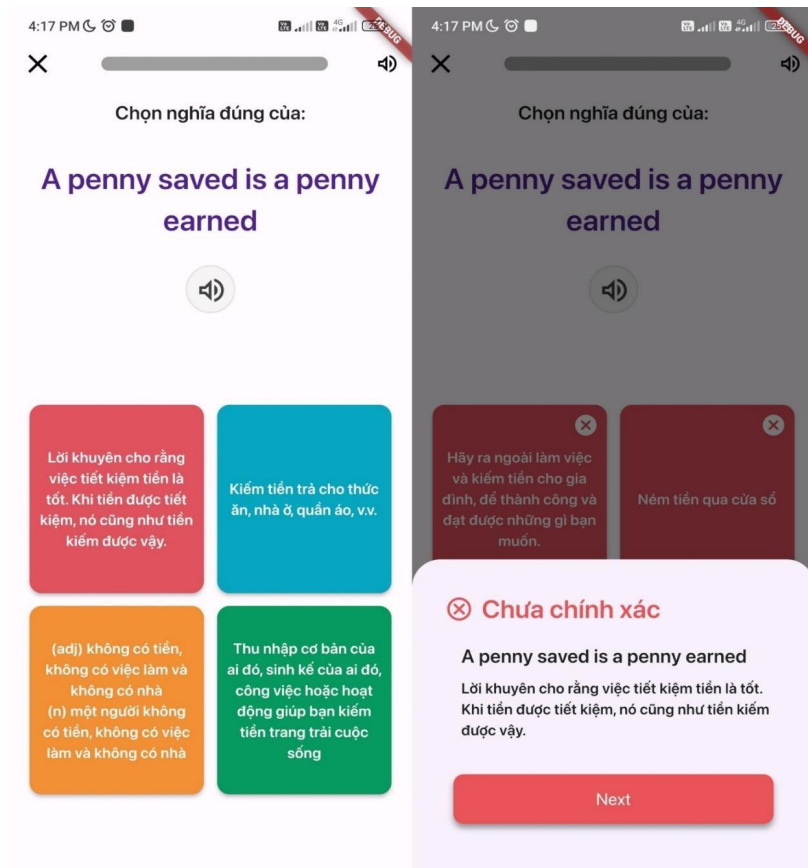
Tính năng "Luyện Phát Âm" không chỉ cung cấp cho người dùng cô giúp rèn luyện khả năng phát âm mà còn giúp tự tin hơn khi giao tiếp bằng tiếng



Hình 4.1.3-1. Bảng IPA chi tiết và cách phát âm

4.1.3.2. Trắc nghiệm Quiz

Tính năng “Trắc nghiệm Quiz” giúp người dùng kiểm tra lại những kiến thức đã được tiếp thu của bản thân. Khi kết thúc ứng dụng sẽ ghi nhận thành tích và thông báo điểm đạt được của người dùng ở cuối mỗi bài trắc nghiệm. Người dùng cần vận dụng kiến thức mình có để chọn định nghĩa, ý hiểu đúng của mỗi cụm từ, mẫu câu hay thành ngữ.

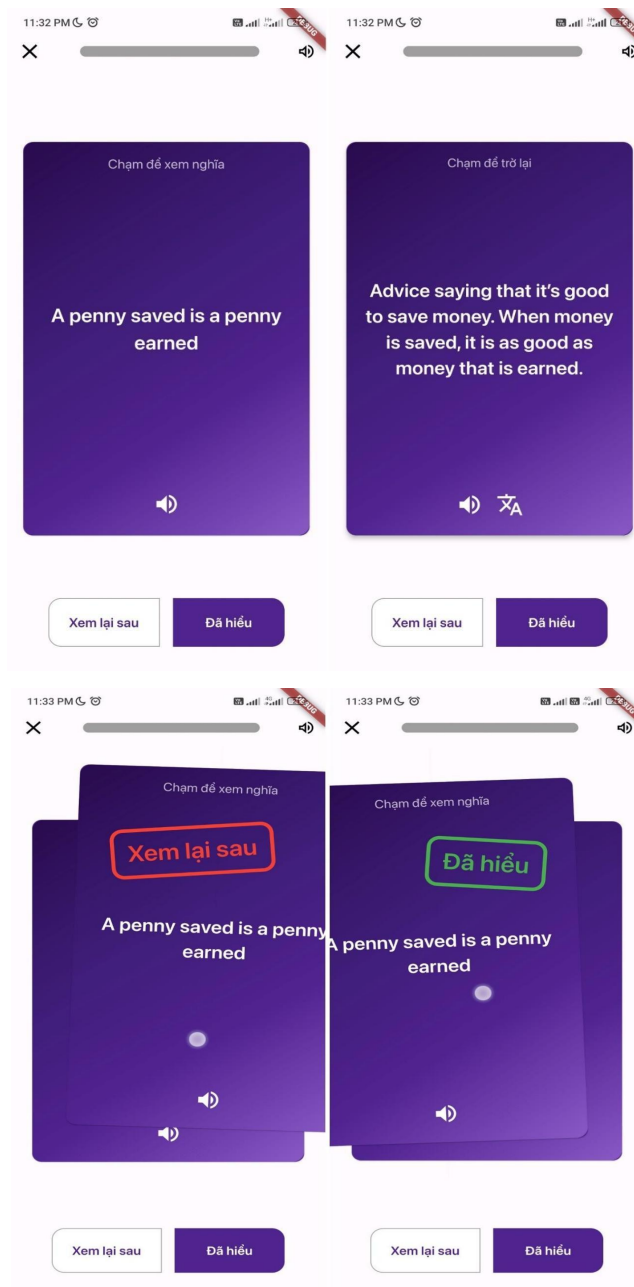


Hình 4.1.3-2. Trắc nghiệm Quiz

4.1.3.3. Flashcard

Flashcard ở ngoài được biết là loại thẻ chứa thông tin trên cả hai mặt giấy, được sử dụng như một công cụ hỗ trợ ghi nhớ trong việc học tập. Mỗi tờ flashcard bao gồm câu hỏi ở mặt trước và câu trả lời ở mặt sau. Mỗi tấm thẻ thường được sử dụng để ghi nhớ từ vựng, ngày tháng lịch sử, công thức hoặc bất kỳ nội dung chủ đề nào có thể học được thông qua hình thức câu hỏi kèm theo câu trả lời. Chính vì thế nhiều người đã sử dụng cách này để học từ vựng Tiếng Anh.

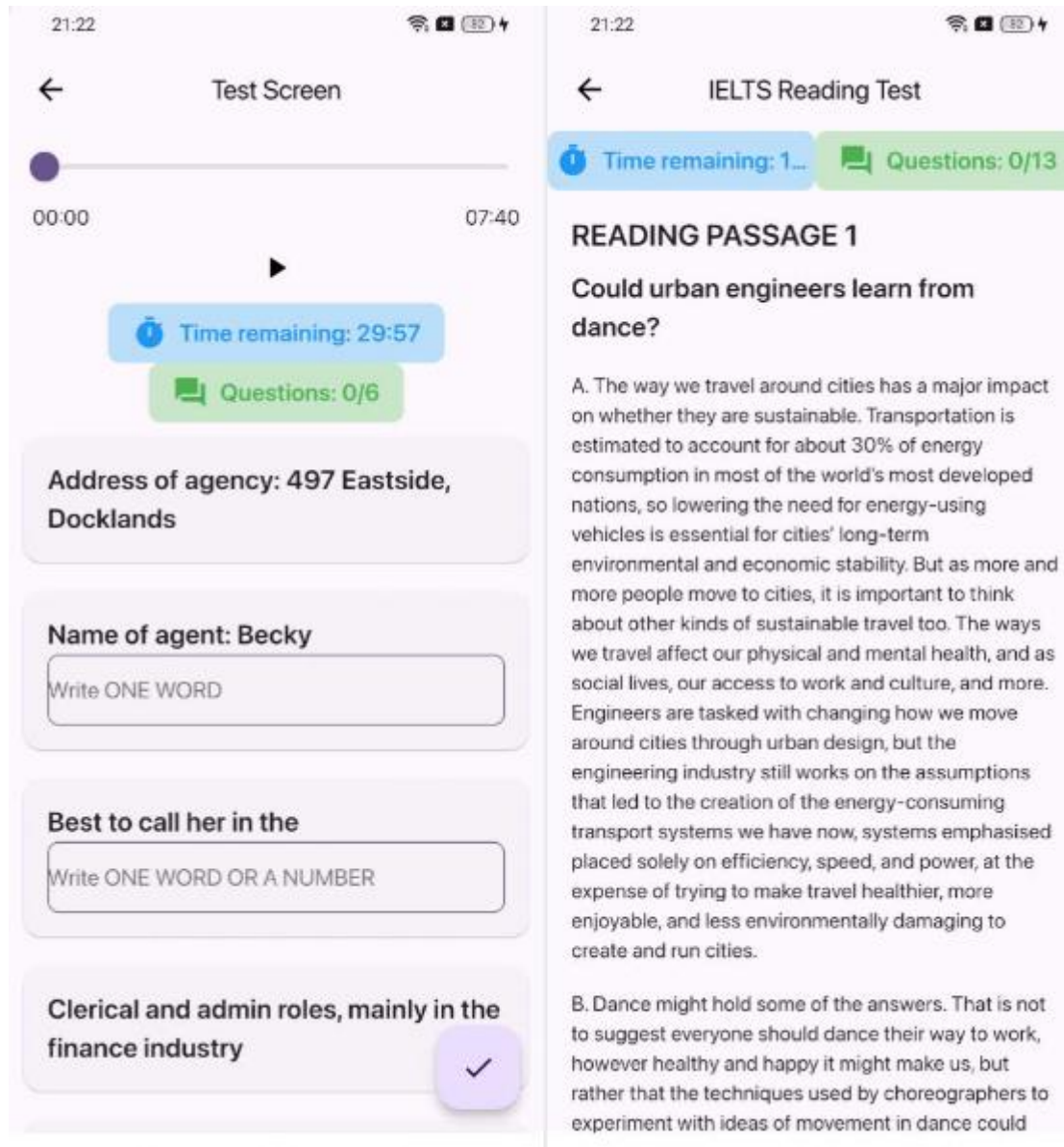
Tính năng “Flashcard” áp dụng trong ứng dụng là một công cụ hữu ích để học từ vựng và cụm từ tiếng Anh một cách hiệu quả. Việc học bằng Flashcard giúp người dùng nắm vững mẫu câu, cụm từ, thành ngữ và định nghĩa, ý nghĩa của nó.



Hình 4.1.3-3. Flashcard

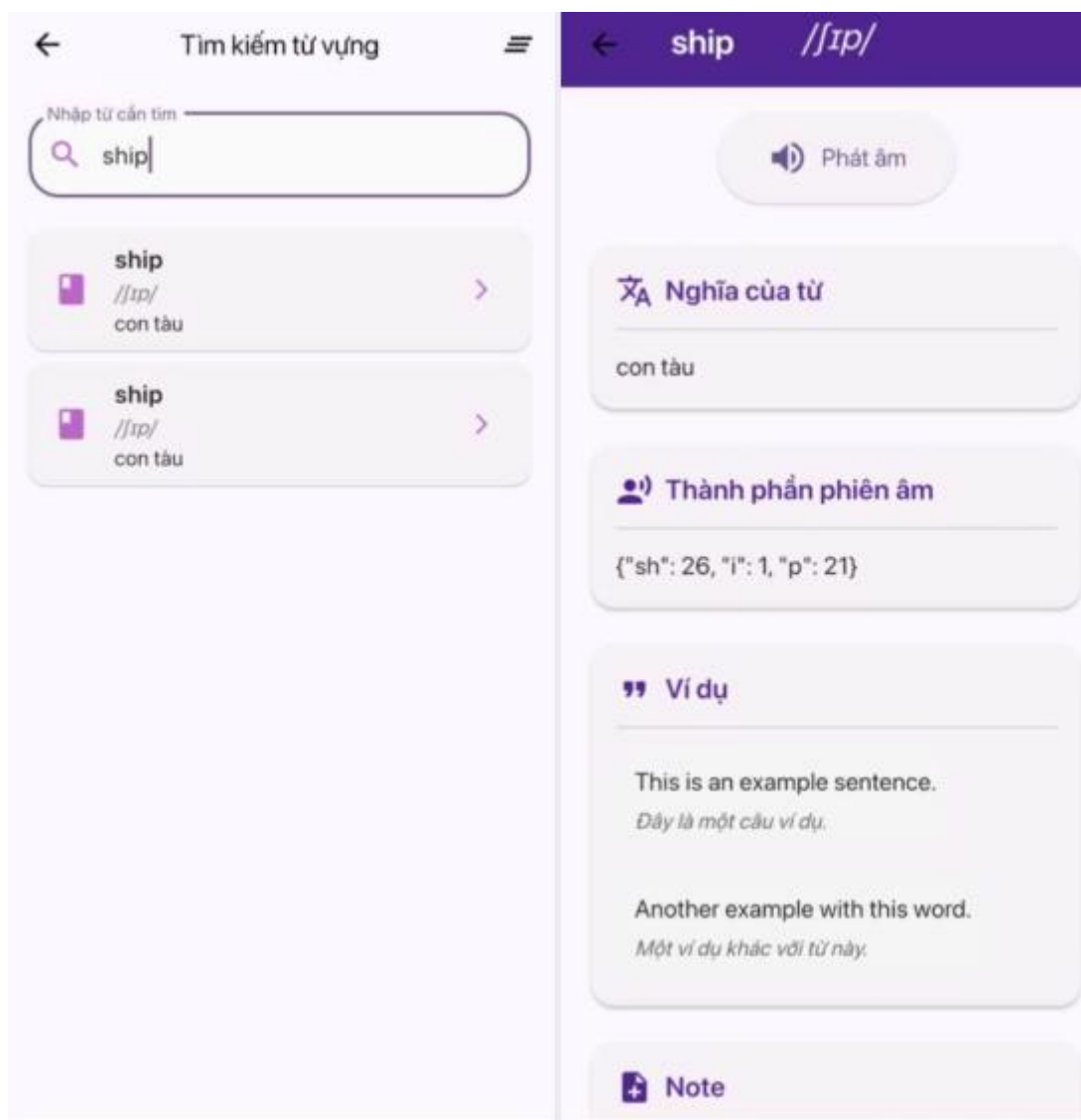
4.1.3.4. Luyện bài thi đọc hiểu

Tính năng “Luyện Reading Test” giúp người dùng làm phần đọc hiểu trong bài thi IELTS. Khi kết thúc ứng dụng sẽ ghi nhận thành tích và thông báo điểm đạt được của người dùng ở cuối mỗi bài. Người dùng cần đọc hiểu và đưa ra câu trả lời ở mục dưới.



Hình 4.1.3-4. Luyện bài thi đọc hiểu

4.1.3.4. Tìm kiếm từ vựng



Hình 4.1.3-5. Tìm kiếm thông tin chi tiết từ vựng

4.1.4. Hoạt động tài khoản

Tính năng “Hoạt động tài khoản” giúp người dùng theo dõi quá trình học tập một cách chi tiết và trực quan.

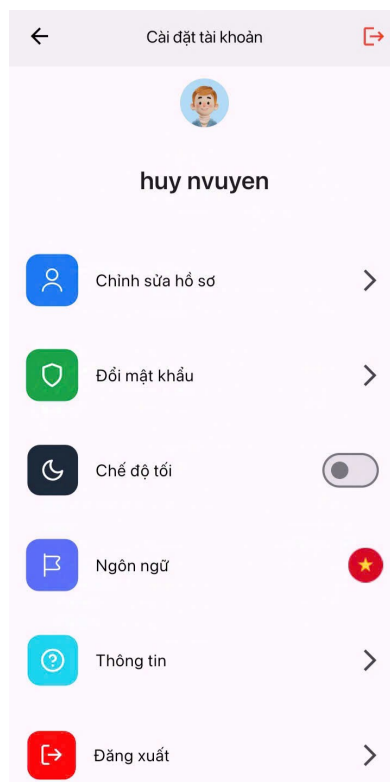


Hình 4.1.4-1. Hoạt động tài khoản

“Hoạt động tài khoản” thể hiện thông tin bao gồm:

- Số ngày duy trì: Cho biết người dùng học liên tục bao nhiêu ngày.
- Thời gian học hôm nay: Hiển thị thông tin tổng thời gian học của người dùng vào ngày hôm đó.
- Study Streak: Là thể hiện chuỗi ngày học liên tiếp của người dùng.
- Hoạt động gần đây: Theo dõi thông tin thời gian mà người dùng đã truy cập vào ứng dụng lần cuối.

4.1.5. Cài đặt tính năng hệ thống



Hình 4.1.5-1. Cài đặt tính năng hệ thống

Công cụ chuyển đổi song ngữ (Anh và Việt): Ứng dụng hỗ trợ song ngôn ngữ, bao gồm tiếng Anh và tiếng Việt. Người dùng có thể chuyển đổi giữa hai ngôn ngữ này để thuận tiện cho việc học tập và hiểu rõ nội dung.

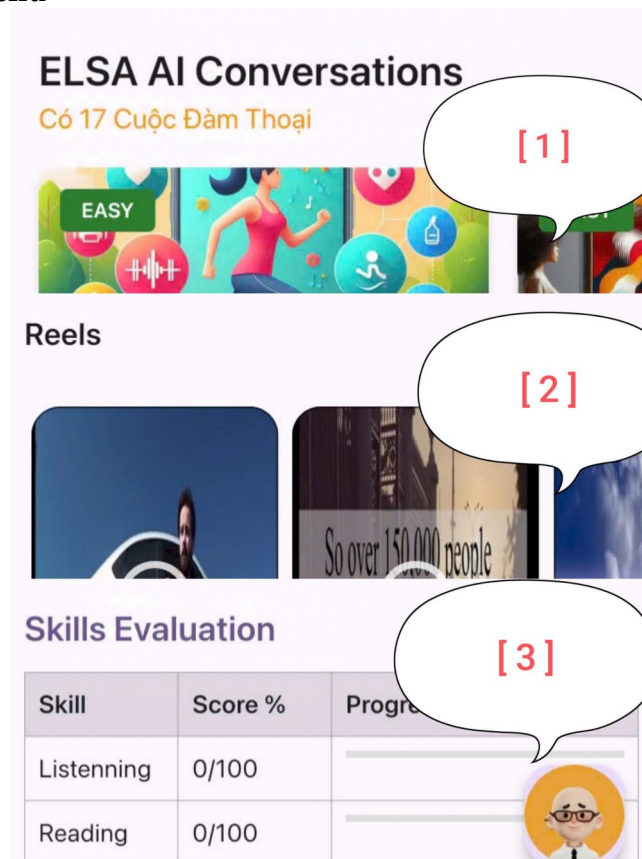


Hình 4.1.5-2. Chuyển đổi song ngữ

Chế độ Ban đêm: Ứng dụng hỗ trợ chế độ ban đêm, cho phép người dùng chuyển đổi giữa giao diện sáng và tối tại bất kỳ thời điểm nào. Chế độ ban đêm giúp giảm mệt mỏi mắt và tiết kiệm pin cho các thiết bị di động. Người dùng có thể chọn chế độ họ ưa thích.

Trong phần “Cài đặt tính năng hệ thống” đi kèm các tính năng như: chỉnh sửa hồ sơ, đổi mật khẩu, thông tin của ứng dụng và đăng xuất.

4.1.6. Trang chủ



Hình 4.1.6-1. Trang chủ

- Hình trên là các tính năng được bố trí trong trang chủ bao gồm:

[1]. Các cuộc hội thoại theo chủ đề (Đã được trình bày ở mục 4.1.1)

[2]. Reels

Tính năng này được lấy cảm hứng từ việc lướt video từ các nền tảng xã hội hiện nay như: Tiktok, Facebook Reels, Youtube Short,.... Từ đó giúp việc học tiếng Anh không còn trở lên nhàm chán.

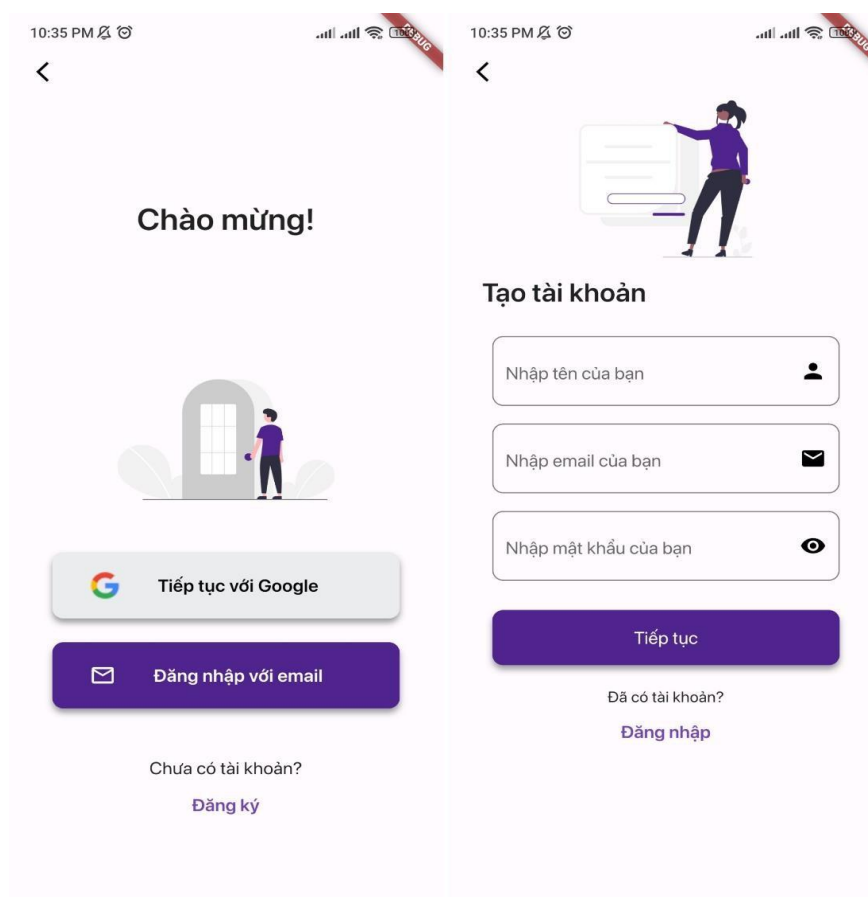
[3]. Tiến độ học tập của kỹ năng

Mục này theo dõi tiến độ học tập của hai kỹ năng Nghe – Đọc.

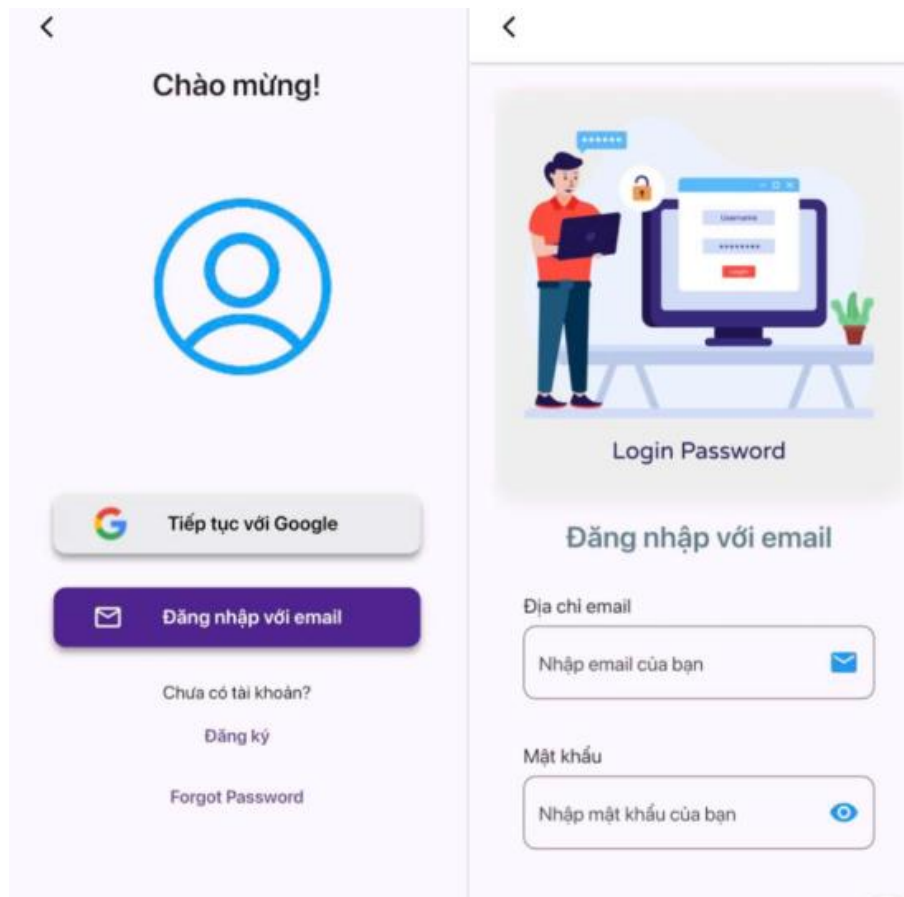
4.1.7. Tính năng xác thực người dùng

Tính năng đăng nhập cho phép người dùng truy cập vào tài khoản cá nhân của họ, nơi họ có thể lưu trữ thông tin cá nhân, theo dõi tiến trình học tập, và đồng bộ hóa dữ liệu qua các thiết bị khác.

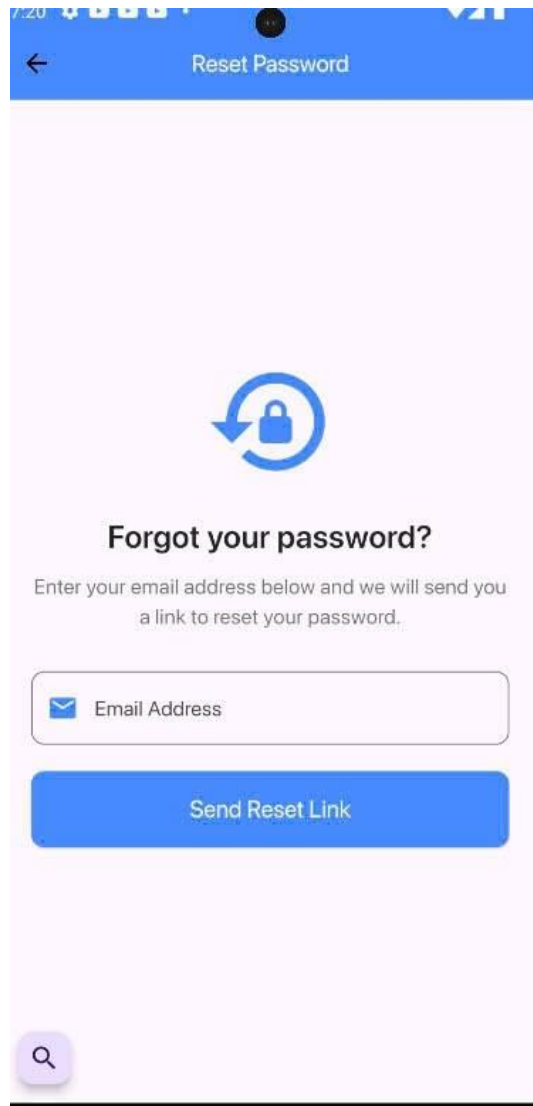
Ứng dụng cung cấp 2 hình thức để đăng nhập là qua tài khoản Google và qua thư điện tử (cần đăng ký).



Hình 4.1.7-1. Tính năng đăng ký



Hình 4.1.7-2. Tính năng đăng nhập

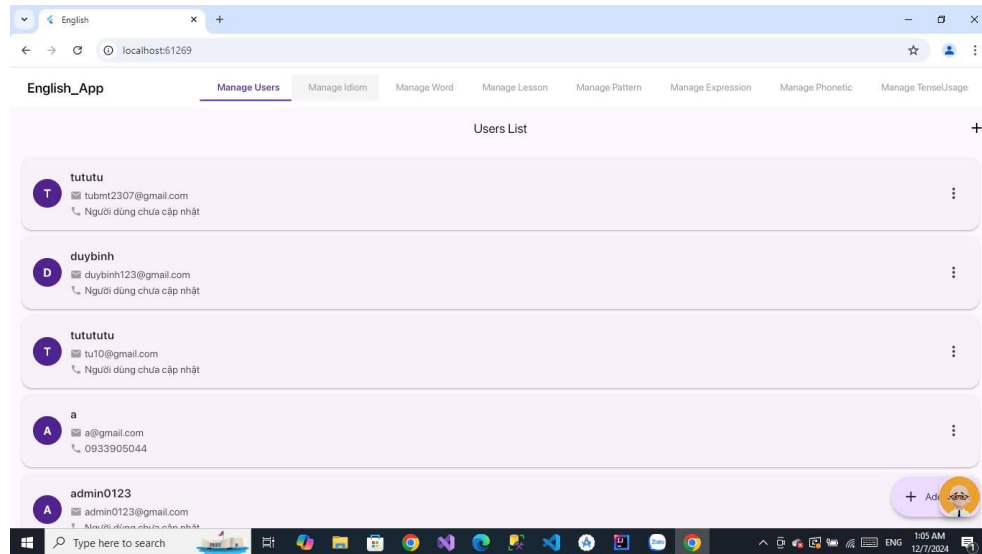


Hình 4.1.7-3. Tính năng quên mật khẩu

4.2. Các chức năng của Admin

4.2.1. Quản lý danh sách người dùng

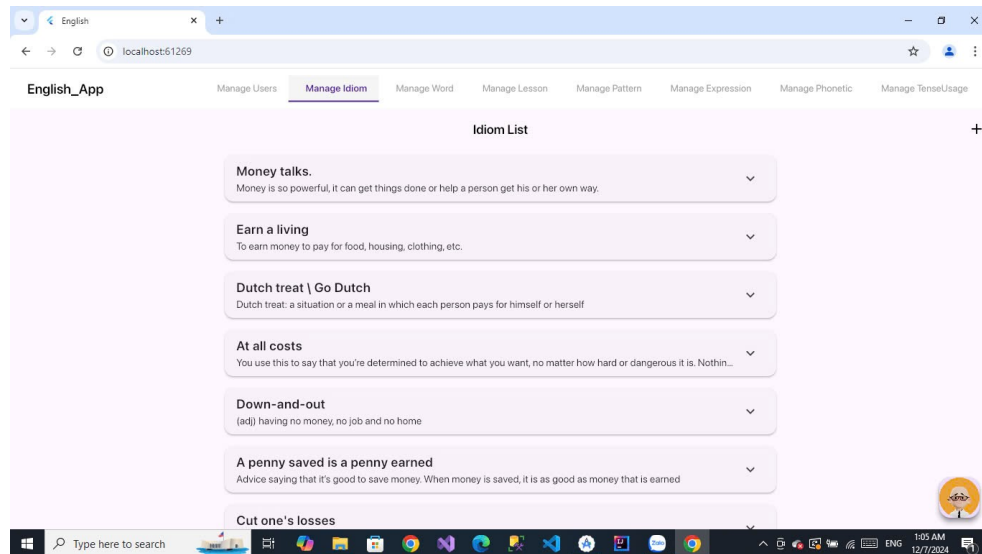
Đây là trang để quản lý từng thông tin người dùng trên hệ thống



Hình 4.2.1. Quản lý danh sách người dùng

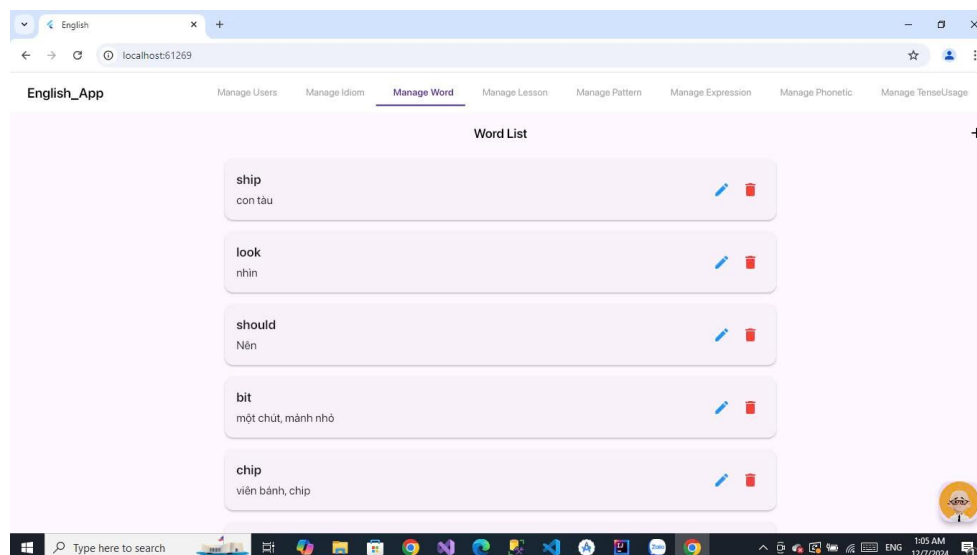
4.2.2. Quản lý danh sách các câu thành ngữ

Đây là trang quản lý các câu thành ngữ được thêm trong mục học tập.



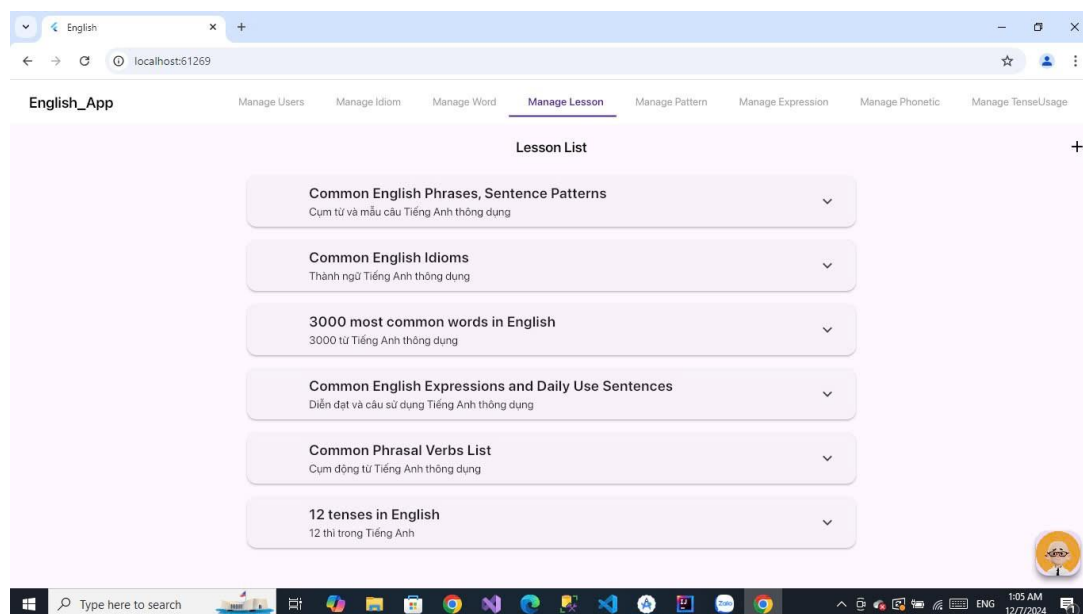
Hình 4.2.2. Quản lý danh sách các câu thành ngữ

4.2.3. Quản lý các từ vựng



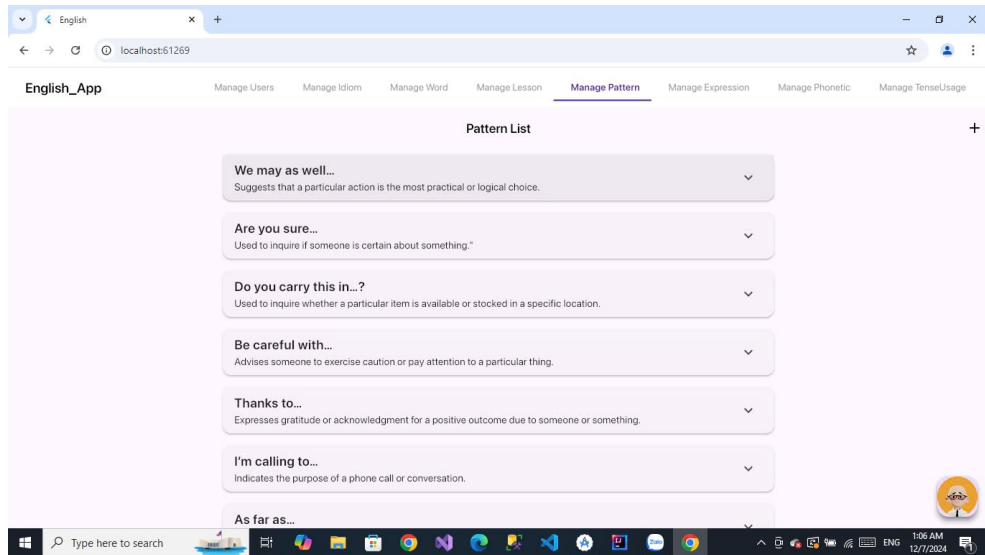
Hình 4.2.3. Quản lý các từ vựng

4.2.4. Quản lý các bài học



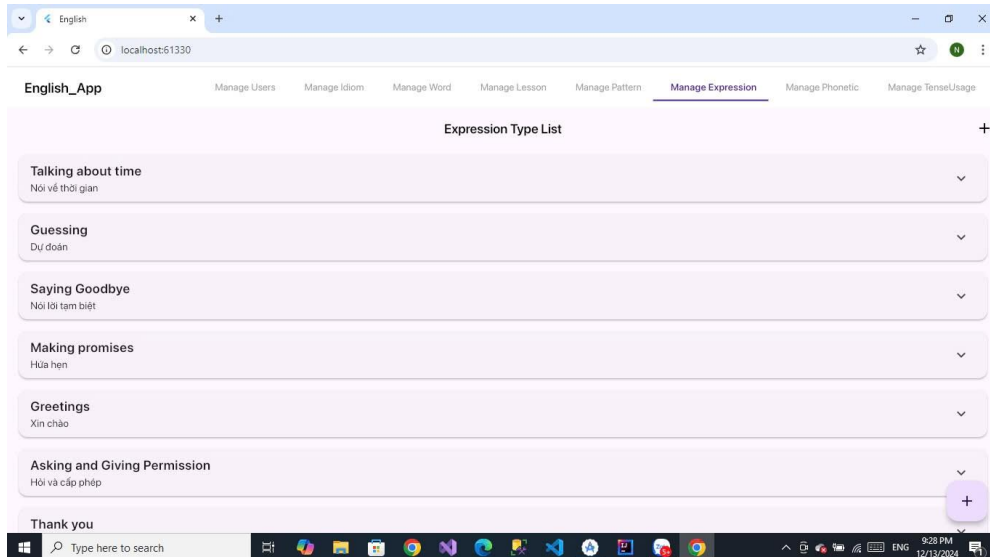
Hình 4.2.4. Quản lý các bài học

4.2.5. Quản lý danh sách các mẫu câu



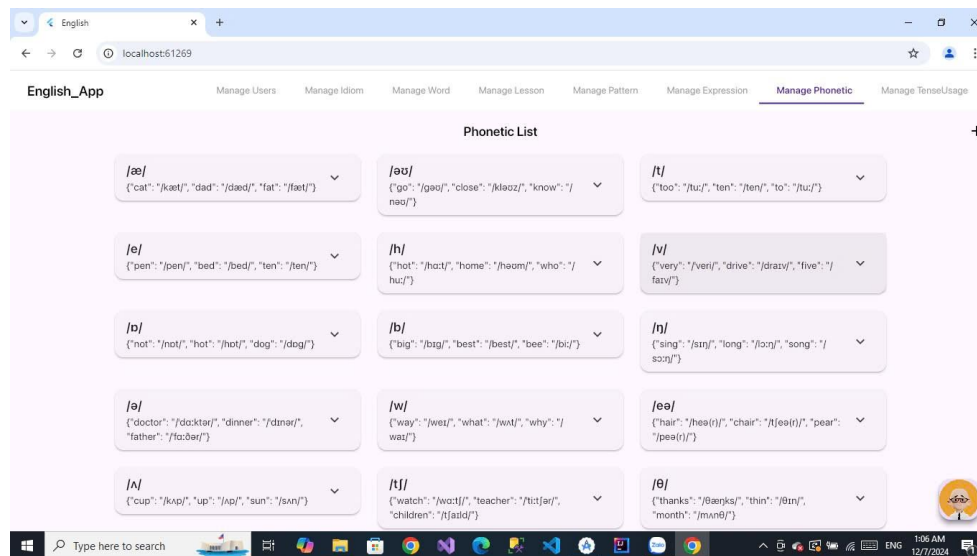
Hình 4.2.5. Quản lý các mẫu câu

4.2.6. Quản lý loại mẫu câu thường dùng



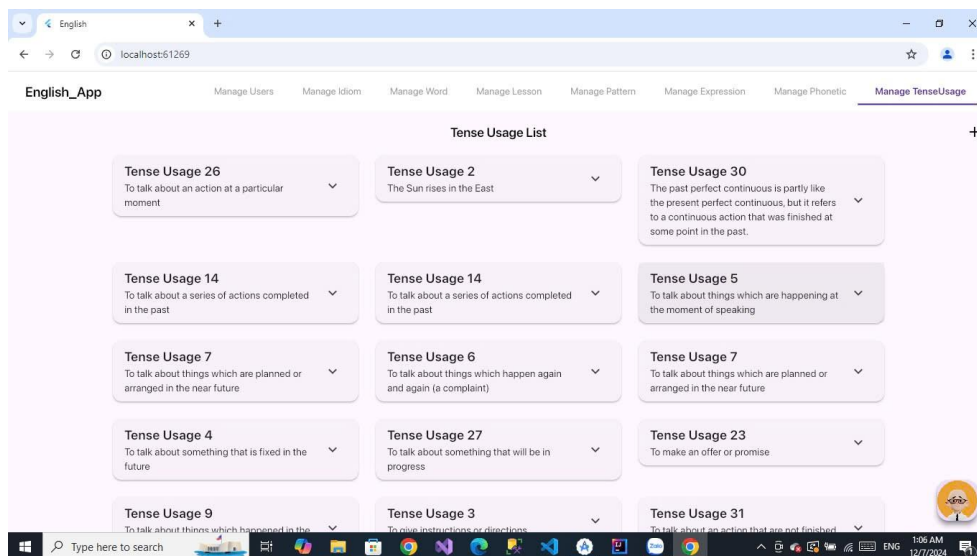
Hình 4.2.6. Quản lý loại mẫu câu thường dùng

4.2.7. Quản lý danh sách âm IPA



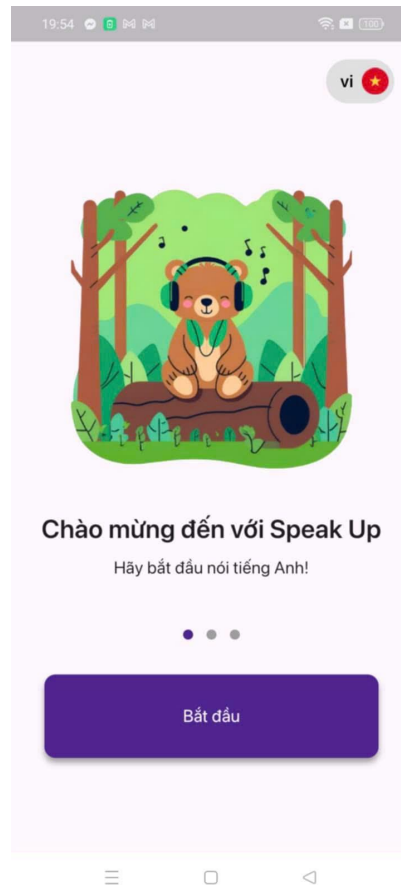
Hình 4.2.7. Quản lý danh sách âm IPA

4.2.8. Quản lý danh sách thì thông dụng



Hình 4.2.8. Quản lý danh sách thì thông dụng

4.3. Giao diện chào mừng hệ thống



Hình 4.3. Giao diện hệ thống

CHƯƠNG 5. KẾT LUẬN

5.1. Kết luận

Từ việc nghiên cứu nhu cầu thị trường, lên kế hoạch, lựa chọn nền tảng, thiết kế cơ sở dữ liệu, phát triển ứng dụng, đến việc đánh giá và kế hoạch tương lai là một cuộc hành trình phát triển ứng dụng đầy thách thức.

Đồ án đã xây dựng một ứng dụng hữu ích, đáp ứng nhu cầu học tập và giao tiếp tiếng Anh của người dùng một cách hiệu quả. Ứng dụng đã tích hợp nhiều tính năng đa dạng đến việc cung cấp kiến thức nền tảng về tiếng Anh, giúp người dùng tự tin sử dụng tiếng Anh trong cuộc sống hằng ngày, đặc biệt trong giao tiếp đó là mục đích mà đồ án hướng tới.

5.2. Hướng phát triển tương lai

Mặc dù ứng dụng tương đối ổn định và hữu ích, tuy nhiên vẫn cần một số việc quan trọng cần triển khai trong tương lai:

- Mở rộng tính năng: Nghiên cứu và thử nghiệm các tính năng như hệ thống xếp hạng người dùng, thông báo nhắc nhở học tập.
- Cải thiện trải nghiệm người dùng: Luôn tập trung vào việc cải thiện trải nghiệm người dùng để tạo ra một ứng dụng học tiếng Anh tốt hơn và hấp dẫn hơn.
- Đưa ứng dụng lên Google Play và Appstore: Để mọi người có thể trải nghiệm và sử dụng, giúp ứng dụng đạt được mục đích chính cuối cùng là có thể hỗ trợ mọi người trong việc học Tiếng Anh, đặc biệt là tiếng Anh giao tiếp.
- Từ những kết quả góp ý của những người sử dụng, đánh giá điểm mạnh và điểm yếu của sản phẩm, giúp hoàn thiện sản phẩm.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Phạm Văn Thịnh, Tìm hiểu về mô hình MVVM, [Online], 2021,

Xem tại: <https://viblo.asia/p/tim-hieu-ve-mo-hinh-mvvm-maGK7vW95j2>

[2] Flutter là gì?, [Online],

Xem tại:

<https://topdev.vn/blog/flutter-la-gi/>

[3] Nguyễn Khắc Bình, Quản lý state trong Flutter với Riverpod, [Online], 2021,

Xem tại: <https://viblo.asia/p/phan-i-quan-ly-state-trong-flutter-voi-riverpod-huong-dan-co-ban-63vKjdq6l2R>

[4] Trần Việt, Clean Architecture là gì?, [Online], 2022,

Xem tại: <https://200lab.io/blog/clean-architecture-uu-nhuoc-va-cach-dung-hop-ly>

[5] Phạm Xuân Nam, Tìm hiểu sơ lược về Firebase, [Online], 2021,

Xem tại: <https://viblo.asia/p/tim-hieu-so-luoc-ve-firebase-Eb85oeOmZ2G>