1st term task: develop an application in Php/mySQL

You have to develop an application to manage a Library. Students that have done it in previous courses develop another program with similar level of difficulty. If a user prefers to use postgreSQL instead of mySQL there is no problem in doing it.

You should at least allow the following:

- Two or three levels of users:
  - Librarian: Can manage users, books and reservations
  - User: Can search and reserve books, manage his/her profile.
  - Administrator: Can manage librarians (optional). If you don't do it you can insert Librarians directly to the DB.

  Every user would be able to see only the operations he/she is allowed to do. Example: Librarians can manage all users but a single user can only manage his own information.

- Registration and login system. A user from home has to be able to subscribe to the Library. You can optionally ask him/her to physically pass to the Library to sign his registration and check his DNI.
- A user has to be able to see all books from home and book one of them for a specific future date. The system has to automatically check if at least one copy of that book is available or not. Then this copy is blocked for the user. You can decide how many books can someone have at home at the same time. Ex: 5 books. It has to be done in a way easy to change (in a constant or in a configuration table of the DB).
- The book can be borrowed a fixed number of days (BORROWED_DAYS). This number should also be easy to modify, as explained previously for the number of books.
- Any user can go to the Library to physically get the reserved book. The Librarian, from her account, will register that the book is taken.
- Any user without reservation can also ask the Librarian to take NOW that book or to a future date. The librarian has to be able to check by the system if it is or not possible and do it.
- The Librarian has also to be able to check which books are taken and which ones are not returned on-time.

- Once a user returns a book (maybe before or after the defined date), the Librarian will set that into the system. The system is not responsible of users that not return the book on-time. Maybe another user is affected by this fact and the Librarian will have to apologise and call the user that still has the book.

The system should be done using Object Oriented Programming in Php, but not necessarily 100%.

You can use the Table and Form classes developed at the class.

You have to provide the teacher (language: English):

1. Program source code
2. .sql file to import a sample DB
3. Technical manual (remember passing the spell checker). A good idea is to write while you develop your program.
4. User manual (idem)

All is packed into a .zip file with your surname and uploaded to the Moodle.

Example of test that you will have to do in front of the teacher to pass the task:

1. Register a user X
2. Login as that user X
3. Login as Librarian
4. Create a book (ex: Lord of the Rings)
5. Create several copies of that books (ex: 3 copies with different state of conservation)
6. Login as User X
7. Find the book Lord of the Rings
8. Book it for July 1st (it is available)
9. Login as another user Y
10. Book it for July 1st (it is available)
11. Login as another user Z
12. Book it for July 1st (it is available)
13. The librarian will try to book the same book for another user for July 1st. (It is NOT available any copy). The system could (or not) suggest when it is available.
14. The librarian will try to reserve the same book for another user July 1st + BORROWED_DAYS (It is NOT available any copy). The system could (or not) suggest when it is available.
15. The librarian will try to reserve the same book for another user July 1st + BORROWED_DAYS + 1 (It IS available).
16. Show how a book is taken from and returned to the Library

How the practial is evaluated:

- To pass the task all material should be provided with a minimum of quality

Library with compulsory functionality providing manuals and .sql file	0-7 points depending on functionality and code Quality
User manual Quality	0-1 extra point
Technical manual Quality	0-1 extra point
Extra features	0-2 extra points

Optional features:

- Be prepared to support more languages
- Administrator user
- Improved layout
- Extras you could think about and negotiate with the teacher