

BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN



ĐỀ TÀI THỰC TẬP CƠ SỞ  
XÂY DỰNG ỨNG DỤNG CHAT VỚI FLUTTER

Sinh viên:

Mã sinh viên:

Lớp:

Giảng viên hướng dẫn:

Chử Thị Mai

B21DCCN082

D21CQCN10-B

Nguyễn Mạnh Hùng

HÀ NỘI - 2/2024



# MỤC LỤC

MỞ ĐẦU -----	2
CHƯƠNG 1: GIỚI THIỆU ỦNG DỤNG CHAT DÙNG FLUTTER -----	3
1.1. Giới thiệu về nhu cầu của việc sử dụng ứng dụng chat -----	3
1.2. Phân tích yêu cầu và các tính năng cơ bản của ứng dụng chat -----	3
1.2.1. Phân tích yêu cầu -----	3
1.2.2. Tính năng cơ bản -----	3
1.3. Phát triển ứng dụng chat sử dụng Flutter -----	4
CHƯƠNG 2: GIỚI THIỆU VỀ CÔNG NGHỆ FLUTTER -----	5
2.1. Giới thiệu về Flutter -----	5
2.1.1. Lịch sử hình thành và phát triển Flutter -----	5
2.1.2. Flutter -----	6
2.2. Ngôn ngữ Dart -----	7
2.2.1. Giới thiệu Dart -----	7
2.2.2. Lý do Flutter sử dụng Dart -----	7
2.2.3. Cách Dart hoạt động -----	8
2.2.4. Dart được thiết kế theo hướng đối tượng -----	8
2.3. Ưu điểm của Flutter -----	10
2.3.1. Được hỗ trợ bởi Google -----	10
2.3.2. Sử dụng một mã nguồn để điều khiển tất cả -----	10
2.3.3. Hiệu suất cao -----	11
2.3.4. Kiểm soát hoàn toàn giao diện người dùng (UI) -----	11
2.3.5. Mã nguồn mở -----	12
2.3.6. Tài nguyên và công cụ dành cho nhà phát triển -----	12
2.3.7. So sánh Flutter với các framework hiện có -----	14
2.4. Cấu trúc của Flutter -----	16
2.4.1. Widget -----	16
2.4.2. StatelessWidget -----	18
2.4.3. StatefulWidget -----	19
2.4.4. Các Widget cơ bản thường sử dụng -----	22
CHƯƠNG 3: ÁP DỤNG FLUTTER ĐỂ XÂY DỰNG ỦNG DỤNG CHAT -	24
3.1. Tài liệu đặc tả chung -----	24
3.1.1. Danh sách các từ chuyên môn trong lĩnh vực của ứng dụng -----	24
3.1.2. Mục đích -----	25
3.1.3. Mô tả hệ thống chi tiết bằng ngôn ngữ tự nhiên -----	25
3.1.4. Sơ đồ tổng quan các usecase của toàn hệ thống -----	28
3.1.5. Mô tả hệ thống chi tiết cho các module -----	28
3.2. Tài liệu pha phân tích -----	36
3.2.1. Viết scenario -----	36
3.2.2. Mô hình hóa các lớp -----	44
3.3. Tài liệu pha thiết kế -----	67
3.3.1. Thiết kế CSDL -----	67
3.3.2. Thiết kế mô hình MVC -----	68
3.4. Pha cài đặt -----	75
3.4.1. Các công cụ và thư viện đã sử dụng trong dự án -----	75

3.4.2. Cây thư mục -----	76
3.4.3. Cơ sở dữ liệu trên Firebase -----	77
3.4.4. Cài đặt -----	78
3.4.5. Các nền tảng có thể chạy -----	85
3.4.6. Public dự án -----	85
<b>KẾT LUẬN -----</b>	<b>86</b>
1. Kết quả đạt được -----	86
2. Khó khăn trong quá trình làm -----	86
3. Hướng phát triển -----	86
<b>TÀI LIỆU THAM KHẢO -----</b>	<b>88</b>
Tài liệu tiếng anh -----	88
Tài liệu trên web -----	88

## MỞ ĐẦU

Sự phát triển vượt trội của Internet trên toàn cầu đã đem đến một cuộc cách mạng về công nghệ trên toàn thế giới. Song song với đó ai cũng sở hữu cho mình ít nhất một thiết bị điện tử như: điện thoại, laptop, ipad,... Do đó, nhu cầu về việc sử dụng các dịch vụ và tiện ích thông qua các ứng dụng ngày càng tăng cao. Lập trình ứng dụng cũng vì thế mà phát triển không ngừng và yêu cầu phải đáp ứng được với nhiều nền tảng khác nhau. Để làm được điều đó, Google đã tạo ra một SDK phát triển ứng dụng di động là Flutter.

Ứng dụng chat hiện nay là một ứng dụng quen thuộc với tất cả mọi người dùng để gửi và nhận tin nhắn, thậm chí còn có thể sử dụng để gọi video call. Với mục đích nghiên cứu và áp dụng những điều đã học được trên trường, ứng dụng chat là một lựa chọn phù hợp.

Ứng dụng chat sử dụng Flutter có thể được cài đặt trên nhiều nền tảng như: Android, IOS, macOS, Window, Linux. Người dùng có thể dễ dàng cài đặt và sử dụng ứng dụng. Với ứng dụng chat, họ có thể nhắn tin trao đổi với nhau, nhận thông báo khi có tin nhắn đến,...

Nội dung của đề tài sẽ được chia thành 3 chương như sau:

- + *Chương 1: Giới thiệu ứng dụng chat dùng Flutter* phân tích yêu cầu, ý tưởng về các tính năng, cơ hội và thách thức khi phát triển ứng dụng với Flutter.
- + *Chương 2: Giới thiệu về công nghệ Flutter* tìm hiểu về ngôn ngữ áp dụng, ưu điểm, cấu trúc lập trình của Flutter
- + *Chương 3: Áp dụng Flutter để xây dựng ứng dụng chat* chi tiết về các bước xây dựng, thiết kế ứng dụng chat

## CHƯƠNG 1: GIỚI THIỆU ỦNG DỤNG CHAT DÙNG FLUTTER

### **1.1. Giới thiệu về nhu cầu của việc sử dụng ứng dụng chat**

Hiện nay xu hướng trò chuyện từ xa ngày càng trở nên phổ biến nên những ứng dụng chat là một công cụ hữu ích để liên lạc từ xa. Ngoài việc dùng để liên lạc qua hình thức nhắn tin, các ứng dụng chat hiện nay còn có các tính năng khác như: gọi video, gửi file, gửi link, gửi định vị,...

Hiện nay, trên thế giới có rất nhiều ứng dụng chat được tạo ra để phục vụ cho nhu cầu của con người. Mỗi ứng dụng đều có những thế mạnh riêng. Và đề tài này sẽ tham khảo 2 ứng dụng chat người Việt Nam sử dụng nhiều nhất là Zalo và Messenger.

Zalo và Messenger đều là 2 ứng dụng trò chuyện. Chúng đều có các tính năng cơ bản như: nhắn tin, gọi điện, gửi file, gửi voice, tạo nhóm chat. Tuy nhiên, chúng cũng có những đặc điểm và những tính năng khác nhau phục vụ cho các yêu cầu khác nhau. Ví dụ như Zalo có thêm tính năng thanh toán bằng ví Zalo Pay, hay yêu cầu để quản lý các tài khoản khác nhau thông qua số điện thoại trong khi đó Messenger quản lý người dùng qua email. Zalo thích hợp để nhắn tin, trò chuyện với bạn bè, người thân. Messenger thích hợp dùng để nhắn tin để hỏi thông tin gì đó vì chỉ cần biết tên tài khoản là có thể tìm ra người đó và trò chuyện ngay lập tức.

Trong đề tài này sẽ tạo ra một ứng dụng chat đơn giản với các tính năng cơ bản mà mọi ứng dụng chat đều phải có là: tạo tài khoản, đăng nhập, đăng xuất, nhắn tin, gửi đính kèm hình ảnh, file, tạo group chat, gửi vị trí.

### **1.2. Phân tích yêu cầu và các tính năng cơ bản của ứng dụng chat**

#### **1.2.1. Phân tích yêu cầu**

Bài toán đặt ra là tạo một ứng dụng chat dùng cho doanh nghiệp. Trong đó có 2 đối tượng chính là Admin và User. Với mỗi đối tượng sẽ có màn hình giao diện hiển thị khác nhau và các tính năng khác nhau được cấp quyền cho mỗi loại người dùng. Ngoài ra còn có đối tượng Leader để quản lý nhóm trong các cuộc trò chuyện nhóm

#### **1.2.2. Tính năng cơ bản**

##### *a) Các tính năng đối với User*

Đối tượng User có thể sử dụng các tính năng sau đối với ứng dụng chat này:

- Nhận và gửi tin nhắn (text, ảnh, voice)
- Gọi video
- Thêm bạn mới
- Tìm kiếm bạn bè

- Xem danh sách thành viên trong nhóm mình tham gia

*b) Các tính năng đối với Leader*

Đối tượng Leader có thể sử dụng các tính năng sau đối với ứng dụng chat này:

- Có các tính năng tương tự User
- Ngoài ra còn có thể quản lý nhóm mà mình làm leader (thêm/xóa thành viên, chuyển quyền trưởng nhóm cho người khác trong nhóm)

*c) Các tính năng đối với Admin*

Đối tượng Admin có thể sử dụng các tính năng sau đối với ứng dụng chat này:

- Quản lý tài khoản người dùng (thêm, xóa, sửa, tìm kiếm)
- Quản lý nhóm (tạo nhóm, thêm/xóa nhóm, tìm kiếm, thêm/xóa thành viên, chuyển quyền trưởng nhóm)

### **1.3. Phát triển ứng dụng chat sử dụng Flutter**

Flutter là một framework phát triển ứng dụng di động và được phát hành bởi Google. Flutter có những ưu điểm sau:

- Chạy được trên nhiều nền tảng khác nhau như: iOS, Android, macOS, web, Linux, Windows.
- Hiệu suất cao
- Kiểm soát hoàn toàn giao diện người dùng
- Cộng đồng lớn

Những ưu điểm trên của Flutter làm cho nó phù hợp để xây dựng ứng dụng chat. Để hiểu rõ hơn về Flutter có thể tìm hiểu chi tiết trong chương 2 của đề tài này.

## CHƯƠNG 2: GIỚI THIỆU VỀ CÔNG NGHỆ FLUTTER

### 2.1. Giới thiệu về Flutter

#### 2.1.1. Lịch sử hình thành và phát triển Flutter

Dựa vào Chapter 1 trong cuốn *Beginning Flutter: A Hands On Guide to App Development* của tác giả Marco L. Napoli [1] và thông tin từ trang Wikipedia [5], lịch sử hình thành và phát triển của Flutter được viết như sau:

- Phiên bản đầu tiên của Flutter được gọi là “Sky” ra mắt tại Dart Developer Summit năm 2015. Eric Seidel (giám đốc kỹ sư của Flutter tại Google) đã xây dựng một dự án thử nghiệm “Dart on mobile” phát hành trên Android Play Store. Tính năng đầu tiên mà Eric thể hiện là hình vuông quay. Mục tiêu đầu tiên của hệ thống: nhanh và phản hồi nhanh, thiết bị có thể hiện thị ổn định ở tốc độ 60 khung hình mỗi giây (Eric muốn fps=120 Hertz nhưng bị hạn chế bởi thiết bị đang sử dụng).
- Trong bài phát biểu quan trọng của Ngày nhà phát triển Google tại Thượng Hải vào 9/2018, Google đã công bố Bản xem trước bản phát hành Flutter 2, bản phát hành chính cuối cùng trước Flutter 1.0.
- Vào 04/12/2018, Flutter 1.0 đã được phát hành tại sự kiện Flutter Live, đánh dấu phiên bản ổn định đầu tiên của framework.
- Vào ngày 11/12/2019, Flutter 1.12 được phát hành tại sự kiện Flutter Interactive. Flutter tại thời điểm này hỗ trợ nền tảng hiển thị thông minh của Google là Google Home Hub. Đây là bước đầu tiên hướng tới việc hỗ trợ các ứng dụng dành cho máy tính để bàn với Chrome OS.
- Ngày 06/05/2020, bộ công cụ phát triển phần mềm Dart ( SDK ) phiên bản 2.8 và Flutter 1.17.0 đã được phát hành, bổ sung hỗ trợ cho Metal API giúp cải thiện hiệu suất trên thiết bị iOS khoảng 50%, cũng như các tiện ích Material mới và theo dõi mạng công cụ phát triển.
- Ngày 03/03/2021, Google đã phát hành Flutter 2 trong sự kiện Flutter Engage trực tuyến. Bản cập nhật lớn này mang đến sự hỗ trợ chính thức cho các ứng dụng dựa trên web với trình kết xuất Canvas Kit mới và các tiện ích dành riêng cho web, hỗ trợ ứng dụng máy tính để bàn truy cập sớm cho Windows , macOS và Linux cũng như API bổ sung vào ứng dụng được cải tiến.
- Vào ngày 08/09/2021, Dart 2.14 và Flutter 2.5 được Google phát hành. Bản cập nhật mang đến những cải tiến cho chế độ toàn màn hình của Android và phiên bản Material Design mới nhất của Google có tên Material You.
- Vào ngày 12/05/2022, Google đã công bố phát hành Flutter 3 và Dart 2.17. Bản cập nhật này đã mở rộng tổng số nền tảng được hỗ trợ lên sáu, bao gồm hỗ trợ ổn định cho Linux và macOS trên cả bộ xử lý Intel và Apple Silicon.

- Vào 30/08/2022, Flutter 3.3 được công bố. Bản phát hành này có tính năng tương tác Objective-C và Swift cũng như bản xem trước ban đầu của công cụ kết xuất mới có tên "Impeller" nhằm mục đích giảm tình trạng giật hình do quá trình biên dịch shader<sup>1</sup> gây ra.

- Vào 25/01/2023, Flutter 3.7 được công bố.

### 2.1.2. *Flutter*

Theo Marco L. Napoli [1], Flutter là:

- Flutter là framework<sup>2</sup> giao diện người dùng có thể di động của Google để xây dựng các ứng dụng hiện đại, native<sup>3</sup> và phản ứng cho iOS và Android. Google cũng đang nghiên cứu tính năng nhúng máy tính để bàn Flutter và Flutter cho Web (Hummingbird) cũng như các thiết bị nhúng khác (Raspberry Pi, già đình, ô tô, v.v.).

- Flutter là một dự án nguồn mở được lưu trữ trên [GitHub](#) với sự đóng góp của Google và cộng đồng.

- Flutter sử dụng Dart, một ngôn ngữ hướng đối tượng hiện đại để biên dịch thành mã native ARM<sup>4</sup> và mã JavaScript<sup>5</sup> sẵn sàng sản xuất.

- Flutter xây dựng giao diện người dùng để phản ánh trạng thái của ứng dụng. Khi trạng thái (dữ liệu) thay đổi, giao diện người dùng sẽ được tải lại và Flutter sẽ tạo một phiên bản mới của widget.

- Flutter cung cấp cho nhà phát triển các công cụ để tạo ra các ứng dụng đẹp mắt và chuyên nghiệp cũng như khả năng tùy chỉnh mọi khía cạnh của ứng dụng: có thể thêm các animation<sup>6</sup> mượt mà, phát hiện cử chỉ và hành vi phản hồi nhanh cho giao diện người dùng.

- Các ứng dụng Flutter mang lại hiệu suất tối ưu cho cả nền tảng iOS và Android.

- Flutter sử dụng tính năng hot reload để làm mới ứng dụng đang chạy trong mili giây khi thay đổi mã nguồn để thêm các tính năng mới hoặc sửa đổi các tính năng hiện có mà vẫn giữ nguyên trạng thái ứng dụng và các giá trị dữ liệu trên màn hình.

---

<sup>1</sup> shader là một chương trình máy tính tính toán mức độ sáng , tối và màu sắc thích hợp trong quá trình hiển thị 3D. [\[Wikipedia\]](#)

<sup>2</sup> framework là một tập hợp các thư viện hoặc lớp có thể tái sử dụng cho một hệ thống phần mềm hoặc hệ thống con. [\[Wikipedia\]](#)

<sup>3</sup> native là ứng dụng được viết bằng ngôn ngữ lập trình và công nghệ phát triển cho nền tảng cụ thể mà nó được thiết kế để chạy trên đó. [\[Wikipedia\]](#) VD: Swift được phát triển riêng cho iOS.

<sup>4</sup> mã native ARM là mã máy hoặc mã nhị phân biên dịch dành riêng cho kiến trúc ARM sử dụng trong các thiết bị nhúng, điện thoại di động, máy tính bảng,... [\[Wikipedia\]](#)

<sup>5</sup> mã JavaScript được thực thi để chạy ứng dụng web hoặc trang web.

<sup>6</sup> animation ở đây là sự chuyển động của các widget trong app.

## 2.2. Ngôn ngữ Dart

### 2.2.1. Giới thiệu Dart

Ngôn ngữ lập trình Dart lần đầu tiên được Google công bố tại hội nghị GOTO vào năm 2011 và Dart 1.0 được phát hành vào cuối năm 2013. Ban đầu Dart được xem như một sự thay thế cho JavaScript (ngôn ngữ lập trình web chính), sự hấp dẫn của Dart đối với các nhà phát triển là tương đối thấp. Tuy nhiên, nhờ sự xuất hiện của Flutter và sự phụ thuộc của nó vào Dart, ngôn ngữ lập trình Dart đã có lượng sử dụng tăng lên rất nhiều. [4]

Dart là ngôn ngữ lập trình hướng đối tượng (OOP), mã được viết theo phong cách class-base giống java và sử dụng cú pháp kiểu C.

### 2.2.2. Lý do Flutter sử dụng Dart

Kể từ khi thành lập, mục tiêu chính của Flutter là trở thành một giải pháp thay thế hiệu suất cao cho các framework đa nền tảng hiện có và cải thiện trải nghiệm của nhà phát triển thiết bị di động. Do đó Flutter phải chọn 1 ngôn ngữ nào đó để đáp ứng được những điều trên và Dart là sự lựa chọn thích hợp nhất. Theo Thomas Bailey & Alessandro Biessek [4] là có 6 lý do:

- **Trình biên dịch Dart:** Dart đủ linh hoạt để cung cấp các cách chạy code khác nhau, vì vậy Flutter sử dụng biên dịch Dart trước thời gian (ahead of time - AOT) với hiệu suất được ưu tiên hàng đầu khi biên dịch phiên bản phát hành của ứng dụng, và nó sử dụng biên dịch ngay lập tức (just in time - JIT) với quá trình biên dịch mã trong thời gian phát triển dưới một giây, nhằm mục tiêu là cung cấp phản hồi nhanh cho code khi thay đổi. Dart IT và AT đề cập đến khi giai đoạn biên dịch diễn ra. Trong AOT, mã được biên dịch trong quá trình xây dựng và trước khi chạy mã; trong IT, mã được biên dịch trong quá trình chạy.

- **Hiệu suất cao:** Do hỗ trợ biên dịch AT của Dart, Flutter không yêu cầu cầu nối chậm giữa các môi trường hệ thống khác nhau, điều này giúp ứng dụng Flutter phản hồi nhanh và cho phép khởi động nhanh.

- **Thu gom rác<sup>7</sup>:** Flutter sử dụng luồng theo functional-style<sup>8</sup> với các đối tượng tồn tại ngắn hạn, và điều này đồng nghĩa với việc có nhiều phân bổ ngắn hạn. Thu gom rác của Dart hoạt động mà không cần khóa, giúp tăng tốc quá trình phân bổ nhanh chóng.

- **Dễ học:** Dart là một ngôn ngữ linh hoạt, mạnh mẽ, hiện đại và tiên tiến. Ngôn ngữ đã được điều chỉnh khi Flutter trở nên phổ biến hơn, với nhiều cú pháp phức tạp và những thay đổi cơ bản về thiết kế, những điều này thực sự giúp ích cho việc tạo ứng dụng Flutter.

- **Giao diện người dùng theo phong cách khai báo:** Trong Flutter, bạn sử dụng một phong cách khai báo để xếp các widget, điều này có nghĩa là các

<sup>7</sup> thu gom rác là một kỹ thuật tự động trong lập trình máy tính, nó tự động phát hiện và giải phóng bộ nhớ không còn được sử dụng nữa.

<sup>8</sup> functional-style là một phương pháp lập trình trong đó các hàm được xem là các đối tượng độc lập, có thể truyền vào đối số cho các hàm khác và có thể được trả về từ các hàm khác.

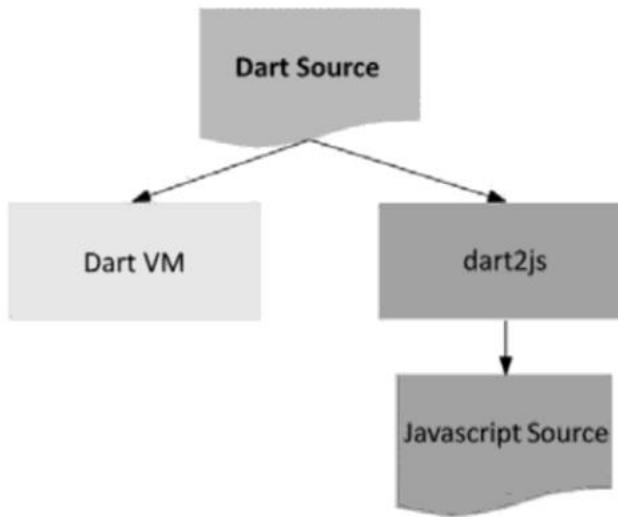
widget là không thay đổi và chỉ là "bản thiết kế" nhẹ. Để thay đổi giao diện người dùng, một widget xây dựng lại chính nó và các cây con của nó.

- **Cú pháp Dart cho việc bố trí:** Khác với nhiều framework có cú pháp riêng biệt cho việc bố trí, trong Flutter, bố trí được chỉ định trực tiếp trong mã Dart. Điều này mang lại tính linh hoạt cao hơn và giảm bớt gánh nặng tư duy cho các nhà phát triển. Flutter cũng có các công cụ tuyệt vời để gỡ lỗi bố trí cũng như hiệu suất hiển thị.

### 2.2.3. Cách Dart hoạt động

Để hiểu được sự linh hoạt của Dart thì cần phải biết cách mà Dart chạy code. Điều này được thực hiện thông qua 2 cách:

- Máy ảo Dart (VMs)
- Biên dịch thành mã JavaScript



Hình 2.1. Các cách để chạy ứng dụng Dart [4]

Gốc của biểu đồ là mã Dart. Điều quan trọng là code và các lựa chọn phụ thuộc đều không phụ thuộc vào cách chạy ứng dụng, nghĩa là không cần thay đổi code để chạy ứng dụng cho từng sự lựa chọn khác nhau.

### 2.2.4. Dart được thiết kế theo hướng đối tượng

Tương tự như hầu hết các ngôn ngữ lập trình hiện đại, Dart được thiết kế để hỗ trợ hướng đối tượng (OOP). Đơn giản là, trong lập trình hướng đối tượng, chúng ta làm việc với các "đối tượng" - chứa dữ liệu (gọi là fields<sup>9</sup>) và đoạn mã (gọi là methods<sup>10</sup>).

Các đối tượng được tạo ra từ các lớp (class) được xác định các thuộc tính và phương thức mà một đối tượng sẽ có. Việc tuân thủ theo các nguyên tắc của lập trình hướng đối tượng (bao đóng<sup>11</sup>, kế thừa<sup>12</sup>, tổ hợp<sup>13</sup>, trừu tượng<sup>14</sup> và đa hình<sup>15</sup>)

<sup>9</sup> field là trường dữ liệu, các thuộc tính của 1 đối tượng (VD: mèo có các thuộc tính về giống loài, màu lông,...)

<sup>10</sup> method là phương thức, thường để chỉ cách hành động của đối tượng (VD: mèo có thể có các phương thức ăn, bắt chuột,...)

<sup>11</sup> bao đóng là che giấu thông tin và cung cấp các giao diện công khai để truy cập đến dữ liệu và chức năng của đối tượng.

<sup>12</sup> kế thừa là cho phép một lớp kế thừa các tính năng của lớp khác, giúp tái sử dụng mã và xây dựng các cấu trúc phân cấp.

sẽ giúp cho code trở nên dễ bảo trì, linh hoạt và tái sử dụng được.

Bốn tính chất OOP của Dart được Thomas Bailey và Alessandro Biessek [4] trình bày như sau:

#### a) *Tính bao đóng (Encapsulation)*

Trong Dart, không có các hạn chế truy cập được chỉ định một cách rõ ràng, như các từ khóa được sử dụng trong Java (protected, private và public). Trong Dart, sự đóng gói xảy ra ở cấp thư viện thay vì ở cấp lớp.

- Dart tạo ra các getter và setter ngầm cho tất cả fields trong một lớp, do đó có thể xác định cách dữ liệu được truy cập bởi consumers<sup>16</sup> và cách nó thay đổi.

- Trong Dart, nếu một định danh (lớp, thành viên của lớp, hàm cấp độ cao, hoặc biến) bắt đầu bằng dấu gạch dưới (\_), nó là riêng tư (private) với thư viện của nó.

#### b) *Tính kế thừa và tổ hợp (Inheritance and composition)*

Kế thừa cho phép chúng ta mở rộng một đối tượng thành các phiên bản chuyên biệt của một số kiểu trừu tượng (abstract). Trong Dart, chỉ cần khai báo một lớp, chúng ta đã ngầm mở rộng class Object.

- Dart hỗ trợ đơn kế thừa trên một lớp, tương tự như Java và hầu hết các ngôn ngữ khác, điều đó có nghĩa là một lớp chỉ có thể kế thừa trực tiếp từ một lớp duy nhất tại một thời điểm.

- Dart có hỗ trợ đặc biệt cho mixins, có thể được sử dụng để mở rộng các chức năng của lớp mà không cần kế thừa trực tiếp, mô phỏng nhiều kế thừa và tái sử dụng mã.

- Dart không chứa chỉ thị final class giống như các ngôn ngữ khác, nghĩa là một lớp luôn có thể được mở rộng (có thể có các lớp con).

#### c) *Tính trừu tượng (Abstraction)*

Theo sau kế thừa, trừu tượng là quá trình định nghĩa một kiểu và các đặc điểm cốt lõi của nó, đi từ các kiểu chuyên biệt đến các kiểu cha.

- Dart chứa các lớp trừu tượng cho phép định nghĩa điều gì đó làm/giới thiệu mà không cần quan tâm đến cách thức thực hiện điều này.

- Dart có khái niệm giao diện ngầm mạnh mẽ, cũng làm mọi lớp trở thành một giao diện, cho phép nó được thực hiện bởi các lớp khác mà không cần mở rộng.

<sup>13</sup> tổ hợp là cho phép sử dụng các đối tượng khác nhau để tạo thành một đối tượng mới, giúp tái sử dụng mã và tạo ra các mối quan hệ phức tạp.

<sup>14</sup> trừu tượng là tạo ra các lớp và phương thức mà chỉ mô tả chức năng chung mà không cần biết chi tiết cụ thể.

<sup>15</sup> da hình là cho phép một phương thức hoạt động khác nhau tùy thuộc vào loại đối tượng được truyền vào, giúp mã trở nên linh hoạt và tái sử dụng được.

<sup>16</sup> consumers đề cập đến các phần của code hoặc các thành phần của ứng dụng mà sử dụng các thành phần khác được cung cấp trong thư viện hoặc lớp.

#### d) *Tính đa hình (Polymorphism)*

Đa hình được đạt được thông qua kế thừa và có thể được coi là khả năng của một đối tượng hoạt động giống như một đối tượng khác (VD: kiểu int cũng là một kiểu num).

- Dart cho phép ghi đè các phương thức của lớp cha để thay đổi hành vi gốc của chúng.

- Dart không cho phép nạp chồng<sup>17</sup> (overloading).

Dưới đây là những thành phần chính của lập trình hướng đối tượng trong ngôn ngữ Dart:

- Lớp (Class): Đây là bản thiết kế để tạo ra một đối tượng.

- Giao diện (Interface): Đây là định nghĩa hợp đồng với một tập hợp các phương thức có sẵn trên một đối tượng. Mặc dù Dart không có kiểu giao diện (interface) rõ ràng, nhưng chúng ta có thể đạt được mục đích của giao diện bằng cách sử dụng các lớp trừu tượng.

- Lớp liệt kê (Enumerated class): Đây là một loại đặc biệt của lớp định nghĩa một tập hợp các giá trị hằng số phổ biến.

- Mixin: Đây là một cách để tái sử dụng mã của một lớp trong nhiều cấu trúc phân cấp lớp khác nhau.

### 2.3. Ưu điểm của Flutter

#### 2.3.1. Được hỗ trợ bởi Google

Flutter và Dart đều do Google tạo ra điều này đảm bảo framework có tất cả các công cụ cần thiết để thành công trong cộng đồng, với sự hỗ trợ từ nhóm Google, sự hiện diện tại các sự kiện lớn như Google I/O và đầu tư vào các cải tiến liên tục trong cơ sở mã. Từ khi ra mắt bản phát hành ổn định đầu tiên trong Sự kiện Flutter Live vào cuối năm 2018, sự phát triển của Flutter được thể hiện rõ ràng [4]:

- Hơn 200 triệu người dùng ứng dụng Flutter
- Hơn 50.000 ứng dụng Flutter trên Play Store
- Gần 500.000 nhà phát triển
- Kho phần mềm phổ biến thứ 18 trên GitHub

#### 2.3.2. Sử dụng một mã nguồn để điều khiển tất cả

Mục tiêu chính của Flutter framework là trở thành một bộ công cụ để xây dựng ứng dụng có hiệu suất, tương tác dễ dàng và tính năng tương đương với các ứng dụng native (ứng dụng tạo trực tiếp cho iOS, Android, macOS, Windows và web) chỉ bằng cách sử dụng một mã nguồn duy nhất.

---

<sup>17</sup> nạp chồng là khả năng tạo nhiều hàm có cùng tên với các cách triển khai khác nhau. [Wikipedia]

### **2.3.3. Hiệu suất cao**

Thomas Bailey và Alessandro Biessek [4] đã trình bày một số điểm về hiệu suất của Flutter: “Hiện tại, thật khó để nói rằng hiệu suất của Flutter trong thực tế luôn tốt hơn các framework khác, nhưng có thể nói rằng mục tiêu của nó là như vậy”. Một số framework hiện nay có dựa vào kết xuất<sup>18</sup> JavaScript và HTML có thể gây ra chi phí cao về hiệu suất vì mọi thứ đều được vẽ trong WebView (một thành phần trực quan giống như trình duyệt web).

Một số ứng dụng sử dụng các tiện ích gốc từ nhà sản xuất (OEM) nhưng phụ thuộc vào một cầu nối để yêu cầu API hệ điều hành để vẽ các thành phần, điều này tạo ra một chướng ngại vật trong ứng dụng vì nó cần một bước bổ sung để vẽ giao diện người dùng (UI).

Một số điểm mang lại hiệu suất tốt cho Flutter [4]:

- **Flutter sở hữu từng điểm ảnh:** Flutter kết xuất ứng dụng pixel từng pixel, tương tác trực tiếp với công cụ đồ họa Skia.

- **Không có lớp bổ sung hoặc lệnh gọi API hệ điều hành bổ sung:** Vì Flutter điều khiển quá trình vẽ ứng dụng, nên không cần các cuộc gọi bổ sung để sử dụng các tiện ích OEM, do đó không có chướng ngại vật.

- **Flutter được biên dịch thành native code:** Flutter sử dụng trình biên dịch Dart AOT để tạo ra native code. Điều này có nghĩa là không có chi phí phát sinh khi thiết lập môi trường để thông dịch mã Dart trực tiếp và nó chạy giống như một ứng dụng native, khởi động nhanh hơn so với các framework cần một loại thông dịch nào đó.

### **2.3.4. Kiểm soát hoàn toàn giao diện người dùng (UI)**

Framework Flutter tự thực hiện toàn bộ UI, vẽ các thành phần của ứng dụng trực tiếp lên canvas [4]:

- **Kiểm soát tất cả các điểm ảnh trên thiết bị:** Bằng cách kiểm soát việc vẽ UI, Flutter cho phép nhà phát triển tạo UI theo cách riêng của họ bằng cách hiển thị API Widgets phong phú và có thể mở rộng, cung cấp các công cụ có thể được sử dụng để tạo UI duy nhất mà không có nhược điểm về hiệu suất và không có giới hạn về thiết kế.

- **Bộ công cụ Platform UI:** Flutter được trang bị các gói cung cấp các widget thiết kế nền tảng, bộ Material trong Android và Cupertino trong iOS.

- **Đáp ứng được yêu cầu thiết kế UI:** Flutter cung cấp một API rõ ràng và mạnh mẽ với khả năng tái tạo các bộ cục trung thành với yêu cầu thiết kế.

- **Giao diện trông mượt mà và tự nhiên:** Flutter cố gắng cung cấp trải nghiệm UI cụ thể cho nền tảng mà ứng dụng đang chạy, vì vậy phông chữ, cử chỉ và tương tác được triển khai theo cách cụ thể cho từng nền tảng, mang lại cảm giác tự nhiên cho người dùng, giống như một ứng dụng native.

<sup>18</sup> kết xuất hay tổng hợp hình ảnh là quá trình tạo ra hình ảnh quang học hoặc không quang học từ mô hình 2D hoặc 3D từ chương trình máy tính. [Wikipedia]

### 2.3.5. Mã nguồn mở

Ngoài việc có Google hỗ trợ thì có cộng đồng đóng góp lại càng quan trọng hơn khi Flutter trở nên phổ biến hơn. Bằng cách là mã nguồn mở, cộng đồng và Google có thể cùng nhau làm các việc sau [4]:

- Hỗ trợ sửa lỗi và tài liệu thông qua sự hợp tác mã nguồn
- Tạo nội dung giáo dục mới về framework
- Hỗ trợ tài liệu và sử dụng
- Đưa ra quyết định cải tiến dựa trên phản hồi thực tế

### 2.3.6. Tài nguyên và công cụ dành cho nhà phát triển

Thomas Bailey và Alessandro Biessek [4] đã giới thiệu về một số tài nguyên và công cụ cho nhà phát triển:

- **Tài liệu và tài nguyên học tập:** Các trang web Flutter rất phong phú dành cho các nhà phát triển đến từ các nền tảng khác nhau, bao gồm các ví dụ và trường hợp sử dụng như: <https://flutter.dev/>

- **Tích hợp với các IDE:** Flutter và Dart có trải nghiệm IDE tích hợp hoàn chỉnh với Android Studio, IntelliJ và Visual Studio Code.

- **Command-line tools:** Dart có các công cụ giúp phân tích, chạy và quản lý các phần phụ thuộc, đây cũng là một phần của Flutter. Ngoài ra, Flutter còn có các lệnh hỗ trợ gỡ lỗi, triển khai, kiểm tra kết xuất bối cảnh và tích hợp với IDE thông qua plugin Dart. Dưới đây là các lệnh có thể sử dụng trong Flutter:

```
Available commands:
Flutter SDK
  bash-completion  Output command line shell completion setup scripts.
  channel          List or switch Flutter channels.
  config           Configure Flutter settings.
  doctor           Show information about the installed tooling.
  downgrade        Downgrade Flutter to the last active version for the current
                   channel.
  precache         Populate the Flutter tool's cache of binary artifacts.
  upgrade          Upgrade your copy of Flutter.

Project
  analyze          Analyze the project's Dart code.
  assemble         Assemble and build Flutter resources.
  build            Build an executable app or install bundle.
  clean             Delete the build/ and .dart_tool/ directories.
  create            Create a new Flutter project.
  drive             Run integration tests for the project on an attached device
                   or emulator.
  gen-l10n          Generate localizations for the current project.
  pub               Commands for managing Flutter packages.
  run               Run your Flutter app on an attached device.
  test              Run Flutter unit tests for the current project.

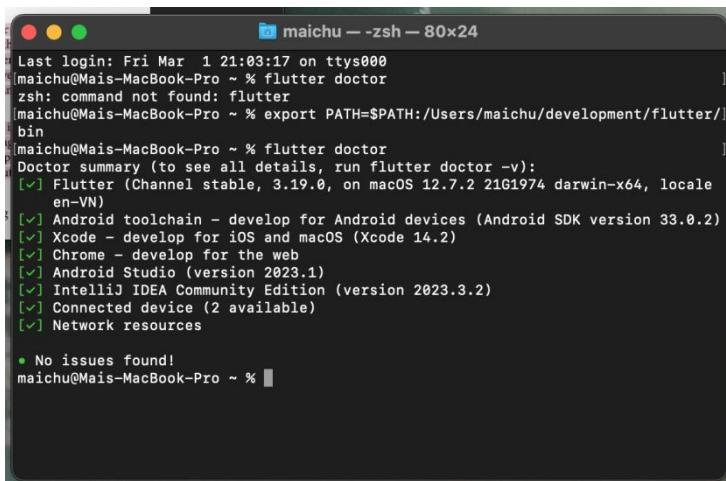
Tools & Devices
  attach            Attach to a running app.
  custom-devices   List, reset, add and delete custom devices.
  devices           List all connected devices.
  emulators         List, launch and create emulators.
  install           Install a Flutter app on an attached device.
  logs              Show log output for running Flutter apps.
  screenshot        Take a screenshot from a connected device.
  symbolize        Symbolize a stack trace from an AOT-compiled Flutter app.

Run "flutter help <command>" for more information about a command.
Run "flutter help -v" for verbose help output, including less commonly used
options.
maichu@Mais-MacBook-Pro ~ %
```

Hình 2.2. Danh sách các lệnh (commands) sử dụng trong Flutter

- **Thiết lập nhanh:** Flutter có lệnh create cho phép tạo một dự án Flutter mới với đầy đủ chức năng. IDE cũng cung cấp tùy chọn menu tạo dự án Flutter phỏng theo chức năng command-line.

- **Chẩn đoán vấn đề môi trường:** Flutter đi kèm với công cụ Flutter Doctor, đây là một công cụ dòng lệnh hướng dẫn nhà phát triển thiết lập hệ thống bằng cách chỉ ra những gì cần thiết để sẵn sàng thiết lập môi trường Flutter.



```
Last login: Fri Mar 1 21:03:17 on ttys000
maichu@Mais-MacBook-Pro ~ % flutter doctor
zsh: command not found: flutter
maichu@Mais-MacBook-Pro ~ % export PATH=$PATH:/Users/maichu/development/flutter/bin
maichu@Mais-MacBook-Pro ~ % flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.19.0, on macOS 12.7.2 21G1974 darwin-x64, locale en-VN)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Xcode - develop for iOS and macOS (Xcode 14.2)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2023.1)
[✓] IntelliJ IDEA Community Edition (version 2023.3.2)
[✓] Connected device (2 available)
[✓] Network resources

• No issues found!
maichu@Mais-MacBook-Pro ~ %
```

Hình 2.3. Chạy lệnh flutter doctor

Kết quả hiển thị khi chạy lệnh “flutter doctor” trong ảnh trên hiển thị các môi trường Flutter, Android toolchain, Xcode, Chrome, Android Studio, IntelliJ đã được thiết lập thành công.

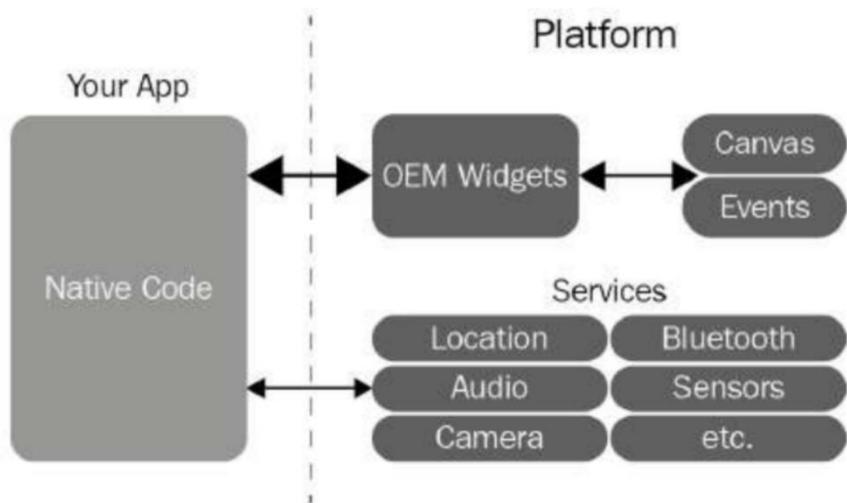
- **Hot reload:** Bằng cách kết hợp các khả năng của ngôn ngữ Dart (chẳng hạn như biên dịch JIT) và sức mạnh của Flutter, nhà phát triển có thể thấy ngay các thay đổi thiết kế được thực hiện đối với mã trong trình mô phỏng hoặc thiết bị.

### 2.3.7. So sánh Flutter với các framework hiện có

Để thấy được ưu điểm của Flutter rõ ràng hơn thì nó cần được so sánh với các nền tảng có sẵn trước đó.

#### a) Nền tảng gốc

Các framework gốc như SDK Android và iOS rất vững chắc. Chúng là sự lựa chọn ổn định nhất để phát triển ứng dụng di động, có rất nhiều ứng dụng có sẵn đã được thử nghiệm kỹ lưỡng và có cộng đồng lớn cũng như các hướng dẫn có sẵn công khai. Sơ đồ sau đây hiển thị hoạt động của các framework phát triển ứng dụng di động gốc:



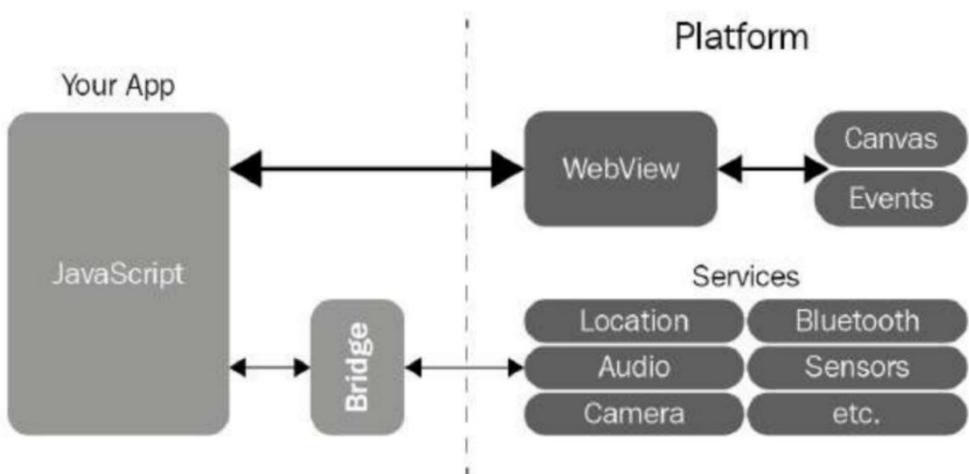
Hình 2.4. Hoạt động của các framework phát triển ứng dụng gốc [3]

Như có thể thấy trong sơ đồ, ứng dụng sử dụng nền tảng gốc giao tiếp trực tiếp với hệ thống giúp cho nó có hiệu suất cao. Tuy nhiên, nó có một nhược điểm: cần học hai ngôn ngữ khác nhau, Kotlin hoặc Java cho Android, Obj-C hoặc Swift cho iOS và SDK.

#### b) Hệ thống WebView

Theo cách tiếp cận đa nền tảng, có thể tải ứng dụng cho cả Android và iOS từ một cơ sở mã duy nhất, giống như trong Flutter.

Nhưng mỗi framework đều có nhược điểm riêng. Các framework dựa trên Cordova, Ionic, PhoneGap và WebView nói chung là các ví dụ tốt về các framework chạy trên nhiều nền tảng, và chúng là các giải pháp đặc biệt tốt cho các nhà phát triển frontend. Tuy nhiên, các framework này hiệu suất thấp và giao diện ứng dụng trong các phương pháp này được tạo thành bằng cách sử dụng WebView để hiển thị HTML. Điều này có nghĩa là ứng dụng về cơ bản là một trang web [3].



Hình 2.5. cách hoạt động của framework dựa trên WebView [3]

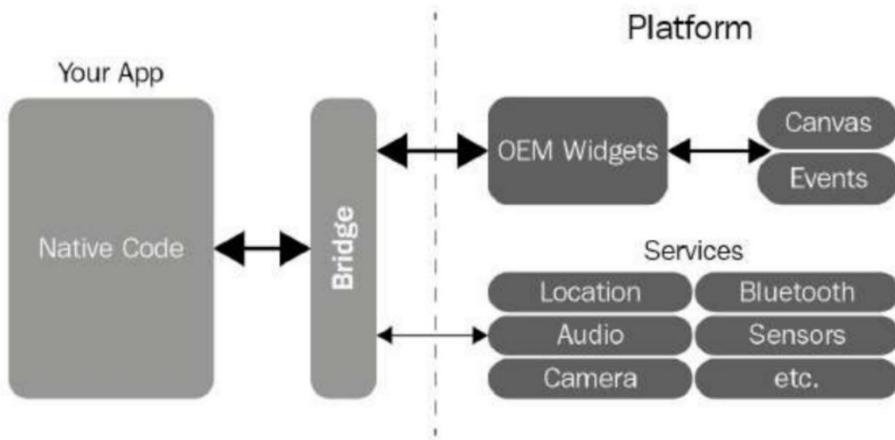
Hệ thống cần có một cầu nối để thực hiện chuyển đổi giữa JavaScript sang hệ thống gốc. Quá trình này sẽ rất chậm, tùy thuộc vào các tính năng nên gây giảm hiệu suất.

### c) Các cách tiếp cận đa nền tảng khác

Hãy xem xét 2 ví dụ sau về cách tiếp cận đa nền tảng [3]:

- Xamarin của Windows là một ví dụ cho việc phát triển đa nền tảng, điều này không thuận tiện lắm, đặc biệt là về hiệu năng và thời gian biên dịch.

- React Native có thể được coi là một trong những nền tảng đa nền tảng tốt nhất, nhưng nó phụ thuộc rất nhiều vào các thành phần OEM.

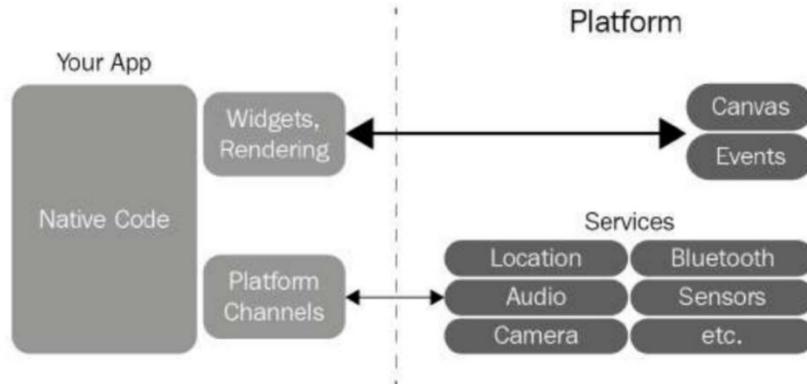


Hình 2.6. Cách hoạt động của React Native [3]

React Native mở rộng khái niệm cầu nối trong hệ thống WebView và sử dụng nó không chỉ cho các dịch vụ mà còn để xây dựng các widget. Do đó, hiệu suất của hệ thống sẽ thấp hơn cả cách tiếp cận của hệ thống WebView.

### d) Cách tiếp cận của Flutter

Flutter hoạt động tốt hơn nhiều so với các giải pháp khác, vì ứng dụng được biên dịch AOT (Ahead Of Time) thay vì JIT (Just In Time) như các giải pháp JavaScript.



Hình 2.7. Cách hoạt động của Flutter [3]

Flutter đã loại bỏ cầu nối và nền tảng OEM và thay vào đó sử dụng Widgets Rendering để làm việc với canvas và các sự kiện. Và nó sử dụng kênh nền tảng (Platform Channels) để sử dụng các dịch vụ. Ngoài ra, không khó để sử dụng API nền tảng với hệ thống nhắn tin không đồng bộ, nghĩa là nếu cần sử dụng một tính năng cụ thể của Android hoặc iOS, có thể thực hiện dễ dàng [4].

## 2.4. Cấu trúc của Flutter

Như đã giới thiệu ở trên, Flutter sử dụng widget để làm tất cả mọi thứ.

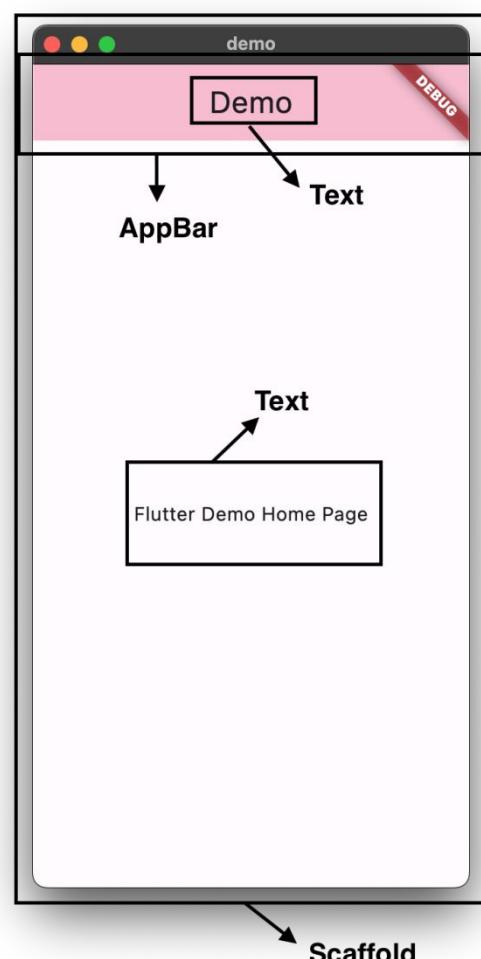
### 2.4.1. Widget

- Widget là một đối tượng dùng để cấu hình các thuộc tính bên trong nó. Khi viết một widget nó sẽ tạo ra đối tượng đó với các thuộc tính (màu sắc, kích thước,...) đã được viết trong widget.

- Widget tree là các widget được viết lồng trong nhau:

Đoạn code VD1:

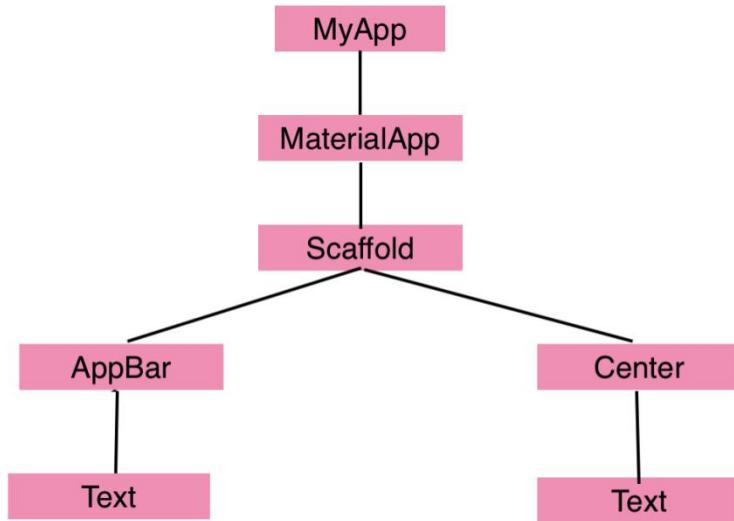
```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('Demo'),
        backgroundColor: Colors.pink[100],),
        body: const Center(
          child: Text('Flutter Demo Home Page'),
        ),
      ),
    );
  }
}
```



Hình 2.8. Chạy ứng dụng

Đoạn code trên hiển thị 1 ứng dụng đơn giản có app bar và một đoạn text hiển thị ở chính giữa ứng dụng.

Để hiểu được lý do các widget kết hợp lồng nhau tạo ra widget tree hãy xem hình dưới đây:



Hình 2.9. Widget tree cho VD1

Trong ví dụ trên các widget con được lồng vào trong các widget cha qua các thuộc tính của các widget cha như: home, body,... và phổ biến nhất trong Flutter là child và children.

Hình 2.9 cho thấy AppBar sẽ được hiển thị với một văn bản bên trong, phần thân (body) sẽ được hiển thị một đoạn văn bản ở chính giữa.

- Các widget được sử dụng trong Flutter có thể là các widget được xây dựng sẵn hoặc do lập trình viên tự tạo.
- Các thành phần quan trọng trong một widget:
  - + context: dùng để cung cấp vị trí của widget trong widget tree [2]. Nó cũng cung cấp các thông tin và dịch vụ cần thiết từ môi trường xây dựng.
  - + state: là thông tin về một cái gì đó được lưu giữ trong bộ nhớ. Trong ngữ cảnh của một ứng dụng, trạng thái là dữ liệu (thông tin) mà nó sử dụng. Trong ngữ cảnh của widget, trạng thái là dữ liệu trong widget đó [2]. Có 2 loại trạng thái là: trạng thái có thể thay đổi và trạng thái không thể thay đổi.

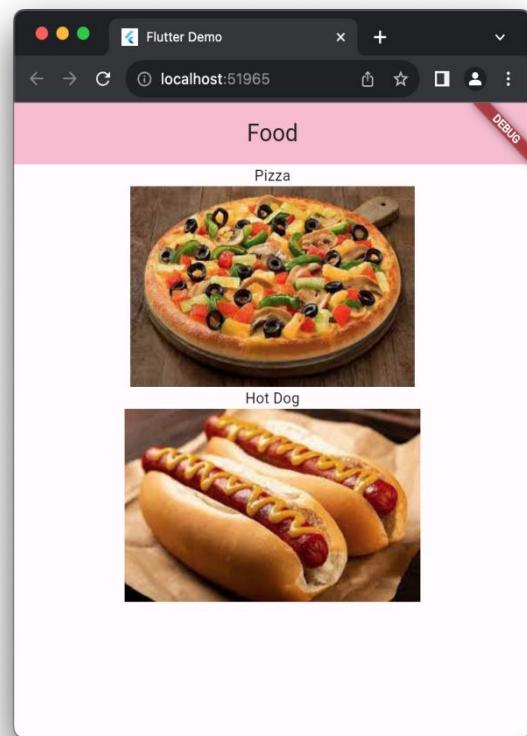
#### 2.4.2. StatelessWidget

- StatelessWidget được sử dụng khi dữ liệu trong widget đó không có gì cần thay đổi. Ví dụ như widget đó chỉ cần hiển thị một đoạn văn bản cố định, hình ảnh cố định,...

- StatelessWidget được tạo bởi widget cha qua phương thức “build”. StatelessWidget được cung cấp các thông tin cần thiết để thực hiện công việc qua phương thức build kế thừa từ widget cha. [2]

### - VD2: Hiển thị ứng dụng sử dụng StatelessWidget

```
class MyHomePage extends StatelessWidget {  
    MyHomePage({key, required this.title}) :  
    super(key: key);  
  
    final String title;  
  
    @override  
    Widget build(BuildContext context) {  
        return new Scaffold(  
            appBar: new AppBar(  
                title: new Text(this.title),  
                backgroundColor: Colors.pink[100],  
            ),  
            body: new Column(children: <Widget>[  
                FoodWidget("Pizza", "link_img1"),  
                FoodWidget("Hot Dog", "link_img2"),  
            ]);  
    }  
}  
  
class FoodWidget extends StatelessWidget {  
    FoodWidget(this.title, this.imageSrc) : super();  
    final String title;  
    final String imageSrc;  
  
    @override  
    Widget build(BuildContext context) {  
        return Center(  
            child: Column(children: <Widget>[  
                Text(title),  
                Image.network(imageSrc),  
            ]);  
    }  
}
```

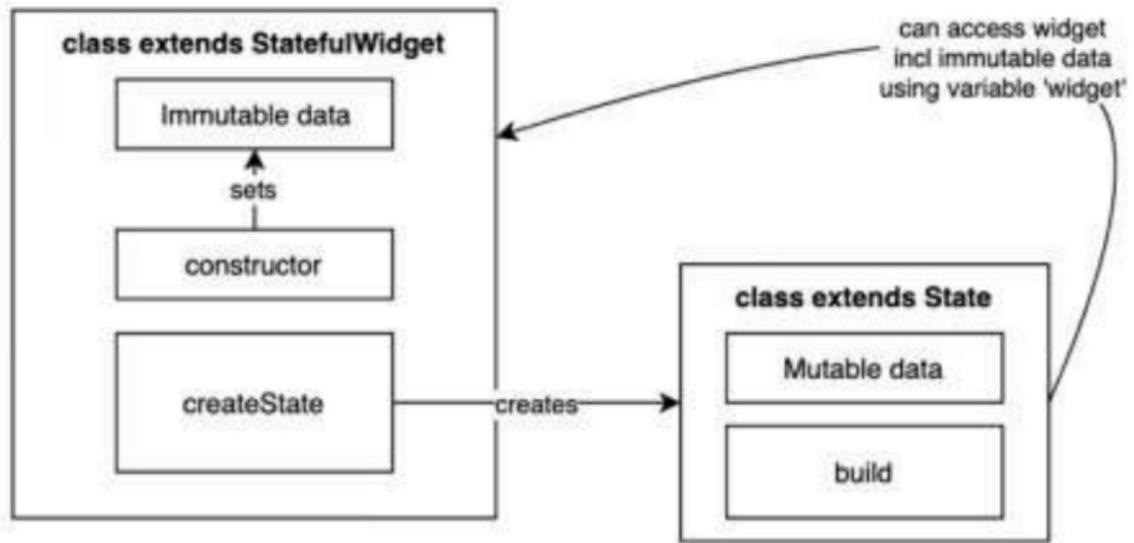


Hình 2.10. VD Stateless Widget

Trong VD trên có 2 stateless widget là FoodWidget và MyHomePage. FoodWidget hiển thị 1 tiêu đề và 1 ảnh cả 2 đều không cần thay đổi nội dung. MyHomePage cũng không cần thay đổi nội dung trong nó.

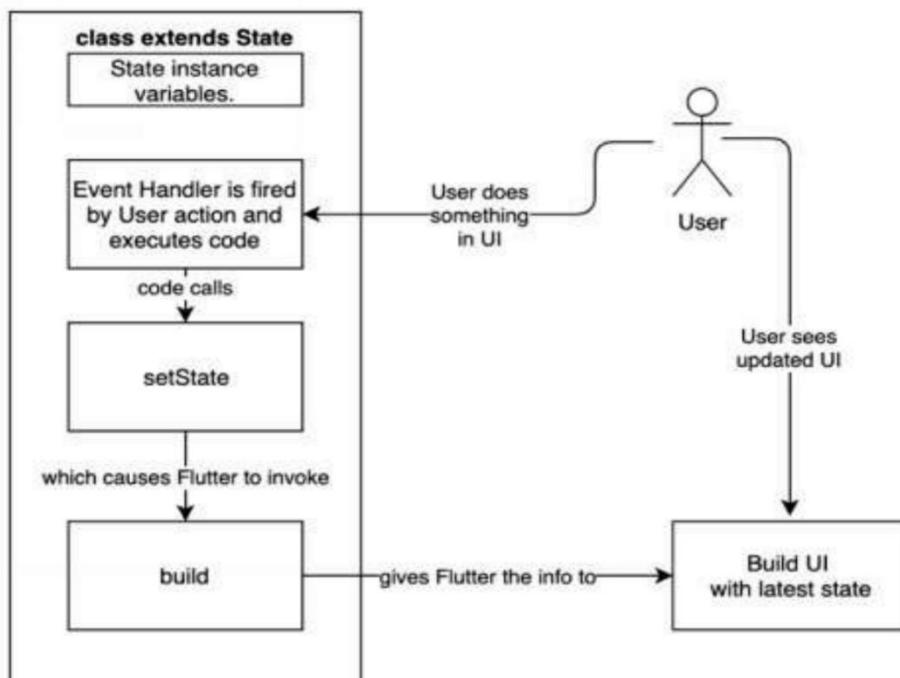
### 2.4.3. StatefulWidget

- StatefulWidget được sử dụng khi xây dựng 1 widget mà trong widget đó có chứa dữ liệu cần cập nhật lại.



Hình 2.11. Cách hoạt động của class extends StatefulWidget [2]

- Một class mở rộng từ StatefulWidget thì nó phải có phương thức createState. Phương thức này trả về một đối tượng State và mục đích chính của class này là tạo một đối tượng State. Dữ liệu trong class này không thể thay đổi. Class này bị loại bỏ và được thay thế khi dữ liệu cần thay đổi và một Widget mới được tạo.



Hình 2.12. Cách hoạt động của class extends State [2]

- Class mở rộng từ State là lớp thực hiện hầu hết công việc. Nó chứa dữ liệu có thể thay đổi. Nó xây dựng giao diện người dùng bằng phương pháp 'build'. Nó có thể phản hồi các sự kiện, như người dùng nhấp vào nút. [2]

### - VD3: Hiển thị ứng dụng sử dụng StatefulWidget

```
class DiceRoll extends StatefulWidget {  
    const DiceRoll({super.key});  
  
    @override  
    State<DiceRoll> createState() {  
        return _DiceRollState();  
    }  
}  
  
class _DiceRollState extends State<DiceRoll> {  
    var activeDiceImage =  
    "assets/images/dice1.png";  
  
    void rollDice() {  
        setState(() {  
            activeDiceImage =  
            "assets/images/dice2.png";  
        });  
    }  
  
    @override  
    Widget build(BuildContext context) {  
        return Column(  
            mainAxisSize: MainAxisSize.min,  
            children: [  
                Image.asset(  
                    activeDiceImage,  
                    width: 200,  
                ),  
                const SizedBox(height: 20,),  
                ElevatedButton(  
                    onPressed: rollDice,  
                    style: ElevatedButton.styleFrom(  
                        // style button  
                    ),  
                    child: const Text("Roll Dice")  
                )  
            ],  
        );  
    }  
}
```



Hình 2.13. VD Statefull Widget

- Ứng dụng trên ban đầu sẽ được thiết lập xúc xắc là 1 nhưng sau khi ấn nút “Roll Dice” thì màn hình sẽ thay bằng ảnh xúc xắc bằng 2.

- Lệnh setState() trong StatefulWidget sẽ kiểm tra lại toàn bộ các widget có trong StatefulWidget sau đó thay những widget bị thay đổi. Nếu màn hình nhiều widget mà trong đó có chứa các widget không thay đổi thì phải tách ra thành các StatelessWidget để giảm thiểu sự kiểm tra khi có lệnh setState().

#### **2.4.4. Các Widget cơ bản thường sử dụng**

Dưới đây sẽ liệt kê các Widget thường sử dụng trong Flutter [1]:

Tên Widget	Chức năng
<b>Scaffold</b>	Triển khai bộ cục trực quan Material Design cơ bản, cho phép bạn dễ dàng thêm nhiều tiện ích khác nhau như AppBar, BottomAppBar, FloatingActionButton, Drawer, SnackBar, BottomSheet, v.v.
<b>AppBar</b>	Thường chứa các thuộc tính tiêu đề, thanh công cụ, hàng đầu và hành động đơn giản (cùng với các nút), cũng như nhiều tùy chọn tùy chỉnh và hiển thị trên cùng của ứng dụng.
<b>SafeArea</b>	Tự động thêm khoảng đệm vừa đủ vào widget con để tránh các thành phần của thiết bị như camera.
<b>Container</b>	Là một tiện ích được sử dụng phổ biến cho phép tùy chỉnh tiện ích con của nó. Có thể dễ dàng thêm các thuộc tính như màu sắc, chiều rộng, chiều cao, phần đệm, lề, đường viền, ràng buộc, căn chỉnh, biến đổi (chẳng hạn như xoay hoặc định cỡ tiện ích), v.v.
<b>Text</b>	Sử dụng để hiển thị một chuỗi ký tự.
<b>RichText</b>	Hiển thị văn bản bằng nhiều kiểu.
<b>Column</b>	Hiển thị các phần tử con của nó theo chiều dọc.
<b>Row</b>	Hiển thị các con của nó theo chiều ngang.
<b>ListView</b>	Hiển thị các phần tử con của nó theo chiều dọc hoặc ngang và có thể cuộn được khi dữ liệu tràn ra khỏi màn hình.
<b>GridView</b>	Hiển thị các phần tử theo dạng lưới.
<b>ScaffoldMessenger</b>	Thẻ thông báo được hiển thị dưới cùng của màn hình.
<b>Overlay</b>	Tạo một lớp phủ trên màn hình
<b>TextField</b>	Tạo ra một trường để nhập văn bản
<b>TextFormField</b>	Tạo ra một trường để nhập văn bản và có kiểm chứng
<b>Button (RaisedButton, FloatingActionButton, FlatButton, IconButton, v.v.)</b>	Có nhiều loại nút để lựa chọn cho các tình huống khác nhau. Có thể tham khảo và xem ví dụ tại <a href="https://docs.flutter.dev/ui/widgets/material2">https://docs.flutter.dev/ui/widgets/material2</a>

<b>Image</b>	Hiển thị hình ảnh từ nguồn cục bộ hoặc nguồn URL (web).
<b>Icon</b>	Sử dụng để vẽ các biểu tượng.
<b>Navigator</b>	Quản lý sự chuyển đổi giữa các trang theo cơ chế ngăn xếp.
<b>Animation</b>	Tạo các chuyển động đẹp mắt cho ứng dụng.

- Trong Flutter ngoài các widget được sử dụng chung cho iOS và Android thì có các widget xử lý riêng cho các phần tử của iOS nằm trong thư viện cupertino: CupertinoButton, CupertinoDatePicker, CupertinoDialog, CupertinoTabBar, CupertinoTabScaffold, CupertinoTabView, CupertinoTextField, v.v.

Bên trên là các widget hay được dùng trong các app viết bởi Flutter. Ngoài ra khi muốn xử lý một số hành động như: mở camera, xử lý hình ảnh,... thì phải dùng thêm các thư viện bên ngoài khác. Dưới đây là một số thư viện:

Tên thư viện	Nguồn	Chức năng
<b>Google Fonts</b>	<a href="https://pub.dev/packages/google_fonts">https://pub.dev/packages/google_fonts</a>	Sử dụng các font chữ của google.
<b>UUID</b>	<a href="https://pub.dev/packages/uuid">https://pub.dev/packages/uuid</a>	Tự động tạo ID.
<b>intl</b>	<a href="https://pub.dev/packages/intl">https://pub.dev/packages/intl</a>	Có các hàm xử lý ngày tháng.
<b>riverpod</b>	<a href="https://pub.dev/packages/flutter_riverpod">https://pub.dev/packages/flutter_riverpod</a>	Quản lý trạng thái tiện ích chéo.
<b>http</b>	<a href="https://pub.dev/packages/http">https://pub.dev/packages/http</a>	Để tương tác với RESTful API.
<b>image picker</b>	<a href="https://pub.dev/packages/image_picker">https://pub.dev/packages/image_picker</a>	Sử dụng ảnh chụp trực tiếp từ camera hoặc lấy ảnh từ thư viện ảnh của thiết bị.

## CHƯƠNG 3:

### ÁP DỤNG FLUTTER ĐỂ XÂY DỰNG ỨNG DỤNG CHAT

#### 3.1. Tài liệu đặc tả chung

##### 3.1.1. Danh sách các từ chuyên môn trong lĩnh vực của ứng dụng

STT	Tên tiếng Việt	Tên tiếng Anh	Giải nghĩa
<b>Danh từ</b>			
1	Người dùng	User	Người dùng ứng dụng chat gồm: nhân viên của doanh nghiệp, người quản lý
2	Nhân viên	Employee	Là nhân viên bình thường của doanh nghiệp.
3	Người quản lý nhóm	Group manager	Là leader của 1 team hoặc trưởng phòng, trưởng bộ phận tùy theo sự phân cấp.
4	Người quản lý người dùng	User administrator	Là sếp cao nhất hoặc một người được thuê về quản lý tất cả các tài khoản.
5	Mật khẩu	Password	Chuỗi ký tự có tối thiểu 8 ký tự và tối đa 20 ký tự
6	Cuộc trò chuyện	Conversation	Là màn hình các tin nhắn mà các nhân viên gửi cho nhau.
7	Tài khoản người dùng	User account	Là tài khoản do doanh nghiệp cấp cho nhân viên.
8	Thanh tìm kiếm	Search bar	Là thanh để gõ tên tên người dùng cần tìm vào để tìm kiếm nhanh.
9	Thông báo	Notification	Là thông báo của doanh nghiệp gửi đến cho nhân viên.
<b>Động từ</b>			
10	Đăng nhập	Sign in/Login	Người dùng đăng nhập vào tài khoản được doanh nghiệp cấp.
11	Đăng xuất	Sign out/Log out	Hệ thống có thể tự đăng xuất sau một khoảng thời gian để đảm bảo bảo mật
12	Đổi mật khẩu	Change password	Là hành động khi người dùng tự đổi mật khẩu.
13	Quên mật khẩu	Forgot password	Khi người dùng ấn quên mật khẩu, hệ thống gửi form chứa link đặt lại mật khẩu về email.
14	Hiển thị các cuộc trò chuyện	View conversations	Các cuộc trò chuyện sẽ được hiển thị theo dạng danh sách
15	Hiển thị các tài khoản	View accounts	Tài khoản của người dùng sẽ được hiển thị theo dạng bảng

			(chỉ có ở tài khoản admin).
16	Thêm bạn	Add friend	Nhập tên người cần tìm vào thanh search, hệ thống hiển thị kết quả gợi ý. Người dùng click vào 1 trong các kết quả hiển thị sẽ dẫn đến cuộc trò chuyện giữa 2 người.
17	Nhận tin nhắn	Get messages	Hệ thống tự động cập nhật khi có tin nhắn mới đến.
18	Tìm kiếm	Search	Hệ thống tìm trong CSDL các thông tin cần tìm.
19	Quản lý nhóm	Group manage	Quản lý 1 nhóm bao gồm các công việc: thêm, xóa thành viên, chuyển quyền leader (có ở tài khoản admin và leader).
20	Quản lý người dùng	User manage	Quản lý người dùng bao gồm các công việc: thêm, xóa, cập nhật, xem tài khoản người dùng (chỉ có ở tài khoản admin).
21	Xác thực tài khoản	Account authentication	Hệ thống kiểm tra xem trong CSDL có tài khoản đó không.
22	Xóa tài khoản	Delete Account	Hệ thống chuyển thuộc tính state của User sang false

### 3.1.2. Mục đích

Phần mềm hỗ trợ việc trao đổi thông tin.

### 3.1.3. Mô tả hệ thống chi tiết bằng ngôn ngữ tự nhiên

#### a) Phạm vi phần mềm

- Hỗ trợ việc trao đổi thông tin trong nội bộ của một doanh nghiệp.
- Các đối tượng có thể sử dụng phần mềm: nhân viên của doanh nghiệp (nhân viên), quản lý của 1 nhóm làm việc (người quản lý nhóm), người quản lý hệ thống (người quản lý người dùng).

#### b) Mô tả

##### \* Đối với tất cả người dùng của hệ thống

- Phải đăng nhập để có thể thực hiện các công việc của mình.
- Nếu quên mật khẩu không đăng nhập được sẽ có chức năng quên mật khẩu
- Sau khi đăng nhập có thể thay đổi mật khẩu cá nhân

- Có thể đăng xuất khỏi hệ thống hoặc hệ thống tự đăng xuất sau một khoảng thời gian.

**\* Đối với nhân viên:**

- Có thể gửi tin nhắn riêng hoặc gửi vào nhóm
- Hệ thống tự động nhận tin nhắn được gửi tới khi có mạng
- Có thể gọi điện
- Có thể thêm bạn sau khi search bằng tên
- Có thể xem danh sách bạn bè/nhóm

**\* Đối với người quản lý nhóm:**

- Có thể làm mọi thứ mà nhân viên có thể làm
- Quản lý nhóm mà mình làm trưởng nhóm

**\* Đối với người quản lý người dùng:**

- Có thể quản lý tất cả các người dùng
- Gửi thông báo đến bất kỳ ai.
- Tạo nhóm cho từng bộ phận trong doanh nghiệp
- Có thể quản lý các nhóm

*c) Nghiệp vụ*

**\* Người dùng:**

- Đối với người dùng nói chung:

- + Phải đăng nhập để thực hiện các hoạt động nghiệp vụ của mình
- + Nếu quên mật khẩu không đăng nhập được sẽ có chức năng quên mật khẩu

- Vậy có thể có các use case:

- + Đăng nhập
- + Quên mật khẩu

- Mô tả các use case:

- + Đăng nhập: use case này cho phép người dùng đăng nhập vào tài khoản của mình.
- + Quên mật khẩu: use case này cho phép hệ thống gửi vào email người dùng form có chứa link đặt lại mật khẩu.

**\* Nhân viên:**

- Đối với nhân viên của doanh nghiệp:

- + Phải đăng nhập để sử dụng hệ thống
- + Sau khi đăng nhập, nhân viên có thể xem danh sách bạn bè/nhóm, tìm kiếm bạn bè, thêm bạn, nhắn tin, gọi điện.

- Vậy có thể có các use case:

- + Đăng nhập được kế thừa từ người dùng
- + Xem danh sách bạn bè/nhóm
- + Tìm kiếm cuộc trò chuyện

+ Thêm bạn

+ Nhắn tin

+ Gọi điện

- Mô tả các use case:

+ Xem danh sách bạn bè/nhóm: use case này cho phép hệ thống hiển thị các cuộc trò chuyện dưới dạng danh sách.

+ Tìm kiếm cuộc trò chuyện: use case này cho phép nhân viên có thể tìm kiếm cuộc trò chuyện theo tên.

+ Thêm bạn: use case này cho phép nhân viên có thể thêm bạn (tạo ra cuộc trò chuyện mới).

+ Nhắn tin: use case này cho phép nhân viên có thể gửi tin nhắn vào trong cuộc trò chuyện nào đó.

+ Gọi điện: use case này cho phép nhân viên có thể gọi điện cho đồng nghiệp.

\* **Người quản lý nhóm:**

- Đối với người quản lý nhóm:

+ Có thể làm mọi thứ nhân viên làm

+ Quản lý nhóm: thêm thành viên, xóa thành viên, xem danh sách các thành viên, chuyển quyền trưởng nhóm, bắt đầu cuộc gọi nhóm.

- Vậy có thể có các use case:

+ Quản lý nhóm

- Mô tả use case:

+ Quản lý nhóm: use case này cho phép người quản lý có thể thêm, xóa thành viên, xem danh sách thành viên của nhóm, chuyển quyền trưởng nhóm, bắt đầu cuộc gọi nhóm.

\* **Người quản lý người dùng:**

- Đối với người quản lý người dùng:

+ Phải đăng nhập để sử dụng được tài khoản quản lý

+ Sau khi đăng nhập, người quản lý người dùng có thể thực hiện nghiệp vụ của mình: quản lý người dùng, quản lý các nhóm chat, gửi thông báo, tạo nhóm, tìm kiếm tài khoản.

- Vậy có thể có các use case:

+ Đăng nhập được kê thừa từ người dùng

+ Quản lý người dùng

+ Gửi thông báo

+ Tạo nhóm

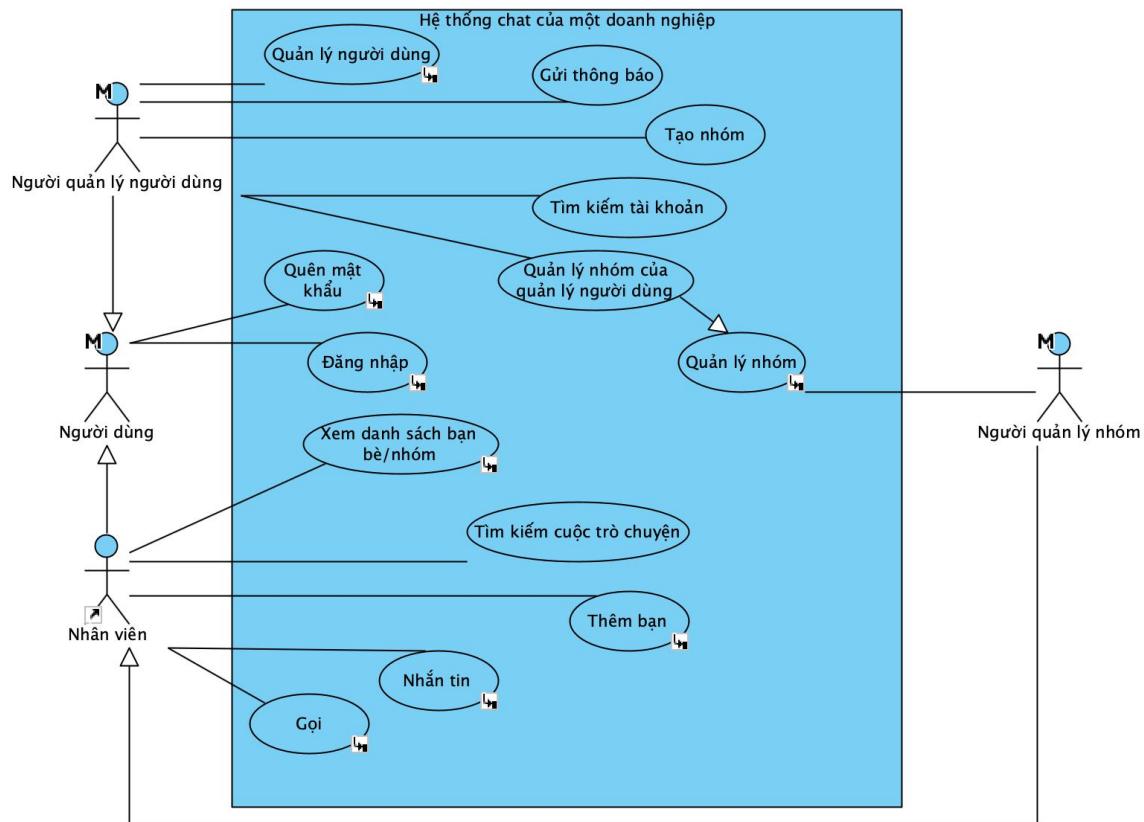
+ Tìm kiếm tài khoản

+ Quản lý nhóm của quản lý người dùng

- Mô tả các use case:

- + Quản lý người dùng: use case này cho phép người quản lý người dùng thêm, xóa, sửa, xem tài khoản của nhân viên.
- + Gửi thông báo: use case này cho phép người quản lý người dùng có thể gửi thông báo tới bất kỳ tài khoản nào được chọn.
- + Tạo nhóm: use case này cho phép người quản lý người dùng có thể tạo nhóm cho tất cả các tài khoản được chọn.
- + Tìm kiếm tài khoản: use case này cho phép người quản lý người dùng có thể tìm kiếm tài khoản người dùng bằng địa chỉ email.
- + Quản lý nhóm của quản lý người dùng: use case này cho phép người quản lý người dùng có quyền quản lý nhóm như người quản lý nhóm.

### 3.1.4. Sơ đồ tổng quan các usecase của toàn hệ thống



Hình 3.1. Use case tổng quan

### 3.1.5. Mô tả hệ thống chi tiết cho các module

#### a) Mô tả hệ thống chi tiết cho module “Quên mật khẩu”

##### \* Đối với người dùng:

- Người dùng quên mật khẩu không đăng nhập được vào hệ thống → Ấn nút quên mật khẩu trên màn hình → Hệ thống hiện ra một biểu mẫu để nhập email → Người dùng nhập email, rồi ấn gửi → Hệ thống xác nhận thành công và gửi cho người dùng email có chứa link đổi mật khẩu → Người dùng quay bấm quay trở lại trang đăng nhập và đăng nhập bằng mật khẩu mới.

##### \* Vậy có các use case:

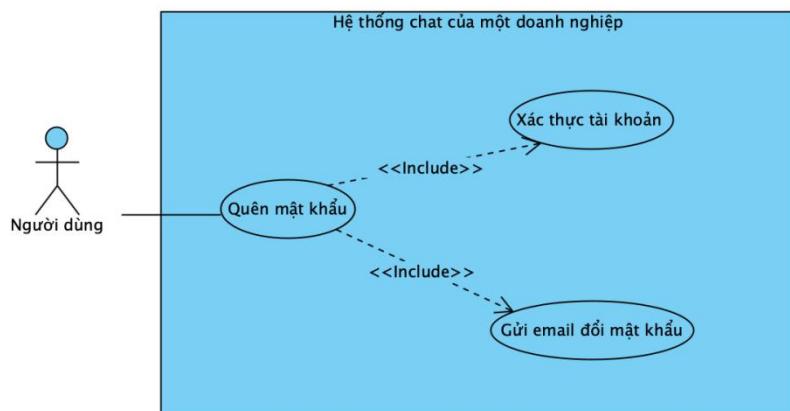
- Quên mật khẩu
- Xác thực tài khoản
- Gửi email chứa link đặt lại mật khẩu

Sau khi người dùng ấn “Send” trong form thì hệ thống thực hiện hàng loạt các hành động xác thực tài khoản và gửi email chứa link đổi mật khẩu do đó chúng đều được include từ hệ thống hiển thị form quên mật khẩu.

#### \* Mô tả các use case:

- Quên mật khẩu: use case này cho phép người dùng thực hiện tính năng quên mật khẩu.
- Xác thực tài khoản: use case này cho phép hệ thống kiểm tra xem địa chỉ email đó có trong CSDL không.
- Gửi email đổi mật khẩu: use case này cho phép hệ thống tự động gửi 1 email có chứa link đổi mật khẩu vào mail của người dùng.

#### \* Sơ đồ use case của module “Quên mật khẩu”:



Hình 3.2. Use case module "Đổi mật khẩu"

#### b) Mô tả hệ thống chi tiết cho module “Đăng nhập”

#### \* Đối với người dùng:

- Người dùng mở ứng dụng → Hiển thị form đăng nhập gồm có: email và mật khẩu → Người dùng nhập email và mật khẩu → Hệ thống kiểm tra email có trong database không và xác nhận mật khẩu có đúng với tài khoản đó không → Người dùng vào được tài khoản sẽ có thể đổi mật khẩu và đăng xuất.

#### \* Vậy có các use case:

- Đăng nhập
- Đổi mật khẩu
- Đăng xuất

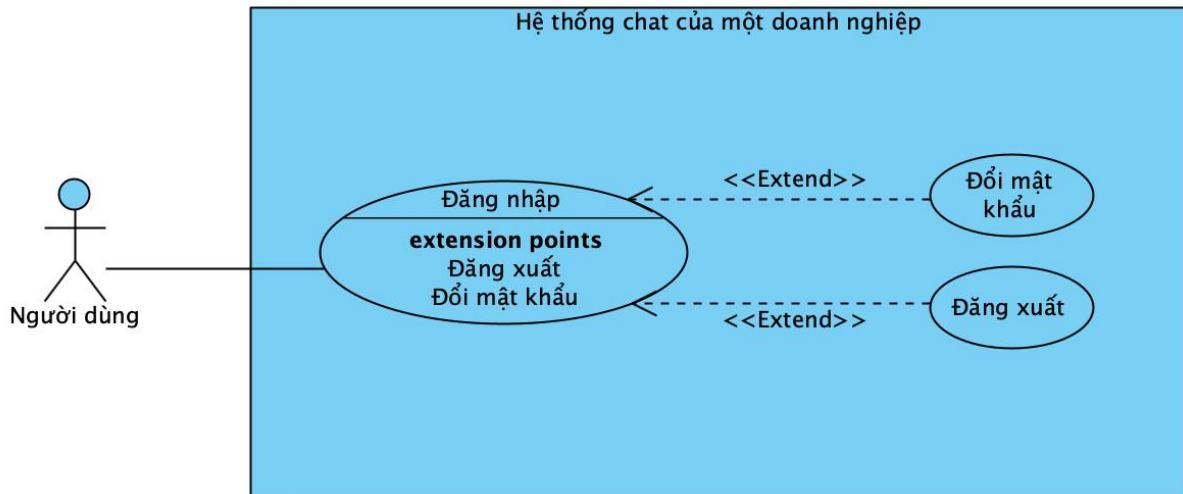
Trong đó đổi mật khẩu và đăng xuất là 2 hành động tùy chọn sau khi đăng nhập. Do đó, đổi mật khẩu và đăng xuất extend từ đăng nhập.

#### \* Mô tả các use case:

- Đăng nhập: use case này cho phép người dùng đăng nhập vào tài khoản.

- Đổi mật khẩu: use case này cho phép người dùng đổi mật khẩu của tài khoản.
- Đăng xuất: use case này cho phép người dùng đăng xuất khỏi hệ thống.

\* Sơ đồ use case của module “Đăng nhập”:



Hình 3.3. Use case module "Đăng nhập"

c) Mô tả hệ thống chi tiết cho module “Xem danh sách bạn bè/nhóm”

\* Đối với nhân viên của doanh nghiệp:

- Nhân viên đăng nhập vào hệ thống thành công → Hệ thống hiển thị danh sách bạn bè/nhóm được sắp xếp theo thứ tự tin nhắn mới đến thì xếp cuộc trò chuyện đó ở đầu → Nhân viên chọn 1 bạn bè/nhóm → Hệ thống hiển thị chi tiết 1 cuộc trò chuyện.

\* Vậy có các use case:

- Xem danh sách bạn bè/nhóm
- Sắp xếp
- Xem chi tiết 1 cuộc trò chuyện

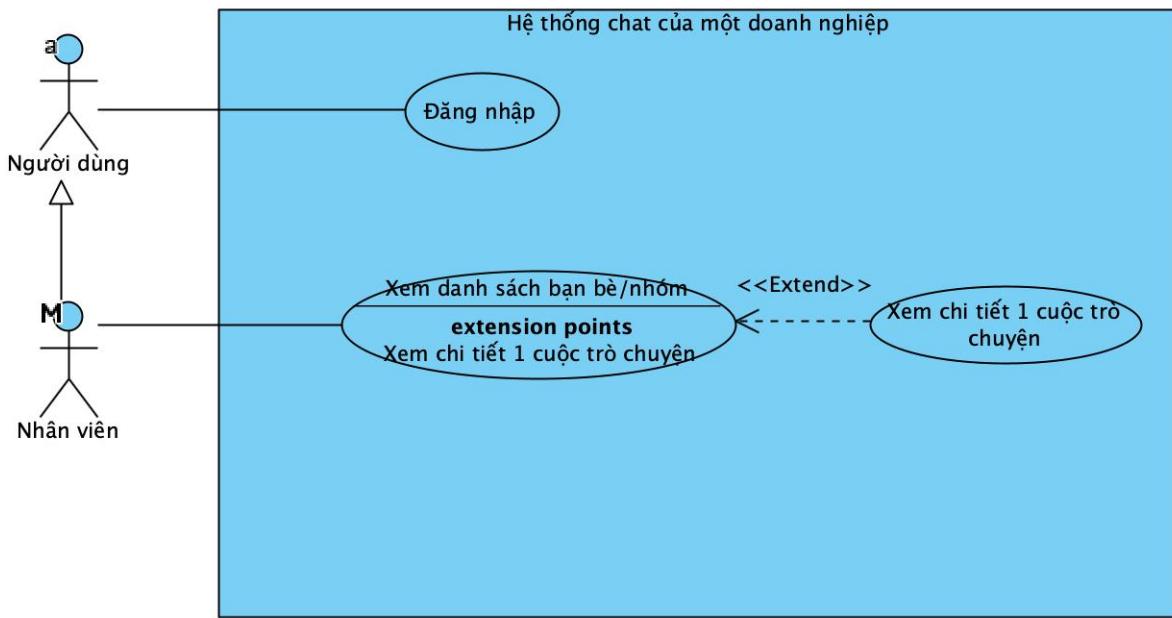
Trong đó xem chi tiết 1 cuộc trò chuyện là hành động tùy chọn và có thể lặp lại nhiều lần nên nó extend từ xem danh sách bạn bè/nhóm.

\* Mô tả các use case:

- Xem danh sách bạn bè/nhóm: use case này cho phép hệ thống hiển thị danh sách bạn bè, nhóm dưới dạng list view.
- Sắp xếp: use case này cho phép hệ thống sắp xếp danh sách bạn bè/nhóm theo tiêu chí tin nhắn đến trong khoảng thời gian gần nhất.

- Xem chi tiết 1 cuộc trò chuyện: use case này cho phép nhân viên xem chi tiết 1 cuộc trò chuyện khi người dùng này chọn cuộc trò chuyện đó.

\* Sơ đồ use case của module “Xem danh sách bạn bè/nhóm”



Hình 3.4. Use case module "Xem danh sách bạn bè/nhóm"

d) Mô tả hệ thống chi tiết cho module “Thêm bạn”

\* **Đối với nhân viên của doanh nghiệp:**

- Nhân viên đăng nhập vào hệ thống thành công → Hệ thống hiển thị danh sách bạn bè/nhóm và dưới AppBar có thanh search → Nhân viên nhập tên của đồng nghiệp → Hệ thống tìm kiếm trong CSDL xem có tên đó không nếu có thì ứng dụng sẽ hiển thị tài khoản đó ngay dưới thanh search → Nhân viên click vào tài khoản đó → Hệ thống hiển thị cuộc trò chuyện mới mà nhân viên có thể bắt đầu nhắn tin.

\* **Vậy có các use case:**

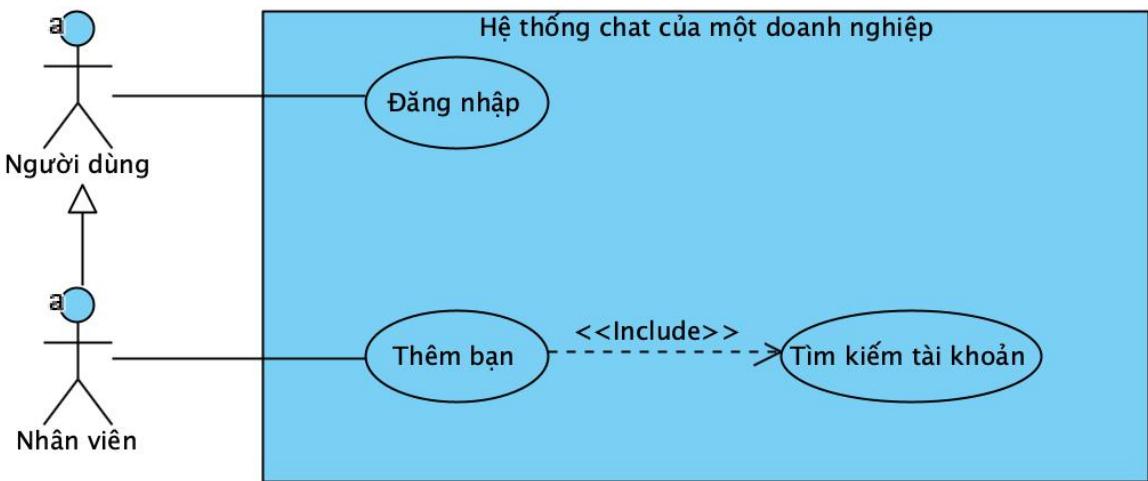
- Thêm bạn
- Tìm kiếm tài khoản

Trong có tìm kiếm tài khoản là hành động bắt buộc phải làm để thêm bạn nên nó include từ thêm bạn.

\* **Mô tả các use case:**

- Thêm bạn: use case này cho phép nhân viên thêm cuộc trò chuyện với một nhân viên chưa có trong danh sách bạn bè.
- Tìm kiếm tài khoản: use case này cho phép nhân viên tìm kiếm tài khoản của nhân viên khác qua tên của họ.

\* **Sơ đồ use case của module “Thêm bạn”**



Hình 3.5. Use case module "Thêm bạn"

e) Mô tả hệ thống chi tiết cho module “Nhắn tin”

\* **Đối với nhân viên của doanh nghiệp:**

- Nhân viên đăng nhập vào hệ thống thành công → Hệ thống hiển thị danh sách bạn bè/nhóm → Nhân viên chọn 1 cuộc trò chuyện → Hệ thống hiển thị chi tiết 1 cuộc trò chuyện → Nhân viên có thể thực hiện 1 trong các hành động sau để gửi tin nhắn: click phần soạn tin nhắn và soạn tin cần gửi. Sau đó nhân viên sẽ click gửi → Hệ thống hiển thị tin nhắn đã gửi vào cuối đoạn chat. Hệ thống sẽ tự động nhận tin nhắn ở bên người nhận khi người gửi nhắn tin.

\* **Vậy có các use case:**

- Xem danh sách bạn bè/nhóm
- Xem chi tiết 1 cuộc trò chuyện
- Nhắn tin
- Nhận tin nhắn
- Gửi tin nhắn văn bản
- Gửi file
- Gửi ảnh
- Gửi voice
- Xóa tin nhắn

Trong đó nhận tin nhắn là do hệ thống tự thực hiện khi phía bên người gửi gửi nên xem chi tiết 1 cuộc trò chuyện include nhận tin nhắn.

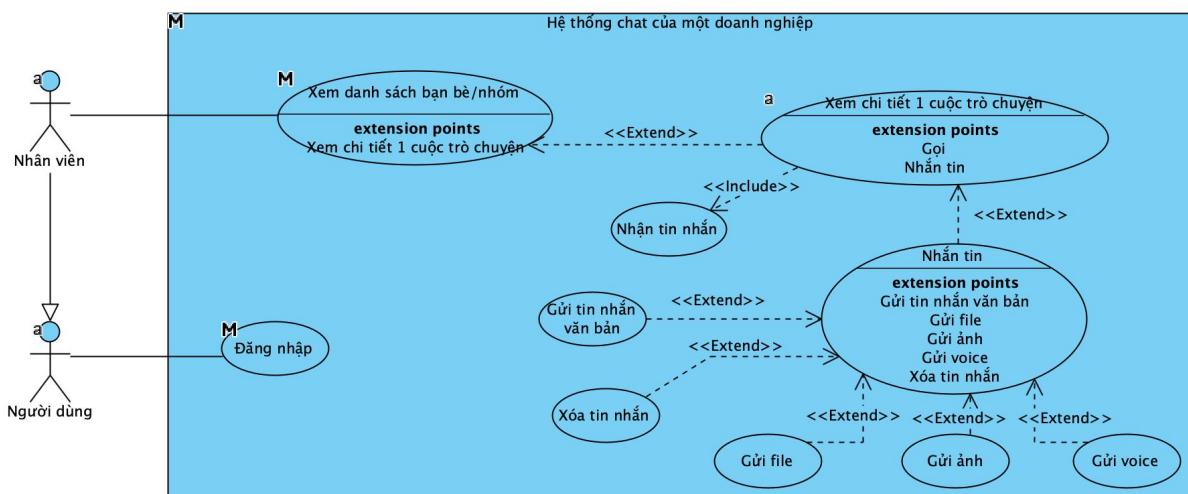
Các hành động gửi tin nhắn văn bản, gửi file, gửi ảnh, gửi voice, xóa tin nhắn là hành động tùy chọn của nhắn tin và có thể thực hiện nhiều lần nên chúng extend từ nhắn tin.

\* **Mô tả các use case:**

- Xem danh sách bạn bè/nhóm: use case này cho phép hệ thống hiển thị danh sách bạn bè, nhóm dưới dạng list view.

- Xem chi tiết 1 cuộc trò chuyện: use case này cho phép nhân viên xem chi tiết 1 cuộc trò chuyện khi người dùng này chọn cuộc trò chuyện đó.
- Nhắn tin: use case này cho phép nhân viên nhắn tin với nhân viên khác hoặc trong nhóm nào đó mà nhân viên là thành viên.
- Nhận tin nhắn: use case này cho phép hệ thống tự động nhận tin nhắn đến khi người gửi gửi tin nhắn đến.
- Gửi tin nhắn văn bản: use case này cho phép nhân viên gửi tin nhắn văn bản.
- Gửi file: use case này cho phép nhân viên gửi file từ thiết bị đang sử dụng.
- Gửi ảnh: use case này cho phép nhân viên gửi ảnh từ thư viện ảnh.
- Gửi voice: use case này cho phép nhân viên gửi được đoạn voice thu âm trực tiếp.
- Xóa tin nhắn: use case này cho phép nhân viên xóa tin nhắn mà nhân viên chọn.

#### \* Sơ đồ use case của module “Nhắn tin”



Hình 3.6. Use case module "Nhắn tin"

f) Mô tả hệ thống chi tiết cho module “Quản lý nhóm”

#### \* Đối với người quản lý nhóm

- Người quản lý nhóm đăng nhập vào hệ thống thành công → Hệ thống hiển thị danh sách bạn bè/nhóm → Người quản lý nhóm chọn xem chi tiết cuộc trò chuyện của nhóm mà mình làm trưởng nhóm → Hệ thống hiển thị cuộc trò chuyện của nhóm được chọn → Người quản lý nhóm có thể thực hiện các hành động sau: xóa thành viên, thêm thành viên, xem danh sách các thành viên, chuyển quyền trưởng nhóm → Hệ thống thực hiện hành động đã chọn.

#### \* Vậy có các use case:

- Quản lý nhóm
- Xóa thành viên
- Thêm thành viên
- Xem danh sách các thành viên

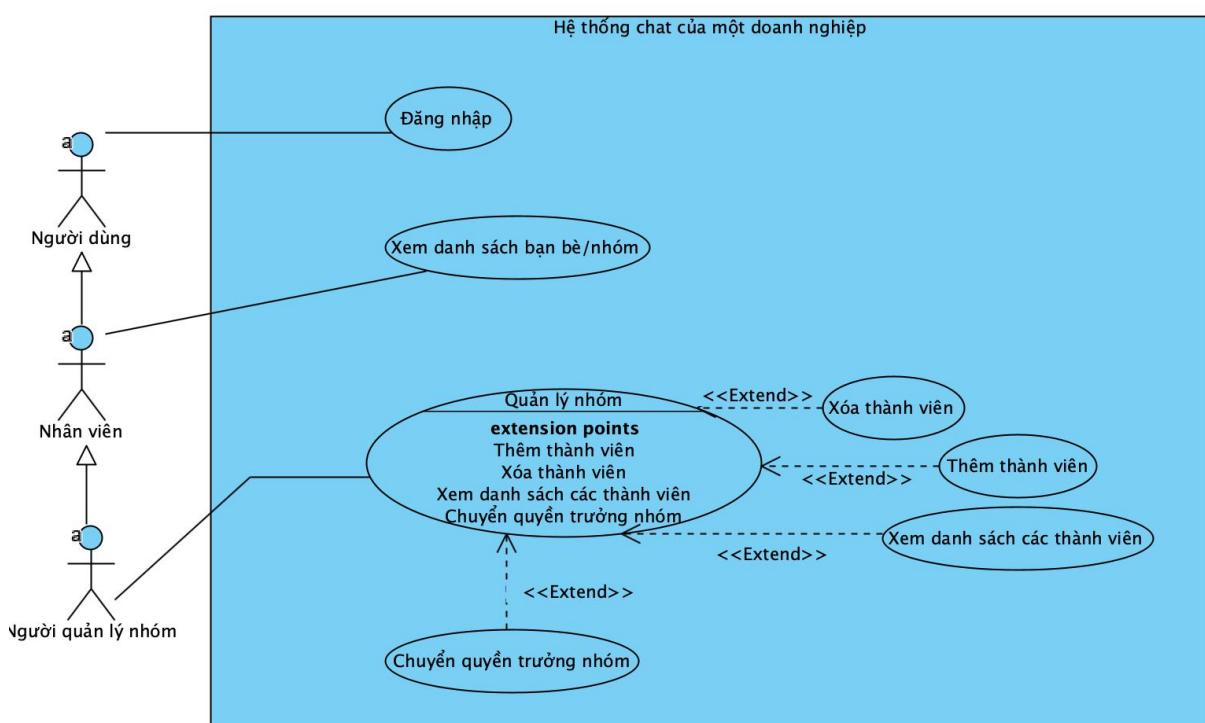
### - Chuyển quyền trưởng nhóm

Trong đó xóa thành viên, thêm thành viên, xem danh sách các thành viên, chuyển quyền trưởng nhóm là hành động tùy chọn của quản lý nhóm và có thể thực hiện nhiều lần. Do đó chúng extend từ quản lý nhóm.

#### \* Mô tả các use case:

- Quản lý nhóm: use case này cho phép người quản lý nhóm thực hiện các hành động quản lý với nhóm.
- Xóa thành viên: use case này cho phép người quản lý xóa thành viên ra khỏi nhóm.
- Thêm thành viên: use case này cho phép người quản lý thêm thành viên vào nhóm.
- Xem danh sách các thành viên: use case này cho phép người quản lý xem danh sách các thành viên của nhóm.
- Chuyển quyền trưởng nhóm: use case này cho phép người quản lý chuyển quyền trưởng nhóm cho thành viên khác trong nhóm.

#### \* Sơ đồ use case của module “Quản lý nhóm”:



Hình 3.8. Use case module "Quản lý nhóm"

#### g) Mô tả hệ thống chi tiết cho module “Quản lý người dùng”

#### \* Đối với người quản lý người dùng

- Người quản lý người dùng đăng nhập vào hệ thống thành công → Hệ thống hiển thị giao diện quản lý người dùng có thanh tìm kiếm tài khoản người dùng dưới AppBar, bảng thông tin của người dùng và nút thêm tài khoản → Người quản lý người dùng có thể thực hiện 1 trong các hành động sau: thêm tài khoản,

cập nhật tài khoản, xóa tài khoản, xem tài khoản, tìm kiếm tài khoản → Hệ thống thực hiện hành động người dùng chọn.

#### \* Vậy có các use case:

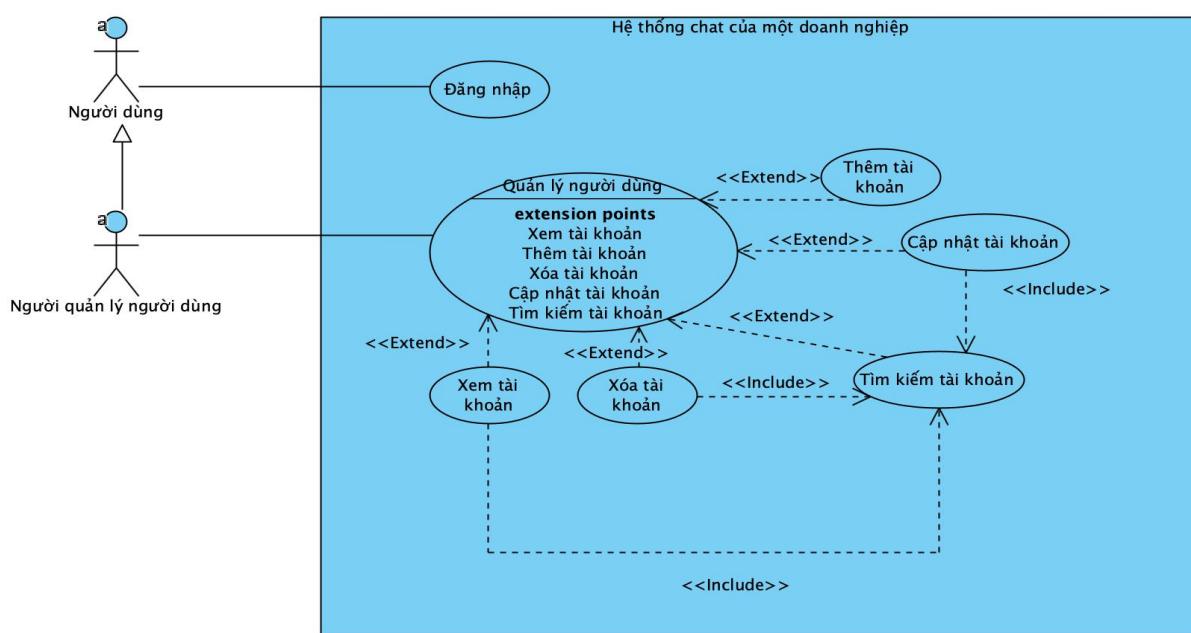
- Quản lý người dùng
- Thêm tài khoản
- Cập nhật tài khoản
- Tìm kiếm tài khoản
- Xóa tài khoản
- Xem tài khoản

Trong đó các hành động trên đều có thể lặp đi lặp lại nhiều lần nên chúng extend từ quản lý người dùng. Tuy nhiên các hành động cập nhật, tìm kiếm, xóa, xem đều phải tìm kiếm tài khoản nên chúng có mối quan hệ include.

#### \* Mô tả các use case:

- Quản lý người dùng: use case này cho phép người quản lý quản lý tài khoản của người dùng.
- Thêm tài khoản: use case này cho phép người quản lý người dùng thêm tài khoản người dùng mới.
- Cập nhật tài khoản: use case này cho phép người quản lý người dùng cập nhật thông tin cho tài khoản người dùng.
- Tìm kiếm tài khoản: use case này cho phép người quản lý người dùng tìm kiếm tài khoản người dùng qua địa chỉ email.
- Xóa tài khoản: use case này cho phép người quản lý người dùng xóa tài khoản người dùng.
- Xem tài khoản: use case này cho phép người quản lý người dùng xem thông tin chi tiết tài khoản người dùng.

#### \* Sơ đồ use case của module “Quản lý người dùng”



Hình 3.9. Use case module "Quản lý người dùng"

### 3.2. Tài liệu pha phân tích

#### 3.2.1. Viết scenario

a) Viết scenario cho module “Quên mật khẩu”

Usecase	Quên mật khẩu
Actor	Người dùng
Tiền điều kiện	Mở app
Hậu điều kiện	Nhận được email mật khẩu mới
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi vào app, giao diện hiện lên form đăng nhập gồm trường nhập email, trường nhập mật khẩu, nút quên mật khẩu.</li> <li>2. Người dùng click vào nút quên mật khẩu.</li> <li>3. Hệ thống hiện ra form quên mật khẩu gồm trường nhập email và nút gửi.</li> <li>4. Người dùng nhập vào email <a href="mailto:MaiCT1@cp.vn">MaiCT1@cp.vn</a> và ấn nút gửi.</li> <li>5. Hệ thống kiểm tra email và thấy đã có trong CSDL, hệ thống gửi form có link reset mật khẩu vào email mà người dùng đăng ký với doanh nghiệp.</li> <li>6. Hệ thống hiện lên trên màn hình dưới trường email dòng chữ “A message has been sent.”.</li> <li>7. Người dùng mở email và thấy mật khẩu mới đã được gửi về.</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>5. Hệ thống kiểm tra email không có trong CSDL</li> <li>5.1. Hệ thống hiện lên trên màn hình dưới trường email dòng chữ “You have not registered an account yet”</li> <li>5.2. Quay lại bước 4 để nhập lại email</li> </ol>

b) Viết scenario cho module “Đăng nhập”

Usecase	Đăng nhập
Actor	Người dùng
Tiền điều kiện	Mở app
Hậu điều kiện	Đăng nhập thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi vào app, giao diện hiện lên form đăng nhập gồm trường nhập email, trường nhập mật khẩu, nút quên mật khẩu.</li> <li>2. Người dùng nhập vào trường email địa chỉ email <a href="mailto:MaiCT1@cp.vn">MaiCT1@cp.vn</a> và nhập vào trường mật khẩu 123456. Sau đó người dùng bấm nút “Login”.</li> <li>3. Hệ thống xác nhận địa chỉ email và mật khẩu trong CSDL đều đúng.</li> <li>4. Hệ thống đăng nhập thành công.</li> </ol>

Ngoại lệ	<p>3. Hệ thống không tìm thấy địa chỉ email trong CSDL.</p> <p>3.1. Hệ thống hiển thị dưới trường email một đoạn text “Couldn't find your account” và reset lại 2 trường email và mật khẩu.</p> <p>3.2. Quay lại bước 2.</p> <p>3. Hệ thống xác nhận địa chỉ email đúng nhưng mật khẩu sai.</p> <p>3.1. Hệ thống hiển thị dưới trường mật khẩu một đoạn text “Your password is incorrect” và reset lại 2 trường email và mật khẩu.</p> <p>3.2. Quay lại bước 2.</p>
----------	---

c) *Viết scenario cho module “Đổi mật khẩu”*

Usecase	Đổi mật khẩu
Actor	Người dùng
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Đổi mật khẩu thành công
Kịch bản chính	<p>1. Sau khi đăng nhập, từ giao diện chính, người dùng click nút trên cùng bên trái.</p> <p>2. Màn hình hiển thị drawer có chức năng đổi mật khẩu.</p> <p>3. Người dùng click vào đổi mật khẩu.</p> <p>4. Hệ thống hiển thị form đổi mật khẩu gồm: trường nhập mật khẩu cũ, nhập mật khẩu mới, xác nhận mật khẩu mới và nút “Change Password”</p> <p>5. Người dùng nhập mật khẩu cũ 123456 vào trường mật khẩu cũ, mật khẩu mới 654321 vào trường mật khẩu mới và 64321 vào trường xác nhận mật khẩu mới. Sau đó người dùng click nút “Change Password”.</p> <p>6. Hệ thống kiểm tra mật khẩu cũ đúng, so sánh mật khẩu mới giống với xác nhận mật khẩu mới. Hệ thống cập nhật mật khẩu mới vào CSDL.</p> <p>7. Hệ thống hiển thị thông báo mật khẩu đã được cập nhật thành công.</p>
Ngoại lệ	<p>5. Người dùng nhập mật khẩu mới 654321 vào trường mật khẩu mới và 123456 vào trường xác nhận mật khẩu mới. Sau đó người dùng click nút “Change Password”.</p> <p>5.1. Hệ thống hiển thị text dưới trường xác nhận mật khẩu “Confirm the password is different from the new password.”</p> <p>5.2. Quay lại bước 5</p> <p>5. Người dùng nhập mật khẩu mới 876 vào trường mật khẩu mới và 12345678 vào trường xác nhận mật khẩu mới. Sau đó người dùng click nút “Change Password”.</p> <p>5.1. Hệ thống so sánh mật khẩu mới không giống với xác nhận</p>

	<p>mật khẩu mới.</p> <p>5.2. Hệ thống hiển thị text dưới trường mật khẩu mới “Your password must contain at least characters.” và ưới trường xác nhận mật khẩu “Confirm the password is different from the new password.”</p> <p>5.3. Quay lại bước 5</p> <p>6. Hệ thống kiểm tra mật khẩu cũ sai.</p> <p>6.1. Hệ thống hiển thị text dưới trường nhập mật khẩu cũ “Your old password does not correct.”</p> <p>6.2. Quay lại bước 5</p>
--	--

d) *Viết scenario cho module “Xem chi tiết một cuộc trò chuyện”*

Usecase	Xem chi tiết một cuộc trò chuyện
Actor	Nhân viên
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Xem được đoạn tin nhắn trong cuộc trò chuyện
Kịch bản chính	<p>1. Sau khi đăng nhập thành công, hệ thống hiển thị giao diện chính là danh sách các đoạn chat của bạn bè/nhóm dưới dạng ListView.</p> <p>2. Người dùng click chọn vào đoạn chat đầu tiên.</p> <p>3. Hệ thống hiển thị các tin nhắn đã gửi và đã nhận trong cuộc trò chuyện.</p>
Ngoại lệ	<p>‘3. Hệ thống hiển thị đoạn text “Let's start the conversation.” ở giữa màn hình khi chưa có tin nhắn nào trong cuộc trò chuyện.</p> <p>‘3. Hệ thống đang tải tin nhắn thì hiển thị vòng tròn chờ.</p>

e) *Viết scenario cho module “Tìm kiếm cuộc trò chuyện”*

Usecase	Tìm kiếm cuộc trò chuyện
Actor	Nhân viên
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Cuộc trò chuyện cần tìm được hiển thị dưới thanh search
Kịch bản chính	<p>1. Sau khi đăng nhập thành công, hệ thống hiển thị giao diện chính, dưới AppBar là thanh search.</p> <p>2. Nhân viên click vào thanh search và gõ tên cuộc trò chuyện muốn tìm: “mai”.</p> <p>3. Hệ thống tìm kiếm trong CSDL danh sách các tài khoản của nhân viên có chứa tên “mai”.</p> <p>4. Hệ thống hiển thị các tài khoản có tên chứa “mai” như “Chu Thi Mai” dưới thanh search theo dạng ListView.</p>

Ngoại lệ	<p>3. Hệ thống không tìm thấy kết quả nào giống hoặc gần giống với “mai” trong CSDL.</p> <p>3.1. Hệ thống hiển thị “No results were found” dưới thanh search.</p> <p>3.2. Quay lại bước 2</p>
----------	---

f) *Viết scenario cho module “Thêm bạn”*

Usecase	Thêm bạn
Actor	Nhân viên
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Cuộc trò chuyện với người bạn mới được thêm vào danh sách bạn bè.
Kịch bản chính	<p>1. Sau khi đăng nhập thành công, hệ thống hiển thị giao diện chính, dưới AppBar là thanh search.</p> <p>2. Nhân viên click vào thanh search và tên của nhân viên khác: “lam”.</p> <p>3. Hệ thống tìm kiếm tài khoản có chứa ‘lam’ trong CSDL.</p> <p>4. Hệ thống hiển thị tài khoản tìm được dưới thanh search khi tìm được tài khoản có tên: Nguyen Thi Thanh Lam (email lamntt@cp.vn).</p> <p>5. Người dùng click vào tài khoản.</p> <p>6. Hệ thống tạo mới thông tin cuộc trò chuyện vào CSDL hiện cuộc trò chuyện chi tiết với đoạn text “Let's start the conversation.” ở giữa màn hình.</p> <p>7. Người dùng bấm quay lại màn hình danh sách các cuộc trò chuyện.</p> <p>8. Hệ thống hiển thị danh sách các cuộc trò chuyện của nhân viên trong đó có cuộc trò chuyện mới.</p>
Ngoại lệ	<p>4. Hệ thống hiển thị “No results were found” dưới thanh search khi không tìm được tài khoản nào có chứa tên ‘lam’</p> <p>4.1. Quay lại bước 2.</p>

g) *Viết scenario cho module “Nhắn tin”*

Usecase	Nhắn tin
Actor	Nhân viên
Tiền điều kiện	Xem chi tiết 1 cuộc trò chuyện
Hậu điều kiện	Gửi tin nhắn thành công.
Kịch bản chính	<p>1. Sau khi hệ thống hiển thị chi tiết đoạn chat của 1 cuộc trò chuyện, nhân viên click vào TextInput dưới cùng màn hình và soạn tin nhắn “Hi. I'm Mai” rồi click gửi.</p> <p>2. Hệ thống hiển thị tin nhắn đã gửi lên trên màn hình được căn lề</p>

	phải.
Ngoại lệ	<p>1. Nhân viên click vào biểu tượng file.</p> <p>1.4. Hệ thống hiển thị màn hình chọn file trong thiết bị.</p> <p>1.5. Nhân viên click vào 1 file.</p> <p>1.6. Tiếp tục bước 3.</p> <p>1. Nhân viên click vào biểu tượng ảnh.</p> <p>1.1. Hệ thống hiển thị màn hình chọn ảnh trong thư viện ảnh của thiết bị.</p> <p>1.2. Nhân viên click vào 1 ảnh.</p> <p>1.3. Tiếp tục bước 3</p> <p>1. Nhân viên click vào biểu tượng mic.</p> <p>1.1. Nhân viên nhấn và giữ để thu âm giọng nói sau đó thả tay ra.</p> <p>1.2. Tiếp tục bước 3.</p>

#### *h) Viết scenario cho module “Thêm thành viên”*

Usecase	Thêm thành viên
Actor	Người quản lý nhóm
Tiền điều kiện	Xem chi tiết 1 cuộc trò chuyện nhóm
Hậu điều kiện	Thành viên mới thêm vào có trong danh sách nhóm
Kịch bản chính	<p>1. Sau khi hệ thống hiển thị chi tiết đoạn chat của 1 cuộc trò chuyện, người quản lý nhóm chọn nút i trên cùng bên phải.</p> <p>2. Hệ thống hiển thị các hành động: thêm thành viên, xem thành viên, rời khỏi nhóm.</p> <p>3. Người quản lý nhóm chọn thêm thành viên.</p> <p>4. Hệ thống hiển thị thanh search ở AppBar.</p> <p>5. Người quản lý nhóm click vào thanh search và gõ ‘mai’.</p> <p>6. Hệ thống tìm kiếm tài khoản có tên chứa ‘mai’ đã nhập trong CSDL.</p> <p>7. Hệ thống hiển thị tài khoản tìm được dưới thanh search khi tìm được tài khoản “Chu Thị Mai” có email maict1@cp.vn và nút “Add” bên cạnh.</p> <p>8. Người quản lý nhóm click “Add”.</p> <p>9. Hệ thống quay lại xem chi tiết cuộc trò chuyện nhóm.</p> <p>10. Người quản lý nhóm chọn nút i trên cùng bên phải.</p> <p>11. Hệ thống hiển thị các hành động: thêm thành viên, xem thành viên, rời khỏi nhóm.</p> <p>12. Người quản lý nhóm chọn xem thành viên.</p> <p>13. Hệ thống hiển thị danh sách thành viên nhóm trong đó có tài khoản của thành viên mới được thêm vào.</p>
Ngoại lệ	<p>1. Hệ thống hiển thị “No results were found” dưới thanh search khi không tìm được tài khoản nào có chứa ‘mai’ trong tên.</p> <p>1.1. Quay lại bước 5.</p>

i) *Viết scenario cho module “Xem danh sách các thành viên”*

Usecase	Xem danh sách các thành viên
Actor	Người quản lý nhóm
Tiền điều kiện	Xem chi tiết 1 cuộc trò chuyện nhóm
Hậu điều kiện	Hiển thị danh sách thành viên nhóm
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi hệ thống hiển thị chi tiết đoạn chat của 1 cuộc trò chuyện, người quản lý nhóm chọn nút i trên cùng bên phải.</li> <li>Hệ thống hiển thị các hành động: thêm thành viên, xem thành viên, rời khỏi nhóm.</li> <li>Người quản lý nhóm chọn xem thành viên.</li> <li>Hệ thống hiển thị danh sách thành viên nhóm.</li> </ol>
Ngoại lệ	Không có

j) *Viết scenario cho module “Xóa thành viên”*

Usecase	Xóa thành viên
Actor	Người quản lý nhóm
Tiền điều kiện	Xem danh sách các thành viên
Hậu điều kiện	Xóa thành viên ra khỏi danh sách nhóm.
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi hệ thống hiển thị danh sách các thành viên trong nhóm và 1 thanh search dưới AppBar, người quản lý nhóm chọn thành viên “Chu Thi Mai”</li> <li>Hệ thống hiển thị các hành động: chuyển quyền trưởng nhóm, xóa thành viên.</li> <li>Người quản lý nhóm chọn xóa thành viên.</li> <li>Hệ thống xóa “Chu Thi Mai” ra khỏi danh sách nhóm. Hệ thống quay lại xem chi tiết cuộc trò chuyện nhóm.</li> <li>Người quản lý nhóm chọn nút i trên cùng bên phải.</li> <li>Hệ thống hiển thị các hành động: thêm thành viên, xem thành viên, rời khỏi nhóm.</li> <li>Người quản lý nhóm chọn xem thành viên.</li> <li>Hệ thống hiển thị danh sách thành viên nhóm và không còn thành viên “Chu Thi Mai”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>Sau khi hệ thống hiển thị danh sách các thành viên trong nhóm và 1 thanh search dưới AppBar, người quản lý nhóm click vào thanh search và nhập “Chu Thi Mai”.</li> <li>Hệ thống tìm trong danh sách nhóm và hiển thị tài khoản thành viên dưới thanh search.</li> <li>Người quản lý nhóm chọn tài khoản thành viên “Chu Thi Mai”.</li> <li>Tiếp tục bước 2.</li> </ol>

k) *Viết scenario cho module “Thêm tài khoản”*

Usecase	Thêm tài khoản
Actor	Người quản lý người dùng
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Hệ thống hiển thị thông báo tài khoản đã được thêm.
Kịch bản chính	<p>1. Sau khi đăng nhập thành công, hệ thống hiển thị giao diện chính, có nút “Add Account” trên bảng hiển thị kết quả.</p> <p>2. Người quản lý người dùng click vào nút “Add Account”.</p> <p>3. Hệ thống hiển thị form điền thông tin tài khoản mới gồm: tên, giới tính, số điện thoại, email, email doanh nghiệp, mật khẩu, bộ phận làm việc, vị trí, vai trò và nút “Add Account”. Trong đó, giới tính, mật khẩu, bộ phận làm việc, vị trí, vai trò đã được điền giá trị có sẵn.</p> <p>4. Người quản lý người dùng nhập vào tất cả các thông tin tên, giới tính, số điện thoại, email, bộ phận làm việc, vị trí, vai trò lần lượt là: Chu Thị Mai, Female, 0987169250, <a href="mailto:chuthimai3723@gmail.com">chuthimai3723@gmail.com</a>, IT, Developer, Employee. Sau đó, người quản lý người dùng click “Add Account”.</p> <p>5. Hệ thống tự sinh trường email doanh nghiệp và trường mật khẩu. Hệ thống lưu thông tin tài khoản và thêm mới vào CSDL.</p> <p>6. Hệ thống hiển thị thông báo “Add new users successfully.”</p>
Ngoại lệ	<p>1. Người quản lý người dùng không nhập đủ tất cả các trường tên, giới tính, số điện thoại, email, bộ phận làm việc, vị trí, vai trò.</p> <p>1.1. Hệ thống hiển thị thông báo lần lượt dưới các trường là: “Please enter a valid user's name.” “Phone Number is incorrect.” “Please enter a valid email address.”</p> <p>1.2. Quay lại bước 4.</p>

l) *Viết scenario cho module “Cập nhật tài khoản”*

Usecase	Cập nhật tài khoản
Actor	Người quản lý người dùng
Tiền điều kiện	Đăng nhập vào tài khoản Admin
Hậu điều kiện	Hệ thống hiển thị thông báo tài khoản đã được cập nhật.
Kịch	1. Sau khi đăng nhập vào tài khoản admin hệ thống hiển thị bảng:

bản chính	Nam e	Gender	Email	Departme nt	Position	Role	
	Chu Thi Mai	Fema l	MaiCT1@cp .vn	IT	Deverlop er	Employ ee	Edit Delete
	2. Người quản lý người dùng tìm đúng tài khoản “ <a href="mailto:MaiCT1@cp.vn">MaiCT1@cp.vn</a> ” và click vào nút “Edit” cùng dòng. 3. Hệ thống hiển thị form sửa tài khoản gồm: ảnh đại diện, tên, số điện thoại, email, email doanh nghiệp, mật khẩu, bộ phận làm việc, vị trí, vai trò và nút “Edit”. Chỉ có các trường số điện thoại, email, mật khẩu, bộ phận làm việc, vị trí, vai trò có thể thay đổi. 4. Người quản lý người dùng thay đổi mật khẩu thành 12345678. Sau đó, người quản lý người dùng click “Edit”. 5. Hệ thống cập nhật thông tin tài khoản vào CSDL. 6. Hệ thống hiển thị thông báo “Update users successfully.”						
Ngoại lệ	4. Người quản lý người dùng thay đổi mật khẩu thành 12345. Sau đó, người quản lý người dùng click “Edit”. 4.1. Hệ thống hiển thị text dưới trường mật khẩu “Your password must contain at least 6 characters.” 4.2. Quay lại bước 4.						

m) Viết scenario cho module “Tạo nhóm”

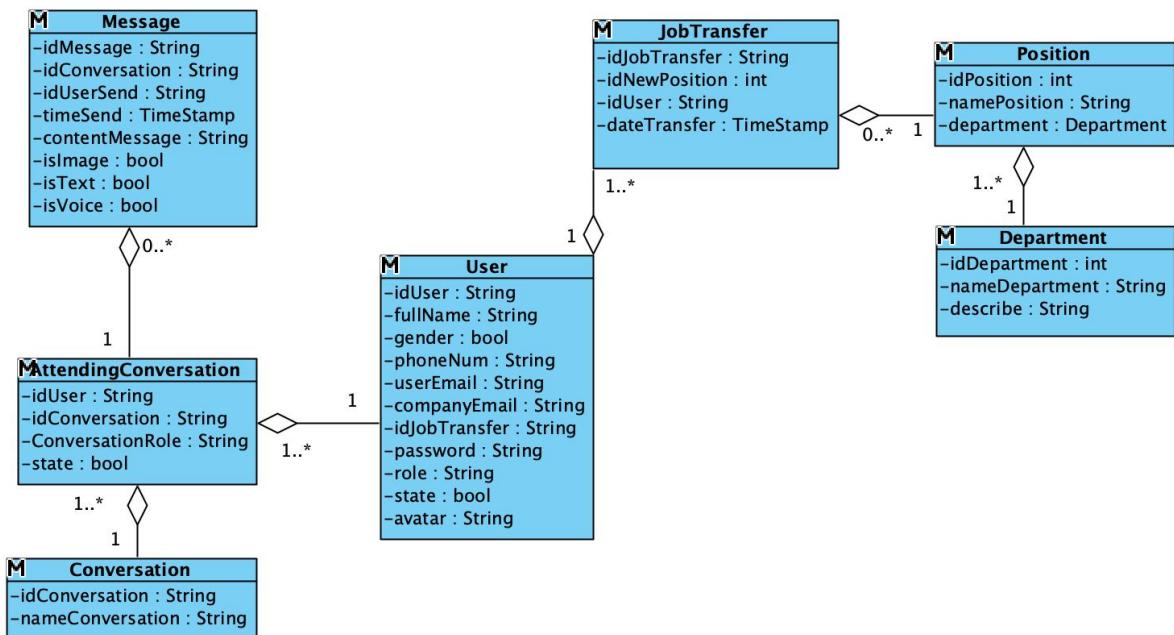
Usecase	Tạo nhóm
Actor	Người quản lý người dùng
Tiền điều kiện	Đăng nhập thành công
Hậu điều kiện	Hệ thống hiển thị nhóm đã được tạo.
Kịch bản chính	1. Sau khi đăng nhập thành công, hệ thống hiển thị giao diện chính, có nút “Add Group” trên bảng hiển thị kết quả. 2. Người quản lý người dùng tìm kiếm lần lượt các tài khoản có trong group: “ <a href="mailto:MaiCT1@cp.vn">MaiCT1@cp.vn</a> ”, “ <a href="mailto:AnNV@cp.vn">AnNV@cp.vn</a> ”, “ <a href="mailto:BaNV@cp.vn">BaNV@cp.vn</a> ” và click vào check box trên dòng của tài khoản. Sau đó, người quản lý người dùng click nút “Add Group”. 3. Hệ thống hiển thị form điền thông tin group mới gồm: tên group, chọn nhóm trưởng và nút “Add Group”. 4. Người quản lý người dùng nhập tên nhóm là “IT1”, click chọn nhóm trưởng. Sau đó click nút “Add Group” 5. Hệ thống lưu thông tin nhóm và thêm mới vào CSDL. 6. Hệ thống hiển thị thông báo “New group has been created.”
Ngoại lệ	4. Người quản lý người dùng nhập tên nhóm là “IT1”, click chọn nhóm trưởng. Sau đó click nút “Add Group” 4.1. Hệ thống tìm kiếm trong CSDL đã có nhóm tên là “IT1”. Hệ

	thông hiển thị thông báo “The group name already exists”
	4.2. Quay lại bước 4.

### 3.2.2. Mô hình hóa các lớp

#### a) Trích lớp thực thể

- User quan hệ nhiều - nhiều với Conversation nên tạo ra bảng AttendingConversation để chuyển thành 2 quan hệ 1 - nhiều.
- AttendingConversation quan hệ 1 - nhiều với Message (1 người tham gia một cuộc trò chuyện có thể gửi nhiều tin nhắn).
- Position quan hệ nhiều - 1 với Department.
- Position quan hệ nhiều - nhiều với User nên tạo ra bảng JobTransfer để chuyển thành 2 quan hệ 1 - nhiều.



Hình 3.10. Trích lớp thực thể

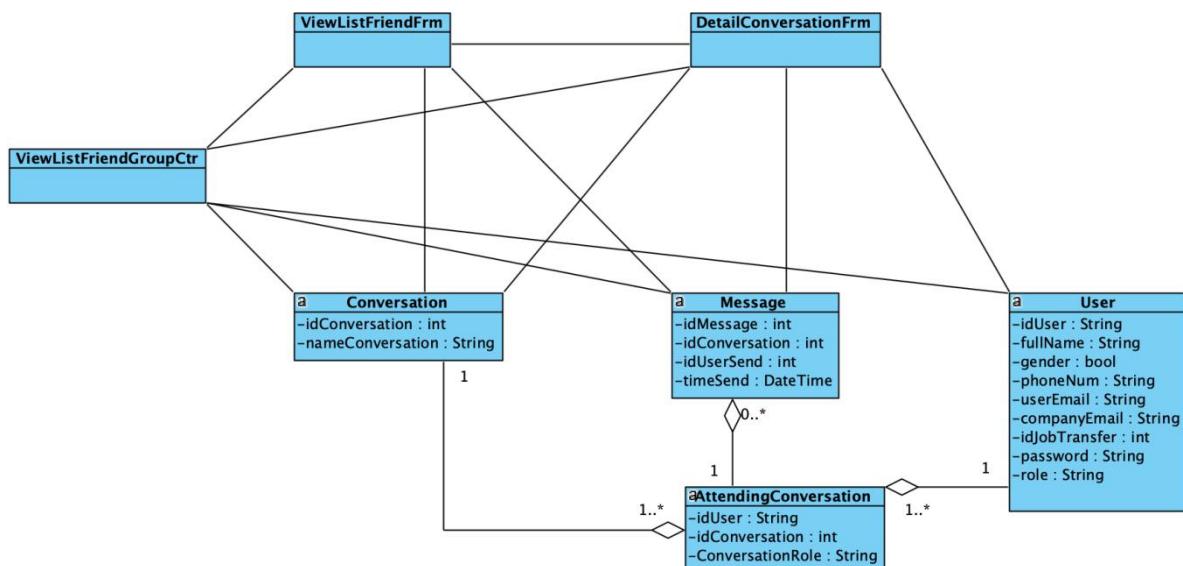
b) Module “Xem danh sách bạn bè/nhóm”

\* Thẻ CRC

ViewListFriendGroupCtr	
<b>Description:</b>	
<b>Attributes:</b>	
Name	Description
<b>Responsibilities:</b>	
Name	Collaborator
Yêu cầu lớp ViewListFriendFrm hiển thị giao diện màn hình chính	ViewListFriendFrm
Yêu cầu lớp DetailConversationFrm hiển thị giao diện chi tiết của 1 cuộc trò chuyện	DetailConversationFrm
Tìm kiếm cuộc trò truyện trong CSDL theo tên	
Tìm kiếm các tin nhắn của 1 cuộc trò chuyện trong CSDL	
Yêu cầu lớp Conversation đóng gói dữ liệu vào đối tượng Conversation	Conversation
Yêu cầu lớp User đóng gói dữ liệu vào đối tượng User	User

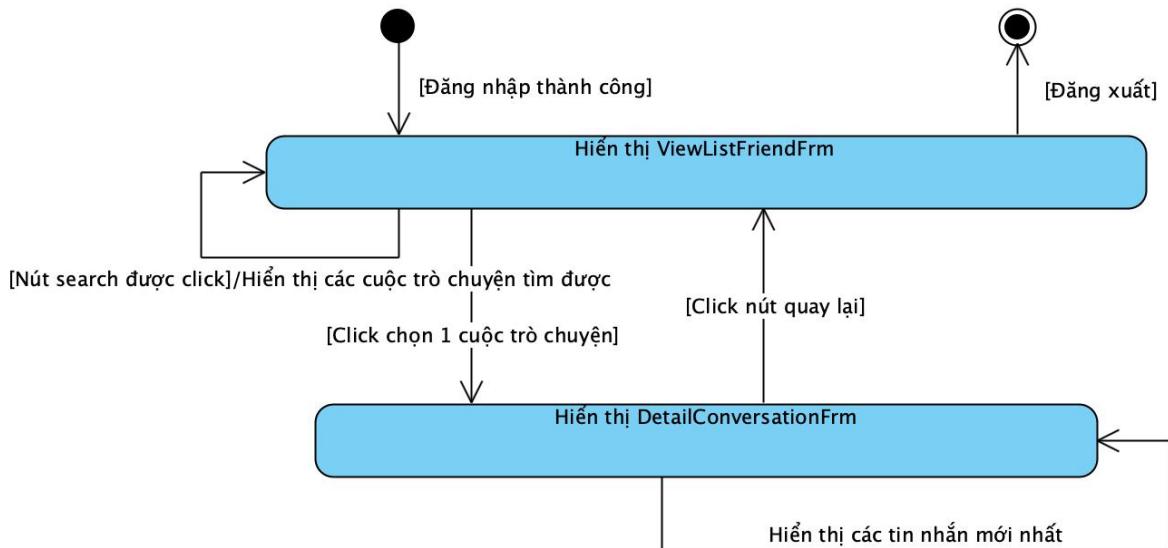
\* Sơ đồ lớp

- Các lớp thực thể: Conversation, User, AttendingConversation.
- Các lớp biên: ViewListFriendFrm, DetailConversationFrm.
- Lớp điều khiển: ViewListFriendGroupCtr.



\* Sơ đồ statechart

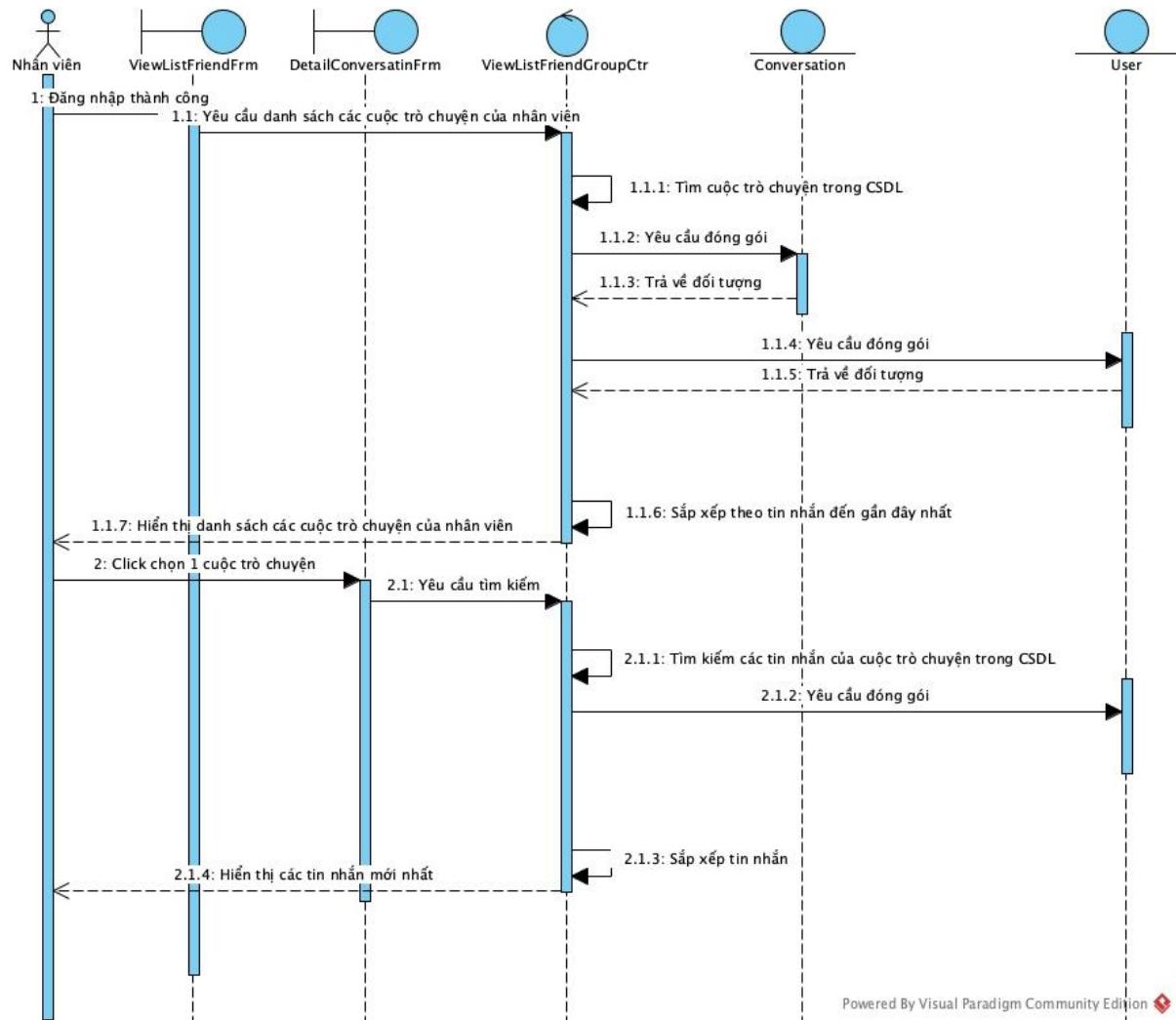
- Có 2 màn hình chính là: ViewListFriendFrm, DetailConversationFrm.



\*Scenario chuẩn cho module “Xem danh sách bạn bè/nhóm”

1. Nhân viên đăng nhập thành công
2. Lớp ViewListFriendFrm gửi yêu cầu tìm danh sách các cuộc trò chuyện của nhân viên đến lớp ViewListFriendGroupCtr.
3. Lớp ViewListFriendGroupCtr tìm kiếm các cuộc trò chuyện của nhân viên trong CSDL.
4. Lớp ViewListFriendGroupCtr yêu cầu đóng gói đối tượng đến lớp Conversation.
5. Lớp Conversation trả về đối tượng Conversation cho lớp ViewListFriendGroupCtr.
6. Lớp ViewListFriendGroupCtr yêu cầu đóng gói đối tượng đến lớp User.
7. Lớp User trả về đối tượng User cho lớp ViewListFriendGroupCtr.
8. Lớp ViewListFriendGroupCtr yêu cầu đóng gói đối tượng đến lớp Message.
9. Lớp Message trả về đối tượng Message cho lớp ViewListFriendGroupCtr.
10. Lớp ViewListFriendGroupCtr sắp xếp cuộc trò chuyện theo các tin nhắn đến gần nhất.
11. Lớp ViewListFriendGroupCtr trả về màn hình hiển thị danh sách các cuộc trò chuyện.
12. Nhân viên chọn 1 cuộc trò chuyện.
13. Lớp DetailConversationFrm yêu cầu tìm kiếm đến ViewListFriendGroupCtr.
14. Lớp ViewListFriendGroupCtr tìm kiếm các tin nhắn của cuộc trò chuyện trong CSDL.
15. Lớp ViewListFriendGroupCtr yêu cầu đóng gói đối tượng đến lớp User.
16. Lớp User trả về đối tượng User cho lớp ViewListFriendGroupCtr.
17. Lớp ViewListFriendGroupCtr yêu cầu đóng gói đối tượng đến lớp Message.
18. Lớp Message trả về đối tượng Message cho lớp ViewListFriendGroupCtr.
19. Lớp ViewListFriendGroupCtr sắp xếp tin nhắn theo trình tự đến gần nhất.
20. Lớp ViewListFriendGroupCtr trả về màn hình hiển thị các tin nhắn mới nhất.

## \* Sơ đồ tuần tự



Powered By Visual Paradigm Community Edition

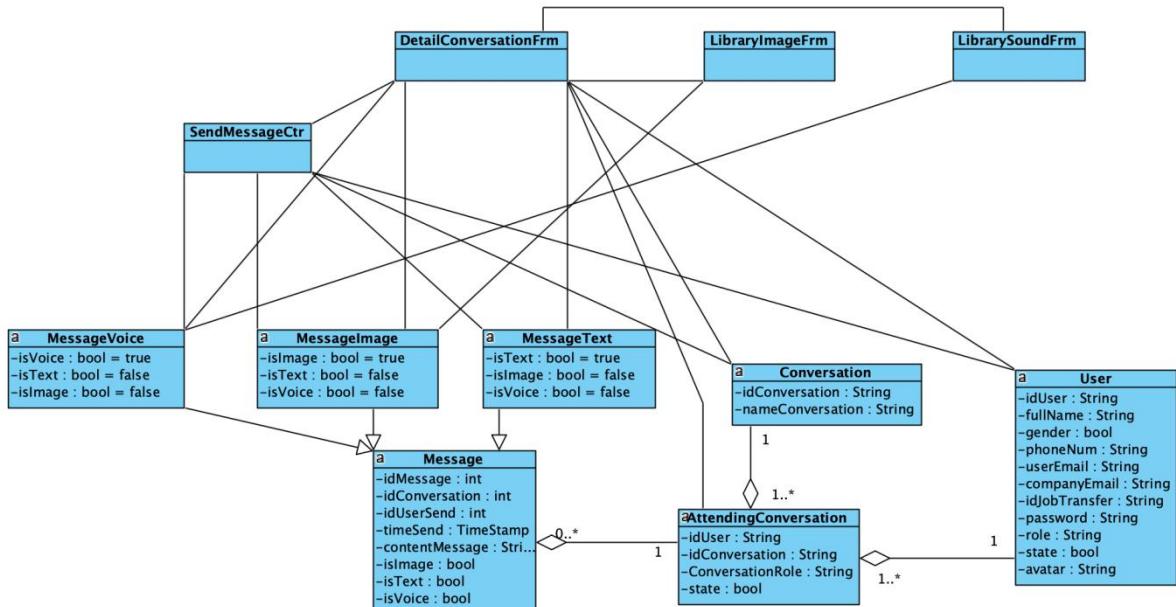
### c) Module “Nhắn tin”

\* Thẻ CRC

SendMessageCtr	
Description:	
Attributes:	
Name	Description
Responsibilities:	
Name	Collaborator
Yêu cầu lớp DetailConversationFrm hiển thị giao diện chi tiết của 1 cuộc trò chuyện	DetailConversationFrm
Yêu cầu lớp LibraryImageFrm hiển thị giao diện chọn ảnh từ thư viện ảnh	LibraryImageFrm
Yêu cầu lớp LibrarySoundFrm hiển thị giao diện chọn file voice từ thư viện âm thanh	LibrarySoundFrm
Tìm kiếm các tin nhắn trong 1 cuộc trò chuyện	
Yêu cầu lớp Conversation đóng gói dữ liệu vào đối tượng Conversation	Conversation
Yêu cầu lớp User đóng gói dữ liệu vào đối tượng User	User
Yêu cầu lớp MessageText đóng gói dữ liệu vào đối tượng MessageText	MessageText
Yêu cầu lớp MessageImage đóng gói dữ liệu vào đối tượng MessageImage	MessageImage
Yêu cầu lớp MessageVoice đóng gói dữ liệu vào đối tượng MessageVoice	MessageVoice

\* Sơ đồ lớp

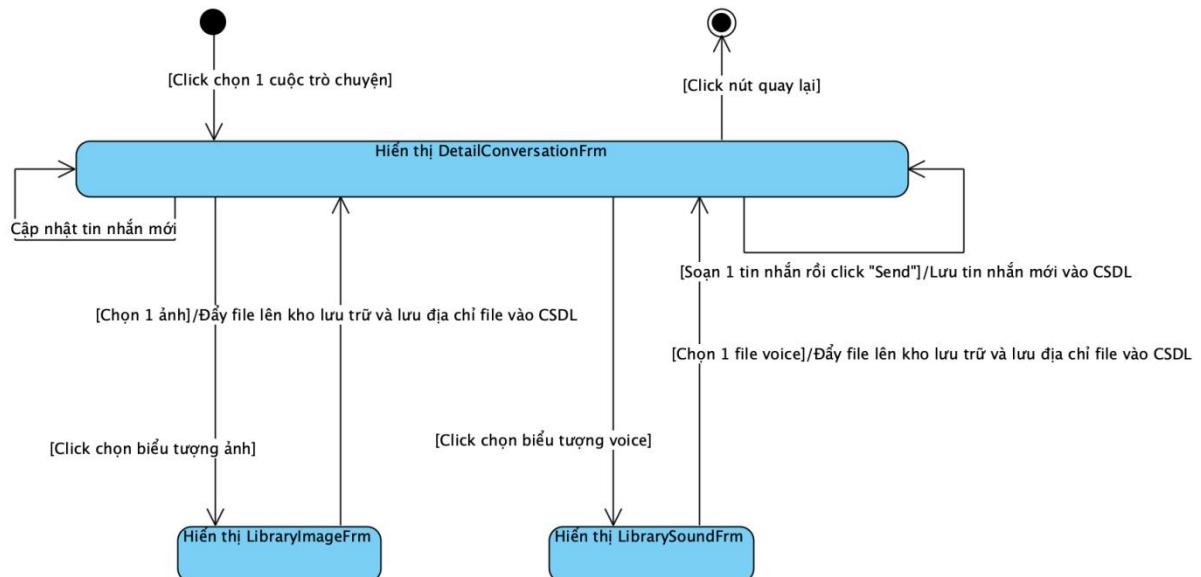
- Lớp thực thể: MessageVoice, MessageImage, MessageText, Conversation, User, AttendingConversation
- Lớp biên: DetailConversationFrm, LibraryImageFrm, LibrarySoundFrm
- Lớp điều khiển: SendMessageCtr



\* Sơ đồ statechart

Có 1 màn hình chính là: DetailConversationFrm

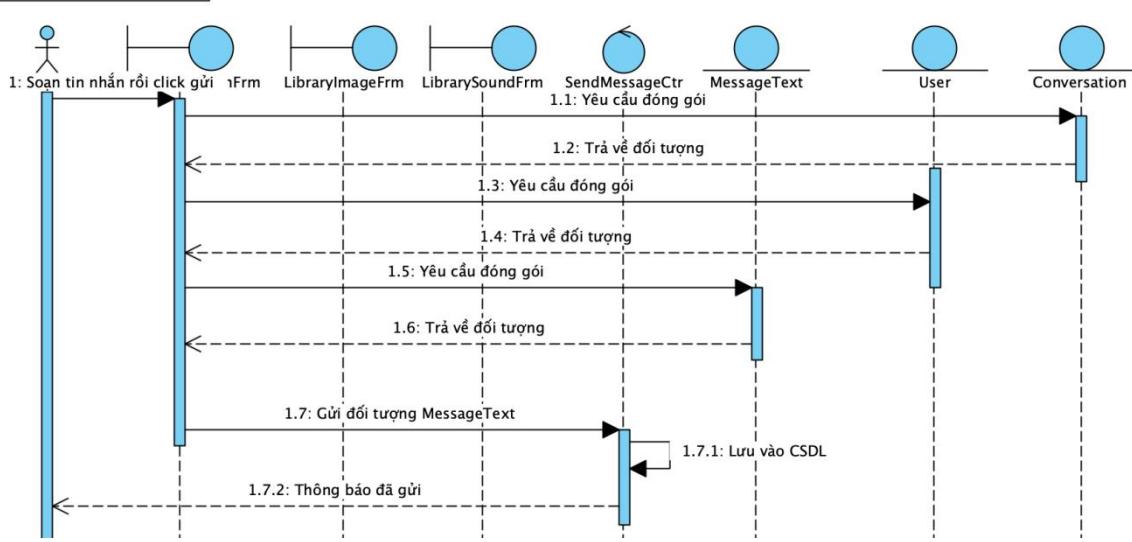
Có 2 màn hình phụ là: LibraryImageFrm, LibrarySoundFrm



\*Scenario chuẩn cho module “Nhắn tin” và sơ đồ tuần tự

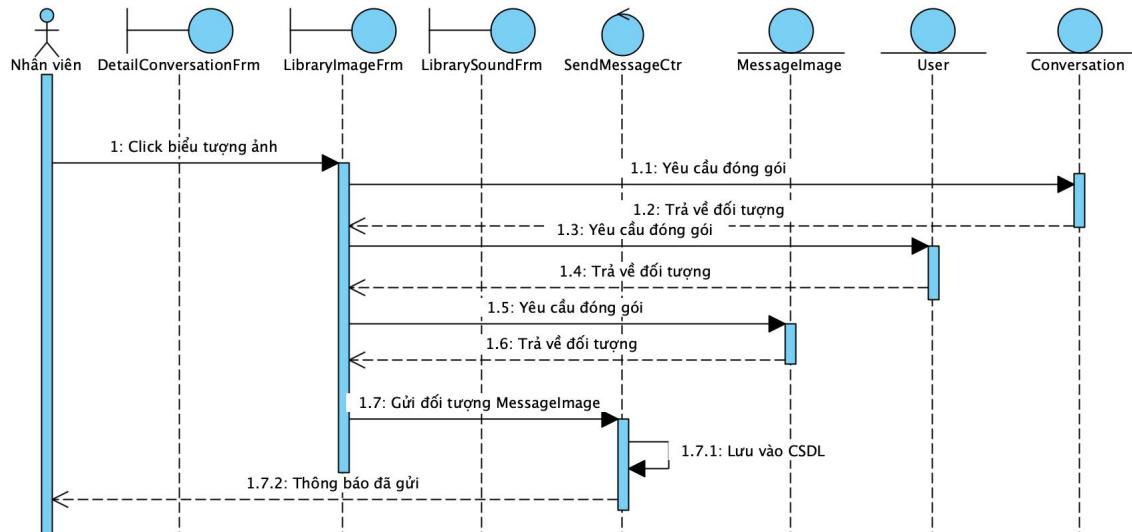
#### - Nhắn tin văn bản

1. Nhân viên soạn tin nhắn rồi click gửi.
2. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailConversationFrm.
4. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp User.
5. Lớp User trả về đối tượng User cho lớp DetailConversationFrm.
6. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp MessageText.
7. Lớp MessageText trả về đối tượng MessageText cho lớp DetailConversationFrm.
8. Lớp DetailConversationFrm gửi đối tượng MessageText cho lớp SendMessageCtr.
9. Lớp SendMessageCtr lưu vào CSDL.
10. Lớp SendMessageCtr trả về thông báo đã gửi thành công.



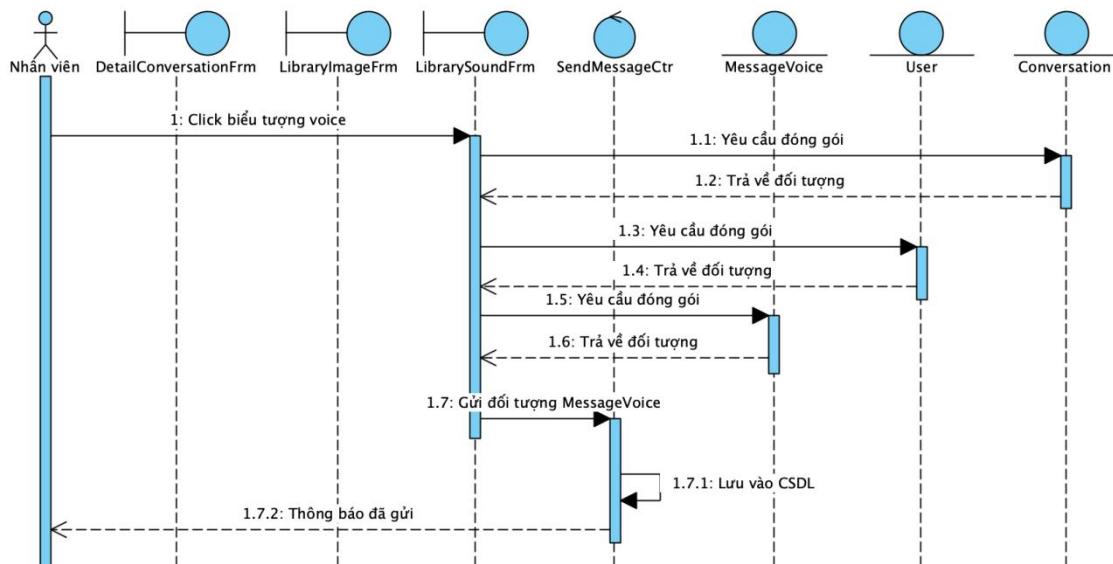
### - Nhắn tin ảnh

1. Nhân viên click vào biểu tượng ảnh rồi chọn ảnh gửi đi.
2. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailConversationFrm.
4. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp User.
5. Lớp User trả về đối tượng User cho lớp DetailConversationFrm.
6. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp MessageImage.
7. Lớp MessageImage trả về đối tượng MessageImage cho lớp DetailConversationFrm.
8. Lớp DetailConversationFrm gửi đối tượng MessageImage cho lớp SendMessageCtr.
9. Lớp SendMessageCtr lưu vào CSDL.
10. Lớp SendMessageCtr trả về thông báo đã gửi thành công.



### - Nhắn tin giọng nói

1. Nhân viên click vào biểu tượng voice rồi thu âm giọng nói.
2. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailConversationFrm.
4. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp User.
5. Lớp User trả về đối tượng User cho lớp DetailConversationFrm.
6. Lớp DetailConversationFrm yêu cầu đóng gói đối tượng đến lớp MessageVoice.
7. Lớp MessageImage trả về đối tượng MessageVoice cho lớp DetailConversationFrm.
8. Lớp DetailConversationFrm gửi đối tượng MessageVoice cho lớp SendMessageCtr.
9. Lớp SendMessageCtr lưu vào CSDL.
10. Lớp SendMessageCtr trả về thông báo đã gửi thành công.



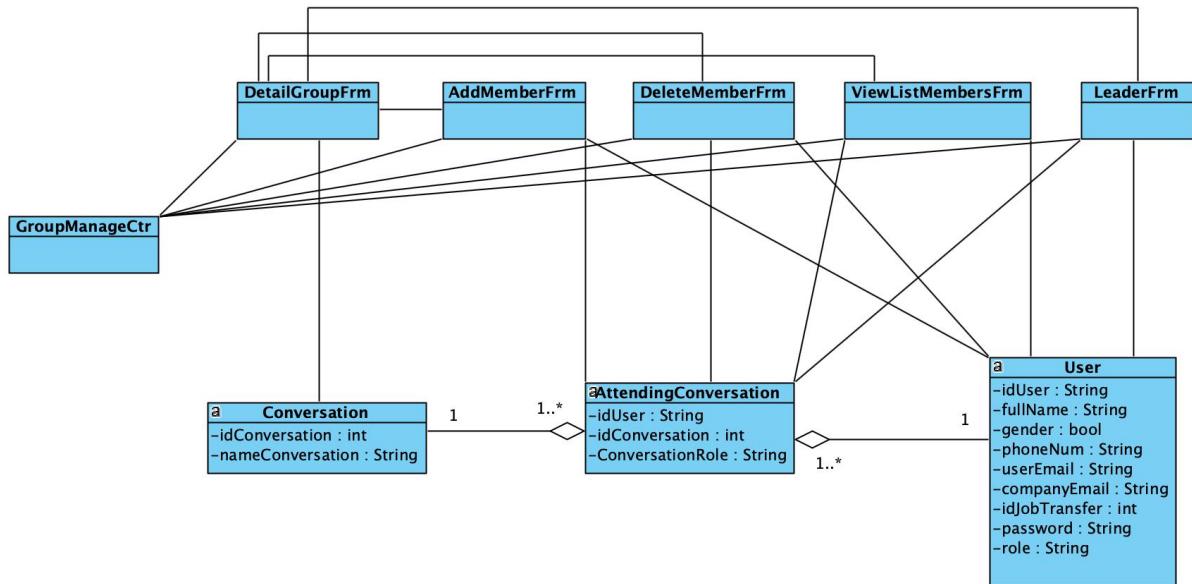
#### d) Module “Quản lý nhóm”

\* Thẻ CRC

GroupManageCtr	
Responsibilities:	
Name	Collaborator
Yêu cầu lớp DetailGroupFrm hiển thị các mục thông tin của nhóm	DetailGroupnFrm
Yêu cầu lớp AddMemberFrm hiển thị giao diện thêm thành viên	AddMemberFrm
Yêu cầu lớp DeleteMemberFrm hiển thị giao diện xóa thành viên	DeleteMemberFrm
Yêu cầu lớp ViewListMembersFrm hiển thị giao diện xem danh sách thành viên của nhóm	ViewListMemberFrm
Yêu cầu lớp LeaderFrm hiển thị giao diện chuyển quyền trưởng nhóm	LeaderFrm
Tìm kiếm người dùng trong CSDL theo tên	
Lưu thông tin thành viên mới của nhóm vào CSDL	
Xóa thông tin thành viên mới của nhóm khỏi CSDL	
Cập nhật thông tin về vai trò trưởng nhóm trong CSDL	
Yêu cầu lớp User đóng gói dữ liệu vào đối tượng User	User
Yêu cầu lớp AttendingConversation đóng gói dữ liệu vào đối tượng AttendingConversation	AttendingConversation

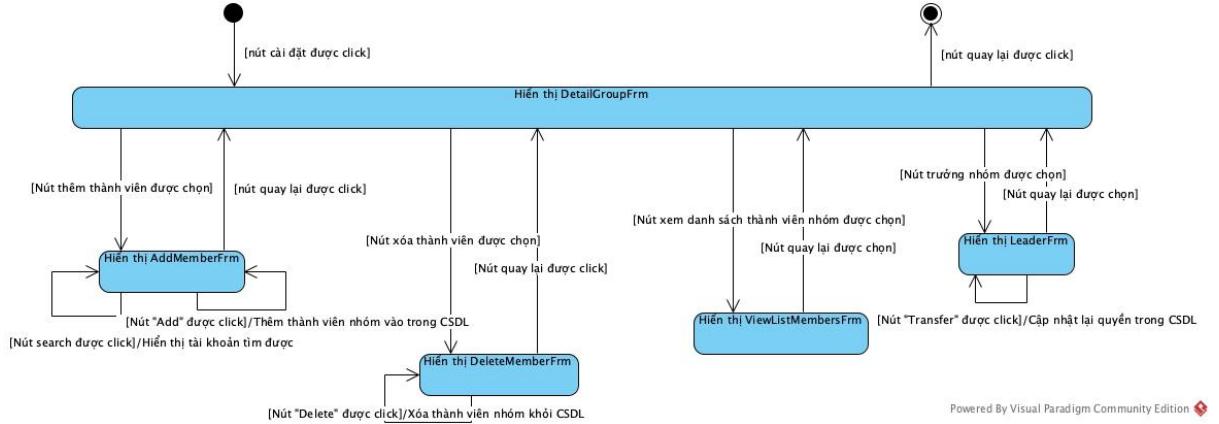
\* Sơ đồ lớp

- Lớp thực thể: Conversation, AttendingConversation, User.
- Lớp biên: DetailGroupFrm, AddMemberFrm, DeleteMemberFrm, ViewListMemberFrm, LeaderFrm.
- Lớp điều khiển: GroupManageCtr.



\* Sơ đồ statechart

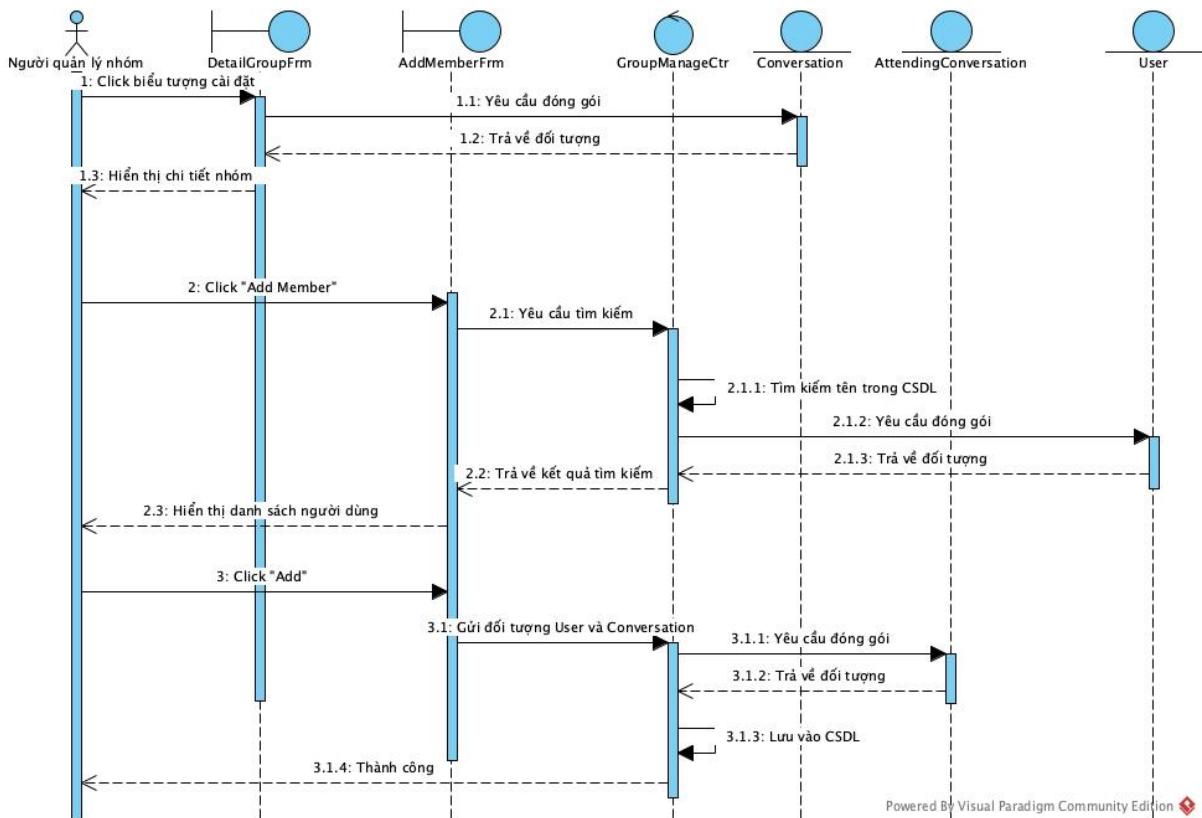
Có 5 màn hình: DetailGroupFrm, AddMemberFrm, DeleteMemberFrm, ViewListMemberFrm, LeaderFrm.



\* Scenario chuẩn cho module “Quản lý nhóm” và sơ đồ tuần tự

#### - Add Member

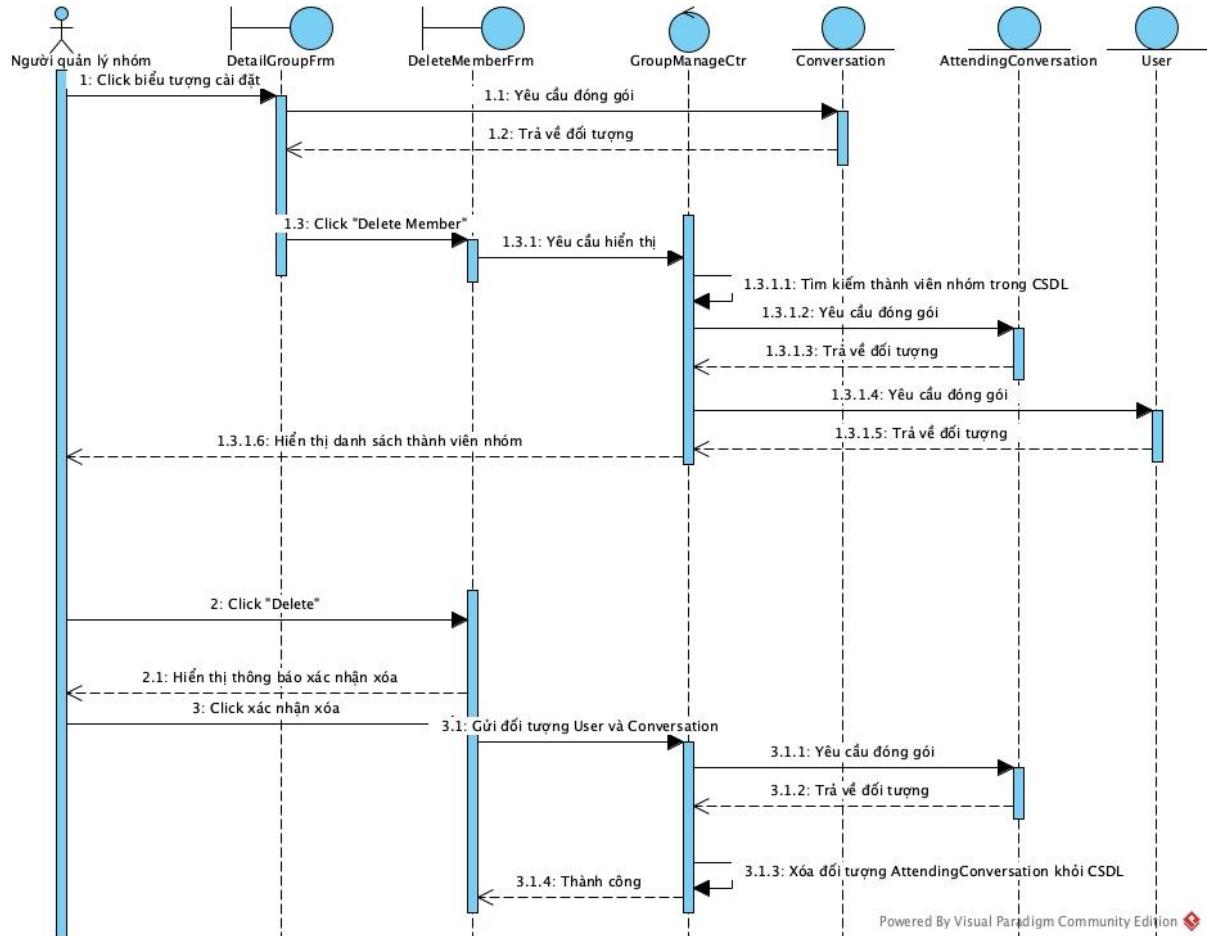
1. Người quản lý nhóm click vào biểu tượng cài đặt trên màn hình chi tiết cuộc trò chuyện nhóm.
2. Lớp DetailGroupFrm gửi yêu cầu đóng gói tới lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailGroupFrm.
4. Lớp DetailGroupFrm hiển thị các mục thông tin chi tiết về nhóm.
5. Người quản lý nhóm click “Add Member”.
6. Hệ thống hiện giao diện AddMemberFrm có thanh tìm kiếm người dùng.
7. Người quản lý nhóm nhập tên tài khoản rồi search.
8. Lớp AddMemberFrm yêu cầu tìm kiếm đến lớp GroupManageCtr.
9. Lớp GroupManageCtr tìm kiếm theo tên trong CSDL.
10. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp User.
11. Lớp User trả về đối tượng User cho lớp GroupManageCtr.
12. Lớp GroupManageCtr trả về kết quả tìm kiếm cho AddMemberFrm.
13. Lớp AddMemberFrm hiển thị danh sách nhân viên trùng tên.
14. Người quản lý người dùng click “Add”.
15. Lớp AddMemberFrm gửi đối tượng User và Conversation đến lớp GroupManageCtr.
16. Lớp GroupManageCtr yêu cầu lớp AttendingCoversation đóng gói.
17. Lớp AttendingCoversation trả về đối tượng AttendingCoversation cho lớp GroupManageCtr.
18. Lớp GroupManageCtr lưu vào CSDL.
19. Lớp GroupManageCtr trả về thông báo thành công cho người dùng.



### - Delete Member

1. Người quản lý nhóm click vào biểu tượng cài đặt trên màn hình chi tiết cuộc trò chuyện nhóm.
2. Lớp DetailGroupFrm gửi yêu cầu đóng gói tới lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailGroupFrm.
4. Lớp DetailGroupFrm hiển thị các mục thông tin chi tiết về nhóm.
5. Người quản lý nhóm click “Delete Member”.
6. Lớp DeleteMemberFrm gửi yêu cầu hiển thị đến lớp GroupManagerCtr.
7. Người quản lý nhóm nhập tên tài khoản rồi search.
8. Lớp GroupManagerCtr tìm kiếm thành viên nhóm trong CSDL.
9. Lớp GroupManagerCtr gửi yêu cầu đóng gói đến lớp AttendingConverstion.
10. Lớp AttendingConverstion trả về đối tượng AttendingConverstion cho lớp GroupManageCtr.
11. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp User.
12. Lớp User trả về đối tượng User cho lớp GroupManageCtr.
13. Lớp GroupManageCtr hiển thị kết quả danh sách thành viên nhóm cho người quản lý nhóm.
14. Người quản lý người dùng click “Delete”.
15. Lớp DeleteMemberFrm hiển thị thông báo xác nhận xóa.
16. Người quản lý người dùng click “Yes”
17. Lớp DeleteMemberFrm gửi đối tượng User và Conversation đến lớp GroupManageCtr.
18. Lớp GroupManageCtr yêu cầu lớp AttendingCoversation đóng gói.

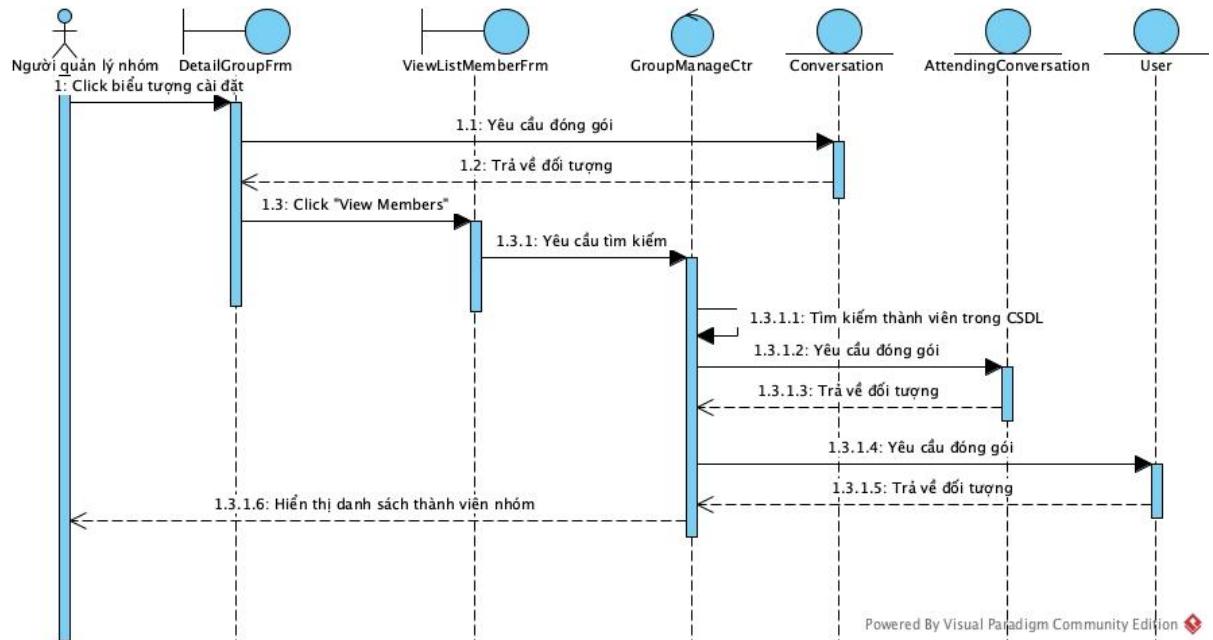
19. Lớp AttendingCoversation trả về đối tượng AttendingCoversation cho lớp GroupManageCtr.
20. Lớp GroupManageCtr xóa đối tượng khỏi CSDL.
21. Lớp GroupManageCtr trả về thông báo xóa thành công cho người dùng.



#### - View List Members

1. Người quản lý nhóm click vào biểu tượng cài đặt trên màn hình chi tiết cuộc trò chuyện nhóm.
2. Lớp DetailGroupFrm gửi yêu cầu đóng gói tới lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailGroupFrm.
4. Lớp DetailGroupFrm hiển thị các mục thông tin chi tiết về nhóm.
5. Người quản lý nhóm click “View Members”.
6. Lớp ViewListMemberFrm gửi yêu cầu tìm kiếm đến lớp GroupManagerCtr.
7. Lớp GroupManagerCtr tìm kiếm thành viên nhóm trong CSDL.
8. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp AttendingConverstion.
9. Lớp AttendingConverstion trả về đối tượng AttendingConverstion cho lớp GroupManageCtr.
10. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp User.
11. Lớp User trả về đối tượng User cho lớp GroupManageCtr.

12. Lớp GroupManageCtr hiển thị kết quả danh sách thành viên nhóm cho người quản lý nhóm.

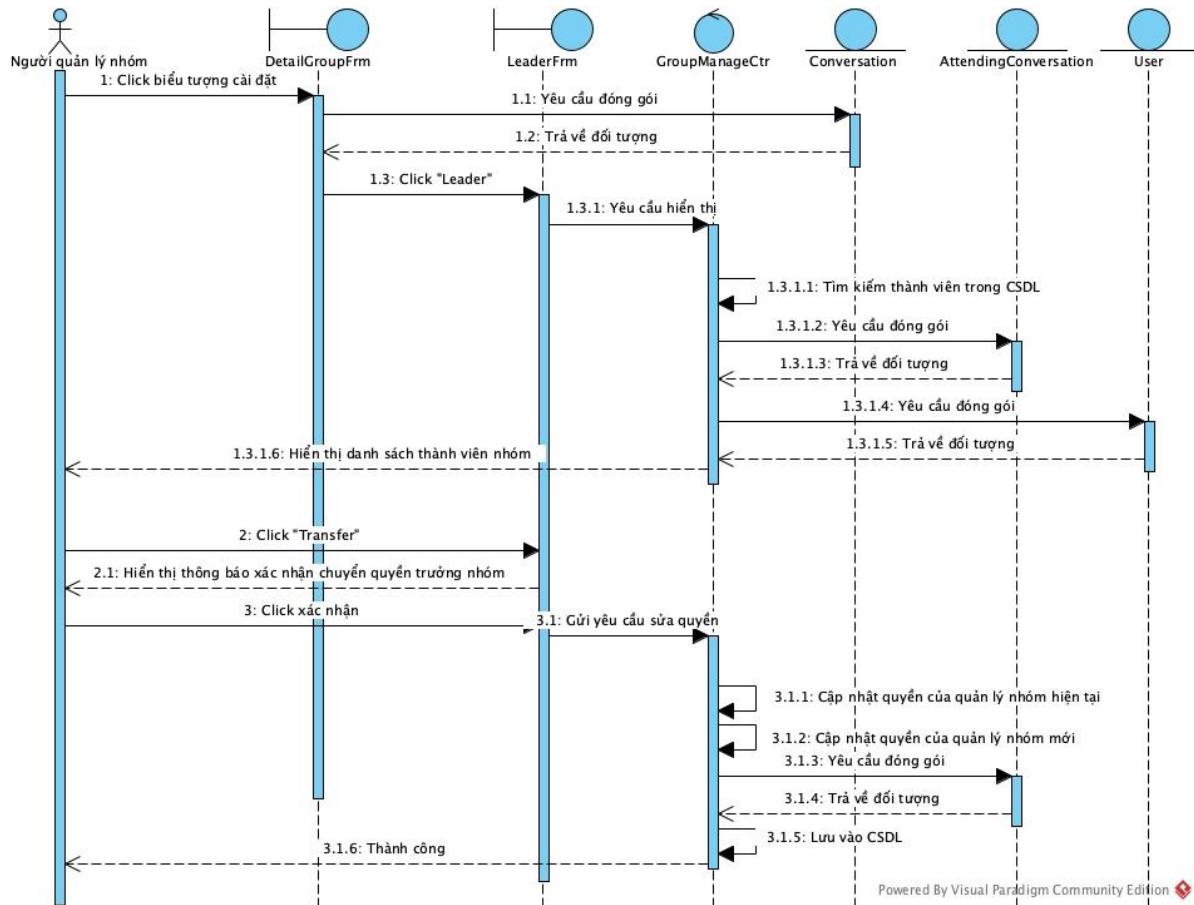


### - Leader

1. Người quản lý nhóm click vào biểu tượng cài đặt trên màn hình chi tiết cuộc trò chuyện nhóm.
2. Lớp DetailGroupFrm gửi yêu cầu đóng gói tới lớp Conversation.
3. Lớp Conversation trả về đối tượng Conversation cho lớp DetailGroupFrm.
4. Lớp DetailGroupFrm hiển thị các mục thông tin chi tiết về nhóm.
5. Người quản lý nhóm click “Leader”.
6. Lớp LeaderFrm gửi yêu cầu hiển thị đến lớp GroupManagerCtr.
7. Lớp GroupManagerCtr tìm kiếm thành viên nhóm trong CSDL.
8. Lớp GroupManagerCtr gửi yêu cầu đóng gói đến lớp AttendingConverstion.
9. Lớp AttendingConverstion trả về đối tượng AttendingConverstion cho lớp GroupManageCtr.
10. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp User.
11. Lớp User trả về đối tượng User cho lớp GroupManageCtr.
12. Lớp GroupManageCtr hiển thị kết quả danh sách thành viên nhóm cho người quản lý nhóm.
13. Người quản lý nhóm click “Transfer”.
14. Hệ thống hiển thị thông báo xác nhận chuyển quyền trưởng nhóm.
15. Người quản lý nhóm click “Yes”
16. Lớp LeaderFrm gửi yêu cầu sửa quyền cho lớp GroupManageCtr.
17. Lớp GroupManageCtr cập nhật quyền của quản lý nhóm hiện tại.
18. Lớp GroupManageCtr cập nhật quyền của quản lý nhóm mới.
19. Lớp GroupManageCtr gửi yêu cầu đóng gói đến lớp AttendingConversation.
20. Lớp AttendingConversation trả về đối tượng AttendingConversation cho lớp GroupManageCtr.

21. Lớp GroupManageCtr lưu thay đổi vào trong CSDL.

22. Lớp GroupManageCtr gửi thông báo đã chuyển quyền thành công.



### e) Module “Quản lý người dùng”

#### \* Thẻ CRC

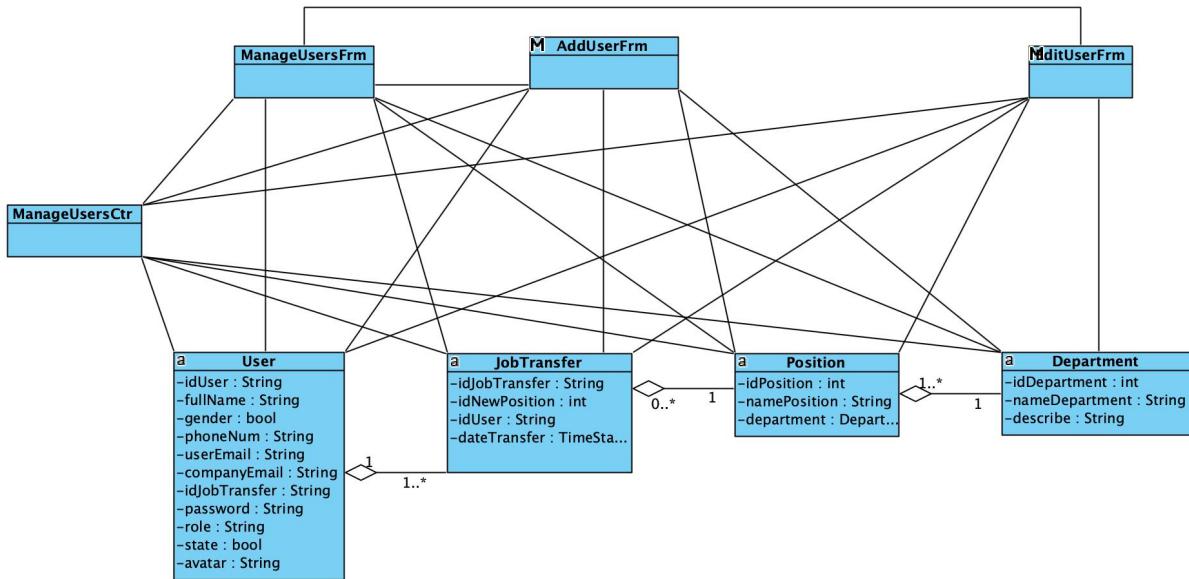
ManageUsersCtr	
Name	Collaborator
Yêu cầu lớp ManageUsersFrm hiển thị giao diện quản lý tài khoản người dùng.	ManageUsersFrm
Yêu cầu lớp AddUserFrm hiển thị giao diện thêm tài khoản người dùng	AddUserFrm
Yêu cầu lớp EditUserFrm hiển thị giao diện sửa tài khoản người dùng	EditUserFrm
Lưu thông tin tài khoản người dùng mới vào trong CSDL	
Tìm kiếm tài khoản người dùng theo tên	
Cập nhật thông tin vừa sửa vào CSDL	
Yêu cầu lớp User đóng gói dữ liệu vào đối tượng Users	User
Yêu cầu lớp JobTransfer đóng gói dữ liệu vào đối tượng JobTransfer	JobTransfer
Yêu cầu lớp Postion đóng gói dữ liệu vào đối tượng Position	Position
Yêu cầu lớp Department đóng gói dữ liệu vào đối tượng Department	Department

Double click: edit Class Name;

#### \* Sơ đồ lớp

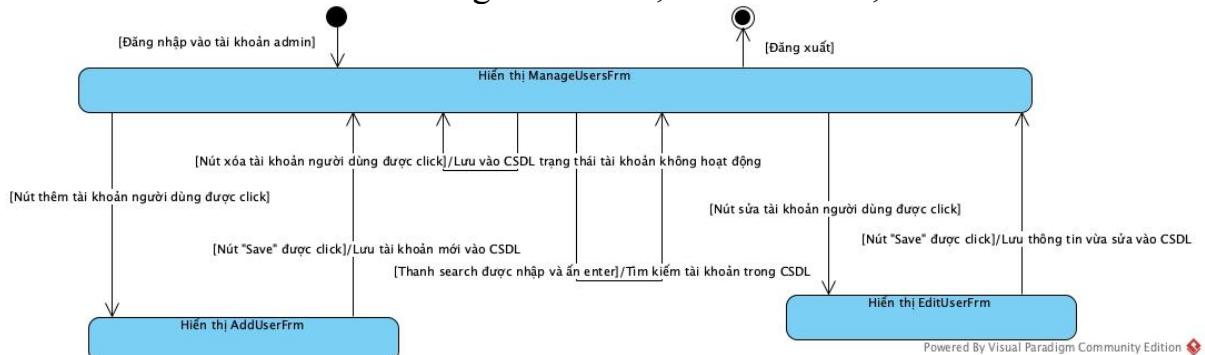
- Lớp thực thể: User, JobTransfer, Position, Department

- Lớp biên: ManageUsersFrm, AddUserFrm, EditUserFrm
- Lớp điều khiển: ManageUsersCtr



\* Sơ đồ statechart

- Có 3 màn hình chính là: ManageUsersFrm, AddUserFrm, EditUserFrm

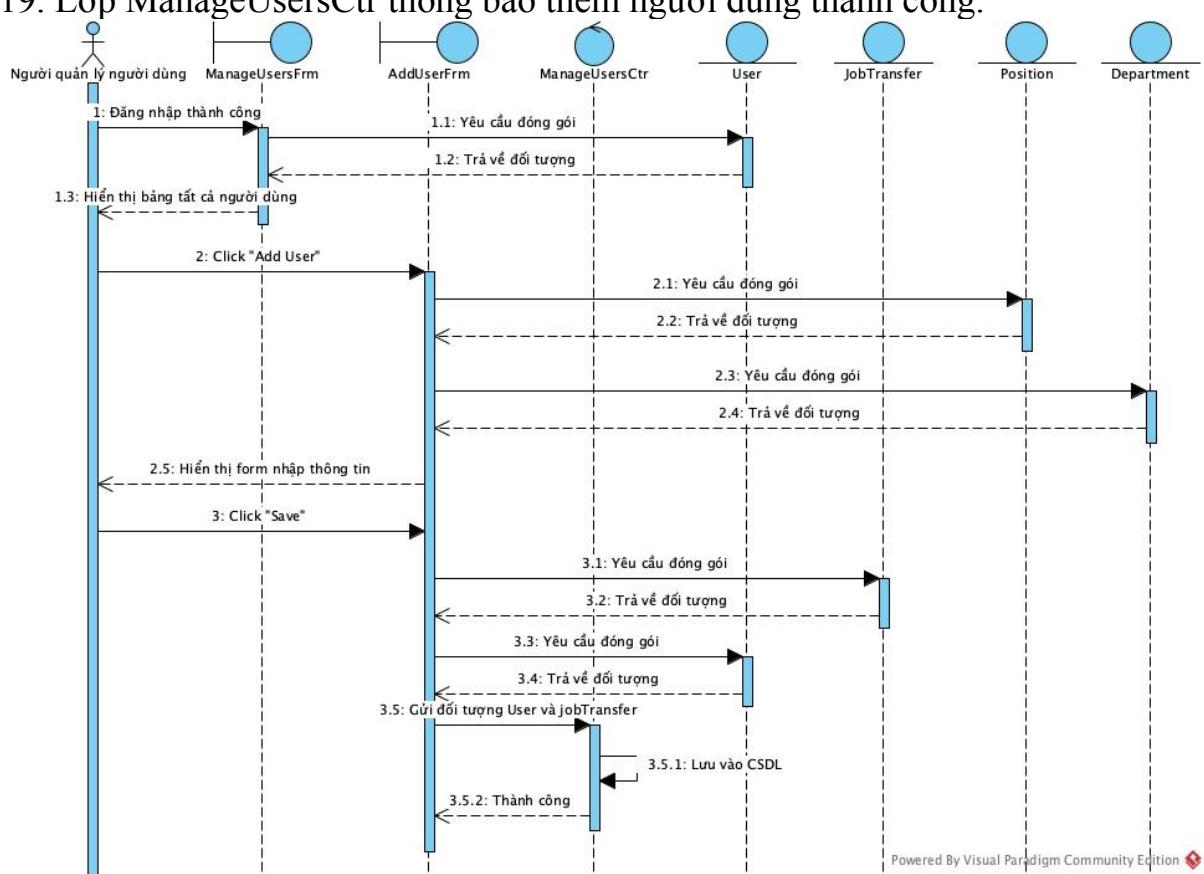


\* Scenario chuẩn cho module “Quản lý người dùng” và sơ đồ tuần tự

#### - Add User

1. Người quản lý người dùng đăng nhập thành công.
2. Lớp ManageUsersFrm gửi yêu cầu đóng gói đến lớp User.
3. Lớp User đóng gói đối tượng User và trả về cho lớp ManageUsersFrm đối tượng User.
4. Lớp ManageUsersFrm hiển thị bảng tất cả người dùng.
5. Người quản lý người dùng click “Add User”.
6. Lớp AddUserFrm hiển thị form nhập thông tin.
7. Lớp AddUserFrm gửi yêu cầu đóng gói đến lớp Position.
8. Lớp Position đóng gói đối tượng Position và trả về cho lớp AddUserFrm đối tượng Position.
9. Lớp AddUserFrm gửi yêu cầu đóng gói đến lớp Department.
10. Lớp Department đóng gói đối tượng Department và trả về cho lớp AddUserFrm đối tượng Department.
11. Lớp AddUserFrm hiển thị form nhập thông tin người dùng mới.
12. Người quản lý người dùng nhập thông tin người dùng mới rồi click “Save”.

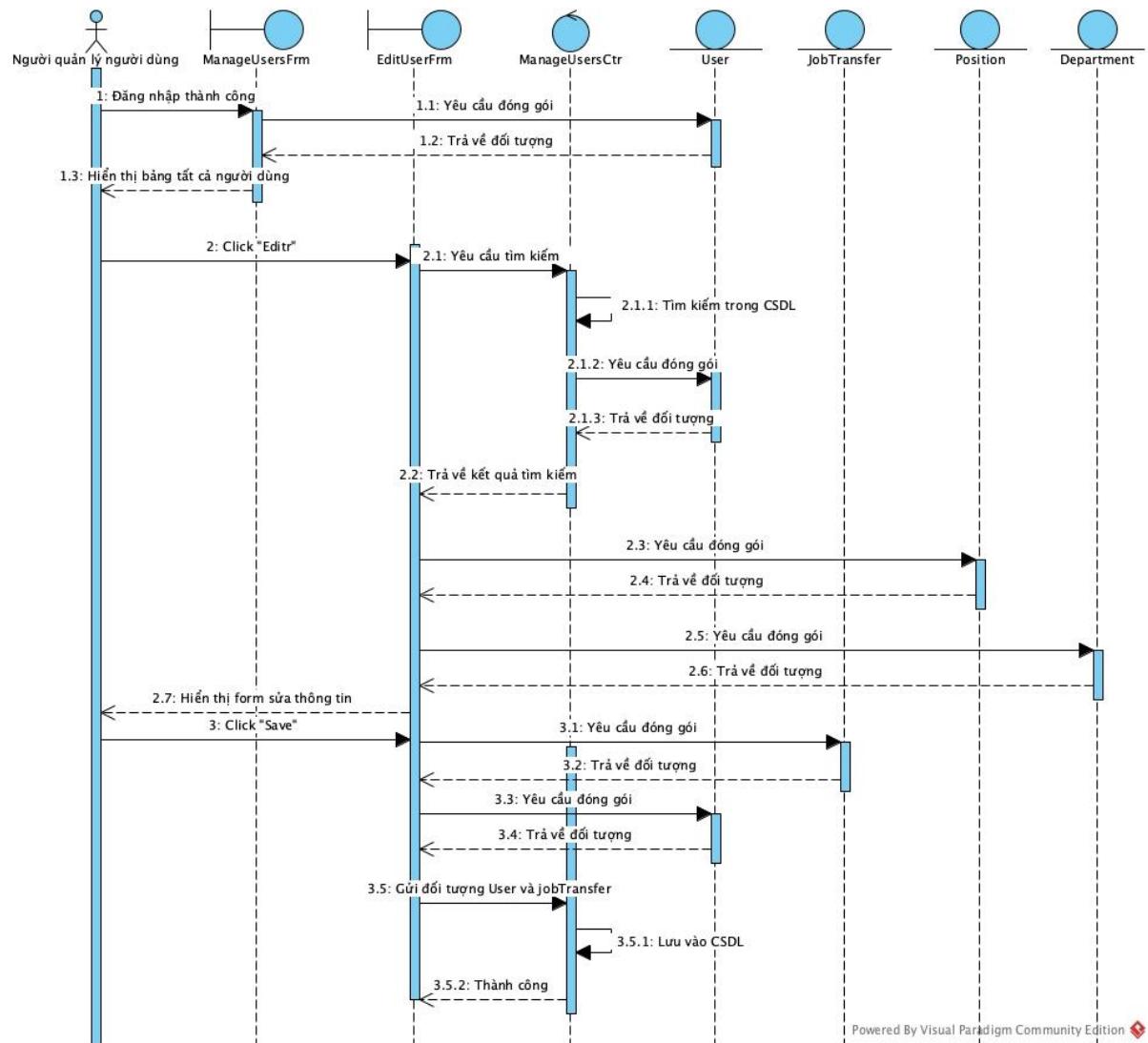
13. Lớp AddUserFrm gửi yêu cầu đóng gói đến lớp JobTransfer.
14. Lớp JobTransfer đóng gói đối tượng JobTransfer và trả về cho lớp AddUserFrm đối tượng JobTransfer.
15. Lớp AddUserFrm gửi yêu cầu đóng gói đến lớp User.
16. Lớp User đóng gói đối tượng User và trả về cho lớp AddUserFrm đối tượng User.
17. Lớp AddUserFrm gửi đối tượng User và JobTransfer cho lớp ManageUsersCtr.
18. Lớp ManageUsersCtr lưu vào CSDL.
19. Lớp ManageUsersCtr thông báo thêm người dùng thành công.



### - Edit User

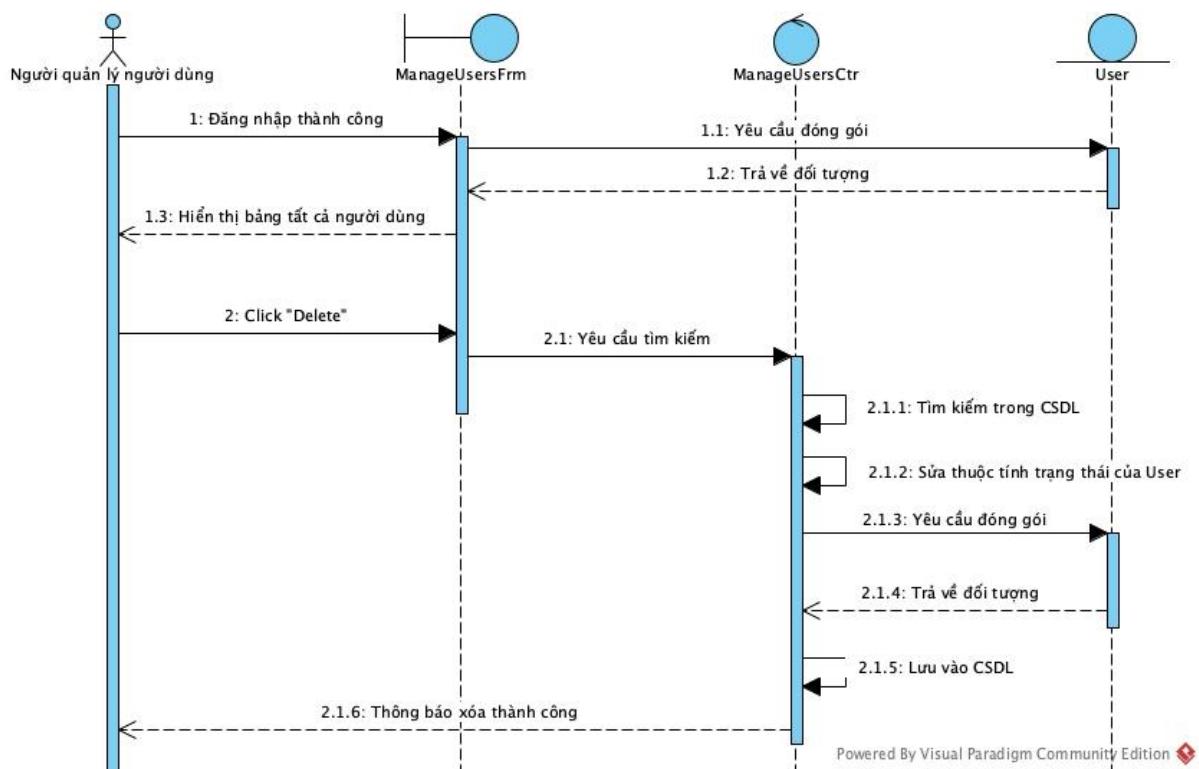
1. Người quản lý người dùng đăng nhập thành công.
2. Lớp ManageUsersFrm gửi yêu cầu đóng gói đến lớp User.
3. Lớp User đóng gói đối tượng User và trả về cho lớp ManageUsersFrm đối tượng User.
4. Lớp ManageUsersFrm hiển thị bảng tất cả người dùng.
5. Người quản lý người dùng click “Edit”.
6. Lớp EditUserFrm gửi yêu cầu tìm kiếm đến ManageUsersCtr.
7. Lớp ManageUsersCtr tìm kiếm trong CSDL thông tin người dùng.
8. Lớp ManageUsersCtr gửi yêu cầu đóng gói đến lớp User.
9. Lớp User đóng gói đối tượng User và trả về cho lớp ManageUsersCtr đối tượng User.
10. Lớp ManageUsersCtr trả về kết quả tìm kiếm cho lớp EditUserFrm.
11. Lớp EditUserFrm gửi yêu cầu đóng gói đến lớp Position.

12. Lớp Position đóng gói đối tượng Position và trả về cho lớp EditUserFrm đối tượng Position.
13. Lớp EditUserFrm gửi yêu cầu đóng gói đến lớp Department.
14. Lớp Department đóng gói đối tượng Department và trả về cho lớp EditUserFrm đối tượng Department.
15. Lớp EditUserFrm hiển thị form sửa thông tin người dùng.
16. Người quản lý người dùng sửa thông tin người dùng mới rồi click “Save”.
17. Lớp EditUserFrm gửi yêu cầu đóng gói đến lớp JobTransfer.
18. Lớp JobTransfer đóng gói đối tượng JobTransfer và trả về cho lớp EditUserFrm đối tượng JobTransfer.
19. Lớp EditUserFrm gửi yêu cầu đóng gói đến lớp User.
20. Lớp User đóng gói đối tượng User và trả về cho lớp EditUserFrm đối tượng User.
21. Lớp EditUserFrm gửi đối tượng User và JobTransfer cho lớp ManageUsersCtr.
22. Lớp ManageUsersCtr lưu vào CSDL.
23. Lớp ManageUsersCtr thông báo sửa người dùng thành công.



### - Delete User

1. Người quản lý người dùng đăng nhập thành công.
2. Lớp ManageUsersFrm gửi yêu cầu đóng gói đến lớp User.
3. Lớp User đóng gói đối tượng User và trả về cho lớp ManageUsersFrm đối tượng User.
4. Lớp ManageUsersFrm hiển thị bảng tất cả người dùng.
5. Người quản lý người dùng click “Delete”.
6. Lớp ManageUserFrm gửi yêu cầu tìm kiếm đến ManageUsersCtr.
7. Lớp ManageUsersCtr tìm kiếm trong CSDL thông tin người dùng.
8. Lớp ManageUsersCtr sửa thuộc tính trạng thái của User.
9. Lớp ManageUsersCtr gửi yêu cầu đóng gói đến lớp User.
10. Lớp ManageUsersCtr lưu vào CSDL.
11. Lớp ManageUsersCtr thông báo xóa người dùng thành công.



### f) Module “Quản lý nhóm của quản lý người dùng”

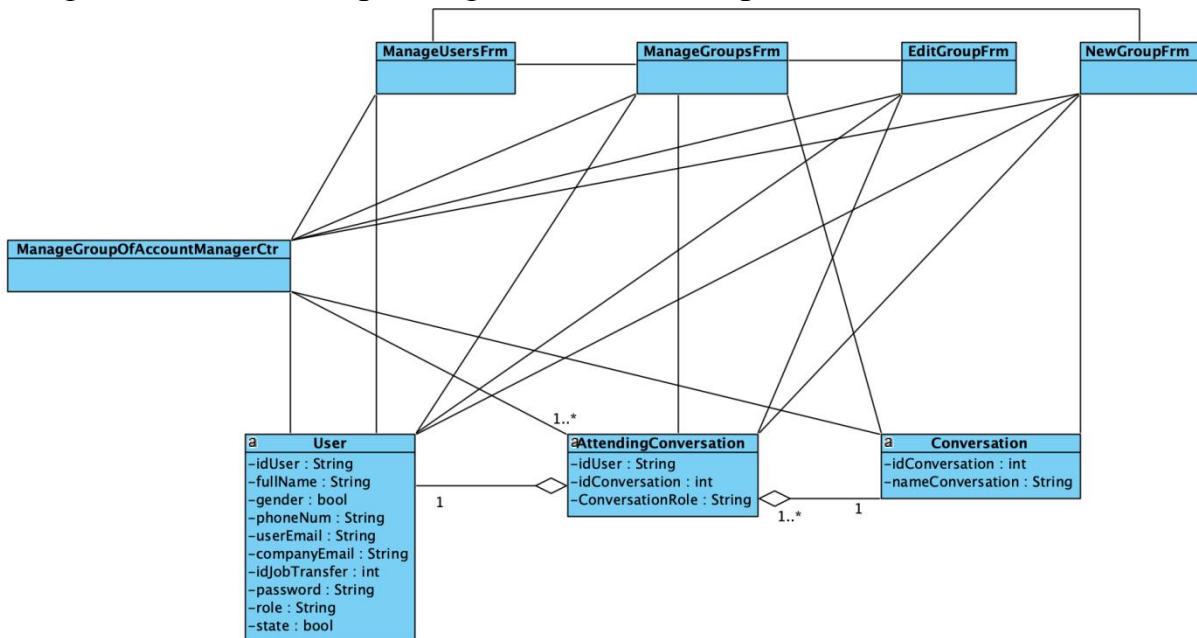
\* Thẻ CRC

ManageGroupOfAccountManagerCtr	
Name	Collaborator
Yêu cầu lớp ManageUsersFrm hiển thị giao diện quản lý tài khoản người dùng	ManageUsersFrm
Yêu cầu lớp ManageGroupsFrm hiển thị giao diện quản lý tất cả các nhóm	ManageGroupsFrm
Yêu cầu lớp EditGroupFrm hiển thị giao diện sửa nhóm	EditGroupFrm
Yêu cầu lớp NewGroupFrm hiển thị giao diện tạo nhóm mới	NewGroupFrm
Lưu thông tin nhóm vào CSDL	
Tìm kiếm nhóm theo tên	
Cập nhật thông tin vừa sửa vào CSDL	
Yêu cầu lớp User đóng gói dữ liệu	User
Yêu cầu lớp Conversation đóng gói dữ liệu	Conversation
Yêu cầu lớp AttendingConversation đóng gói dữ liệu	AttendingConversation

Double click: edit Responsibility Collaborator; Right click: add Responsibility; remove Responsibility

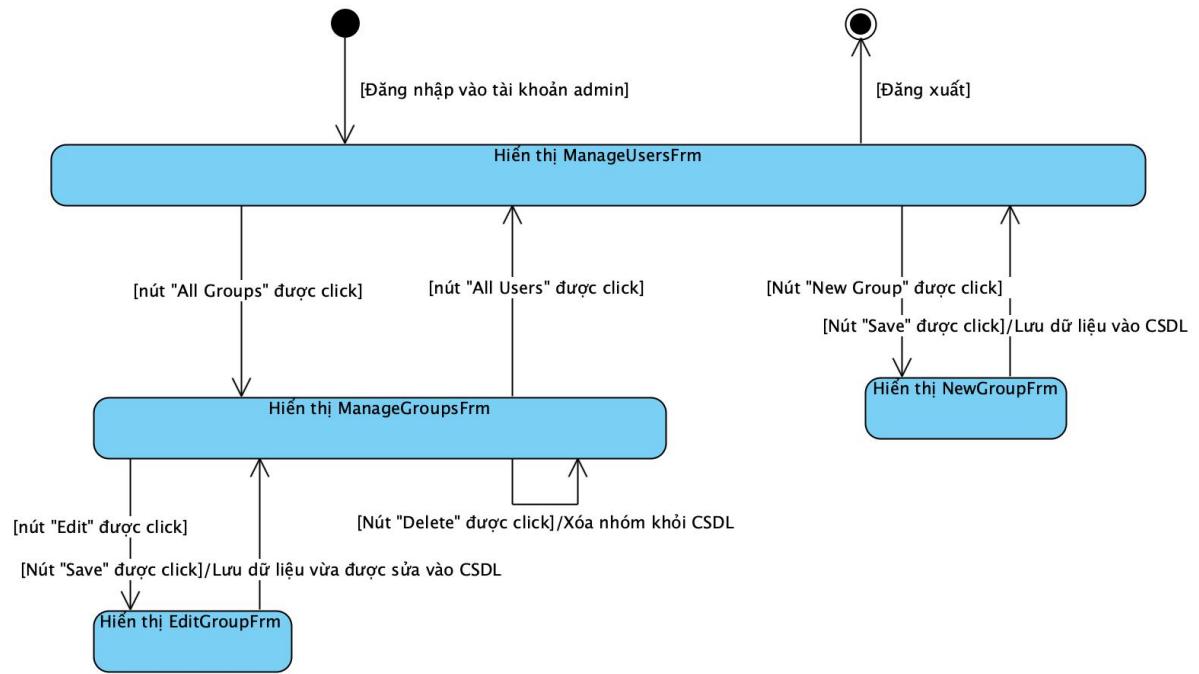
\* Sơ đồ lớp

- Lớp thực thể: User, AttendingConversation, Conversation
- Lớp biên: ManageUsersFrm, ManageGroupsFrm, EditGroupFrm, NewGroupFrm
- Lớp điều khiển: ManageGroupOfAccountManagerCtr



\* Sơ đồ statechart

- Có 4 màn hình chính: ManageUsersFrm, ManageGroupsFrm, EditGroupFrm, NewGroupFrm

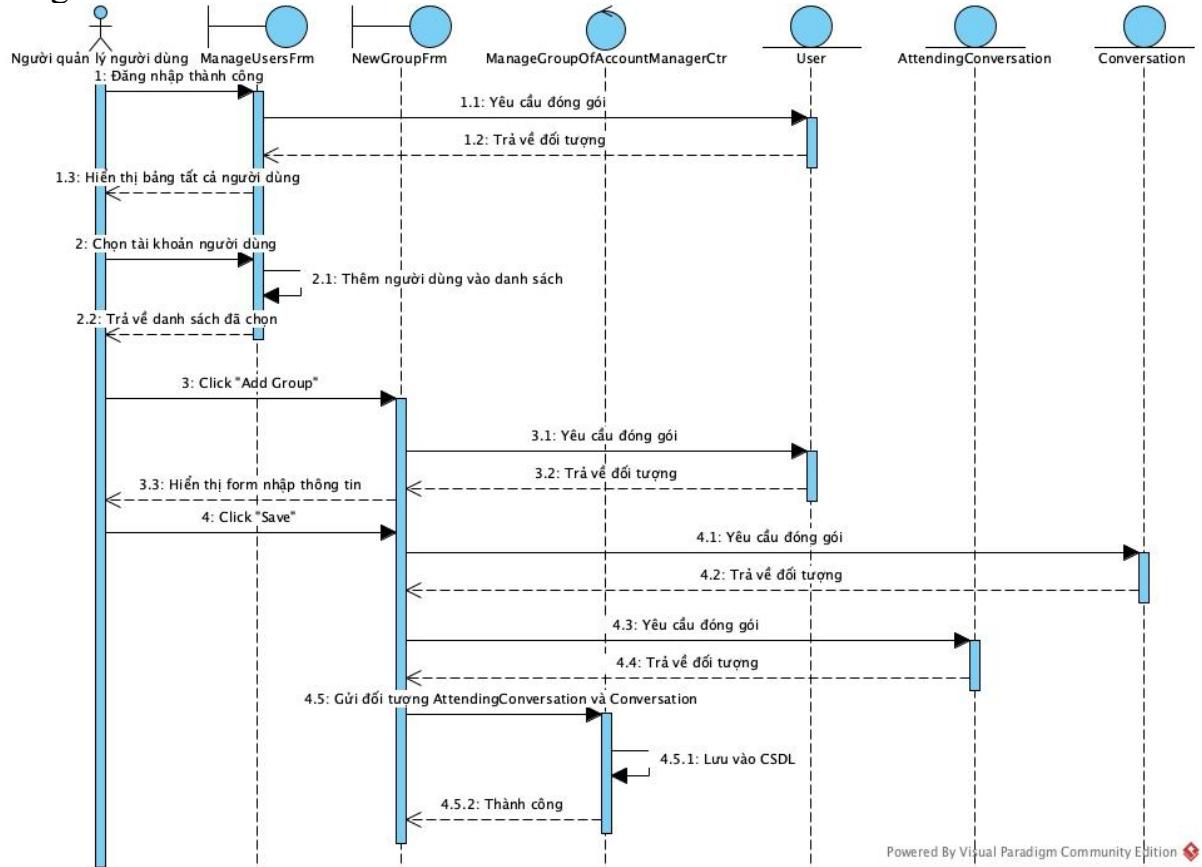


\* Scenario chuẩn cho module “Quản lý nhóm của quản lý người dùng” và sơ đồ tuần tự

#### - Add Group

1. Người quản lý người dùng đăng nhập thành công.
2. Lớp ManageUsersFrm gửi yêu cầu đóng gói đến lớp User.
3. Lớp User đóng gói đối tượng và trả về cho lớp ManageUsersFrm đối tượng User.
4. Lớp ManageUsersFrm hiển thị tất cả người dùng.
5. Người quản lý người dùng chọn tài khoản người dùng.
6. Lớp ManageUsersFrm thêm người dùng vào danh sách.
7. Lớp ManageUsersFrm trả về danh sách đã chọn hiển thị cho người quản lý người dùng.
8. Người quản lý người dùng click “Add Group”
9. Lớp NewGroupFrm gửi yêu cầu đóng gói đến lớp User.
10. Lớp User đóng gói đối tượng và trả đối tượng về cho lớp NewGroupFrm.
11. Lớp NewGroupFrm hiển thị form nhập thông tin nhóm.
12. Người quản lý người dùng nhập thông tin của nhóm và click “Save”
13. Lớp NewGroupFrm gửi yêu cầu đóng gói đến lớp Conversation.
14. Lớp Conversation đóng gói đối tượng và trả đối tượng về cho lớp NewGroupFrm.
15. Lớp NewGroupFrm gửi yêu cầu đóng gói đến lớp AttendingConversation.
16. Lớp AttendingConversation đóng gói đối tượng và trả đối tượng về cho lớp NewGroupFrm.
17. Lớp NewGroupFrm gửi đối tượng AttendingConversation và Conversation đến lớp ManageGroupOfAccountManagerCtr.
18. Lớp ManageGroupOfAccountManagerCtr lưu vào CSDL.

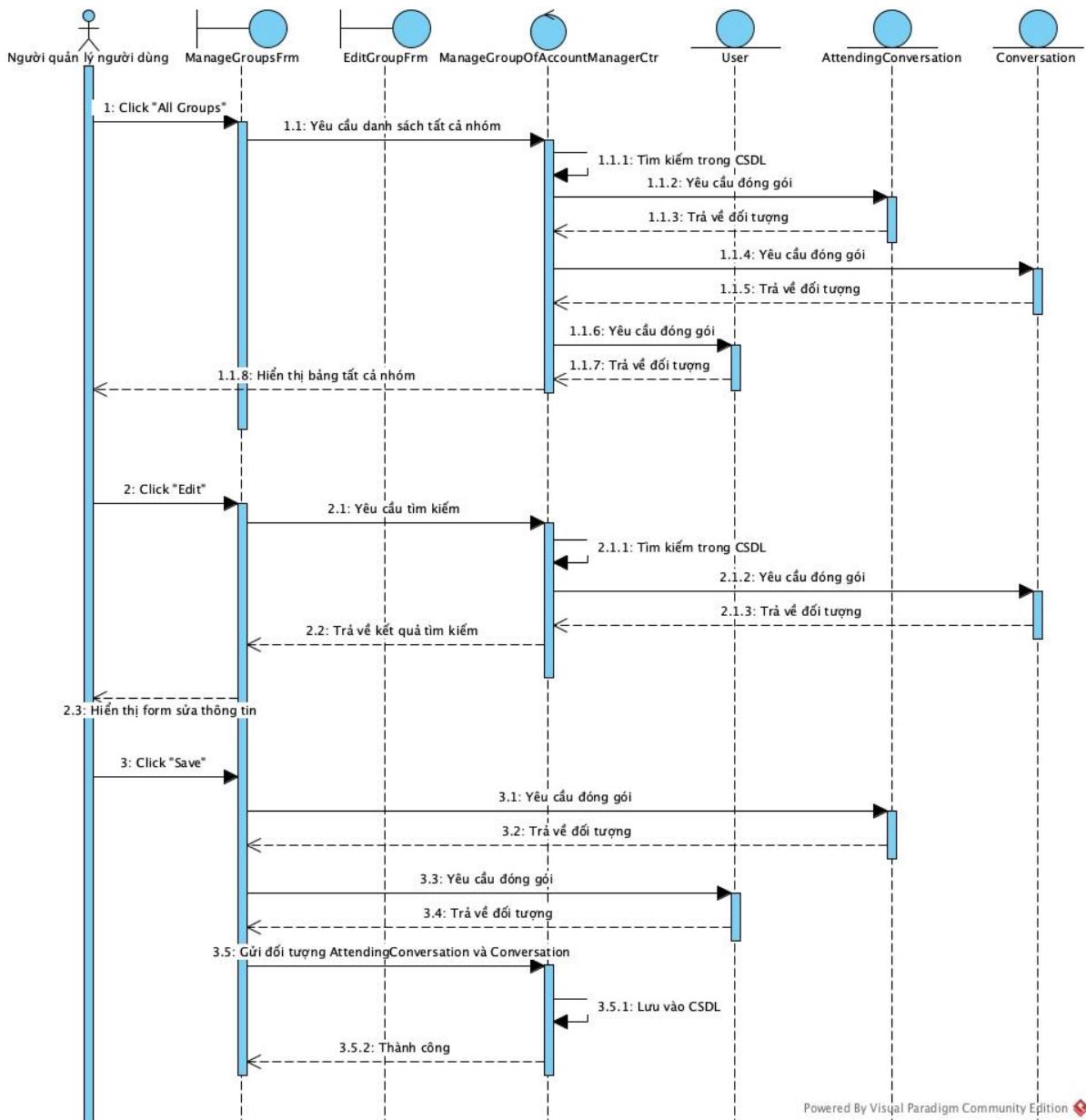
19. Lớp ManageGroupOfAccountManagerCtr thông báo tạo group mới thành công.



### - Edit Group

1. Người quản lý người dùng click “All Groups”
2. Lớp ManageGroupsFrm gửi yêu cầu danh sách tất cả các nhóm đến lớp ManageGroupOfAccountManagerCtr.
3. Lớp ManageGroupOfAccountManagerCtr tìm kiếm trong CSDL.
4. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp AttendingConversation.
5. Lớp AttendingConversation trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
6. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp Conversation.
7. Lớp Conversation trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
8. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp User.
9. Lớp User trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
10. Lớp ManageGroupOfAccountManagerCtr trả về hiển thị bảng tất cả các nhóm.
11. Người quản lý người dùng click “Edit”

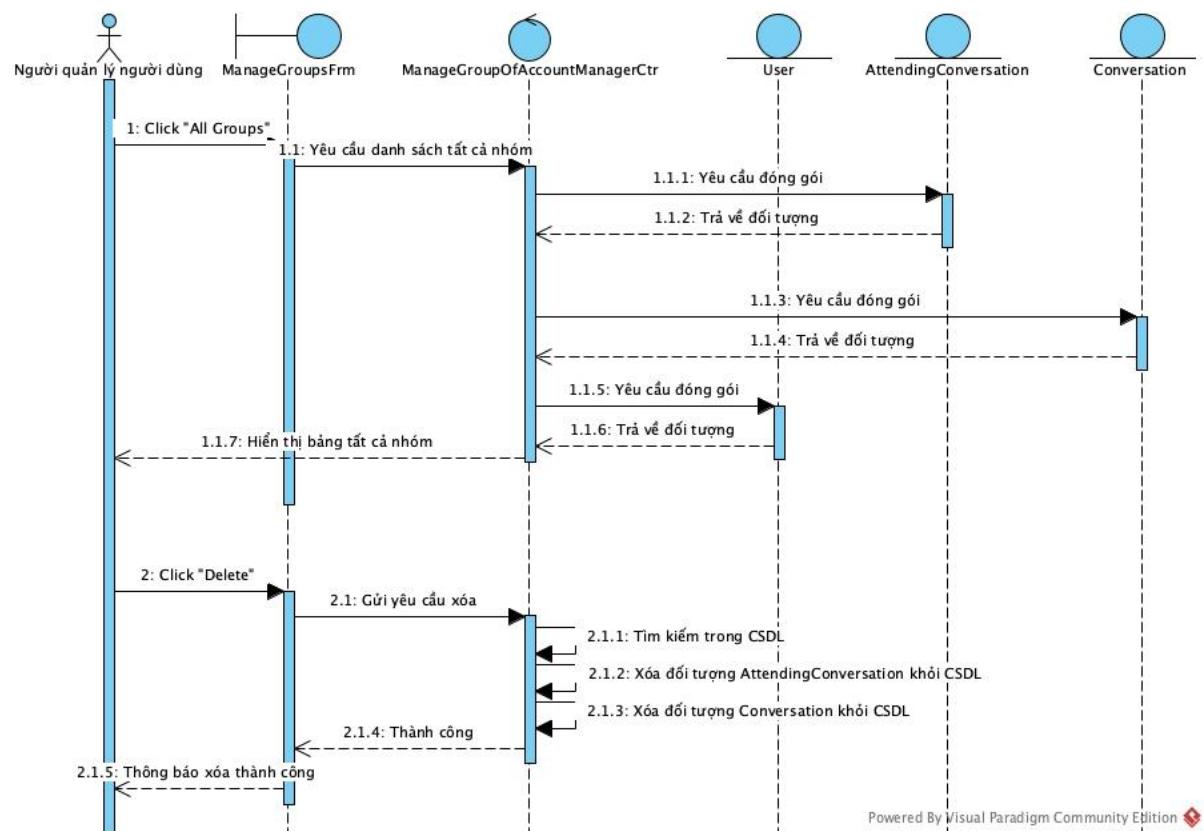
12. Lớp ManageGroupsFrm gửi yêu cầu tìm kiếm đến lớp ManageGroupOfAccountManagerCtr.
13. Lớp ManageGroupOfAccountManagerCtr tìm kiếm trong CSDL.
14. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói đến Conversation.
15. Lớp Conversation trả về đối tượng đã đóng gói cho lớp ManageGroupOfAccountManagerCtr.
16. Lớp ManageGroupOfAccountManagerCtr trả về kết quả tìm kiếm cho lớp ManageGroupsFrm.
17. Lớp ManageGroupsFrm hiển thị form sửa thông tin nhóm.
18. Người quản lý người dùng sửa thông tin và click “Save”
19. Lớp ManageGroupsFrm gửi yêu cầu đóng gói đến AttendingCoversation.
20. Lớp AttendingCoversation trả về đối tượng đã đóng gói cho lớp ManageGroupsFrm.
21. Lớp ManageGroupsFrm gửi yêu cầu đóng gói đến User.
22. Lớp User trả về đối tượng đã đóng gói cho lớp ManageGroupsFrm.
23. Lớp ManageGroupsFrm gửi đối tượng AttendingConversation và Conversation đến lớp ManageGroupOfAccountManagerCtr.
24. Lớp ManageGroupOfAccountManagerCtr lưu thay đổi vào CSDL.
25. Lớp ManageGroupOfAccountManagerCtr thông báo thay đổi thành công.



### - Delete Group

1. Người quản lý người dùng click “All Groups”
2. Lớp ManageGroupsFrm gửi yêu cầu danh sách tất cả các nhóm đến lớp ManageGroupOfAccountManagerCtr.
3. Lớp ManageGroupOfAccountManagerCtr tìm kiếm trong CSDL.
4. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp AttendingConversation.
5. Lớp AttendingConversation trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
6. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp Conversation.
7. Lớp Conversation trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
8. Lớp ManageGroupOfAccountManagerCtr gửi yêu cầu đóng gói dữ liệu đến lớp User.

9. Lớp User trả về đối tượng đã được đóng gói cho lớp ManageGroupOfAccountManagerCtr.
10. Lớp ManageGroupOfAccountManagerCtr trả về hiển thị bảng tất cả các nhóm.
11. Người quản lý người dùng click “Delete”
12. Lớp ManageGroupsFrm gửi yêu cầu xóa đến lớp ManageGroupOfAccountManagerCtr.
13. Lớp ManageGroupOfAccountManagerCtr tìm kiếm trong CSDL.
14. Lớp ManageGroupOfAccountManagerCtr xóa đối tượng AttendingConversation khỏi CSDL.
15. Lớp ManageGroupOfAccountManagerCtr xóa đối tượng Conversation khỏi CSDL.
16. Lớp ManageGroupOfAccountManagerCtr thông báo xóa thành công.



### 3.3. Tài liệu pha thiết kế

#### 3.3.1. Thiết kế CSDL

\* Các bảng CSDL:

- tblUser
- tblPosition
- tblDepartment
- tblJobTransfer
- tblAttendingConversation
- tblConversation
- tblMessage

\* Các thành phần trong các bảng

- tblUser(**idUser**, fullName, gender, phoneNum, userEmail, companyEmail, idJobTransfer, password, role, state, avatar)

- tblPosition(**idPosition**, namePosition, department)

- tblDepartment(**idDepartment**, nameDepartment, describe)

- tblJobTransfer(**idJobTransfer**, idNewPosition, idUser, dateTransfer)

- tblAttendingConversation(**idUser**, **idConversation**, conversationRole, state)

- tblConversation(**idConversation**, nameConversation)

- tblMessage(**idMessage**, idConversation, idUserSend, timeSend, isText, isImage, isVoice, contentMessage)

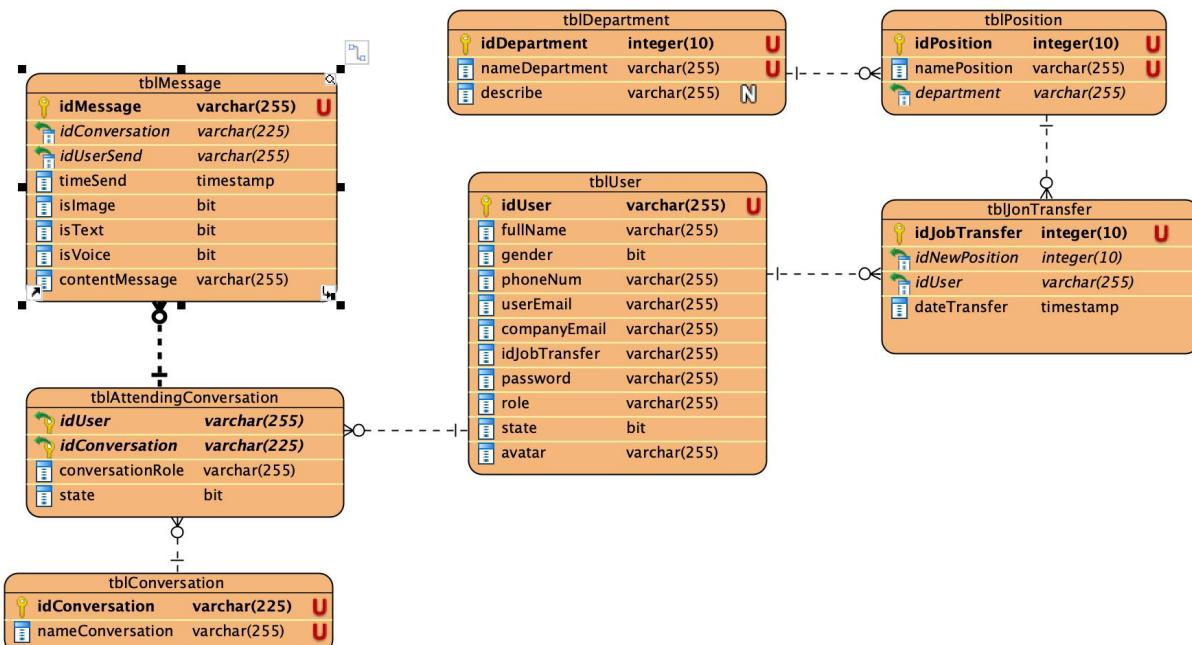
\* Các quan hệ:

- User quan hệ nhiều - nhiều với Conversation nên tạo ra bảng AttendingConversation để chuyển thành 2 quan hệ 1 - nhiều.

- AttendingConversation quan hệ 1 - nhiều với Message (1 người tham gia một cuộc trò chuyện có thể gửi nhiều tin nhắn).

- Position quan hệ nhiều - 1 với Department.

- Position quan hệ nhiều - nhiều với User nên tạo ra bảng JobTransfer để chuyển thành 2 quan hệ 1 - nhiều.



Hình 3.3.1. CSDL của hệ thống

### 3.3.2. Thiết kế mô hình MVC

#### a) Module “Quản lý người dùng”

Người quản lý người dùng trong module “Quản lý người dùng” có thể thực hiện được các tính năng:

- Thêm người dùng

- Xóa người dùng

- Sửa thông tin người dùng

Do đó trong lớp điều khiển ManageUsersDAO phải có các phương thức để thực hiện các hành động tương tác với CSDL tương ứng:

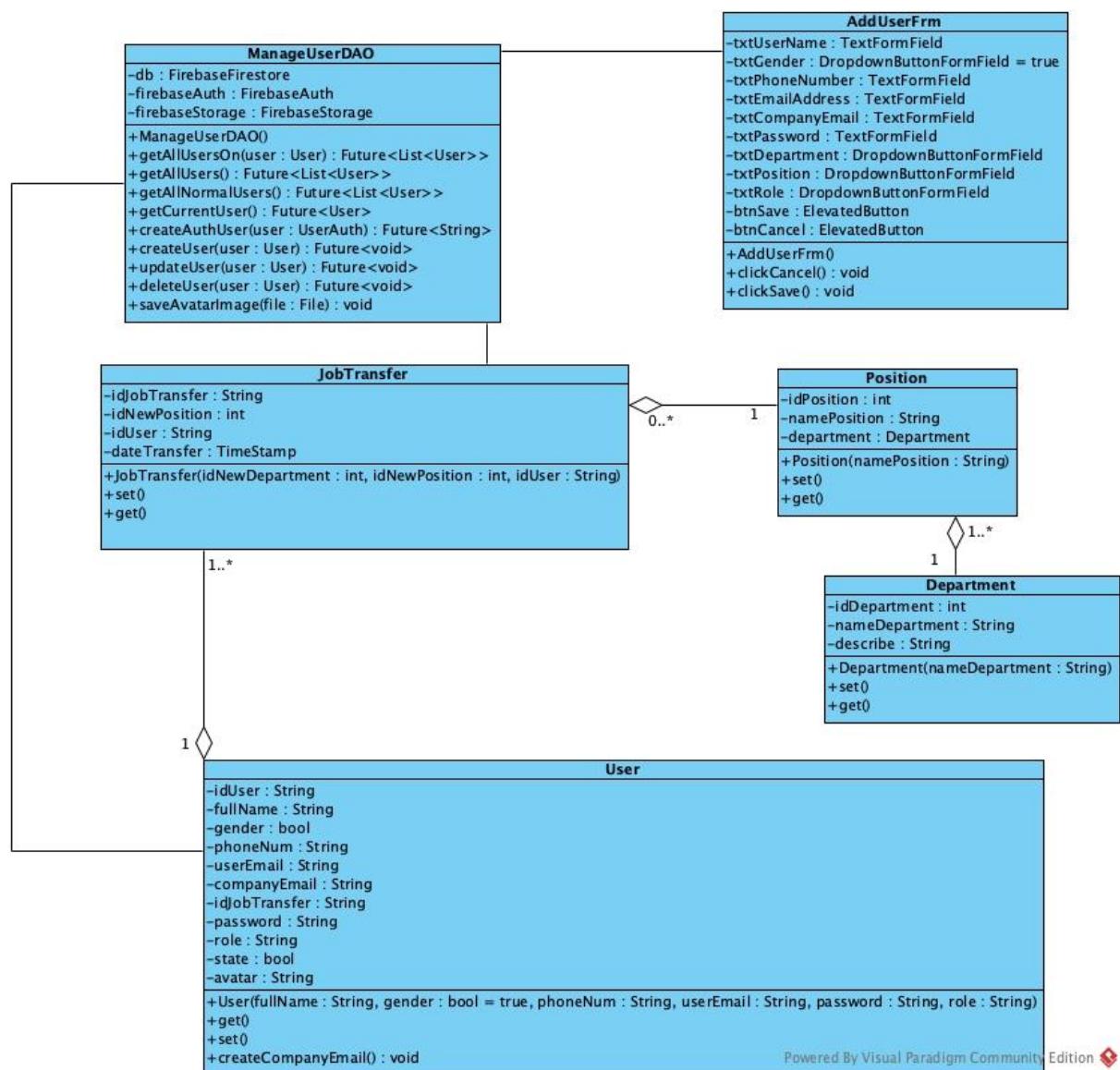
- createUser()
  - + Trong createUser() cần sử dụng thêm hàm addJobTransfer()
- deleteUser()
- updateUser()

#### \*Add User

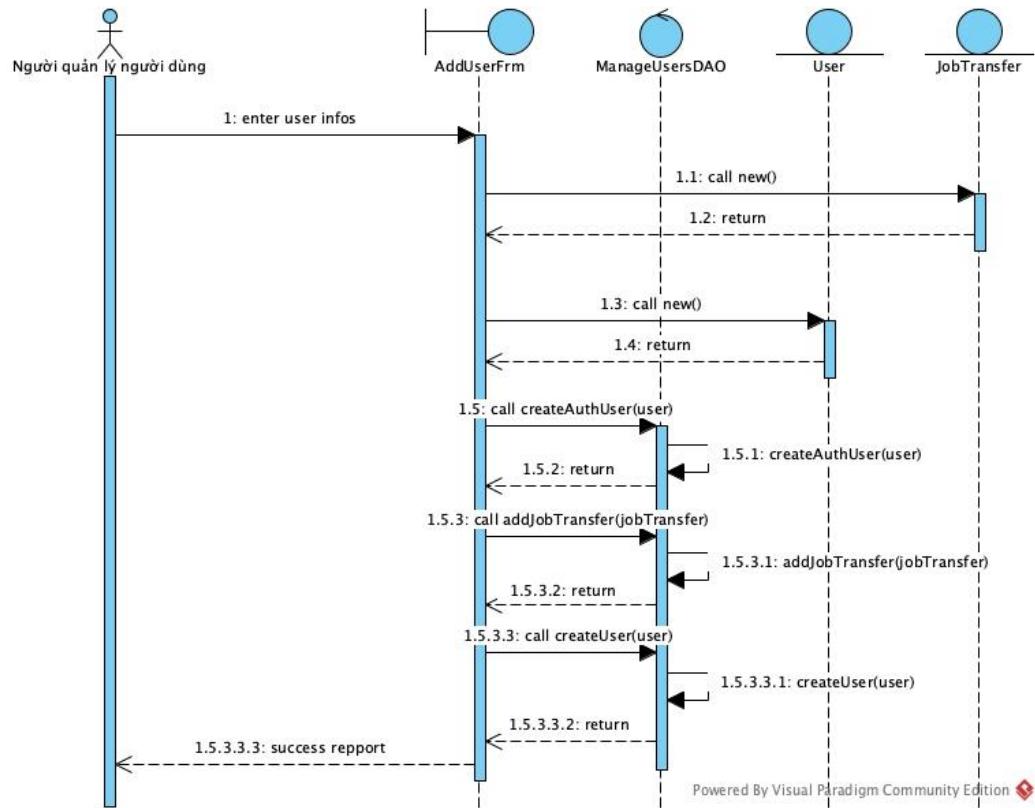
- AddUserFrm cần có các trường để nhập các thuộc tính liên quan đến User: name, gender, phone number, email address, company email, password, department, position, role và có 2 button save và cancel. Trong đó company email và password sẽ ở chế độ chỉ đọc.

- Do có 2 button save và cancel nên phải có 2 hàm để xử lý 2 button khi được click là clickSave() và clickCancel()

#### - Hoàn thiện sơ đồ lớp



## - Sơ đồ tuần tự

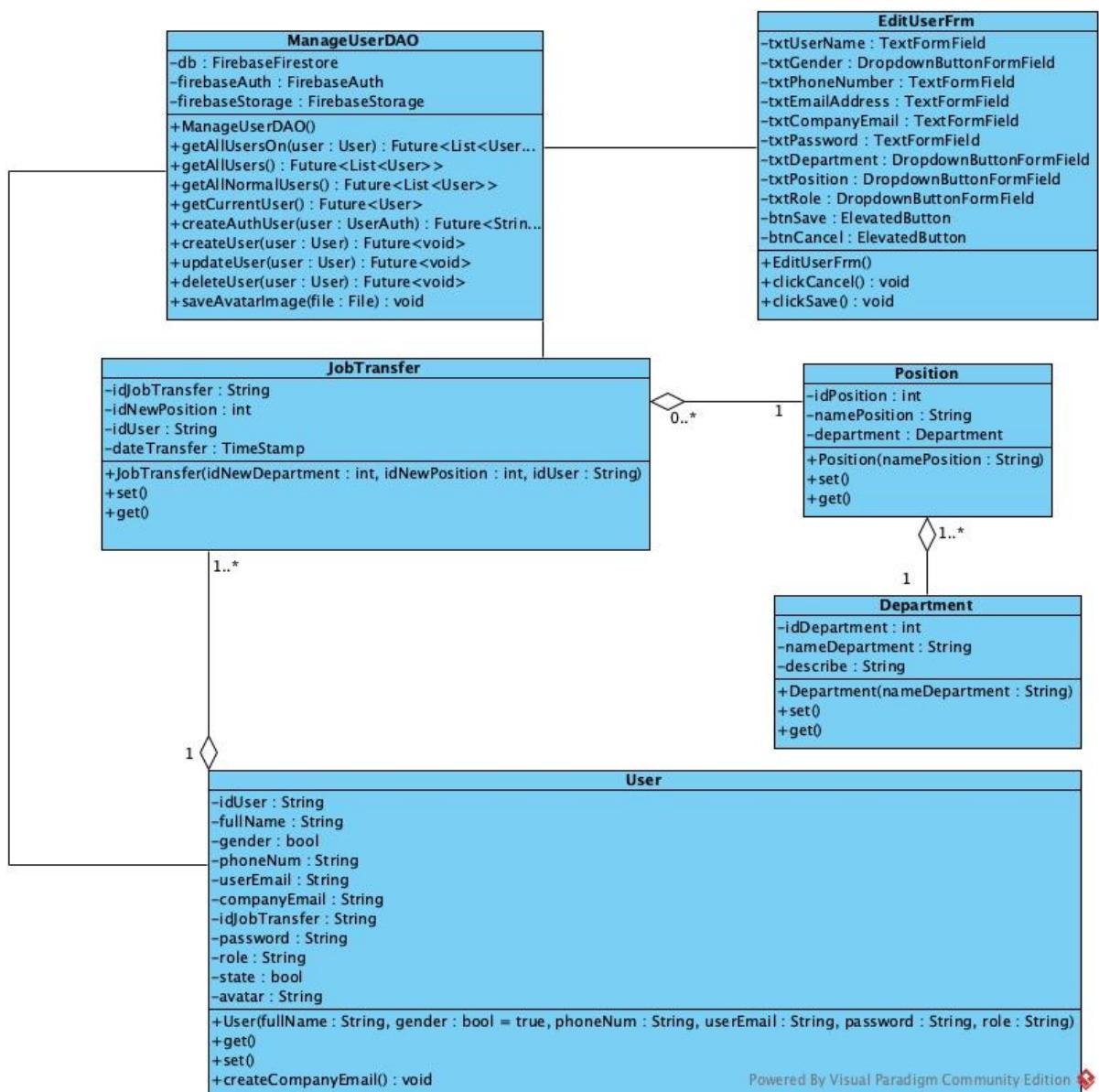


### \* Edit User

- EditUserFrm cần có các trường để sửa các thuộc tính liên quan đến User: name, gender, phone number, email address, company email, password, department, position, role và có 2 button save và cancel. Trong đó company email và password sẽ ở chế độ chỉ đọc. Các trường đều có giá trị cũ đã điền vào các trường.

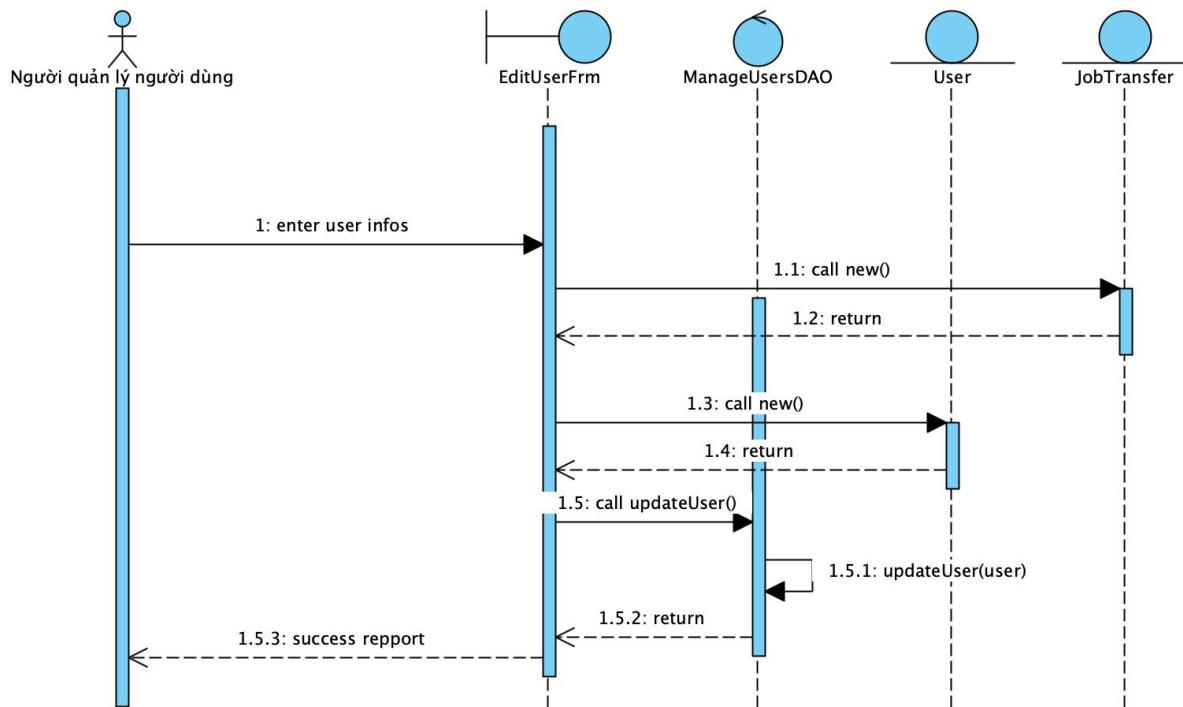
- Do có 2 button save và cancel nên phải có 2 hàm để xử lý 2 button khi được click là clickSave() và clickCancel()

## - Hoàn thiện sơ đồ lớp



Powered By Visual Paradigm Community Edition

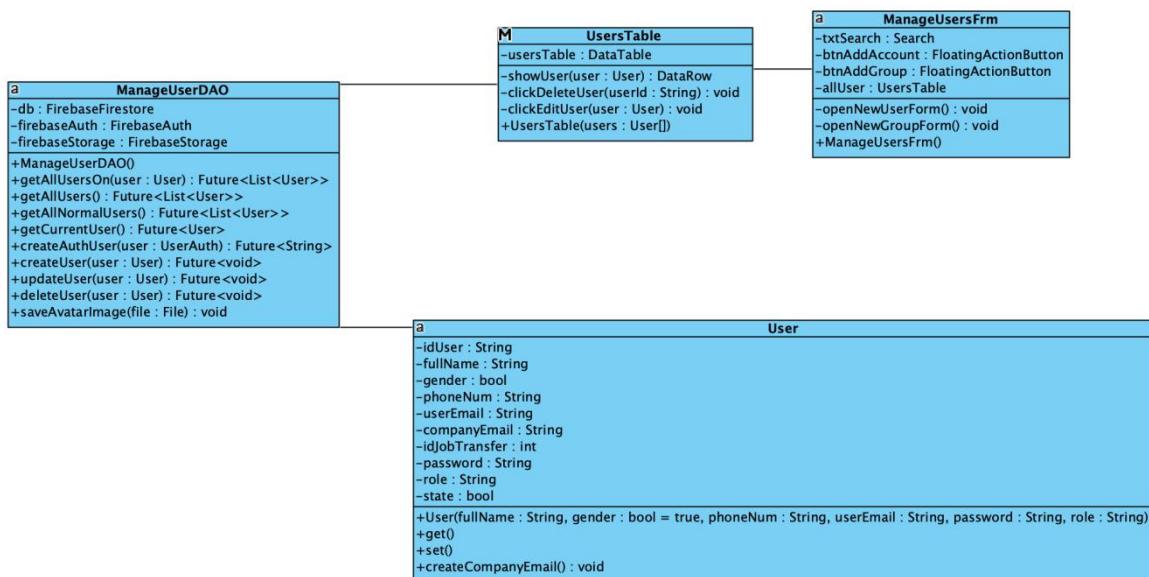
## - Sơ đồ tuần tự



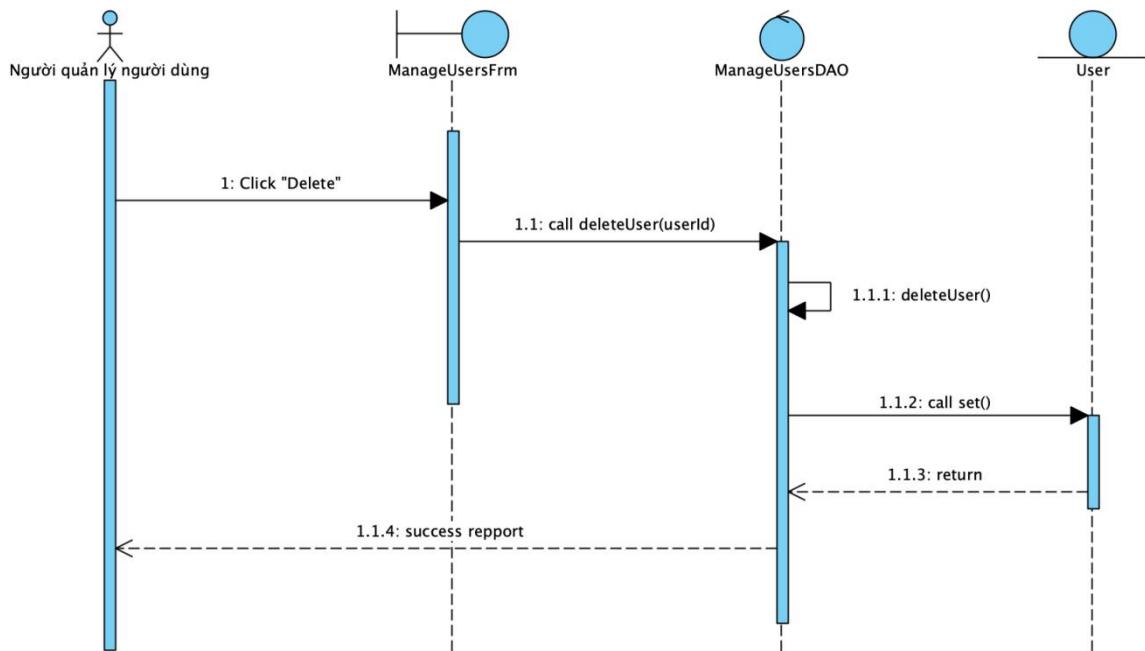
### \* Delete User

- Tính năng xóa người dùng sẽ thực hiện khi click nút Delete tương ứng với người dùng hiển thị trong UsersTable của ManageUsersFrm.
- Do đó trong UsersTable sẽ có hàm xử lý khi click và nút delete: clickDelete()

### - Hoàn thiện sơ đồ lớp



## - Sơ đồ tuần tự



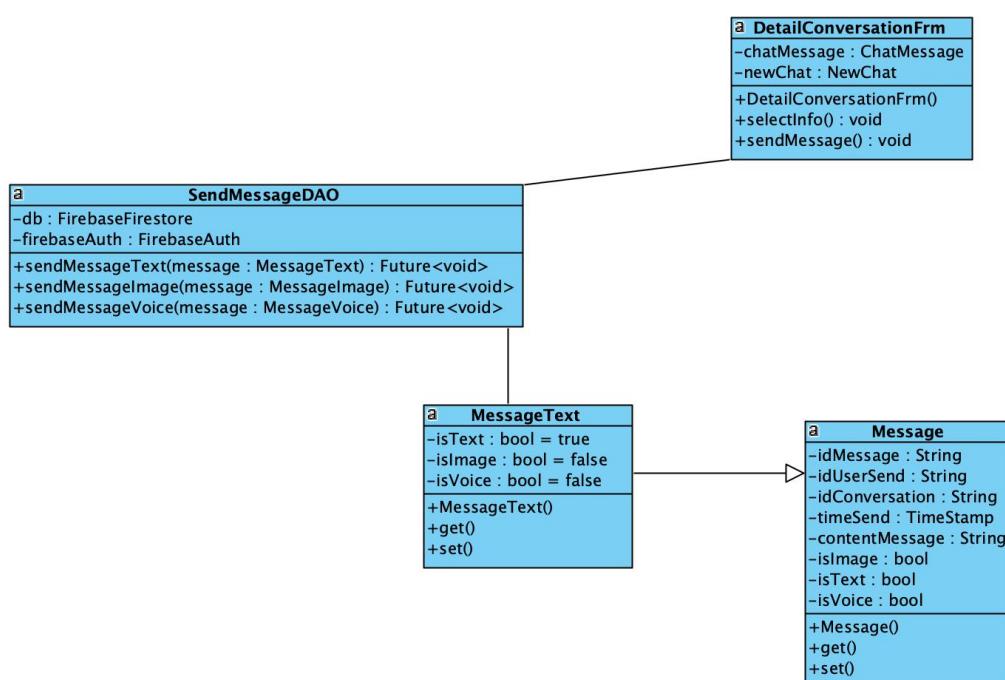
## b) Module “Nhắn tin”

Gửi tin nhắn bao gồm gửi tin nhắn văn bản, voice, ảnh nên tương ứng sẽ có 3 hàm sendMessageText, sendMessageVoice, sendMessageImage trong lớp SendMessageDAO

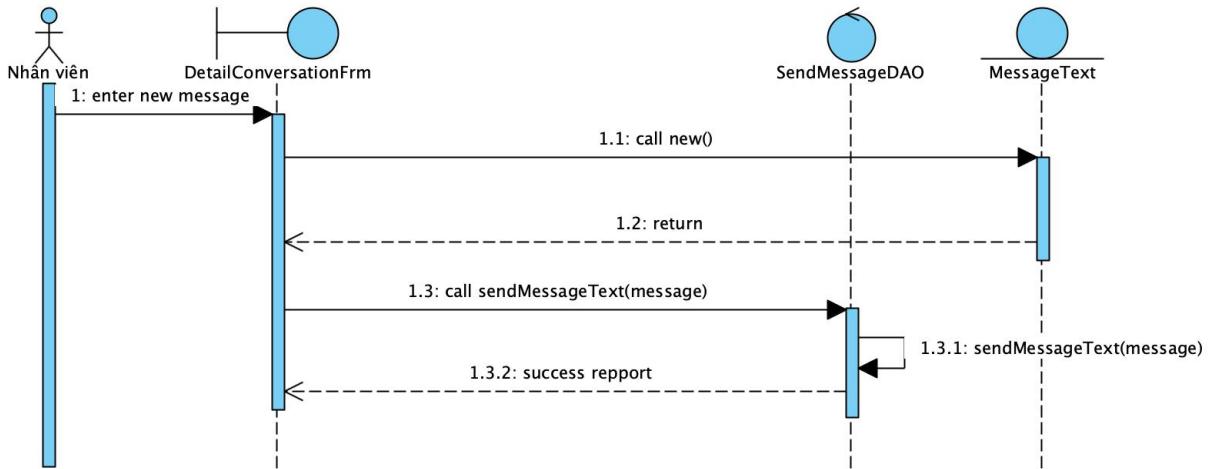
Lớp biên DetailConversationFrm được chia nhỏ ra thành 2 thành phần là hiện thị tin nhắn đã gửi và phần nhập tin nhắn mới để gửi là chatMessage và newChat.

- \* Gửi tin nhắn văn bản

### - Hoàn thiện sơ đồ lớp



## - Sơ đồ tuần tự



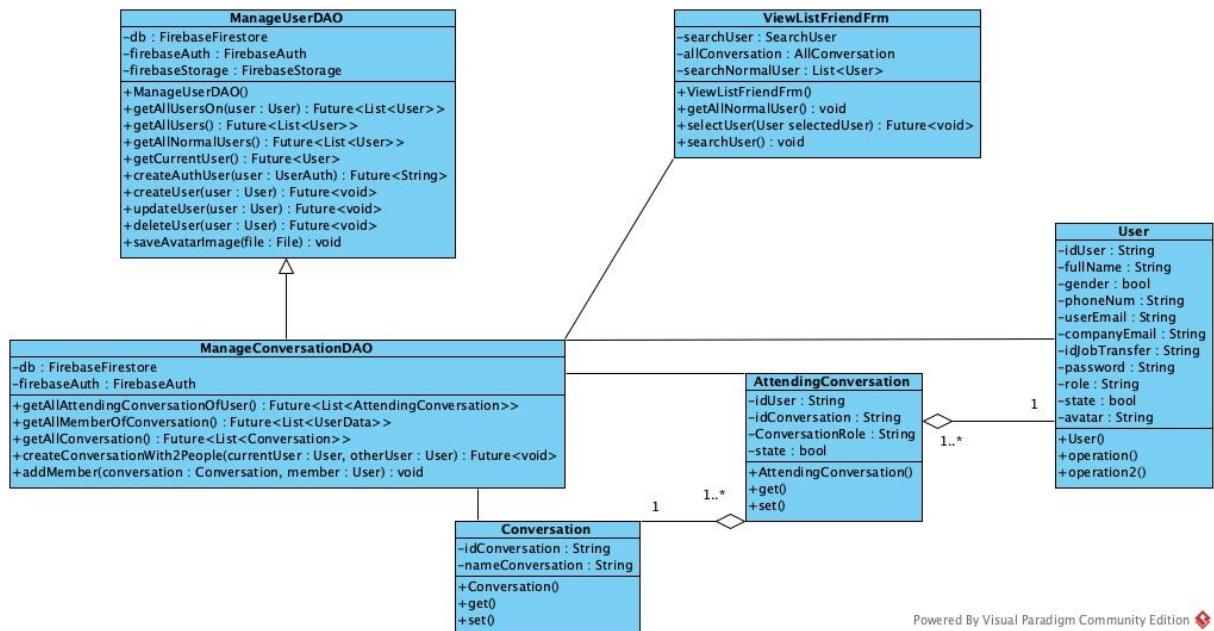
### c) Thêm bạn

Thêm bạn gồm các công việc search tài khoản theo tên, chọn 1 bạn cần thêm mới rồi hiển thị conversation mới. Do đó cần các hàm getAllNormalUser() để lấy dữ liệu của tất cả các tài khoản thường, hàm searchUser() để search ra các kết quả và hàm selectUser() là hàm thực hiện khi chọn 1 tài khoản trong danh sách kết quả.

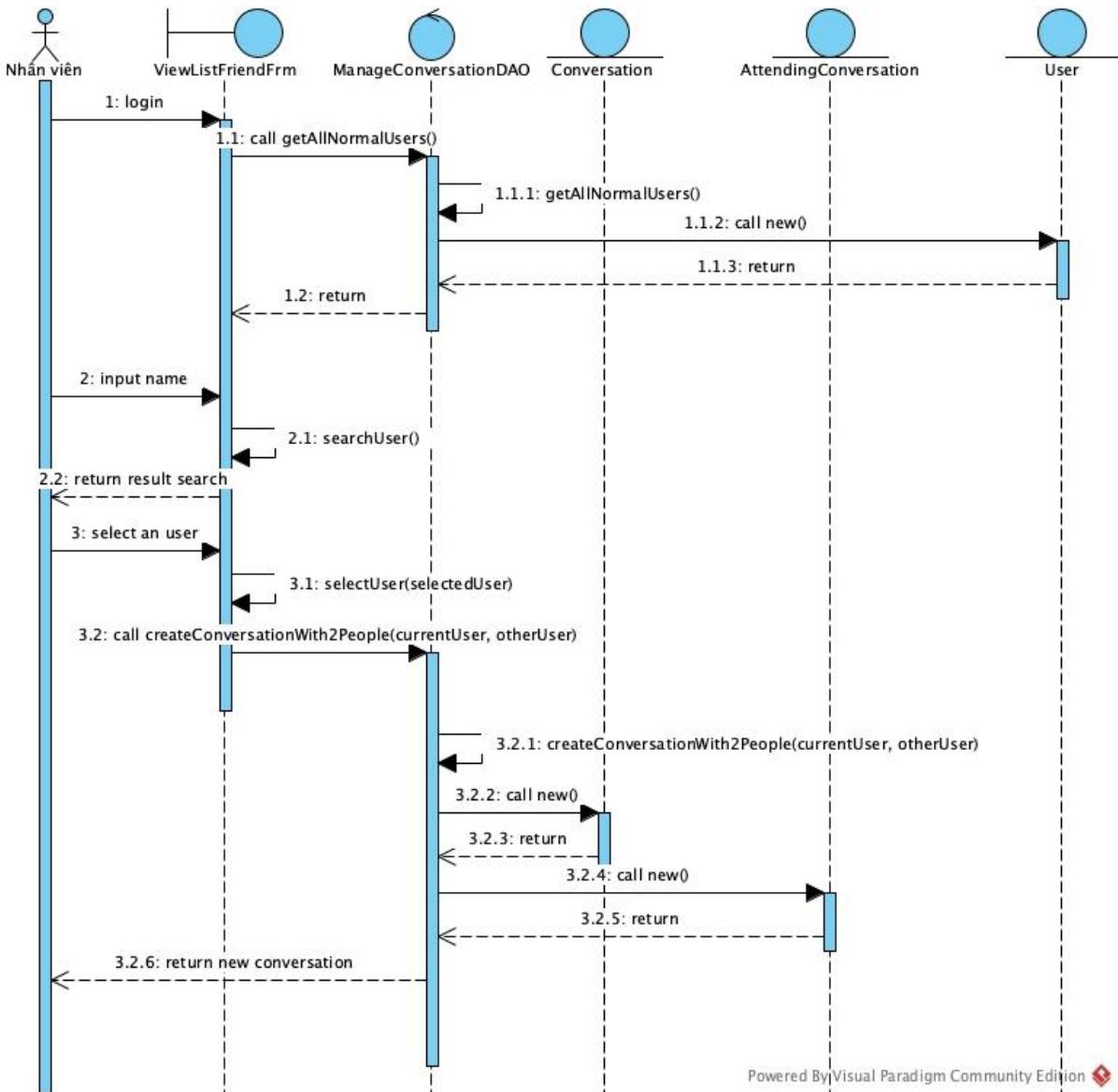
Để lấy được getAllNormalUser() thì cần hàm getAllNormalUsers() trong ManageConversationDAO để tương tác với CSDL.

Để tạo được cuộc trò chuyện mới xử lý trong hàm selectUser() thì trong lớp ManageConversationDAO cần có hàm createConversationWith2People()

## - Hoàn thiện sơ đồ lớp



## - Sơ đồ tuần tự



### 3.4. Pha cài đặt

#### 3.4.1. Các công cụ và thư viện đã sử dụng trong dự án

##### a) Các công cụ sử dụng

Các công cụ được sử dụng trong đề tài này sẽ được trình bày cụ thể trong bảng dưới đây:

Công cụ sử dụng	Chức năng	Version
Flutter	Framework	3.19.0
Dart	Ngôn ngữ lập trình	3.3.0
ChatGPT	Fix bug, tra cứu các widget	3.5
Stack overflow	Fix bug	-
Google translate	Dịch tài liệu	-
IntelliJ IDEA CE	Môi trường lập trình	2023.3.2

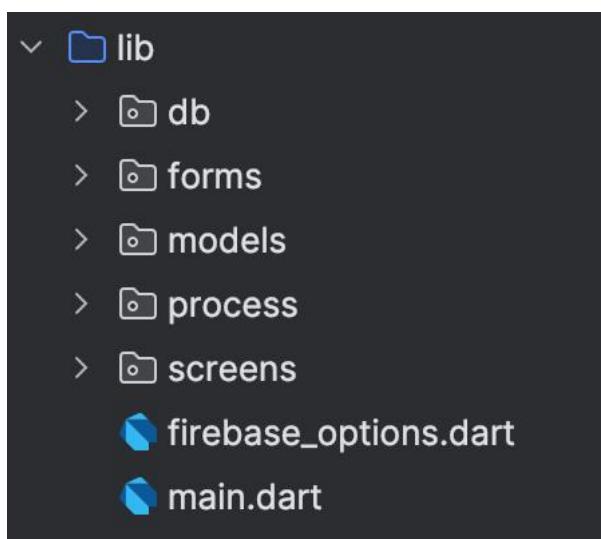
Android Studio	Chạy thiết bị ảo Android	2023.1.1 Patch 1
XCode	Chạy thiết bị ảo iPhone 14 Pro - iOS 16.2	14.2
GitHub	Quản lý source code	-
Visual Paradigm CE	Vẽ các thiết kế	17.1
Firebase	Cơ sở dữ liệu NoSQL được lưu trữ đám mây cho phép lưu trữ và đồng bộ dữ liệu	-

b) Các thư viện đã sử dụng trong dự án

Thư viện sử dụng	Chức năng	Version
firebase_auth	Dùng để xác thực tài khoản (login, logout, token)	^4.19.4
firebase_core	Cho phép kết nối nhiều ứng dụng Firebase	^2.30.1
firebase_storage	Lưu trữ ảnh trên firebase	^11.7.4
cloud_firestore	Lưu trữ CSDL trên firebase	^4.17.2
uuid	Tạo mã định danh giả	^4.4.0
image_picker	Chọn ảnh trong thư viện của thiết bị, truy cập máy ảnh để chụp ảnh	^1.1.1

### 3.4.2. Cây thư mục

Code chính nằm trong folder lib:

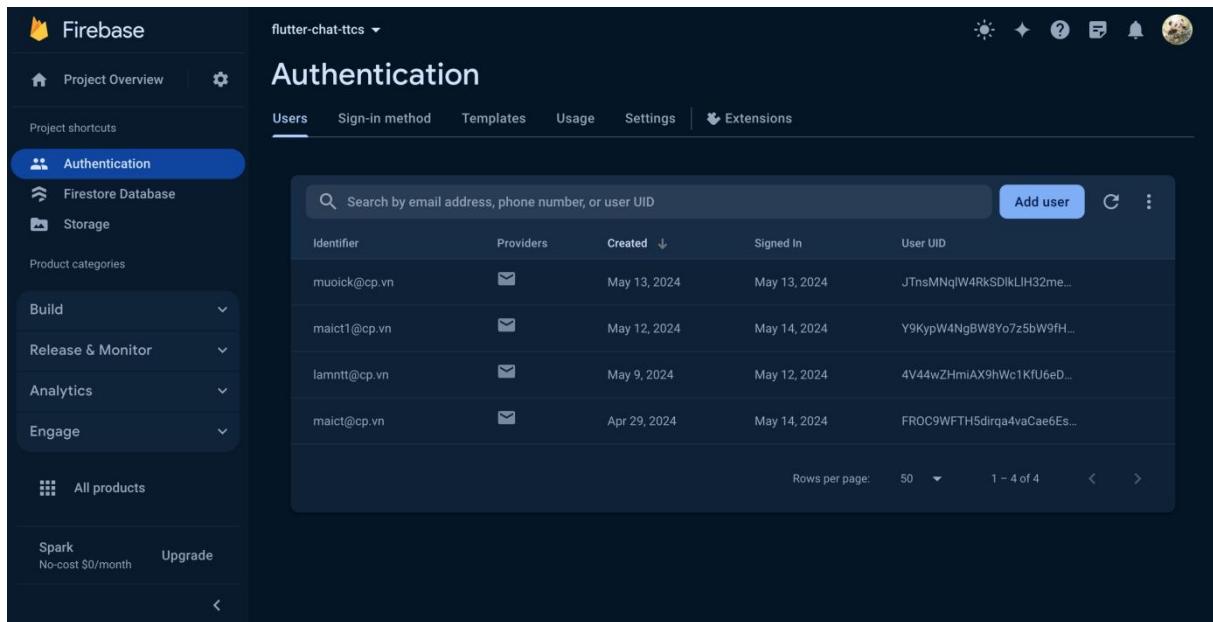


- db: chứa các class xử lý tương tác với CSDL.
- forms: chứa các class trả về widget thành phần tạo nên screen.
- models: chứa các class đóng gói các đối tượng.
- process: chứa các class để xử lý tên, email, tạo mật khẩu ngẫu nhiên.
- screens: chứa các màn hình chính.
- file firebase\_options.dart: chứa tham số để liên kết firebase dùng cho từng nền tảng.
- file main.dart: chạy dự án.

### 3.4.3. Cơ sở dữ liệu trên Firebase

#### a) Firebase Authentication

Firebase Authentication dùng để xác thực tài khoản (login, logout, token). Ứng dụng sẽ gửi tin hiệu xác thực đến Firebase Authentication. Firebase Authentication sẽ trả lại response.



The screenshot shows the Firebase console's Authentication section. On the left sidebar, 'Authentication' is selected under 'Project shortcuts'. The main area displays a table of users with columns: Identifier, Providers, Created, Signed In, and User UID. The table contains four entries:

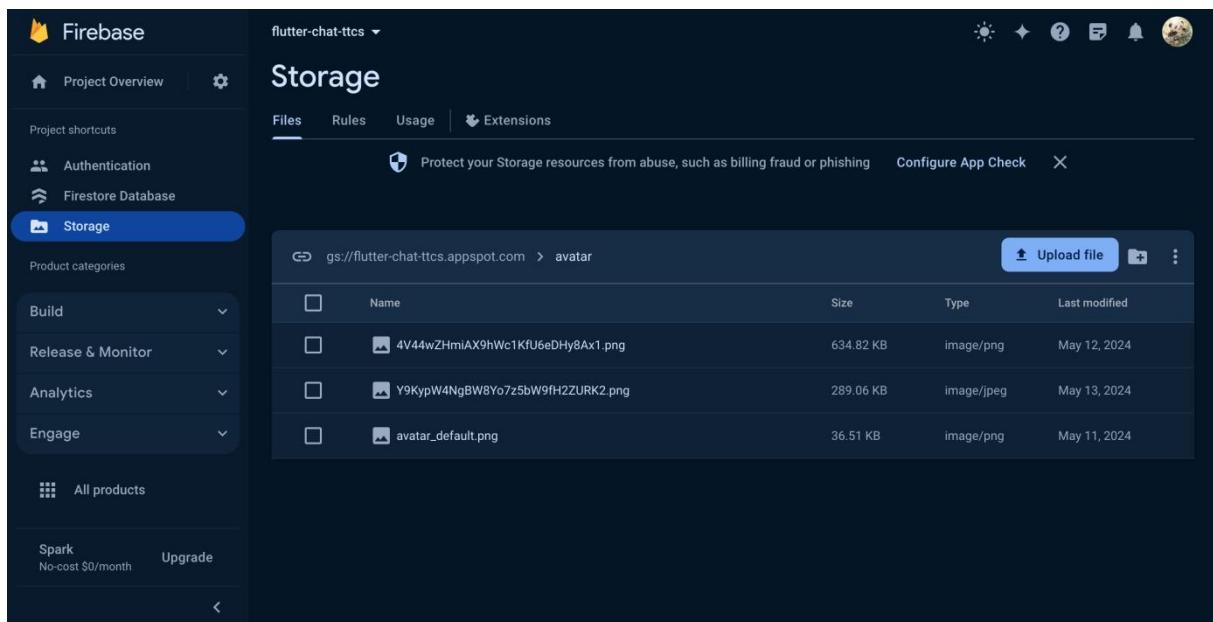
Identifier	Providers	Created	Signed In	User UID
muoick@cp.vn	✉️	May 13, 2024	May 13, 2024	JTnsMNqjW4RkSDIkLH32me...
maiict1@cp.vn	✉️	May 12, 2024	May 14, 2024	Y9KypW4NgBW8Yo7z5bW9fH...
lamntt@cp.vn	✉️	May 9, 2024	May 12, 2024	4V44wZHmiAX9hWc1KfU6eD...
maiict@cp.vn	✉️	Apr 29, 2024	May 14, 2024	FROC9WFTH5dirqa4vaCae6Es...

At the bottom, there are buttons for 'Rows per page' (50), '1 - 4 of 4', and navigation arrows.

Hình 3.4.3a: Firebase Authentication

#### b) Firebase Storage

Firebase Storage sử dụng để lưu trữ avatar của người dùng. Trong ảnh dưới đây avatar của người dùng được lưu trong folder “avatar”.



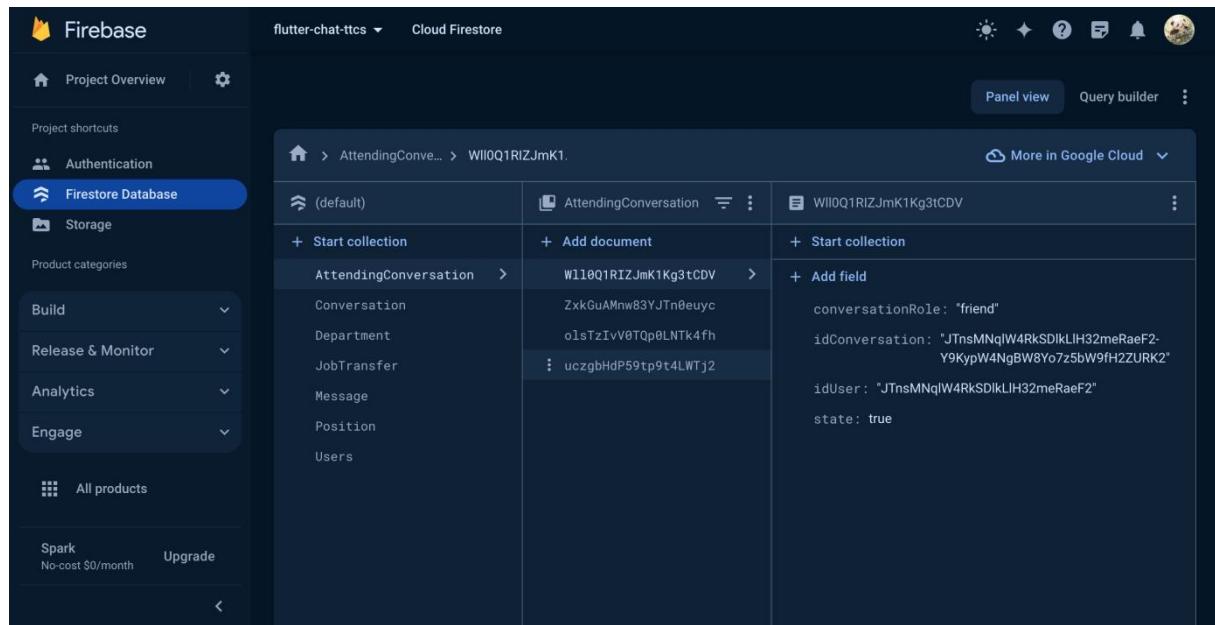
The screenshot shows the Firebase console's Storage section. On the left sidebar, 'Storage' is selected under 'Project shortcuts'. The main area displays a table of files in the 'gs://flutter-chat-ttcs.appspot.com/avatar' folder. The table has columns: Name, Size, Type, and Last modified. The table contains three entries:

Name	Size	Type	Last modified
4V44wZHmiAX9hWc1KfU6eDHy8Ax1.png	634.82 KB	image/png	May 12, 2024
Y9KypW4NgBW8Yo7z5bW9fH2ZURK2.png	289.06 KB	image/jpeg	May 13, 2024
avatar_default.png	36.51 KB	image/png	May 11, 2024

Hình 3.4.3b: Firebase Storage

### b) Firebase Firestore Database

Firebase Firestore Database dùng để lưu trữ dữ liệu thiết kế theo bảng CSDL trong pha thiết kế.



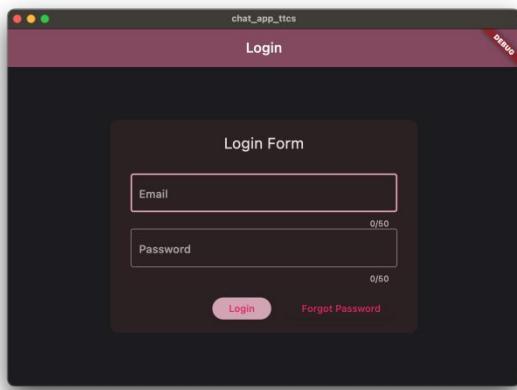
The screenshot shows the Firebase console interface for a project named "flutter-chat-ttcs". The left sidebar has "Firestore Database" selected. The main area displays a hierarchical database structure under "AttendingConversation" with a document ID "WII0Q1RIZJmK1Kg3tCDV". This document contains fields like "conversationRole: "friend"" and "idUser: "JTsMNqlW4RkSDlkLH32meRaeF2"". A "More in Google Cloud" button is visible at the top right of the main panel.

Hình 3.4.3c: Firebase Firestore Database

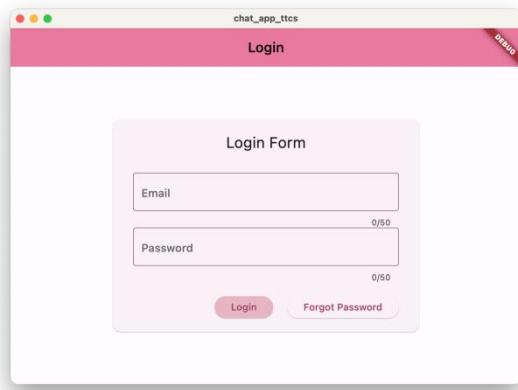
### 3.4.4. Cài đặt

#### a) Giao diện login

Ứng dụng có 2 theme là dark và light được thay đổi theo theme của hệ thống.



Hình 3.4.4a (1)  
Giao diện login theme dark



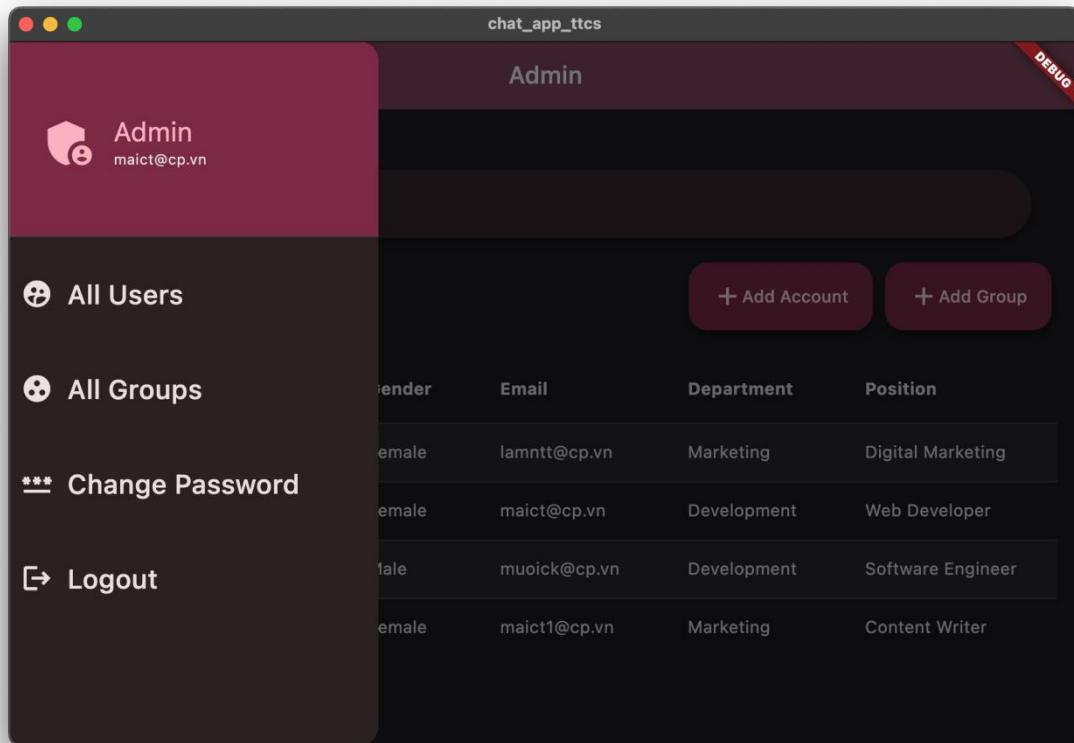
Hình 3.4.4a (2)  
Giao diện login theme light

#### b) Drawer của admin

Thanh drawer chứa:

- Thông tin về email của tài khoản admin

- Các nút chuyển trang: All User chuyển đến trang quản lý người dùng, All Groups chuyển đến trang quản lý các nhóm, Change Password chuyển đến trang đổi mật khẩu.
- Nút Logout dùng để đăng xuất khỏi tài khoản hiện tại.



Hình 3.4.4b:Thanh drawer của admin

### c) Trang quản lý tài khoản người dùng của admin

Trang quản lý tài khoản chứa thông tin chung của người dùng và các nút để quản lý. Thanh search dùng để search tài khoản để tạo ra cuộc hội thoại mới để admin nhắn tin thông báo với tài khoản khác.

The two screenshots show the "User Management" screen of the application. The top navigation bar is purple with the word "Admin" and a "DEBUG" indicator. A search bar labeled "Search" is at the top. Below it are two buttons: "+ Add Account" and "+ Add Group". The main content area displays a table of user data with columns: Name, Gender, Email, and Department. The data includes:

Name	Gender	Email	Department
Nguyen Thi Thanh Lam	Female	lamntt@cp.vn	Marketing
Chu Thi Mai	Female	maict@cp.vn	Development
Chu Kim Muoi	Male	muoick@cp.vn	Development
Chu Thi Mai	Female	maict1@cp.vn	Marketing

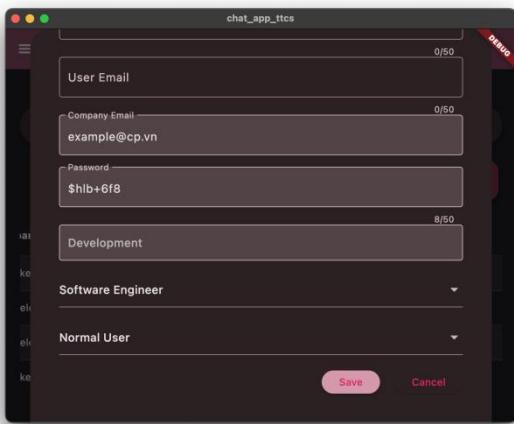
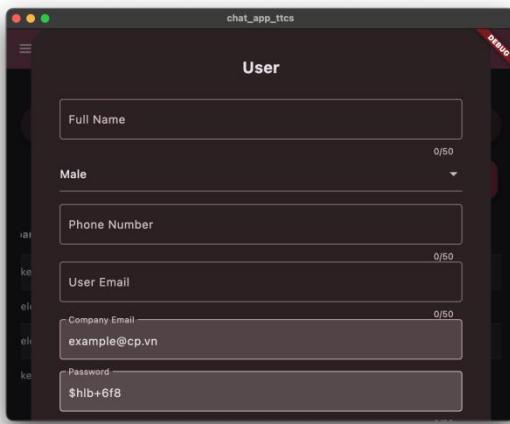
On the right side of each row are three buttons: "Edit", "Delete", and another unlabeled button. The second screenshot shows a similar table with different data:

Role	Position	Department
Normal User	Digital Marketing	Marketing
Admin	Web Developer	Development
Normal User	Software Engineer	Development
Normal User	Content Writer	Marketing

Hình 3.4.4c: Trang quản lý tài khoản người dùng

d) Trang thêm mới tài khoản của admin

Trang thêm tài khoản mới gồm các trường tên, giới tính, SĐT, email, password, phòng ban, vị trí và vai trò cùng với 2 nút save và cancel.

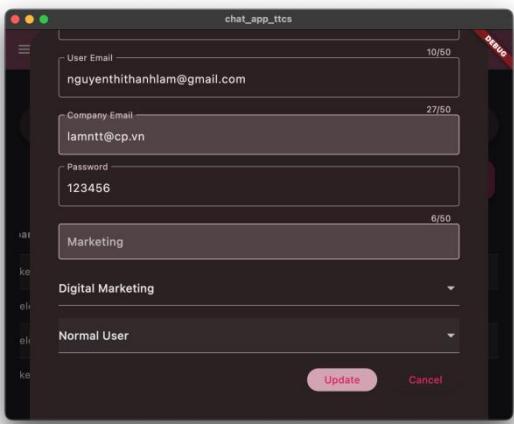
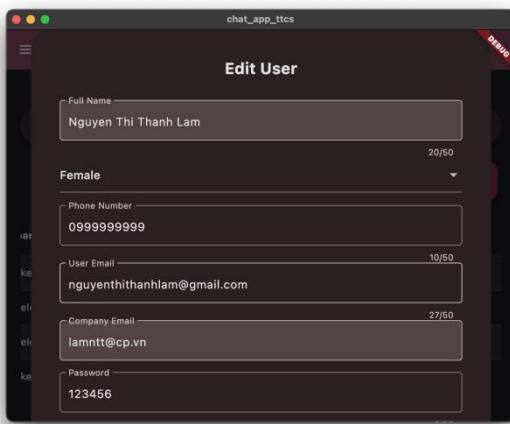


The screenshots show two views of a user creation form. The left view shows fields for Full Name, Gender (Male), Phone Number, User Email, Company Email, and Password. The right view shows fields for User Email, Company Email, Password, Department (Development), Position (Software Engineer), and Role (Normal User). Both screens have 'Save' and 'Cancel' buttons at the bottom.

Hình 3.4.4d: Trang thêm tài khoản mới

e) Trang sửa thông tin tài khoản của admin

Trang sửa thông tin của 1 tài khoản gồm các trường giống với trang thêm tài khoản mới tuy nhiên các trường sẽ được điền sẵn thông tin của tài khoản đã chọn.

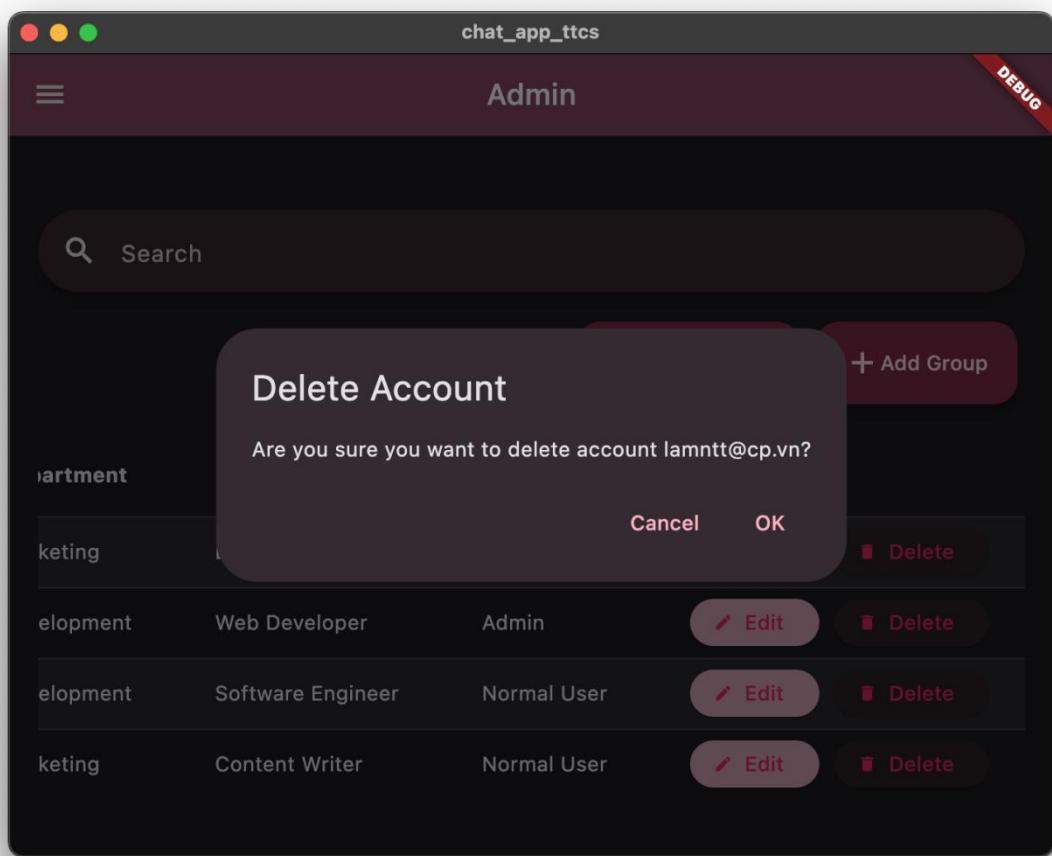


The screenshots show two views of an edit user form. The left view shows fields for Full Name, Gender (Female), Phone Number, User Email, Company Email, and Password. The right view shows fields for User Email, Company Email, Password, Department (Marketing), Position (Digital Marketing), and Role (Normal User). Both screens have 'Update' and 'Cancel' buttons at the bottom.

Hình 3.4.4e: Trang sửa tài khoản

f) Thông báo xác nhận xóa tài khoản của admin

Thông báo xác nhận xóa sẽ được hiện lên khi click delete.

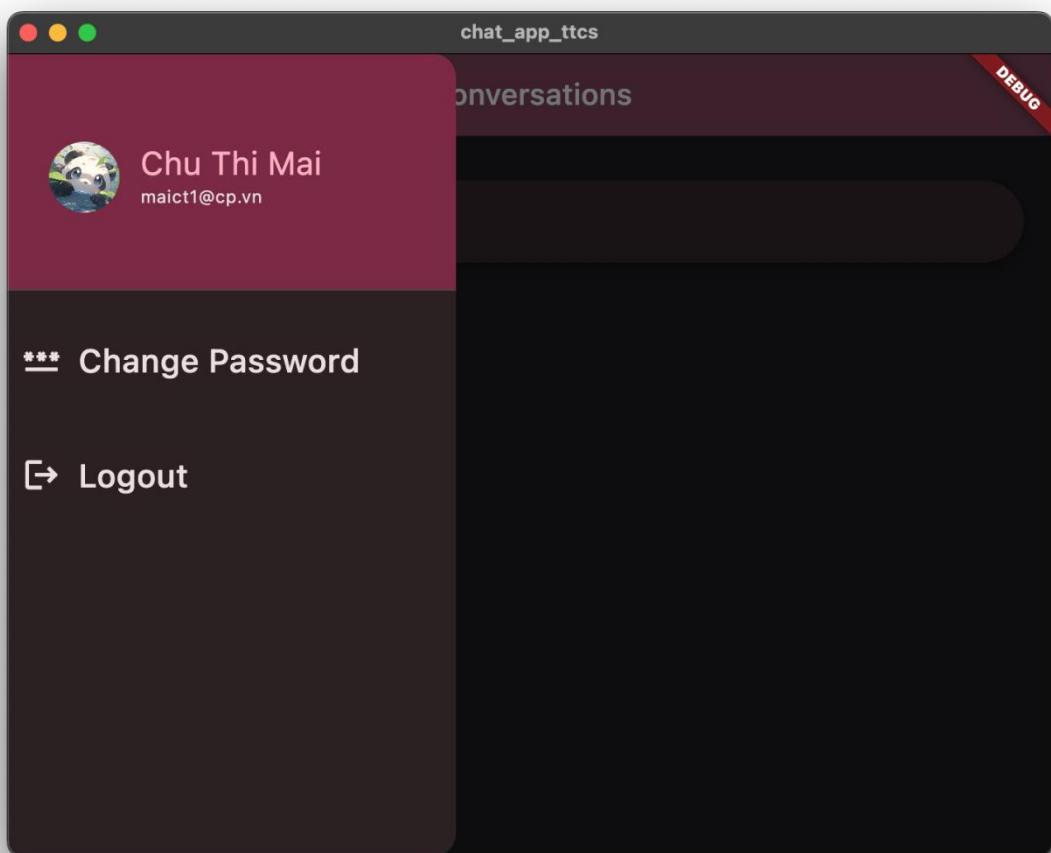


Hình 3.4.4f: Dialog thông báo xác nhận xóa tài khoản

*g) Drawer của tài khoản thường*

Drawer hiện thông tin tên và email của tài khoản, hiện thanh Change Password và Logout.

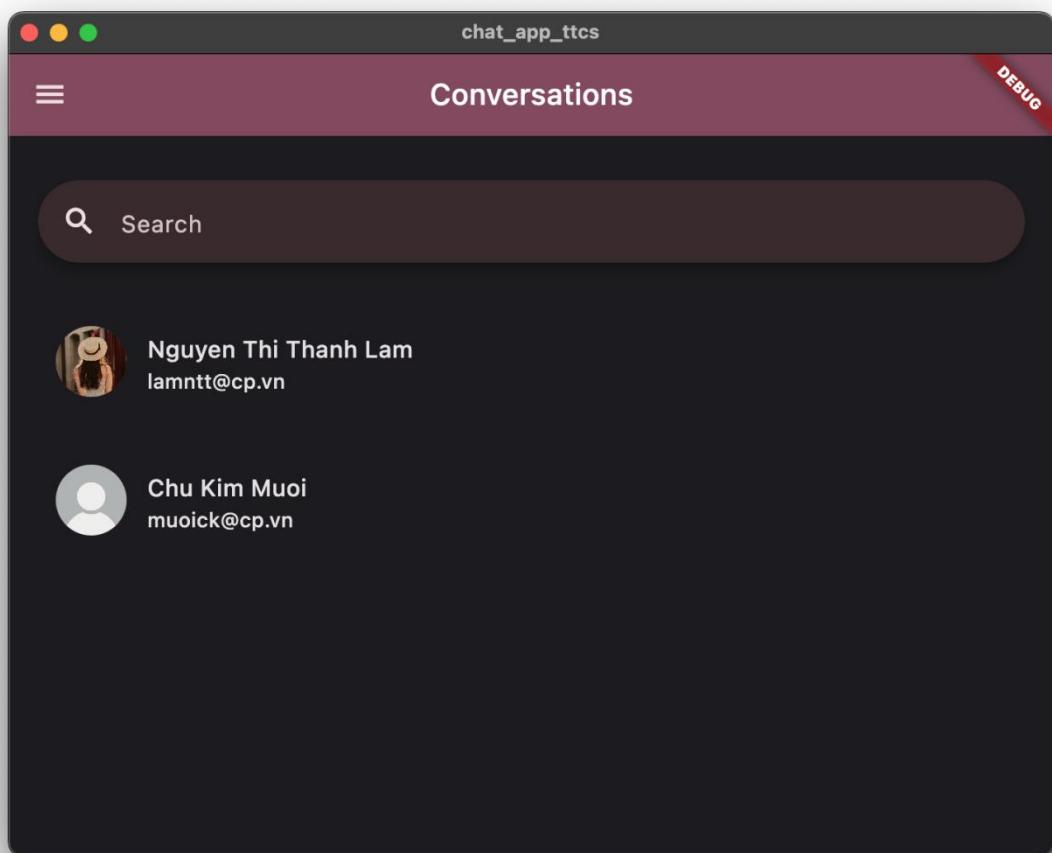
Người dùng double click vào ảnh đại diện thì sẽ hiện ra thư viện ảnh để chọn ảnh đại diện mới.



Hình 3.4.4g: Drawer của tài khoản thường

*h) Danh sách các cuộc hội thoại trong tài khoản thường*

Danh sách các cuộc hội thoại mà có người dùng tham gia sẽ được hiển thị sau khi người dùng login vào tài khoản thường.

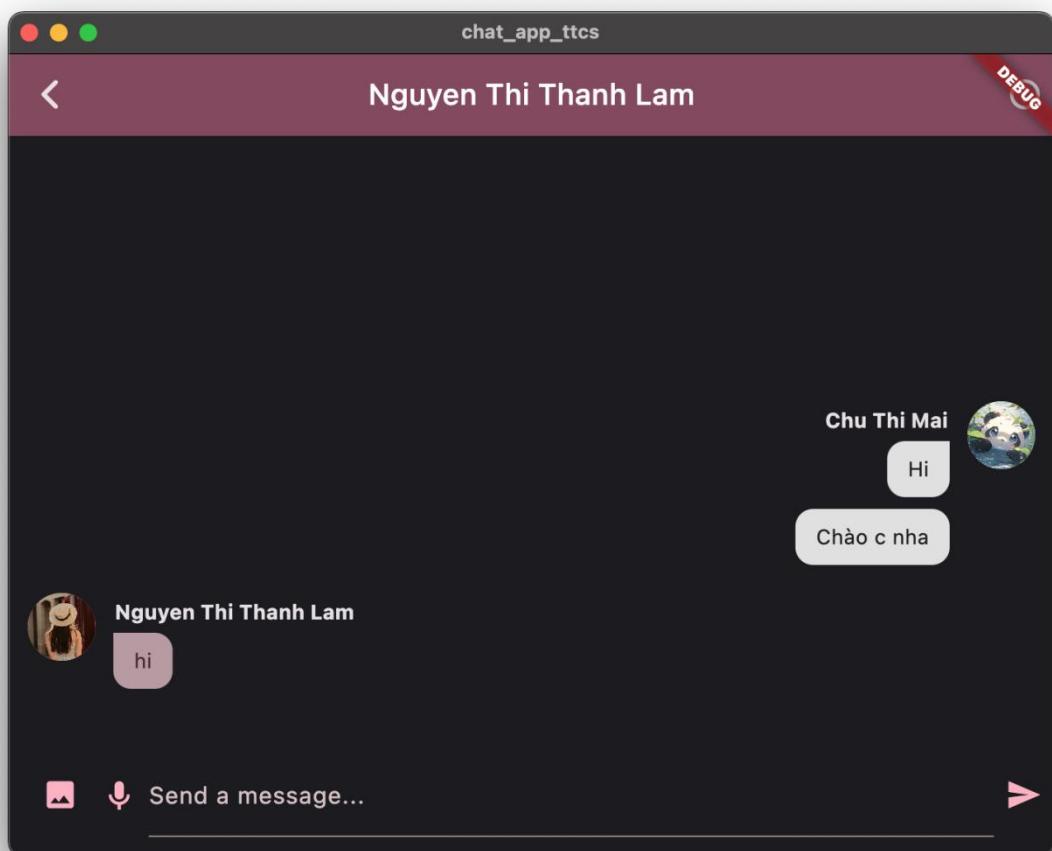


Hình 3.4.4h: Danh sách các cuộc hội thoại

i) *Cuộc hội thoại trong tài khoản thường*

Cuộc hội thoại hiển thị các tin nhắn giữa 2 bên:

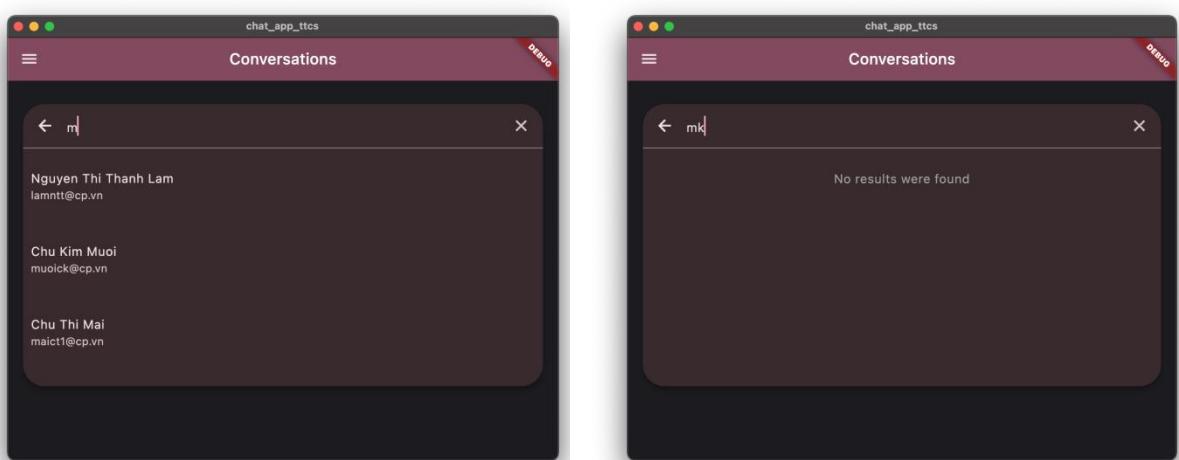
- Tin nhắn của người dùng đang đăng nhập sẽ được hiển thị bên tay phải
  - Tin nhắn của đối phương sẽ được hiển thị bên tay trái
- Dưới cuộc hội thoại là thanh soạn tin nhắn và nút gửi tin nhắn.



Hình 3.4.4i: Hiển thị đoạn chat trong cuộc hội thoại

#### j) Thanh search

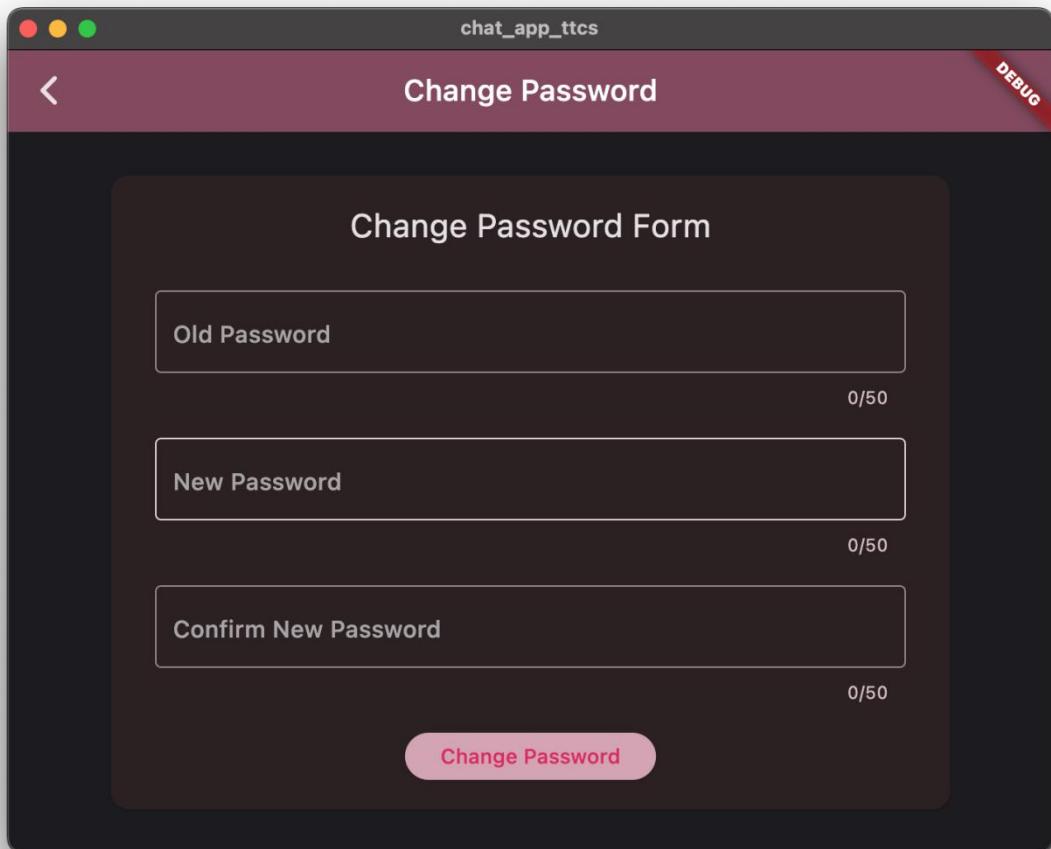
Thanh search sẽ hiển thị cái tài khoản có chứa ký tự đã gõ và chỉ hiển thị tài khoản người dùng thường.



Hình 3.4.4j: Thanh search sau khi submit trong 2 trường hợp  
tìm thấy và không tìm thấy kết quả.

### k) Trang đổi mật khẩu

Trang đổi mật khẩu gồm các trường: mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới.



Hình 3.4.4k: Trang đổi mật khẩu

### 3.4.5. Các nền tảng có thể chạy

Platform	Min Version	Đã test
Android	minSdkVersion = 21	✓
iOS	MinimumOSVersion = 13.0	✓
macOS	Minimum Deployments = 10.11	✓
Web	-	✓
windows	-	✗

### 3.4.6. Public dự án

[https://github.com/chuthimai/chat\\_app\\_ttcs](https://github.com/chuthimai/chat_app_ttcs)

## KẾT LUẬN

Trong quá trình thực hiện đề tài “Ứng dụng chat với Flutter”, kết quả đạt được đáp ứng được 1 số tính năng đã được thiết kế trước đó. Tuy nhiên bởi nhiều lý do khách quan cũng như chủ quan mà đề tài này chắc chắn vẫn còn nhiều thiếu sót.

Rất mong được thầy nhận nhận xét, đánh giá và góp ý để đề tài này được hoàn thiện và có tính thực tế hơn.

Dưới đây là một số điểm tự đánh giá về đề tài.

### 1. Kết quả đạt được

Hoàn thành ứng dụng với tính năng cơ bản:

- Ứng dụng có 2 mức tài khoản là admin và normal user.
- Admin có thể thực hiện các tính năng thêm, sửa, xóa tài khoản.
- Normal user có thể tìm kiếm các tài khoản normal user khác và có thể tạo cuộc trò chuyện với nhau.
- Nhận và gửi được tin nhắn văn bản.
- Đổi được ảnh đại diện của người dùng.
- Đổi mật khẩu
- Ứng dụng đã thử chạy được trên 4 nền tảng là: Android, iOS, macOS và web.

### 2. Khó khăn trong quá trình làm

*Một số khó khăn gặp phải trong quá trình thực hiện đề tài:*

- Xử lý, điều khiển luồng
- Cài đặt hệ thống với các nền tảng khác nhau
- Chia các thư mục sao cho hợp lý
- Thiết kế các class
- Xử lý giao diện bị tràn thành phần
- ...

*Khó khăn chưa được khắc phục:*

- Khi delete user màn hình của admin bị logout ra xong login lại. Đã nghĩ đến cách xử lý khác là sẽ đăng nhập vào tài khoản khi người dùng đã xóa đăng nhập sau đó sẽ lấy được user rồi kiểm tra user đã delete chưa thông qua state. Tuy nhiên cách làm này nếu mạng nhanh chậm thì hệ thống vẫn login vào được hệ thống 1 lúc rồi tự out ra và nó không hiển thị được thông báo vì đã bị xóa mất context.
- Thanh search chỉ search một lần, khi không chọn mà search tiếp thì lúc submit ko hiện ra kết quả như mong muốn.
- Các tính năng liên quan đến group, tính năng quên mật khẩu chưa lập trình.
- Tính năng gửi ảnh và gửi voice

### 3. Hướng phát triển

- Xây dựng thêm các tính năng liên quan đến chat nhóm và quản lý nhóm.

- Xây dựng backend bằng Python để có thể gửi email quên mật khẩu về email của người dùng.
- Thêm tính năng gửi ảnh và voice.

## TÀI LIỆU THAM KHẢO

### Tài liệu tiếng anh

- [1] Marco L. Napoli. (2019). *Beginning Flutter: A Hands On Guide to App Development*. In John Wiley & Sons, chapter 1.
- [2] Mark Clow. (2019). *Learn Google Flutter Fast: 65 Example Apps*. Amazon, part 15, 16, 17 & 28.
- [3] Prajyot Mainkar & Salvatore Giordano. (2019). *Google Flutter Mobile Development Quick Start Guide: Get up and running with iOS and Android mobile app development*. Packt Publishing, chapter 1.
- [4] Thomas Bailey & Alessandro Biessek. (2021). *Flutter for Beginners: An Introductory Guide to Building Cross-Platform Mobile Applications with Flutter 2 and Dart*. Packt Publishing, chapter 1, 2, 4 & 5.

### Tài liệu trên web

- [5] *Flutter (software)*. (2017). Truy cập 28/2/2024, từ [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)).