DISTRIBUTED SYSTEMS LABORATORY

Exercise #2 - Date Delivery: 28/5/2023

A. You are asked to build in Java RMI a minimum requirements client/server application (which can be accessed by many clients at the same time), through which each client will be able to reserve one or more seats for a Theater Performance at Theater X.
Consider that the following seats are available from the theater in total:

- 100 PA type seats (Square - Zone A) that cost 45 Euros each
- 200 seats type PB (Square - Zone B) which cost 35 Euros each - 400 seats type PG (Square - Zone C) which cost 25 Euros each - 225 seats type KE (Central Exit) which cost 30 Euros each - 75 PG (Side Theories) seats that cost 20 Euros each

For the implementation you should write a server (TÿServer), a client (TÿClient), a remote interface (TÿInterface), and the implementation of this interface (TÿImpl).

The client should be run on the command line as follows:

ÿ **java TÿClient:** if no other parameters are specified, the program should simply print on the screen exactly how (with what parameters) the user should run the command.

ÿ **java TÿClient list <hostname>:** to print as a result on the screen a list of the available ones seats (for any type and cost) in the theater - e.g. in the following form:

1 seats Square - Zone A (code: PA) - price: 45 Euros 2
seats Square - Zone B (code: PB) - price: 35 Euros 3
seats Square - Zone C (code: PG) - price: 25 Euros 4
seats Central Outer Seat (code: KE) - price: 30 Euros and
5 seats Side Seats (code: ÿÿ) - price: 20 Euros

where k1,k2,k3,k4,k5 indicate the current number of available seats of each type.

ÿ **java TThe Client book <hostname> <type> <number> <name>:** to reserve <number> seats of type/code <type> to <name>. A relevant success or failure message should also be printed on the screen (if eventually the requested positions were not found available - this can happen even though the user had initially seen them available because in the meantime someone else may have caught up and closed some of them). If fewer seats are found available, the user should also be asked if they wish to book only those seats. In any case, the total cost of the reservation will also be refunded.

ÿ **java TÿClient guests <hostname>:** to print a list of all the customers of the theater (that is, of those who have a reservation) and the reservations that each of them has made (for how many and what type/code of seats and total cost).

ÿ **java TTheClient cancel <hostname> <type> <number> <name>:** to cancel the corresponding <number> positions of type/code <type> for which the user <name> had booked. To as a result, a list of the remaining seats (except those canceled) for which the same user has any valid reservation, or an appropriate failure message in case the seats to be canceled were not found, is returned.

B. You extended the Java RMI application you made in A. so that it also supports the following:

ÿ In the event that during the **'book'** function, a failure message was returned to the client, the user should also be asked at the end if they want to subscribe to the theater lists for immediate notification when a reservation for the specific type of seats requested is canceled. If the user answers positively, the registration should be made (they should be kept by the theater server separate notification lists for each type of position).

ÿ Every time a **'cancel' operation is performed,** the server should immediately send notifications to those clients who have subscribed to the notification list for the specific type of seats. The notification should simply be the printing of a relevant message on the client's screen.

Deliverables: code, commentary/documentation, test runs