**Homework assignment**

**Task: Write a simple library application in .Net c#**

**Procedural requirements:**

* Assignment will be delivered by 16:30pm, Wednesday.
* Deadline is 10am, Saturday, the same week. However, if you finish the assignment sooner, send it in faster as well. Keep in mind email services may block code attachments, so use an online service to deliver the solution package.
* Log the time spent on different types of components, ie coding, test development, documenting etc. We recommend toggl.com or similar tool to help with tracking and aggregating data, and send a detailed report along with the documentation mentioned below.
* The type or design of the user interface for the application is not in focus, so choose the approach you are most comfortable/efficient with.
* Structure the code, solution components and QA elements as you see fit. In general the solution should represent what you yourself consider as following best practices.
* Include a documentation in English with the solution, that at bare minimum includes the following:
  + Instructions what needs to be done to compile and run it.
  + Describe the technical and implementation choices you made, end provide a quick explanation as to why you made them. Include possible improvement steps if you think that they can be improved with more work.
* If you face an ambiguous situation where you would in a business setting require clarification with an analyst, make a decision on your own and document your reasoning for it, and move on.

**Functional Requirements**

* Clients are the people who use the services of a library. They borrow and return books. They have a name and a date of birth. They may be denied to borrow more books if they have 3 books already borrowed (no more than one book for untrustworthy clients), or if any of the books they have borrowed is over it’s due to return date. A client may be refused a book as well, if the client’s age is younger than the age requirement for a book. The Client will be flagged to “Untrustworthy” after 3 times of overdue return dates.
* Books have a title, name, and an identifier code that represents a shelf number and a number which represents the position of the book on the assigned shelf (ie 10-19 - 10 being the shelf number, and 19 being the 19th book on the shelf). Books also have an optional age limit and an attribute that describes maximum number days a book can be borrowed out for.
* Library also has shelves which have numbers and a description as to where they are placed in the library. Each shelf houses maximum 20 books.
* User can perform standard CRUD operations on clients, books and shelves. In addition the user can register the books given out and returned by clients.
* If you feel that some checks/functionality should be added to the scope, you are free to do so, but please document your logic why and how you got to that.