By: Nicoline Nymand-Andersen
Friday, 08 December, 2023

1. Setup information

For this project I decided to implement Redis using Python. I have submitted two files: *main.py* and *library.py*. The former provides the execution of the main functions (provided by the latter, library.py file) and interaction with the user.

1.1. **Necessary steps to run Redis Client.** In order to run the file, a linux (subsystem) is required. In my case, I set up a wsl (Windows Subsystem for Linux), from which I was able to run redis. I ran my code using Linux distribution Ubuntu. The files *main.py* and *library.py* must be mounted on this system.

Requirements for running the code:

Mounting the folder:

```
sudo mount -t drvfs 'Path/to/files/on/windows' /mnt/architectures
```

Pip must be installed:

```
sudo apt install python3-pip
```

Redis must be installed:

```
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share
/keyrings/redis-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https:/
/packages.redis.io/deb $(lsb_release -cs) main" | sudo tee /etc/apt/sources.
list.d/redis.list

sudo apt-get update
sudo apt-get install redis
```

redis must be installed for python:

```
pip3 install redis
```

The server must be started:

```
sudo service redis-server start
```

You can check if the server works, if Pong is returned:

```
redis-cli
> ping
```

1.2. **Run the code.** Please navigate to where the files are located (done during mounting). From here, you can initiate the program by running:

```
python3 main.py
```

This should initiate the toy application of a database of books that can be borrowed in a library. The library is accessible from two seperate interfaces, each of which limit the interaction with the library:

(1) Librarian Interface, which allows for the end-user (i.e. the librarian) to
   (a) retrieve some book information given an isbn,
   (b) add a book to the library, which publishes the news to different channels. If the book already exists, more copies can be added to or removed from the database.
   (c) remove a book from the library given an isbn.

(2) Customer Interface, which allows for the end-user (i.e. the customer) to
   (a) retrieve book information given an isbn,
   (b) borrow a book given an isbn,
   (c) return a book given an isbn,
   (d) subscribe to a channel given a channel name (restricted to author name, language, genre or year of publication)
   (e) unsubscribe to a channel given a channel name (restricted to author name, language, genre or year of publication)
   (f) retrieve news from subscribed channels, which listens to whether any news will be posted.

At each step, it is possible to return to the previous page. Only the first presented menu allows a total exit from the program.

2. Supported scenarios and examples

2.1. **Scenario 1: Publish a book.** In order to publish a book, you must be a librarian. For this reason, the user must insert the option 2. Considering the book only is published when it is added to the library, the librarian must enter 2. Once it has successfully been added to the library, it also has been published. The details of adding a book to the library can be seen below, figure: 1. For proof, see next scenario and figure: 2.

```
nicilini@LAPTOP-GQIPNUAH:/mnt/architectures$ python3 main.py
Welcome to the National Library of Narnia

Choose an option:
1. Customer
2. Librarian
3. Exit
Enter your choice: 2

You are in librarian mode. What would you like to do ?
1. Get book information
2. Add a book
3. Remove a book
4. Return to the main menu
Enter your choice: 2
Please provide the isbn: 978-0141182636
Please provide the title: The Great Gatsby
Please provide the author: F Scott Fitzgerald
Please provide the number of copies: 10
Please provide the language of the book: English
Please provide the genre of the book: Realism
Please provide the year of publication of the book: 1925
The book 978-0141182636 has successfully been added to the library.
```

Figure 1. The scenario of publishing a book

2.2. **Scenario 2: Subscribe to a channel.** In order to subscribe to a channel, you must be a customer. First the customer must choose the option of subscribing to a channel (option 4) and then the customer must retrieve the information posted to that channel. The left side of figure: 2 displays the publishing of The Great Gatsby to the channel, while the right side displays the successful subscription to that channel. As the listening function is activated by the user specifying he would like to retrieve news from subscribed channels (option 6), the librarian publishes the book. The news is sent to the user in real time, announcing a newly published book and also naming the ISBN of the newly published book.

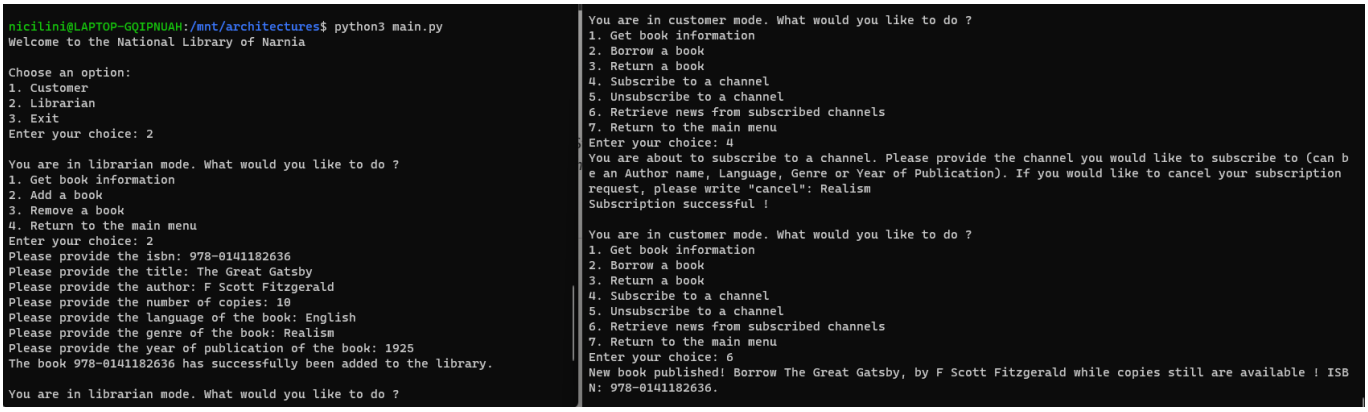

FIGURE 2. The scenario of subscribing to a channel

2.3. **Scenario 3: Borrow and Return books.** In order to borrow or return books, you must be a customer. Therefore, the user must insert 1, as seen in figure: 3. Then the customer is confronted with a menu, displaying all the options a customer has. As the customer would like to borrow a book, the number 2 and the isbn are inserted. If the borrowing procedure is successful, the customer is returned a prompt, congratulating him on successfully borrowing the book. Finally, figure: 3 also displays that the book, in fact, was borrowed, as now the number of copies has been reduced by one (9 copies left).

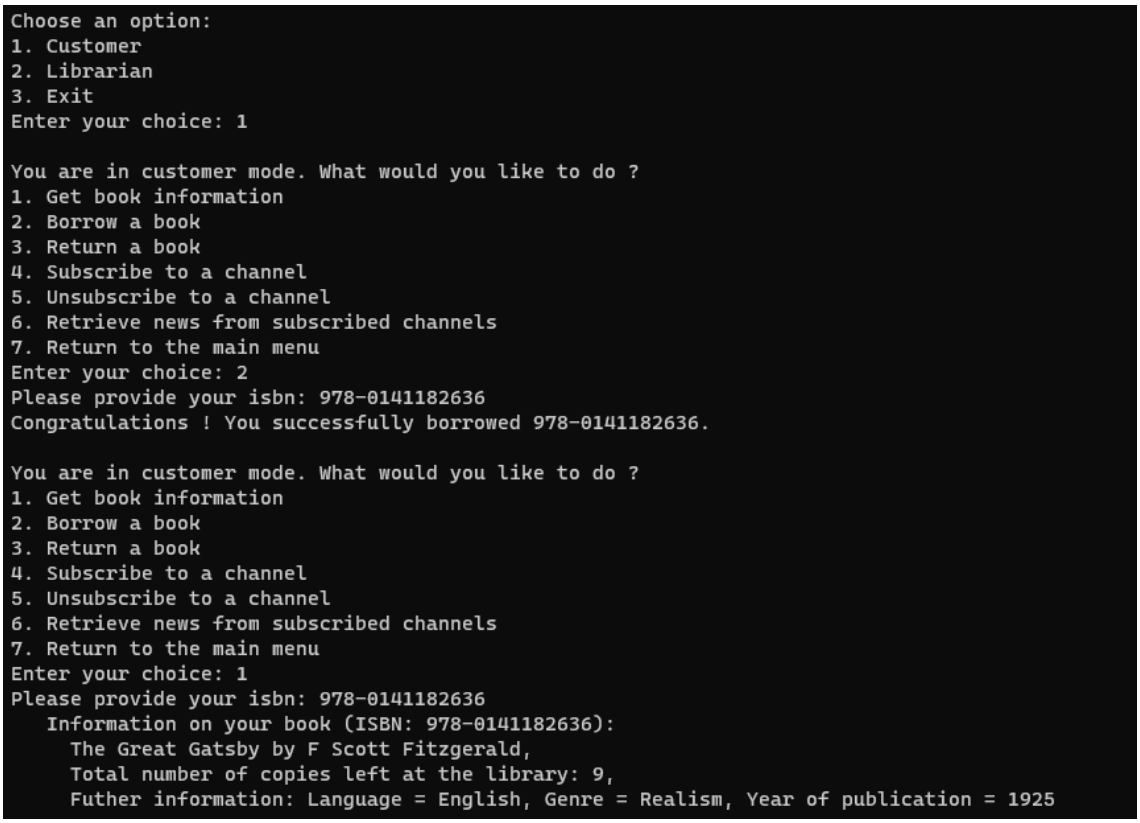

FIGURE 3. The scenario of borrowing a book

In order for the customer to return this book, he must insert the option 3 and fill in the isbn. As displayed in figure: 4 the customer inputs this information and is confronted with a prompt telling him that the transaction was successful. Checking again the number of books in the library, we can see it has been increased by one (thus returning to a total of 10 copies).

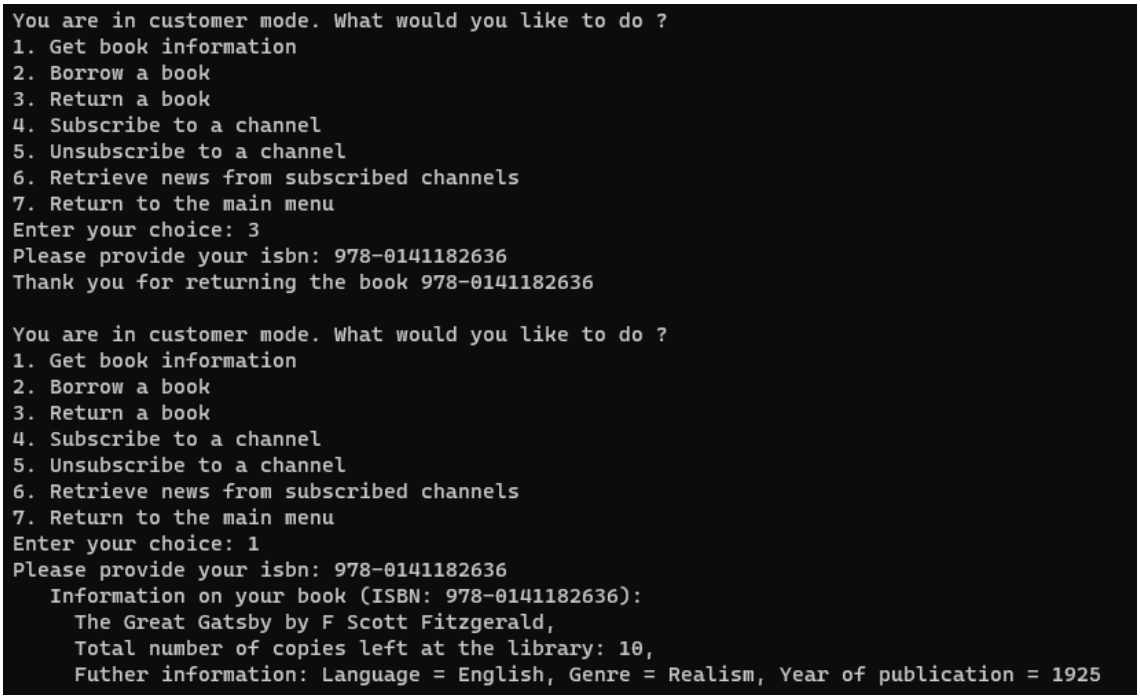

FIGURE 4. The scenario of returning a book

2.4. **Scenario 4: Detecting expired books.** Once books expire, these do not exist in the system anymore. For this reason, we expect the system to let us know that the book unfortunately does not exist in the system. For this, we can enter customer or librarian mode, and get information on the book (option 1). For this experiment, I waited to for the book to expire (set to 5 minutes), to perform these steps. The results can be seen in the screenshot provided below, figure: 5.

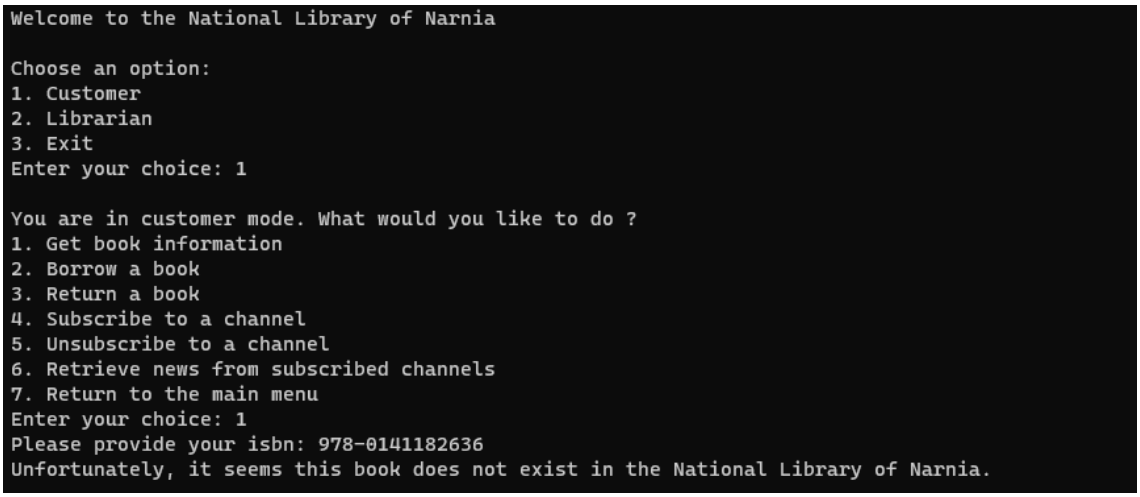

FIGURE 5. The scenario of retrieving an expired book

2.5. **Further Scenarios.** The code supports a variaty of further scenarios, but providing all of these would be rather extensive. For this reason, I will list a couple of further functionalities provided by the code. The librarian is furthermore able to:

- remove books from the library. When this is done, the book is deleted and news is published to each of its channels, informing on its removal.

- add or remove copies. If the librarian tries to publish a new book which already exists in the database, the librarian gets the option to add or remove copies of the book. If the librarian wishes to do neither, the opportunity to return to the librarian menu is given.

The customer is furthermore able to:

- unsubscribe from a channel. If the customer no longer is interested in following a channel, the customer has the possibility to unsubscribe.

Finally, the system also handles a couple of exceptions. For example, if the librarian wants to remove more copies than are in stock, the librarian is confronted with a transaction failure, stating how many copies maximally can be removed. Another example is that if a user is trying to retrieve information on a book that does not exist, the user is told that there are no copies of that book. A final example is in case the user inputs an option, which is not recognised as an option. In this case, the user is prompted with the information, that he must select one of the valid options.