The restaurant

Events portray the activities that take place when a group of students, enrolled in *Sistemas Distribuídos*, go to a famous restaurant downtown for a gourmet dinner to celebrate the beginning of the second semester. There are three main locations in the restaurant that should be accounted for: the *table* where the students sit to have their meal, the *kitchen* where the chef prepares it according to the orders placed by the students, and the *bar* where the waiter stands waiting for service requests. There are, furthermore, three kinds of interacting entities: the *chef* , the *waiter* and N *students*.

The activities are organized as described below

- the students arrive one by one at random times and sit at the table, chatting with one another while waiting for the group to be complete;
- once a students sits, the waiter brings her/him a copy of the menu so that she/he can select among the offered dishes the ones of her/his preference;
- the first student to arrive gathers the individual plate choices of her/his companions and prepares the order for the whole group;
- once the order has been completed, she/he calls the waiter and informs him about its content;
- the waiter, then, goes to the kitchen and places the order to the chef;
- the order consists of M courses per participant in the dinner;
- the waiter serves them in succession, only passing to the next course when signaled by the last student to finish eating that everybody is ready;
- in the end, the student that was the last to arrive signals the waiter to bring her/him the bill after both her/his companions and her/himself have finished the dessert and pays it in full as a form of penalty for being late;
- all students leave together the restaurant and go home to study because an assignment deadline is soon due.

Assume there are seven students and that the order consists of three courses per participant in the dinner: a starter, a main course and a dessert. Write a simulation of the life cycle of the students, the chef and the waiter using the client-server model with server replication, where the students, the chef and the waiter are the *clients* and the access to the information sharing regions are the services provided to them by the *servers*.

The operations that were assigned to activities previously carried out in the information sharing regions (for the already implemented concurrent version), must now be assigned to independent requests performed on the servers, seen as remote objects, through the remote method of invocation.

One aims for a solution to be written in Java, to be run in Linux under Java RMI, either in a concentrated manner (on a single platform), or in a distributed fashion (up to 7 different platforms), and to terminate (it must contemplate service shutdown). A *logging* file, that describes the evolution of the internal state of the problem in a clear and precise way, must be included.

Guidelines for solution implementation

1. Specify for each representative server of an information sharing region the structure of the messages to be exchanged.
2. Specify the general organization of the servers architecture.
3. Specify the general organization of the clients architecture.
4. Sketch the interaction diagram which describes in a compact, but precise, way the dynamics of your solution. Go back to steps 1, 2 and 3 until you are satisfied the description is correct.
5. Proceed to its coding in Java as specific reference data types.
6. Specify the mapping of the servers and the clients onto multiples nodes of the parallel machine and write the shell scripts which enable the deployment and the execution of the different modules the application is composed of.
7. Validate your solution by taking several runs and checking for each, through the detailed inspection of the logging file, that the output data is indeed correct.