Template code review

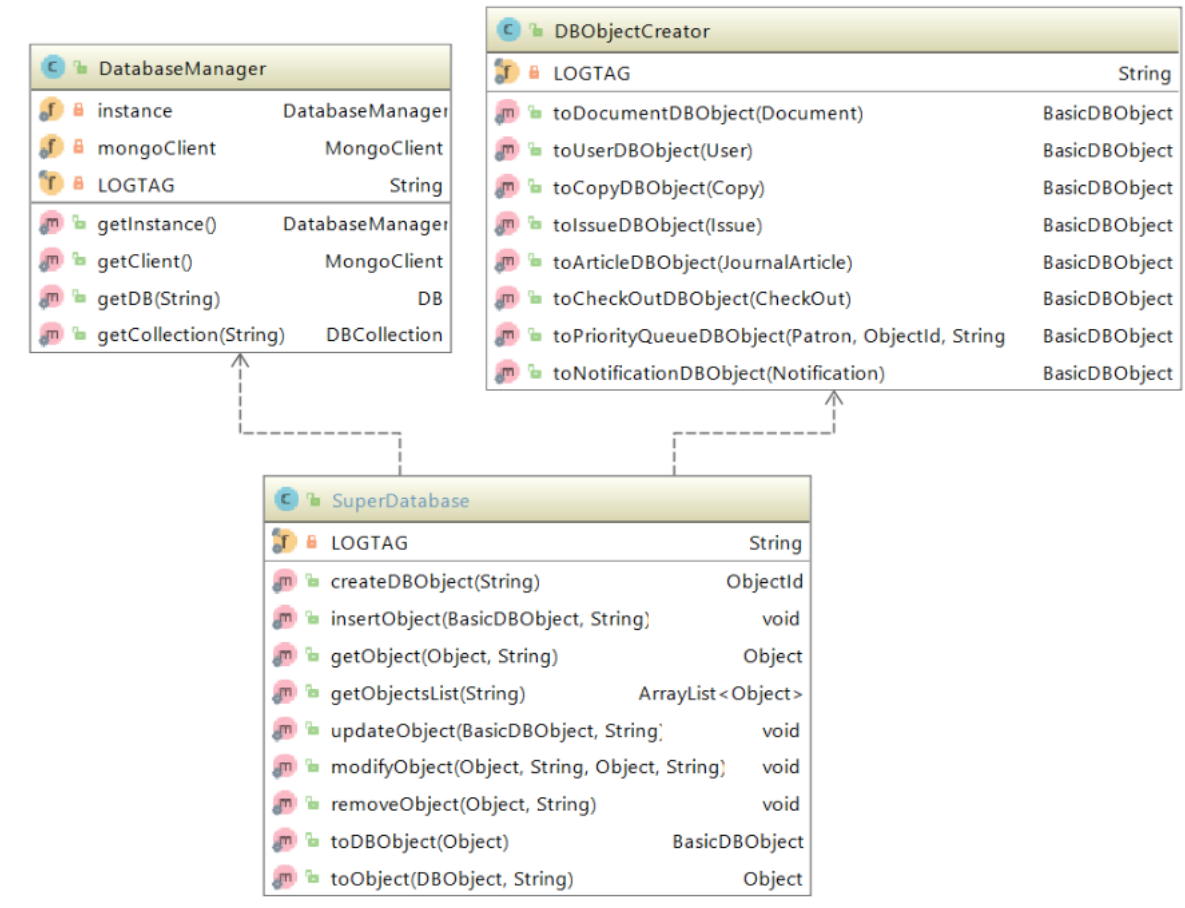
Overview

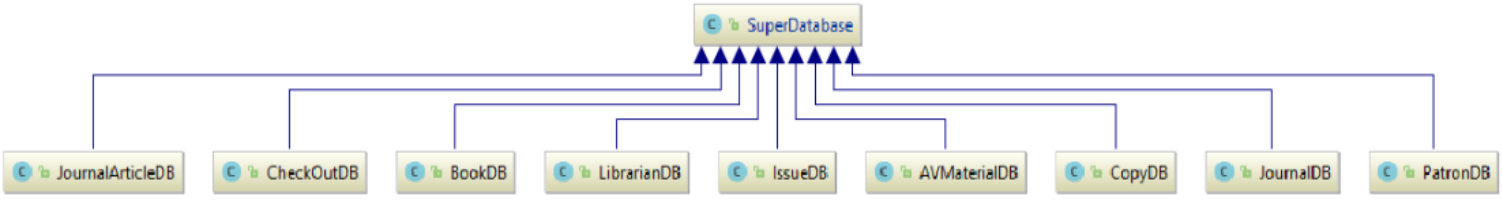
The piece of code is used to make basic queries to the MongoDB database. To access collections of the database the class addresses to the DatabaseManager class. The methods supported creation, insertion, get, update, modification, deletion operations on objects of classes: Book, Journal, AVMaterial, CheckOut, Copy, Issue, JournalArticle, Librarian, Patron.

Goals

Implementation of the usable interface between the system and other database classes and the actual database

Class Diagrams





Code

Categories for review

1. Design decisions

2. API design

3. Architecture (including inheritance hierarchy, if any)

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

5. Exceptions handling - Contracts

[5.1] Reviewer1: you often use DatabaseManager.getCollection(collectionName), but you do not check whether the collection with “collectionName” exist

[5.1.1] Owner: Thank you for your feedback. I realised that it may cause some problems like creation of new collections that are irrelevant or even more write/read some object in a different collection. I will fix it by implementing another method that will check the collectionName at the beginning of each method and will put logs to throw exception. Do you agree with such solution? Also, don’t you think that we could move this issue to the category 5 Exceptions handling - Contracts?

[5.1.2] Reviewer1: 1.I think it is a good solution of the problem
2.Yes

Comment: *The issue was moved to category 5 from category 4.* [CLOSED]

[5.2] Reviewer3: There is a possible case when we have correct collection name but there is no such ID in the collection. It wouldn`t break anything perhaps, but I guess it should be shown somehow that such problem occured.

[5.2.1] Owner: Thank you for your comment. Yes, indeed, such situation might occur when an id could not be found. However, no exceptions or error will be thrown since MongoDB would just ignore ids that do not exist in a collection. Anyway, I agree that it would be helpful to write some logs informing about such situation. I will implement a pre-conditions checking existence of the id in the collection. [CLOSED]

[5.3] Reviewer4: In function toDBObject, Argument object can be null, but you haven’t check it.

[5.3.1] Owner: Yes, you are right. However, it will not cause any bugs because conditions of form “object instanceof Class” won’t violate any exception and the last ‘else’ condition will be run. But it might be good to put a log in that place. [CLOSED]

6. Programming style, names

7. Comments and documentation

[7.1] Reviewer5: at the line 155 in class SuperDatabase.java should be written “In case” instead of “Incase” - this is grammatically incorrect;

[7.1.1] Owner: Thank you, I will fix it! [CLOSED]

[7.2] Reviewer1: Please, try to write comments shorter, they just should briefly explain the features of the class

[7.2.1] Owner: Some developers would not agree with you because usually comments, especially Java Docs, should explicitly explain the method. For example, if you open the classes from java libraries you could see tens of lines of comments for each method. However, I would agree that my comments might be not

understandable. So, if you agree I would give more better explanation for some methods. Could you say which methods' comments are not well-explained?
[7.2.2] Reviewer1: your comments are understandable, but it is very difficult to read them and do not lose your point of view
[7.2.3] Owner: I think that I need to ask you to explain your opinion with more details because I cannot understand in which way I could improve my comments. Could you give some examples what should be changed or how it could look better? [TO BE DISCUSSED]