VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY

Computer Networks (CO3094) - CC02

Assignment for

Chat Application

Advisor: Nguyễn Phương Duy
Student: Lê Khánh Duy - 2052003.
Trần Minh Thức - 2053486.
Nguyễn Quốc Minh Thư - 2052736.
Nguyễn Hoàng Anh Thư - 2053478.

HO CHI MINH CITY, November 2022

Contents

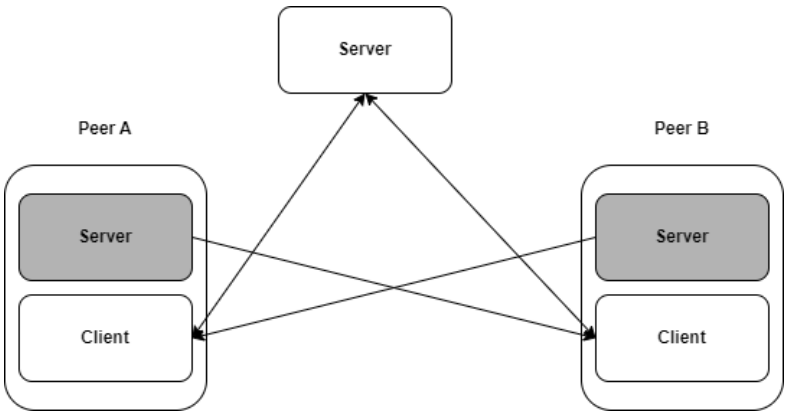
Member list & Workload

No.	Full name	Student ID	Problems	Percentage of work
1	Lê Khánh Duy	2052003	- Job -	..%
2	Trần Minh Thức	2053486	-Job -	..%
3	Nguyễn Hoàng Anh Thư	2053478	- Job. -	..%
4	Nguyễn Quốc Minh Thư	2052736	- Job -	..%

1 Phase 1

1.1 General Architecture

Description: In this application, there will be ***a center server*** that connect to ***some clients*** which also be in ***the P2P relationship with each other***. The connection of both Client-Server and Peer to Peer will use ***TCP/IP protocol***. Username will be used to differentiate those client.



Hình 1: Architecture

1.2 Functions

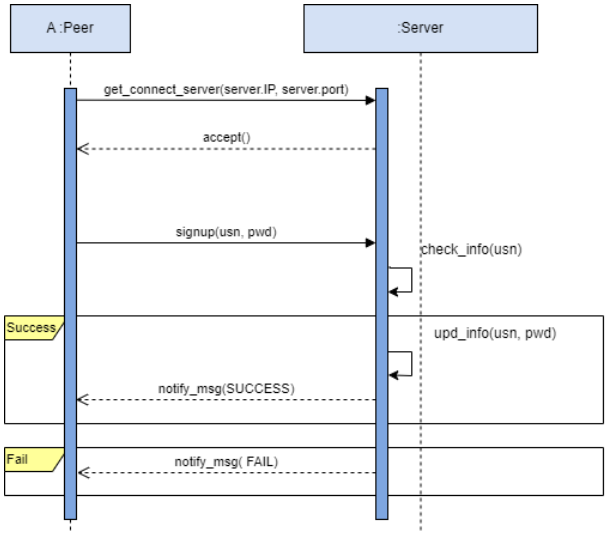
Requirement: Define specific functions of the chat application

- **Server - Client:**
  - Server:
    1. Receive request sign up, login from client.
    2. Manage the list of user being online.
    3. Return the friend list of user when receive the request from client.
  - Client:
    1. Send request sign up, login to server.
    2. Send add friend request.
    3. Send get friend list request.
    4. Create a room chat with a peer in the friend list received from server and through server.
- **Peer to Peer:**
  1. Send message to peer.
  2. Send file to peer.

1.3 Protocols

Requirement: Define the communication protocols used for each function

1.3.1 Sign up

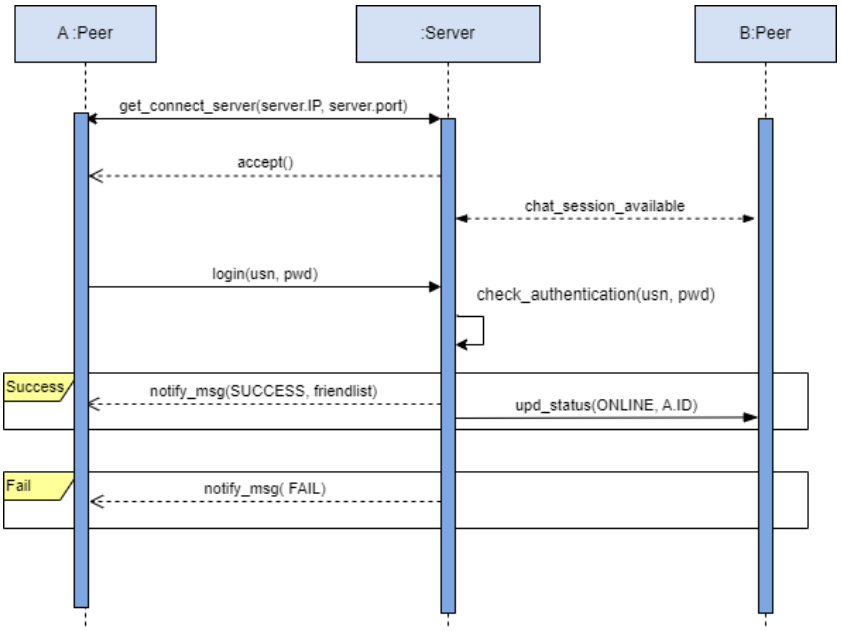


Hình 2: Sign up

♣ Description

Use Case Name	Sign up
Description	A peer want to create the account in server
Actor(s)	Peer A, Server
Trigger	Peer A sent the signup(usn, pwd) to the Server.
Precondition(s)	Peer A must connect to server and server accepted
Postcondition(s)	Peer A create a account in server successfully
Main flow	1. Peer A send the request to connect server through get_connect_sever (server.IP,server.port) and wait for accepting 2. Server send the respond accept to the server. 3. Peer A send the signup(usn,pwd) to server to create the account. 4. Server with check the info to consider either usn is unique or not through check_info(usn). 5. Username is unique, so server update info of user (username and password) - save it in the database (upd_info(usn,pwd)) and then send the notify message success to peer A (notify_msg(SUCCESS)).
Exception flow	5a. Username is not unique, so server send the notify message fail to peer A (notify_msg(FAIL) *Go back to step 3.*

1.3.2 Log in

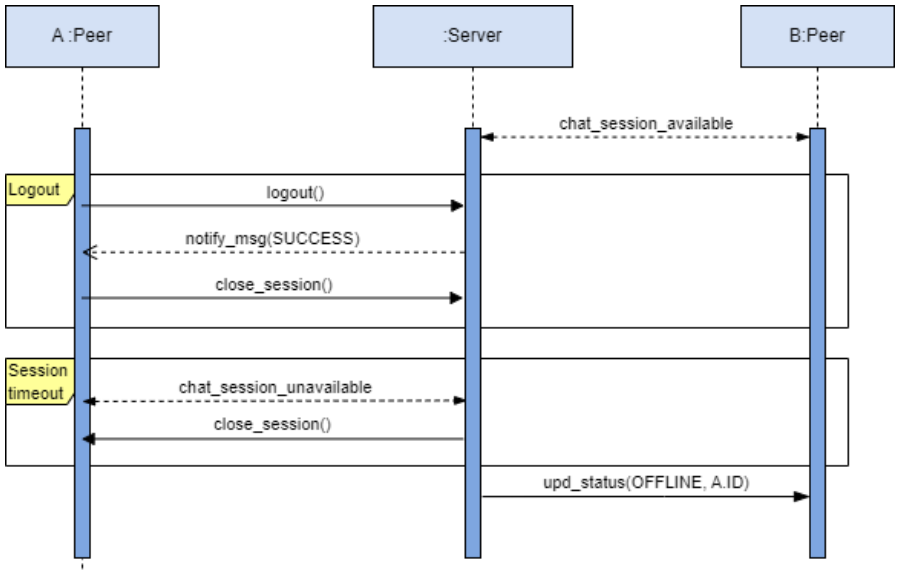


Hình 3: Log in

♣ Description

Use Case Name	**Log in**
Description	A peer want to log in account in server
Actor(s)	Peer A, Server, Peer B
Trigger	Peer A sent the login(usn, pwd) to the Server.
Precondition(s)	Peer A must connect to server and server accepted
Postcondition(s)	Peer A receive the notify message success and can send message to friend.
Main flow	1. Peer A send the request to connect server through get_connect_sever (server.IP,server.port) and wait for accepting 2. Server send the respond accept to the server and also alert to other peer that can chat to peer A. 3. Peer A send the login(usn,pwd) to server to log in server. 4. Server with check the authentication to consider either usn and password is in database or not through check_authentication(usn,pwd). 5. Username and password are eligible, so server update the status of user then send it (ONLINE and port of peer A) to other peer (upd_status(ONLINE,A.ID)) and then send the notify message success and return the friendlist of A to peer A (notify_msg(SUCCESS, friendlist)).
Exception flow	5a. Username or password is ineligible, so server send the notify message fail to peer A (notify_msg(FAIL)) *Go back to step 3.*

1.3.3 Log out

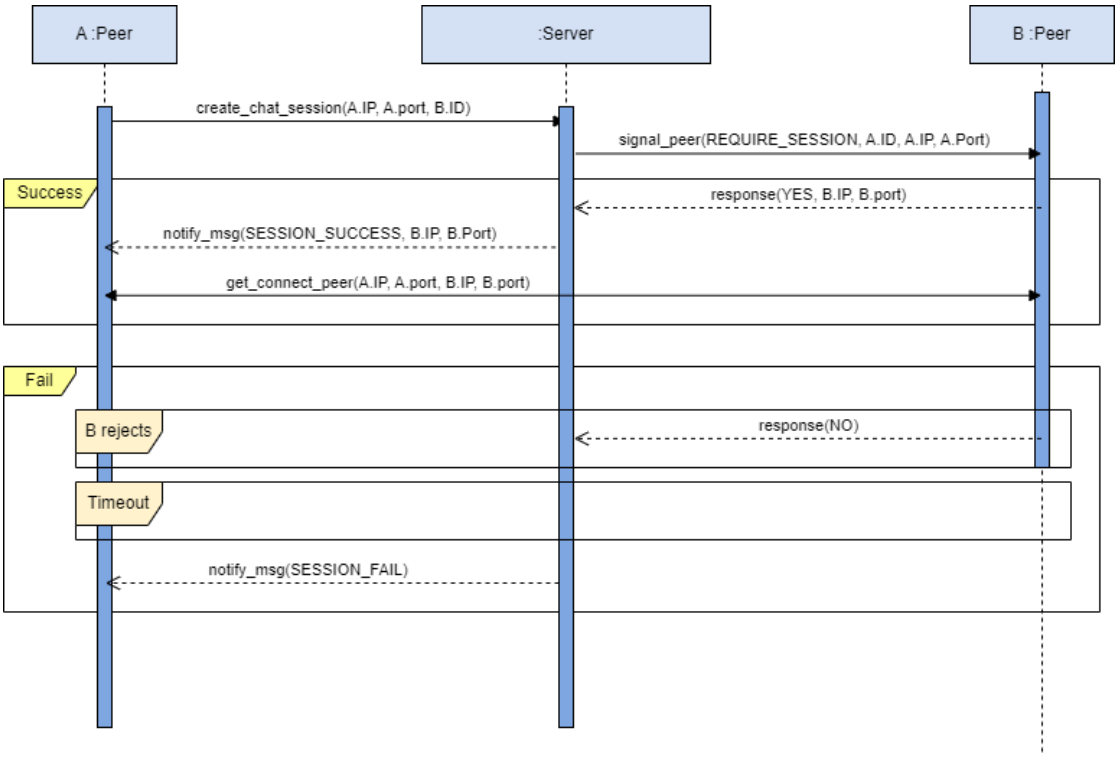


Hình 4: Log out

♣ Description

Use Case Name	**Log in**
Description	A peer want to log out account in server
Actor(s)	Peer A, Server, Peer B
Trigger	Peer A sent the logout request to the Server, or Session timeout.
Precondition(s)	Peer A must log in to server
Postcondition(s)	
Main flow	1. Peer A send the request log out to server and wait for accepting 2. Server accept and send notify success message to peer A. 3. Peer A close all thread and session. 4. Server update status OFFLINE of peer A to other peer (upd_status(OFFLINE, A.ID).
Alternative flow	1a. After TTL (time to live) up (peer A do not do any action on protocol), all chat session will be unavailable. 2a. Server will close all thread and section of peer A. 3a. Server update status OFFLINE of peer A to other peer (upd_status(OFFLINE, A.ID).

1.3.4 Create chat session

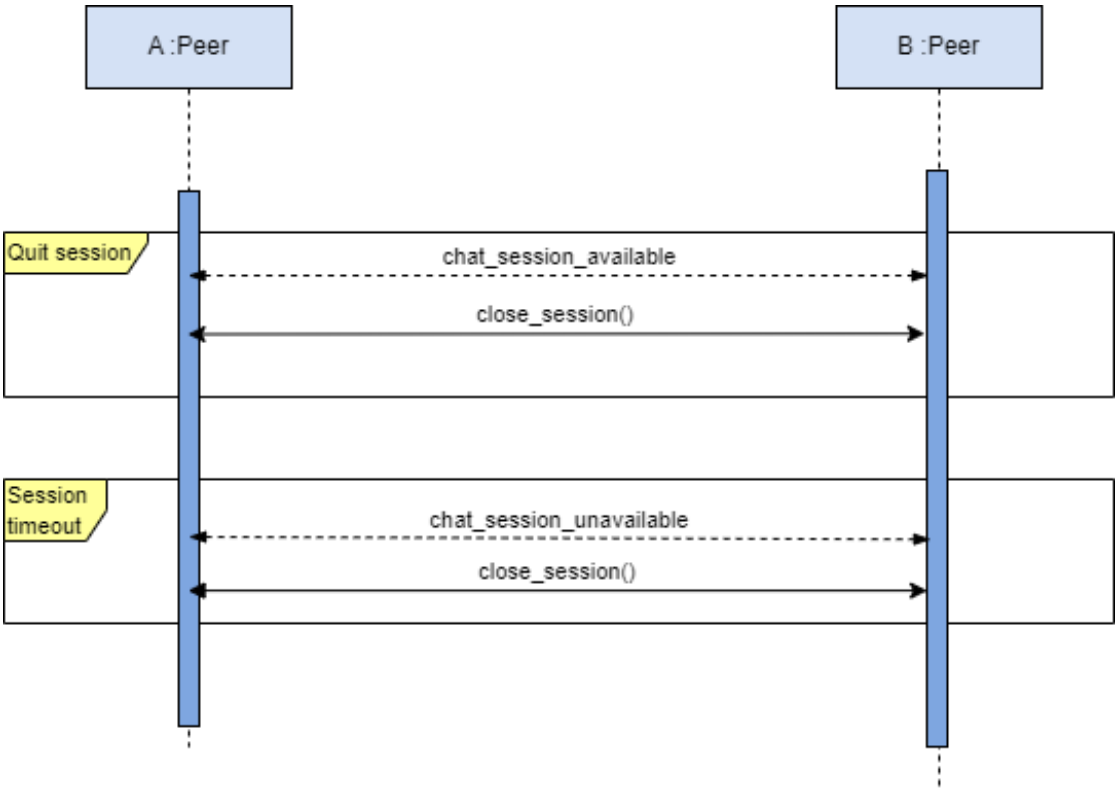


Hình 5: Initiate chat session

♣ Description

Use Case Name	**Initiate chat session**
Description	A peer initiate a chat session with other peer
Actor(s)	Peer A, Server, Peer B
Trigger	Peer A sent the *create_chat_session (A.IP, A.Port, B.ID)* to the Server.
Precondition(s)	Peer A must login to the app chat successfully
Postcondition(s)	Peer A create chat session with peer B successfully
Main flow	1. Server sent the connection notification from peer A to peer B 2. Peer B sends *response(YES, B.IP,B.port)* 3. Server send connection success message to peer A. 4. Initiate a chat session between A and B
Exception flow	2a. Peer B sends *reponse(NO)* to server *Server sends connection failure message to peer A. End usecase.* 2b. Peer B Timeout *Server sends connection failure message to peer A. End usecase*

1.3.5 Close chat session



Hình 6: Initiate chat session

♣ Description

Use Case Name	**Close chat session**
Description	A peer close a chat session that has already open with another peer.
Actor(s)	Peer A, Peer B
Trigger	Peer A or Peer B sent the *close_ session()* to the other. The time for that chat room is out.
Precondition(s)	Peer A and Peer B must already have opened a chat_session
Postcondition(s)	Close the chat session between A and B.
Main flow	1. Chat session is available between A and B. 2. Peer A or Peer B sent the *close_ session()* to the other. 3. The one received the *close_ session()* will close the session with its peer.
Alternaltive flow	1a.Chat session is unavailable between A and B. One of them lost the connection. *The rest one will wait until the timeout for that chat session, then close the session.*

2 Phase 2

In this section, we present our Chat Application.