|  |
| --- |
| Software Requirements Specification document for a Public Library System |
|  |
| June 11  Authored by: Ilias El Bouchtaoui 0996736 |

Table of Contents

[1. INTRODUCTION 3](#_Toc101563547)

[PURPOSE 3](#_Toc101563548)

[DOCUMENT CONVENTIONS 3](#_Toc101563549)

[INTENDED AUDIENCE AND READING SUGGESTIONS 3](#_Toc101563550)

[PROJECT SCOPE 3](#_Toc101563551)

[REFERENCES 4](#_Toc101563552)

[2. OVERALL DESCRIPTION 4](#_Toc101563553)

[PRODUCT PERSPECTIVE 4](#_Toc101563554)

[PLS REQUIREMENST 5](#_Toc101563555)

[Administrator requirements 5](#_Toc101563556)

[Member requirements 6](#_Toc101563557)

[System requirements 6](#_Toc101563558)

[Other requirements 7](#_Toc101563559)

[OPERATING ENVIRONMENT 8](#_Toc101563560)

[DESIGN and IMPLEMENTATION CONSTRAINTS 8](#_Toc101563561)

[PRODUCT FEATURES 8](#_Toc101563562)

[3. INTERFACE REQUIREMENTS 12](#_Toc101563563)

[USER INTERFACES 12](#_Toc101563564)

[4. USER MANUAL 12](#_Toc101563565)

[System Requirements 12](#_Toc101563566)

[Working with the PLS 12](#_Toc101563567)

[Starting the program 13](#_Toc101563568)

[Admin 13](#_Toc101563569)

[Member 17](#_Toc101563570)

# INTRODUCTION

## PURPOSE

The purpose of the PLS management system is to ease book loan management of a library.

## DOCUMENT CONVENTIONS

This document uses the following conventions.

|  |  |
| --- | --- |
| Admin | Administrator |
| CSV | Comma Separated Value |
| JSON | JavaScript Object Notation |
| PLS | Public Library System |
| RAM | Random Access memory |
| UML | Unified Modeling Language |
| MoSCoW | Must Have, Should Have, Could Have |

## INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for Public Library System, and it is restricted within the college premises. This has been implemented under the guidance of a college. This project is useful for the PLS management team and as well as to the Members.

## PROJECT SCOPE

This assignment consists of the design and implementation of a Public Library System, which will be called PLS, afterward. The required functionalities of the system are according to the descriptions provided in this document, which might be slightly different from a real public library system; rather, these functionalities are defined according to the course intended learning outcome and assessment objectives. Such a library system usually involves a database, but here the assignment is limited to storing data in RAM and files. The system will consist of a console based (textual) interface that gives access to all functions of the system.

Only the first three of the SDLC Phases are covered and should be worked out.

* Requirement gathering and analysis
* Design
* Implementation or coding

The whole issue of the security is left out of scope unless explicitly stated out elsewhere in this document.

## REFERENCES

<https://nl.wikipedia.org/wiki/Unified_Modeling_Language>

<https://miro.com/templates/moscow-matrix/>

<https://github.com/0996736Ilias/schoolopdracht>

# OVERALL DESCRIPTION

## PRODUCT PERSPECTIVE

A PLS stores the following information.

Book Loan details:

This includes books, customers, book items and book loan details.

## PLS REQUIREMENST

The PLS is to ease the process of book item loan and will have the following requirements:

The are two roles that can login tot the system PLS:

* Customer
* Administrator

The username for the admin is “Admin” with a password = “admin123”. If a person uses these credentials to login this user will be recognized as an Administrator otherwise it’s a member (customer).

These two roles have different functions. The prioritization will be done according to the MoSCoW (M, S and W) method.

MoSCoW is an acronym for Must Have, Should Have, Could Have, and Won’t Have. These four priority categories make up the four segments in the matrix. “Must Have” items are necessary for delivery; “Should Have” items are important but not necessary; “Could Have” items are nice to have (they are not priorities, but your team can work on them if time and resources permit); and “Won’t Have” items do not fit into the scope of the current project.

In our case alle the minimum requirements in the assignment are Must Haves (M).

### Administrator requirements

Requirements related to members:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| See the list of members | M |
| Add, edit, and delete a member from the system. Including username and password | M |
| Check the status of book items currently loaned by a member | M |
| Load and add a list of members to the system, all at once (using a csv file) | M |

Requirements related to the catalog:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| Check the catalog (list of books) | M |
| Add, edit, and delete a book to/from the catalog | M |
| Search a book in the catalog | M |
| Load and add a list of books to the catalog, all at once (using a json file) | M |

Requirements related to book items:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| See the list of book items in the library | M |
| Add, edit, and delete a book item to/from the library | M |
| Search a book item and its availability in the catalog library | M |
| Lend a book item to a member | M |

Requirements related to system administration:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| Make backups of the system | M |
| Restore a specific backup of the system | M |

### Member requirements

Requirements related to the catalog:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| Check the catalog (list of books) | M |
| Search a book in the catalog | M |

Requirements related to book items:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| See the list of book items in the library | M |
| Search a book item and its availability in the catalog library | M |
| Loan a book item | M |
| A member can borrow a maximum of 3 books simultaneously | M |
| A member can borrow a book for a maximum period of 1 month (30 days) | M |
| Return a loaned book item | M |

### System requirements

All the following requirements are Must haves.

Configuration of the system:

When the system starts, it must check the data file(s) for catalog, book items, members, etc.

If the data file(s) are already there, the system should recognise it, otherwise the system may need to properly initialize and configure the data file(s).

Requirements related to loan administration:

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| The library should initially have 3 copies (book items) of each book | M |
| Book search function must accept book title or author as search key | M |
| Book search function must accept partial keys, lowercase, and uppercase letters (it should not be case-sensitive). For example, a search with author= “Hem” or “hemi” or “ming” must return “Ernest Hemingway”, if such author exists in the data file. | M |

Requirements related to system administration:

Name of backup files should contain the date of backup, plus a serial number for the multiple backup on the same date

### Other requirements

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| All data of PLS must be saved on a file after every transaction of the system.  Members are unique and should not be repeated. | M |
| Books are unique. There will be only one representation of a book in the system. | M |
| Book items are unique | M |
| There is no special formatting requirements for the passwords | M |
| a username must be lowercase | M |
| Menus should be chosen by one digit or one letter, only. See example below | M |

Menu example:

Enter 1 or 2 to choose a menu:

[1] Add member

[2] Search Book

## OPERATING ENVIRONMENT

Operating environment for PLS is as listed below.

Client/server : Local Client system

Operating system: Windows. Mac OS

database: csv, json

platform: Python

## DESIGN and IMPLEMENTATION CONSTRAINTS

The PLS system must be built only in Python 3 using only modules in the standard library of python 3.

## PRODUCT FEATURES

The major features of PLS as shown in below UML Diagram.

I have chosen to make a Member and a Admin class including a database (json) file for each object. I Could have done this with a Boolean field in the person table. That would be much better.

I dint use unique id’s for the book items. That too would have made the system professional but didn’t have the time to realize it.

Much of the names for the functions could be better. I realized that it is more readable to use functional names like Save instead of WriteTodabase for Book Class. Or GetBook() instead of readFromdatabase().

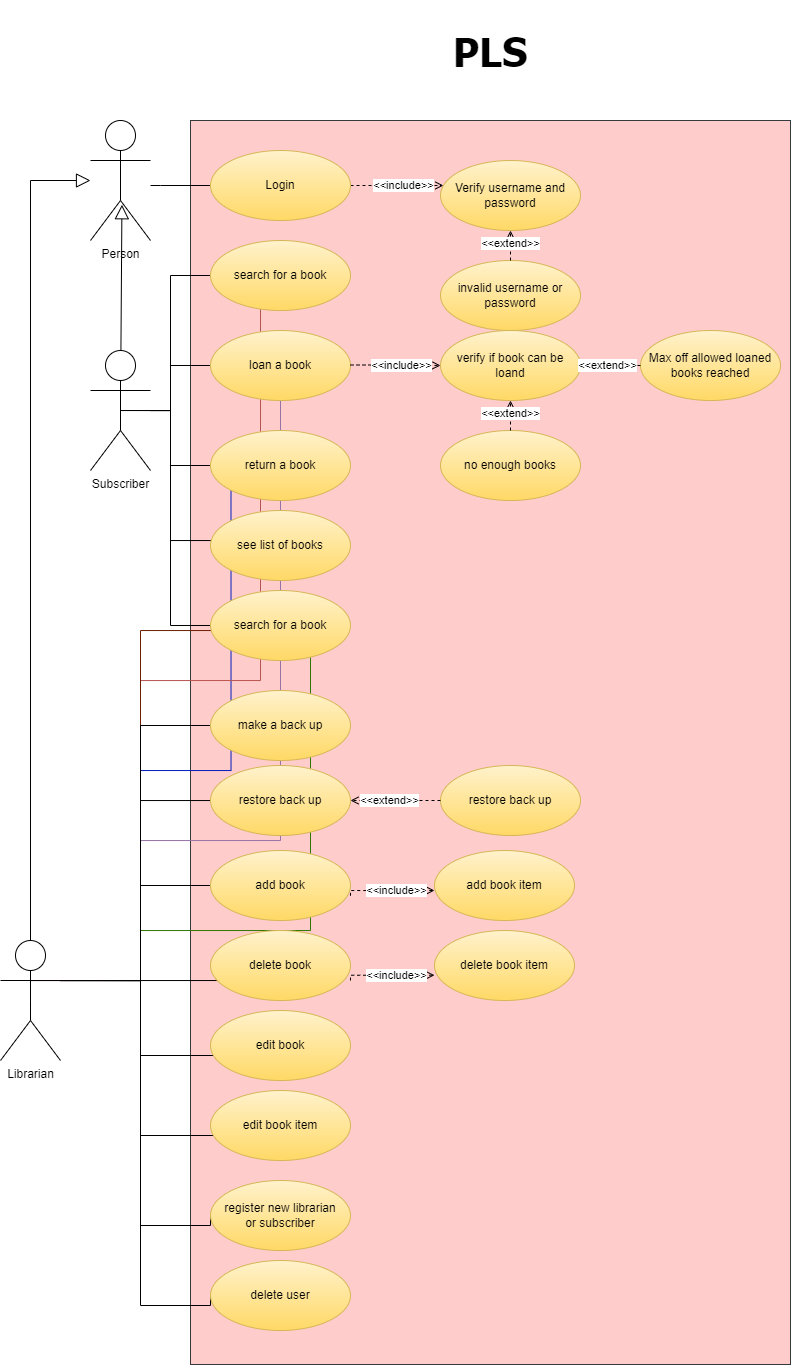
Below is the UML of the PLS.

UML Diagram

Timeline

Description automatically generated

Use Case Diagram



# INTERFACE REQUIREMENTS

## USER INTERFACES

There will be no special requirements for the user interface others than stated in requirements.

A Command window will be used as the interface of the PLS.

# USER MANUAL

## System Requirements

Processor: 1 GHz or faster x86 or 64-bit processor; Intel processor (Mac)  
Memory : 1 GB RAM (32-bit or Mac) /2 GB RAM (64-bit)  
Operating System : Windows Or Mac OS

Software: Python 3.0 installed and working  
Tools : Command line tool

We will run our program on on **CMD**or **command prompt**as CMD is the default command-line interpreter on Windows.

We assume that you have a knowledge of using windows and starting a cmd session.

## Working with the PLS

First you have to login before you can do anything with the PLS.

You can login in as an Administrator (Admin) or as a Member.

As an Admin you have the following functionality:

* See the list of members,
* Add, edit, and delete a member from the system,
* Check the status of book items currently loaned by a member,
* Load and add a list of members to the system, all at once (using a csv file),
* Check the catalog (list of books),
* Add, edit, and delete a book to/from the catalog,
* Search a book in the catalog,
* Load and add a list of books to the catalog, all at once (using a json file),
* See the list of book items in the library,
* Add, edit, and delete a book item to/from the library,
* Search a book item and its availability in the catalog library,
* Lend a book item to a member,
* Make backups of the system,
* Restore a specific backup of the system

As a Member you have the following functionality:

* Check the catalog (list of books),
* Search a book in the catalog,
* See the list of book items in the library,
* Search a book item and its availability in the catalog library,
* Loan a book item,
* Return a loaned book item

## Starting the program

1. Open a command tool like cmd.exe on windows,
2. Navigate to your directory where the PLS resides, **CD <”Your directory”>**
3. Start the program by typing “python PLS”

## Admin

* Type a username = *admin*
* Press <ENTER>
* Type password = *admin123*
* Press <ENTER>.

Now you are an Administrator of the PLS.

The next menu will be presented to you :

1. Search book
2. Search book item
3. Add book
4. book list
5. user list
6. Delete a user
7. Register new user
8. Logout
9. Make backup
10. Restore backup
11. Edit user
12. Edit book
13. Edit book item
14. Delete book item
15. Delete book
16. Return book

Choose from 1 to 16 and press <ENTER>.

1. Search Book.  
   This gives the possibility to search a book. You can search in the title, author or country.  
   Choose title, author or country. Press <ENTER>  
   Type your search criteria. Press <Enter>

If there is aa book matching your search criteria it will be presented to you otherwise it will message “No results found” and go back to the main menu.

1. Search book item  
   Give an exact text you want to search in the book items  
   Press <ENTER>  
   A List of book items will be shown.
2. Add book  
   Add a book to the PLS  
   Fill in all the required fields.  
   At the end a book with x copies (book items) will be added to the PLS.
3. book list  
   Presents a list of all the books in the library.
4. User list  
   List all the users in the PLS.
5. Delete a user  
   Enter the userid as shown in the userlist (to delete the user).  
   Press <Enter>  
   A user will be removed from the PLS after the PLS is closed an started.
6. Register new user  
   To add a new user as Member or Admin.  
   Fill in alle the required fields,  
   At the end choose for Admin of Member.
7. Logout  
   To log you out and present the login screen.
8. Make backup  
   Make a backup of the PLS.
9. Restore backup  
   To restore a backup.  
   PLS will preset you with all backups.  
   Select one of the backups and press <ENTER>  
   The chosen backup will be used as a restore.
10. Edit user  
    Select a user number to edit.  
    Give the new values for the fields of the user.
11. Edit book  
    Type in the number of the book.  
    Edit the fields
12. Edit book item  
    Give the ISBN number to edit a book item.   
    Press <ENTER>  
    Fill in the required fields.
13. Delete book item  
    To remove a book item from the PLS.  
      
    Enter ISBN number  
    Press <ENTER>
14. Delete book  
    Delete a book from the PLS.  
      
    Type in the number of the book.  
    Press <ENTER>
15. Return book  
    To return a book to the library.  
    Give the ISBN number of the book and press <ENTER>  
    Give the user id (number) and press <ENTER>

## Member

1. Search book

2. Logout

3. Book list

4. Return book

5. Edit current user

1 to 3 are the same as the admin. See admin manual.

Number 4 is the same as admin except for the user. You don’t need an user number. You are the user.

The same with choice number 5.