Cairo University
Faculty of Engineering
CMP(N) 306
Advanced Programming Techniques

Socket Programming in Java

Requirement:

In this requirement, we will implement a **connection-oriented** network communication with socket programming using server-client architecture for a hospital system. Clients can ask the server to make an appointment to a doctor or cancel the appointment. Each doctor has pre-defined set of available time slots.

Define the following classes:

1. **Class Doctor:** contains the following data members: id, name, an array of booleans called timeslots indicating if the doctor is available during each timeslot or has an appointment, and an array of strings called patients containing the patient's name of each time slot if he had an appointment with the doctor in the corresponding timeslot. The size of the timeslots and the patients arrays are the same. Each time slot is defined by its index in the timeslots and the patients arrays (no need to store the day, hour, etc.)

2. **Class Hospital:** contains an array of doctors as a data member and the following:
    a. **Constructor**: reads doctor's data from file and fills the array of doctors
    b. **MakeAppointment**: takes the doctor id, the index of his time slot to reserve, and the patient's name, and makes the appointment if possible. The function updates the doctor data members appropriately.
    c. **CancelAppointment**: takes the doctor id, the index of his time slot to reserve, and the patient's name, and cancels the appointment if possible. The function updates the doctor data members appropriately.
    d. **Print**: for each doctor in the hospital, it prints the id of the doctor and the name of his patient in each time slot index.

3. **Class Client**:
    a. Prompts the user to enter his name (the patient's name)
    b. Opens two sockets with the server, one on the make appointment port 6666, and the other on the cancel appointment port 6667
    c. Sends the patient's name to both sockets
    d. In a loop,
        i. Prompts and asks the user if he wants to make or cancel an appointment
        ii. Reads the doctor id and the timeslot index from the user
        iii. Sends the doctor id and the timeslot index to the appropriate socket
        iv. Reads the response message from the server
        v. Prints the response message to the user on console

4. **Class Server**:
    a. It contains a hospital object that contains many doctors
    b. The server should be able to take requests from clients **in parallel** on both ports such that the server can make appointments for many clients in parallel, cancel appointments from many clients in parallel, make appointments and cancel appointments from many clients in parallel. **Use multithreading for that**.

    c. For serving a make appointment request coming on port 6666:
        i. Reads the patient's name from the client socket
        ii. Reads the doctor id and the timeslot index sent from the client socket
        iii. Calls function makeAppointment of class hospital to make an appointment for this patient name to the read doctor id at the read timeslot index
        iv. Sends a response message to the client socket with a descriptive message indicating whether:
            1. Making the appointment is done successfully (Success)
            2. the doctor id is not found in hospital (Failure)
            3. the timeslot index is out of boundary (Failure)
            4. the doctor is already busy at this timeslot (Failure)
        v. Calls function print of the hospital object to print it on the server console

    d. For serving a cancel appointment request coming on port 6667:
        i. Reads the patient's name from the client socket
        ii. Reads the doctor id and the timeslot index sent from the client socket
        iii. Calls function cancelAppointment of class hospital to cancel the appointment of this patient from the read doctor id and timeslot index
        iv. Sends a response message to the client socket with a descriptive message indicating whether:
            1. Cancelling the appointment is done successfully (Success)
            2. the doctor id is not found in hospital (Failure)
            3. the timeslot index is out of boundary (Failure)
            4. the doctor doesn't have an appointment at this timeslot (Failure)
            5. the doctor has an appointment to a different patient name at this timeslot (Failure)
        v. Calls function print of the hospital object to print it on the server console