

C340 Concurrency
Compulsory Coursework

To be handed in to G14
Deadline: Dec 11 1998, 12:00am

A London Jazz club that is very popular (probably because it is linked to a Pizza restaurant) offers dinner during Jazz concerts. As it is so popular, it needs to improve its booking system, because customers, phoning to reserve tables, overload the members of staff. The owner Ella, who happens to be a techie decides that she wants to go for a Web-based system. She buys a machine, installs Java and a web-server, gets an Internet connection in place, and thinks about the requirements for the system.

She wants an applet that shows the tables of the restaurant and indicates those that are still vacant using green colour. To book a table she would like customers to choose a free table by entering the number of the table and by entering the customer's name. The system would then respond with a reservation number that clients keep so that they can obtain and pay for their tickets at the entrance of the Jazz restaurant. While reservations are made, the applets would write log information on a file. Ella then wants to print the log file each night. The people manning the entrance then know which customers have reserved tables.

Half way into writing the Java applet, Ella recognizes that her programming skills are not sufficient - she had not anticipated that more than one customer might use the applet at the same time. She calls a friend, who has access to a bunch of bright students...

Part 1: (30%)

Construct and submit an FSP model of the table booking system and demonstrate that your model does not permit double bookings. (Hint: It is only necessary to model a few Web browsers and a few tables).

Part 2: (30%)

Develop and submit an object-oriented design using the UML for the booking system. Identify the different classes involved in the applet and show their attributes and operations. Also show the inheritance, aggregation and dependency relationships between the classes.

Part 3: (40%)

Implement your design in Java. Use concurrent threads but do not attempt a distributed solution using sockets or RMI. Make sure that there is a direct correspondence between the Java classes, the classes' associations, operations and attributes with the UML design you produced to respond to Part 2. Submit a URL of a web page that embeds the reservation applet and has a link to the source code. Also submit a set of test cases that demonstrate that the classes implement the FSP model properly.