

Distributed Programming
Web Programming Test Assignment
Submission deadline 17th July 2015, h 23.59

The project consists in developing a simplified version of a web application for managing booking operations related to the conference hall of a large hotel. There is an upper limit on the number of persons who may stay in the hall at the same time. The hall can be divided into smaller parts of arbitrary size for hosting smaller conferences, but the sum of the numbers of participants of the various conferences taking place at the same time can never exceed the maximum capacity of the whole hall. A booking for the hall must include the number of participants and the start and end time (each one specified as hour and minute). For simplicity, assume all the conferences take place in the same day. The system **MUST** be able to efficiently manage the bookings according to the following specifications:

1. All the users who access the website can see, without any authentication or registration mechanism, the list of bookings, each one with its number of participants, start time and end time, but without the username of who made the booking. The list is sorted according to the number of participants, with the bookings having the higher numbers of participants first.
2. Each user can freely register to the website, just by providing a username and a password. If a given username has already been chosen by another registered user, the system **MUST** deny the registration and ask the user for another username in order to always guarantee username uniqueness.
3. An authenticated user must be able to see, in his/her personal page, the bookings already made by other users, including the username of who made each booking. In the same page, an authenticated user must be able to request a booking for a new conference, by specifying the number of participants, the start time and the end time. The booking operation must be successfully completed only if the request is compatible with the other, already present, bookings, i.e. if it does not totally or partially overlap in time with other already existing bookings or if it overlaps, but the total number of participants does never exceed, at any time, the total capacity of the conference hall.
4. An authenticated user **MUST** be allowed to display at any time the information related to his/her bookings and to cancel one of his/her bookings, if desired.
5. In the submitted project, the total capacity of the conference hall must be of 100 persons. Moreover, there must be at least three bookings with numbers of participants equal to 60, 50 and 40, there must exist at least three registered users, with usernames $u1$, $u2$ and $u3$ and passwords $p1$, $p2$ and $p3$. Each one of the previously mentioned bookings must have been performed by a different user ($u1$, $u2$ or $u3$). Two of the previously mentioned bookings must be partially overlapped by at least 1 hour.
6. Authentication must be done by username and password when required, and it must remain valid if the user does not remain idle for more than 2 minutes. If a user attempts to perform any one of the operations that require authentication after an inactivity time longer than 2 minutes, the operation must have no effect, and the user must be forced to re-authenticate with username and password. The system must force the use of the HTTPS protocol for authentication and in every part of the website accessible only through authentication.
7. The general appearance of the web pages must include: a header in the upper part of the page, a navigation bar on the left side with all the links needed to perform the various operations, and a central part which is used for the main operation.
8. Cookies and Javascript must be enabled, otherwise the website may not work properly (in that case, for what

concerns cookies, the user must be alerted and the website navigation must be forbidden, for what concerns Javascript the user must be informed). Forms should be provided with small informational messages in order to explain the meaning of the different fields. These messages may be put within the fields themselves or may appear when the mouse pointer is over them.

9. The graphical layout must be consistent, that is, the pages must be as much as possible uniform among all the different browsers.

10. Extra requirements only for students having the 8-credits course: design an XML format that can be used to represent a set of bookings, according to the above specifications. Specify the designed format by means of an XML schema named `bookings.xsd`. This schema must be made available in the main folder of the application (along with the index page). Then, add an administration page to your application. This page must be available through HTTPS only (not through HTTP), and it must include a form by which the administrator can upload a collection of bookings, specified in an XML document written according to the designed format. The form should also include the input of the (required) administrator's credentials (a password). The XML document is considered by the server only if the provided credentials are present and correct. The admin password can be hardcoded in the application or stored in the DB as you prefer. This password must be set to "admin" in the deployed application. The uploaded bookings will be added to the existing ones in the application if compatible. If at least one of the bookings is not compatible, the whole request must be refused and no change made to the set of bookings.

Submission instructions:

The instructions already published in the Material folder of the course web page for the installation on the `cclix11.polito.it`, still hold. Furthermore, you need to submit your project in a zip file to the following web site: <https://pad.polito.it/enginframe/dp1/dp1.xml> (from inside the Politecnico network) or <https://pad.polito.it:8080/enginframe/dp1/dp1.xml> (from outside).

In addition:

1. The sql script included in the zip file (submitted to `pad.polito.it`) to create the database must have a name with the following pattern: `sXXXXXXX.sql` (where XXXXXX is your own student id).
2. The starting page of your web site must be put in a file named `index.html` or `index.php`
3. DO NOT use absolute links

WARNING: The system that accepts your projects, works in an **automatic way and it will stop accepting submissions at the scheduled deadline. For this reason, we recommend you DO NOT submit your work in the very last minutes before the final deadline.**

In case of any doubt and question related to the understanding of the assignment, please firstly visit the forum in the course website in order to check if other students have already asked the same question. Otherwise use the forum (not the teacher email) to ask your question so that the response will be available to all the students.