LynxLMS

Introduction

LynxLMS is a library management system developed as a course project for Introduction to Programming II course (BS1 S18). This project uses Python 3.6.3 and PyQT 5 (Graphical User Interface), Peewee 2.10.2 (Object-Relational Mapping) frameworks.

Installation

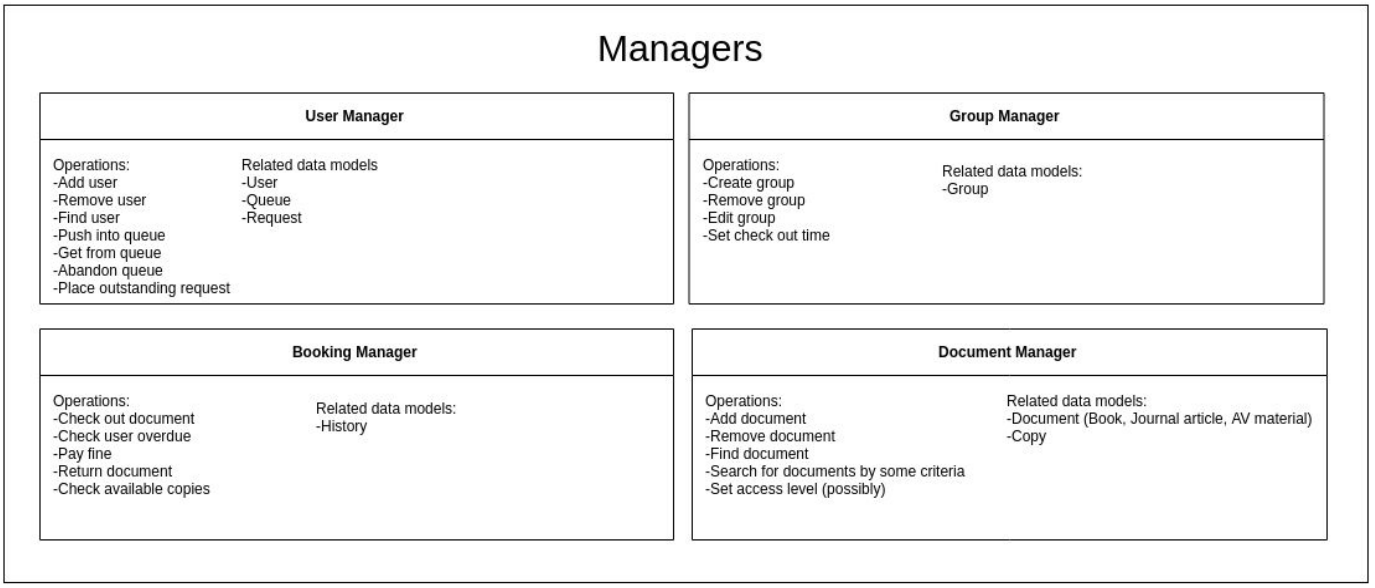
Download and install Python 3.6.3 (or higher). Clone or download all source files from github (https://github.com/Lgmrszd/LynxLMS) and install requirements using command

```
python3 -m pip install -r requirements.txt
```

To run application simply run main.py

Working principles

We have distributed all operations on database through 4 managers



User manager performs operations on users, group manager on groups and document manager on documents and copies, also it deals with priority queue for documents and with outstanding requests. Booking

manager (or booking system) performs such operations as check out document copy, return copy, pay fine and renew. Also this module allows user to view check out history and check available copies of certain document. Several important points about working principles of specific parts of project are presented below.

Documents

- Document has several copies.
- Librarian is able to check out certain copy to certain user.
- It is possible to add copy only if information of document is already in system.
- If document has no copies or all of them are deleted, document is considered as deleted.
- It is not possible to delete information about document or copy completely from system (for librarian), but they can be marked as removed and then they cannot be checked out.

Users

- As for documents it is not possible to remove user from system completely, but it is possible to change group of user to ‘Deleted’ and then check out will be forbidden for this user.

Queue

- If there is now any available copy of certain document left, user is placed in queue
- If someone returns or adds copy, user with highest priority will receive this copy
- The queue can be abandoned, all users will be notified

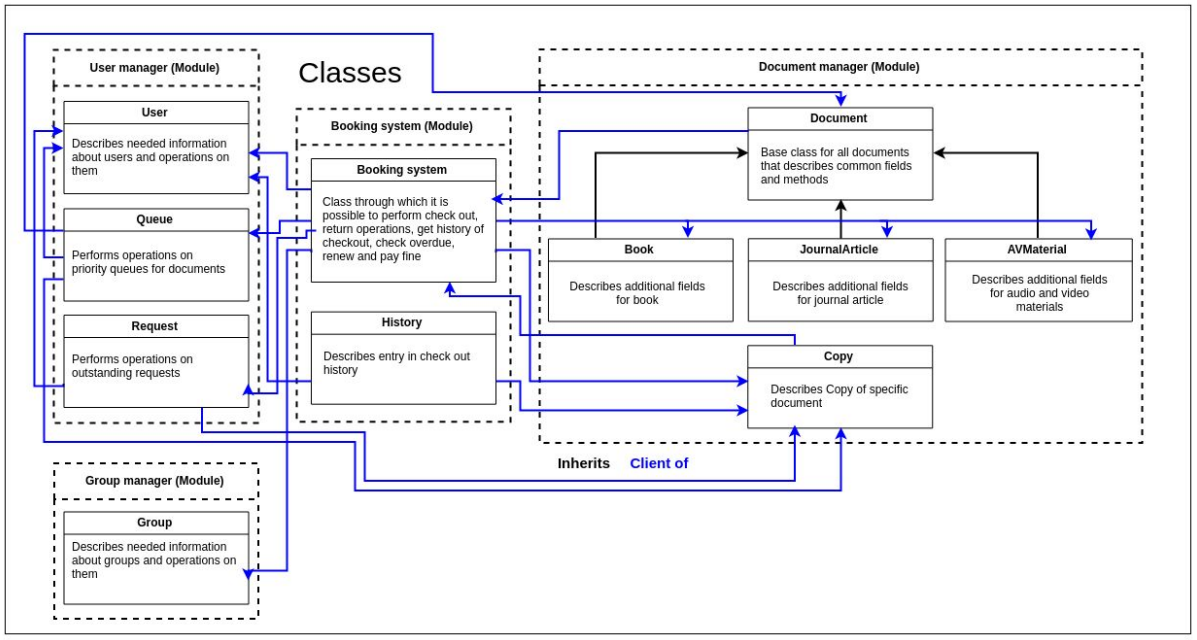
Outstanding request

- If there is now available copy for certain document left, librarian can place an outstanding request for specific user and this document.
- After request is placed, queue for the document abandoned and no one can renew this copy during the request.
- Request cancels when user for whom this request was placed receive copy of document.
- First available copy will be checked out to user immediately.

Check out history

- Every check out and return operations are written in history.
- It is not possible to edit or remove any information from check out history using LMS.

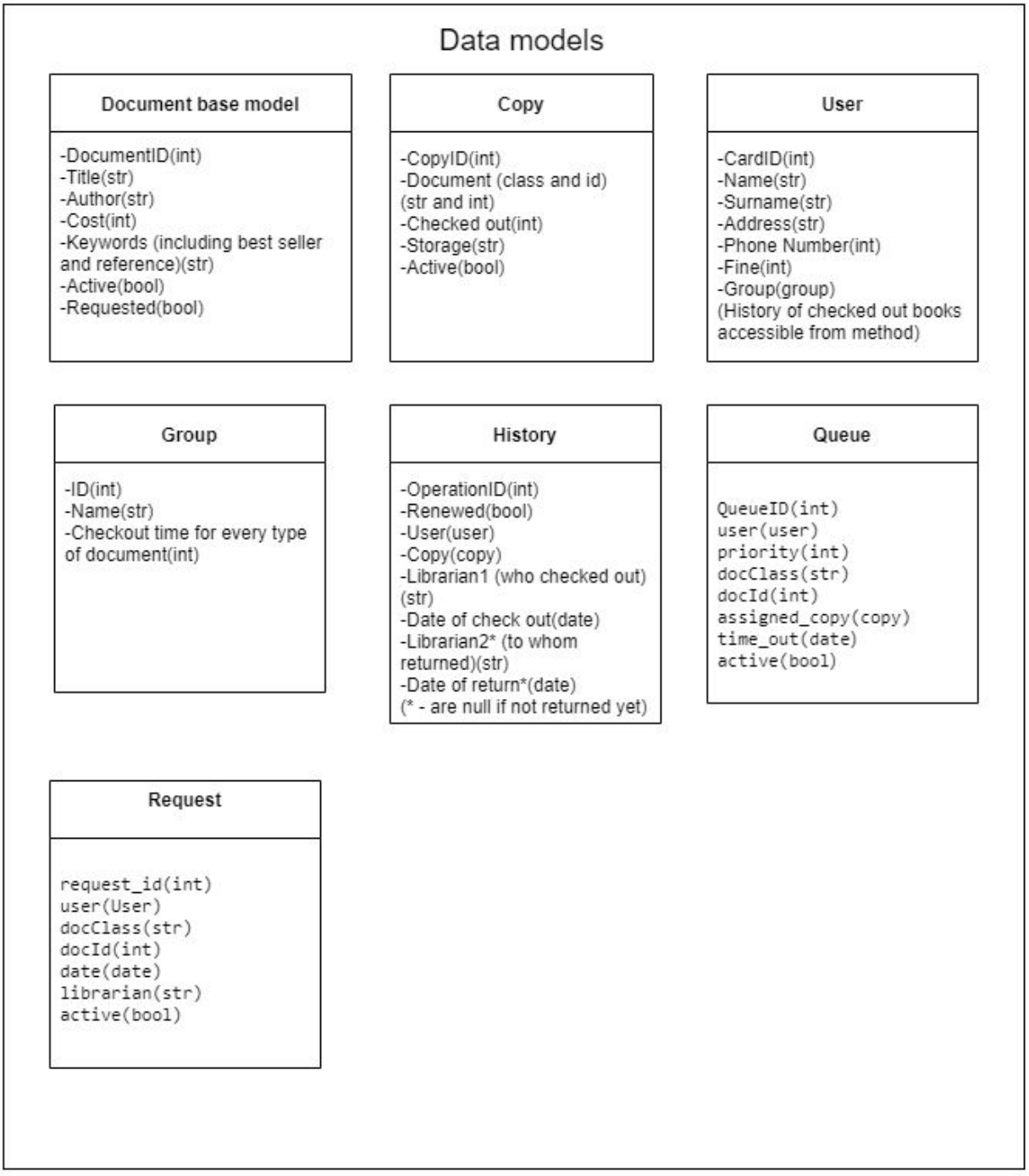
Representation of classes



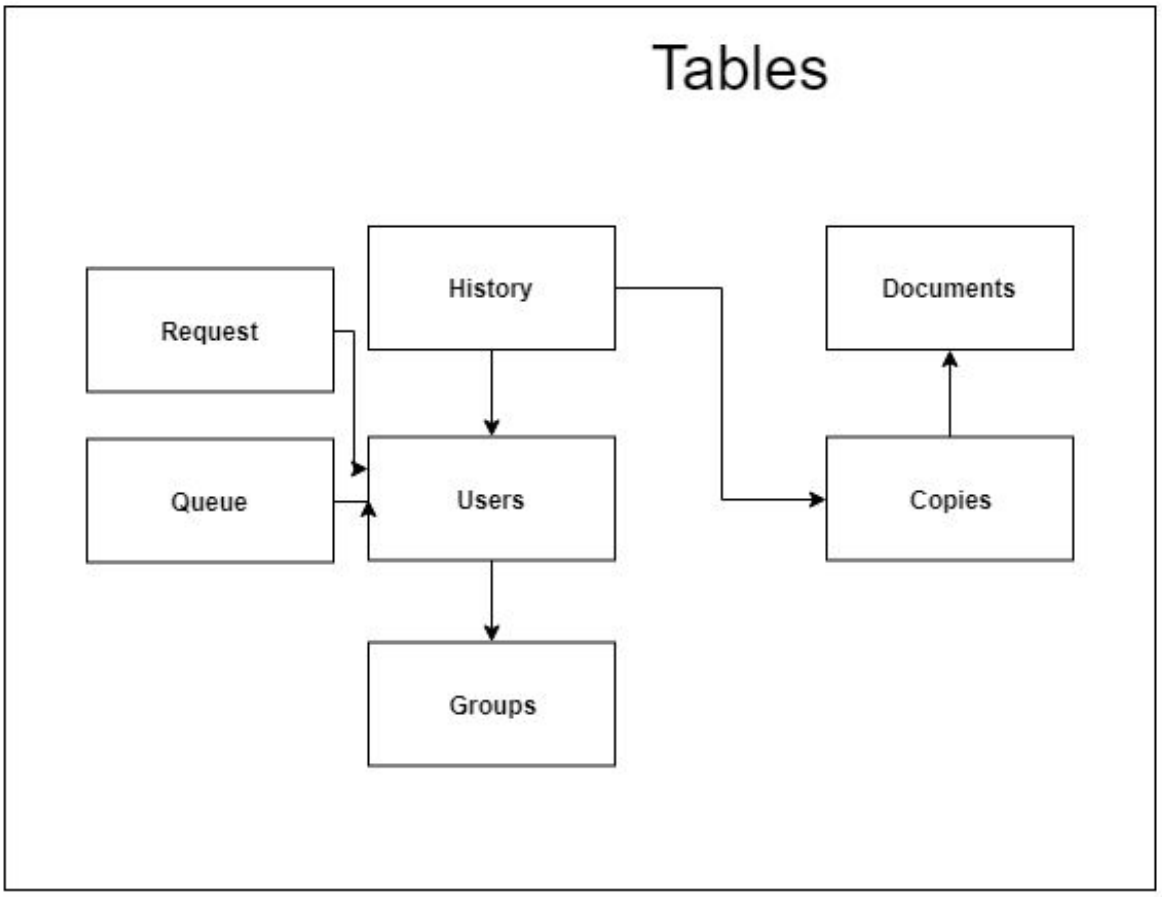
Every document type is represented as separate class that inherits from common document class.

Database

In this project SQLite is used as a primary database. All data is represented in several data models.



Copies, users, groups, history and each document type (excluding base model) are represented in separate tables. Connection between tables is shown in diagram below



Final Notes

If you have any issue with this application, please report about it using Github. Also, if you want to contact us, use email a.antonov@innopolis.ru .
Relevant version of documentation available on github or google docs (https://docs.google.com/document/d/1CmiCC5MhD1s7IM_LnhWjaoEQb3ODOV5LDWXsXAB5TcE/edit?usp=sharing)