

# HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

# ST

ONE LOVE. ONE FUTURE.



ĐẠI HỌC  
BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# Xây dựng ứng dụng chat room trực tuyến kết hợp chia sẻ file

Nhóm 11

Môn học: Thực hành lập trình mạng

ONE LOVE. ONE FUTURE.

- Mục tiêu chính là xây dựng một hệ thống cho phép nhiều người dùng:
- **Tham gia vào các phòng chat (chat rooms):** Nơi họ có thể gửi và nhận tin nhắn văn bản (text) ngay lập tức (real-time).
- **Chia sẻ tệp tin:** Cho phép người dùng tải lên (upload) và gửi các tệp tin (như hình ảnh, tài liệu, file .zip...) cho những người khác trong cùng một phòng chat.
- Ứng dụng này phát triển dưới dạng ứng dụng máy tính (desktop).

## Mục tiêu và phạm vi của dự án

- Hệ thống sẽ được xây dựng bằng ngôn ngữ C++, áp dụng mô hình client-server với các chức năng chính:
- **Quản lý User:** Đăng ký, Đăng nhập.
- **Chat Room:** Tạo phòng, Tham gia phòng, Thoát phòng, Gửi tin nhắn chung (broadcast) trong phòng, Gửi yêu cầu kết bạn.
- **Chia sẻ File:** Tải file lên (upload) một phòng, Tải file về (download) từ phòng.
- **Server:** Xử lý đồng thời nhiều Client (đa luồng - pthread), quản lý trạng thái online/offline của người dùng, phân phối tin nhắn và điều phối việc truyền file.
- **Client:** Ứng dụng giao diện.

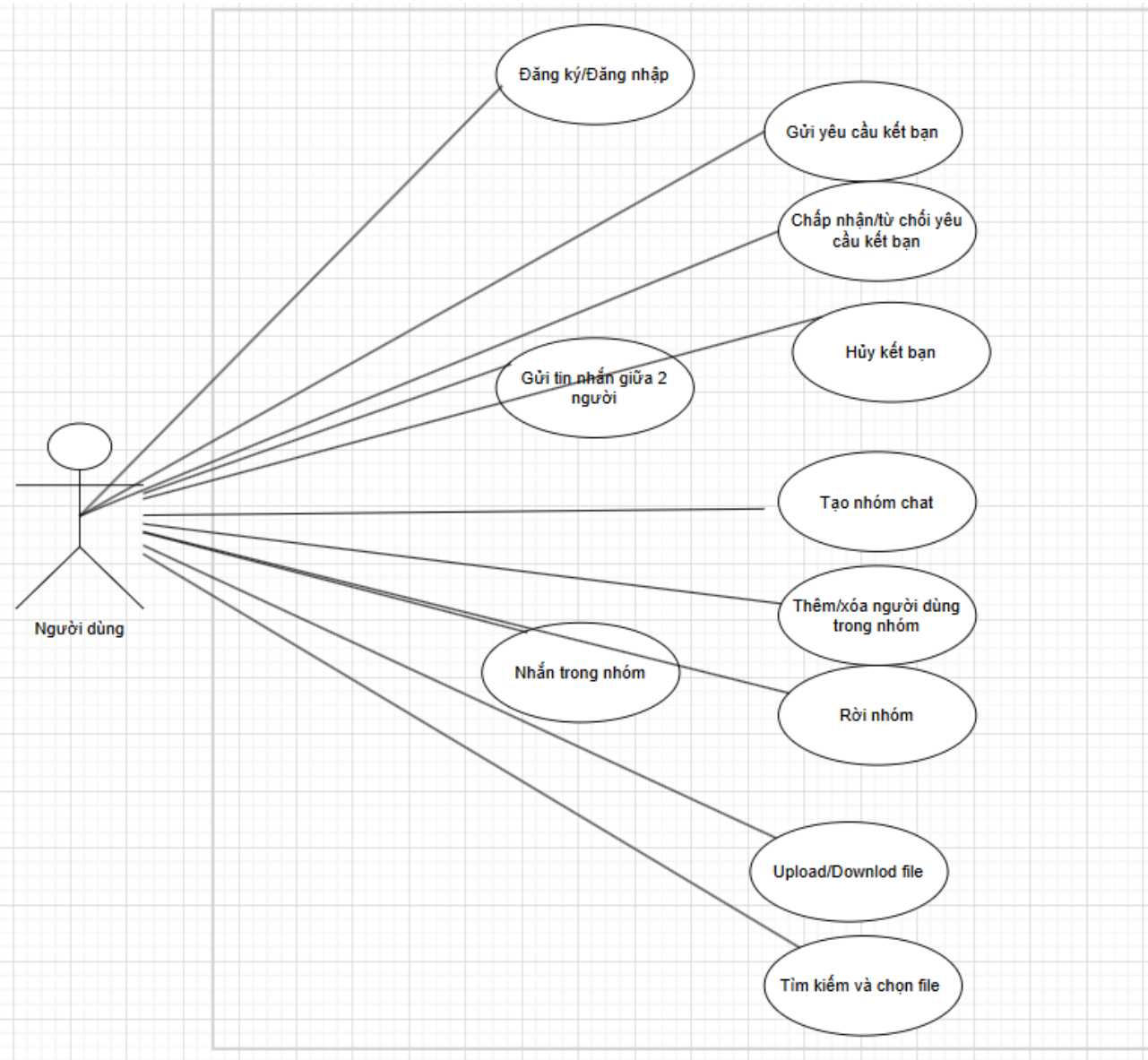
## Khởi động và xác thực

- Người dùng mở ứng dụng Qt (Client).
- Client tự động kết nối đến Server qua C++ Socket.
- Giao diện Đăng nhập / Đăng ký hiện ra.
- Người dùng nhập thông tin. Client gửi yêu cầu đến Server.
- Server xác thực với CSDL, trả về thông báo thành công (cùng với token phiên, danh sách bạn bè, tin nhắn offline) hoặc thất bại.

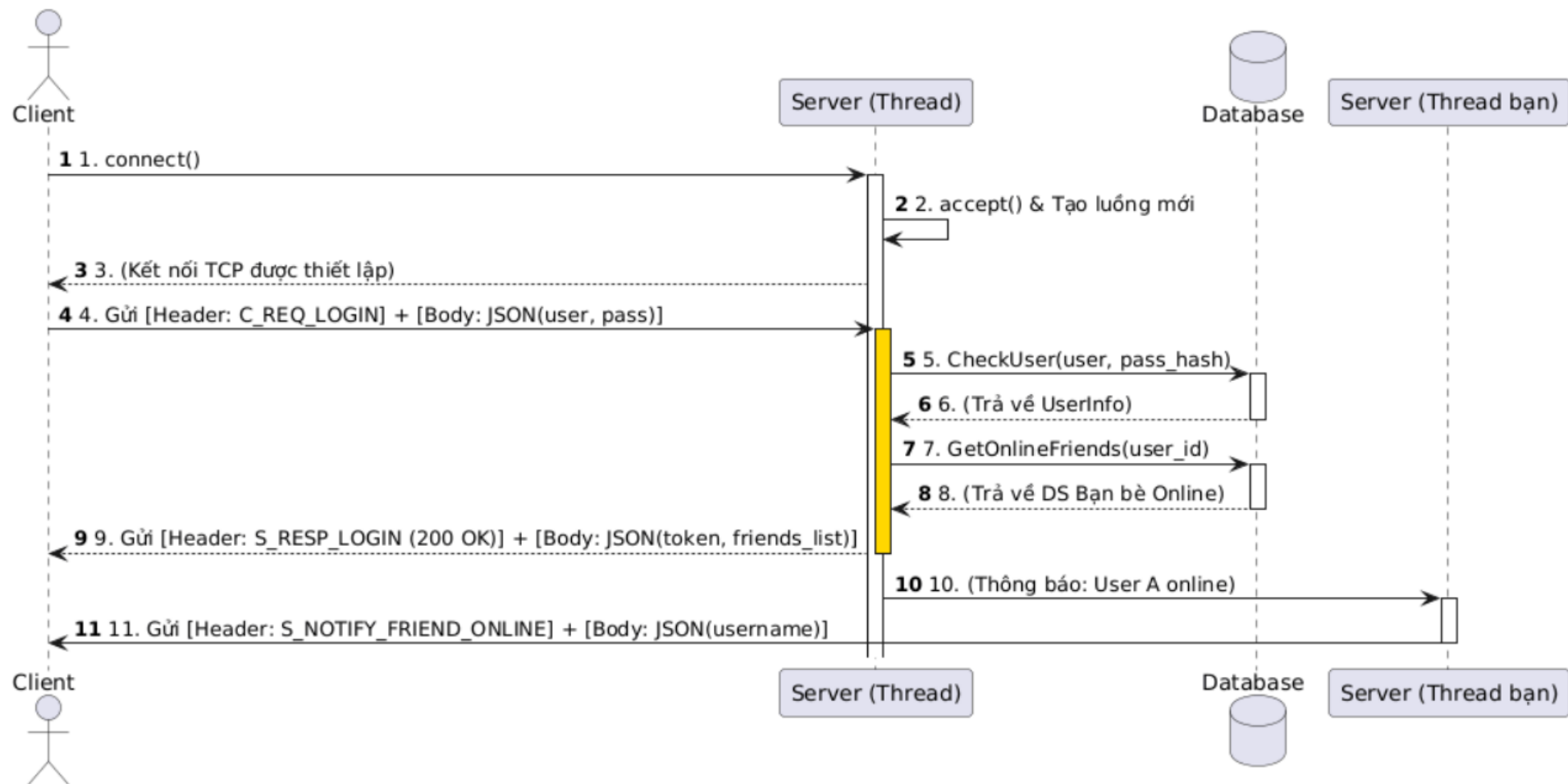
## Giao diện Chính (Main Lobby):

- Danh sách Bạn bè: Hiển thị tên bạn bè và trạng thái (Online, Offline, InGame).
- Danh sách Nhóm Chat: Các nhóm người dùng đã tham gia.

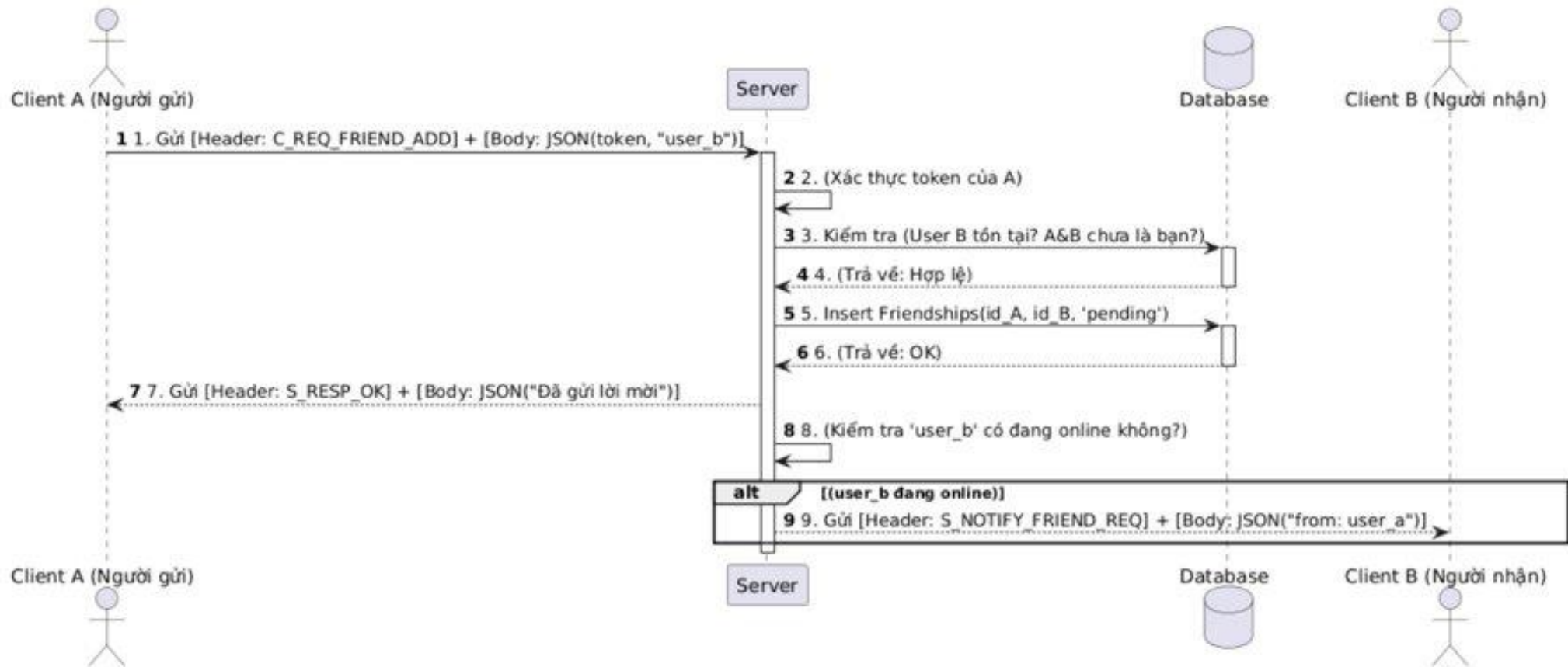
# Biểu đồ Use Case



# Biểu đồ trình tự: Luồng đăng nhập và thông báo người dùng online

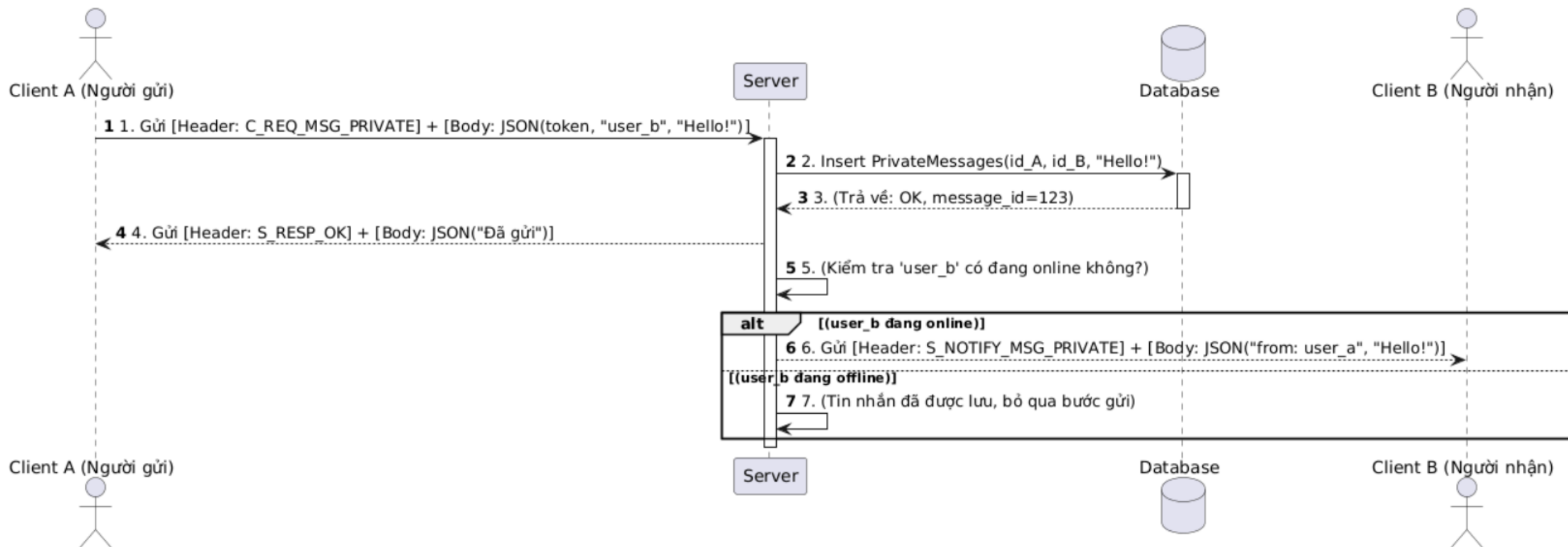


# Biểu đồ trình tự: Luồng gửi lời mời kết bạn

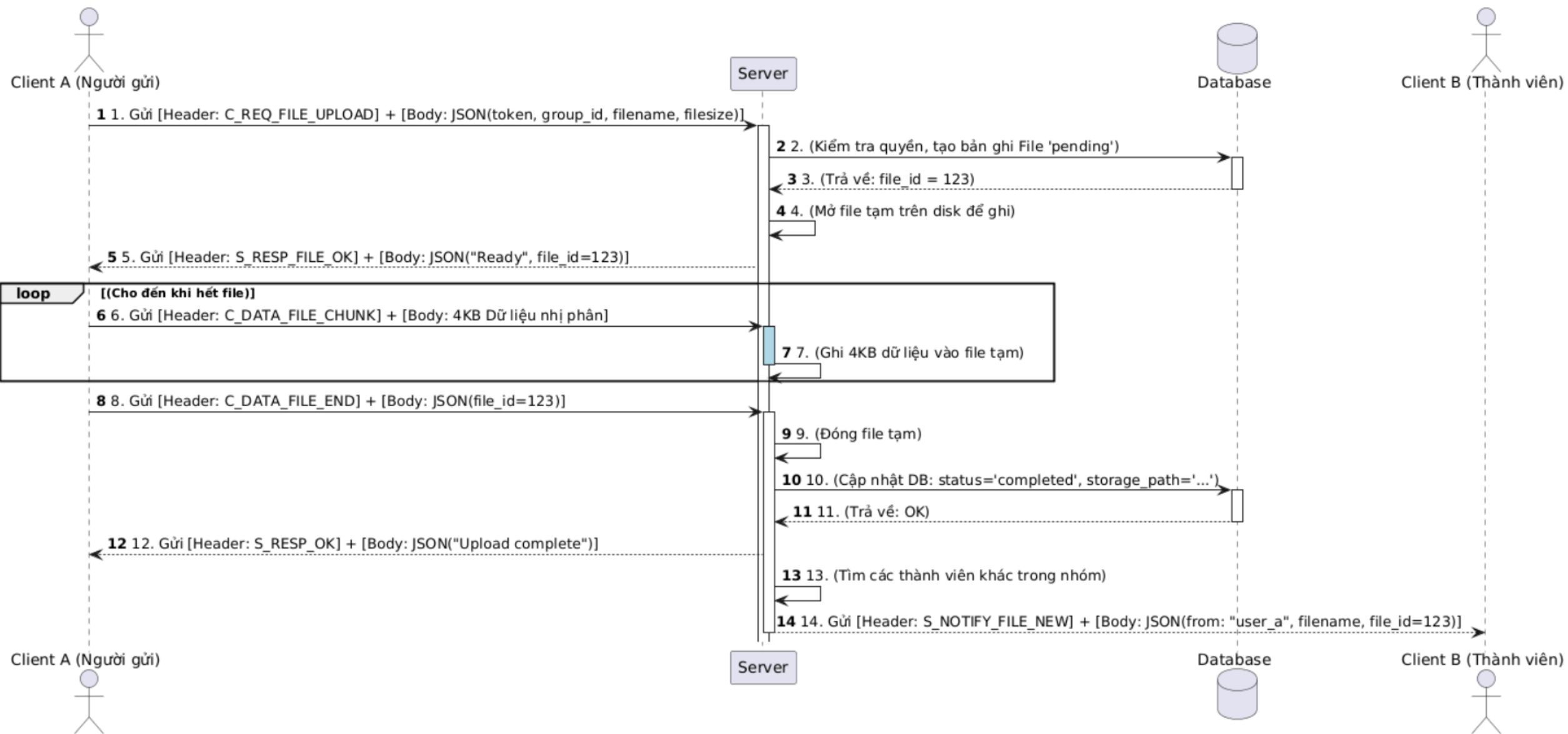




# Biểu đồ trình tự: Gửi tin nhắn 1-1



# Biểu đồ trình tự: Tải file lên nhóm



## Thiết kế giao thức tầng ứng dụng

- Kiến trúc sử dụng: kiến trúc client server
- Cấu trúc gói tin: giao thức tuân theo cấu trúc **Header + Body**
  - **Header:** Có độ dài cố định, chứa thông tin về loại lệnh (VD: 1 = Login, 2 = SEND\_MESSAGES,...), mã trạng thái (VD: 200 = OK, 404 = NOT\_FOUND,..), thời gian gửi, độ dài của phần Body
  - **Body:** Có độ dài thay đổi được chỉ định trong Header. Sử dụng JSON dưới dạng chuỗi để làm nội dung Body.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm xác thực (1xx)				
101	C_REQ_LOGIN	C → S	{"username": "u1", "pass_hash": "..."} 	Client yêu cầu đăng nhập.
102	S_RESP_LOGIN	S → C	(200 OK) {"token": "...", "friends_online": [...]}	Server phản hồi đăng nhập (kèm DS bạn bè online).
103	C_REQ_REGISTER	C → S	{"username": "u1", "pass_hash": "..."} 	Client yêu cầu đăng ký.
104	S_RESP_REGISTER	S → C	(201 OK) {"message": "Register OK"}	Server xác nhận đăng ký thành công.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm trạng thái (2xx)				
201	S_NOTIFY_FRIEND_ONLINE	S → C	{"username": "u4"}	Server báo cho bạn biết u4 vừa online.
202	S_NOTIFY_FRIEND_OFFLINE	S → C	{"username": "u2"}	Server báo cho bạn biết u2 vừa offline.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm quản lý bạn bè (3xx)				
101	C_REQ_FRIEND_ADD	C → S	{"token": "...", "target_username": "u5"}	Client gửi lời mời kết bạn (FC05).
102	S_NOTIFY_FRIEND_REQ	S → C	{"from_username": "u1"}	Server báo cho u5 biết u1 muốn kết bạn.
103	C_RESP_FRIEND_REQ	C → S	{"token": "...", "from_username": "u1", "accept": true}	Client (u5) phản hồi lời mời kết bạn.
104	S_NOTIFY_FRIEND_ACCEPT	S → C	{"username": "u5"}	Server báo cho u1 biết u5 đã đồng ý.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm chat 1-1 (4xx)				
401	C_REQ_MSG_PRIVATE	C → S	{"token": "...", "target_username": "u2", "message": "Hi"}	Client gửi tin nhắn 1-1 (FC03).
402	S_NOTIFY_MSG_PRIVATE	S → C	{"from_username": "u1", "message": "Hi"}	Server chuyển tiếp tin nhắn 1-1 cho người nhận.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Chat nhóm (5xx)				
501	C_REQ_GROUP_CREATE	C → S	{"token": "...", "group_name": "Class 101"}	Client yêu cầu tạo nhóm mới.
502	S_RESP_GROUP_CREATE	S → C	(201 OK) {"group_id": 1, "group_name": "..."}	Server xác nhận tạo nhóm.
503	C_REQ_GROUP_JOIN	C → S	{"token": "...", "group_id": 1}	Client yêu cầu tham gia nhóm (FC07).
504	S_NOTIFY_GROUP_JOIN	S → C	{"username": "u1", "group_id": 1}	Server báo các thành viên (và người mới) là u1 đã vào nhóm.
505	C_REQ_GROUP_LEAVE	C → S	{"token": "...", "group_id": 1}	Client yêu cầu rời nhóm (FC07).
506	S_NOTIFY_GROUP_LEAVE	S → C	{"username": "u1", "group_id": 1}	Server báo các thành viên là u1 đã rời nhóm.
507	C_REQ_MSG_GROUP	C → S	{"token": "...", "group_id": 1, "message": "Hello all"}	Client gửi tin nhắn vào nhóm (FC08).
508	S_NOTIFY_MSG_GROUP	S → C	{"from_username": "u1", "group_id": 1, "message": "..."}	Server broadcast tin nhắn cho các thành viên.



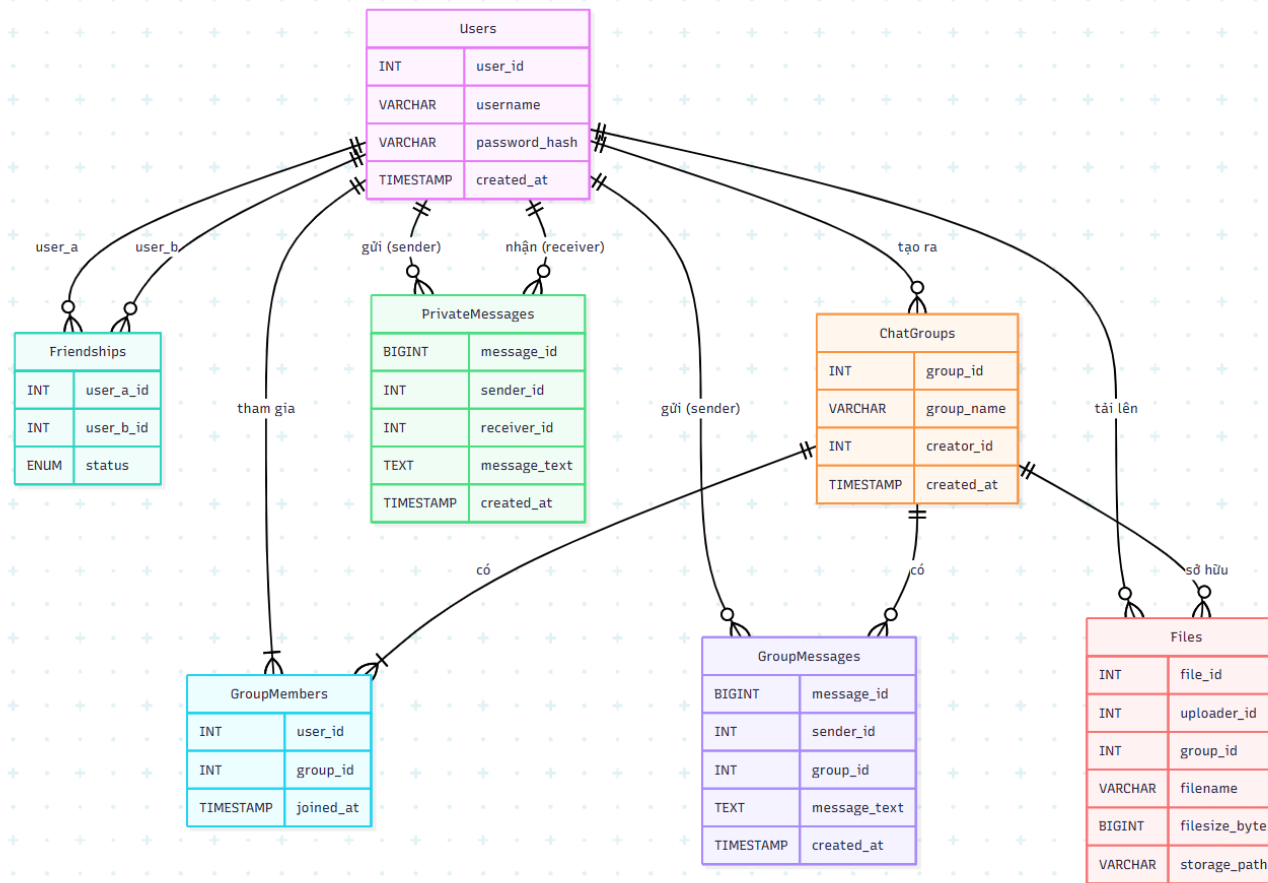
# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm chia sẻ file (6xx)				
601	C_REQ_FILE_UPLOAD	C → S	{"token":"...", "group_id":1, "filename":"a.zip", "filesize":10240}	Client "xin phép" được tải file lên.
602	S_RESP_FILE_OK	S → C	(200 OK) {"file_id": 123, "message": "Ready"}	Server đồng ý, trả về file_id để Client bắt đầu gửi.
604	C_DATA_FILE_END	C → S	{"token": "...", "file_id": 123}	Client báo đã gửi xong chunk cuối cùng.
605	S_NOTIFY_FILE_NEW	S → C	{"from_user":"u1", "group_id":1, "filename":"a.zip", "file_id":123}	Server báo các thành viên nhóm có file mới.
606	C_REQ_FILE_DOWNLOAD	C → S	{"token": "...", "file_id": 123}	Client yêu cầu được tải file.
607	S_RESP_FILE_START	S → C	(200 OK) {"filename":"a.zip", "filesize":10240}	Server đồng ý, báo Client chuẩn bị nhận file.
609	S_DATA_FILE_END	S → C	{"file_id": 123, "message": "Download complete"}	Server báo đã gửi xong chunk cuối cùng.

# Thiết kế giao thức tầng ứng dụng

ID	Tên lệnh	Hướng	JSON Body (Ví dụ)	Mô tả
Nhóm các gói tin có phần Body là <b>Dữ liệu nhị phân thô</b>				
1603	C_DATA_FILE_CHUNK	C → S	[Dữ liệu nhị phân (vd: 4KB)]	Client gửi một khối (chunk) của file.
1608	S_DATA_FILE_CHUNK	S → C	[Dữ liệu nhị phân (vd: 4KB)]	Server gửi một khối (chunk) của file.

# Thiết kế cơ sở dữ liệu



## Phạm Tiến Thọ:

- Xử lý luồng (Stream Handling): 1 điểm
- Thiết kế giao diện đồ họa người dùng (GUI): 3 điểm
- Tìm kiếm và lựa chọn tệp tải xuống: 3 điểm
- Ghi log hoạt động: 1 điểm
- Gửi tin nhắn ngoại tuyến: 1 điểm
- **Tổng: 9 điểm**

## Nguyễn Mạnh Tuấn:

- Gửi và nhận tin nhắn giữa hai người dùng: 1 điểm
- Ngắt kết nối: 1 điểm
- Tạo nhóm chat: 1 điểm
- Thêm người dùng vào nhóm chat: 1 điểm
- Xóa người dùng khỏi nhóm chat: 1 điểm
- Rời khỏi nhóm chat: 1 điểm
- Gửi và nhận tin nhắn trong nhóm chat: 1 điểm
- Tải lên/Tải xuống tệp: 2 điểm
- **Tổng: 10 điểm**

## Nguyễn Đức Tín:

- Cài đặt cơ chế I/O cho socket trên máy chủ: 2 điểm
- Đăng kí và quản lý tài khoản: 2 điểm
- Đăng nhập và quản lý phiên làm việc: 2 điểm
- Gửi yêu cầu kết bạn: 1 điểm
- Chấp nhận/Từ chối yêu cầu kết bạn: 1 điểm
- Hủy kết bạn: 1 đi
- **Tổng: 9 điểm**

- **Tuần 7:** Thiết kế cơ sở dữ liệu, cài đặt môi trường cần thiết
- **Tuần 8:** Xây dựng Server đa luồng đơn giản và Client kết nối tới Server
- **Tuần 9:** Kết nối tới CSDL, thực hiện chức năng đăng nhập , đăng kí, tích hợp các giao thức
- **Tuần 10:** Hoàn thành luồng tạo phòng, vào phòng và gửi tin nhắn chung, Client có thể vào 1 phòng và nhìn thấy tin nhắn của nhau
- ...



**HUST**

**THANK YOU !**



[hust.edu.vn](http://hust.edu.vn)



[fb.com/dhbkhn](https://fb.com/dhbkhn)