CS408 – Computer Networks – Spring 2022
Homework 1 – Creating accounts
Deadline 14.10.2022 (Thursday), 23.00

- **No collaboration is allowed. You are not allowed to ask and get help from your classmates.**
- Your application should have a graphical user interface (GUI). It is not a console application! Both server and client should be implemented.
- You must use pure TCP sockets as mentioned in the socket lab sessions. You are not allowed to use any other socket classes.
- Please submit to SUCourse+ in a zipped format and delete the content of debug folders in your project directory before submission. More submission related info is given at the end of this document.
- Test your program before submission.
- For eventual questions related to the Homework 1 **you should exclusively contact your TA Müge Kuşkon** (preferably) during her zoom office hours announced on SUCourse or you can send her an e-mail (mugekuskon@sabanciuniv.edu).
- For clarification purposes, on SUCourse under the "Assignments and Term project" tab you have a clarification video on how should the server and clients behave. If a detail is not specified in the video or this document, you can implement it in the way that you think it should be done.
- In package with this assignment, you also have the grading criteria by which your assignment will be graded.

The aim of this assignment is to develop an application where the client will connect to the server and then will be able to create an account via Graphical User Interface (GUI). IP and port number will be taken as inputs from the user GUI via textboxes and a connect button should be present in order to connect the client. Cases such as entering wrong IP or port number should be checked and error messages should be shown accordingly in the Client GUI or Server GUI.

After the connection is established, the client will enter his/her name, surname, username and password in textboxes and a button which will allow the user to create an account. A rich textbox that shows the errors, notifications or actions must be present. The client should also check whether the name, surname, username or password is empty or not. If any of them is empty, the user should enter it otherwise he/she will not be able to create an account. Furthermore, the password should be in a hidden form while the client enters the credentials.

Moreover, when the user tries to create an account, the server should check only whether the **username** is in the text file that it creates ***(database.txt)***. If his/her username is in the ***database.txt***, no account should be created. Also, when a new user is added to the database, all information related to the user (username, name, surname and password) should be stored in that ***database.txt***. Finally, the user should be able to disconnect via a button from the server

Server Specifications:

- There is only one server running on this system.
- 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.
- The server accepts any user which enters the correct IP and port number via the connect button on the client GUI.
- Only users who don't reside in the user database (***database.txt***.) can create an account. In other words, user with a username that exists in the user database cannot create another account with the same username, thus the server prevents him/her to create an account and report an error message on the server GUI. (e.g. "There is already an account with this username")
- When a new account is created, the server should record all of the user information (name, surname, username, password) in a text file (***database.txt***.)
- All activities of the server should be reported using a rich text box on the Server GUI including the usernames of the users that created a new account, messages (A client is connected! Already existing username! B has created an account! etc.), and errors (Check the port number! etc.). We cannot grade your assignment if we cannot follow what is going on.
- When the server application is closed (even abruptly), nothing should crash!
- Other behaviors of the server can be observed from the clarification video on SUcourse+.
- Make sure that you have correctly implemented all of the server requirements by which you will be graded according to the grading file that came in package with the assignment!

Client specifications:

- The IP address and the port number **must not be hardcoded** and must be taken from the client GUI.
- The password of the client should be entered in a hidden form in the client GUI.
- All activities of the client should be reported using a rich text box on the client GUI including messages ("Successfully disconnected" etc.), errors ("Please enter you surname!", "Check your port number!", "There's already an account with this username" etc.) and success message of creating an account. We cannot grade your assignment if we cannot follow what is going on.
- Each client must have a unique username. This username must be entered using the client GUI. This username is also sent to the server. The server checks the presence of accounts in the database using **client's usernames only**.
- The client should be able to disconnect.
- 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.
- Other behaviors of the clients can be observed from the clarification video on SUcourse+.
- Make sure that you have correctly implemented all of the server requirements by which you will be graded according to the grading file that came in package with the assignment!

Submission:

- Submit your work to SUCourse+ as a single zipped file at the designated place
- 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, where XXXX is your SUNet ID and Y is the project step (1, 2 or 3) (e.g. artrimk-Kjamilji-Artrim). Put your Server and Client folders into this folder.
- Compress your XXXX-Lastname-OtherNames folder using ZIP.
- You might be invited for a demonstration of your work. If so, you will be informed about the date and other details about the demo later on.
- 24 hours late submission is possible with 10 points penalty (out of 100).

Good luck!

CS408 Team (Müge Kuşkon and Artrim Kjamilji)