**BỘ CÔNG NGHIỆP**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**Báo cáo thực tập tốt nghiệp**

**Xây dựng ứng dụng chat cho điện thoại trên nền tảng Android và Nodejs**

|  |  |
| --- | --- |
| Giáo viên: | Ths Trần Hùng Cường |
| Lớp: | CNTT5 |
| Khoá: | 12 |
| Học và tên: | Vũ Văn Hoàng |

Hà nội, Năm 2021

LỜI MỞ ĐẦU

Hiện nay trong lĩnh vực lập trình ứng dụng trên điện thoại được chia làm 2 theo 2 hệ điều hành nổi tiếng là IOS và Android. Với lập trình trên IOS các ngôn ngữ lập trình được sử dụng phổ biến như Objective-C, Swift, Python… và môi trường để có thể lập trình được ứng dụng ios phải được phát triển trên nền tảng cơ bản như MacOS. Đối với Android ngôn ngữ lập trình được sử dụng phổ biến như Java, Darf, C++, Koltin, … đa phần là những ngôn ngữ lập trình quen thuộc, nền tảng. Thực tế 1 ứng dụng được phát triển cho Android không thể tương thích cho hệ điều hành IOS và ngược lại, vì vậy việc phát triển ứng dụng cho điện thoại tương thích với cả 2 hệ điều hành thực sự gặp khó khăn. Đó là lý do khi bạn dùng 1 ứng dụng trên điện thoại, có thể bạn sẽ nhận ra người phát triền luôn để 1 nhắc nhở như ‘Ứng dụng này đang được phát triển trên Android’ hoặc đơn giản là vậy. Từ năm 2016 Facebook phát triển ra 1 thư viện cho Javascript là Reactjs sau này được phát triển trở thành 1 framework giúp cho việc lập trình web trở nên dễ dàng và đạt hiệu xuất cao hơn, chưa dừng lại ở đó Facebook tiếp tục dẫn đầu về công nghệ cho ra ngay sau đó là React-native 1 framwork dành cho lập trình điện thoại, được phát triển cơ bản trên React. Với React-native lập trình viên không còn phải lo ứng dụng có thể chạy trên đơn nền tảng. Khi lập trình 1 ứng dụng sử dụng React-native bạn có thể chạy chúng trên cả 2 nền tảng IOS và Android.

MỤC LỤC

[Danh mục các kí hiệu, viết tắt 5](#_Toc67139909)

[Chương 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU 6](#_Toc67139910)

[1.1 Lý do và mục đích chọn đề tài 6](#_Toc67139911)

[1.2 Nội dung của đề tài 6](#_Toc67139912)

[Chương 2. Kiến thức cơ sở 7](#_Toc67139913)

[2.1 Tổng quan về Javascript 7](#_Toc67139914)

[2.2 Javascript và Java 7](#_Toc67139915)

[2.3 Framework React Native 7](#_Toc67139916)

[2.4 React Native hoạt động như thế nào? 8](#_Toc67139917)

[2.5 NodeJS 9](#_Toc67139918)

[2.6 Cơ sở dữ liệu 9](#_Toc67139919)

[Chương 3. Phân tích thiết kế hệ thống 10](#_Toc67139920)

[3.1 Các tác nhân chính và use case 10](#_Toc67139921)

[3.1.1 Mô tả use case Đăng nhập 11](#_Toc67139922)

[3.1.2 Mô tả use case Đăng kí 11](#_Toc67139923)

[3.1.3 Mô tả use xem danh bạ 12](#_Toc67139924)

[3.1.4 Mô tả use case Gửi tin nhắn 12](#_Toc67139925)

[3.1.5 Mô tả use case gửi tệp dữ liệu 13](#_Toc67139926)

[3.1.6 Mô tả use case gửi ảnh trực tiếp 13](#_Toc67139927)

[3.1.7 Mô tả use case gọi điện 14](#_Toc67139928)

[3.1.8 Mô tả use case gọi video 14](#_Toc67139929)

[3.1.9 Mô tả use case tạo nhóm nhắn tin 15](#_Toc67139930)

[3.1.10 Mô tả use case xem thông tin cá nhân 15](#_Toc67139931)

[3.1.11 Mô tả use case chỉnh sửa thông tin cá nhân 16](#_Toc67139932)

[3.1.12 Mô tả use case chỉnh sửa ảnh đại diện 16](#_Toc67139933)

[3.2 Phân tích use case 17](#_Toc67139934)

[3.2.1 Phân tích các use case 17](#_Toc67139935)

[Chương 4. KẾT QUẢ THỰC TẬP 20](#_Toc67139936)

[4.1 Môi trường, thiết bị yêu cầu 20](#_Toc67139937)

[4.1.1 Scripts 20](#_Toc67139938)

[4.2 Giao diện hệ thống 21](#_Toc67139939)

[4.2.1 Giao diện đăng nhập 21](#_Toc67139940)

[4.2.2 Giao diện đăng kí 22](#_Toc67139941)

[4.2.3 Giao diện danh sách chat 23](#_Toc67139942)

[4.2.4 Giao diện danh bạ 24](#_Toc67139943)

[4.2.5 Giao diện danh sách nhóm 25](#_Toc67139944)

[4.2.6 Giao diện tạo nhóm 26](#_Toc67139945)

[4.2.7 Giao diện chat 29](#_Toc67139946)

[4.2.8 Giao diện gửi ảnh, video, tệp 31](#_Toc67139947)

[4.2.9 Giao diện khi được gọi tới 33](#_Toc67139948)

[4.2.10 Giao diện bắt đầu gọi 35](#_Toc67139949)

[4.2.11 Giao diện gọi video 36](#_Toc67139950)

[4.2.12 Giao diện thông tin cá nhân 38](#_Toc67139951)

[Chương 5. TÀI LIỆU THAM KHẢO 39](#_Toc67139952)

# Danh mục các kí hiệu, viết tắt

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **English** | **Tiếng việt** |
| CSDL | Database | Cơ sở dữ liệu |
| DBMS | Database Management System | Hệ quản trị cơ sở dữ liệu |
| I/O | Input/Output | Đầu vào / Đầu ra |
| IT | Infomation technology | Công nghệ thông tin |
| JS | JavaScript | JavaScript |
| RN | React Native |  |
| RDBMS | Relational Database Management System | Hệ quản trị cơ sở dữ liệu quan hệ |

# TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## Lý do và mục đích chọn đề tài

Tại thời điểm hiện tại, công nghệ hỗ trợ cho việc lập trình ứng dụng điện thoại đang rất phát triển. Vì muốn tìm hiểu và triển khai, nên em đã chọn đề tài về lập trình ứng dụng trên nền tảng Android.

## Nội dung của đề tài

Lập trình ứng dụng chat cho điện thoại trên nền tảng Android, Nodejs.

# Kiến thức cơ sở

## Tổng quan về Javascript

JavaScript là một ngôn ngữ hướng đối tượng, đa nền tảng được sử dụng để làm cho các trang web trở nên tương tác (ví dụ: có các hoạt ảnh phức tạp, các nút có thể nhấp, menu bật lên, v.v.). Ngoài ra còn có các phiên bản JavaScript phía máy chủ nâng cao hơn như Node.js, cho phép bạn thêm nhiều chức năng vào trang web hơn là tải xuống tệp (chẳng hạn như cộng tác trong thời gian thực giữa nhiều máy tính). Bên trong môi trường máy chủ (ví dụ: trình duyệt web), JavaScript có thể được kết nối với các đối tượng của môi trường của nó để cung cấp khả năng kiểm soát theo chương trình đối với chúng.

## Javascript và Java

Về cơ bản thì JavaScript giống với Java, nhưng đối với JavaScript thì bản thân nó không quá tuân thủ các nguyên tắc về kiểu dữ liệu như Java, giúp cho việc lập trình với JavaScript thật sự thoải mái và bất quy tắc, vì vậy Js được kết hợp với HTML khá tốt, nó giúp cho người mới tiếp cận với lập trình web, hay các ứng dụng cơ bản không còn quá khó khăn, vì không cần quá bận tâm đến kiểu dữ liệu, các khái niệm cơ bản trong lập trình.

## Framework React Native

React Native là một khung mã nguồn mở để xây dựng các ứng dụng Android và iOS bằng cách sử dụng React và các khả năng gốc của nền tảng ứng dụng. Với React Native, bạn sử dụng JavaScript để truy cập các API của nền tảng cũng như để mô tả giao diện và hành vi của giao diện người dùng của bạn bằng cách sử dụng các thành phần React: gói mã có thể sử dụng lại, có thể lồng vào nhau.

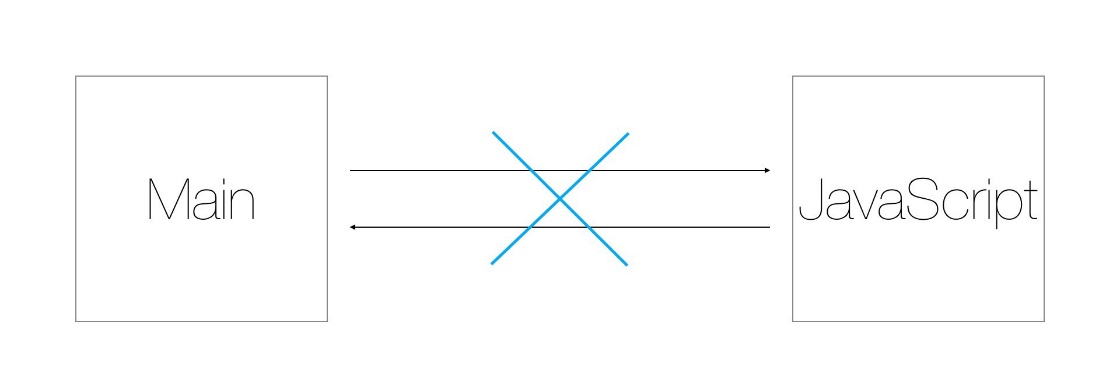
## React Native hoạt động như thế nào?

Có hai luồng quan trọng đang chạy trong mỗi ứng dụng React Native.

Một trong số đó là luồng chính, cũng chạy trong mỗi ứng dụng gốc tiêu chuẩn. Nó xử lý việc hiển thị các thành phần của giao diện người dùng và xử lý các cử chỉ của người dùng.

Cái còn lại dành riêng cho React Native. Nhiệm vụ của nó là thực thi mã JavaScript trong một công cụ JavaScript riêng biệt. JavaScript xử lý logic nghiệp vụ của ứng dụng. Nó cũng xác định cấu trúc và các chức năng của giao diện người dùng.

Hai luồng này không bao giờ giao tiếp trực tiếp và không bao giờ chặn lẫn nhau:



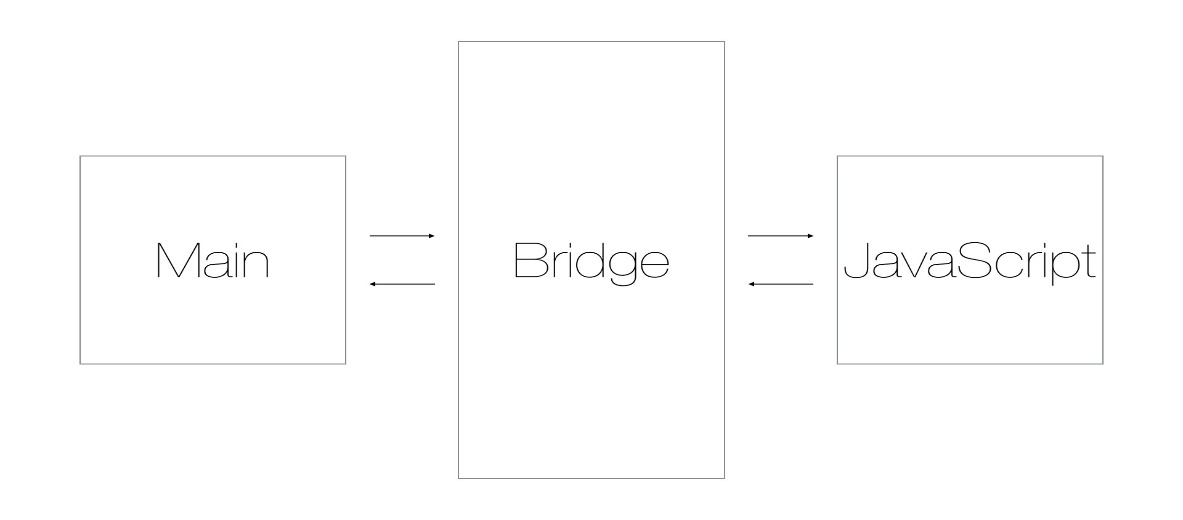
Làm thế nào để các chủ đề tương tác?

Giữa hai luồng này là cái gọi là cầu nối, là cốt lõi của React Native. Cây cầu có ba đặc điểm quan trọng.

**Asynchronous.** Nó cho phép giao tiếp không đồng bộ giữa các luồng. Điều này đảm bảo rằng chúng không bao giờ chặn lẫn nhau.

**Batched**. Nó chuyển các thông điệp từ luồng này sang luồng khác theo cách tối ưu hóa.

**Serializable.** Hai luồng không bao giờ chia sẻ hoặc hoạt động với cùng một dữ liệu. Thay vào đó, chúng trao đổi các tin nhắn tuần tự.



## NodeJS

Node.js là một môi trường thời gian chạy JavaScript đa nền tảng và mã nguồn mở. Nó là một công cụ phổ biến cho hầu hết mọi loại dự án.

Node.js chạy công cụ JavaScript V8, cốt lõi của Google Chrome, bên ngoài trình duyệt. Điều này cho phép Node.js hoạt động rất hiệu quả.

Ứng dụng Node.js được chạy trong một quy trình duy nhất mà không cần tạo luồng mới cho mọi yêu cầu. Node.js cung cấp một tập hợp các nguyên thủy I/O không đồng bộ trong thư viện tiêu chuẩn của nó để ngăn chặn mã JavaScript bị chặn và nói chung, các thư viện trong Node.js được viết bằng cách sử dụng các mô hình không chặn, làm cho hành vi chặn trở thành ngoại lệ thay vì bình thường.

Khi Node.js thực hiện một thao tác I/O, chẳng hạn như đọc từ mạng, truy cập cơ sở dữ liệu hoặc hệ thống tệp, thay vì chặn luồng và lãng phí chu kỳ CPU chờ đợi, Node.js sẽ tiếp tục hoạt động khi phản hồi trở lại.

Điều này cho phép Node.js xử lý hàng nghìn kết nối đồng thời với một máy chủ duy nhất mà không gây ra gánh nặng quản lý đồng thời luồng, vốn có thể là một nguồn lỗi đáng kể.

Node.js có một lợi thế duy nhất bởi vì hàng triệu nhà phát triển giao diện người dùng viết JavaScript cho trình duyệt hiện có thể viết mã phía máy chủ ngoài mã phía máy khách mà không cần phải học một ngôn ngữ hoàn toàn khác.

## Cơ sở dữ liệu

Cơ sở dữ liệu là một ứng dụng riêng biệt để lưu trữ một bộ sưu tập dữ liệu. Mỗi cơ sở dữ liệu có một hoặc nhiều API riêng biệt để tạo, truy cập, quản lý, tìm kiếm và sao chép dữ liệu mà nó nắm giữ.

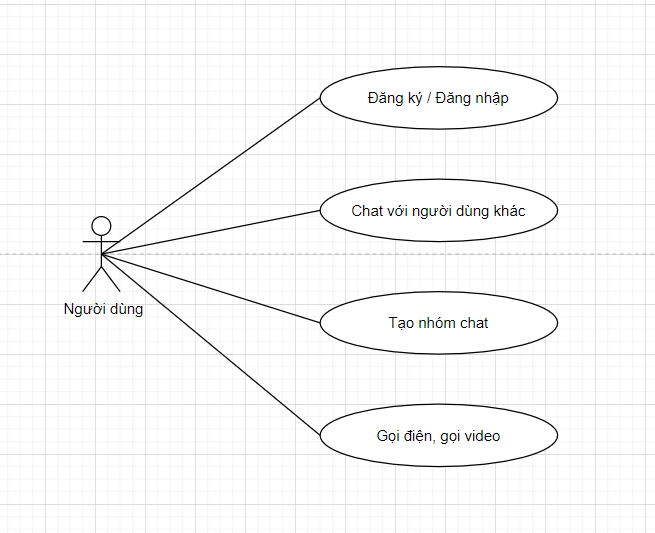
MySQL là 1 hệ thống quản lí cơ sở dữ liệu có quan hệ, sử dụng SQL là ngôn ngữ truy vấn.

* MySQL rất dễ dàng sử dụng, nó là mã nguồn mở và hoàn toàn miễn phí
* Với những người bắt đầu sử dụng cơ sở dữ liệu, MySQL rất dễ dàng để cài đặt và kết nối, khả năng lưu trữ đủ lớn dành cho việc tiếp cận và được hỗ trợ trên nhiều hệ điều hành.

# Phân tích thiết kế hệ thống

## Các tác nhân chính và use case

* Người dùng
* Use case người dùng



### Mô tả use case Đăng nhập

Người dùng đăng nhập vào hệ thống để sử dụng các chức năng của hệ thống

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Người dùng bắt đầu với màn hình đăng nhập, người dùng chỉ cần nhập số điện thoại đã đăng kí trên hệ thống sau đó bấm nút đăng nhập.
  + Luồng rẻ nhánh:
    1. Tại bước 2 của luồng cơ bản, nếu người dùng nhập sai số điện thoại hệ thống sẽ bắt người dùng nhập lại.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.

### Mô tả use case Đăng kí

Người dùng đăng kí để sử dụng hệ thống

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Người dùng bấm vào đăng kí, ứng dụng sẽ chuyển đến màn hình đăng kí.
    2. Người dùng điền thông tin đăng kí gồm tên và số điện thoại sau đó bấm nút đăng kí.
  + Luồng rẽ nhánh:
    1. Nếu người dùng nhập thiếu trường, số điện thoại không hợp lệ, hệ thống sẽ báo lỗi cho người dùng, yêu cầu người dùng nhập chính xác.
    2. Nếu số điện thoại đã tồn tại trong hệ thống, thông báo lỗi, yêu cầu người dùng nhập lại.
    3. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.

### Mô tả use xem danh bạ

Người dùng xem người dùng có trong hệ thống

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng chọn vào biểu tượng danh bạ ở góc dưới cùng màn hình điện thoại
    2. Hệ thống sẽ trả về cho người dùng tất cả người dùng trong hệ thống.
  + Luồng rẽ nhánh:
    1. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đã đăng nhập

### Mô tả use case Gửi tin nhắn

Người dùng gửi tin nhắn cho người khác

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng chọn 1 người để di chuyển đến màn hình nhắn tin
    2. Người dùng nhập nội dung tin nhắn rồi bấm vào biểu tưởng gửi tin nhắn.
  + Luồng rẽ nhánh:
    1. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đã đăng nhập.

### Mô tả use case gửi tệp dữ liệu

Người dùng muốn gửi tệp dữ liệu cho người dùng khác

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng đã ở màn hình nhắn tin, người dùng chọn vào biểu tượng gửi file, ứng dụng sẽ cho phép người dùng chọn kiểu dữ liệu muốn gửi lên bao gồm:

ảnh, tệp, video, ảnh động…

* + 1. Người dùng chọn kiểu dữ liệu muốn gửi lên rồi bấm biểu tượng gửi.
  + Luồng rẽ nhánh:
    1. Lỗi đến từ điện thoại của người dùng, thông báo từ hệ điều hành điện thoại sẽ được hiện ra.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đã đăng nhập

Ứng dụng đã được cấp quyền đầy đủ bởi điện thoại người dùng bao gồm: Quyền truy cập file, truy cập máy ảnh, truy cập bộ nhớ (mọi yều cầu này đều an toàn, được xác minh)

### Mô tả use case gửi ảnh trực tiếp

Người dùng gửi ảnh trực tiếp bằng cách chụp 1 ảnh mới

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng đang ở màn hình nhắn tin, người dùng chọn biểu tượng file gửi lên, rồi chọn vào biểu tượng camera.
    2. Người dùng chụp ảnh theo mục đích rồi chọn ‘OK’ để lưu lại ảnh trên tin nhắn sau đó ấn gửi.
  + Luồng rẽ nhánh:
    1. Lỗi đến từ điện thoại của người dùng, thông báo từ hệ điều hành điện thoại sẽ được hiện ra.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đã đăng nhập

Ứng dụng đã được cấp quyền truy cập vào camera của điện thoại (yêu cầu truy nhập này là an toàn)

### Mô tả use case gọi điện

Người dùng gọi điện cho người dùng khác

Luồng sự kiện:

* + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng đang ở màn hình nhắn tin, sau đó người dùng bấm vào biểu tượng điện thoại.
    2. Ứng dụng sẽ bắt đầu kết nối đến người dùng còn lại.
  + Luồng rẽ nhánh:
    1. Lỗi đến từ điện thoại của người dùng, hoặc kết nối mạng không ổn định, thông báo từ hệ điều hành điện thoại sẽ được hiện ra.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đăng nhập.

### Mô tả use case gọi video

Người dùng gọi video cho người dùng khác

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng chọn đang ở màn hình nhắn tin và chọn vào biểu tượng máy quay để gọi video.
    2. Ứng dụng sẽ yêu cầu đến người dùng còn lại.
  + Luồng rẽ nhánh:
    1. Lỗi đến từ điện thoại của người dùng, hoặc kết nối mạng không ổn định, thông báo từ hệ điều hành điện thoại sẽ được hiện ra.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đăng nhập

Ứng dụng đã được cấp quyền truy cập camera

### Mô tả use case tạo nhóm nhắn tin

Người dùng tạo nhóm và mời người được trong nhóm.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Người dùng chọn vào biểu tượng nhóm chat ở dưới cùng màn hình.
    2. Người dùng chọn vào biểu tưởng tạo nhóm ở góc phải trên của màn hình.
    3. Người dùng sẽ chọn tên nhóm, kiểu của nhóm bao gồm:
       - Public: nhóm công khai ai cũng có thể vào
       - Private: nhóm riêng tư, chỉ được nhìn thấy khi bạn được mời vào nhóm.
       - Password protected: Nhóm được bảo vệ, yêu cầu vào nhóm khi nhập đúng mật khẩu đã được đặt bởi chủ nhóm trước đó
    4. Người dùng bấm tạo nhóm.
  + Luồng rẽ nhánh:
    1. Nếu người dùng để trống tên nhóm, yêu cầu người dùng nhập đầy đủ.
    2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Tiền điều kiện:

Người dùng đăng nhập.

### Mô tả use case xem thông tin cá nhân

Người dùng xem thông tin cá nhân.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng chọn vào biểu tượng mở rộng ở dưới cùng màn hình.
    2. Màn hình chuyển sang giao diện thông tin cá nhân, hệ thống hiển thị thông tin người dùng đã đăng kí
  + Luồng rẽ nhánh:
    1. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Các yêu cầu đặc biệt:

Người dùng phải đăng nhập

### Mô tả use case chỉnh sửa thông tin cá nhân

Người dùng cập nhật thông tin cá nhân

* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Usecase bắt đầu khi người dùng đang ở màn hình xem thông tin cá nhân, người dùng bấm nút cập nhật.
2. Các ô thông tin người dùng chuyển thành input có thể thay đổi.
3. Người dùng điền thông tin mới và bấm nút lưu.
   * **Luồng rẽ nhánh**
     1. Người dùng nhập vào thông tin không hợp lệ hoặc bỏ trống, hệ thống thông báo lỗi và yêu cầu người dùng nhập lại.
     2. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.

* Các yêu cầu đặc biệt

Người dùng đã đăng nhập.

### Mô tả use case chỉnh sửa ảnh đại diện

Người dùng cập nhật ảnh đại diện

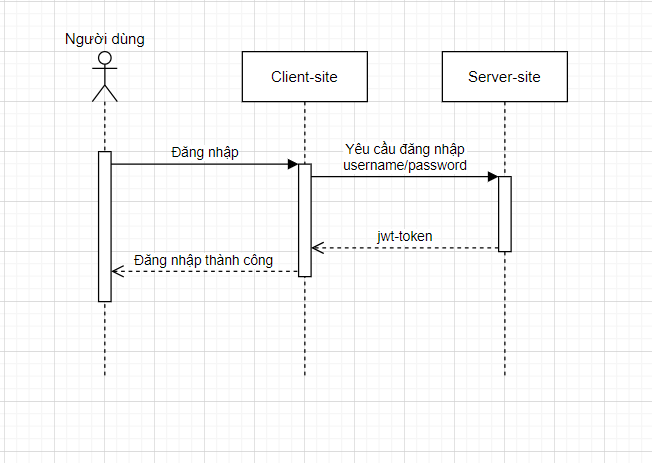
* **Luồng sự kiện**
  + **Luồng cơ bản:**
    1. Người dùng ở màn hình xem thông tin cá nhân, người dùng ấn vào ảnh đại diện của mình.
    2. Ứng dụng truy nhập vào thư viện ảnh của máy, người dùng chọn ảnh muốn tải lên sau đó ấn cập nhật.
  + **Luồng rẽ nhánh**
    1. Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.
* Các yêu cầu đặc biệt

Người dùng đã đăng nhập.

## Phân tích use case

### Phân tích các use case

1. **Phân tích use case Đăng nhập**
   1. Biểu đồ trình tự

****

1. **Phân tích use case đăng kí**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
2. **Phân tích use case gửi tin nhắn**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
3. **Phân tích use case xem danh bạ**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
4. **Phân tích use case gửi tệp dữ liệu**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
5. **Phân tích use case ảnh trực tiếp**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
6. **Phân tích use case gọi điện**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
7. **Phân tích use case gọi video**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
8. **Phân tích use case tạo nhóm chat**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
9. **Phân tích use case xem thông tin cá nhân**
   1. Biểu đồ trình tự
   2. Biểu đồ lớp phân tích
10. **Phân tích use case chính sửa thông tin cá nhân**
    1. Biểu đồ trình tự
    2. Biểu đồ lớp phân tích
11. **Phân tích use case chỉnh sửa ảnh đại diện**
    1. Biểu đồ trình tự
    2. Biểu đồ lớp phân tích

# KẾT QUẢ THỰC TẬP

## Môi trường, thiết bị yêu cầu

Yarn

NodeJs v10^

Android Studio

Java sdk

Máy điện thoại Android ( nếu có )

### Scripts

Vào thư mục root code, mở cmd:

* Tải thư viện cần thiết

> yarn

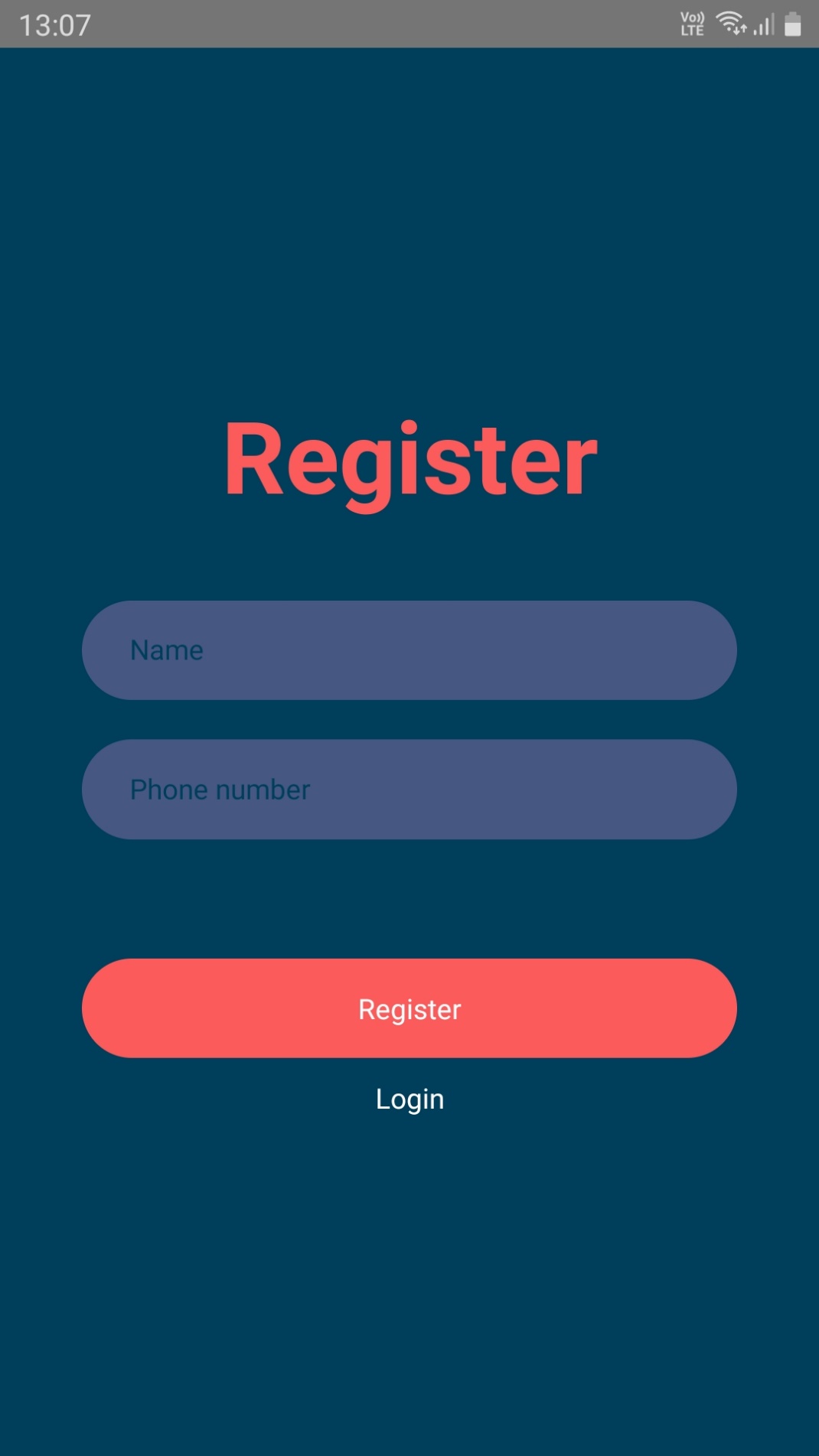
* Chạy chương trình:
  + Android
    - > yarn android
  + IOS
    - > yarn ios
* Testing
  + > yarn test
* Build
  + > yarn build

## Giao diện hệ thống

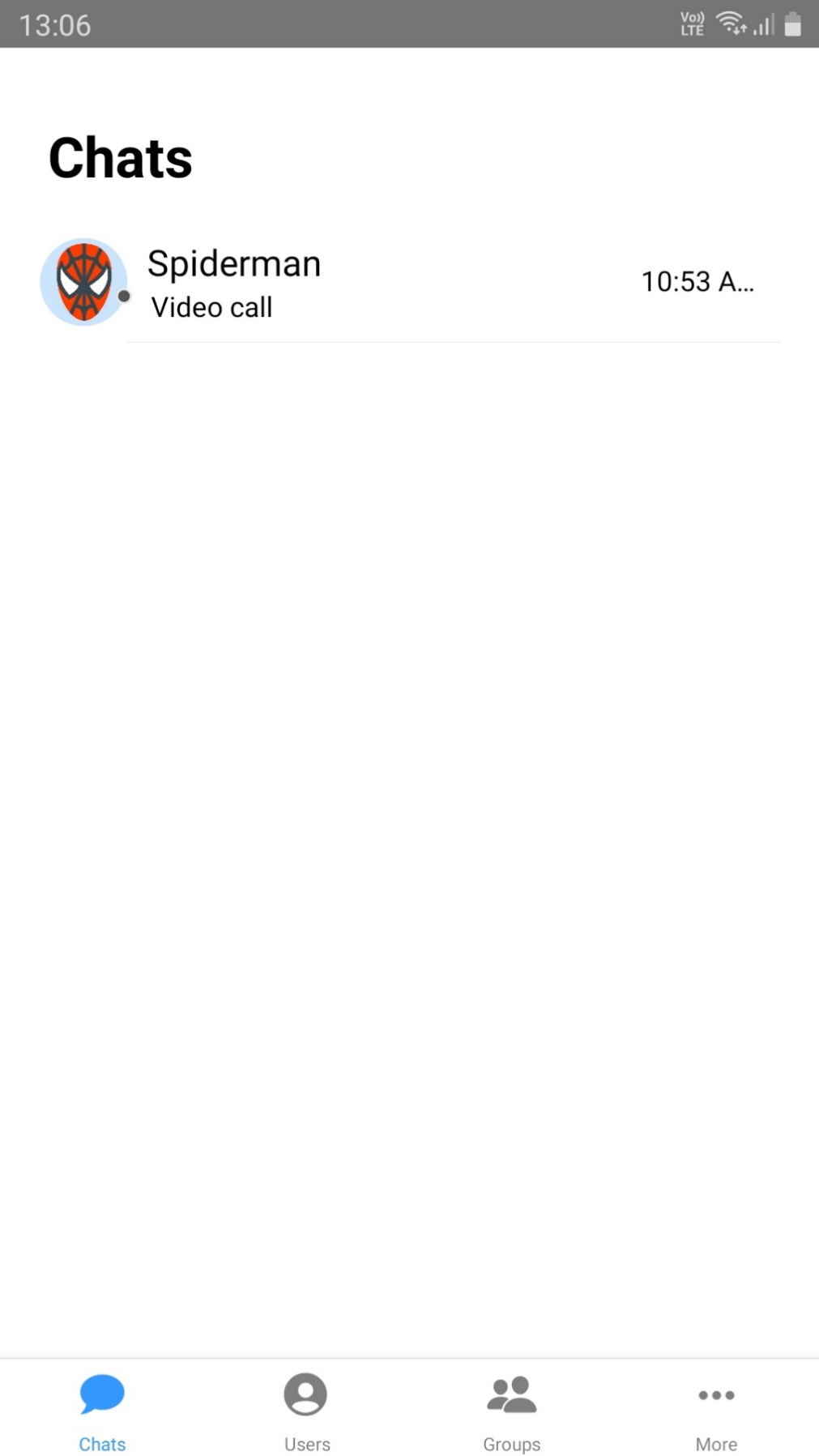
### Giao diện đăng nhập



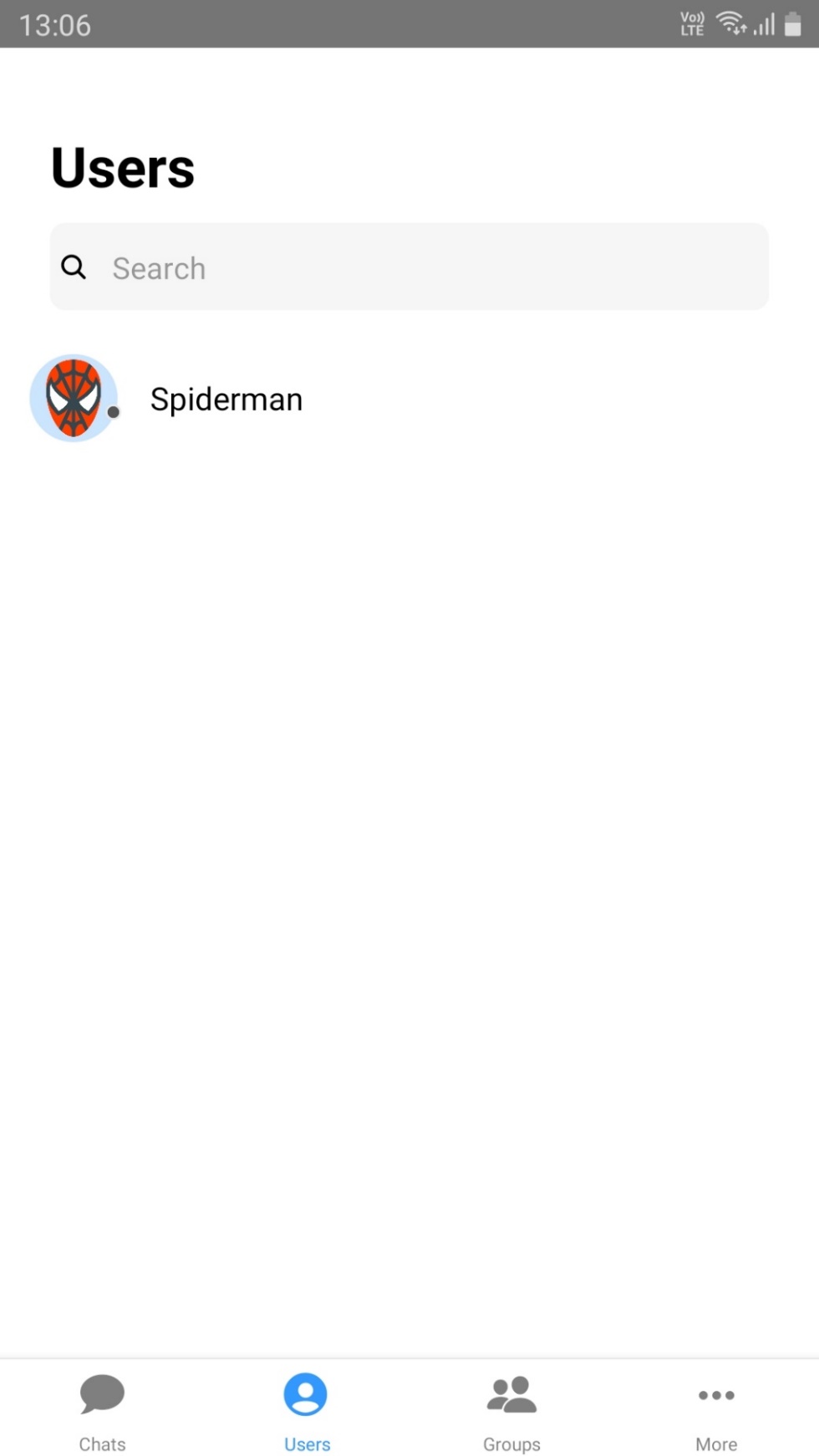
### Giao diện đăng kí



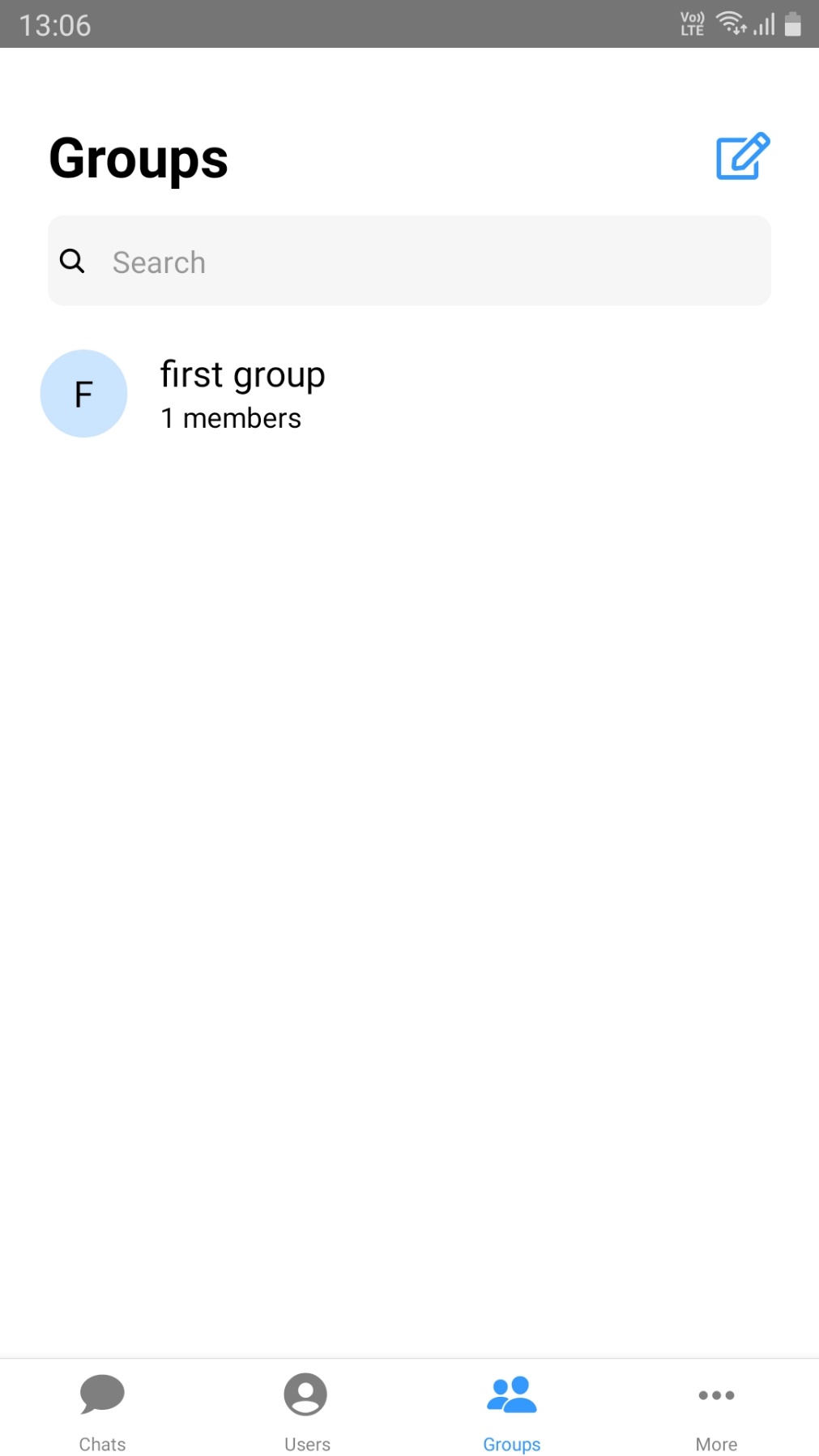
### Giao diện danh sách chat



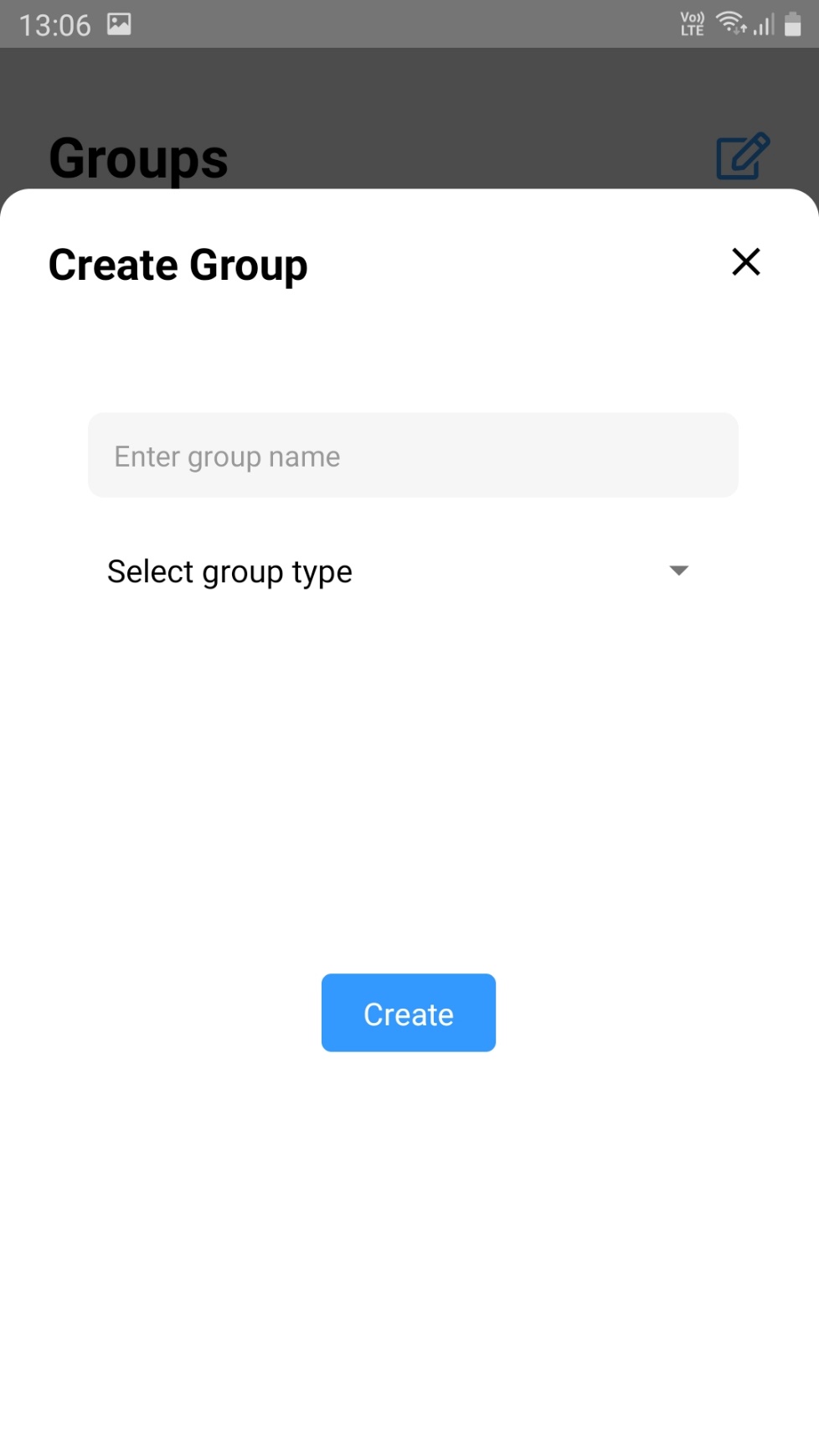
### Giao diện danh bạ

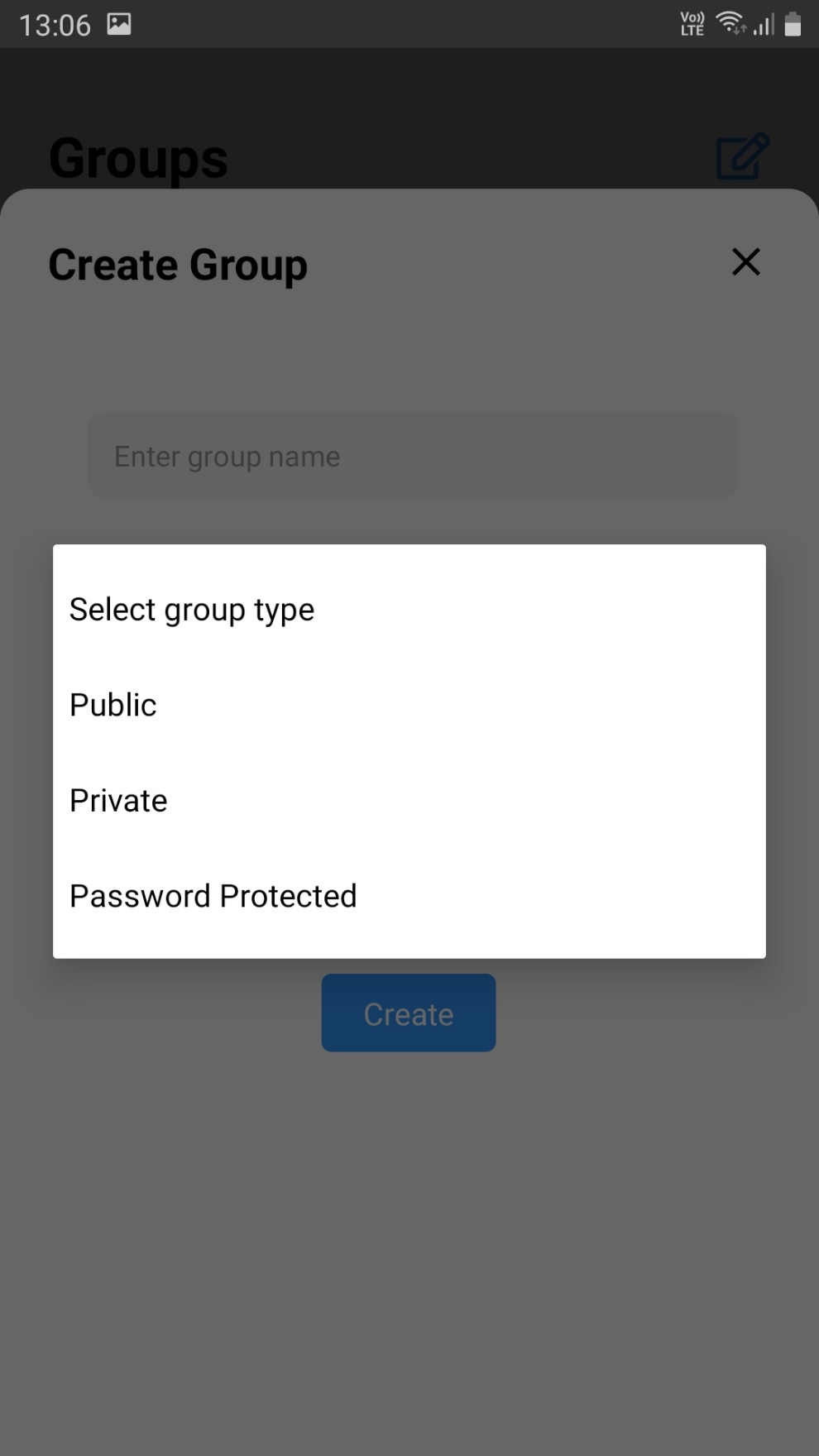
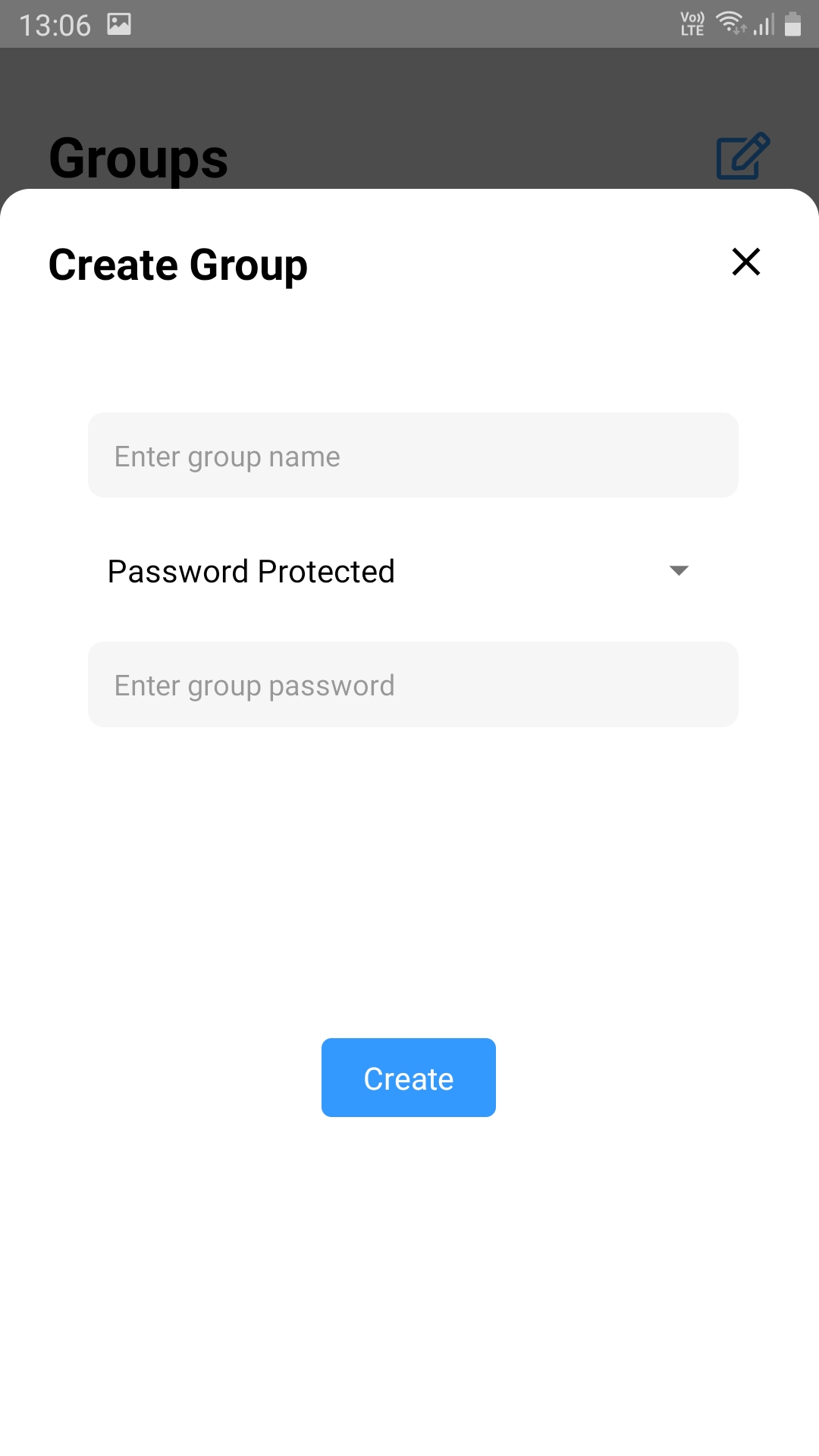


### Giao diện danh sách nhóm

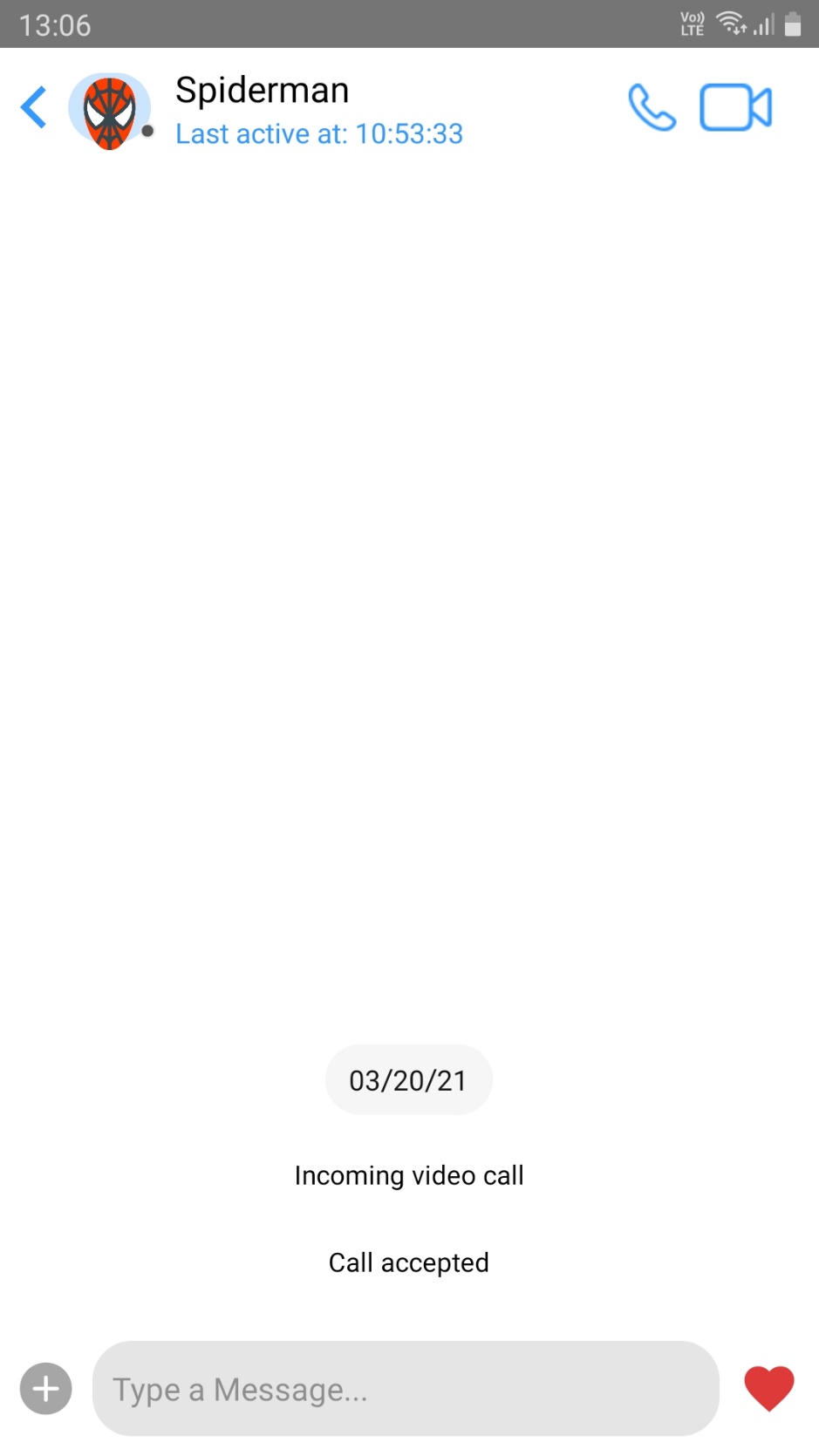


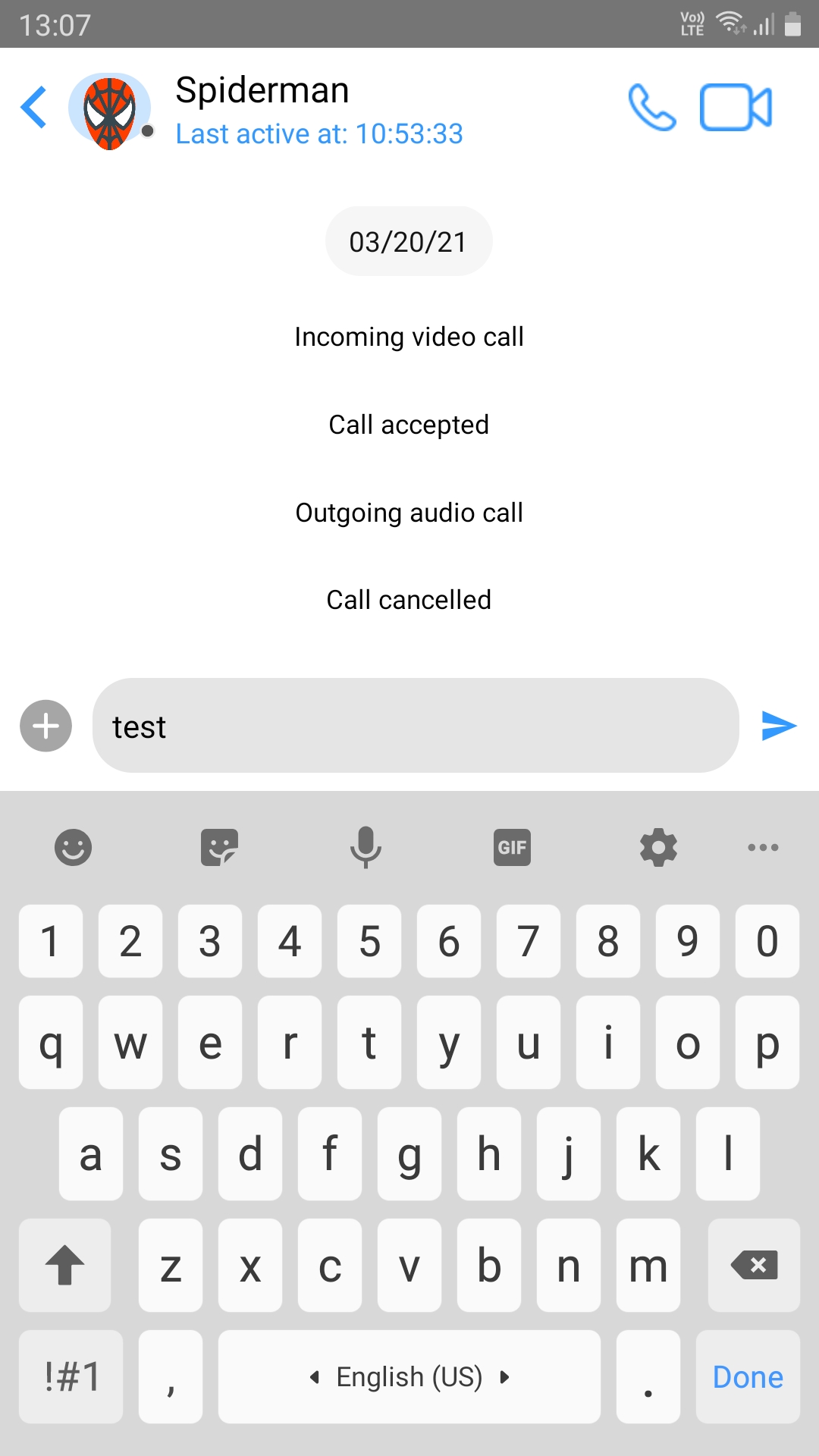
### Giao diện tạo nhóm



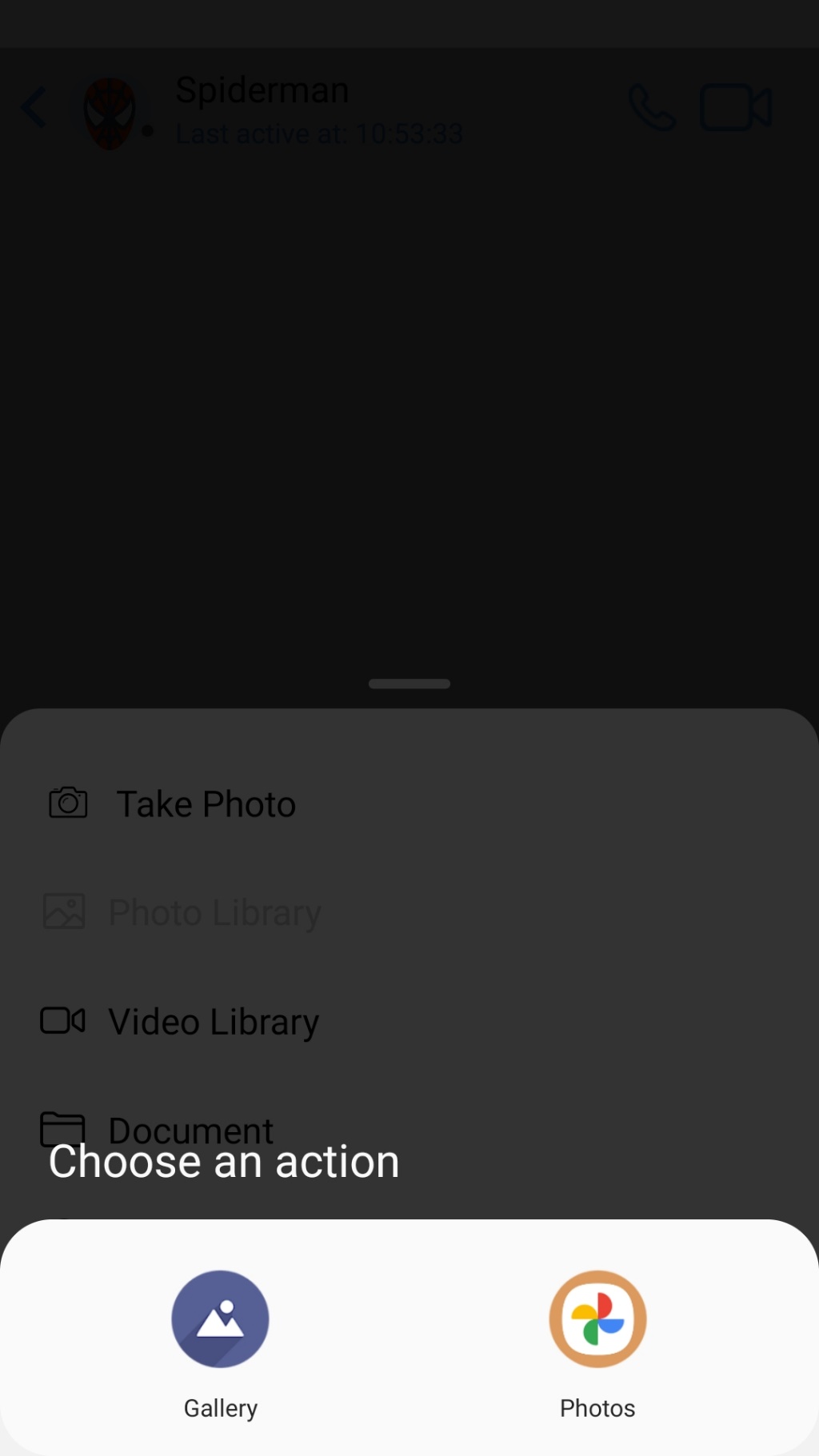
 

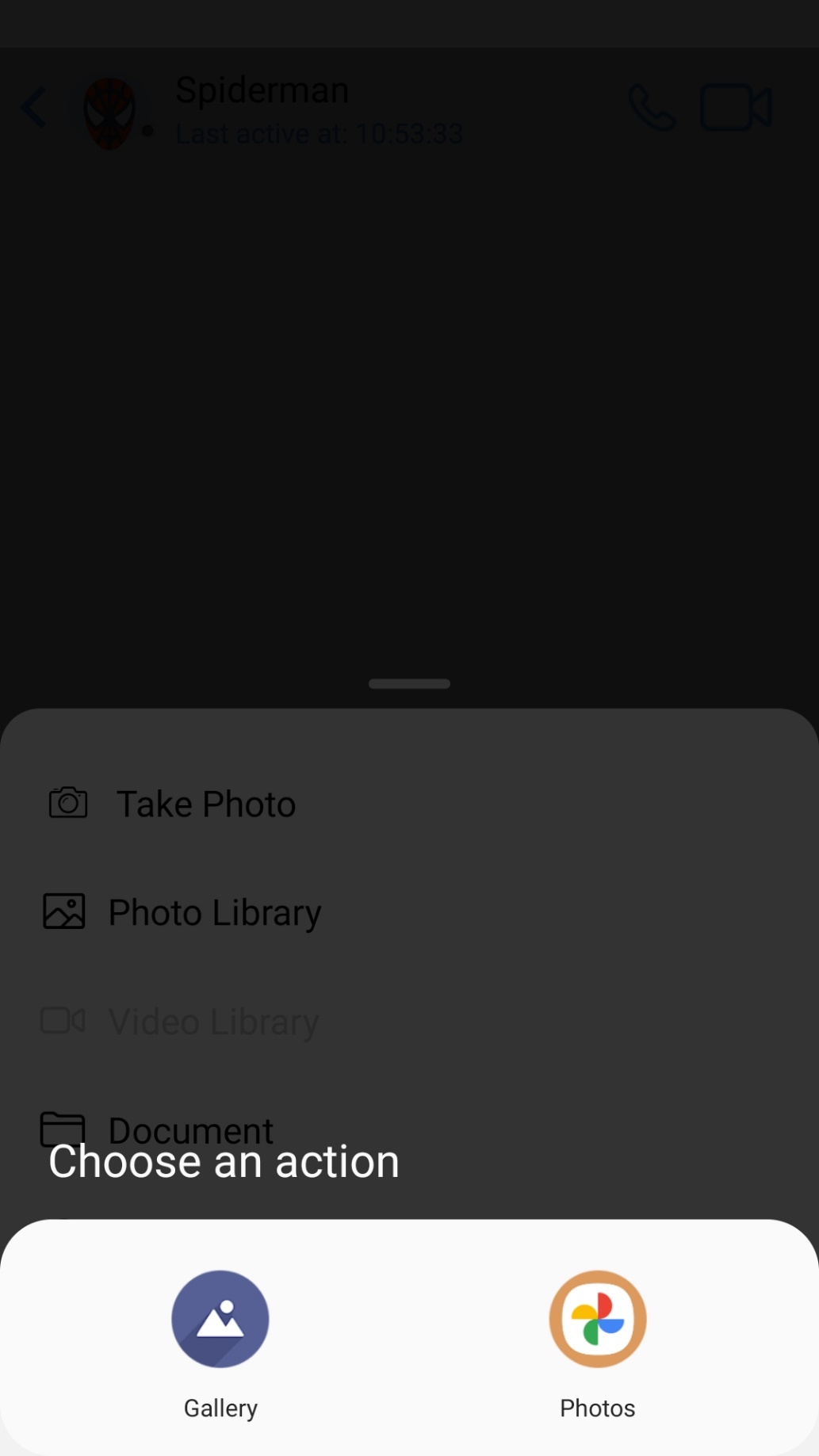
### Giao diện chat



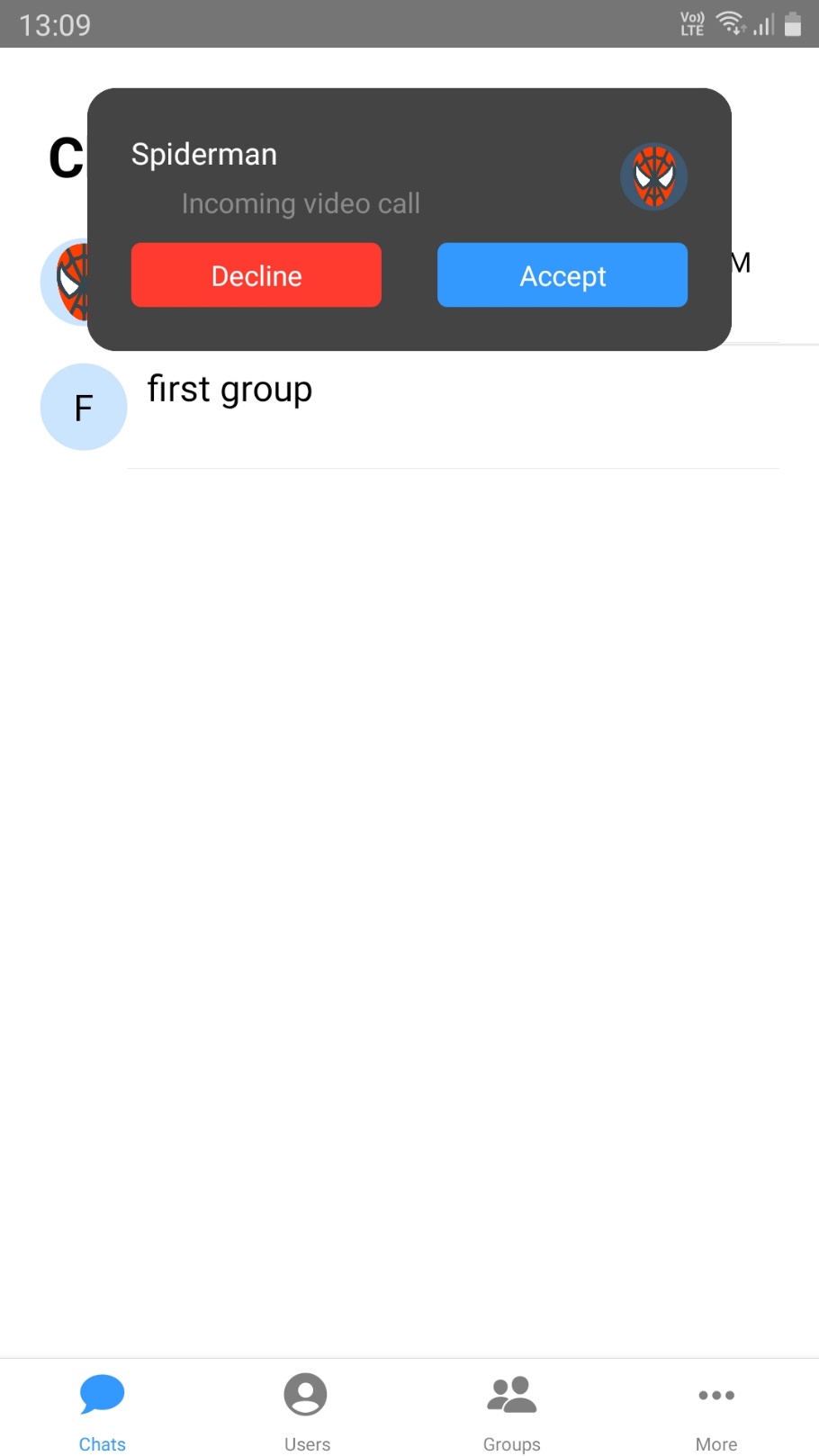


### Giao diện gửi ảnh, video, tệp





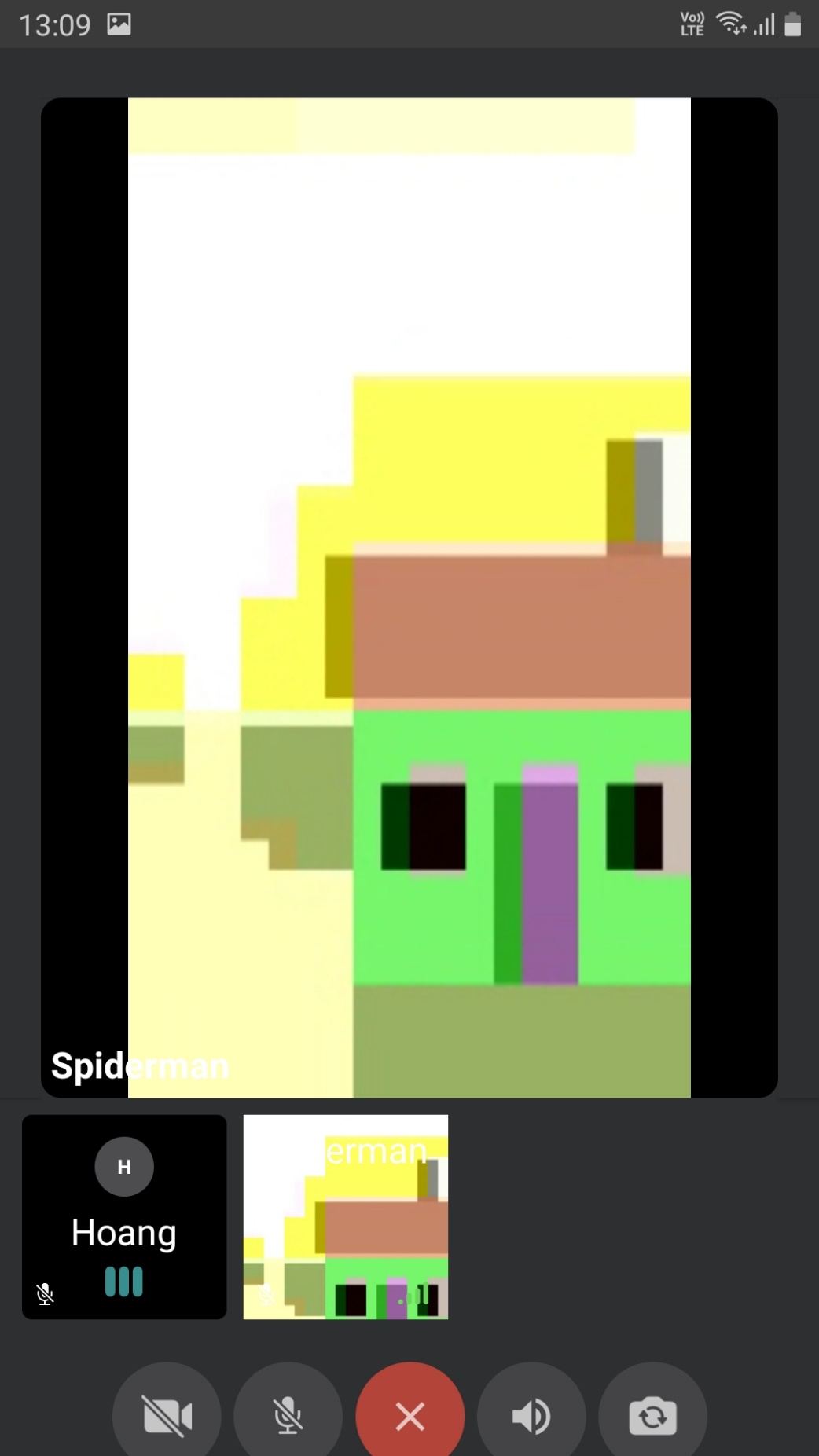
### C:\Users\ADMIN\OneDrive\Desktop\audico.jpgGiao diện khi được gọi tới

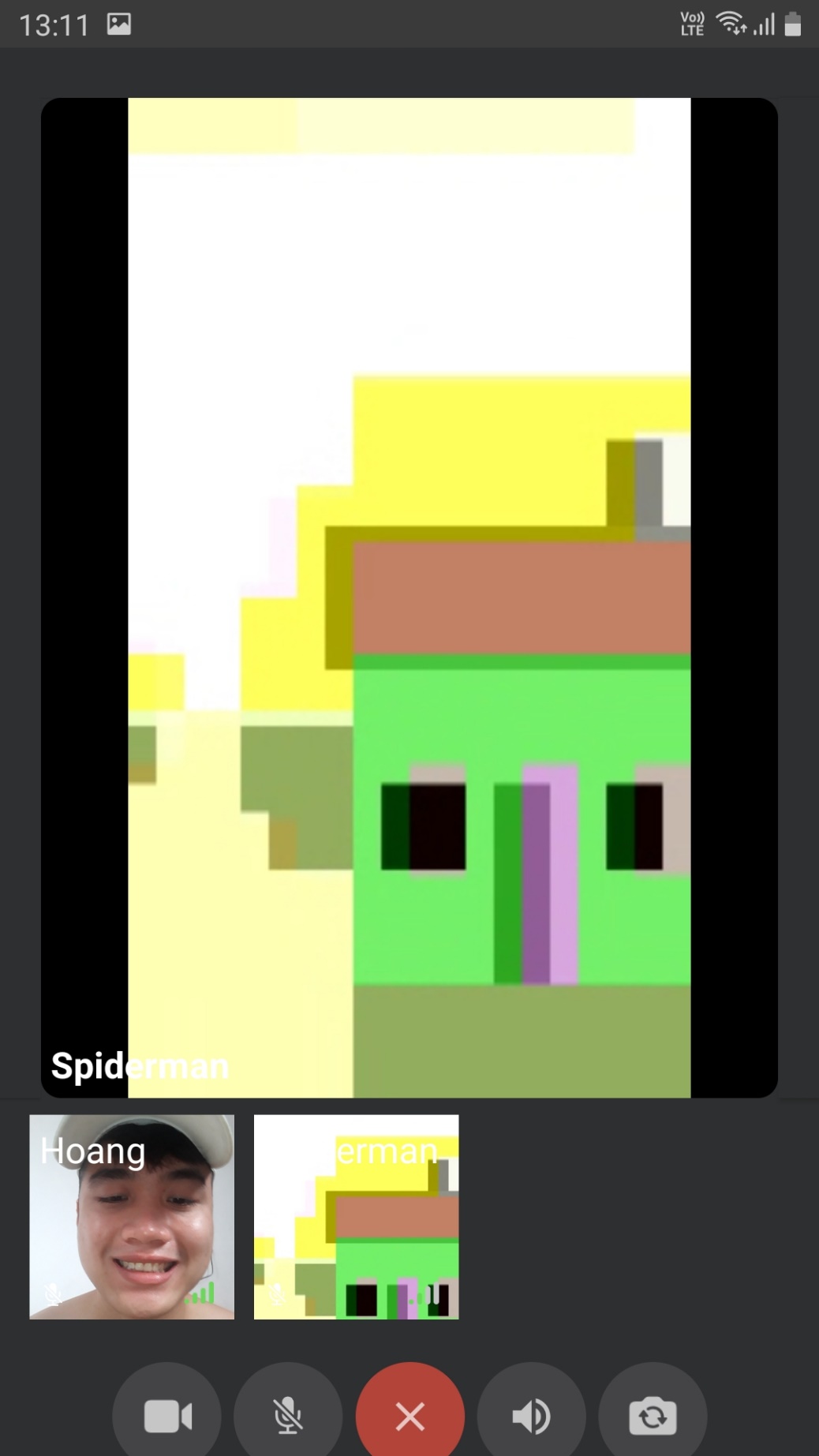


### Giao diện bắt đầu gọi

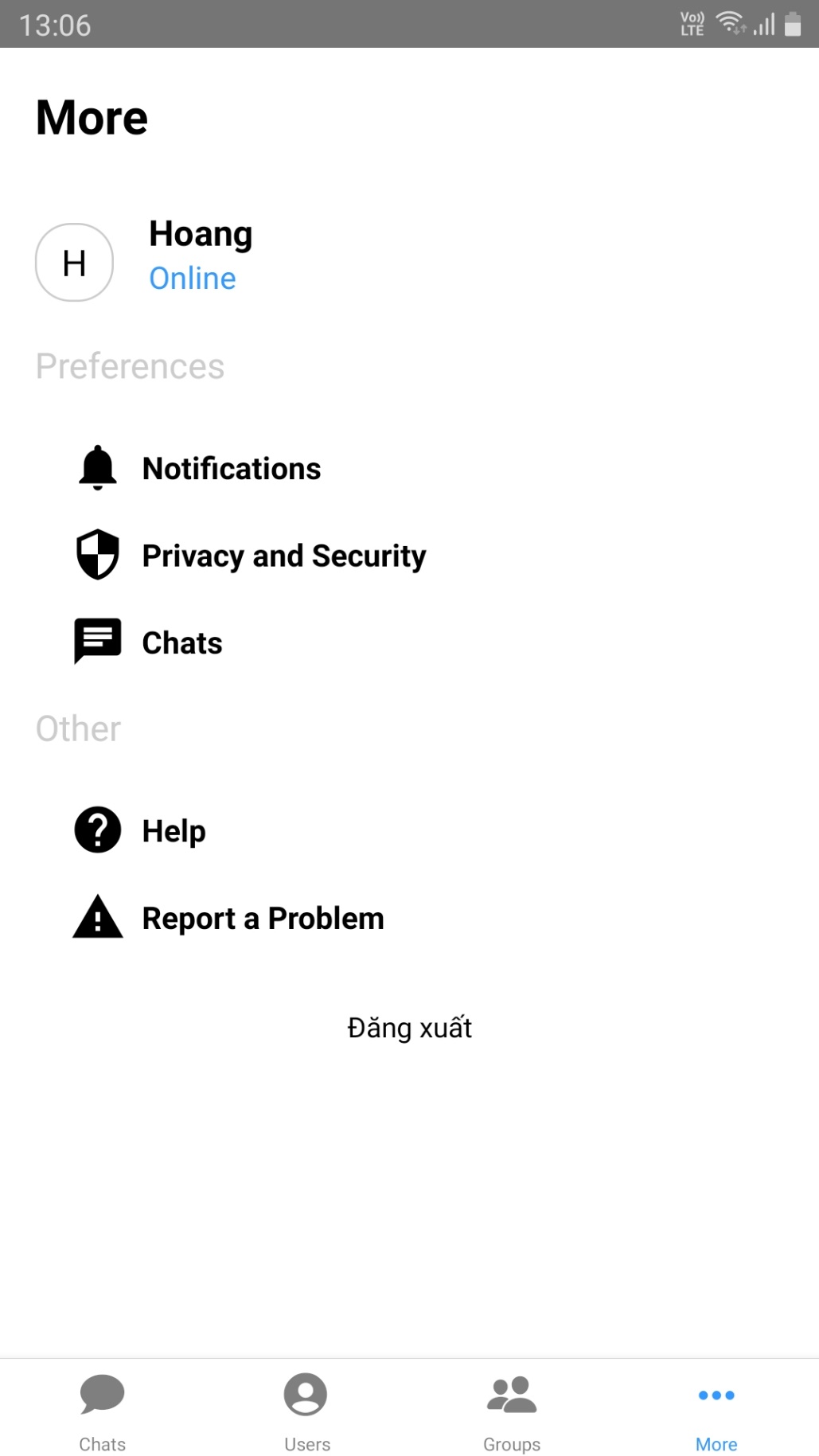


### Giao diện gọi video





### Giao diện thông tin cá nhân



# TÀI LIỆU THAM KHẢO

* <https://reactnative.dev/docs/intro-react>
* <https://nodejs.org/en/>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
* <https://www.tutorialspoint.com/mysql/mysql-introduction.htm>