Book Management System (BMS)

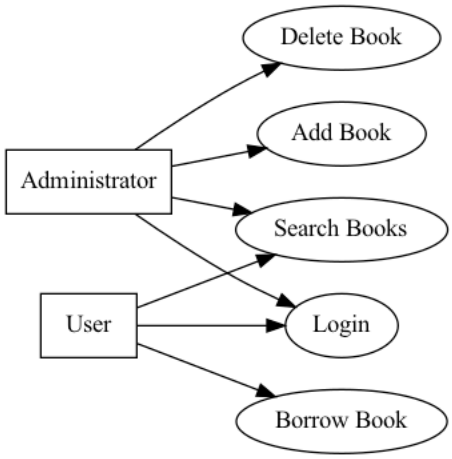
BMS is a robust application designed to manage books and users efficiently. Built with Python3 and SQLite, the system provides role-based functionality for administrators and users. Administrators can add, delete, and search for books. Users can search for books, borrow books. The system incorporates authentication, ensuring secure login for both roles.
The BMS offers a command-line interface for interaction and uses PEP 8-compliant, maintainable code to ensure high-quality software practices.

This project has these .py files and .txt files.

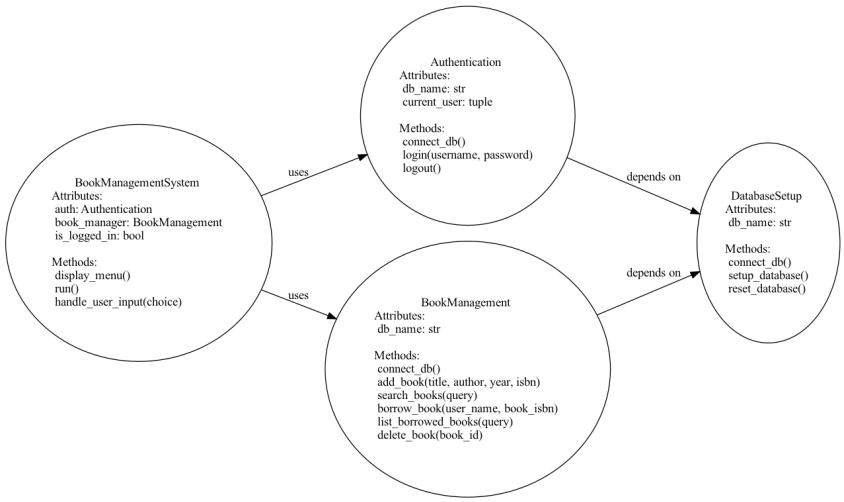
1. database.py
2. authentication.py
3. book_management.py
4. main.py
5. Users.txt
6. Books.txt

The use case and UML diagram for this application:

1. **Use Case**



2. UML diagram



Assumptions:

1. Assume that all users will register manually at the library, they won't be able to register using this system. They can only login, logout, search for books, borrow books, admins can add books.
2. I have two .txt files one is users.txt and the other one is books.txt, where the users.txt has users info - user_name, password for the system and their user_type, are they admin or user. The books.txt has title, author, year, isbn.

Steps to run the program:

1. First run database.py to create the database. A database with the filename book_management.db is created. This has two tables, users and books. The users table has attributes: id, user_name, password, user_type. The books table has attributes: id, title, author, year, isbn.

```
(base) pushwitha@Pushwithas-MacBook-Pro BMS % python3 database.py
Database setup complete.
(base) pushwitha@Pushwithas-MacBook-Pro BMS %
```

2. To check the contents of the table, I print the contents of the database:

```
Contents of table 'Users':
id, username, password, user_type
(1, 'pushwitha', 'test123', 'admin')
(2, 'shashank', 'test234', 'user')
(3, 'jnanashree', 'test345', 'user')
(4, 'mala', 'test456', 'user')
(5, 'rathna', 'test567', 'admin')
```

```
Contents of table 'Books':
id, title, author, year, isbn
(1, 'Artificial Intelligence: A Guide to Intelligent Systems', 'Michael Negnevitsky', 2005, '9780321204660
')
(2, 'Superintelligence: Paths Dangers Strategies', 'Nick Bostrom', 2014, '9780198739838')
(3, 'Human Compatible: Artificial Intelligence and the Problem of Control', 'Stuart Russell', 2019, '97805
25557999')
(4, 'Life 3.0: Being Human in the Age of Artificial Intelligence', 'Max Tegmark', 2017, '9781101970317')
(5, 'Artificial Intelligence: A Modern Approach', 'Stuart Russell and Peter Norvig', 2021, '9780134610993'
)
(6, 'Deep Learning', 'Ian Goodfellow Yoshua Bengio and Aaron Courville', 2016, '9780262035613')
(7, 'The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World', 'Pedro
Domingos', 2015, '9780465065707')
(8, 'Prediction Machines: The Simple Economics of Artificial Intelligence', 'Ajay Agrawal Joshua Gans and
Avi Goldfarb', 2018, '9781633695672')
(9, 'Rebooting AI: Building Artificial Intelligence We Can Trust', 'Gary Marcus and Ernest Davis', 2019, '
9781524748258')
(10, 'Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy', "Cathy O'Ne
il", 2016, '9780553418835')
(11, 'The Social Construction of Reality A Treatise in the Sociology of Knowledge', 'Peter L. Berger and T
homas Luckmann', 1966, '9780385058988')
(12, 'Thinking Fast and Slow', 'Daniel Kahneman', 2011, '9780374533557')
(13, 'The Tipping Point How Little Things Can Make a Big Difference', 'Malcolm Gladwell', 2000, '978031634
6627')
(14, 'Outliers The Story of Success', 'Malcolm Gladwell', 2008, '9780316017930')
(15, 'Guns Germs and Steel The Fates of Human Societies', 'Jared Diamond', 1997, '9780393317558')
(16, 'Sapiens A Brief History of Humankind', 'Yuval Noah Harari', 2014, '9780062316110')
(17, 'Homo Deus A Brief History of Tomorrow', 'Yuval Noah Harari', 2016, '9780062464316')
(18, 'The Structure of Scientific Revolutions', 'Thomas S. Kuhn', 1962, '9780226458120')
(19, 'Bowling Alone The Collapse and Revival of American Community', 'Robert D. Putnam', 2000, '9780743203
043')
(20, 'The Spirit Level Why Equality is Better for Everyone', 'Richard Wilkinson and Kate Pickett', 2009, '
9781608193417')
```

3. Next I check the authentication of the users using the authentication.py file, to make sure that the registered users can login. Below is the screenshot of running authentication.py file. This is for admin user, it's the same for regular user as well.

```
(base) pushwitha@Pushwithas-MacBook-Pro BMS % python3 authentication.py

Options:
1. Login
2. Logout
3. Exit
Enter your choice: 1
Enter username: pushwitha
Enter password: test123
Login successful! Welcome, pushwitha (admin).

Options:
1. Login
2. Logout
3. Exit
Enter your choice: 2
Goodbye, pushwitha!

Options:
1. Login
2. Logout
3. Exit
Enter your choice: 3
Exiting program. Goodbye!
```

4. First I want to show what happens when I run main.py – for the admin user and the regular user, and then I'll show the book_management.py file.

```
(base) pushwitha@Pushwithas-MacBook-Pro BMS % python3 main.py

Main Menu:
1. Login
2. Exit
Enter your choice: 1
Enter username: pushwitha
Enter password: test123
Login successful! Welcome, pushwitha (admin).

Admin Menu:
1. Add Book
2. Delete Book
3. Search Books
4. Logout
Enter your choice: 4
Goodbye, pushwitha!

Main Menu:
1. Login
2. Exit
Enter your choice: 1
Enter username: jnanashree
Enter password: test345
Login successful! Welcome, jnanashree (user).

User Menu:
1. Search Books
2. Borrow Book
3. Logout
Enter your choice:
```

When I run the main.py file, the user will will login using the credentials. The system will identify if the admin logged in or user logged in. Based on the admin / user it will give the user menu. So, when a user logs in and wants to see all the books available, this is an example of how it is displayed, they can check out whatever book they want from the list. When an admin logsin, they will be able to add books to the database.

```
(base) pushwitha@Pushwithas-MacBook-Pro BMS % python3 main.py

Main Menu:
1. Login
2. Exit
Enter your choice: 1
Enter username: jnanashree
Enter password: test345
Login successful! Welcome, jnanashree (user).

User Menu:
1. Search Books
2. Borrow Book
3. Logout
Enter your choice: 1

Enter search criteria (leave blank if not applicable):
Title:
Author:
Year:
ISBN:

Books Found:
Title: Artificial Intelligence: A Guide to Intelligent Systems, Author: Michael Negnevitsky, Year: 2005, I
SBN: 9780321204660

Title: Superintelligence: Paths Dangers Strategies, Author: Nick Bostrom, Year: 2014, ISBN: 9780198739838

Title: Human Compatible: Artificial Intelligence and the Problem of Control, Author: Stuart Russell, Year:
 2019, ISBN: 9780525557999

Title: Life 3.0: Being Human in the Age of Artificial Intelligence, Author: Max Tegmark, Year: 2017, ISBN:
 9781101970317

Title: Artificial Intelligence: A Modern Approach, Author: Stuart Russell and Peter Norvig, Year: 2021, IS
BN: 9780134610993

Title: Deep Learning, Author: Ian Goodfellow Yoshua Bengio and Aaron Courville, Year: 2016, ISBN: 97802620
35613
```

5. When I run book_management.py this is what we get, the users can perform whatever operations they want to. Whenever the user borrows a book, its updated in a new table in the database BorrowedBooks.

```
(base) pushwitha@Pushwithas-MacBook-Pro BMS % python3 book_management.py

Book Management Options:
1. Add Book
2. Delete Book
3. Search Books
4. Borrow Book
5. List Borrowed Books
6. Exit
Enter your choice:
```