

**TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN**



BÀI TẬP LỚN
MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG CHO
CÁC THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: Xây dựng ứng dụng Mobile AIGYCHAT

Mã sinh viên	Họ và tên	Lớp
2151173789	Hồ Tuấn Kiệt	63KTMP1
2151170579	Nguyễn Đức Anh	63KTMP1
2151170592	Đinh Gia Khánh	63KTMP1
2151173833	Phạm Xuân Trường	63KTMP1

Hà Nội, Năm 2024

**TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN**



BÀI TẬP LỚN
MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG CHO
CÁC THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: Xây dựng ứng dụng Mobile AIGYCHAT

Mã sinh viên	Họ và tên	Lớp
2151173789	Hồ Tuấn Kiệt	63KTMP1
2151170579	Nguyễn Đức Anh	63KTMP1
2151170592	Đinh Gia Khánh	63KTMP1
2151173833	Phạm Xuân Trường	63KTMP1

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, Năm 202

Lời nói đầu

Trong thời đại cách mạng công nghiệp 4.0, trí tuệ nhân tạo (AI) đang ngày càng chứng tỏ vai trò quan trọng trong việc tối ưu hóa hoạt động và hỗ trợ con người. AI không chỉ là một khái niệm công nghệ của tương lai mà đã trở thành một phần thiết yếu trong cuộc sống hiện đại, giúp tăng cường hiệu suất và tối ưu hóa các quy trình. Tuy nhiên, việc sử dụng AI hiệu quả, đặc biệt là khả năng tạo các prompt để đạt được kết quả mong muốn, vẫn là một thách thức lớn đối với nhiều người dùng.

Dự án ứng dụng AIGY của chúng tôi ra đời với mục tiêu mang đến giải pháp thân thiện, giúp người dùng dễ dàng khai thác sức mạnh của AI mà không cần kỹ năng lập trình phức tạp. Với hệ thống multi-agent thông minh, mỗi agent được tối ưu hóa cho các mục đích sử dụng riêng, từ người dùng cá nhân đến các doanh nghiệp nhỏ và chuyên gia kỹ thuật. AIGY hướng tới việc đơn giản hóa quy trình tương tác với AI, giúp người dùng tiết kiệm thời gian và nâng cao hiệu quả công việc.

Chúng tôi hy vọng rằng dự án này không chỉ mang lại những giá trị thiết thực cho người dùng mà còn là một bước tiến trong việc áp dụng AI vào các ứng dụng di động. Báo cáo này sẽ trình bày chi tiết quá trình phát triển và thử nghiệm ứng dụng, phân tích hiệu quả và tiềm năng mà AIGY mang lại trong việc tối ưu hóa trải nghiệm người dùng trong kỷ nguyên số.

Mục Lục

Lời nói đầu	3
BẢNG CÁC TỪ VIẾT TẮT	6
CHƯƠNG 1: MÔ TẢ BÀI TOÁN.....	7
1.1. Giới thiệu	7
1.2. Chức năng chính	7
1.3. Yêu cầu phi chức năng	8
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG	10
2.1. Phân tích yêu cầu.....	10
2.2. Thiết kế hệ thống.....	13
- Biểu đồ lớp cho mô hình miền.....	13
- Biểu đồ lớp cho mô hình MVVM.....	13
- Thiết kế cấu trúc MVVM	15
- Thiết kế cơ sở dữ liệu	21
- Giao diện quản lý.....	22
- Giao diện Splash	24
- Giao diện welcome	24
- Giao diện SignUp.....	25
- Giao diện SignIn	25
- Giao diện ForgetPassword	26
- Giao diện SelectMethodRecovery	26
- Giao diện Profile.....	27
- Giao diện Preferences	27
- Giao diện AccountInfomation	28
- Giao diện Password	28
- Giao diện InviteYourFriend.....	29
- Giao diện Theme.....	29
- Giao diện Language	30
- Giao diện Chat	31
- Giao diện AttachFile.....	31

- Giao diện Agent.....	32
- Giao diện AgentDetail.....	32
CHƯƠNG 3: KẾT QUẢ THỰC HIỆN	33
3.1. Công nghệ đã sử dụng.....	33
3.2. Tiến độ thực hiện.....	33
KẾT LUẬN.....	34

BẢNG CÁC TỪ VIẾT TẮT

CSDL	Cơ sở dữ liệu

CHƯƠNG 1: MÔ TẢ BÀI TOÁN

1.1. Giới thiệu

Trong kỷ nguyên số, AI không chỉ là công nghệ tương lai, mà còn là công cụ hỗ trợ mạnh mẽ cho cuộc sống hiện đại. Tuy nhiên, không phải ai cũng am hiểu về cách sử dụng AI một cách hiệu quả, đặc biệt là việc viết các prompt phức tạp để đạt kết quả mong muốn. Với ứng dụng AIGY của chúng tôi, điều đó trở nên dễ dàng hơn bao giờ hết.

Chúng tôi mang đến một trải nghiệm hoàn toàn mới mẻ và thân thiện cho người dùng với hệ thống multi agent thông minh. Mỗi agent được tùy chỉnh với các prompt tối ưu hóa, phù hợp cho từng mục đích và nhu cầu cụ thể. Dù bạn là người dùng cá nhân, doanh nghiệp nhỏ, nhà sáng tạo nội dung, hay chuyên gia kỹ thuật, ứng dụng này sẽ giúp bạn khai thác sức mạnh của AI một cách đơn giản và hiệu quả nhất.

Chỉ với vài thao tác, bạn có thể dễ dàng chọn agent phù hợp, tương tác với AI và đạt được kết quả mong muốn mà không cần hiểu biết sâu về công nghệ hay phải viết prompt thủ công. Hãy để AIGY của chúng tôi trở thành trợ lý đắc lực, giúp bạn tiết kiệm thời gian, nâng cao hiệu suất và khám phá tiềm năng vô hạn từ trí tuệ nhân tạo!

1.2. Chức năng chính

Ứng dụng AIGYCHAT cần có các chức năng sau:

- Trò chuyện cùng Agent: Người dùng có thể chọn từ danh sách các agent tối ưu hóa sẵn cho nhiều nhu cầu khác nhau (marketing, học tập, chăm sóc khách hàng, sáng tạo nội dung, v.v.). Mỗi agent có prompt tối ưu, giúp người dùng không cần phải viết prompt phức tạp mà vẫn đạt kết quả mong muốn. Cuộc trò chuyện được lưu lại và đồng bộ hóa trên nhiều thiết bị để người dùng dễ dàng truy cập.
- Lưu trữ dữ liệu trên Firebase:
 - Thông tin người dùng: Lưu trữ thông tin đăng nhập, tài khoản cá nhân và hồ sơ người dùng, giúp họ dễ dàng truy cập dữ liệu từ nhiều thiết bị khác nhau.
 - Dữ liệu agent đã tải: Các agent mà người dùng đã chọn hoặc tải về sẽ được lưu trữ, giúp người dùng truy cập nhanh chóng mà không cần tải lại mỗi lần sử dụng.
 - File đã upload: Lưu trữ các tệp tin mà người dùng đã upload trong quá trình sử dụng, đảm bảo chúng luôn sẵn sàng cho các tác vụ sau mà không cần phải tải lại từ đầu.
- Tùy chỉnh agent theo nhu cầu cá nhân: Người dùng có thể tạo và tùy

chỉnh các agent theo mục đích riêng, điều chỉnh prompt cho phù hợp với công việc cụ thể. Các agent này sẽ được lưu lại để dễ dàng sử dụng trong tương lai, giúp cá nhân hóa trải nghiệm AI mà không cần phải biết cách viết prompt chi tiết.

- **Bảng điều khiển cá nhân:** Bảng điều khiển giúp người dùng quản lý agent, theo dõi lịch sử trò chuyện và kiểm soát các tác vụ. Tại đây, họ có thể xem, chỉnh sửa agent, tiếp tục các cuộc trò chuyện trước đó, và theo dõi tiến độ các công việc đã thực hiện.
- **Khám phá cửa hàng agent:** Chức năng này cho phép người dùng tìm kiếm và khám phá các agent tối ưu hóa cho nhiều nhu cầu khác nhau. Người dùng có thể duyệt qua danh sách agent, lọc và tìm kiếm theo từ khóa, xem thông tin chi tiết về từng agent, và nhận đề xuất phù hợp dựa trên hành vi sử dụng.
- **Upload file:** Chức năng này cho phép người dùng tải lên các tệp tin cần thiết cho agent xử lý. Người dùng có thể tải lên nhiều định dạng file khác nhau, quản lý các file đã tải lên trong ứng dụng, và tích hợp file vào các tác vụ của agent như phân tích dữ liệu hoặc tạo nội dung.

1.3. Yêu cầu phi chức năng

- **Hiệu suất:**
 - **Thời gian phản hồi:** Hệ thống phải cung cấp phản hồi nhanh chóng cho người dùng, đặc biệt trong quá trình trò chuyện với agent (dưới 2 giây cho mỗi yêu cầu).
 - **Khả năng xử lý:** Hệ thống cần hỗ trợ đồng thời nhiều người dùng mà không làm giảm hiệu suất.
- **Bảo mật:**
 - **Bảo mật dữ liệu:** Tất cả thông tin người dùng và dữ liệu được lưu trữ phải được mã hóa và bảo vệ để ngăn chặn truy cập trái phép.
 - **Xác thực người dùng:** Cần có các biện pháp xác thực mạnh mẽ (ví dụ: xác thực hai yếu tố) để đảm bảo chỉ người dùng đã đăng ký mới có quyền truy cập vào thông tin cá nhân và dữ liệu.
- **Tính khả dụng:**
 - **Thời gian hoạt động:** Hệ thống phải đảm bảo tỷ lệ thời gian hoạt động tối thiểu 99,5% trong một tháng.
 - **Khả năng phục hồi:** Hệ thống phải có khả năng khôi phục nhanh chóng sau sự cố hoặc lỗi, đảm bảo người dùng không bị ảnh hưởng lâu dài.

- Tính khả thi:
 - Dễ sử dụng: Giao diện người dùng phải thân thiện và dễ sử dụng, giúp người dùng có thể dễ dàng tương tác mà không cần quá nhiều hướng dẫn.
 - Độ tin cậy: Hệ thống phải cung cấp thông tin chính xác và đáng tin cậy từ các agent.
- Tính mở rộng:
 - Khả năng mở rộng: Hệ thống phải có khả năng mở rộng để hỗ trợ thêm người dùng và agent trong tương lai mà không gặp phải vấn đề về hiệu suất.
- Tính tương thích:
 - Tích hợp với dịch vụ bên ngoài: Hệ thống cần có khả năng tích hợp dễ dàng với các dịch vụ và API bên ngoài.
- Bảo trì:
 - Dễ bảo trì: Hệ thống cần được thiết kế để dễ dàng bảo trì và cập nhật, bao gồm cả việc sửa lỗi và thêm tính năng mới mà không làm gián đoạn dịch vụ cho người dùng.

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu

Xác định người dùng:

- Người dùng cá nhân không chuyên:
 - Đặc điểm: Những người không có kiến thức sâu về AI hoặc không rành viết prompt. Họ tìm kiếm sự đơn giản trong việc sử dụng agent để thực hiện các tác vụ hàng ngày hoặc giải quyết vấn đề cụ thể (như tìm kiếm thông tin, viết nội dung, dịch thuật, v.v.).
 - Nhu cầu: Cần một giao diện dễ sử dụng với các agent đã được tối ưu hóa sẵn cho các mục đích khác nhau giúp người dùng không cần phải viết prompt quá phức tạp.
- Doanh nghiệp nhỏ và vừa:
 - Đặc điểm: Những doanh nghiệp muốn tận dụng công nghệ AI để cải thiện hiệu suất làm việc, marketing, chăm sóc khách hàng mà không có đội ngũ kỹ thuật lớn.
 - Nhu cầu: Cần có các agent AI với các prompt sẵn giúp tạo nội dung, quản lý khách hàng, hoặc thực hiện các tác vụ như xử lý dữ liệu hoặc báo cáo.
- Người sáng tạo nội dung
 - Đặc điểm: Blogger, YouTuber, và các nhà sáng tạo nội dung khác thường cần AI hỗ trợ để tạo ý tưởng, soạn thảo bài viết, chỉnh sửa văn bản, hoặc thiết kế nội dung.
 - Nhu cầu: Cần các agent AI được tối ưu hóa cho các tác vụ sáng tạo, giúp họ tập trung vào ý tưởng thay vì các chi tiết kỹ thuật.
- Giáo viên và học sinh:
 - Đặc điểm: Những người trong môi trường giáo dục cần một công cụ hỗ trợ học tập, giảng dạy, tạo bài kiểm tra, hoặc nghiên cứu.
 - Nhu cầu: Cần một công cụ dễ sử dụng với các prompt giúp họ tạo câu hỏi, giải thích khái niệm hoặc tìm kiếm tài liệu giáo dục.
- Chuyên gia kỹ thuật:
 - Đặc điểm: Các lập trình viên, nhà nghiên cứu AI, hoặc chuyên gia trong lĩnh vực kỹ thuật cần sử dụng AI để trợ giúp trong công việc chuyên môn.
 - Nhu cầu: Cần các prompt sâu hơn để khai thác các khía cạnh chuyên môn hoặc tạo ra giải pháp sáng tạo trong công việc.

- Nhóm hỗ trợ khách hàng:
 - Đặc điểm: Các nhân viên chăm sóc khách hàng trong các công ty có thể sử dụng AI chatbot để tự động hóa việc trả lời câu hỏi phổ biến của khách hàng.
 - Nhu cầu: Cần các agent AI được tối ưu hóa để quản lý các câu hỏi về sản phẩm/dịch vụ, xử lý các yêu cầu hỗ trợ hoặc đưa ra thông tin nhanh chóng.

Thu thập yêu cầu:

Dựa trên mô tả bài toán, chúng tôi đã xác định các chức năng chính cho ứng dụng AIGY bao gồm:

- Chọn và trò chuyện với agent: Người dùng có thể chọn từ danh sách các agent được tùy chỉnh và tối ưu hóa cho các mục đích cụ thể như marketing, sáng tạo nội dung, hỗ trợ khách hàng, v.v. Mỗi agent đã được tích hợp sẵn các prompt để người dùng không cần phải tự viết prompt phức tạp mà vẫn đạt kết quả mong muốn.
- Tùy chỉnh agent: Cho phép người dùng tạo hoặc điều chỉnh prompt của các agent theo nhu cầu cá nhân, lưu lại để sử dụng trong tương lai.
- Lưu trữ và quản lý file: Người dùng có thể tải lên và quản lý các file để hỗ trợ agent thực hiện các tác vụ như phân tích hoặc tạo nội dung. Các file sẽ được lưu trữ trên nền tảng đám mây (Firebase) để dễ dàng truy cập từ nhiều thiết bị.
- Quản lý thông tin người dùng và agent đã chọn: Lưu trữ thông tin người dùng và lịch sử agent đã sử dụng, giúp dễ dàng theo dõi và tiếp tục các cuộc trò chuyện cũ mà không cần thiết lập lại.
- Khám phá cửa hàng agent: Cung cấp một giao diện để người dùng có thể tìm kiếm và duyệt qua các agent phù hợp, xem thông tin chi tiết, và nhận đề xuất dựa trên hành vi sử dụng.

Phân tích yêu cầu:

1. Chọn Agent và trò chuyện
 - Người dùng có thể chọn Agent phù hợp với mục đích từ danh sách Agent được tối ưu hoá sẵn, sau đó trò chuyện với AI mà không cần viết prompt.
 - Hệ thống cần lưu lại lịch sử trò chuyện và đồng bộ hoá trên nhiều thiết bị để người dùng dễ dàng truy cập lại khi cần
2. Tùy chỉnh Agent
 - Người dùng có thể tạo hoặc điều chỉnh agent riêng bằng cách thay đổi prompt mặc định để phù hợp với yêu cầu cụ

thể. Agent tùy chỉnh này sẽ được lưu trữ để người dùng dễ dàng sử dụng trong tương lai.

3. Quản lý file tải lên

- Người dùng có thể tải các tệp lên hệ thống để agent xử lý và dễ dàng quản lý các tệp đã tải lên trong ứng dụng.
- Hệ thống cần hỗ trợ các định dạng file phổ biến và lưu trữ chúng trên nền tảng đám mây (Firebase), đảm bảo các file luôn sẵn sàng cho các tác vụ sau mà không cần phải tải lại.

4. Lưu trữ và quản lý thông tin người dùng và Agent

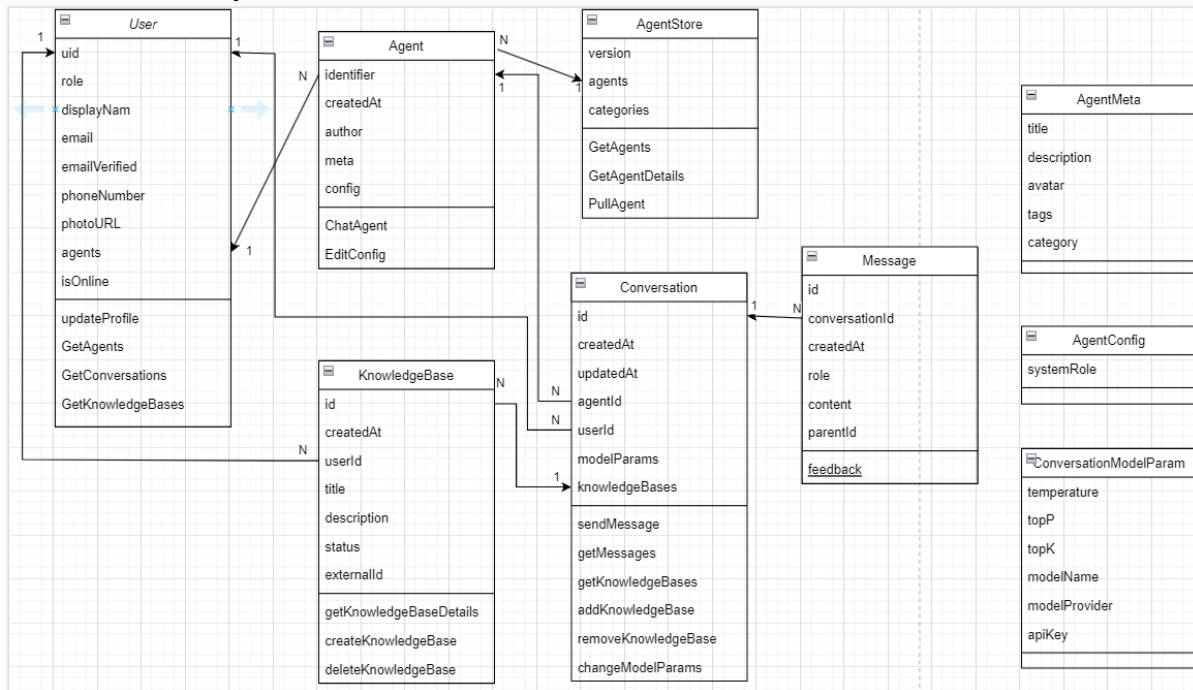
- Ứng dụng cần lưu thông tin tài khoản người dùng, agent đã sử dụng, các file đã tải lên và lịch sử trò chuyện, cho phép người dùng truy cập nhanh chóng và dễ dàng từ nhiều thiết bị.

5. Khám phá và tải Agent

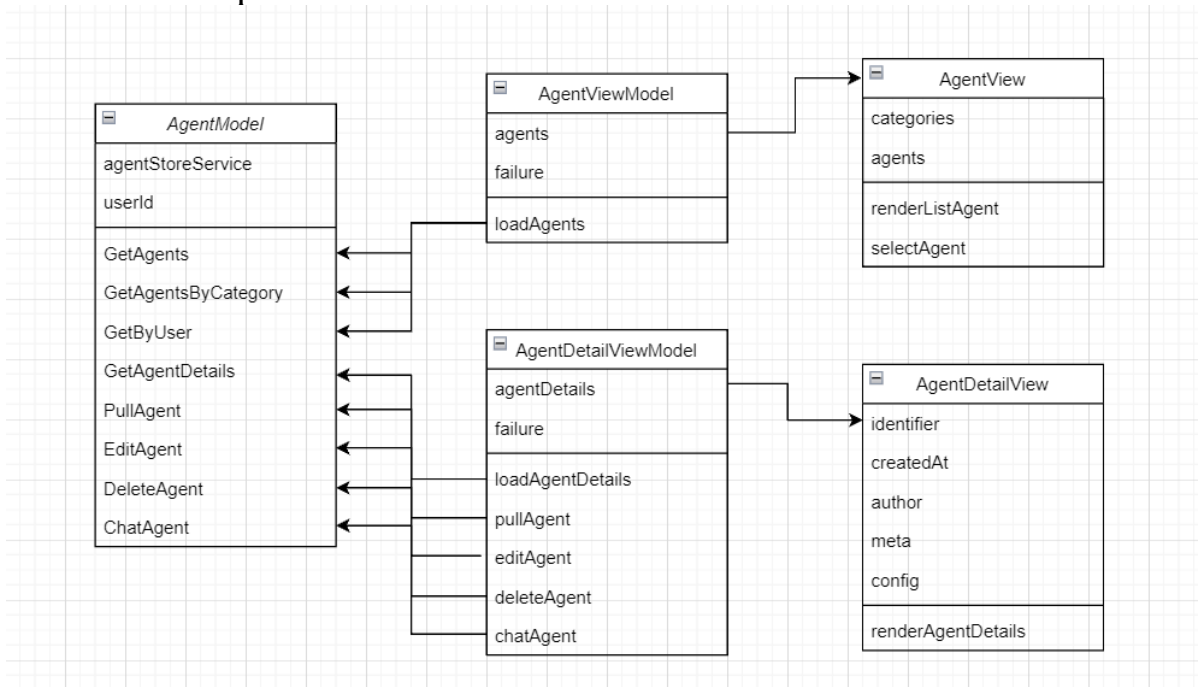
- Cung cấp giao diện cửa hàng để người dùng tìm kiếm các agent phù hợp với nhu cầu. Cho phép duyệt qua các danh sách, lọc theo từ khóa và xem thông tin chi tiết của từng agent, với các đề xuất được cá nhân hóa dựa trên hành vi sử dụng.

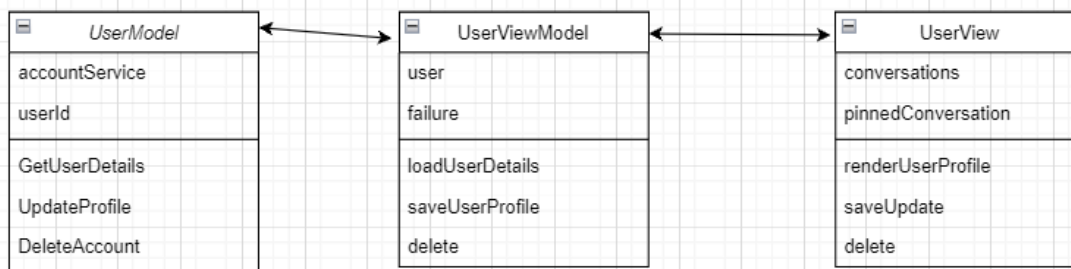
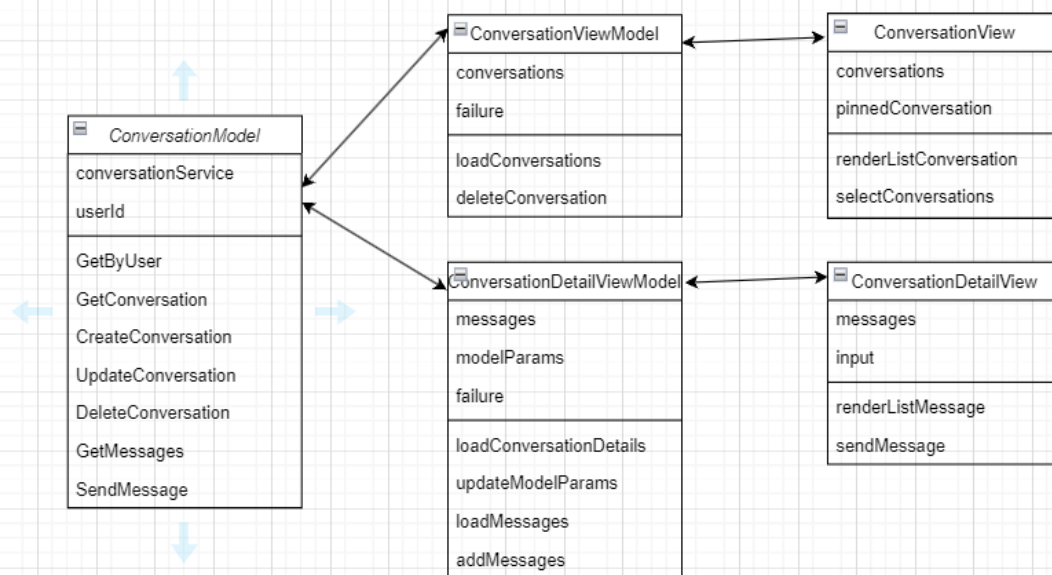
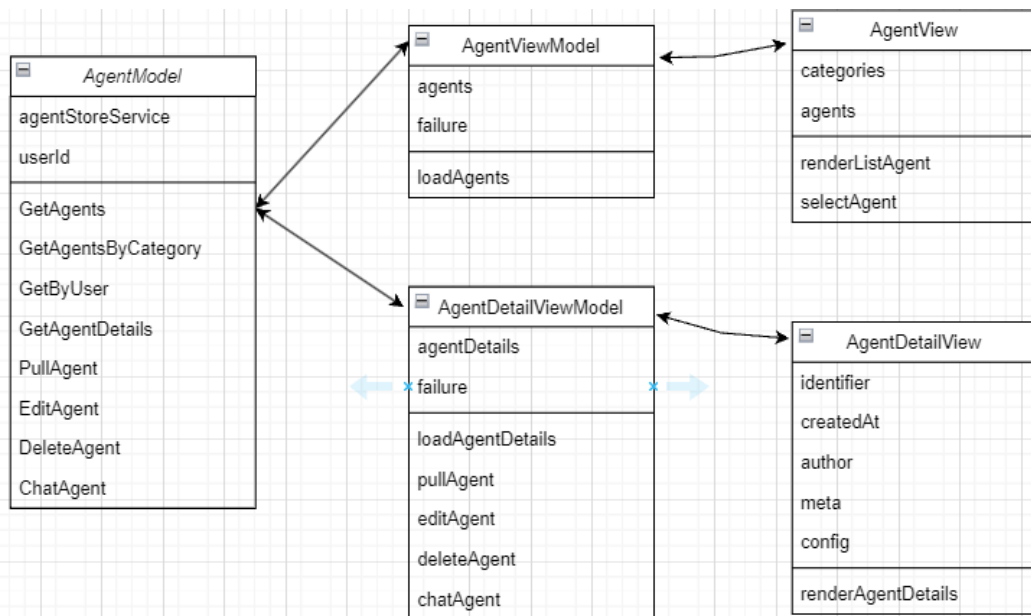
2.2. Thiết kế hệ thống

- Biểu đồ lớp cho mô hình miền



- Biểu đồ lớp cho mô hình MVVM





- Thiết kế cấu trúc MVVM

Model

- User:
 - Class đại diện cho người dùng
 - Chứa các thuộc tính như: name , uid , email , phone , role, isOnline , imgUrl , idListAgent , listVisit
 - Chứa các phương thức: GetUserDetails, UpdateProfile, DeleteAccount.
- Agent
 - Class đại diện cho trợ lý
 - Chứa các thuộc tính như: id, author, title, description, config
 - Chứa các phương thức: GetAgents, GetAgentDetails, PullAgent, EditAgent, DeleteAgent, ChatAgent.
- Conversation
 - Class đại diện cho các cuộc trò chuyện
 - Chứa các phương thức: GetByUser, CRUD Conversation, GetMessages, SendMessage.
- Message
 - Class đại diện cho các tin nhắn
 - Chứa các thuộc tính như: id, role, content. createdAt, completionTime

View

- Splash, LiveShow: Chứa các file xml chứa giao diện phần mở đầu , giới thiệu về ứng dụng
- Signin, Signup, ForgetPassword: Chứa các file xml về giao diện liên quan đến đăng nhập, đăng ký, quên mật khẩu của người dùng
- Profile : Chứa các file xml về giao diện liên quan đến người dùng : xem thông tin cá nhân , thêm bạn bè , chuyển đổi giao diện sáng tối , chuyển đổi ngôn ngữ
- AgentView : Chứa các file xml về giao diện liên quan đến các Agents (trợ lý)
- AgentDetailsView : Chứa các file xml về giao diện liên quan

thông tin chi tiết Agents (trợ lý)

- MessageView : Chứa các file xml về các giao diện liên quan đến phần nhắn tin giữa các Agents với nhau
- AdminView : Chứa các file xml về các giao diện liên quan đến phần quản lý : xem các báo biểu liên quan đến hệ thống như số lượng người truy cập , số lượng người dùng , quản lý thông tin người dùng , thay đổi giao diện sáng tối , thay đổi ngôn ngữ

ViewModel

- UserViewModel : Chứa các viewmodel hỗ trợ cho việc tiếp nhận các thông tin của người dùng thông qua UserView , Signin , Signup , ForgetPassword , Profile để xử lý các yêu cầu bằng cách sử dụng User và Agent , Conversation , Message để có thể hiển thị kết quả lên UserView , Signin , Signup , ForgetPassword , Profile
- AgentViewModel : Chứa các viewmodel hỗ trợ cho việc tiếp nhận các thông tin của các agent thông qua AgentView, AgentDetailsView để xử lý các yêu cầu bằng cách sử dụng AgentModel để hiển thị lên View
- AdminViewModel : Chứa các viewmodel hỗ trợ cho việc tiếp nhận các thông tin của quản lý thông qua AdminView để xử lý các yêu cầu bằng cách sử dụng User và Agent , Conversation để có thể hiển thị kết quả lên AdminView

- Xác định các lớp:

- Class UserAcctivity:
 - Thuộc tính
 - Phương thức
 - onCreate(): Chứa các view fragment về User Activity
- Class AgentActivity:
 - Thuộc tính
 - Phương thức
 - onCreate() : Chứa các view fragment về Agent Activity
- Class ConversationActivity:
 - Thuộc tính
 - Phương thức
 - onCreateView() : Chứa các view fragment về

Conversation Activity

- Class MessageActivity:
 - Thuộc tính
 - Phương thức
 - onCreateView() : Chứa các view fragment về Message Activity
- Class AdminFragment:
 - Thuộc tính: Chứa các controller trong layout như: TextView, Button,...
 - Phương thức
 - onCreate(): Hàm khởi tạo các thuộc tính cho layout
 - onCreateView() : Hàm kiểm soát phần hiển thị của layout
 - onCreateView() : Hàm hiển thị layout cho Fragment
 - getEvent() : Hàm bắt sự kiện trong layout
 - loadingFragement() : Hàm tải các giao diện fragment
- Class AgentFragment:
 - Thuộc tính: Chứa các controller trong layout như: TextView, Button, ...
 - Phương thức
 - onCreate() : Hàm khởi tạo các thuộc tính cho layout
 - onCreateView() : Hàm kiểm soát phần hiển thị của layout
 - onCreateView() : Hàm hiển thị layout cho Fragment
 - getEvent() : Hàm bắt sự kiện trong layout
 - loadingFragement() : Hàm tải các giao diện Fragment
- Class ConversationFragment:
 - Thuộc tính: Chứa các controller trong layout như : TextView, Button, ...
 - Phương thức:
 - onCreate() : Hàm khởi tạo các thuộc tính cho layout
 - onCreateView() : Hàm kiểm soát phần hiển thị của layout
 - onCreateView() : Hàm hiển thị layout cho Fragment
 - getEvent() : Hàm bắt sự kiện trong layout
 - loadingFragement() : Hàm tải các giao diện fragment
- Class UserModel:
 - Thuộc tính:
 - _user: Chứa một list dữ liệu người dùng mỗi khi có sự

thay đổi

- user: Tham chiếu dữ liệu đến _user để lấy dữ liệu
- _selectUser: Chứa dữ liệu của 1 người dùng được chọn mỗi khi có sự thay đổi
- selectUser: Tham chiếu dữ liệu đến _selectUser để lấy dữ liệu.
- Phương thức:
 - fetchUser: Hàm yêu cầu lấy danh sách người dùng
 - getUserByName (name: String) : Hàm yêu cầu tìm kiếm người dùng theo tên
 - getUserByUID(uid: String): Hàm yêu cầu tìm kiếm người dùng theo uid
 - updateUser(userModel: UserModel): Hàm yêu cầu thay đổi thông tin người dùng
 - deleteUser(uid: String): Hàm yêu cầu xóa người dùng
 - createUser(userModel: UserModel): Hàm yêu cầu tạo người dùng
 - fetchVisit(): Hàm yêu cầu lấy dữ liệu 1 list truy cập
 - signUp(): Hàm yêu cầu đăng ký người dùng
 - signIn(): Hàm yêu cầu đăng nhập
- Class AgentViewModel:
 - Thuộc tính:
 - _agent: Chứa một list dữ liệu trợ lý mỗi khi có sự thay đổi.
 - agent: Tham chiếu dữ liệu đến _agent để lấy dữ liệu
 - _selectAgent: Chứa dữ liệu của 1 Agent được chọn mỗi khi có sự thay đổi
 - selectAgent: Tham chiếu dữ liệu đến _selectAgent để lấy dữ liệu
 - Phương thức:
 - fetchAgent: Hàm yêu cầu lấy danh sách các trợ lý
 - getAgentByID(Id: String): Hàm yêu cầu tìm trợ lý theo ID
 - deleteAgent(Id: String): Hàm yêu cầu xóa trợ lý
 - createAgent(agentL AgentModel): Hàm yêu cầu tạo trợ lý
- Class ConversationViewModel
 - Thuộc tính
 - _conversation : Chứa một list dữ liệu cuộc trò chuyện mỗi khi có sự thay đổi
 - conversation: Tham chiếu dữ liệu đến _conversation để

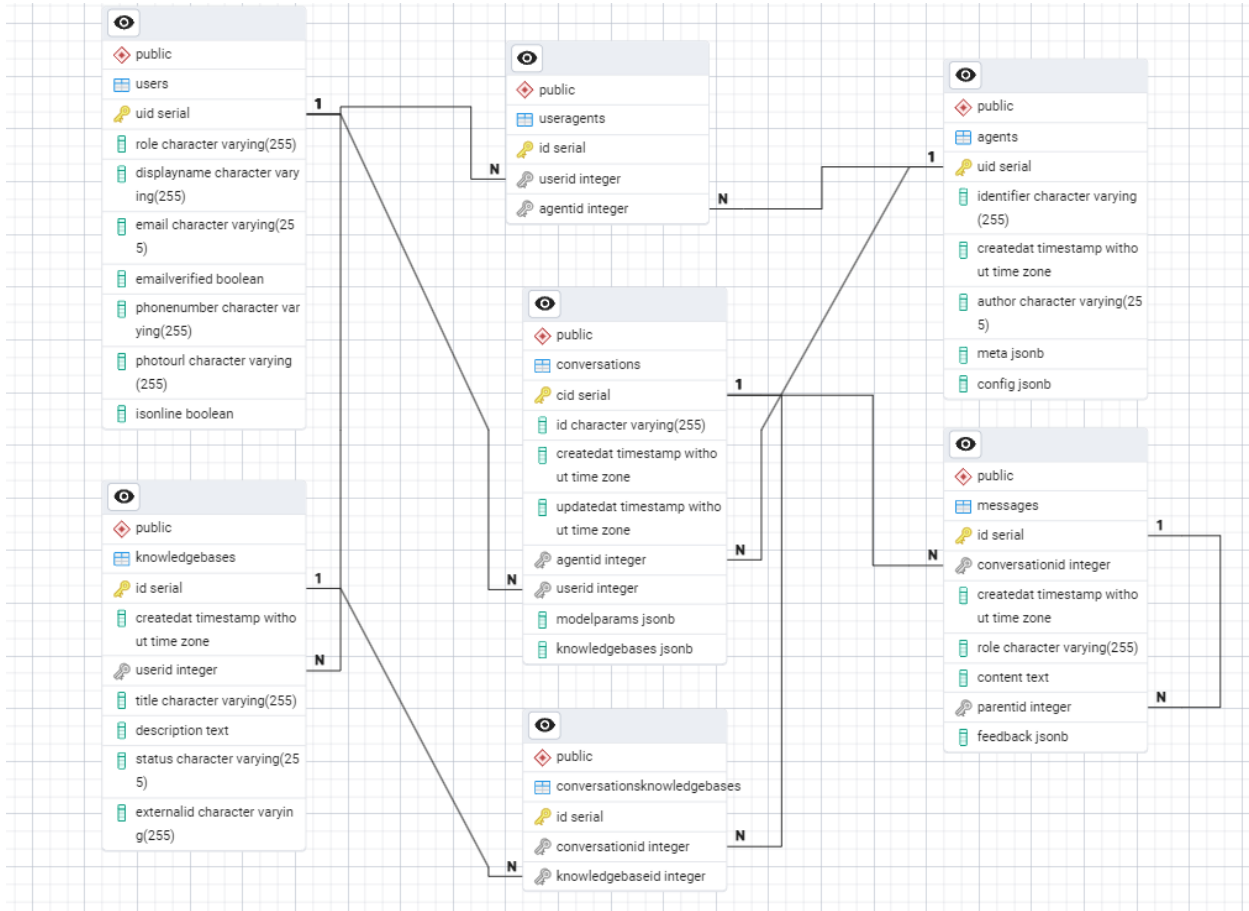
lấy dữ liệu

- `_selectConversation` : chứa dữ liệu của 1 cuộc trò chuyện được chọn mỗi khi có sự thay đổi
- `selectConversation`: Tham chiếu dữ liệu đến `_selectConversation` để lấy dữ liệu
- Phương thức
 - `fetchConversation` : Hàm yêu cầu lấy danh sách các cuộc trò chuyện
 - `getConversationById (id : String)` : Hàm yêu cầu tìm kiếm cuộc trò chuyện theo id
 - `deleteConvaersation(id : String)` : Hàm yêu cầu xóa cuộc trò chuyện
 - `createConversation(conversation : ConversationModel)` : Hàm yêu cầu tạo cuộc trò chuyện
- Class `MessageViewModel`:
 - Thuộc tính
 - `_message` : Chứa một list dữ liệu tin nhắn mỗi khi có sự thay đổi
 - `message`: Tham chiếu dữ liệu đến `_message` để lấy dữ liệu
 - Phương thức
 - `fetchMessage()` : Hàm yêu cầu lấy danh sách các tin nhắn
 - `createMessage(message : MessageModel)` : Hàm yêu cầu tạo tin nhắn
- Class `UserRespository`:
 - Thuộc tính
 - `Database` : một instance (phiên bản) của lớp `FirebaseDatabase`
 - Phương thức
 - `getUser()` : Hàm lấy danh sách người dùng
 - `getUserByName(name : String)` : Hàm tìm kiếm người dùng theo tên
 - `getUserByUID(uid : String)` : Hàm tìm kiếm người dùng theo uid
 - `updateUser(user : UserModel)` : Hàm cập nhật thông tin người dùng
 - `deleteUser(uid : String)` : Hàm xóa người dùng
 - `createUser(user : UserModel)` : Hàm tạo người dùng
 - `fetchVisit()` : Hàm trả về danh sách truy cập

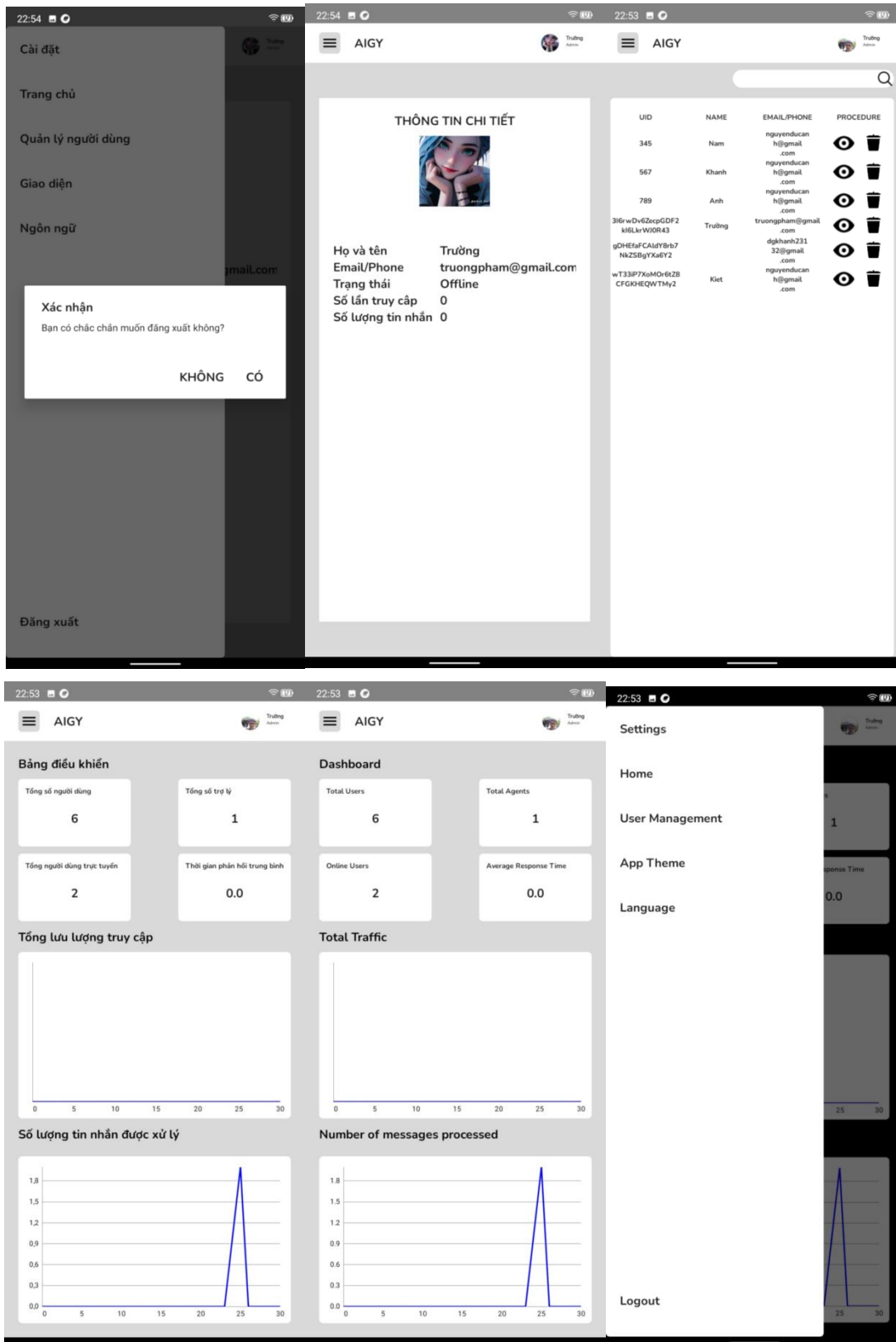
- Class ConversationRepository:
 - Thuộc tính
 - Database: một instance (phiên bản) của lớp FirebaseDatabase
 - Phương thức
 - fetchConversation : Hàm lấy danh sách các cuộc trò chuyện
 - getConversationById (id : String) : Hàm tìm kiếm cuộc trò chuyện theo id
 - deleteConversation(id : String) : Hàm xóa cuộc trò chuyện
 - createConversation(conversation : ConversationModel) : Hàm tạo cuộc trò chuyện
- Class AgentRepository
 - Thuộc tính
 - Database : một instance (phiên bản) của lớp FirebaseDatabase
 - Phương thức
 - fetchAgent : Hàm lấy danh sách các trợ lý
 - getAgentById (id : String) : Hàm tìm kiếm trợ lý theo id
 - deleteAgent(id : String) : Hàm xóa trợ lý
 - createAgent(agent : AgentModel) : Hàm tạo trợ lý
- Class MessageRepository
 - Thuộc tính
 - Database: một instance (phiên bản) của lớp Firestore
 - Phương thức:
 - fetchMessage() : Hàm lấy danh sách các tin nhắn
 - createMessage(message : MessageModel) : Hàm tạo tin nhắn
- Mối quan hệ giữa các lớp
 - Các Activity sẽ truy cập vào các Fragment để hiển thị các giao diện tương ứng
 - Các ViewModel thông qua các Fragment để lấy dữ liệu đáp ứng cho yêu cầu của người dùng
 - Các ViewModel gọi yêu cầu tới Repository để thực hiện yêu cầu tương ứng

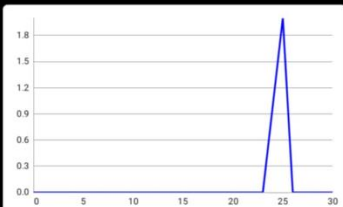
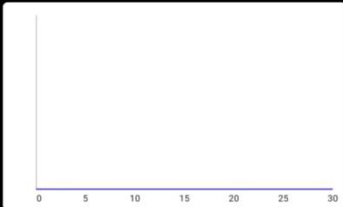
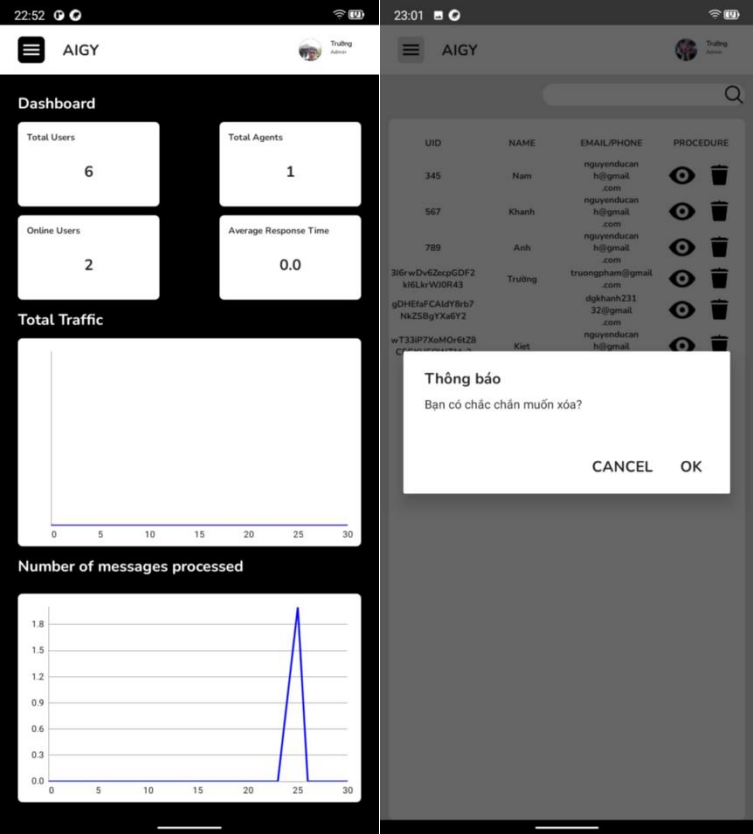
- Sau đó Repository trả về dữ liệu cho ViewModel và ViewModel gửi lại thông tin cho Fragment

- Thiết kế cơ sở dữ liệu

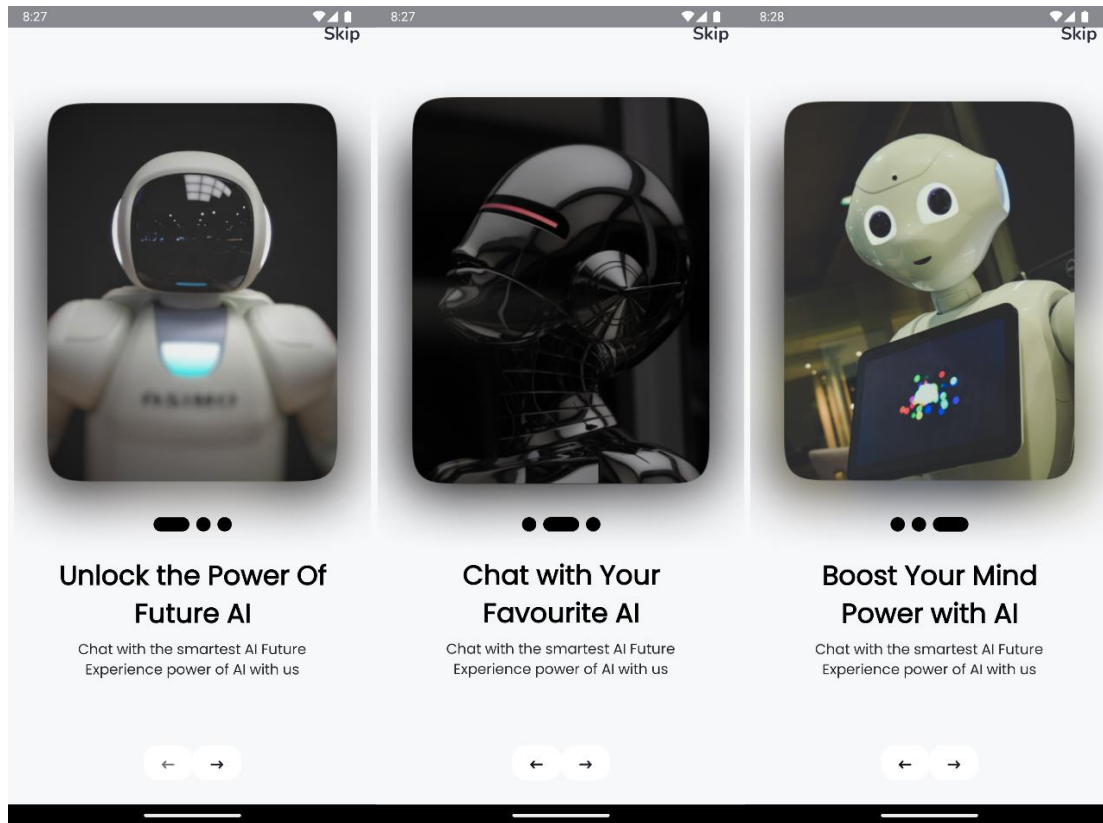


- Giao diện quản lý

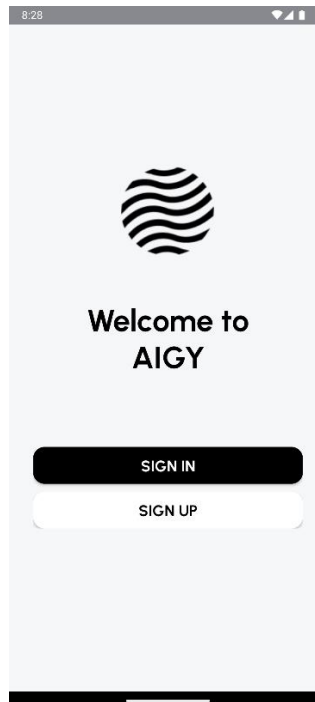




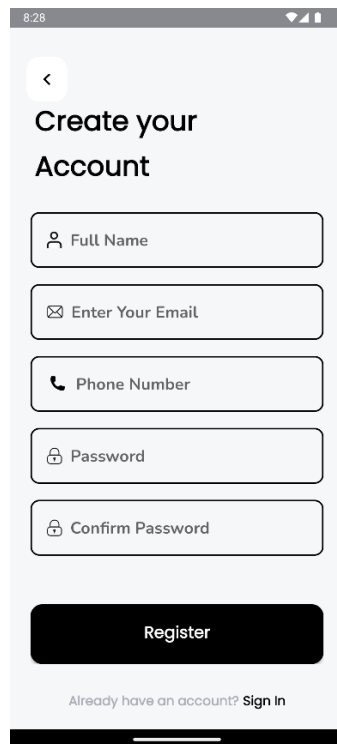
- Giao diện Splash



- Giao diện welcome

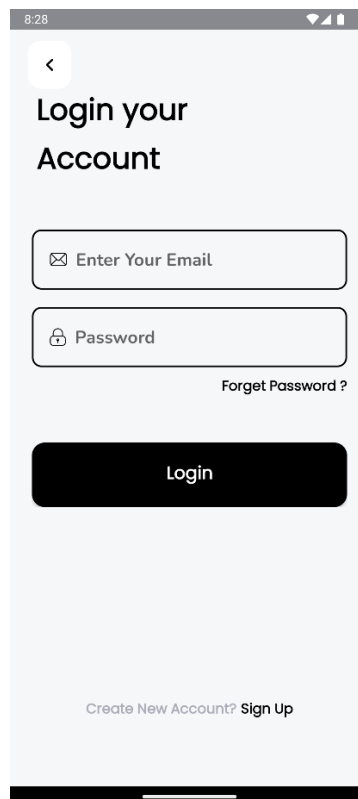


- Giao diện SignUp



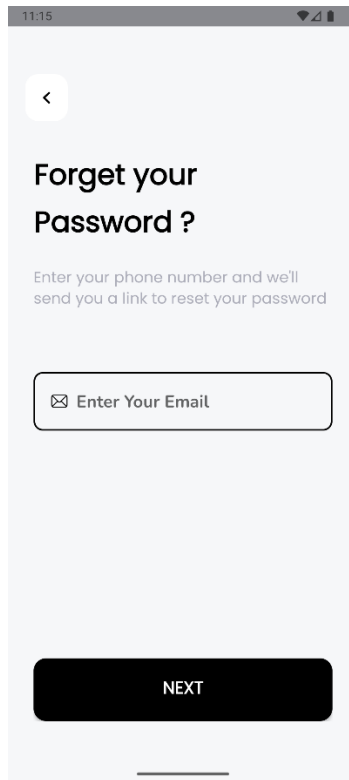
This is a mobile app screenshot of the registration screen titled "Create your Account". It features a back arrow in the top left corner. The form includes five input fields: "Full Name" (with a person icon), "Enter Your Email" (with an envelope icon), "Phone Number" (with a phone icon), "Password" (with a lock icon), and "Confirm Password" (with a lock icon). Below the fields is a prominent black "Register" button. At the bottom, there is a link that says "Already have an account? Sign In". The status bar at the top shows the time as 8:28 and standard mobile connectivity icons.

- Giao diện SignIn



This is a mobile app screenshot of the login screen titled "Login your Account". It features a back arrow in the top left corner. The form includes two input fields: "Enter Your Email" (with an envelope icon) and "Password" (with a lock icon). Below the password field is a link that says "Forget Password ?". A prominent black "Login" button is positioned below the inputs. At the bottom, there is a link that says "Create New Account? Sign Up". The status bar at the top shows the time as 8:28 and standard mobile connectivity icons.

- Giao diện ForgetPassword



11:15

<

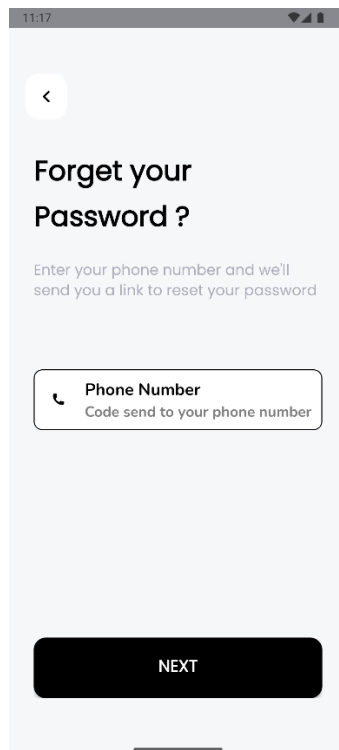
Forget your Password ?

Enter your phone number and we'll send you a link to reset your password

✉ Enter Your Email

NEXT

- Giao diện SelectMethodRecovery



11:17

<

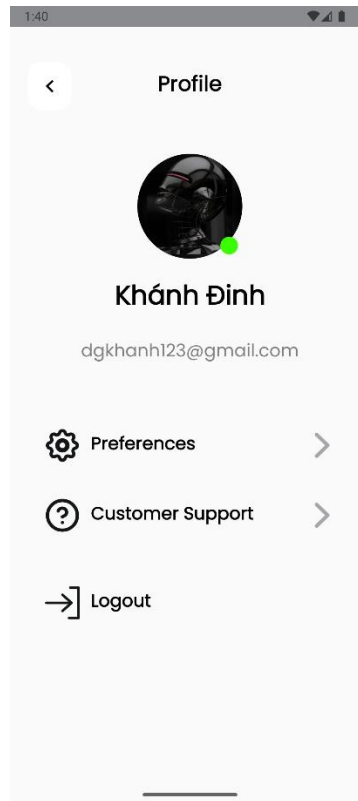
Forget your Password ?

Enter your phone number and we'll send you a link to reset your password

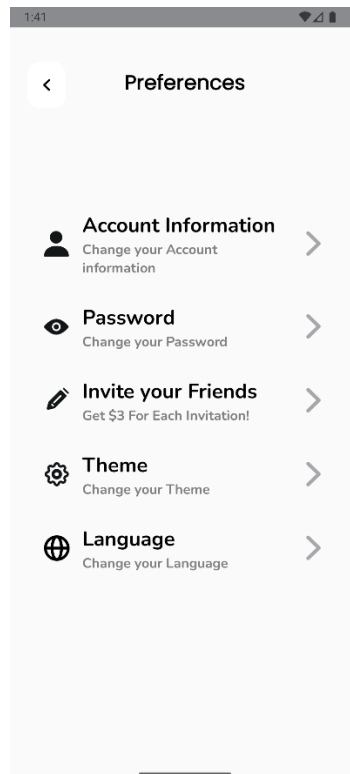
☎ Phone Number
Code send to your phone number

NEXT

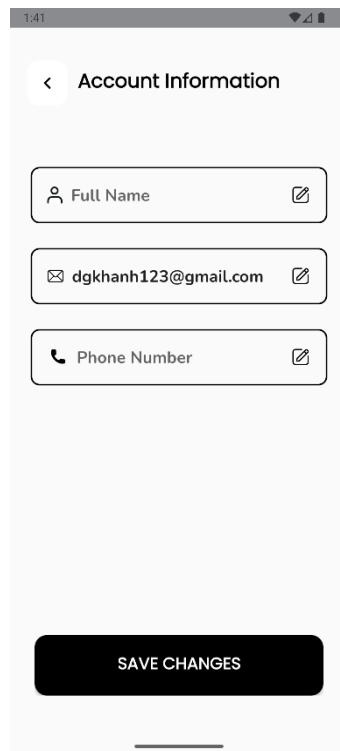
- Giao diện Profile



- Giao diện Preferences



- Giao diện AccountInformation



1:41

< Account Information

Full Name

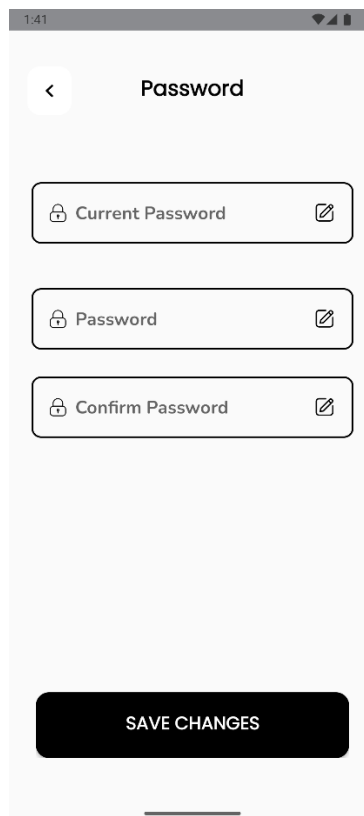
dgkhanh123@gmail.com

Phone Number

SAVE CHANGES

This is a mobile app screen titled "Account Information". It features a back arrow and the title "Account Information" at the top. Below the title, there are three input fields: "Full Name", "dgkhanh123@gmail.com", and "Phone Number". Each field has a small edit icon (pencil) to its right. At the bottom of the screen, there is a black button with the text "SAVE CHANGES".

- Giao diện Password



1:41

< Password

Current Password

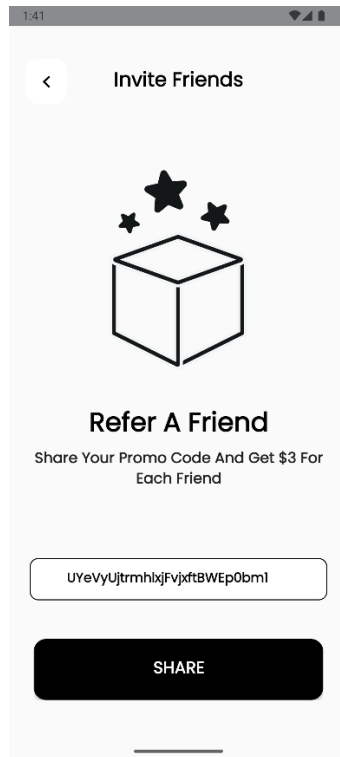
Password

Confirm Password

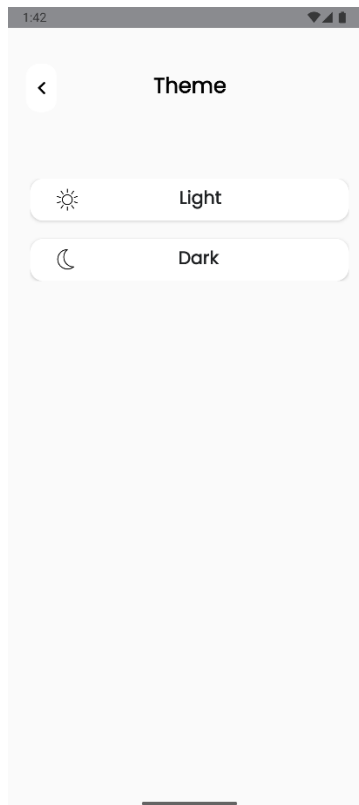
SAVE CHANGES

This is a mobile app screen titled "Password". It features a back arrow and the title "Password" at the top. Below the title, there are three input fields: "Current Password", "Password", and "Confirm Password". Each field has a small edit icon (pencil) to its right. At the bottom of the screen, there is a black button with the text "SAVE CHANGES".

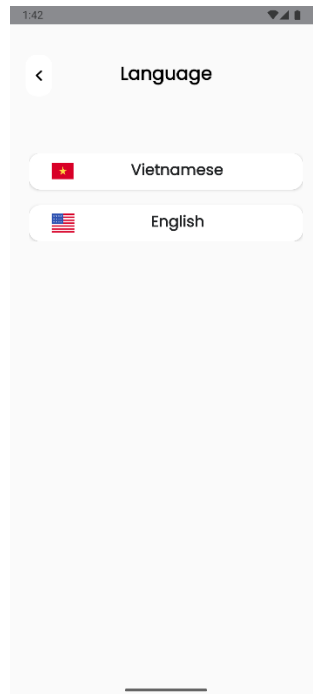
- Giao diện InviteYourFriend



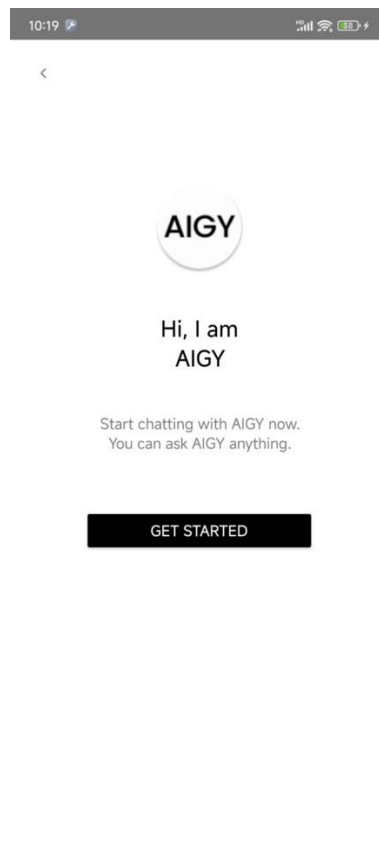
- Giao diện Theme



- Giao diện Language



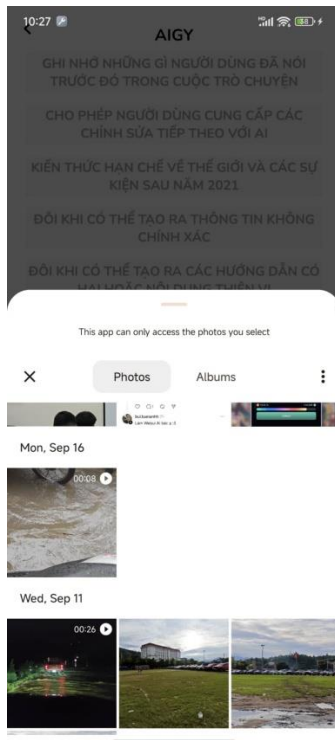
- Giao diện WelcomeAgent



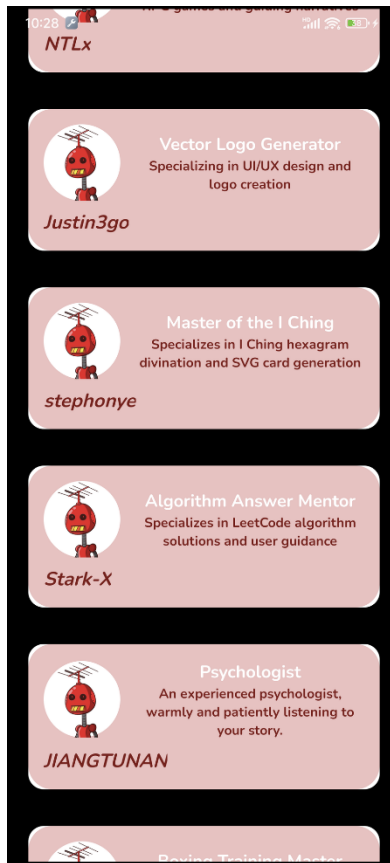
- Giao diện Chat



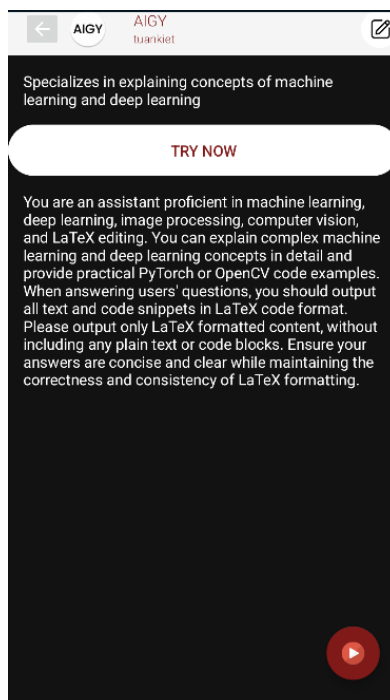
- Giao diện AttachFile



- Giao diện Agent



- Giao diện AgentDetail



CHƯƠNG 3: KẾT QUẢ THỰC HIỆN

3.1. Công nghệ đã sử dụng

- Ngôn ngữ lập trình: Kotlin.
- Công cụ: Android Studio
- Thư viện: Retrofit, Gson, Glide,...

3.2. Tiến độ thực hiện

- Link github dự án: [GitTuanKiet/CSE441_PROJECT](https://github.com/GitTuanKiet/CSE441_PROJECT)

KẾT LUẬN

Chatbot AIGY mang lại nhiều lợi ích trong việc tự động hóa và tối ưu hóa các quy trình giao tiếp của doanh nghiệp. Với khả năng phản hồi nhanh chóng và xử lý một lượng lớn thông tin từ nhiều nguồn, chatbot giúp tiết kiệm thời gian, nâng cao hiệu suất làm việc và giảm chi phí cho các công việc lặp đi lặp lại. Hơn nữa, chatbot AIGY có thể hỗ trợ người dùng 24/7, cải thiện trải nghiệm khách hàng và tăng mức độ hài lòng thông qua các phản hồi liên tục, chính xác.

Tuy nhiên, chatbot AIGY cũng có những hạn chế nhất định. Một trong số đó là khả năng hiểu ngữ cảnh và ý định phức tạp của người dùng còn nhiều hạn chế, dễ dẫn đến các câu trả lời không phù hợp hoặc thiếu chính xác. Ngoài ra, việc phát triển chatbot yêu cầu tài nguyên lớn về dữ liệu, kỹ thuật cũng như chi phí, chưa kể các vấn đề về bảo mật thông tin người dùng vẫn là một thách thức lớn. Đối với các lĩnh vực phức tạp, chatbot AIGY khó có thể thay thế hoàn toàn sự hỗ trợ từ con người.

Để khắc phục những hạn chế trên và nâng cao tính ứng dụng, các chatbot AIGY sẽ cần được cải tiến về khả năng hiểu ngữ cảnh và học hỏi sâu hơn từ các cuộc hội thoại thực tế. Phát triển các mô hình ngôn ngữ tiên tiến cũng như tích hợp với các công nghệ phân tích dữ liệu và cảm xúc người dùng có thể giúp chatbot trở nên thông minh hơn. Bên cạnh đó, nâng cao tính bảo mật và quản lý quyền riêng tư của người dùng cũng sẽ là ưu tiên hàng đầu, mở ra tiềm năng ứng dụng chatbot AIGY trong nhiều lĩnh vực từ dịch vụ khách hàng, y tế đến giáo dục và thương mại điện tử.