Development of Web Applications from Server-side

Project: Library Management Program

General Description

You are going to develop an application to manage a modern and complete library with different classes of elements (books, magazines, DVDs...) and different classes of protection levels (long booking, short booking, no booking, reserved...).

Anyone must be able to search the catalogue of this library through its website and to register to become a member of the library. Obviously, some compulsory information must be required to do so.

Members, which log in using a username and a password they can set at registration and change at any time, can search the catalogue and book an element of it for any future date in which it is available, if its level of protection allows it.

Librarians, that are another type of users with more privileges, can perform the same tasks than members and besides manage (add, remove and change) the elements of the catalogue and the users. Moreover, they can book an element for any user, lend any available element with the required level of protection (previously booked by a user or not) and perform a return.

Finally, the administrator has all the privileges: he or she can set the parameters for the library management and can create, change and delete librarians too.

This kind of program would fit a library like the one we have here at IES Joan Ramis i Ramis. If you are interested in do so, I could ask the library commission about the possibility of actually using one of your programs.

Specifications

All the specifications and features the program must include are clearly explained in the next section. They have been classified in four different classes:

- Compulsory specifications of type I:
  - Marked with a (I) at the beginning of the line.
  - Must be satisfied at the first deadline.
- Compulsory specifications of type II:
  - Marked with a (II) at the beginning of the line.
  - Must be satisfied at the second deadline.
- Compulsory specifications of type III:
  - Marked with a (III) at the beginning of the line.
  - Must be satisfied at the third deadline.
- Additional (non-compulsory) specifications:
  - Marked with (add) at the beginning of the line.

    - You don't need to implement them, but if you do, your mark can improve significantly.
    - You can add any other additional specification that you think that could be useful in a real world management of a library.

Detailed description of the program

I have tried to provide a very thorough description of the program I request. Nevertheless, I must confess that probably I have forgotten to mention some details.

For this reason, we are using the next few days to improve this wording if needed. I ask you to read carefully this section and to inform me about any inconsistency, contradiction or forget. I will think of it and change whatever I think I must change.

- Website:
    - The website must provide pages or tools to perform every task:
        - (I) A cover, with a logo, a menu, an area for highlights and a search bar. It has to show an area to sign in.
        - (I) A catalogue page, with search tools and results area.
        - (I) An element template to show the data about any element from the catalogue, or to introduce this data. (They can be two different pages if you prefer).
        - (I) A register page.
        - (I) A user template to show the data about any user.
        - (I) A booking and lending page. (It can be a tool in the element template or the catalogue).
        - (I) A management page, from which the work of the librarian is easily done.
    - (I) Every webpage of the website must adapt itself to the type of user that is visiting it. In particular, the navigation menu must only show the pages to where the user has granted access.
- Types of users and privileges (each level has the privileges of the previous one plus the listed below):
    - (I) Everyone (even non-members) can consult the catalogue.
    - (II) Everyone (even non-members) can become a member by filling a form.
    - (II) Members can book an available element of the catalogue for any available date.
    - (II) Members can edit its own profile information at any time.
    - (II) Librarians can add and remove members, change any member information, lend and return an element, and add and remove elements from the catalogue.

  - (III) Administrators can add and remove any kind of user and edit the parameters of the management system.
- User management:
  - (I) Hardcoded series of users to show everything works.
  - (II) Migrate user data to a DB.
  - (I) A user profile must include:
    - ID, name and surnames, ID card, telephone, e-mail, address, etc.
    - User and password
    - (II) Photograph
    - Type of user
    - Privileges (directly linked with the type of user)
    - (add) Further profile
  - (II) Automatic register and unregister of users (directly performed by users), when they are of member type.
  - (III) Other types of users only can be created by the administrator, through a special interface to which only this user has access.(
  - (II) Users can change their profiles (except protected data) at any moment after the registration.
  - (add) The system provides a mechanism for the confirmation of the e-mail users have provided.
  - (add) The system obeys the law about personal data protection.
  - (add) You can link privileges with the type of user, as proposed above, or you can treat them independently, with some default settings, and provide an interface for the administration to change them freely.
- Catalogue and catalogue management.
  - (I) The catalogue contains information about all the elements that can be found in the library (books, magazines, DVDs, etc.). For every element, it includes:
    - ID, type of element (book, magazine, etc.), title, author, subject(s), publishing company, year, edition number, ISBN, etc.
    - Level of protection (long booking, short booking, no booking, reserved...)
    - State of conservation
    - Wether it is available, booked or borrowed.
  - (I) At first deadline, this catalogue can be hardcoded (not database is needed).
  - (II) Naturally, the catalogue must be stored as a Database. You have to

provide a DB file with at least 20 different fully-informed elements, to make possible the assessment of your project.

  - (II) You have to provide a booking and borrowing interface for every user. It may change a bit from one type of user to other, in order to satisfy the specifications.
  - (II) A catalogue is a kind of living thing, to which elements are constantly added and removed. You have to provide an interface and a procedure for the librarians to do so.
    - (add) Do not completely erase from the system the removed elements. Just move them out of the catalogue to another storing place.
  - (add) It's good to listen to the suggestions of members, so it would be a good idea to provide a way for members to indicate books, magazines or other types of elements they would like to find in the library.
- Catalogue search:
  - (I) It can be searched using:
    - Title (whole or a part)
    - Author
    - Subject
    - Any other feature you think could be useful.
- Library administration:
  - (I) Form to register and unregister users (only show it, do not need to work. If it works for the session, better).
  - (II) Register and unregister form connected to the DB and working.
  - (I) Form to introduce or delete elements of the catalogue. (Again, not necessarily working).
  - (II) Once selected an element from the catalogue, any user (member, librarian or administrator) must easily know if it is available, booked, borrowed or not returned at any time.
  - (II) The system must provide a method to obtain a list of the pending returns for a certain date (today by default). With this tool, the librarian should be able to easily phone or write to every member on the list. It must be easy to indicate if any of the elements has been returned. Obviously, the date of this return must be kept and stored.
  - (add) When a return date has passed and the return has not taken place, the system automatically sends an e-mail to the user to recall him or her about the return. The sending of this mail must be stored in the database.
  - (III) History of an element: the information about who has borrowed an element, when, etc. must be included in the database. It must be easily found for the librarians.
  - (III) History of a member: the information about which books has

borrowed a member, for how long, wether he or she returns the elements in time, etc. must be included in the database too. It must be easily found for the librarians.

  - (add) If you decide to provide a way for the members to suggest the acquisition of books, you could provide a method to obtain a list of this suggestions and to indicate if they have been followed.

- Library parameters, that control some details about the way the library works.
  - (I) Use this parameters, wherever they are stored (at first deadline, hardcoded):
    - Number of elements a member can borrow at the same time.
    - Levels of protection and conservation.
    - Lending period (depending on the type of element, its level of protection and conservation)
    - Privileges of each type of user.
    - Penalty for a delay in the return (depending on the length of the delay)
  - (III) Move this information to a file or a database.
  - (add) Provide an interface for the administrator to edit this parameters.
  - (add) You could inform here about the data you want to have about users and elements, and indicate which pieces are compulsory and which are not.
- Other specifications:
  - Provide a method to overwrite the system date without having to edit the server data or restarting the system. This way it will be much easier to check if your program does what it has to do.

Delivery and deadlines

Several dates has been set:

1. **Previous deadline:** Monday, 14th November 2016, 00:00 h (that is midnight form Sunday 13th to Monday 14th)
   - You have to finish the plan for your program and deliver it before this deadline. Deliver:
     - Class diagram
     - Entity-relationship diagram
     - Navigation map

   - The delivery of all these documents is compulsory.

2. **1st deadline:** Saturday, 17th December 2016, 00:00 h (that is midnight from Friday 16th to Saturday 17th)

- Your program must fulfill the compulsory specifications of type I.
- You need to deliver:
  - The program source code.
  - A technical manual with an explanation of the functionalities you have implemented.
  - A user manual.

- The delivery of all these documents is compulsory.

- Everything must be uploaded to Moodle as a ZIP file with this name:

  NameSurname_Library(I).zip

- Compulsory specifications of type II will not be considered at this point as worthy of an improvement of your mark, but any other additional contributions that you make will be, up to the maximum established.

3. Up to Tuesday, 20th December, 21:30 h
   - Time for the teacher to ask for any detail to any student about his or her project.
4. **2nd deadline:** Monday, 23rd January 2017, 00:00 h (that is midnight from Sunday 22nd to Monday 23rd)
   - Your program must fulfill the compulsory specifications of both type I and type II.
   - You need to deliver:
     - The program source code.
     - The DB file.
     - A complete technical manual.
     - A user manual.
     - A presentation of your project, to be used for you in your public defense of it. You will have 30 minutes to perform this defense.

   - The delivery of all these documents is compulsory.

   - Everything must be uploaded to Moodle as a ZIP file with this name:

     NameSurname_Library(II).zip

   - Any additional features that you introduce in your program will be worthy of an improvement of your mark, up to the maximum established.
5. **3rd deadline:** Monday, 13th February 2017, 00:00 h (that is midnight from Sunday 12th to Monday 13th)

- Your program must fulfill the compulsory specifications of types I, II and III.
- You need to deliver:
    - The program source code.
    - The DB file.
    - A complete technical manual.
    - A user manual.
    - A presentation of your project, to be used for you in your public defense of it. You will have 30 minutes to perform this defense.

- The delivery of all these documents is compulsory.

- Everything must be uploaded to Moodle as a ZIP file with this name:
    - NameSurname_Library(II).zip
- Any additional features that you introduce in your program will be worthy of an improvement of your mark, up to the maximum established.

6. Up to Friday, 17th February 2017, 21:30 h:
    - Performing of the defenses of your projects.
    - Each defense must go on for 30 minutes. Your control of time is one of the assessment issues.
    - Each speaker will be randomly chosen right before the moment of his or her speech.
7. **Last deadline:** Saturday, 18th March 2017, 00:00 h (that is the midnight from Friday 17th March to Saturday 18th March)
    - Any improvement of your projects will be accepted up to this point.
    - Send the improvements by e-mail to vgomez@iesjoanramis.org.

Assessment Issues

The mark your projects deserve, will be formed by different parts, that will evolve from a deadline to the next:

- Previous deadline: No assessment will be performed, but if you do not deliver these files, you won't be assessed ever.
- First and second deadlines:
  - Your program: up to 8 points:
    - Functionalities: up to 5,5 points.
    - Usability and design: 2,5 points.
  - Your documentation: up to 2 points.
- Third deadline:
  - Your program: up to 8 points:
    - Functionalities: up to 6,5 points.
    - Usability and design: 1,5 points.
  - Your documentation: up to 1 point.
  - Your public defense: up to 1 point.

Moreover, you have to take into account the following statements:

*To obtain a 5 over 10 you need to completely and satisfactory fulfill all the specifications marked as compulsory for each deadline (type I for 1st deadline and so on).

*You need to get at least a 5 over 10 (as a whole) in the first delivery of your project (1st deadline) to pass the first term. This is completely independent of the other marks you have collected.

*You need to get at least a 5 over 10 (as a whole) in the last delivery of your project to pass the course. This is completely independent of the other marks you have collected.

*As is usual practice in this course, you can keep on working on your projects after these deadlines and deliver them at any other date. If your work is worthy an improvement of your mark, you will have it.

*As the final mark of the course is calculated from the set of all your marks, the most important thing is the mark you have been given at the final deadline.

*No copied works will be accepted. If the teacher suspects that your work has been performed by a third party, you will have to prove otherwise. In the meantime, your mark will be 0 and, therefore, you will not pass the course.