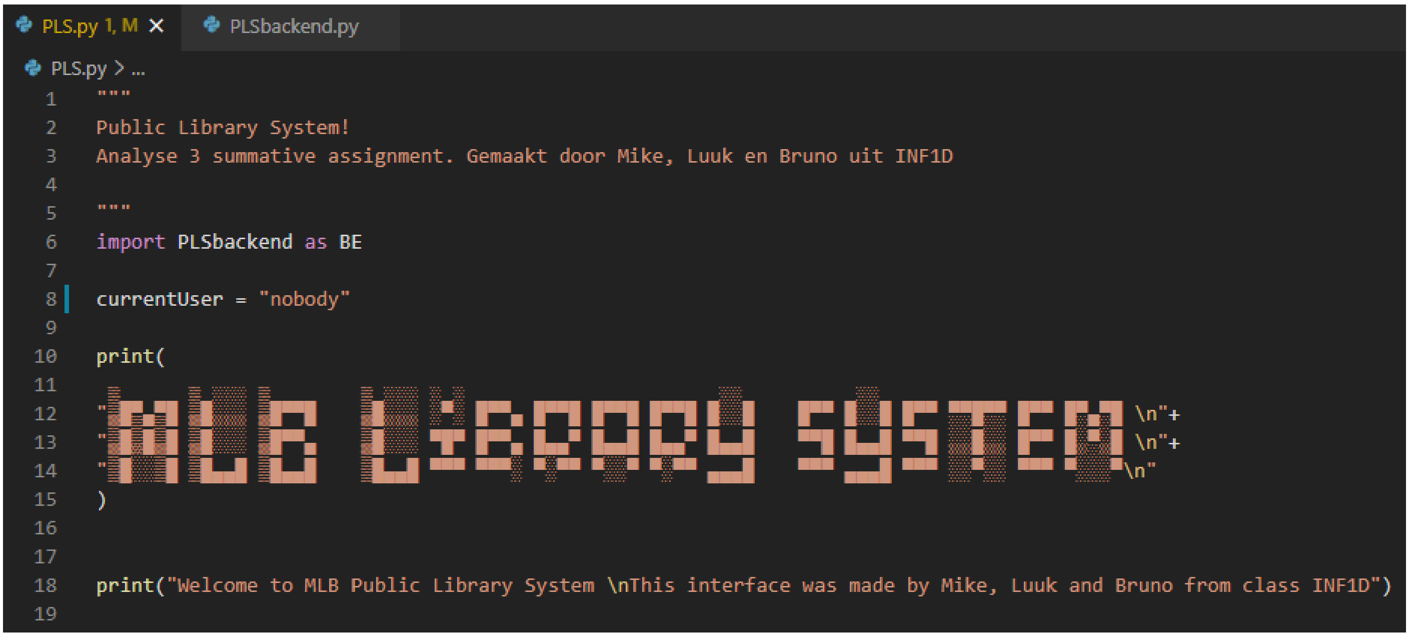
MLB Public Library System User manual

Summative assignment Analysis 3 (2021)



```python
"""
Public Library System!
Analyse 3 summative assignment. Gemaakt door Mike, Luuk en Bruno uit INF1D

"""
import PLSbackend as BE

currentUser = "nobody"

print(

"[illegible] \n"+
"[illegible] \n"+
"[illegible]\n"
)


print("Welcome to MLB Public Library System \nThis interface was made by Mike, Luuk and Bruno from class INF1D")

```

Made by:
Mike Jansen (1003769)
Luuk Selen (1009198)
Bruno van der Stel (1005978)

From class:
INF1D

Teachers:
Aleksandra Wypych
Johan Bosman

[Patch notes 2.0 (Retake)]

- Added functional Subscriber Company class

- Expanded loan book functionality to customers

- Several quality of life improvements when traversing menu’s

- Tweaked JSON integration

- Updated manual accordingly

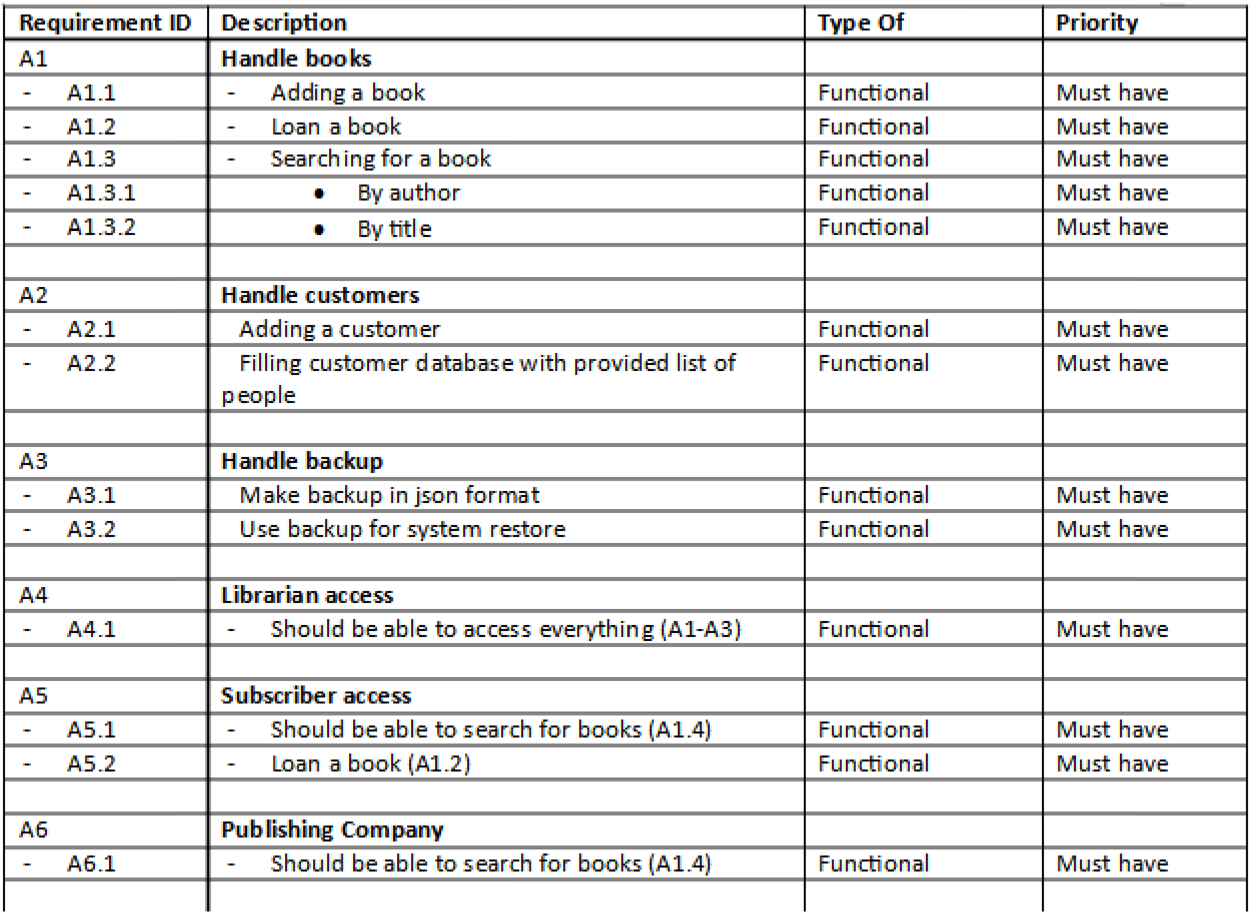
- Added a guide that clearly explains how to loan a book as a customer

Index

1. The System Requirements Specification (SRS)

As specified in the assignment, the system must meet several requirements.

Our SRS:



Requirement ID	Description	Type Of	Priority
A1	**Handle books**		
- A1.1	- Adding a book	Functional	Must have
- A1.2	- Loan a book	Functional	Must have
- A1.3	- Searching for a book	Functional	Must have
- A1.3.1	• By author	Functional	Must have
- A1.3.2	• By title	Functional	Must have
A2	**Handle customers**		
- A2.1	Adding a customer	Functional	Must have
- A2.2	Filling customer database with provided list of people	Functional	Must have
A3	**Handle backup**		
- A3.1	Make backup in json format	Functional	Must have
- A3.2	Use backup for system restore	Functional	Must have
A4	**Librarian access**		
- A4.1	- Should be able to access everything (A1-A3)	Functional	Must have
A5	**Subscriber access**		
- A5.1	- Should be able to search for books (A1.4)	Functional	Must have
- A5.2	- Loan a book (A1.2)	Functional	Must have
A6	**Publishing Company**		
- A6.1	- Should be able to search for books (A1.4)	Functional	Must have

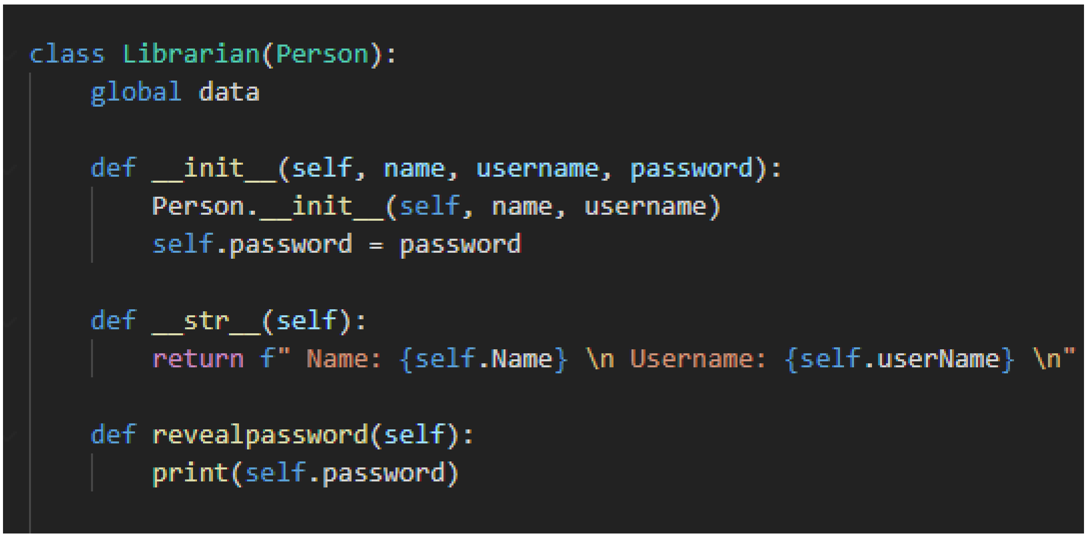
2. The system design.

This system is designed from an Object Oriented Programming (OOP) perspective, meaning that the classes and their objects are the focal point of the design. The system consists of several classes, with some of them creating objects, and some of them remaining abstract classes. This implies that there will be classes that inherit from superclasses.

For example, the Librarian, Subscriber and PublishingCompanyUser class share the same superclass (Person). These subclasses will have objects defining individual users. But other classes do not have subclasses, nor inherit from other classes. An example being the important PublicLibrary class. For this reason, the PublicLibrary class features static methods, as it does not instantiate objects.

This design paradigm is seen throughout the program as a whole.

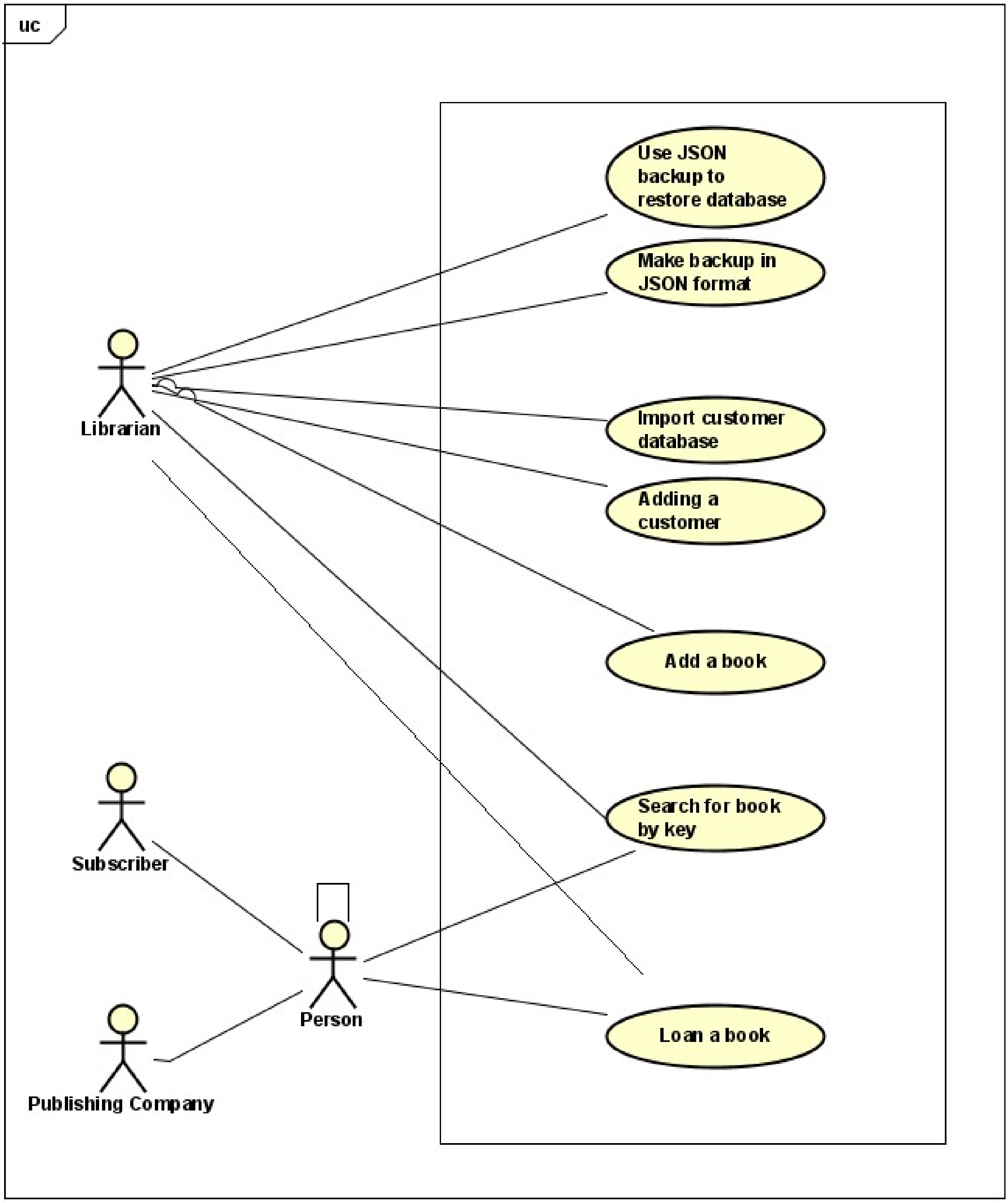
From a technical perspective, this system features a front-end and a back-end portion. These are seperated by different files (PLS.py and PLSbackend.py, respectively). The backend contains all the classes and several helper functions, whereas the front-end features the user interface and the menu’s through which the user will navigate and interact with the system’s functionality.



```
class Librarian(Person):
    global data

    def __init__(self, name, username, password):
        Person.__init__(self, name, username)
        self.password = password

    def __str__(self):
        return f" Name: {self.Name} \n Username: {self.userName} \n"

    def revealpassword(self):
        print(self.password)
```

Below you will see the Use Case Diagram for this system. Note how the librarian serves as not only the employee that adds and edits books, this user also serves as an administrator that can create backups, load backups and make new customer accounts.

The subscriber and publishing company users do not have this functionality.



3. How to use the MLB Public Library SystemUser

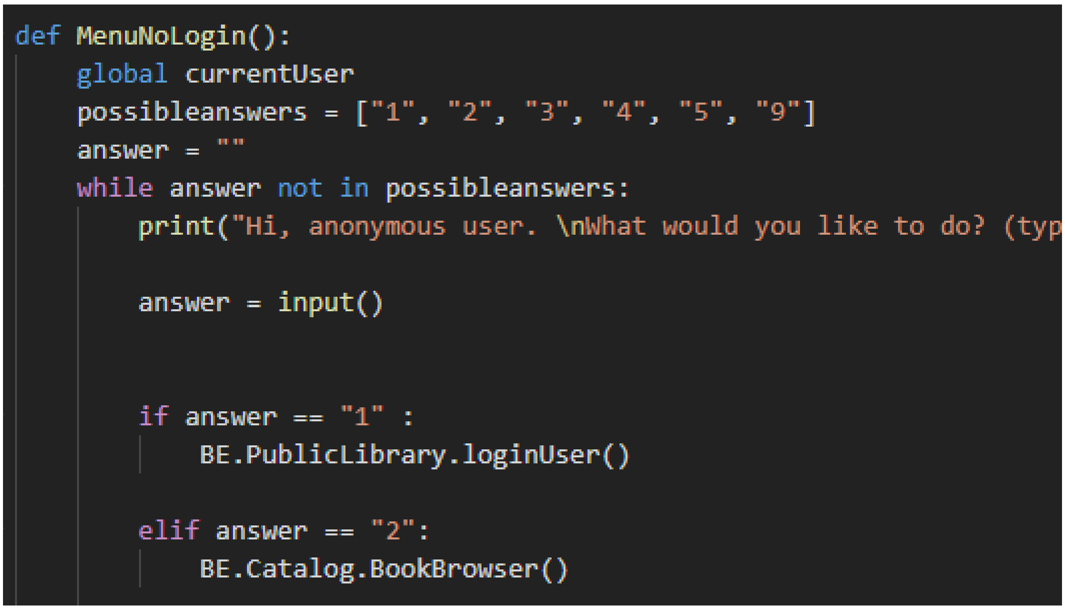
The administrator / librarian login credentials are:
*Username: **admin***
*Password: **admin***

The administrator / librarian is the only user that can access all of the functions. These login credentials are imperative for that purpose.

Upon running the file, the user will be greeted in the main menu. Navigating through the menu’s of this system is done by typing the numbers corresponding to the choice you want to make.

For example, if you want to login and the program tells you,
“1. Login” ,
the user will only need to type “1” as their choice. Anything other than just the number will not be recognised as a valid input. If the input is invalid, the program will tell you that the answer was not recognised.

The admin can login using this menu.



```
def MenuNoLogin():
    global currentUser
    possibleanswers = ["1", "2", "3", "4", "5", "9"]
    answer = ""
    while answer not in possibleanswers:
        print("Hi, anonymous user. \nWhat would you like to do? (typ

        answer = input()


        if answer == "1" :
            BE.PublicLibrary.loginUser()

        elif answer == "2":
            BE.Catalog.BookBrowser()
```

Some menu’s will require the user to enter different bits of information. For example, when the librarian is making a new customer account, the program will ask for several inputs, which correspond with the data fields for the new user.

The main functionality of the admin / librarian are :

1. Add single book to database.
2. Add multiple books to database using JSON
3. Create new subscriber.
4. Create new Librarian
5. Loan books + loan administration
6. Load / make system backup.
7. Browse Books
8. Add customers via CSV file
9. Logout.

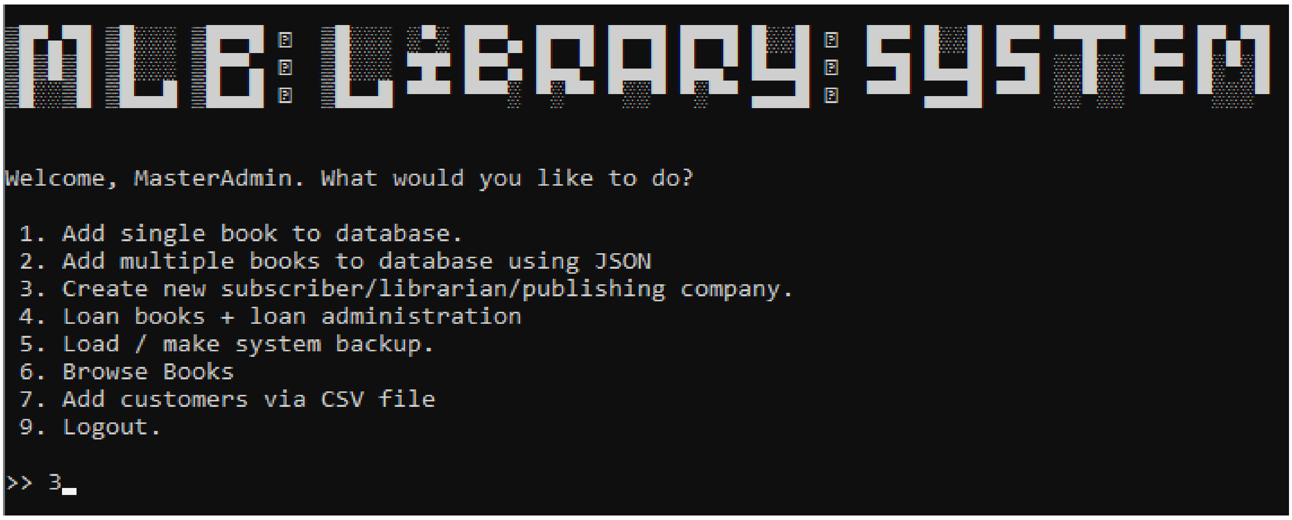
The titles of these functions should be self-explanatory. Remember that the user of this interface interacts by pressing **only the number that corresponds with the function.**

The functions will guide you through what is expected from the user via print statements.

4. (Guide) Interacting with the system as a customer

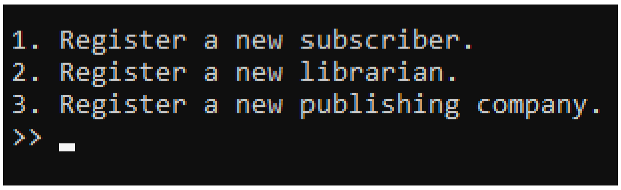
1. To interact with the system as a customer, you first need a customer account.
Only the admin can create customer accounts. To do so, you first need to log in as an admin. Remember that the credentials for logging in are
Username: admin
Password: admin

2. after logging in as an admin, you can create a customer account by going to corresponding menu (option **3**)



```
Welcome, MasterAdmin. What would you like to do?

 1. Add single book to database.
 2. Add multiple books to database using JSON
 3. Create new subscriber/librarian/publishing company.
 4. Loan books + loan administration
 5. Load / make system backup.
 6. Browse Books
 7. Add customers via CSV file
 9. Logout.

>> 3
```

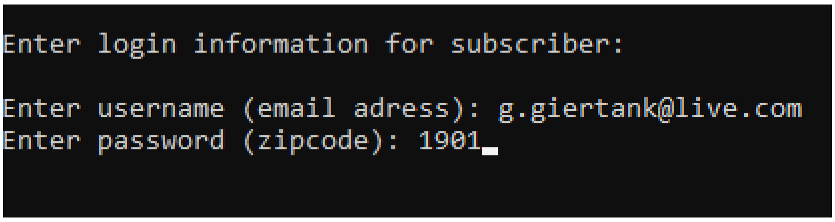
3. After selecting this menu, you will be prompted to create either a subscriber, librarian or publishing company account



```
1. Register a new subscriber.
2. Register a new librarian.
3. Register a new publishing company.
>>
```

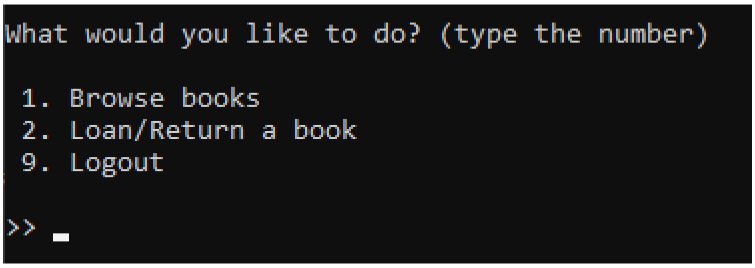
4. Select option 1 to enter the subscriber creation screen. You will be asked to enter the user's language, first name, last name, e-mail address, and so on.

5. After doing so, the program will save this account and you are now able to log in with this new account.
Your username will be the e-mailaddress for this new account.
The password will be the zipcode for this account.



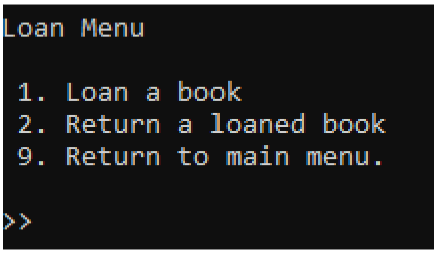
```
Enter login information for subscriber:

Enter username (email adress): g.giertank@live.com
Enter password (zipcode): 1901
```

6. When logging in, you can do several things.



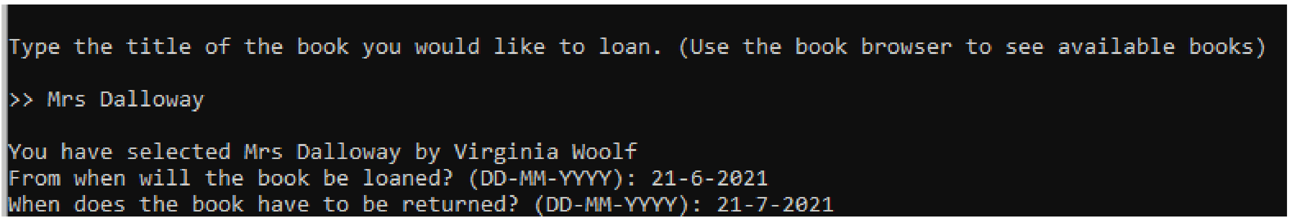
```
What would you like to do? (type the number)

 1. Browse books
 2. Loan/Return a book
 9. Logout

>> _
```

You can browse books, loan books and return books as a subscriber. Pressing the number that corresponds with the function brings you there.



```
Loan Menu

 1. Loan a book
 2. Return a loaned book
 9. Return to main menu.

>>
```

7. When pressing 1 to loan a book, you can type in the name of the book you want to loan, and enter the dates on which you will loan the book and when you will return the book



```
Type the title of the book you would like to loan. (Use the book browser to see available books)

>> Mrs Dalloway

You have selected Mrs Dalloway by Virginia Woolf
From when will the book be loaned? (DD-MM-YYYY): 21-6-2021
When does the book have to be returned? (DD-MM-YYYY): 21-7-2021
```

8. Returning the book can be done by entering the loan menu. Returning a loan book can be done by the loaner OR by a librarian.

9. That’s it! You are done. This is how you loan books and return books in our Public Library System.

General notes:

A **publishing company account** has the same functionality as a subscriber. However, the publishing company account does not require an address, e-mail, or other peripherals that a normal subscriber would. Instead, this account needs a name, a username and a password.

When loading major JSON files such as the backup and customer lists, you may need to restart the program to save the changes.