

* Diploma in software development (Level 6) – Year 1 – Online Campus – 20/11/2022 2202-Online-DSD
* Student ID: 270195879 - Jisu Jeon.
* Student ID: 270181332 – Ricardo Diaz Rodriguez.
* Project start date: 22/10/2022
* Project end date: 20/11/2022
* Name of the project: Library Information System.

1.PROJECT DESCRIPTION

GOAL.

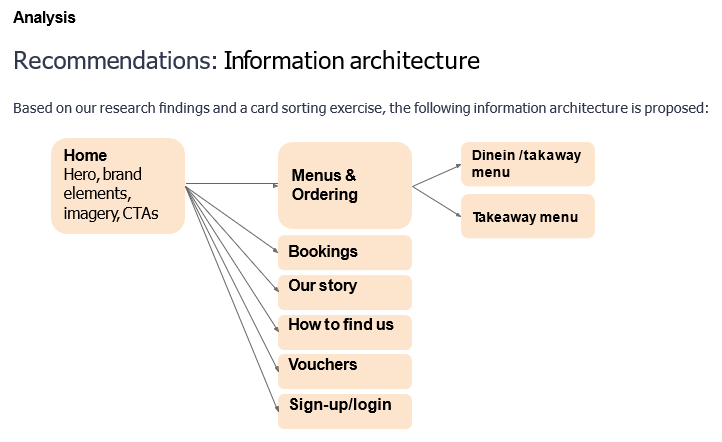
To design and develop library information system to manage all the in-house functions of a library and perform basic library activities such as registration, management of members, managing book status, etc.

OBJECTIVES.

* To investigate to establish the client’s business needs.
* To determine software process model to be used.
* To identify functional and non-functional requirements
* To propose functionalities given the result of the investigation.
* To provide a UI design system adjusting it to the client’s needs.
* To gather accurate information through the user testing tool Maze.
* Details about project timelines, constrains team members and responsibilities.
* Software Process Model (Iterative).
* List of tools and technologies (documented).
* Revisions to original project description.

2.SYSTEM MODELLING.

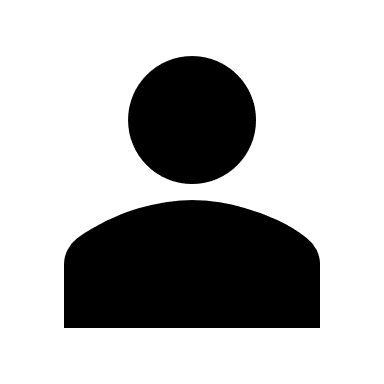
* Description of overall system architecture



* Components Functionalities and Interfaces.
* Case Diagrams.

**Use case: Search books**

**Diagram**



SEARCH-BOOK

MEMBER

SEARCH BOOK

**Brief Description**

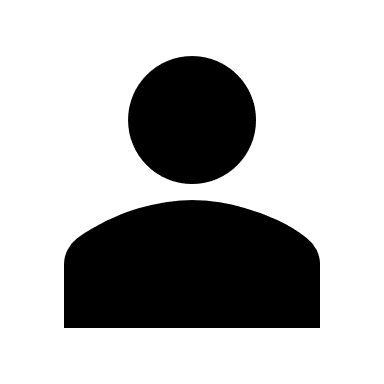
The member access to the app, searches the book using either name or author.

**Step by step description:**

1. Member chooses to search by book name, author name, category or keyword.
2. System displays books matching the data inputted by the member.
3. Member chooses a book,
4. System displays the book information as well as availability for each location.
5. Member can choose to put the item on hold if its unavailable.
6. System puts the book on hold if member chooses to and informs the member with booking details.

**Use Case: Pre-book**

**Diagram**



PRE-BOOK

MEMBER

**Brief Description**

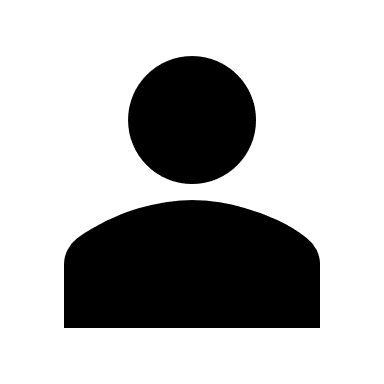
The member access to the app, searches the book using either name or author then, depending on availability to pre-book a book.

**Step by step description:**

1. Member select a book from categories, catalogue list, recently added etc.
2. System displays the book selected.
3. Member add the book clicking on pre-book button,
4. System displays success window.

**Use case: Login**

**Diagram**



ADMIN

LOGIN

**Brief Description**

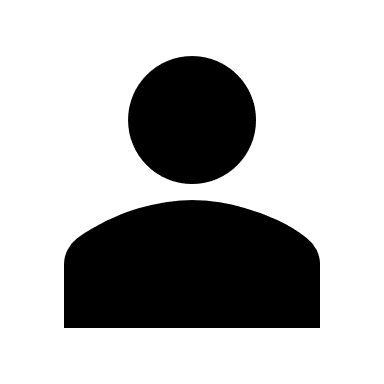
Administrators login

**Step by step description:**

1. Admin launches the app.
2. Admin fill the login details boxes.
3. Admin clicks on admin login,
4. System automatically redirect to the window “Admin Profile”.

**Use case: Add book**

**Diagram**



ADMIN

ADD BOOK

**Brief Description**

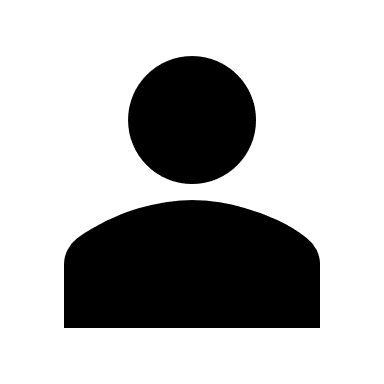
The member access to the app, clicks on hamburger menu to access to the window “Add Book”.

**Step by step description:**

1. Admin log in as administrator.
2. System automatically redirect to Admin Profile window.
3. Admin clicks on hamburger menu.
4. Admin clicks on Add Book link.
5. Admin adds information required.
6. Admin clicks on diskette icon to save the changes.
7. System displays success window.

**Use case: Sign up**

**Diagram**



MEMBER

SIGN UP

**Brief Description**

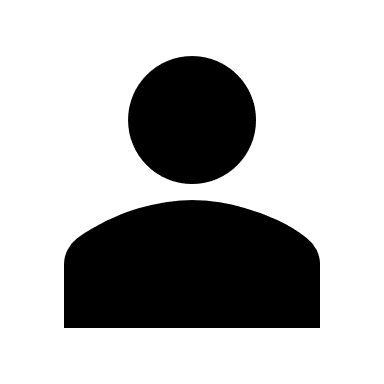
The member access to the app to sign up and explore the library system.

**Step by step description:**

1. Member launches the app.
2. Member fills up sign up detail’s boxes.
3. Member clicks on sign up button,
4. System automatically redirect to the window “Home”.

**Use case: Sign up**

**Diagram**



EDIT CATALOGUE

ADMIN

**Brief Description**

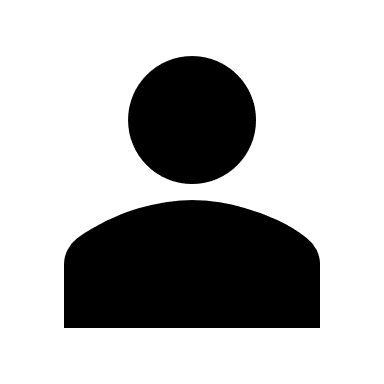
The admin access to the app and login as administrator to edit book’s catalogue.

**Step by step description:**

1. Admin log in as administrator.
2. System automatically redirect to Admin Profile window.
3. Admin clicks on hamburger menu.
4. Admin clicks on Catalogue link.
5. Admin can search by book name, author name, category or keyword.
6. Admin clicks on edit button.
7. System redirect to “Modify Catalogue” window.
8. Admin edits information needed or adds pictures.
9. Admin clicks on diskette icon to save the changes.
10. System displays success window.

**Use case: Modify members**

**Diagram**



MODIFY MEMBERS

ADMIN

**Brief Description**

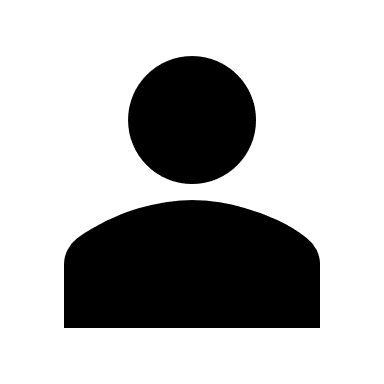
The admin access to the app and login as administrator to modify existing members.

**Step by step description:**

1. Admin log in as administrator.
2. System automatically redirect to Admin Profile window.
3. Admin clicks on hamburger menu.
4. Admin clicks on Members link.
5. Admin can search by member’s name and surname or ID, sort or filter users list.
6. Admin clicks on settings icon.
7. System redirect to “Member Details Admin Screen” window.
8. Admin edits information needed.
9. Admin clicks on diskette icon to save the changes.
10. System displays success window.

**Use case: Add members**

**Diagram**



ADD MEMBERS

ADMIN

**Brief Description**

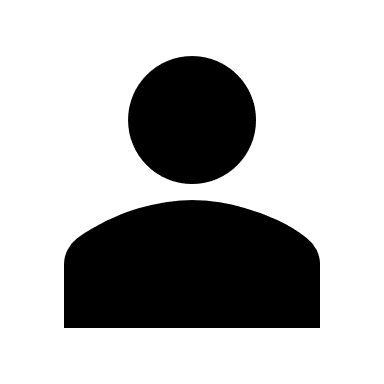
The admin access to the app and login as administrator to add new members.

**Step by step description:**

1. Admin log in as administrator.
2. System automatically redirect to Admin Profile window.
3. Admin clicks on hamburger menu.
4. Admin clicks on Members link.
5. Admin clicks on add user icon located at top app bar.
6. System redirect to “Add New Members” window.
7. Admin fills up sign up credentials.
8. Admin clicks on sign up button.
9. System displays success window.

**Use case: Log messages**

**Diagram**



LOG MESSAGES

ADMIN

**Brief Description**

The admin access to the app and login as administrator to log messages according status of books.

**Step by step description:**

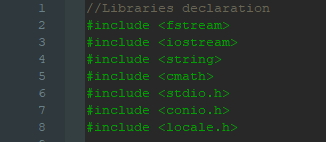
1. Admin log in as administrator.
2. System automatically redirect to Admin Profile window.
3. Admin Reports icon at bottom app bar.
4. System redirect to “Logs-Screen” window.
5. Admin chooses between three options, returned books, due-date books, or overdue books.
6. Admin clicks action button to log messages.
7. System displays success window.

3.IMPLEMENTATION

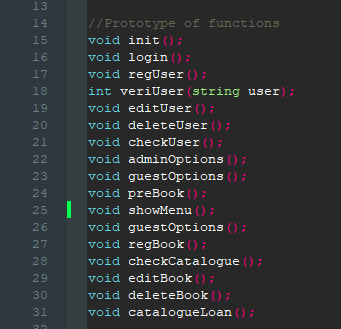
* Screenshots and code explanations.

Libraries declaration:

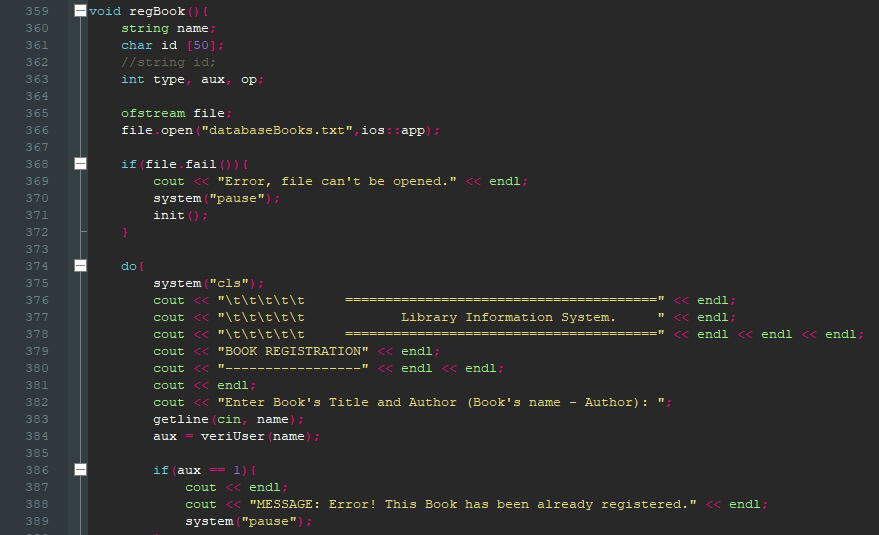
A very important library is fstream as we need to create files, write information to files and read information from them.



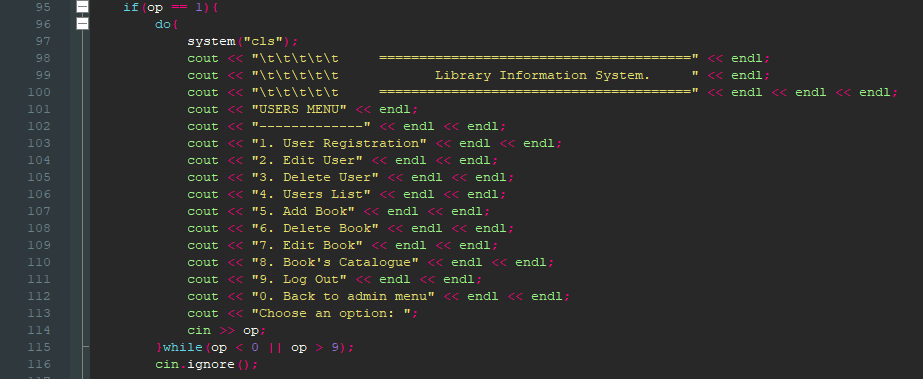
Prototype of functions helps to easily read the code. The main function calls init(); which starts a “do while” loop and shows a menu to register, login or exit the program. Depending on the choice made by the user an if conditional will direct the program to the function, in this case the functions are login(); to access functions depending on the type if user, regUser(); to register new users and exit(0); to close the program.



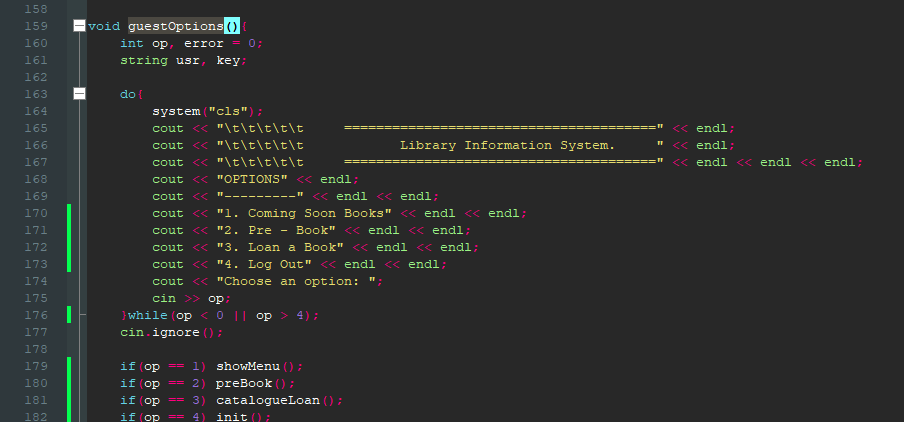
The program has many functions with especial functionalities as registering a new book, create a file for the first time and then just write the file with new records.



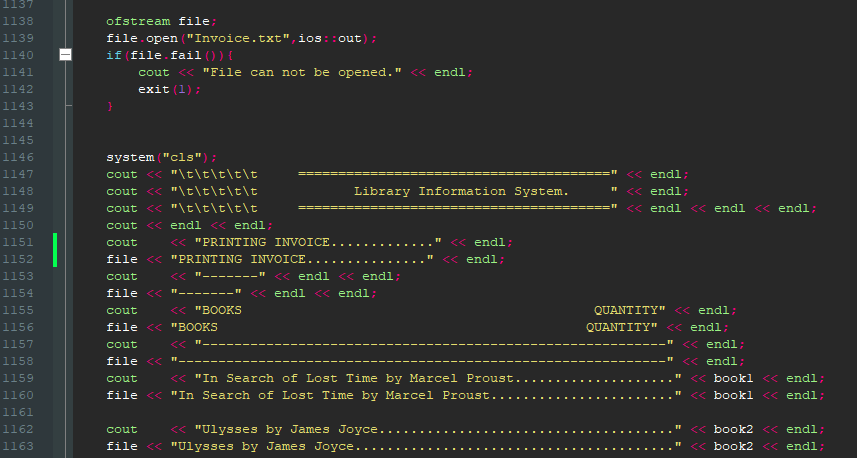
The program has different type of user access such as administrator or member.



Member’s menu. “Do while” loop that direct the program to exclusive members options such as loan a book or pre – book a book.



The system prints an invoice depending on the book that members want to pre-book and it logs a .txt file.



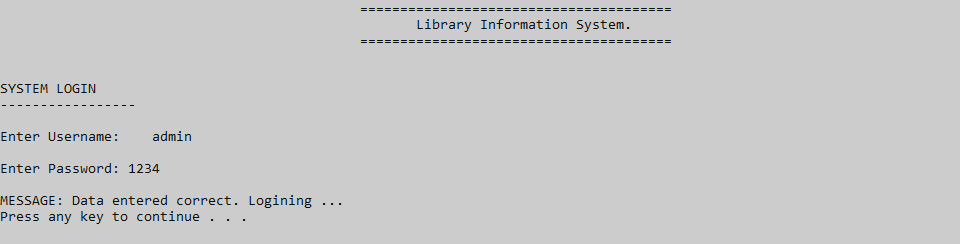
4.FUNCTIONALITY TESTING

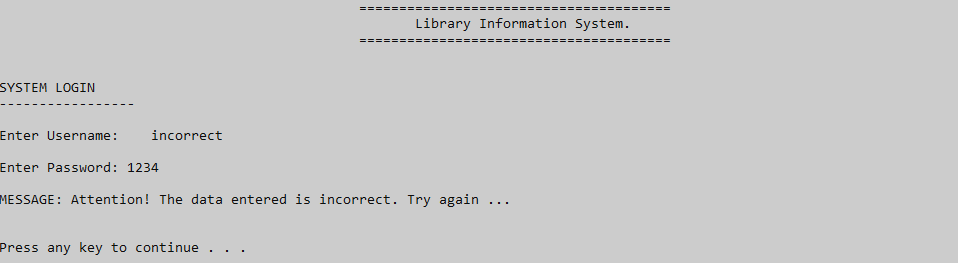
* Black Box Testing.10 Test cases to test functionality.

Functional testing is the most appropriate to be applied on the current project as this kind of test covers how well the system executes its functions.

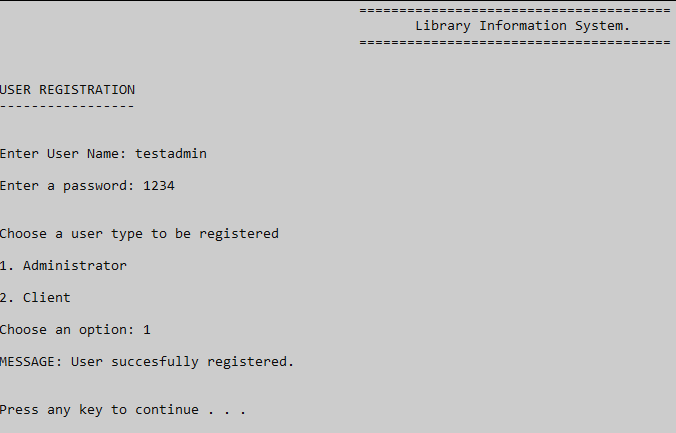
Technique of black box testing: Functional requirement / Use case.

1. Case description: Admin login screen. The screen has two fields to fill username and password and the system should show an error screen when login credentials are not correct. No error observations.

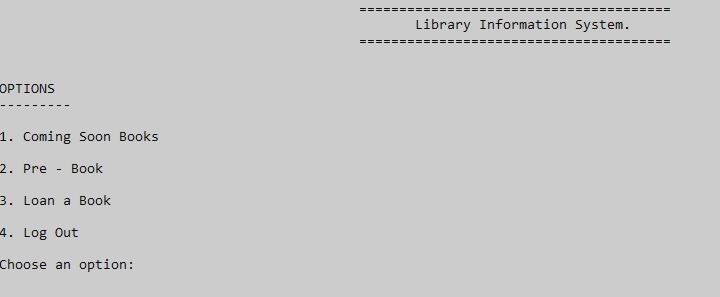




1. Case description: Register screen. The screen has two fields to fill username and password the system should show a success screen when the registered data fits the parameters. Errors observations, sometimes the program cannot open the file.

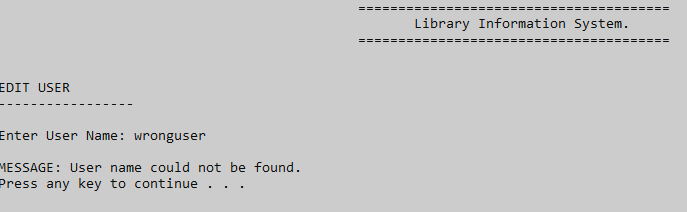


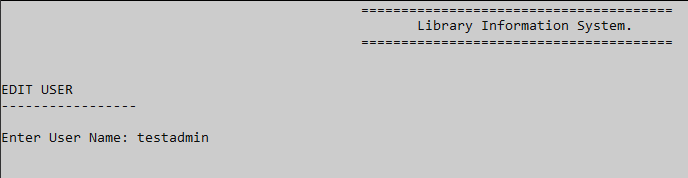
1. Case description: Member check coming soon books. The screen shows four options and just returns data of books. Pre-book option added. No error observations.

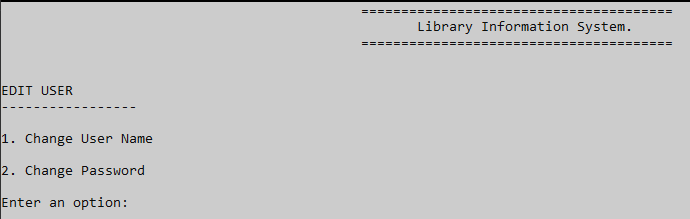


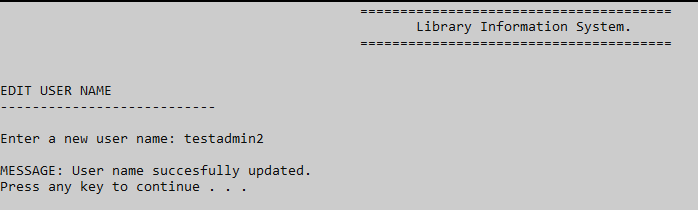


1. Case description: Edit users. The screen ask user to input user’s name, if correct, the system shows two fields to change username and password to change details, if wrong, systems print an error message user could not be found. No error observations.

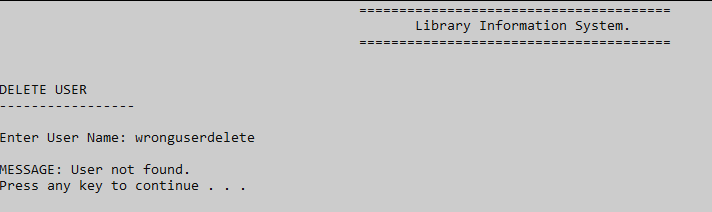


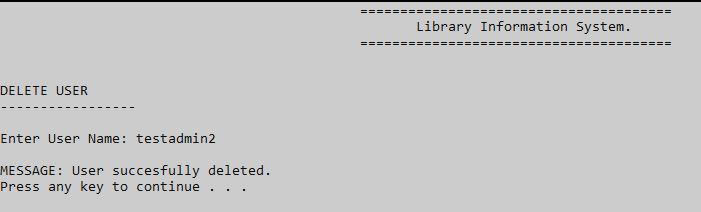




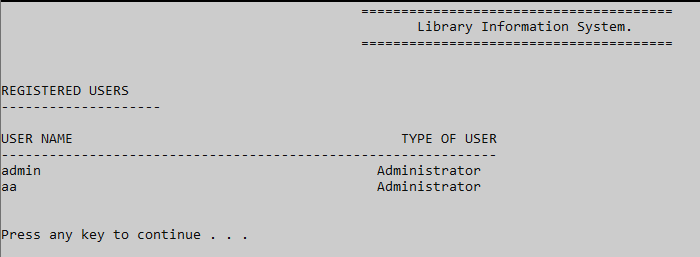


1. Case description: Delete users. The screen asks to the user to enter user’s name, if correct, the system deletes it, if wrong, systems print an error message user could not be found. No error observations.



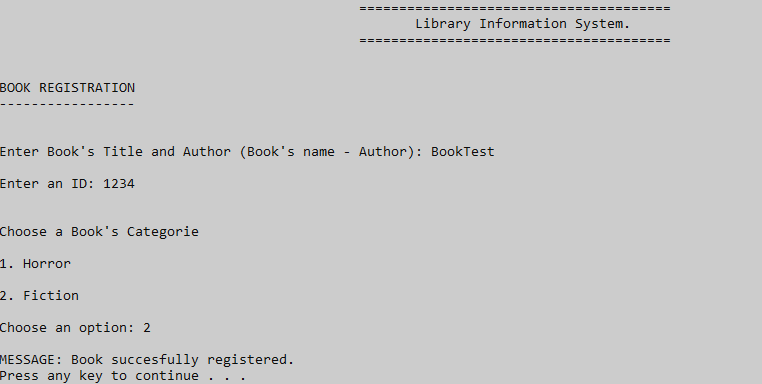


1. Case description: Users database. The screen just return data from the document used to store user’s information. No error observations.

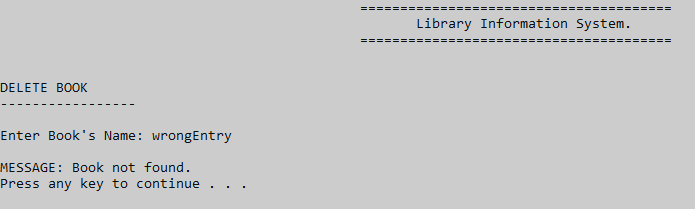


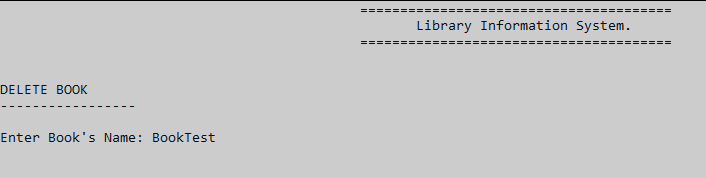
1. Case description: Add new book. The screen asks to the admin to enter the name of the book, an ID and a category. Errors observed when saving data on the .txt document. It might be by entering special characters or spaces between book’s title and author.

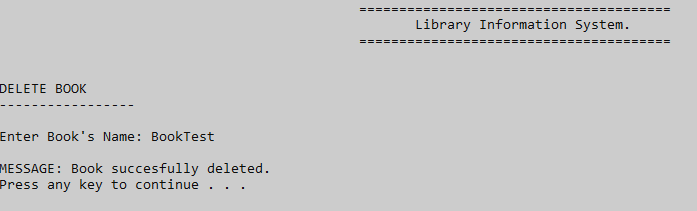
Recommendation: To change the way that information is stored.



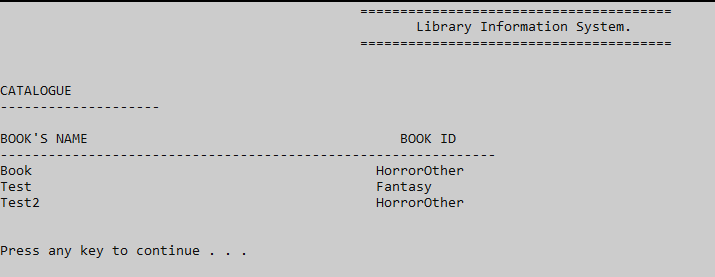
1. Case description: Delete book. The screen asks to the admin to enter the name of the book, if the book don’t exist, the system shows an error “Book not found”, if the book exists on the database, the system deletes it.



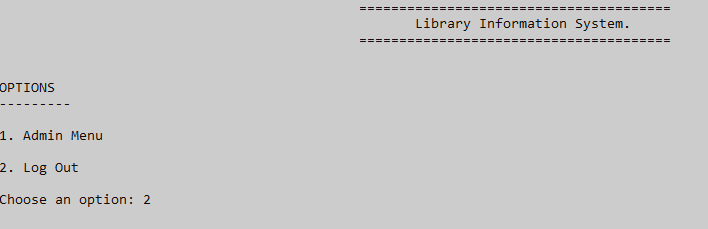


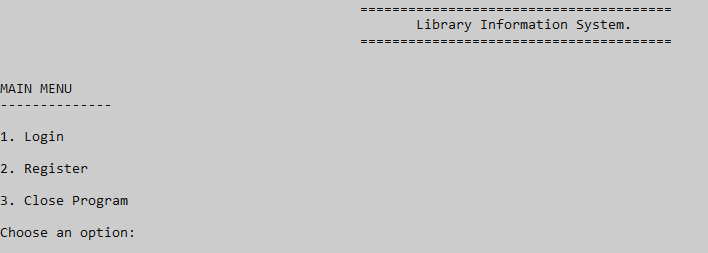


1. Catalogue: Book’s Catalogue. The system just returns data of catalogue registered by admins. No errors or observations.



1. Case description: Log out. A logout screen is settled on main screens. No errors or observations.





* Testing results.

Black box testing helps to understand and improve functionalities. On this testing, some errors came up and some observations were made.

In general terms, system storage need to be improved as well as the distribution of some menus, a written report need to be send to developers in order to adjust functionalities.

5.USER DOCUMENTATION

* User installation guide.
* User manual or user guide (Screenshots of all applications flows)

6.SELF REFLECTION REPORT

* Self-reflection report including aspects that worked well and did not work well.

7.PRESENTATION

* To include slides on this document or explanation.