

CS 408 - Computer Networks - Spring 2022 – Phase II

SUBook, a Social Networking Application

Demo: to be announced

Project Phase (Step) 2 - Deadline: 03.06.2022 (Friday), 11.59 (around noon):

Second phase of the project is built on top of the first step. In this phase, you will modify previous client and server modules to add more functionalities on top of them.

In addition to sending posts and requesting posts feed features as it was done in Step 1, the following features are to be added to the application.

1. Requesting post feed posted by friends only.
2. Adding and removing friends.
3. Being able to see your friend list.
4. Deleting own posts.

Each user can *add as a friend* another user. The concept of *adding a friend* is basically adding yourself to your friend's and your friend to your list of friends. So, when A is adding B as a friend, B is automatically added to A's friend list and vice versa. No acceptance or rejection is done. If a user is trying to add someone which is not in the database as a friend, in both server and client GUIs error messages should appear. The same process should be done when a user tries to add himself/herself as a friend.

Another mechanism that should be implemented in this phase is the ability of clients to request the posts only from the client's friends. In the previous phase, a client was able to request all posts by all clients. This feature should stay, but in addition to that in this last phase a client can also request the posts posted by only the **users' friends** and his/her **own** posts.

Moreover, the user should be able to see his/her friend list. You can use a list box to display the friends or displaying it to the rich text box. When a user removes a friend, she/he must be removed from the friend list and vice versa.

Finally, a client can delete a post of her/his own. For this purpose, a unique post identifier is needed, and the post ID will be used for this purpose. To delete a post, the user should send the post ID to the server and after the server deletes it, it should inform the client. If there is no such post or there is an ownership problem, then error messages should appear on both client and server GUIs.

As in the phase 1, all the operations must be clearly shown on the client and server GUI.

Furthermore, in package with the Term Project – Step (Phase) II, you will get the grading criteria upon which your Phase II will be graded. We will determine the points for each criterion after the deadline of Phase II. Furthermore, in case you might find it helpful, you will get the codes/solutions of Phase I so you can build on top of them by doing the necessary changes in the code (additions, deletions, etc.). This is mostly done for those that didn't submit Phase I. Of course, those that submitted the solutions for Phase I, we encourage them to work on their own submitted codes for Phase II. Also, on SUCourse+ you will have a demo guide video which briefly explains everything you need to know and do for Phase II.

Term Project – Phase II - Server Specifications:

- The port number on which the server listens is **not to be hardcoded**; it should be taken from the Server GUI.
- The server will start listening on the specified port. It must handle multiple clients simultaneously. To do so, whenever a client is connected to the listening port, the corresponding socket should be added to a list and the server should continuously accept other client sockets while listening.
- Only users who reside in the user database can connect to the server as a client. In other words, no user with a username that does not exist in the user database can connect, and its error message should be given in both GUI.
- All activities of the server should be reported using a rich text box on the Server GUI including the usernames of the connected clients, the usernames of the disconnected clients, error messages (Problem with port number, not a database user etc.) as well as all the post sharing and post feed request details along with adding or removing friends' notifications. **We cannot grade your project if we cannot follow what is going on; so, the details contained in this text box is very important.**
- When the server application is closed (even abruptly), nothing should crash! Also, the process in the operating system regarding the server should be terminated properly.
- When a user requests posts feed of only his/her friends or own, the server should take the required posts from the post text file and send it to the server. Also, the post ID should be incremented according to the posts in the text file. (Do not start it from 0 when the server restarts.)
- When a user adds/removes a friend, the server should keep the friend relationship of clients and act accordingly.
- When a user requests to delete a post of his/her own, the server should delete the post from the post text file.

Term Project – Phase II - Client specifications:

- The server's IP address and the port number **must not be hardcoded** and must be entered via the client GUI.
- There could be any number of clients in the system.
- If the server or the client application closes, the other party should understand disconnection and act accordingly. Your program must not crash!
- All activities of the client should be reported using a rich text box on the client GUI including sharing posts, error messages, and posts feed request details. **We cannot grade your project if we cannot follow what is going on, so the details contained in this text box is very important.**
- Each client can disconnect from the system at any time. Disconnection can be done by pressing a disconnect button on client GUI and by just closing the client window.
- If the client application is closed (even abruptly), nothing should crash! Also, the process in the operating system regarding the client should be terminated properly.
- Both connection and post transfer operations will be performed using TCP socket.
- When a user requests the post feed (for own posts, friend's posts, or all posts), all necessary posts should be seen in the client GUI.
- A user should be able to add a friend to his/her friend list, he/she should be added also in his/her friend's list also. The client should be able to see his/her friend's lists.
- The client should be able to delete his/her post using the post ID.

-----END OF PHASE 2-----

Group Work

- You can work in groups of **one, two, three and maximum four** people for all steps. There is a shared Excel file with the formed groups, so you can decide to join one. No group changes are allowed after the first step (phase). Any group changes must be performed before the submission of the first step. However, we do not recommend changing groups once you start the project since moving codes might lead to plagiarism.

- Equal distribution of the work among the group members is essential. All members of the group should submit all the codes for both client and server. All members should be present during the demos. In case of any dispute within the group, please do not allow the problematic group members to submit the code, so that he/she will not be graded. However, if a group member submits the same code, then the other members automatically accepts his/her contribution. In such a case, the same group continues with the second step. In other words, submitting step (phase) 1 together and separating for the other steps is **not** possible.
- If a particular student does not submit the first step of the project (due to being without a group or being excluded from a group due to low contribution/effort), he/she can join another group for the second step.
- Similarly, if a particular group member does not work enough in the second step, please do not let him/her submit. You can also inform us about this.
- Your TA Müge Kuşkon (mugekuskon@sabanciuniv.edu) is responsible for keeping track of the groupings. You can contact her if you have any questions related to the term project.
- You have a chance to form your own groups. However, if you cannot find enough people to form a group, then you have to accept to work with people that we assign. If you do not like to work people that you do not know, then form your own groups or work yourself!
- **If someone does both of the phases alone, he/she will get an extra 3% towards the final grade**

Programming Rules

- Preferred languages are C#, Java and Python, but C# is strongly recommended
- Your application should have a graphical user interface (GUI). **It is not a console application!**
- You must use pure TCP sockets as mentioned in the socket lab sessions. No other socket classes is allowed.
- Your program should be portable. It should not require any dependencies specific to your computer. We will download, compile and run it. If it does not run, it means that your program is not running. So do test your program before submission.

Submission

- Submit your work to SUCourse+. Each step will be submitted and graded separately.
- Delete the content of debug folders in your project directory before submission.
- Create a folder named **Server** and put your server related codes here.
- Create a folder named **Client** and put your client related codes here.
- Create a folder named **XXXX_Lastname_OtherNames_StepY**, where XXXX is your SUNet ID and Y is the project step (1 or 2) (e.g. mugekuskon_Kuskon_Muge_Step2). Put your Server and Client folders into this folder.
 - Compress your XXXX_Lastname_OtherNames_StepY folder **using ZIP or RAR**.
- You will be invited for a demonstration of your work. Date and other details about the demo will be announced later.
- **24 hours late submission is possible with 10 points penalty (out of 100).**

For questions and support related to Phase II, you should exclusively send an email to your TA Müge Kuşkon (mugekuskon@sabanciuniv.edu) or attend her online office hours given below:

1. 26.05.2022 (Thursday), 10.40-11.30
2. 26.05.2022 (Friday), 10.40-12.30
3. 30.05.2022 (Monday), 09.40-12.30
4. 02.06.2022 (Thursday), 10.40-12.30

through this Zoom link:

<https://sabanciuniv.zoom.us/j/2457215757?pwd=ZE1lTjdUd1JUyTJRQmx1KzNYVnIvZz09>

Good luck!

CS204 Team (Müge Kuşkon, Artrim Kjamili)