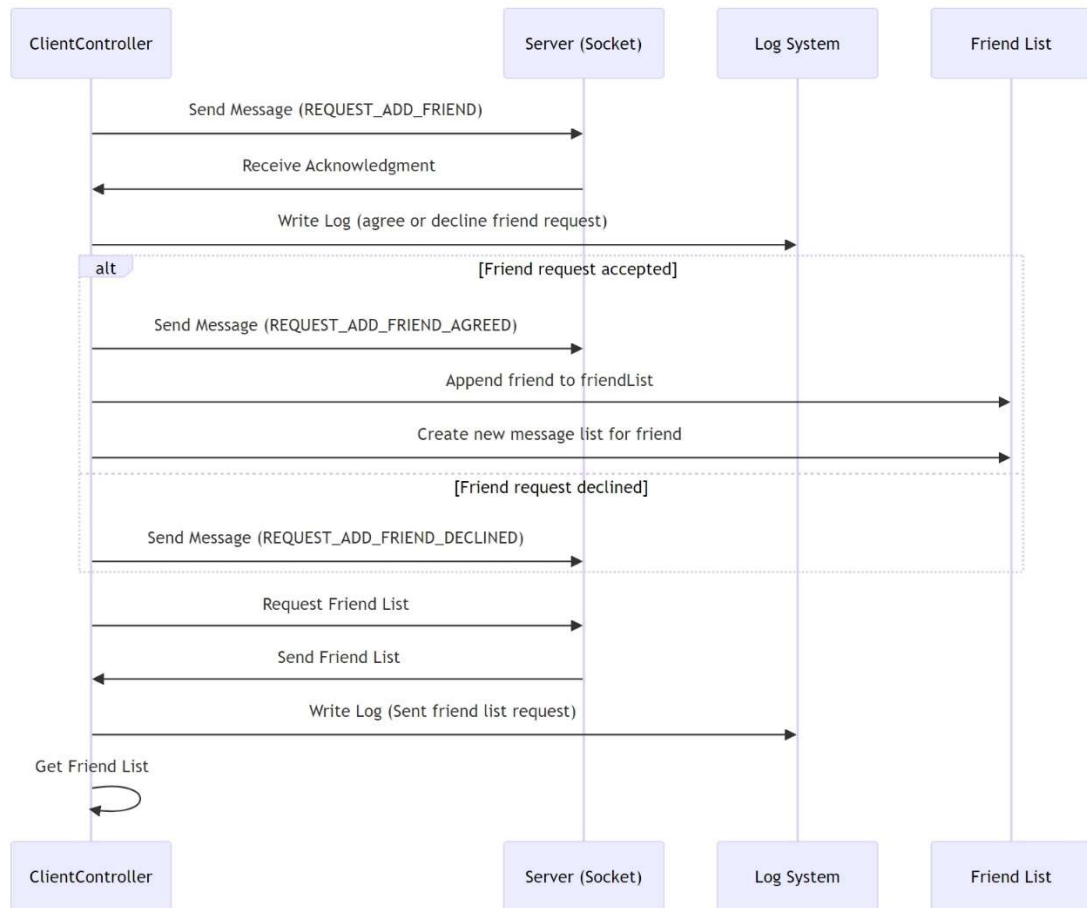
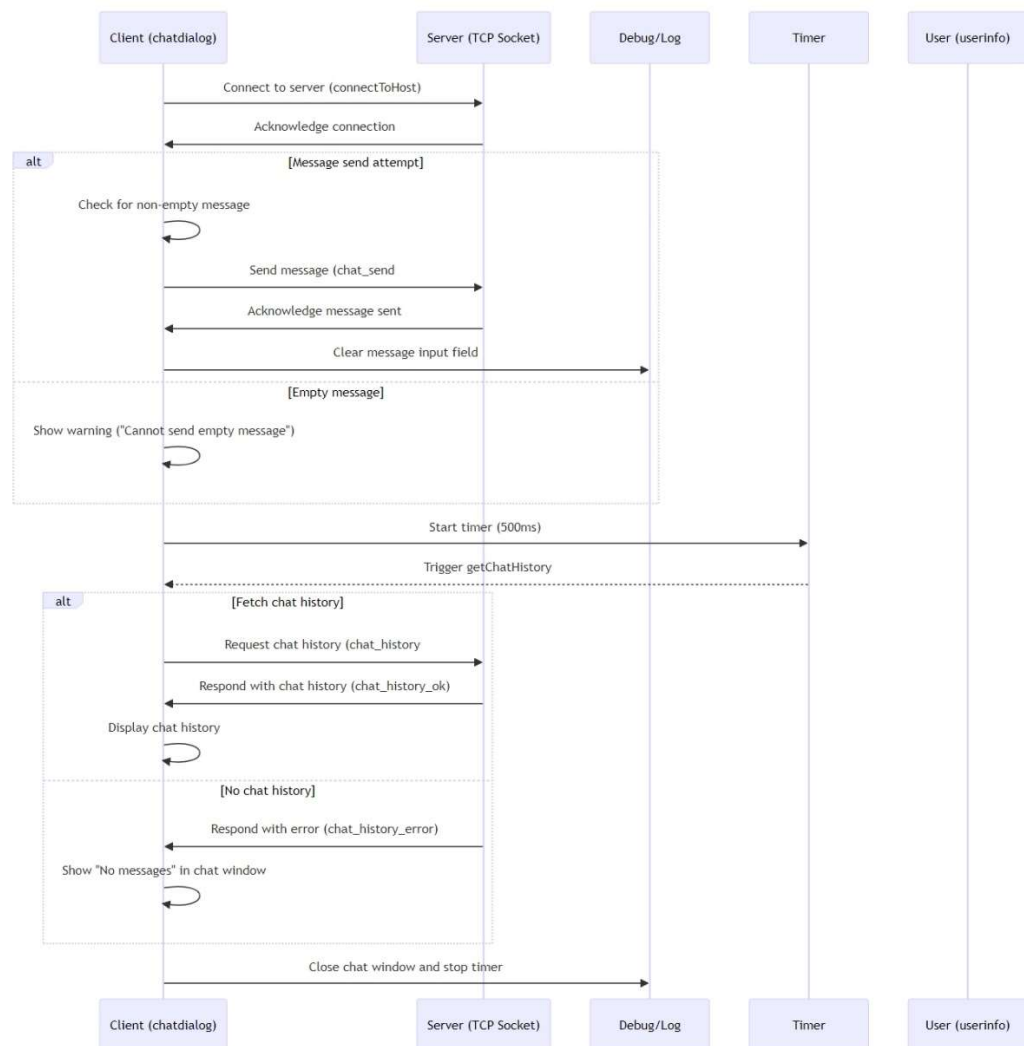


Task 2: IM System Conversation Design

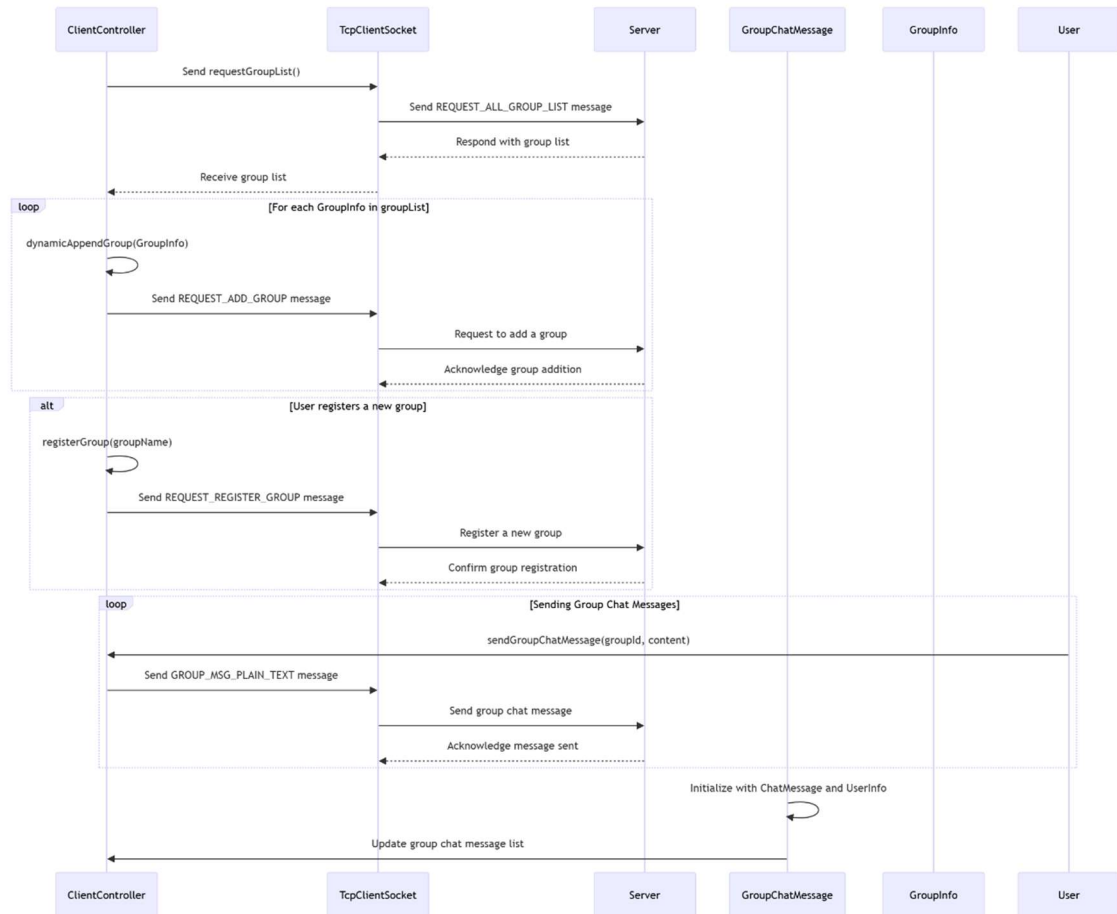
1. Add Friend



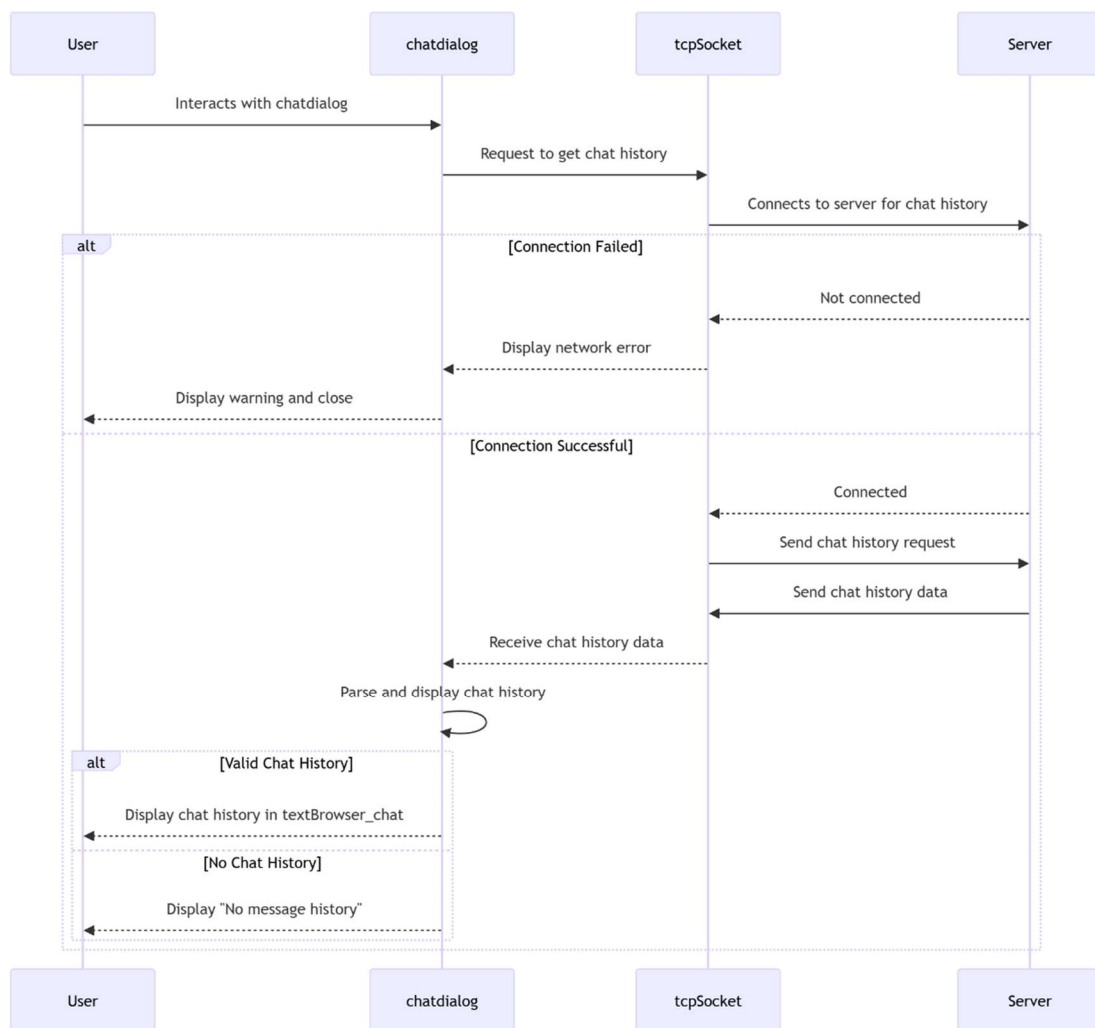
2. Private Chat



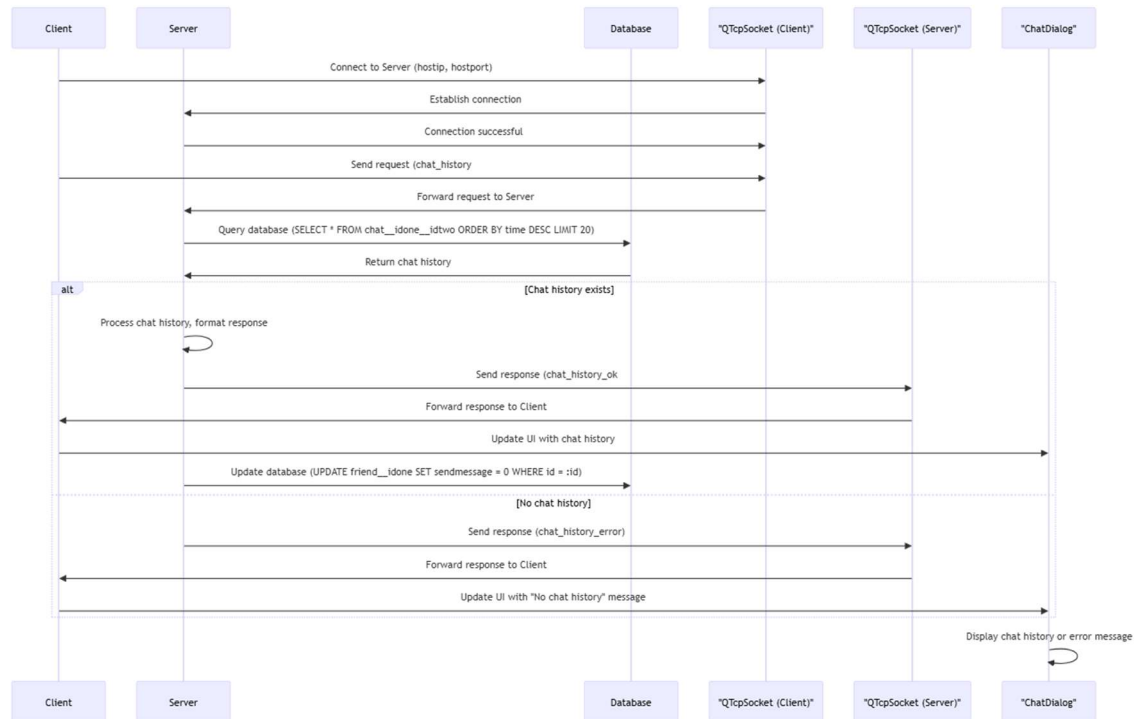
3. Group Chat



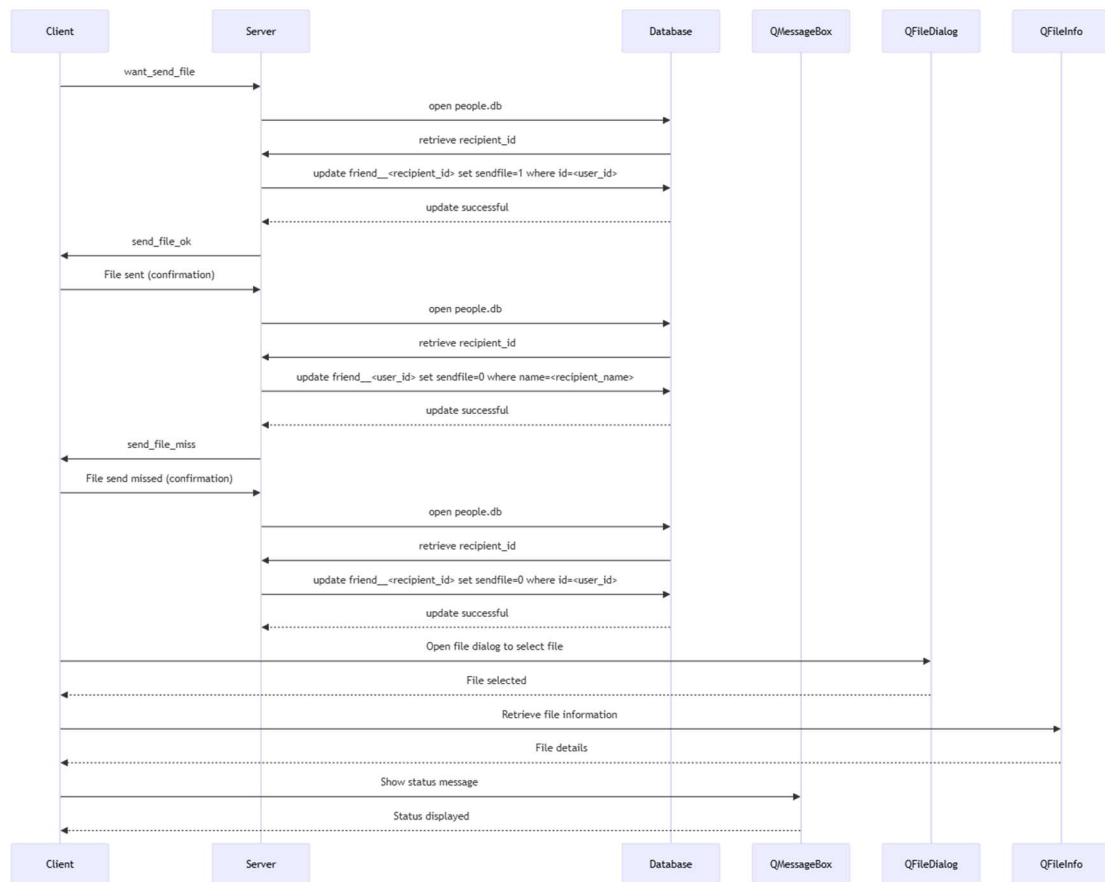
4. Retrieve Offline Messages



5. Chat History



6. Send & Receive Files



Task 3: Protocol Specification

1. Register

- Description: Register a new user account.

- Request Format:

```
class Register {  
  
    public:  
  
        std::string username; // The desired username  
  
        std::string password; // The desired password  
  
        std::string email;    // The user's email address  
  
};
```

- Response Format:

```
class RegisterResponse {  
  
    public:  
  
        bool success;          // true if registration is successful, false otherwise  
  
        std::string message;    // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Registration successful."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Username already exists."  
  
}
```

2. Login

- Description: Login to the IM system.

- Request Format:

```
class Login {  
  
    public:  
  
        std::string username; // The user's username  
  
        std::string password; // The user's password  
  
};
```

- Response Format:

```
class LoginResponse {  
  
    public:  
  
        bool success;           // true if login is successful, false otherwise  
  
        std::string message;    // Success message or error description  
  
        std::string token;      // Session token if login is successful  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Login successful.",  
  
    "token": "abc123token"  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Invalid username or password."  
  
}
```


3. Logout

- Description: Logout from the IM system.

- Request Format:

```
class Logout {  
  
    public:  
  
        std::string token; // The session token of the user  
  
};
```

- Response Format:

```
class LogoutResponse {  
  
    public:  
  
        bool success; // true if logout is successful, false otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Logout successful."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Invalid session token."  
  
}
```

4. GetFriendList

- Description: Retrieve the list of friends for the logged-in user.

- Request Format:

```
class GetFriendList {  
  
    public:  
  
        std::string token; // The session token of the user  
  
};
```

- Response Format:

```
class GetFriendListResponse {  
  
    public:  
  
        bool success; // true if the list is retrieved successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
        std::vector<std::string> friends; // List of friend usernames  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Friend list retrieved successfully.",  
  
    "friends": ["friend1", "friend2", "friend3"]  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to retrieve friend list."  
  
}
```

5. GetGroupList

- Description: Retrieve the list of groups the user belongs to.

- Request Format:

```
class GetGroupList {  
  
    public:  
  
        std::string token; // The session token of the user  
  
};
```

- Response Format:

```
class GetGroupListResponse {  
  
    public:  
  
        bool success; // true if the list is retrieved successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
        std::vector<std::string> groups; // List of group names  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Group list retrieved successfully.",  
  
    "groups": ["group1", "group2", "group3"]  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to retrieve group list."  
  
}
```

6. GetGroupMembers

- Description: Retrieve the list of members in a specific group.

- Request Format:

```
class GetGroupMembers {  
  
public:  
  
    std::string token; // The session token of the user  
  
    std::string groupName; // The name of the group  
  
};
```

- Response Format:

```
class GetGroupMembersResponse {  
  
public:  
  
    bool success; // true if the list is retrieved successfully, false  
otherwise  
  
    std::string message; // Success message or error description  
  
    std::vector<std::string> members; // List of group members' usernames  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Group members retrieved successfully.",  
  
    "members": ["member1", "member2", "member3"]  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to retrieve group members."  
  
}
```

7. SendFriendMessage

- Description: Send a message to a specific friend.

- Request Format:

```
class SendFriendMessage {  
  
public:  
  
    std::string token;        // The session token of the user  
  
    std::string recipient;    // The username of the recipient friend  
  
    std::string message;      // The message content  
  
};
```

- Response Format:

```
class SendFriendMessageResponse {  
  
public:  
  
    bool success;             // true if the message is sent successfully, false  
otherwise  
  
    std::string message;      // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Message sent successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to send message."  
  
}
```

8. SendGroupMessage

- Description: Send a message to a specific group.

- Request Format:

```
class SendGroupMessage {  
  
public:  
  
    std::string token;      // The session token of the user  
  
    std::string groupName; // The name of the group  
  
    std::string message;    // The message content  
  
};
```

- Response Format:

```
class SendGroupMessageResponse {  
  
public:  
  
    bool success;          // true if the message is sent successfully, false  
otherwise  
  
    std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Message sent to group successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to send message to group."  
  
}
```

9. ReceiveFriendMessage

- Description: Receive a message from a specific friend.

- Request Format:

```
class ReceiveFriendMessage {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
};
```

- Response Format:

```
class ReceiveFriendMessageResponse {  
  
    public:  
  
        bool success;        // true if the message is received successfully, false  
otherwise  
  
        std::string sender; // The username of the sender  
  
        std::string message; // The message content  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "sender": "friend1",  
  
    "message": "Hello!"  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to receive message."  
  
}
```

10. ReceiveGroupMessage

- Description: Receive a message from a specific group.

- Request Format:

```
class ReceiveGroupMessage {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string groupName; // The name of the group  
  
};
```

- Response Format:

```
class ReceiveGroupMessageResponse {  
  
    public:  
  
        bool success  
;        // true if the message is received successfully, false otherwise  
  
        std::string sender; // The username of the sender  
  
        std::string message; // The message content  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "sender": "group_member1",  
  
    "message": "Hello, group!"  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to receive group message."  
  
}
```


11. AddFriend

- Description: Send a friend request to another user.

- Request Format:

```
class AddFriend {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string username; // The username of the person to be added as a friend  
  
};
```

- Response Format:

```
class AddFriendResponse {  
  
    public:  
  
        bool success;        // true if the friend request is sent successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Friend request sent successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to send friend request."  
  
}
```

12. CreateGroup

- Description: Create a new group.

- Request Format:

```
class CreateGroup {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string groupName; // The name of the group  
  
};
```

- Response Format:

```
class CreateGroupResponse {  
  
    public:  
  
        bool success;        // true if the group is created successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Group created successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to create group."  
  
}
```

13. RequestAddToGroup

- Description: Request to join a group.

- Request Format:

```
class RequestAddToGroup {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string groupName; // The name of the group  
  
};
```

- Response Format:

```
class RequestAddToGroupResponse {  
  
    public:  
  
        bool success;        // true if the request is sent successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Request to join group sent successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to send request to join group."  
  
}
```

14. AcceptGroupAdd

- Description: Accept a request to join a group.

- Request Format:

```
class AcceptGroupAdd {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string groupName; // The name of the group  
  
};
```

- Response Format:

```
class AcceptGroupAddResponse {  
  
    public:  
  
        bool success;        // true if the request is accepted successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Request to join group accepted."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to accept request to join group."  
  
}
```

15. Heartbeat

- Description: Send a heartbeat signal to keep the connection alive.

- Request Format:

```
class Heartbeat {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
};
```

- Response Format:

```
class HeartbeatResponse {  
  
    public:  
  
        bool success;        // true if the heartbeat is acknowledged, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Heartbeat acknowledged."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to acknowledge heartbeat."  
  
}
```

16. FileTransfer

- Description: Transfer a file to another user or group.

- Request Format:

```
class FileTransfer {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string recipient; // The username of the recipient or group name  
  
        std::string fileName; // The name of the file  
  
        std::vector<char> fileData; // The file data  
  
};
```

- Response Format:

```
class FileTransferResponse {  
  
    public:  
  
        bool success;    // true if the file is transferred successfully, false  
otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "File transferred successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to transfer file."  
  
}
```

17. AudioVideoCall

- Description: Initiate an audio or video call.

- Request Format:

```
class AudioVideoCall {  
  
    public:  
  
        std::string token;    // The session token of the user  
  
        std::string recipient; // The username of the recipient or group name  
  
        bool isVideoCall;    // true for video call, false for audio call  
  
};
```

- Response Format:

```
class AudioVideoCallResponse {  
  
    public:  
  
        bool success;        // true if the call is initiated successfully, false otherwise  
  
        std::string message; // Success message or error description  
  
};
```

- Success Response Example:

```
{  
  
    "success": true,  
  
    "message": "Call initiated successfully."  
  
}
```

- Failure Response Example:

```
{  
  
    "success": false,  
  
    "message": "Failed to initiate call."  
  
}
```