Template code review

Overview

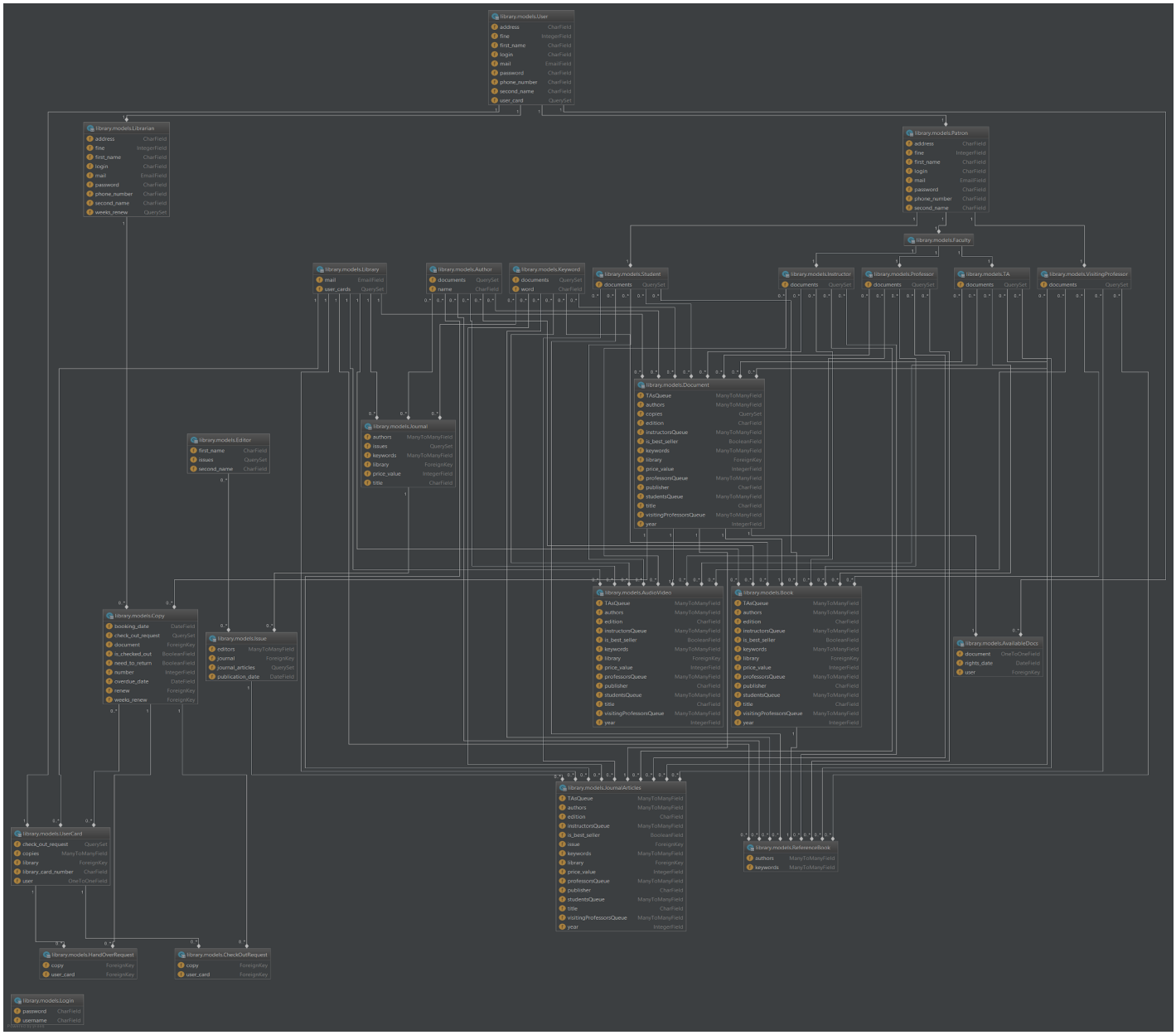
There was used Django framework for the project. The Python code below is about basic representation of the main classes (e.g. User, Document and UserCard) for making a functional backend.

Goals

Correct every debatable LOC. Find bugs in the system and make the code ability more efficient.

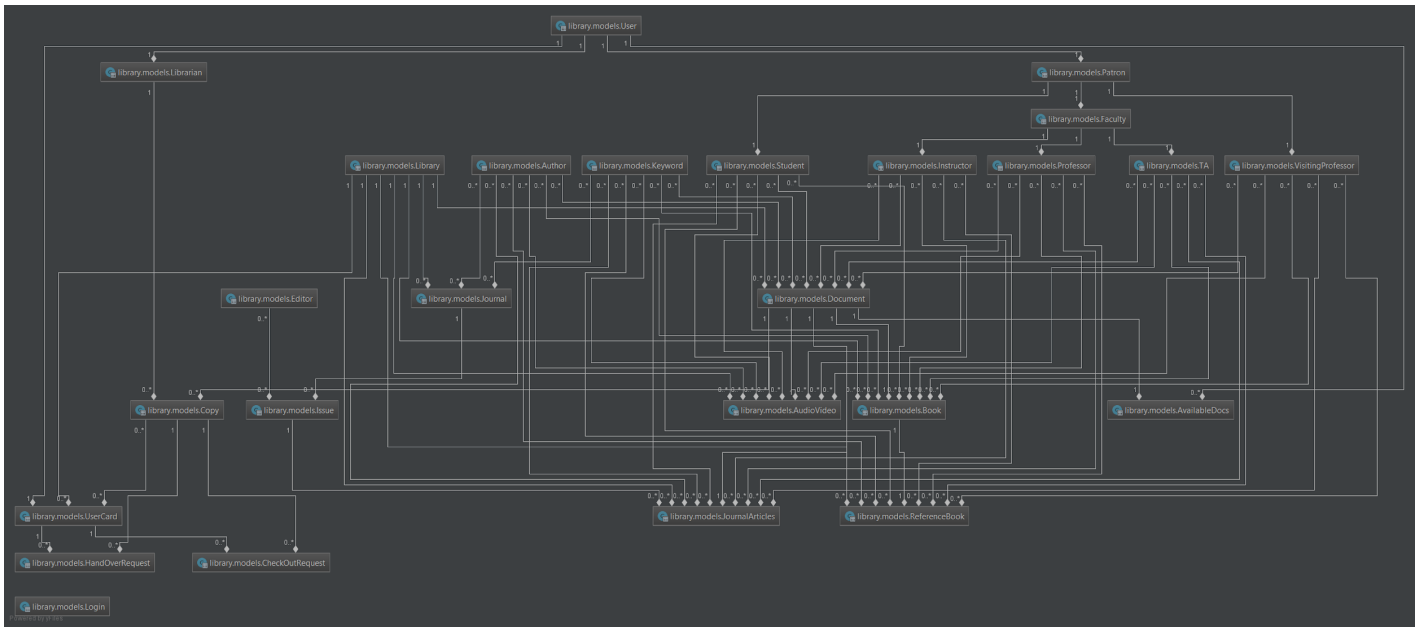
Usage Scenarios

Being the part of the main classes in the whole project it is used for making basic operations in library from creating the user till the booking the document. Looking at the diagram, it is possible to claim that there is a strong cohesion between classes. They are necessary for the basis of sqlite3 and creating the model Library for the further web-representation of the project.



Class Diagrams

Class structure of the whole project which reflexed in the database. There you can see all connections and relations between classes in models.



Code

Making review

Please, go to the link xxx for writing all your comments. There is some example for the review:

1. Design

[1.1] NameTTT: You should use HASH_TABLE for 'yyy' instead of ARRAY.

[1.1.1] Owner: the decision of using ARRAY is based on the fact that ARRAYs are lighter than HASH_TABLES

[1.1.2] NameTTT: That might be true, but you are compromising execution time, 'yyy' is used to perform several searches. Having a HASH_TABLE one can have constant complexity in searching

[1.1.3] Owner: Ok, I will change the design and implementation. [CLOSED]

2. Implementation

[2.1.] NameTTT: It seems unnecessary to pass `a_block_type' argument into `end_process_block', since it's always the top of the block stack. It seems a bit of an overkill to me to introduce this redundancy just to do the check (maybe clients could do the check instead)...

[2.1.1] Owner: Let us discuss this issue during the meeting [TO-BE-DISCUSSED]

Categories for review

1. Design decisions (5)

[1.1] Reviewer1: Line 260 What's doing this method? Not setters, no return value, no calls in code

[1.1.1] Owner: You are absolutely right. Moreover, this method is never going to be used at the whole project. I guess it was created with some ideas of implementation which was not integrated into it. I would simply delete it.

[1.1.2] Reviewer1: Ok, it's solved [TO BE FIXED]

[1.2] Reviewer2: Lines 31-35: There is no need to make 5 queues for each document. The users should have priorities and you need to create only one queue for the document based on the users' priorities.

[1.2.1] Reviewer1: I agree, it decreases scalability of your system in case of adding user types

[1.2.2] Owner: Yes, I totally agree with you. It was only in order to save time before the delivery, because your approach demands the really priority queue implementation instead of "imitation" it. [TO BE IMPLEMENTED]

[1.3] Reviewer3: A good habit is created a special variable for long boolean (and not all) equation which tells for programmist that it means
Like on line 206: " if document.studentsQueue.count() + document.TAsQueue.count() + document.instructorsQueue.count() + document.visitingProfessorsQueue.count() + document.professorsQueue.count() > 0:" maybe better create variable 'has_queue' or 'queue_not_empty' or smth else

[1.3.1] Owner: Yes, you are right. It is really fascinating habit. Will fix it. [TO BE IMPLEMENTED]

[1.4] Reviewer3: library_card_number = models.CharField(max_length=128)
What guarantied unique of library_card_number?

[1.4.1] Owner: Nothing and nobody, except of librarian.

[1.4.2] Reviewer3: But librarian is a human and human can make mistakes. What will you do if a librarian makes two users with identity card number?

[1.4.3] Owner: Ok-ok, I can write a special method for generating unique card_numbers or something like this. [CLOSED]

[1.5] Reviewer3: Queue implementation not modularity. If we want to add a new type of patron we need create a new queue and add conditions in all methods that use a queue.

[1.5.1] Reviewer3: Save priority in the dictionary with different values like
priority = ['Student': 100, 'TA': 80, 'Instructor': 60, 'Visiting Professor': 40, 'Professor': 20]. It's not all implementations of queue but sketch

[1.5.2] Owner: Accepted, a useful observation [TO BE IMPLEMENTED]

2. API design (5)

[2.1] Reviewer1: Line 141: I think that "fine" is more related to UserCard class, than to User

[2.1.1] Owner: Yes, there will no be problems to handle it in the whole project. Thanks [TO BE IMPLEMENTED]

[2.2] Reviewer2: Line 27: "Document" class should not contain field "is_best_seller".

This field is related to "Book" class.
[2.2.1] Owner: Thank you for the question. It is really strange that this field appeared exactly in the "Document" class, not in the "Book" one. I will correct it [TO BE FIXED]
[2.3] Reviewer3: Save special values in constant. Strings ('Student', 'TA' etc), integers (number of days for check out etc) and all custom parameters because if you want to change it easier modify one constant than search all suggestions in whole project
[2.3.1] Owner: If you insist on it, I will correct it. But for me it mostly does not matter
[2.3.2] Reviewer3: It is not obligatory just a recommendation
[2.3.4] Owner: Anyway, thank you [TO-BE-FIXED]
[2.4] Reviewer3: The class document requires that all documents need parameteres is_besteseller, edition, publisher, year and more but different types of documnets has different parameters in modularity system
[2.4.1] Owner: Yes-yes, it has already been notified. Will fix it [TO BE IMPLEMENTED]
[2.5] Reviewer3: You have not reference journals, but requirements says "The library also has reference books and magazines, which cannot be
checked out"
[2.5.1] Owner: Could you identify the difference between reference magazines and the reference books?
[2.5.2] Reviewer3: I think that magazine is a journal, not a book but maybe I wrong
[2.5.3] Owner: Ahah, ok. If it does matter for you, I will fix it. I mean, that I can simply create the reference literature class instead of all these ones. [TO BE IMPLEMENTED]

3. Architecture (including inheritance hierarchy, if any) (1)

[3.1] Reviewer2: Line 93: The class "Journal" should inherit from the class "Document".
[3.1.1] Owner: Before the decision to fix it or no: the reason of such implementation is only about special cases for journals which is impossible to book. Am I right or no?
[3.1.2] Reviewer1: Reference book is not available to book too, but it do inherit from class Book, which is a child of Document
[3.1.3] Owner: Yes, there are really problems about the logic of implementation. Will fix it. [TO BE FIXED]

4. Implementation techniques, in particular choice of data structures and algorithms (7)

[4.1] Reviewer1: Line 252: Returning of "VisitingProfessor" makes whole function not applicable for representing user type in interface
[4.1.1] Owner: Do you mean the returning of the String type object makes the function not applicable?
[4.1.2] Reviewer1: Returning "VisitingProfessor" without space between words make this function not applicable to pass value in user interface. I suggest you just add one space
[4.1.3] Owner: Oh, excuse me, I even did not notice that fact. Yes. of course, I will add the space beetween these two words.
[4.1.4] Reviewer1: Perfect, reusability is moving up! [TO BE FIXED]

[4.2] Reviewer1: Line: 209&210 and 211&212 Is it possible to change a loop on exact query to database? Search by id, for example. Same situation with lines 209 and 210.
[4.2.1] Owner: It is very interesting question. By the way, will it be more efficiently about the memory or execution time? Because to find something in the database is all about cycle, which is looking for the object. No?
[4.2.2] Reviewer1: If you can write better than the creators of database, then loop is better. Otherwise internal search will reduce amount of input data from database and slightly increase speed. And in the first case if your database will contain great count of documents and copies, memory will be eaten
[4.2.3] Owner: Ok, I have caught the idea. The query to the database is much better. [TO BE FIXED]
[4.3] Reviewer3: In class UserCard why library_card_number has type string and max_length 128 symbols? Usually, the number is a number(integer or long)
[4.3.1] Owner: Maybe, there will be the card with both numbers and letters. Although, you are right. It does not matter so much, but I can fix it [CLOSED]
[4.4] Reviewer3: n method return_doc in class Patron do you notify user in queue (if queue exist) that now copy of document is availible to check out?
[4.4.1] Owner: Yes. [CLOSED]

[4.5] Reviewer3: In class Patron exist method has_overdue but not exist a method that calculates overdue (The overdue fine is a hundred rubles per item per day, but cannot be higher than the value of the overdue item)
[4.5.1] Owner: It calculates via database methods, but i can rewrite it to a special function. [CLOSED]
[4.6] Reviewer3: Can't find a method how the librarian can see who check out concrete copy of the document ("For each copy, we need to know whether it is currently checked out and by whom (c) Requirements)
[4.6.1] Owner: It executes via database methods, but i can rewrite it to a special function. [CLOSED]
[4.7] Reviewer3: Renew method violence requirements, because not checked that renew can do only once (except visiting professor). And method don't change overdue_date
[4.7.1] Owner: Accepted [TO BE IMPLEMENTED]

5. Exceptions handling - Contracts (3)

[**5.1] Reviewer1: Line 163: Here is possibility, that "first_in_queue" returns None, isn't it? That may fail execution of lines 164&165**
[5.1.1] Owner: I guess there will be an exception (otherwise - none returning). I will handle it, thanks [TO BE IMPLEMENTED]
[5.2] Reviewer3: In class book, when book is bestseller and faculty check out it his booking period = 2 weeks, but requiments says what "Books are checked out for three weeks, unless they are checked out by a faculty member, in which case the limit is 4 weeks (regardless the book is best seller)"
[5.2.1] Owner: Ok, thank you for the correction. Will fix it [TO BE IMPLEMENTED]

[5.3] Reviewer3: Copies are stored in a certain place inside the library, e.g., a room, level. (c) Professor Rivera
[5.3.1] Owner: Then could you suggest the implementation for it?
[5.3.2] Reviewer3: Just add one more parameter for copy of document - shelf
[5.3.3] Owner: Ok, thanks. Will fix it. [TO BE IMPLEMENTED]

6. Programming style, names (7)

[6.1] Reviewer1: Line 174: "pass" is useless
[6.1.1] Owner: Yes, let's consider it to be a typo. Will delete it [CLOSED]
[6.2] Reviewer1: Line 138: Second name in English have difference from last name or surname, rename it please
[6.2.1] Owner: I do not believe the difference to be significant for correcting. However, since the renaming is not difficult, I can do it.
[6.2.2] Reviewer1: You can ask others, by the way, and change or not change name of this variable according to them
[6.2.3] Owner: I have asked. The surname is more used. I will rename it.
[6.2.4] Reviewer1: Solved [TO BE FIXED]
[6.3] Reviewer4: According to PEP8 programming style documentation for python the max length of line should be less or equal to 79 symbols in line. Here the lines where this rule disarranged:

- **library = models.ForeignKey(Library, on_delete=models.DO_NOTHING, related_name='documents')**
- **instructorsQueue = models.ManyToManyField("Instructor", related_name='documents')**
- **visitingProfessorsQueue = models.ManyToManyField("VisitingProfessor", related_name='documents')**
- **professorsQueue = models.ManyToManyField("Professor", related_name='documents')**
- **library = models.ForeignKey(Library, on_delete=models.DO_NOTHING, related_name='journals')**
- **journal = models.ForeignKey(Journal, on_delete=models.DO_NOTHING, related_name='issues')**
- **issue = models.ForeignKey(Issue, on_delete=models.DO_NOTHING, related_name='journal_articles')**
- **document = models.ForeignKey(Document, on_delete=models.DO_NOTHING, related_name='copies')**
- **renew = models.ForeignKey("Librarian", on_delete=models.DO_NOTHING, related_name='renew', null=True)**
- **weeks_renew = models.ForeignKey("Librarian", on_delete=models.DO_NOTHING, related_name='weeks_renew', null=True)**
- **user = models.OneToOneField(User, on_delete=models.DO_NOTHING, related_name='user_card')**

- **library = models.ForeignKey(Library, on_delete=models.DO_NOTHING, related_name='user_cards')**
- **user = models.ForeignKey(User, on_delete=models.DO_NOTHING, related_name='available_documents')**
- **self.user.available_documents.exclude(document=self.document, user=self.user)**
- **first.available_documents.create(document=self.document, user=first)**
- **documents = self.user_card.library.documents.filter(authors__name__contains=name).distinct()**
- **documents = self.user_card.library.documents.filter(title__contains=name).distinct()**
- **documents = self.user_card.library.documents.filter(keywords__word__in=words).distinct()**
- **if document.studentsQueue.count() + document.TAsQueue.count() + document.instructorsQueue.count() + document.visitingProfessorsQueue.count() + document.professorsQueue.count() > 0:**
- **CheckOutRequest.objects.create(user_card=self.user_card, copy=copy)**
- **# return copy of the document to the library. If it is not possible returns False**
- **HandOverRequest.objects.create(user_card=self.user_card, copy=copy)**

[6.3.1] Owner: Thank you for paying attention to it. Will correct it [TO BE FIXED]

[6.4] Reviewer4: According to PEP8 programming style documentation for python importing of libraries should be in that order:

1) Import of standard libraries

2) Import of other libraries

3) Import modules of current projects

!!!! Each import group should be separated by empty line

BUT you have:

- **from library.models import ***
- **import datetime**
- **import re**
- **from django.db import models**
- **from django.utils.timezone import now**

[6.4.1] Reviewer1: Also this horizontal scrollbar on Gitlab is awkward due to big lines

[6.4.2] Owner: Ok-ok, will fix it [CLOSED]

[6.5] Reviewer4: According to PEP8 programming style documentation you should avoid inline comments that explain obvious things or didn't explain nothing (def renew(self, queue=Copy.renew): # 13 requirement

[6.5.1] Reviewer1: If this is for internal usage, why not: understandable and does make a reference

[6.5.2] Owner: Would agree with Ilya. [CLOSED]

[6.6] Reviewer3: Maybe a better name for the function if_overdue is is_overdue because 'is_smth' name usually is used for functions that return a boolean value

[6.6.1] Owner: Ok, yes, it is important notes. I will correct it [TO BE FIXED]

[6.7] Reviewer3: Maybe a better name for the function 'overdue' in class Copy is 'time_of_overdue' because it return difference between current date and overdue_date in time format
[6.7.1] Owner: Ok, you are right. Will fix it [TO BE FIXED]

7. Comments and documentation (2)

[7.1] Reviewer2: Line 21: The class "Document" has not any comments or documentation
[7.1.1] Owner: Since Professor Rivera asked owners to represent the code with no more than 200 lines, I simply did not add some pieces of documentation on it. I can do it for you in order to make the code understanding better. [CLOSED]
[7.2] Reviewer4: Some functions aren't commented:
def booking_period(self, user):

def queue_type(self, doc, user):

def first_in_queue(self, doc):

def booking_period(self, user):

def booking_period(self, user):

def booking_period(self, user):

def if_overdue(self):

def overdue(self):

def check_date(self):

def has_overdue(self):

def type(self):

def faculty_card(self, login, password, name):

[7.2.1] Owner: Yes, will fix it [TO BE FIXED]