Project 1 (Total points: 20)

Due time: Oct. 24th, 2023 11:59pm

The purposes of this project:

(1) Understanding how to implement the client and server application architecture with TCP sockets
(2) Understanding how to implement a multithreading server to serve multiple clients simultaneously
(3) Understanding how to synchronize shared resources among threads

After you submit your project, **please download it and make sure you submit it successfully and correctly**.

Implement a very simple "GoFundMe" simulation server and client with Java TCP socket programming. The server should support multiple clients (>=2) through multithreading programming and one server thread serves one client. Especially pay attention to data synchronization, e.g. when multiple clients donate to the same fundraising event.

For the server and clients, they work together to provides the following functions, which are allowed users to select:

- Create a new fundraising event (event name, target amount, deadline) on the server through client requests. For graduate students, you need store the event data to a file. For undergraduate students, you can store them in RAM. (It means every time you restart your server; all event data are gone. But for the graduate students, you need load data from the file.)
- List current and past fundraising events in separate categories and list them with indices according to the deadline in their own category.
- Donate to a current fundraising event through user inputs (e.g. event index and amount)
- Check the balance and deadline of current and past fundraising events

Requirements:

1. Log requests and/or responses on the server and/or client side through printing out information (e.g. client IP address, port number, request type etc.) and make it clear for users to understand the context and easy to use
2. Please submit a compressed file which contains your source code and user manual (with screenshots), and a short video recording which you and/or your teammate (at most 2 persons for a team) demonstrate how to use your client and server to illustrate all functions. Please domo multiple clients using the server at the same time.
3. Please make sure your code is executable. You can also provide a simple README on how to run your code.
4. Please comment your source code briefly.
5. Remember to deal with invalid user input and exceptions. Make your program to be robust.

6. We just implement a simple “GoFundMe” simulation and some important functions (e.g. payment, user/event verification etc.) are out of our capabilities.