CSE 4074 – Programming Assignment
Due 22.12.2019 Sunday, 23:59 (no extension)

Socket Programming – Travel Agency

Consider a customer who needs to plan a trip. A trip plan consists of a hotel reservation in a requested time interval and a flight reservation for requested number of travelers. Specifically:
A customer informs the travel agency of the arrival and departure date of his trip, his preferred hotel and airline and the number of travelers. Then travel agency contacts to the hotel and airline and ask for availability for corresponding dates. If there are available rooms for given number of people in the hotel and available seats in the flight for a given number of travelers, then the travel agency finalizes the trip. (You may assume that all of the rooms are for single person and you may assume that there exists single flight in a day for each airline). When the trip is finalized, the updates are reflected in the databases of the hotel and the airline. If the preferred hotel or flight is unavailable, then the databases remain unchanged and the travel agency contacts to alternative hotel(s) or airline(s) and propose an itinerary to the customer. If the customer accepts the proposed itinerary, then the travel agency can finalize the trip as before. Otherwise, the travel agency will keep on generating another itinerary until it runs out of options.

Implementation Details: You can use any programming language (Java, Python or C/C++). The customer communicates only with the travel agency (not with the hotels and the airlines). The customer uses TCP sockets to communicate with the travel agency. The travel agency communicates with the hotels and airlines using HTTP. (You have to implement web services using TCP sockets. Unfortunately you are not allowed to use high level web service libraries/frameworks). Each hotel and airline has its own database to keep track of availability. The travel agency does not have a direct access to these databases.
Client software will contain a simple GUI for the customer to enter dates, preferred hotels, preferred airlines and number of travelers. The result of the entry should be displayed on the GUI and the user should be able to accept or reject the alternative itinerary if proposed.

You may implement your code for single customer and single travel agency. However, there should be at least two hotels and two airlines in order to demonstrate different scenarios. **You should use only socket programming. Do not use any sophisticated libraries downloaded from the web.** You can use a simple text document for databases, but it would be preferable if you use a database tool.

You will give a demo about your project. Demo schedule will be announced by Ozan Neli later. In the demo, you should be able to show the following cases:

1. The travel agency finalizes the trip using the preferred hotel and airline.
2. The travel agency proposes alternative itinerary due to unavailability of hotel or flight, and the customer accepts it.
3. The travel agency proposes alternative itinerary but the customer rejects it, and the travel agency proposes another itinerary.

Bonus: You'll get extra 15 points if you implement travel agency multithreaded in order to serve for multiple customers and consider concurrency issues among multiple customers. You should submit a detailed report to illustrate your solution.
Bonus: You'll get extra 15 points if you implement your system such that number of hotels and airlines may change dynamically.

You are expected to do your project in groups of **two**. **However the group information should be sent to Ozan Neli before November 26th, Tuesday. No groups can be formed after that time.**

What to submit? - You can submit your projects in a zip file which contains your well COMMENTED source code and DETAILED report via google classroom. Please do not forget to write the names of people in your groups. Detailed report should include description of the protocol used between customer and travel agency, flow diagrams, important implementation details, and a user manual that describes how to run and test your code.