

**Distributed Programming**  
*Web Programming Test Assignment*  
**Submission deadline 25th June 2015, h 23.59**

The project consists in developing a simplified version of a web application for managing spare time activities with a limited number of places (for example soccer, volley, swimming, etc...). To participate to the activities, a reservation is necessary. A reservation includes one adult and, optionally, one or more children, up to a maximum of three children (thus you can assume that an adult may participate by accompanying up to three children). The system **MUST** be able to efficiently manage the reservation mechanism according to the following specifications:

1. All the users that access the website can see, without any authentication or registration mechanism, the activities list, the number of reserved places so far for each activity and the maximum number of places available for that particular activity. The available places can be used for adults as well as for children, given that children **MUST** be accompanied. Activities are ordered by the number of available places (activities with higher places availability are listed first). For simplicity, you can assume that each activity does not contain any temporal information related to when the activity starts or ends. In other words, it will be always possible to reserve places for each activity (subject to availability, of course).
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 (who is expected to be an adult) can make reservations. In case the user is interested in including children in the reservation, the reservation process **MUST** be atomic so as to ensure that the whole reservation is entirely accepted (or possibly rejected). It is not possible to accept a reservation if not all the requested places are available. Furthermore, it is not possible to add children to a reservation **AFTER** the reservation has been created. Another requirement is that reservations with a number of children greater than 3 **MUST** be rejected. It is not required to insert the name (or any other ID) for each child during the reservation process.
4. An authenticated user **MUST** be allowed to display at any time the information related to his reservation and, possibly, cancel his reservation. A cancelled reservation **MUST** be deleted completely (all the places must be released). If places were also booked for children, they must be released accordingly.
5. In the submitted project there must be three activities with a maximum number of places of 6, 8 and 4; furthermore, there must exist three user, *u1*, *u2* and *u3* with passwords *p1*, *p2* and *p3*. Each user must have performed at least two reservations for two different activities not completely overlapped with other users. At least one reservation for each user must include at least one child.
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 activity reservations, 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 reservations, 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 reservations will be added to the existing ones in the application.

#### **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 project, 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.