

# HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

# KHOA CÔNG NGHỆ THÔNG TIN

====================

# Tài Liệu Bài Tập Lớn

Lập trình mạng

***ĐỀ TÀI***

**Ứng dụng chat TCP SOCKET**

# Lớp học phần: 04 Nhóm BTL: 11

**Sinh viên:** Nguyễn Công Huy – B18DCCN267

Vũ Minh Quang - B18DCCN487

Đinh Văn Đô - B18DCCN157

Nguyễn Viết Cường - B18DCCN069

1. Phân công công việc:

- Nguyễn Công Huy: phân tích thiết kế hệ thống, phân công công việc, code module đăng nhập đăng ký, chat đơn, gửi emoji

- Vũ Mạnh Quang: code module kết bạn, module xem, sửa thông tin cá nhân user

- Đinh Văn Đô: code module group chat, viết tài liệu

- Nguyễn Viết Cường: code module group chat, viết tài liệu

1. Mô tả hệ thống bằng ngôn ngữ tự nhiên và yêu cầu, phạm vi hệ thống

1/ Mô tả

- Hệ thống chat cho phép người dùng đăng ký tài khoản, đăng nhập, tìm kiếm và chat riêng với các user khác, tạo phòng chat nhóm, kết bạn, sửa thông tin tài khoản cá nhân, chat thì có thể là gửi tin nhắn text hoặc gửi icon emoji, hiển thị trạng thái online, đối với phòng chat đơn thì trạng thái online của người dùng còn lại trong phòng phải được cập nhật real time.

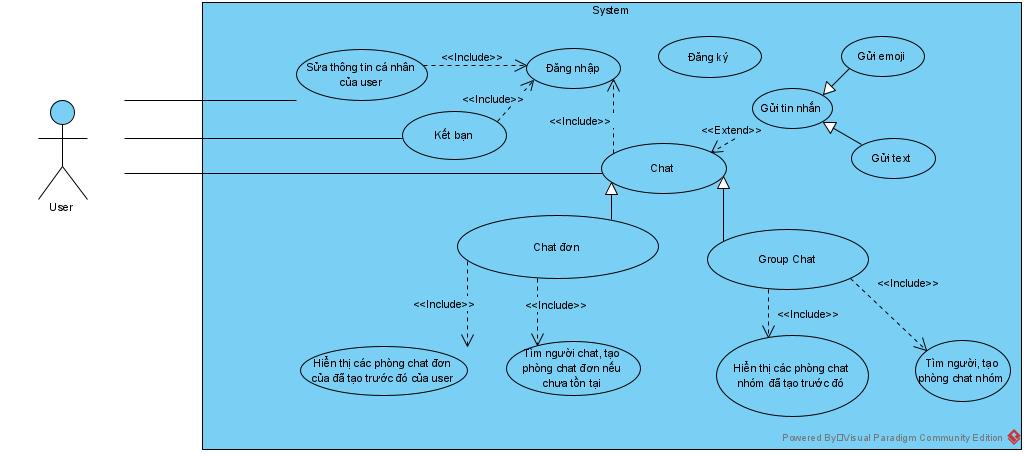
2/ Yêu cầu, phạm vi hệ thống

- Hệ thống có 1 server có nhiệm vụ tiếp nhận yêu cầu từ client rồi gửi lại dữ liệu cho client hoặc chuyển tiếp cho client khác

- Hệ thống phải lưu lại được thông tin các phòng chat đơn, các phòng chat nhóm của từng user một, lưu được thông tin tin nhắn của từng user trong từng phòng để lần sau vào thấy được các tin nhắn cũ và các tin nhắn đã được user khác gửi đến trong thời gian 1 user offline, lưu trạng thài kết bạn.

- Sử dụng Java Swing để code UI, Java TCP Socket để code chức năng gửi nhận các gói tin

1. Biểu đồ use case toàn hệ thống

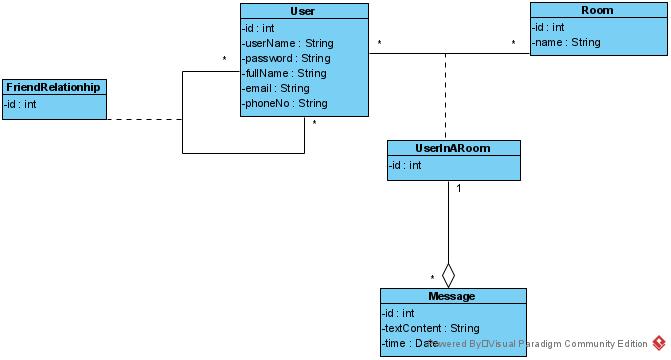


1. Biểu đồ lớp pha phân tích:

- 1 User có thể ở trong nhiều phòng, 1 phòng có thể có 2 hay nhiều user(phòng nào có 2 người là phòng chat đơn) do đó có thêm 1 bảng UserInARoom đại diện cho quan hệ giữa 1 thực thể user và 1 thực thể phòng

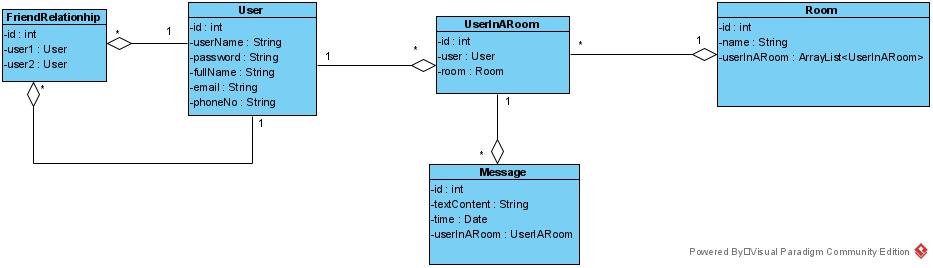
- 1 người ở trong 1 phòng sẽ có thể có nhiều tin nhắn, 1 tin nhắn thì thuộc 1 người trong 1 phòng

- 1 User sẽ có thể là bạn với nhiều User khác, mỗi User khác đó lại có thể là bạn với nhiều User, vì vậy cần có thêm một bảng là FriendRelationship đại diện cho 1 quan hệ bạn bè giữa 2 thực thể User xác định.



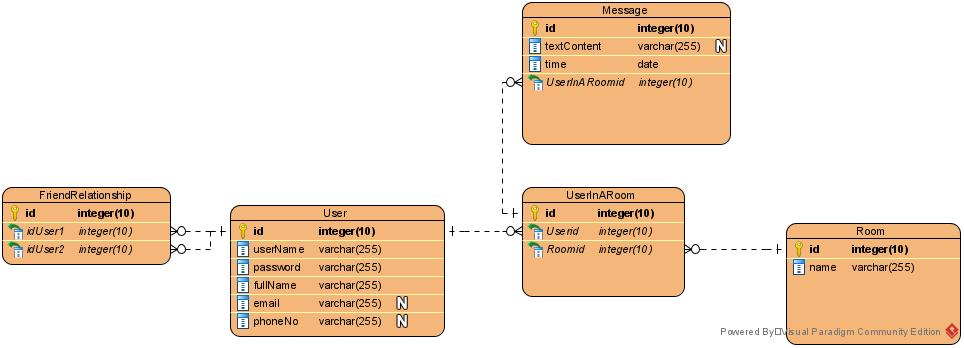
1. Biểu đồ lớp pha thiết kế

- Phá quan hệ association class, thêm id, cho các lớp thực thể tham gia làm thuộc tính của các lớp thực thể khác



1. Database

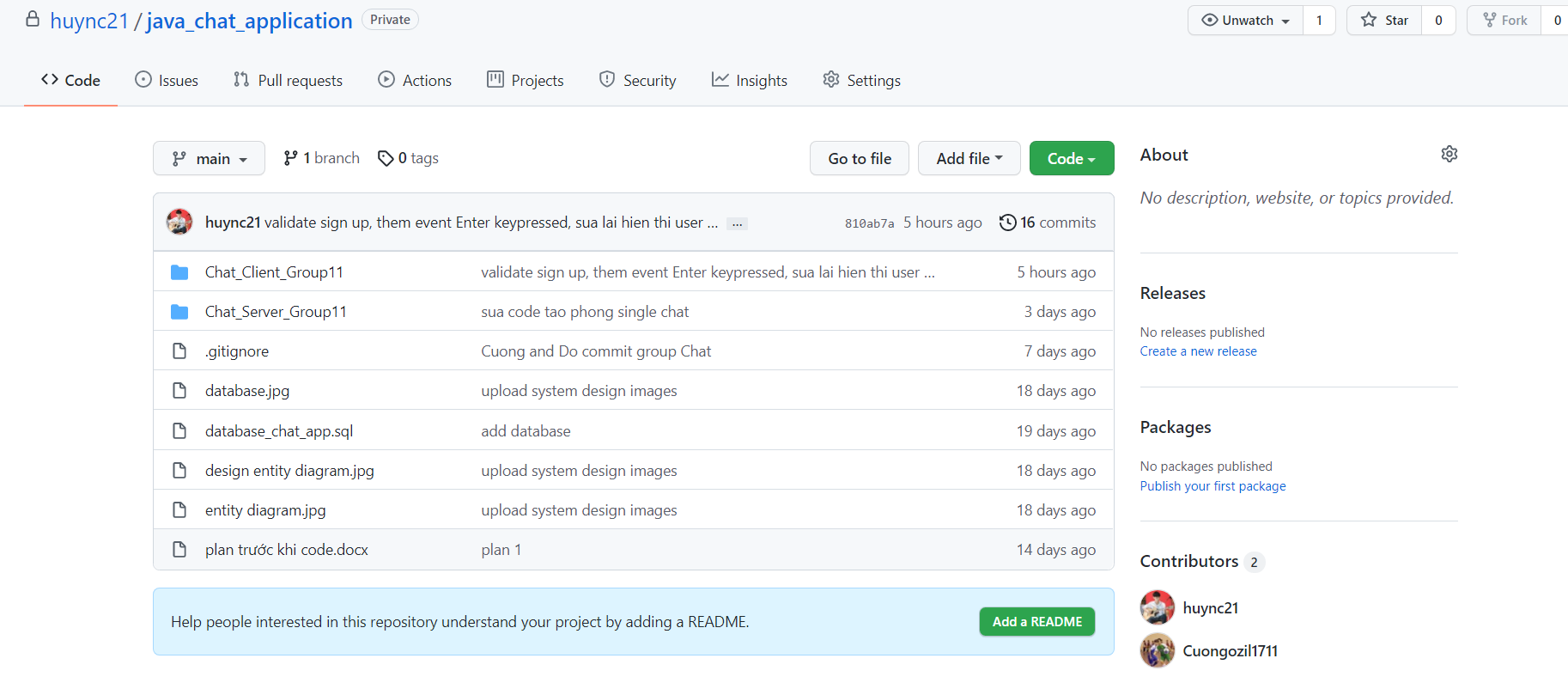
- Cơ sở dữ liệu: bỏ đi các thuộc tính ở các lớp thực thể, thêm khóa ngoại khóa chính



1. Quá trình thực hiện
2. Quản lý source code:

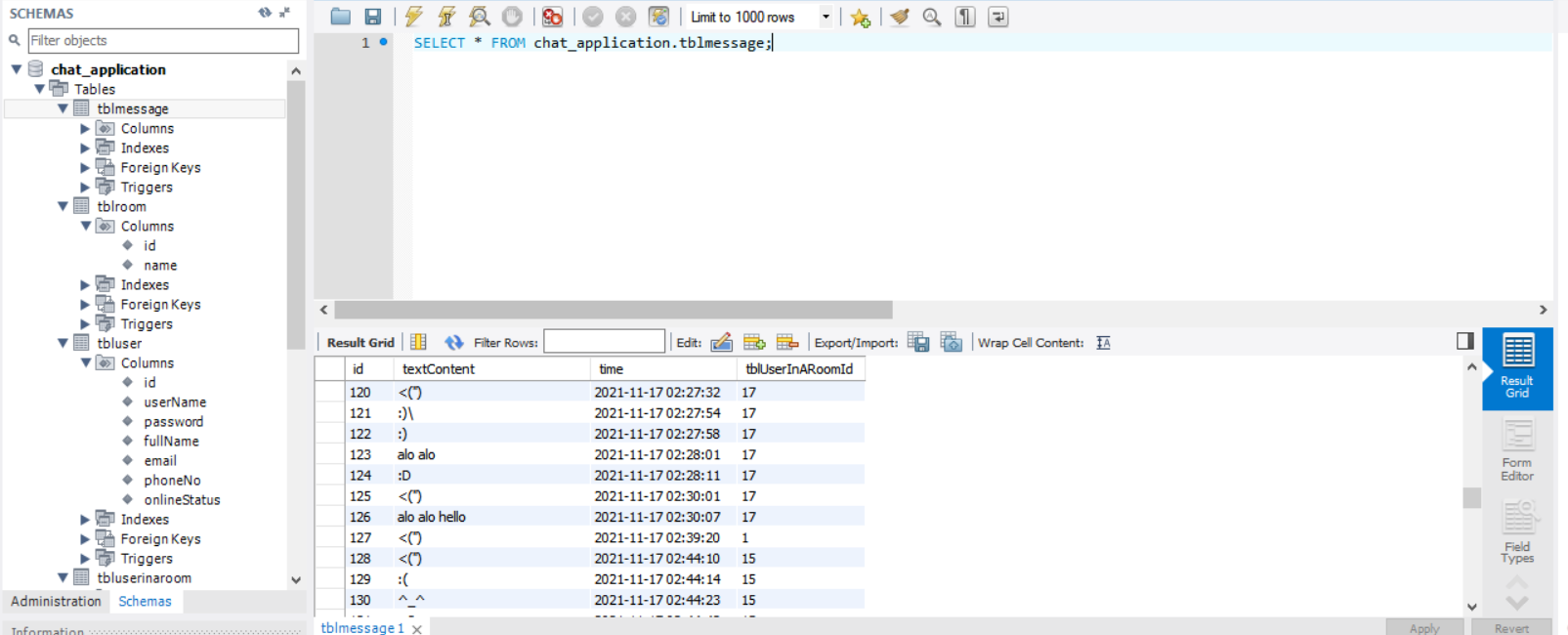
- Sử dụng git hub để làm việc

- Thống nhất phiên bản MySQL WorkBench 8.0, Java JDK 8



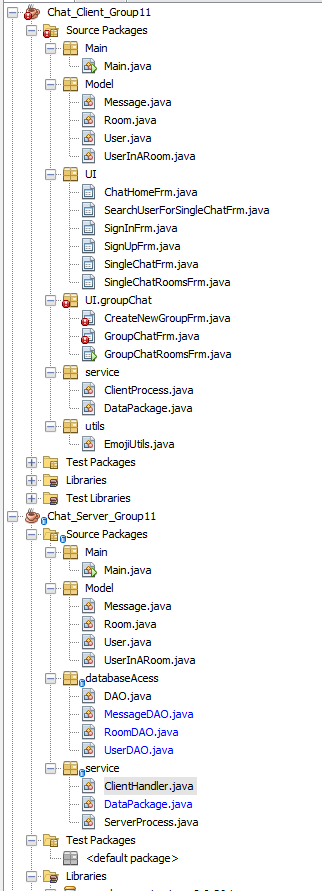
1. Tạo data base:

Sử dụng MySQL WorkBench



1. Cấu trúc project (sử dụng Netbeans):

- Bao gồm 1 project Client và 1 project Server



1. Phương thức gửi nhận của hệ thống:

ClientHandler

ClientProcess

Client 1

Server

Chứa socket Chứa socket

ServerProcess

ClientHandler

ClientProcess

Client 2

Chứa socket Chứa socket

…... ……….. Chứa ServerSocket

ClientHandler

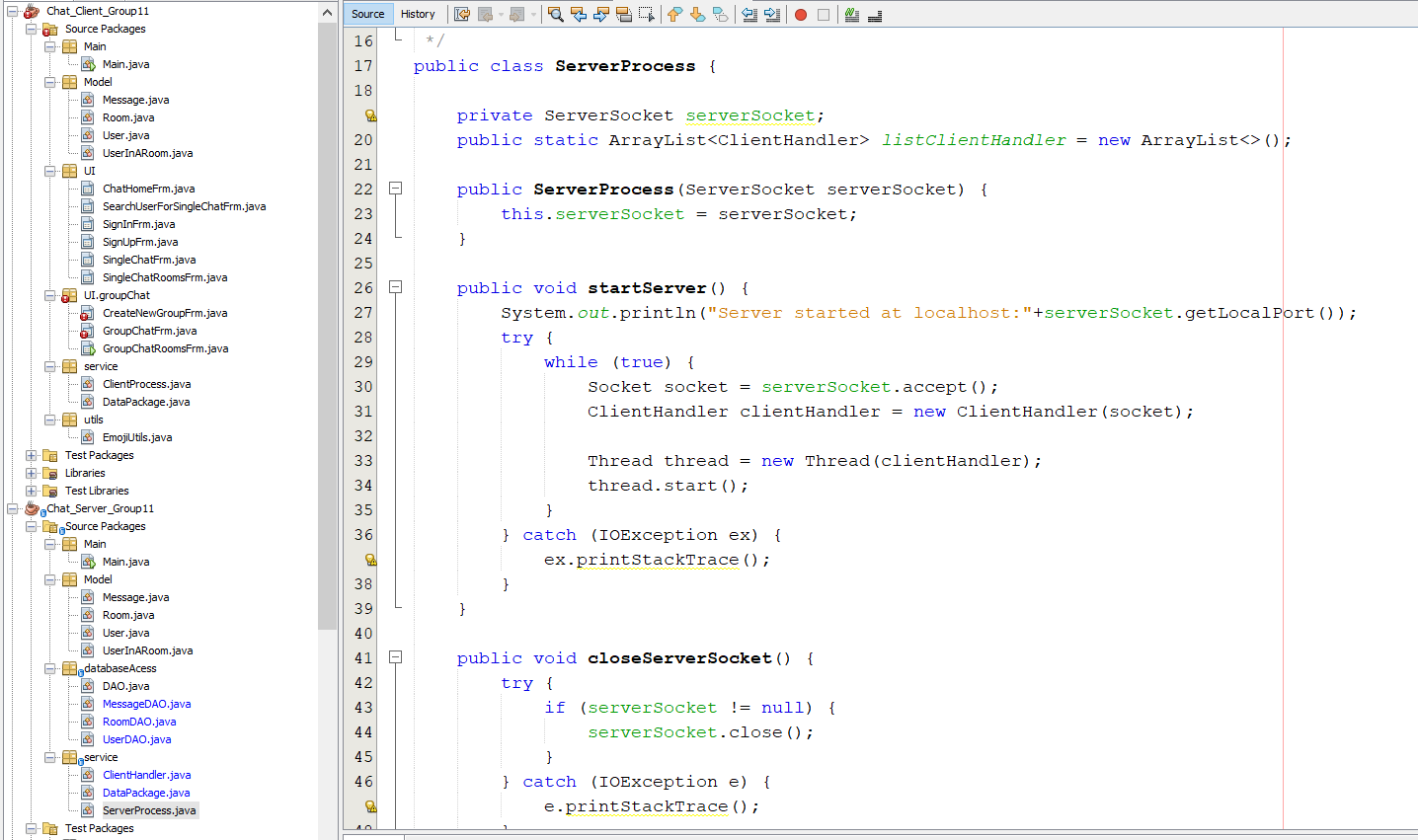
ClientProcess

Client n

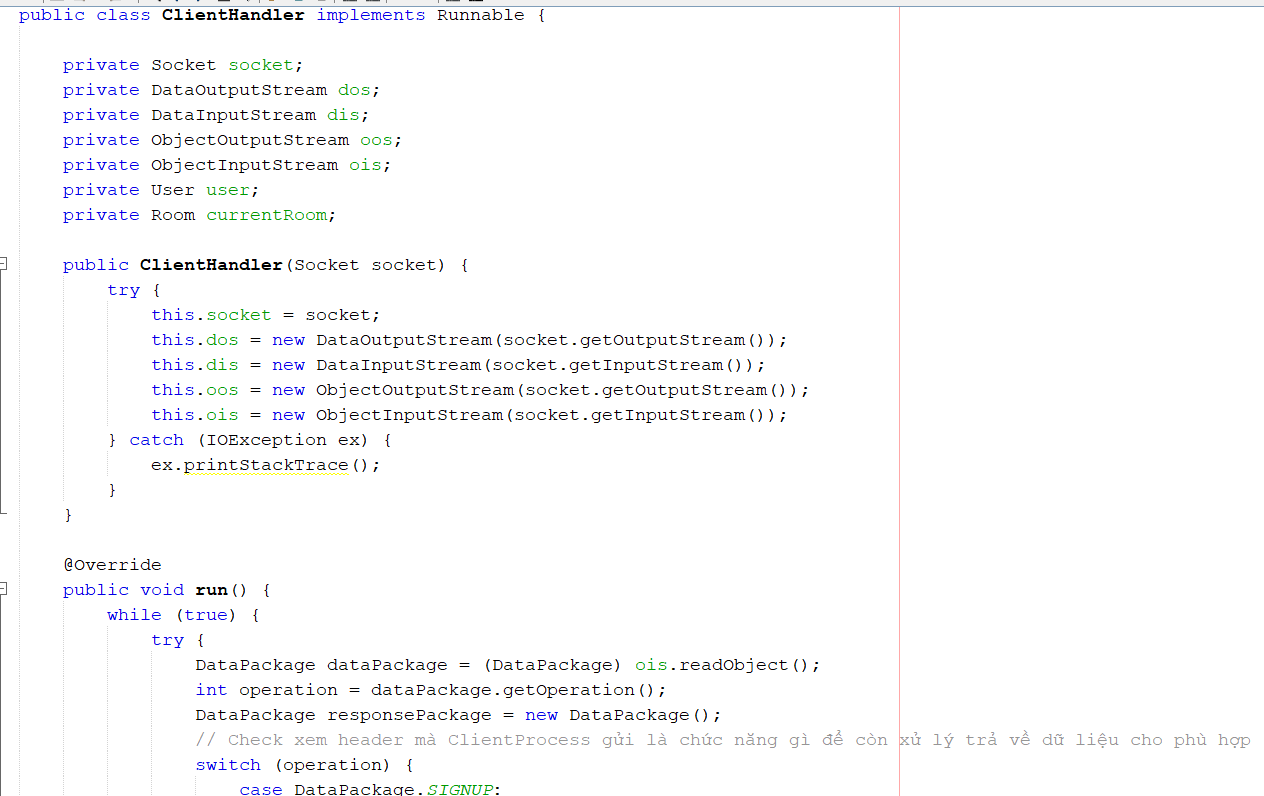
Chứa socket Chứa socket

**CLIENTS (Single Thread)** **SERVER(Multithread)**

Đầu tiên bên phía server sẽ có 1 lớp là ServerProcess, lớp này sẽ có 1 thuộc tính ServerSocket, ở vòng while(true) sẽ liên tục chấp nhận các kết nối từ phía các client gửi tới, mỗi lần chấp nhận kết nối, ta sẽ khởi tạo 1 luồng ClientHandler bên phía server, nhét socket lấy được vào hàm khởi tạo, mỗi ClientHandler này sẽ làm nhiệm vụ giao tiếp với 1 tiến trình ClientProcess bên phía Client



Mỗi ClientHandler bên phía server sẽ có 1 socket riêng, từ đó lấy được các luồng đọc ghi từ socket đó để giao tiếp với 1 ClientProcess bên phía client:



Nếu như bên phía server là cơ chế đa luồng, có nhiều tiến trình ClientHandler chạy cùng lúc, thì bên phía client chỉ có 1 tiến trình ClientProcess chạy duy nhất:

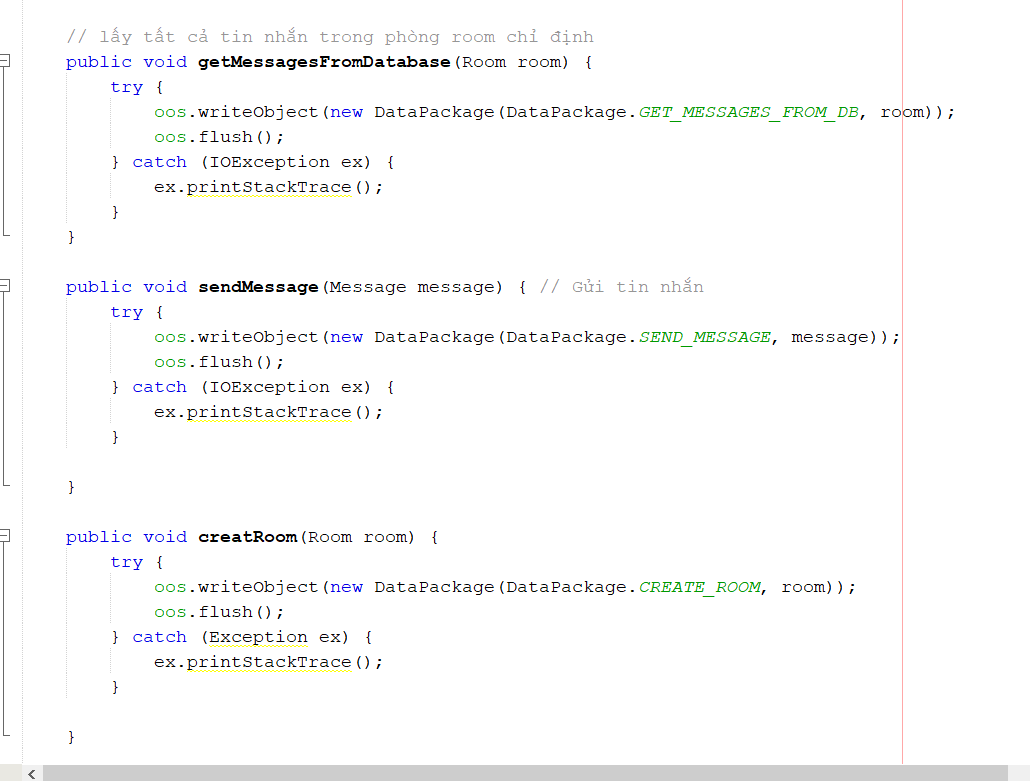


Client và server sẽ trao đổi thông tin dữ liệu thông qua đối tượng DataPackage, với trường data là dữ liệu gửi đi có kiểu là Object, operation là chức năng cần thực hiện với data đó, để kiểu int, được xác định bởi các biến static int trong lớp này, và statusMessage để báo xem gói tin nhận về có bị lỗi hay đã ok rồi:



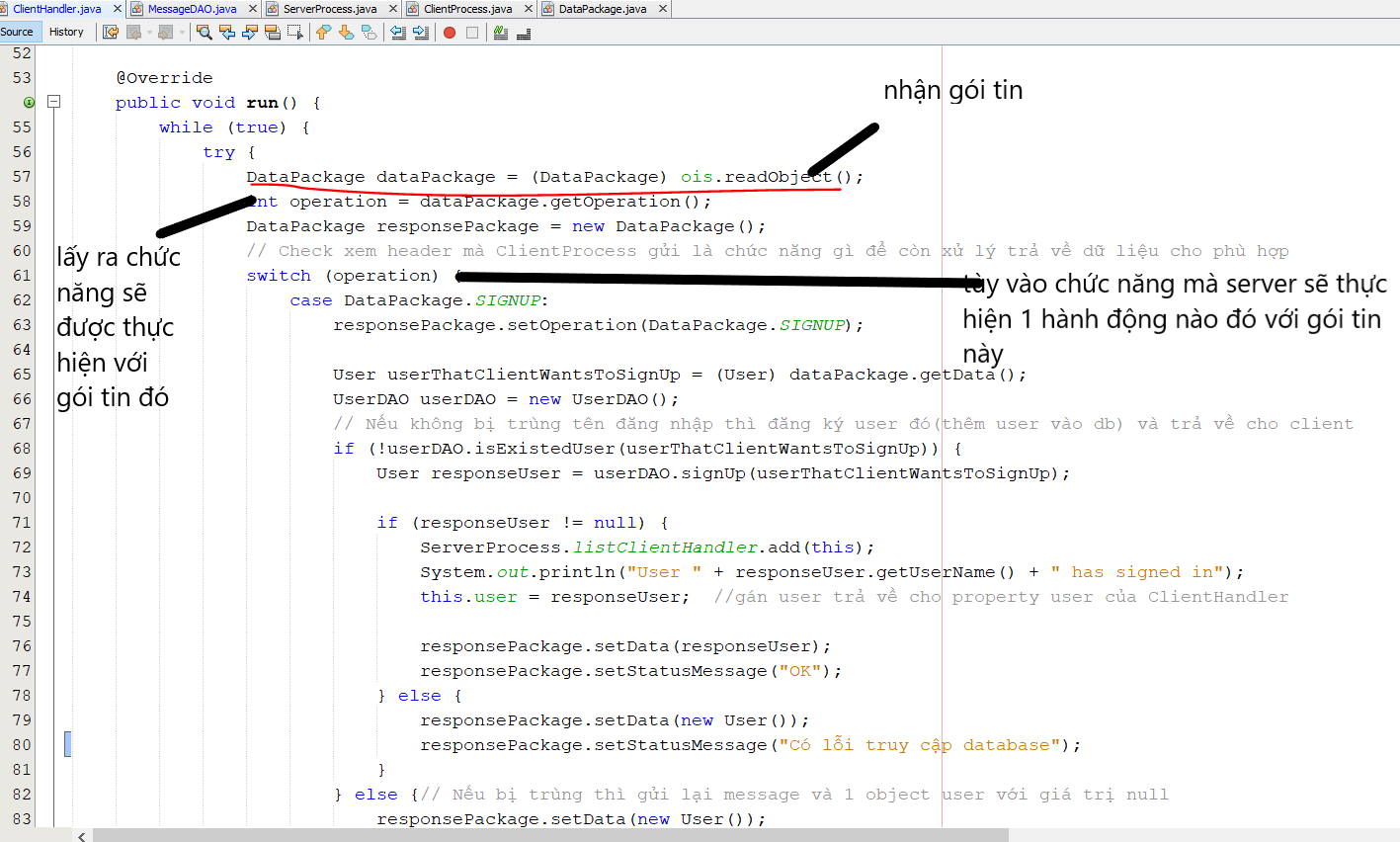
Đó là khung trao đổi dữ liệu, giờ nói đến cách client và server giao tiếp với nhau:

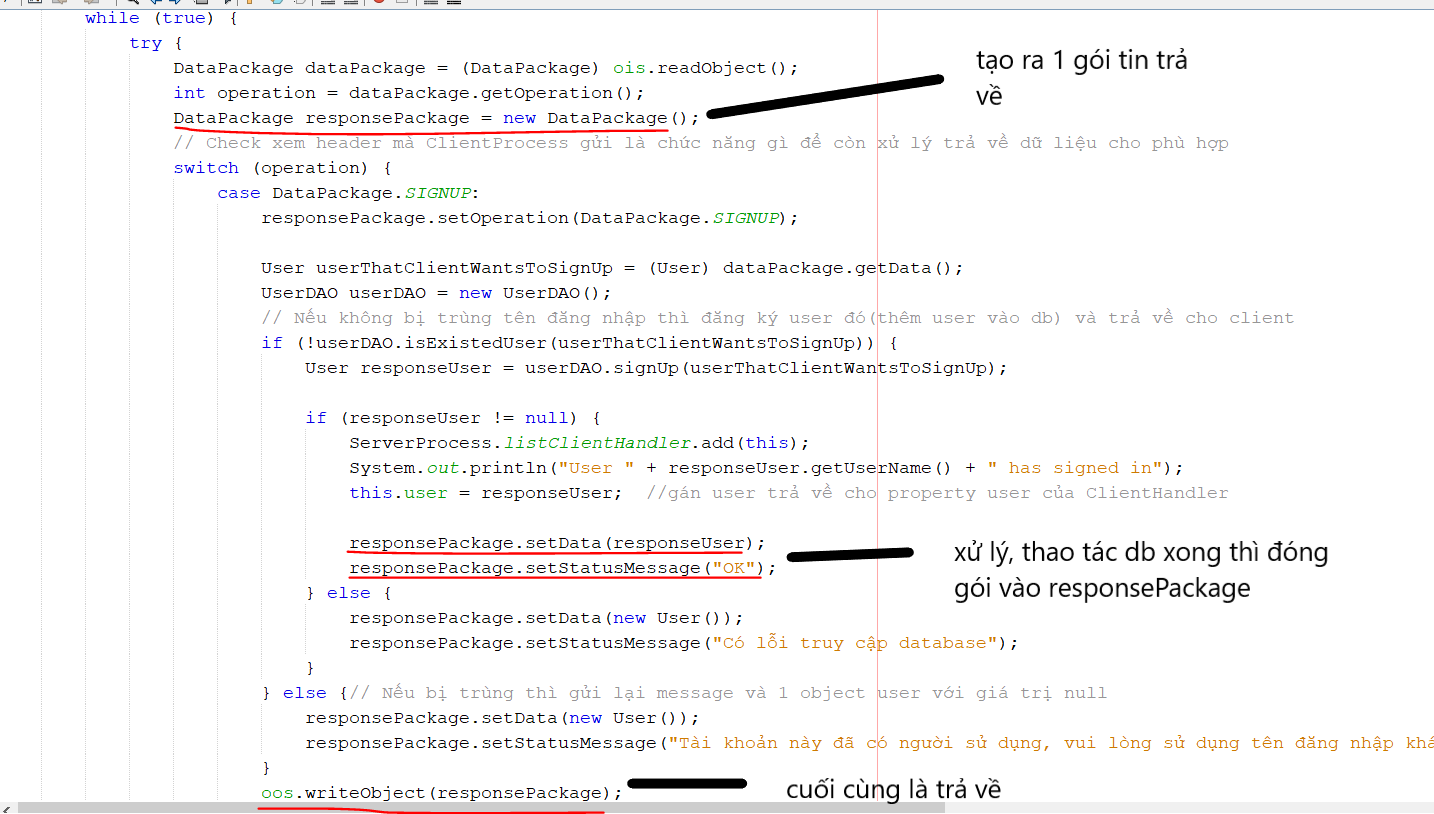
Đầu tiên trong lớp ClientProcess bên phía client, ta có các hàm để gửi yêu cầu tới server, các hàm này sẽ được giao diện bên phía swing gọi khi người dùng thực hiện một event nào đó:



Khi gói tin được gửi tới phía server, ở bên server, tại lớp ClientHandler chứa luồng while(true) liên tục nhận các gói tin DataPackage yêu cầu từ bên máy khách và thực hiện xử lý các chức năng tùy vào thuộc tính operation của gói tin đó

Ví dụ với chức năng SignUP đăng ký





Ở bên phía ClientProcess bên phía máy khách, trong hàm khởi tạo của nó, ta cũng tạo ra 1 luồng liên tục nhận các gói tin DataPackge trả về từ phía Server, và tùy vào trường operation của gói tin, ClientProcess sẽ quyết định làm gì với dữ liệu data trong gói tin đó, có thể là dùng dữ liệu đó để gọi UI bên swing thực hiện 1 chức năng nào đó chẳng hạn:



1. Các bước gửi nhận tin nhắn trong 1 phòng chat đơn hay group chat: