

Distribution and Integration Technologies

Assignment #1

Intranet Distributed Application (using .NET Remoting)

Restaurant order and account system

2019

Scenario

A restaurant needs to automatize the dining room orders, allowing them to be quickly communicated to the kitchen and bar tenders, to be prepared as soon as possible.

Also, the same system should maintain a complete record of all orders, compute the bills of each table, and maintain a record of the total amount received in the day.

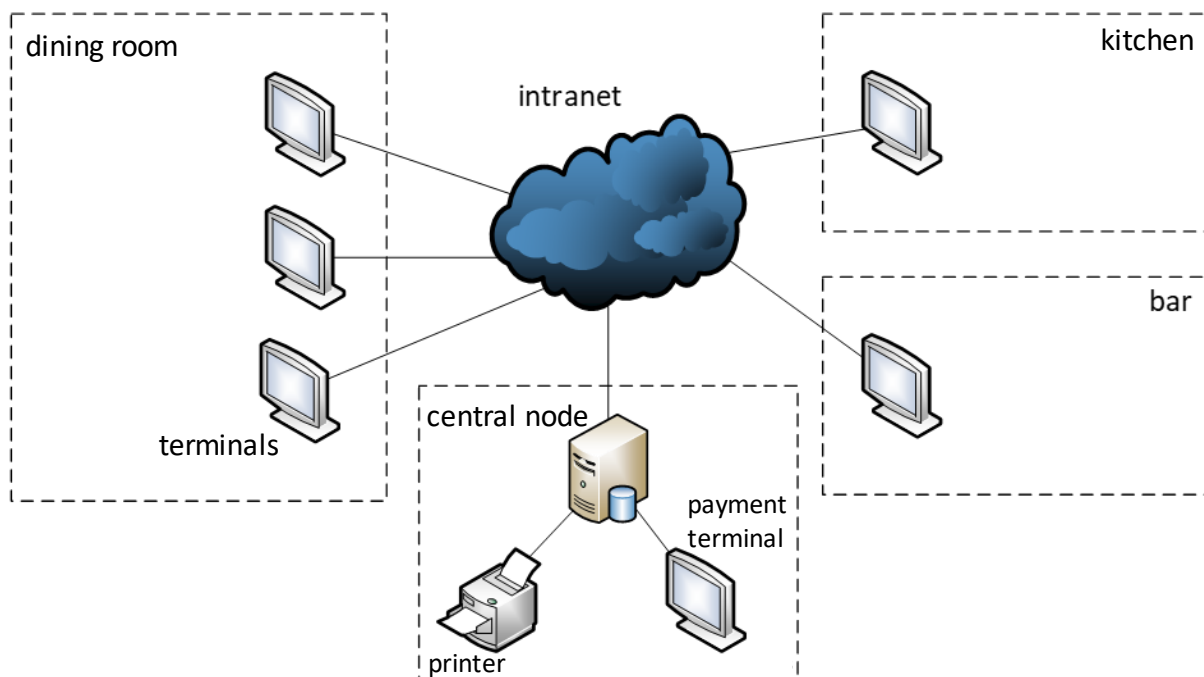
In the dining room there are several terminals (client computers) where the waiters can make new orders (in any of the terminals), opening a new table or adding to the orders already assigned to a table. Those orders can have as destination the kitchen (dishes) or the bar (drinks).

When the orders become prepared, cooks and bar tenders should signal the event, letting the waiters know that they can fetch the orders and serve them to the tables.

In the bar area and the kitchen there is also a terminal where the orders should appear, already divided (bar orders in the bar terminal and kitchen orders in the kitchen terminal).

When a free bar tender or cook picks one order to prepare it, he should signal that in the terminal changing its status from 'not picked' (the original status of any order) to 'in preparation', preventing other person to prepare the same order.

When the order is prepared (drink or dish) the same person should go to the terminal (bar or kitchen) and change its status to 'ready', and that should be signaled or shown in the dining room terminals.



Payment zone

Near the dining room there is a small zone with a computer connected to a printer and terminal for closing a table and emit an invoice for the payment. This central computer can execute remote objects dealing with the tables, the orders and their statuses, and the payments (to simplify you can use memory data structures for the information storage – e.g.

a *dataset*, an *arraylist*, a *map*, or any other). The remote object (or objects) should implement methods for the client applications running in the terminals, and events that should maintain all displayed information current, in every terminal, in real time.

An order in the system should be characterized by a unique identifier, a description, a quantity, destination table, type (kitchen or bar), price, and a state ('not picked', 'in preparation', 'ready', and 'paid').

When a table is closed in the payment terminal, an invoice should be printed containing a list of the orders for that table, their unit and quantity price, and the total value to pay. The printer can be simulated by a console application running in the central node. The system should also have available a small application for indicating the total value collected so far, and the number of tables served.

Dining room terminals

The application executing in these terminals should allow the specification of new orders from a predefined list of possible products (dishes or drinks), associated with a quantity and a table. Use a list, menu, or button collection for choosing the products. The orders should be automatically signaled in the right place (kitchen or bar).

When any order becomes ready, that should be signaled in the graphic interface of the dining room application in a well visible way (e.g. using a special screen area, a change of color, ...).

Kitchen and bar terminals

There is only one terminal in each of these places, running the same application, but configured for the place where it is executing.

New orders placed in the dining room should appear in this application interface, according to the type of order (dishes in the kitchen and drinks in the bar).

When one of the cooks or bar tenders picks a new order to prepare it (e.g. clicking on it) its status changes from 'not picked' to 'in preparation' (e.g. moving it from one list to another or changing its color). When the order is ready, again the responsible should change its state to 'ready' (e.g. clicking or double clicking it on the interface). Also, in the dining room applications, that indication should appear, allowing some waiter to pick it and serve it to the associated table.

Implementation

This system of applications and functionalities should be written using .NET, with a set of methods and events in remote objects in .NET Remoting. The user interfaces should be graphic (GUI, using for instance WinForms) but simple and intuitive to use. The application simulating the printer could be textual (console). You can always include other functionalities considered useful (will be valued) and implement everything using the best practices for the technologies in place.

Delivery

You should prepare a file, containing the developed code (central server, printer simulation, payment application, dining room application, kitchen and bar application), a report describing the general project architecture and the relevant details (remote objects, their methods, events and subscribers, ...), functionalities included, and screen captures illustrating the main sequences of use.

The working of the system will also be demonstrated in class. You can run everything in the same computer.