Code review

Overview

This code is a part of our connection with database class. This part is fully connected with tests (assertions). We use LinQ to SQL and Azure server in our implementation.

Goals

Decrease amount of features and make it more flexible (and may be make them fully covering).

Usage Scenarios

Helpful functions for tests.

Class Diagrams

(here is not really necessarily)

Code

Categories for review

1. Design decisions

[1.1] Reviewer1: 14, 153 lines. There may be a collision of users with the same name. I think we need to add 1 more test property. maybe mail or something else.
[1.1.1] Owner: It's exactly so, but this methods just for tests, so if you remember in test we use just name of the users, but not email or something like that.
[1.1.2] Reviewer1: I am done.
[1.1.3] Owner: Thank you for comments. Discussed and [CLOSED]
[1.2.1] Reviewer2: line 153. It's not quite clear which name we should put in this function. First name, last name or the full name.
[1.2.2] Owner: In our implementation we use only first name of the user in registration, but anyway good comment. [CLOSED]
[1.3.1] Reviewer2: line 137-143. Not clear which type user gets after an upgrade
[1.3.2] Owner: It's internal method for testing (for example upgrade from student to faculty). [CLOSED]
[1.4.1] Reviewer2: line 13. I think this function is not necessary because we don't usually need to know patron's line in table.

[1.4.2] Owner: I think this is old function that was used in old implementation, but thank you for comment [CLOSED]

2. API design

3. Architecture (including inheritance hierarchy, if any)

[3.1] Reviewer3: this code uses **different types of users as fields of objects of one class**, i think it is better to have different classes for every user type - increase flexibility of code
[3.1.1] Owner: Cannot understand your question, by your comment it's part of polymorphism technique. Clarify your question please.
[3.1.2] Reviewer3: I mean that it is better to have different classes for every user type (STUDENT, PROFESSOR, LIBRARIAN etc) than just using field userType of objects of one class.
[3.1.3] Owner: Maybe you didn't understand that's query in BD [CLOSED]

4. Implementation techniques, in particular choice of data structures and algorithms

5. Exceptions handling - Contracts

[5.1] Reviewer4: Are there any contracts in code to control it's condition?
[5.1.1] Owner: It's functions for check conditions in other class.
[5.1.2] [CLOSED]
[5.2] Reviewer4: A good idea would be to add exceptions in the code that prevent the null input to the method.
[5.2.1] Owner: Please more concrete. [CLOSED]
[5.3] Reviewer3: Methods that uses linq to sql can produce exceptions and leave code in "broken" state. We need contracts/assertions.
[5.3.1] Owner: Not really clear where it could be produced [TO-BE-IMPLEMENTED]

6. Programming style, names

[6.1] Reviewer1: line 137. "UpgradeUser" has argument "ut". It is unclear name for argument.
[6.1.1] Owner: It's c - style don't repeat same name, if you can see in the next line we increase usertype.
[6.1.2] Reviewer1: Principle of programming: Use abstraction for simplicity. If I would be a client of your
code, I can't look at code inside your methods. And I will no know how to use it.
[6.1.3] Anyway it's discussable question [TO-BE-IMPLEMENTED]
[6.2] Reviewer5: line 6, 33, 44, 74, 82. This is not entirely clear why does field BookID called as it is. According to the code it may contain not only ID of some book, but any other document too (e.g. article or av material)
[6.2.1] Owner: It's probably lie on old implementation (when we have only books), but it's reasonable question. We will change it in DB. Thank you for your comment. [TO-BE-IMPLEMENTED]
[6.3] Reviewer4: Not clear why there's an argument called "test" in method "GetUserFineForDoc", also no comments to explain it
[6.3.1] Owner: This is because we use foreach query for test existence of this book in user checkouts. Maybe we should write comments for it. Thank you for your comment.

[6.3.2] Discussed and [TO-BE-IMPLEMENTED]
[6.4] Reviewer1: line 137. Why the name of feature is "upgrade"? We assume that admin can decrease privileges of librarians. I think it should be "**update**".
[6.4.1] Owner: You are absolutely right, and we have this method, but because of difficult signature and high probability of mistake in update, we use upgrade, just for.up user in some level (faculty, VP).

7. Comments and documentation

[7.1] Reviewer1: Why some comments was made on russian? And big part of methods doesn't have documentation or comments.
[7.1.1] Owner: Let's discuss this issue during the meeting [CLOSED]
[7.2] Reviewer4: There is no clear structure of comments, it is not clear what and what it refers to.
[7.2.1] Owner: Read previous comments in this section. What is unclear (about English comments)?
[7.2.2] Reviewer4: The point is that the comments are not structured (Even in russian).
[7.2.3] Owner: May be you didn't understand C# language [CLOSED]
[7.3] Reviewer4: Misuse of TODO
[7.3.1] Owner: Read previous comments in this section.[CLOSED]