PLS Document

Hogeschool Rotterdam



Table of Contents

[Document information 1](#_Toc131964498)

[Document version 1](#_Toc131964499)

[Authors 1](#_Toc131964500)

[Software requirements specifications 2](#_Toc131964501)

[Introduction 2](#_Toc131964502)

[Purpose 2](#_Toc131964503)

[Intended use 2](#_Toc131964504)

[Scope 2](#_Toc131964505)

[Functional requirements 3](#_Toc131964506)

[Class diagram 7](#_Toc131964507)

[Class diagram 7](#_Toc131964508)

[Catalog 7](#_Toc131964509)

[Description 7](#_Toc131964510)

[Book 7](#_Toc131964511)

[Description 7](#_Toc131964512)

[Book item 7](#_Toc131964513)

[Description 7](#_Toc131964514)

[Loan item 7](#_Toc131964515)

[Description 7](#_Toc131964516)

[Person 7](#_Toc131964517)

[Description 7](#_Toc131964518)

[Library admin 8](#_Toc131964519)

[Description 8](#_Toc131964520)

[Member 8](#_Toc131964521)

[Description 8](#_Toc131964522)

[Use Case diagrams 9](#_Toc131964523)

[Member use case 9](#_Toc131964524)

[Admin use case 10](#_Toc131964525)

[User manual 11](#_Toc131964526)

[Login 11](#_Toc131964527)

[Managing members 12](#_Toc131964528)

[Get all members 12](#_Toc131964529)

[Add new member 12](#_Toc131964530)

[Edit member 13](#_Toc131964531)

[Delete member 14](#_Toc131964532)

[Member import file requirements 15](#_Toc131964533)

[List member imports (CSV) 15](#_Toc131964534)

[Import members from CSV 15](#_Toc131964535)

[Managing catalog 16](#_Toc131964536)

[List all books in catalog 16](#_Toc131964537)

[Add a new book 16](#_Toc131964538)

[Edit an existing book 17](#_Toc131964539)

[Delete an existing book 18](#_Toc131964540)

[List available import files 18](#_Toc131964541)

[Books import file requirements 18](#_Toc131964542)

[Import books from import file 19](#_Toc131964543)

[Search specific book 20](#_Toc131964544)

[Manage library 21](#_Toc131964545)

[List all book items 21](#_Toc131964546)

[Search book item 22](#_Toc131964547)

[Add new book item 23](#_Toc131964548)

[Edit book item 24](#_Toc131964549)

[Delete book item 25](#_Toc131964550)

[Loan a book item 26](#_Toc131964551)

[List loaned items 27](#_Toc131964552)

[Manage back-ups 28](#_Toc131964553)

[List all back-ups 28](#_Toc131964554)

[Create back-up 28](#_Toc131964555)

[Load back-up 29](#_Toc131964556)

# Document information

## Document version

|  |  |  |
| --- | --- | --- |
| Version | Date | Notes |
| 1.0 | 10-03-2022 | Document creation |
| 1.1 | 11-03-2022 | Add diagrams |
| 1.2 | 25-03-2022 | Modified based on feedback |
| 1.3 | 28-03-2022 | Updated diagrams |
| 1.4 | 04-04-2022 | Updated diagrams to better reflect requirements |
| 1.5 | 07-04-2022 | Added requirements |
| 1.6 | 10-04-2022 | Updated requirements based on new insights |
| 1.7 | 14-04-2022 | Added User manual chapter |
| 1.8 | 08-04-2023 | Updated and improved for retake |

## Authors

|  |  |
| --- | --- |
| Name | Student number |
| Jeremey Sas | 1028294 |
| Michael Heerkens | 1028285 |
| Tobias Roessingh | 1042623 |

# Software requirements specifications

## Introduction

### Purpose

The purpose of the Public Library System (PLS) is to manage the inventory of a library through a console application.

### Intended use

The system is supposed to be used by either members or employees of the library. It is intended to be used as an inventory management system.

### Scope

Members of the system (Customers of the library) are able to search for books and check if they are available. When a book is available the member is able to loan the book and can loan the book for a maximum of 30 days and can’t loan more than three books at once.

The library Admin can manage the library inventory, manage its members and loan books to specific members. This user is also privileged to create backups of the system and restore them from a valid backup file.

## Functional requirements

The functional requirements are divided in four categories based on the priority, these priorities are set following the MoSCoW method.

Must have requirements are required for base functionality of the system and must be in the end product.

Should have requirements are required for the system functionalities but are dependent on the requirements of the Must have requirements.

Could have requirements are not mandatory for the system functionalities. These requirements will be added to the system if the time allows for these requirements

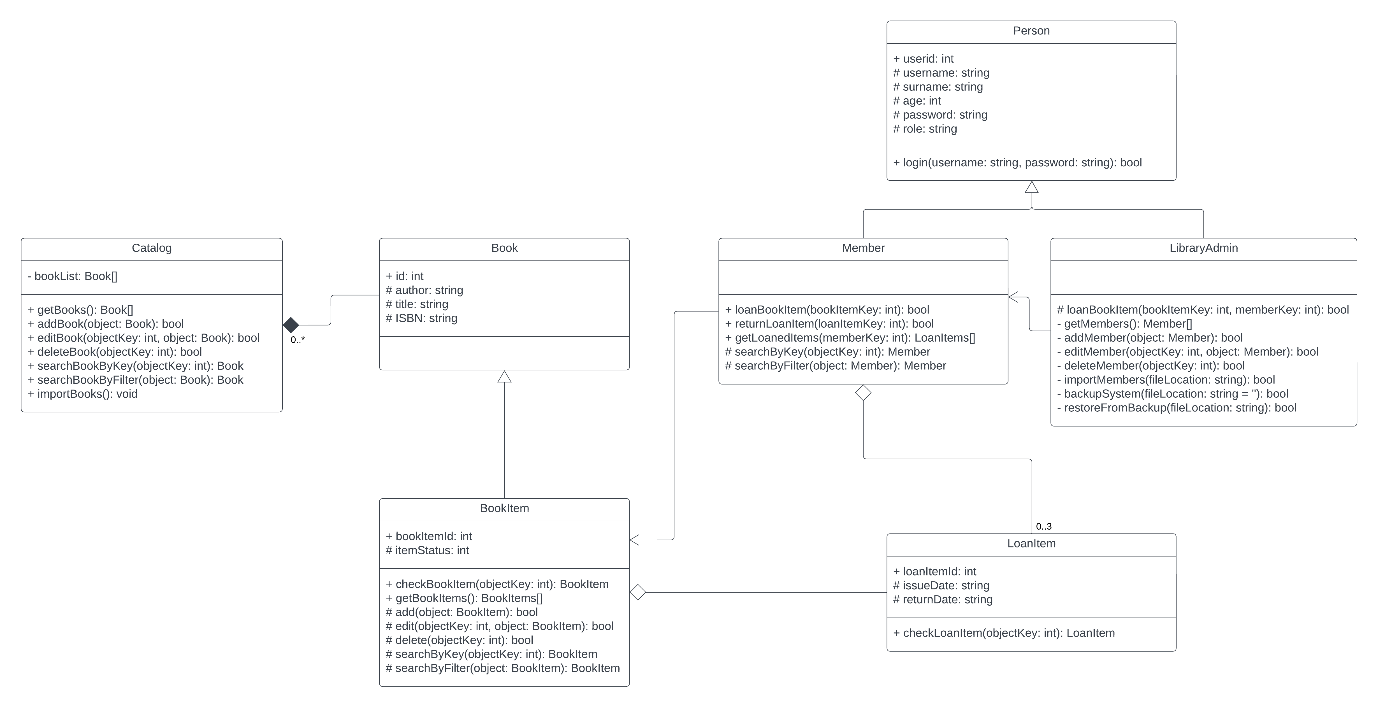
Won’t have requirements are not mandatory for the system and will not be in this implementation of the system. These requirements can be added in a later project to improve ease of use, user interface and speed improvements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Requirement | Priority | Relations | Type | |
| 1 | View members list | Must | 2, 39 | | F, A |
| Description | An administrator needs to be able to view the list of members | | | | |
| 2 | Member management | Must | 1, 42 | | F, A |
| Description | An administrator needs to be able to add, edit and delete members from and to the system | | | | |
| 3 | Check loaned items | Must |  | | F, A |
| Description | An administrator needs to be able to check the currently loaned items per member | | | | |
| 4 | View catalog | Must | 5, 22, 28, 40 | | F, A, M |
| Description | A user (Admin and Member) needs to be able to view the catalog (list of books) | | | | |
| 5 | Catalog management | Must | 4, 43 | | F, A |
| Description | An administrator needs to be able to add, edit and delete books from and to the catalog | | | | |
| 6 | View book items | Must | 7, 9, 24, 28, 41 | | F, A, M |
| Description | A user (Admin and Member) needs to be able to view the list of book items in the system | | | | |
| 7 | Book item management | Must | 6, 25 | | F, A |
| Description | An administrator needs to be able to add new book items to the system | | | | |
| 8 | System back-up | Must |  | | F, A, S |
| Description | An administrator needs to be able to make a back-up of the system in JSON format | | | | |
| 9 | Loan book item | Must | 6, 26 | | F, M, L |
| Description | A member needs to be able to loan an available book item | | | | |
| 10 | User logon | Must |  | | F, A, M, S |
| Description | All users need to be able to logon to the system | | | | |
| 11 | Check mandatory files | Must | 34 | | F, S |
| Description | On startup the system will check all required data files | | | | |
| 12 | Functionality before logon | Must |  | | NF, A, M |
| Description | Before the logon the only two options for a user to choose from is ‘Login’ or ‘Exit’ | | | | |
| 13 | Book item copies | Must |  | | NF, S |
| Description | Each book item has three copies | | | | |
| 14 | Book search | Must |  | | NF, S |
| Description | A title or author are required when searching for books in the system | | | | |
| 15 | Back-up formatting | Must |  | | NF, S |
| Description | A back-up file name contains the current date and serial number to distinguish multiple back-ups | | | | |
| 16 | Data saving | Must |  | | NF, S, FM |
| Description | All data is saved after each transaction in the system | | | | |
| 17 | Duplicates | Must |  | | NF, S, FM |
| Description | The system must prevent duplicate entities that are being created | | | | |
| 18 | Username formatting | Must |  | | NF, S, M |
| Description | A member username must contain lowercase formatting | | | | |
| 19 | Defining members | Must |  | | NF, S, M, A |
| Description | The username and password of a member are defined by the library administrator | | | | |
| 20 | Navigation | Must |  | | NF, S |
| Description | The system navigation will be chosen by digits | | | | |
| 21 | Import members | Should |  | | F, A, M, FM |
| Description | An administrator needs to be able to import members via CSV file | | | | |
| 22 | Search books | Should | 4, 33 | | F, A |
| Description | An administrator needs to be able to search books in the catalog | | | | |
| 23 | Import books | Should |  | | F, A, FM |
| Description | An administrator needs to be able to import books from a JSON file | | | | |
| 24 | Search book items | Should | 6, 33 | | F, A |
| Description | An administrator needs to be able to search a book item in the system | | | | |
| 25 | Check availability | Should | 7 | | F, A |
| Description | An administrator needs to be able to check availability of a book item in the system | | | | |
| 26 | Lend book item | Should | 9 | | F, A, L |
| Description | An administrator needs to be able to lend a book item to a member | | | | |
| 27 | Restore system | Should |  | | F, A, S, FM |
| Description | An administrator needs to be able to restore the system via a back-up | | | | |
| 28 | Search book | Should | 4, 33 | | F, M |
| Description | A member needs to be able to search for a book in the catalog | | | | |
| 29 | Search book item | Should | 6, 33 | | F, M |
| Description | A member needs to be able to search book item in the system | | | | |
| 30 | Return loaned item | Should |  | | F, M, L |
| Description | A member needs to be able to return loaned item(s) | | | | |
| 31 | Application interface | Should |  | | F, S |
| Description | The main interface in console application | | | | |
| 32 | Admin credentials | Should |  | | F, A, S |
| Description | Admin credentials are hardcoded | | | | |
| 33 | Partial filters | Should | 22, 24, 28, 29 | | F, S |
| Description | Filters need to accept partial parameters | | | | |
| 34 | File availability | Should | 11, 35 | | F, S, FM |
| Description | Only when the required data file is available, it will be loaded into the system | | | | |
| 35 | File creation | Should | 34 | | F, S, FM |
| Description | The system creates new data files when the required data files are not available | | | | |
| 36 | Loan limit | Should | 37, 38 | | NF, S, M, L |
| Description | A member can only loan a book for a maximum of 30 days | | | | |
| 37 | Loan maximum | Should | 36, 38 | | NF, S, M, L |
| Description | A member can loan a maximum of three books simultaneously | | | | |
| 38 | Loan one book | Should | 36, 37 | | NF, S, M, L |
| Description | A member can only loan one book item of its multiple copies | | | | |
| 39 | Filter members list | Could | 1 | | F, S, A |
| Description | The system needs to be able to filter the list of members | | | | |
| 40 | Filter catalog | Could | 4 | | F, S, A, M |
| Description | The system needs to be able to filter the catalog | | | | |
| 41 | Filter book items | Could | 6 | | F, S, A, M |
| Description | The system needs to be able to filter the book items | | | | |
| 42 | Export members | Could | 2 | | F, S, A, FM |
| Description | The administrator can export the list of members to CSV file | | | | |
| 43 | Export books | Could | 5 | | F, S, A, FM |
| Description | The administrator can export the list of books to JSON file | | | | |
| 44 | Graphical interface | Won’t |  | | NF, S |
| Description | The system will not have any fancy interfaces and mouse pointer actions, for this version it will just be a command-line application | | | | |
| 45 | Location management | Won’t |  | | F, S |
| Description | The system cannot manage location of the library system, there will always be one location, the catalog of the system it is running on | | | | |

|  |  |
| --- | --- |
| Type Legend | |
| F | Functional |
| NF | Non-Functional |
| A | Administrative |
| M | Members |
| S | System |
| L | Loaning |
| FM | File management |

# Class diagram

## Class diagram



## Catalog

### Description

The catalog contains a list of all the books available at the library. This object contains all the information used to manage the inventory of books at the library.

## Book

### Description

The book contains all base information of a book item, like genre, title and author. This is also the object used to fill the catalog. This is a definition item and cannot be used as physical item.

## Book item

### Description

The book item is the physical object of a book. This can be used to loan to members of the library system.

## Loan item

### Description

The loan item will describe a connection between a member and the book item that has been loaned to the member.

## Person

### Description

This is a base for both the Library Admin and the Member. This will contain basic information used in the derived objects.

## Library admin

### Description

This is the administrator of the public library system. This user has a higher privilege level and can manage the system.

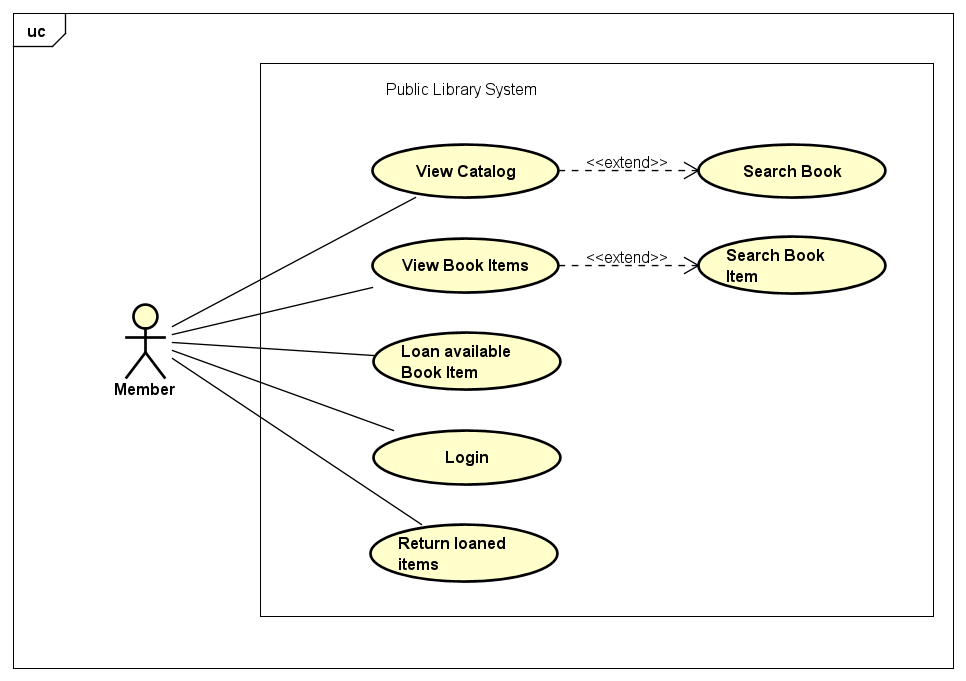
## Member

### Description

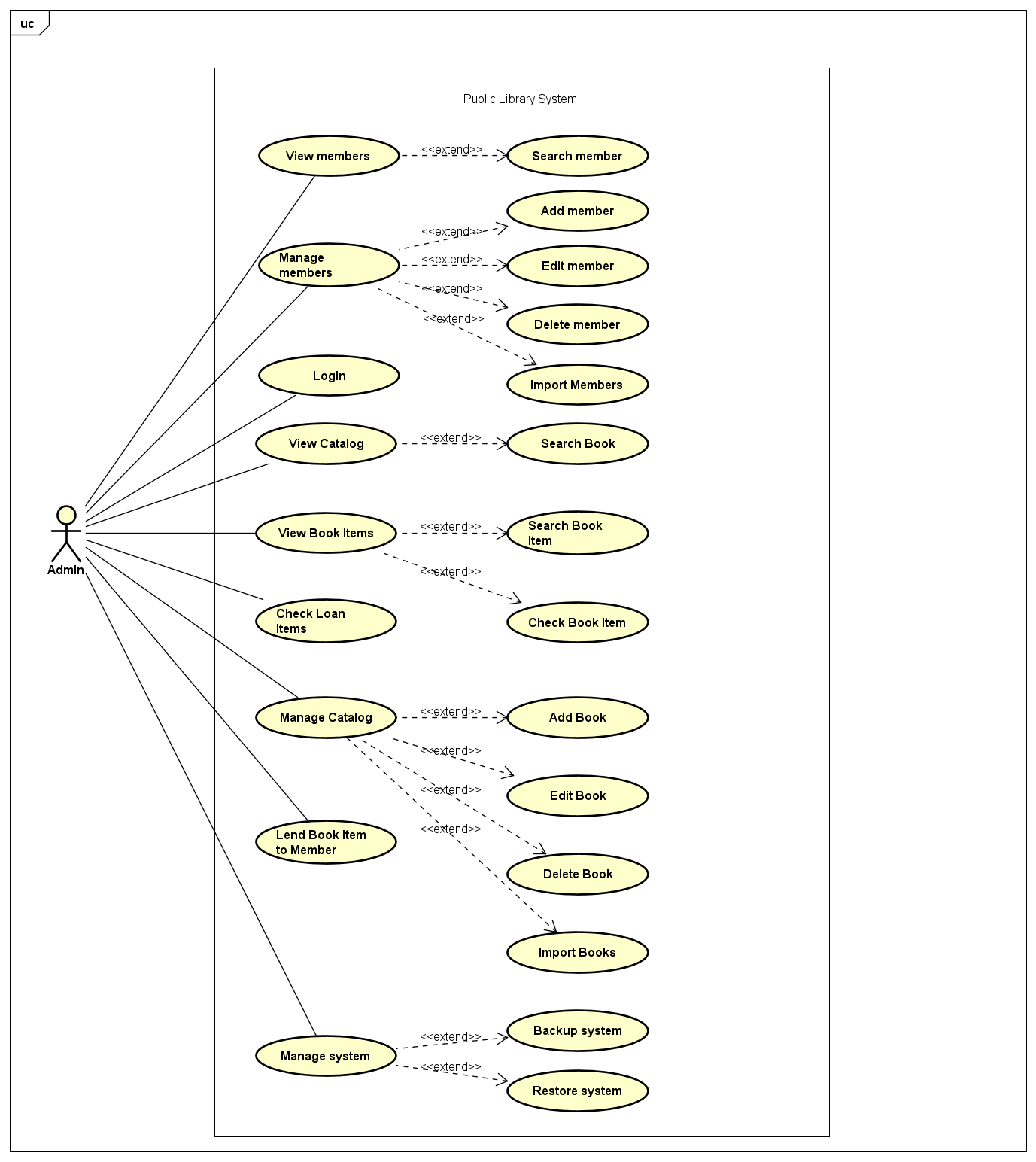
A member can loan book items, search for books and return loaned items. This user is created for customers of the library.

# Use Case diagrams

### Member use case



### Admin use case



# User manual

## Login

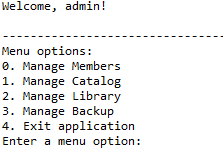
Enter username and press enter, see screenshot below:



Enter the password and press enter to enter the application, see screenshot below:

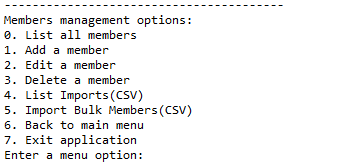


Main menu screenshot below:



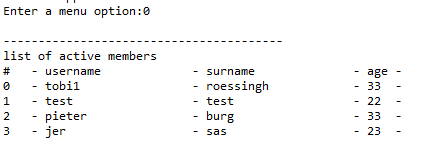
## Managing members

From the main menu select the option 0 and press enter to go to the member management menu, see screenshot below:



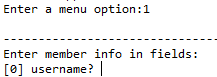
### Get all members

From the member management menu, select 0 and press enter to retrieve the list of all members, see screenshot below:

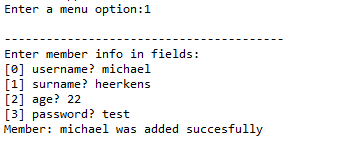


### Add new member

From the member management menu, press option 1 and press enter to begin adding a new member, see screenshot below:

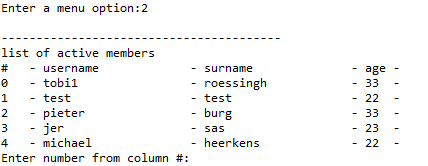


After this, fill in the data fields and press enter after filling the fields



### Edit member

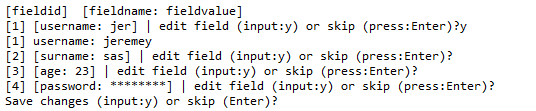
From the member management menu, press option 2 and press enter to begin editing an existing member, see screenshot below:



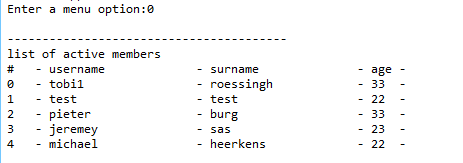
After this the user can select to either modify a field, by pressing y and enter, or to skip to the next field by pressing enter, see screenshot below:



At the end of all inputs the user is required to confirm the modification by pressing y and enter or only enter to cancel the modifications.

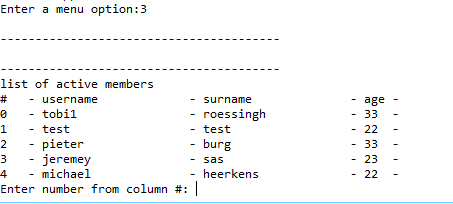


After this the member will be modified, see screenshot below:

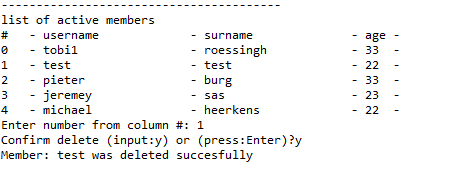


### Delete member

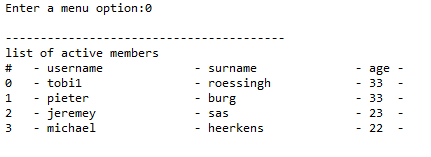
From the member management menu, press option 3 and press enter to begin deleting an existing member, see screenshot below:



After selecting the user to delete the user is required to confirm the action by pressing y and enter, or to cancel it with only pressing enter, see screenshot below:



After this the member is deleted, see screenshot below:



### Member import file requirements

File requirements:

* File extension: .csv
* Location relative to PLS.py is: data/import.
* Contents format example:

"userid","username","surname","age","password","role"

1,"tobias","roessingh","33","test123","member"

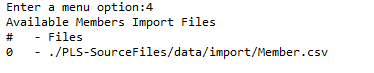
2,"michael ","heerkens","33","test123","member"

3,"jeremey ","sas","33","test123","member"

* Only lowercase letters are allowed for username column.
* The separator: Comma (,)

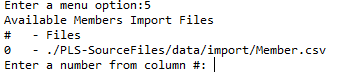
### List member imports (CSV)

From the member management menu, press option 4 and press enter to list all available CSV files to use as import, see screenshot below:

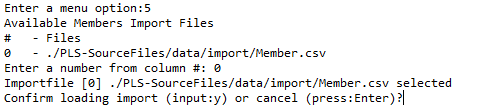


### Import members from CSV

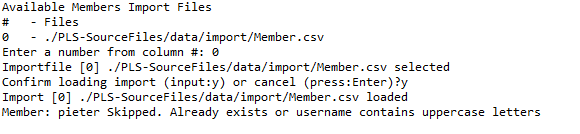
From the member management menu, press option 5 and press enter start importing users from the CSV import file, see screenshot below:



After this press any available import file to use and press enter, see screenshot below:



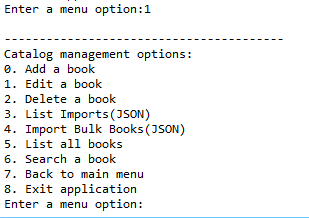
After this this the user is required to either confirm the import, by pressing y and enter, or cancel the import by only pressing enter, see screenshot below:



Note: If a member already exists, the import will skip importing this member.

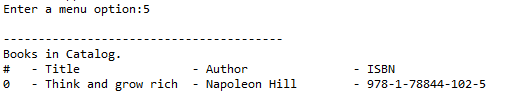
## Managing catalog

From the main menu select the option 1 and press enter to go to the catalog management menu, see screenshot below:



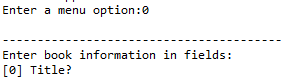
### List all books in catalog

From the catalog management menu, press option 5 and press enter to list all the books in the catalog, see screenshot below:

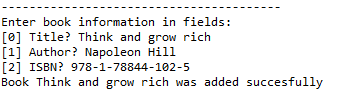


### Add a new book

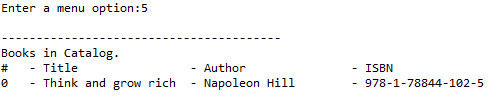
From the catalog management menu, press option 0 and press enter to begin adding a new book, see screenshot below:



After filling all the fields, the user is met with the following message:

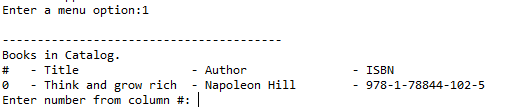


After this the book has been added:

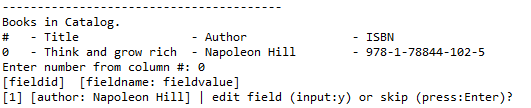


### Edit an existing book

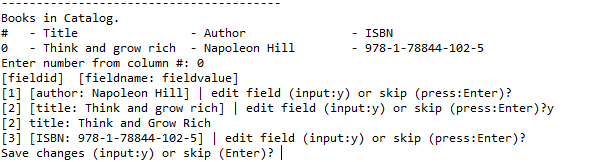
From the catalog management menu, press option 1 and press enter to begin editing a book, see screenshot below:



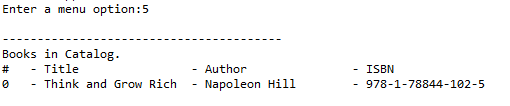
From here select a book by entering the id and pressing enter:



After this the user can select to either modify a field, by pressing y and enter, or to skip to the next field by pressing enter. At the end the user needs to confirm the action by pressing y and enter, or to cancel the modification by only pressing enter, see screenshot below:

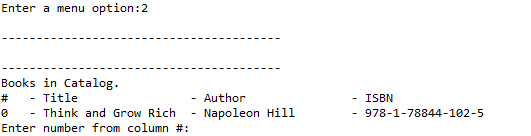


After this the book has been modified:

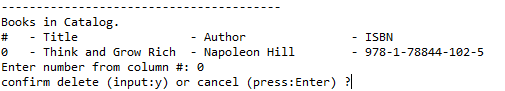


### Delete an existing book

From the catalog management menu, press option 2 and press enter to begin deleting a book, see screenshot below:

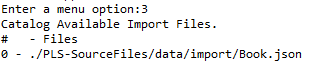


Select a book by entering the id and pressing enter, after this the user needs to confirm the action by pressing y and enter, or cancel the deletion by only pressing enter, see screenshot below:



### List available import files

From the catalog management menu, press option 3 and press enter to begin editing a book, see screenshot below:



### Books import file requirements

File requirements:

* File extension: .json
* Location relative to PLS.py is: data/import.
* Contents format example:

[

    { *"id"*: 0, *"author"*: "J.K. Rowling", *"title"*: "Harry Potter 1", *"ISBN"*: "978-6-6442-4397-3" },

    { *"id"*: 0, *"author"*: "J.K. Rowling", *"title"*: "Harry Potter 2", *"ISBN"*: "978-6-1276-3992-7" },

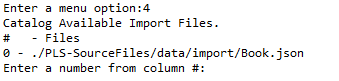
    { *"id"*: 0, *"author"*: "J.K. Rowling", *"title"*: "harry potter 3", *"ISBN"*: "978-7-0392-1683-1" },

    { *"id"*: 0, *"author"*: "JRR Tolkien", *"title"*: "Lord of the Rings", *"ISBN"*: "1234234234" }

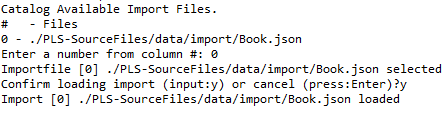
]

### Import books from import file

From the catalog management menu, press option 4 and press enter to begin editing a book, see screenshot below:

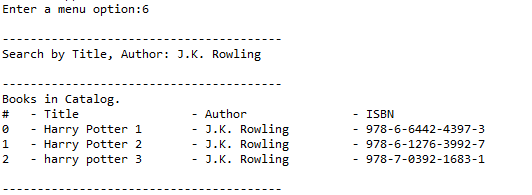


The user can select the file to import the books from and press enter, after this the user needs to confirm the input by pressing y and enter, or cancel it by nly pressing enter, see screenshot below:



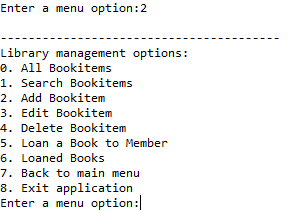
### Search specific book

From the catalog management menu, press option 6 and press enter to search a book, after this the user can enter a title or the author’s name to search all the book, see screenshot below:



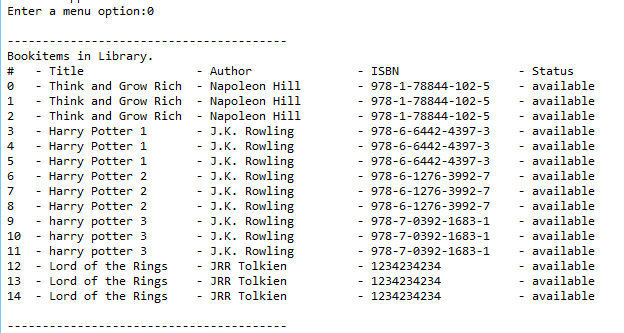
## Manage library

From the main menu select the option 2 and press enter to go to the library management menu, see screenshot below:



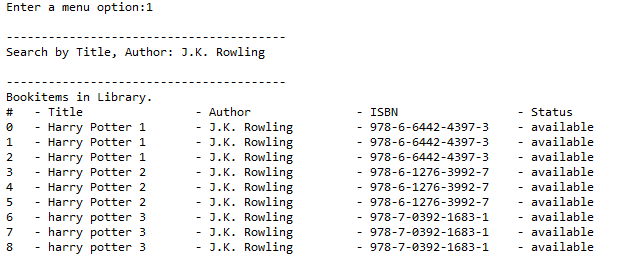
### List all book items

From the library management menu, press option 0 and press enter to list all book items, see screenshot below:



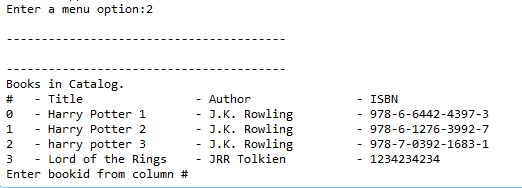
### Search book item

From the library management menu, press option 1 and press enter to search a book item, after this the user can enter either the title or author of a book to search book items, see screenshot below:

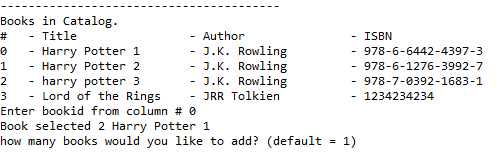


### Add new book item

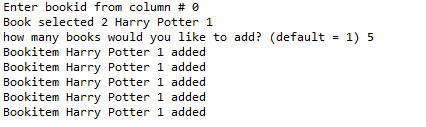
From the library management menu, press option 2 and press enter to add a new book item, see screenshot below:



After this the user can select a book to create a new book item by entering the book id and pressing enter, see screenshot below:

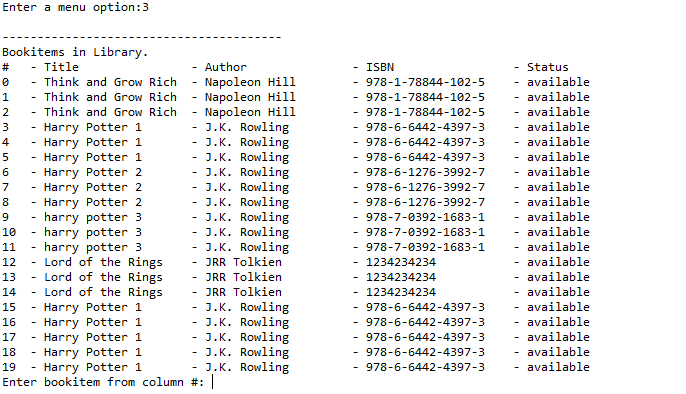


After this the user can select how many book items should be added of this book by entering the count and pressing enter, see screenshot below:



### Edit book item

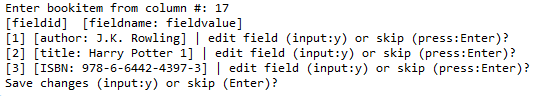
From the library management menu, press option 3 and press enter to edit a book item, see screenshot below:



Enter a book item id to start editing the book item, see screenshot below:

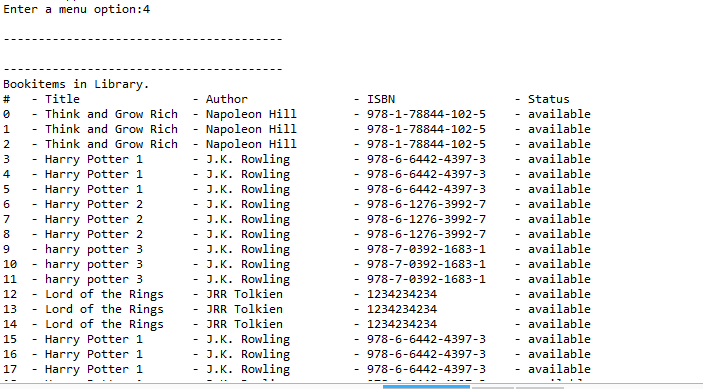


After editing all the fields, the user needs to confirm the modification by pressing y and enter, or cancel by only pressing enter, see screenshot below:

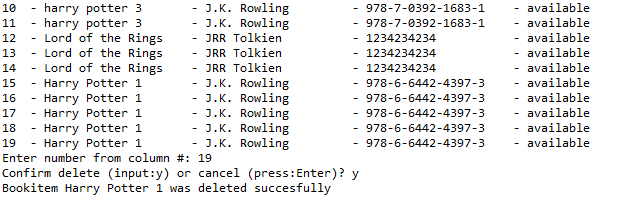


### Delete book item

From the library management menu, press option 4 and press enter to delete a book item, see screenshot below:

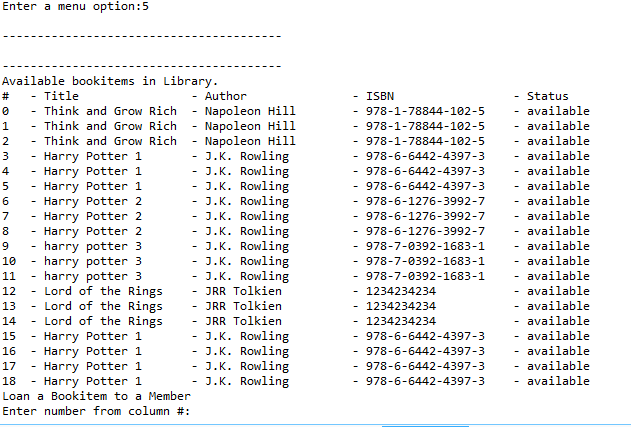


After this the user needs to select the item to delete by entering the id and pressing enter, after this the user needs to either confirm the action by pressing y and enter, or cancel the deletion by only pressing enter, see screenshot below:

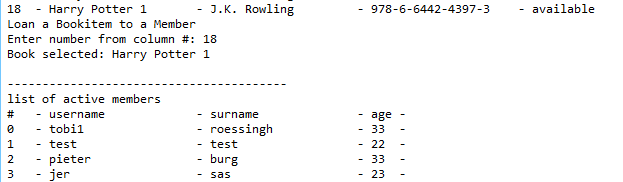


### Loan a book item

From the library management menu, press option 5 and press enter to loan a book item, see screenshot below:



The user can select an available book item by entering the id and pressing enter, after this the user needs to select a member to loan the item to, see screenshot below:

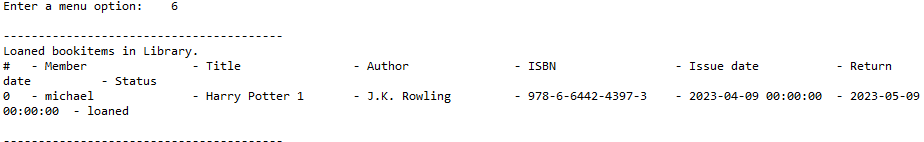


After entering the member id and pressing enter the book item will be loaned to the selected member, see screenshot below:



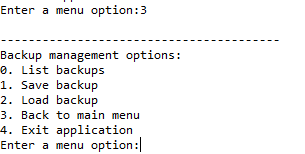
### List loaned items

From the library management menu, press option 6 and press enter to list all the loaned book items, see screenshot below:



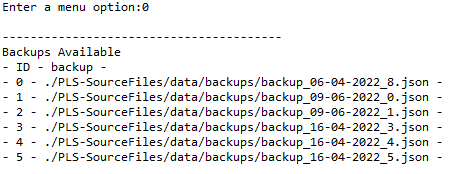
## Manage back-ups

From the main menu select the option 3 and press enter to go to the back-up management menu, see screenshot below:



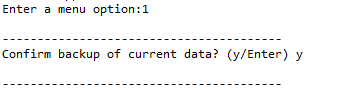
### List all back-ups

From the back-up management menu, press option 0 and press enter to list all the back-ups, see screenshot below:



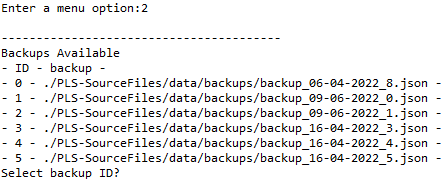
### Create back-up

From the back-up management menu, press option 1 and press enter to create a new back-up, after this the user needs to confirm the action by pressing y and enter, see screenshot below:



### Load back-up

From the back-up management menu, press option 2 and press enter to load a back-up, after this the user needs to select a back-up, see screenshot below:



After this the user can select a back-up by entering an id and pressing enter, after this the user needs to confirm the action, see screenshot below:

