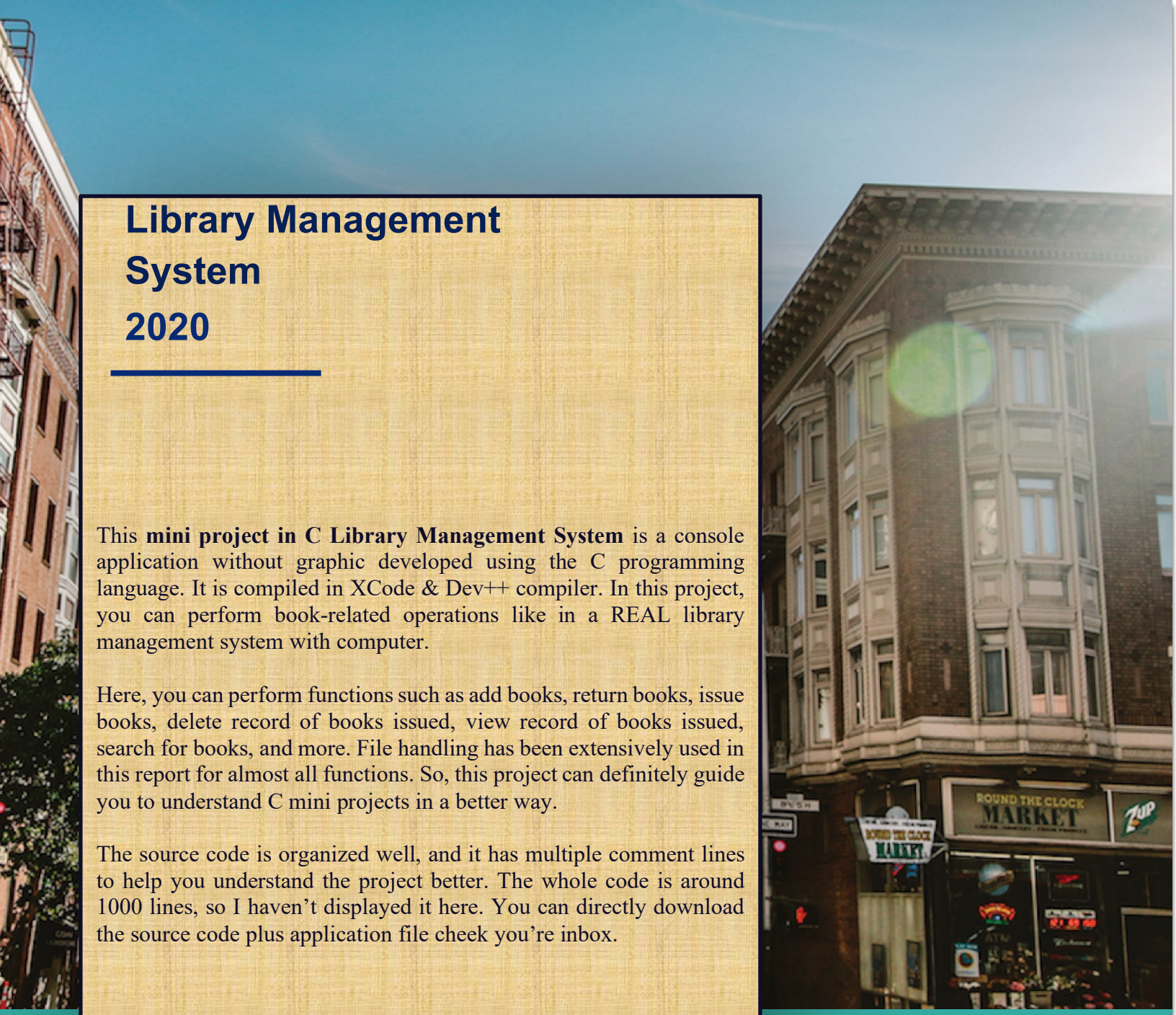
Library Management System 2020



This **mini project in C Library Management System** is a console application without graphic developed using the C programming language. It is compiled in XCode & Dev++ compiler. In this project, you can perform book-related operations like in a REAL library management system with computer.

Here, you can perform functions such as add books, return books, issue books, delete record of books issued, view record of books issued, search for books, and more. File handling has been extensively used in this report for almost all functions. So, this project can definitely guide you to understand C mini projects in a better way.

The source code is organized well, and it has multiple comment lines to help you understand the project better. The whole code is around 1000 lines, so I haven't displayed it here. You can directly download the source code plus application file cheek you're inbox.

MAY 12

Team name: NETWORKLAND™
Authored by: Yassen & Nouhaila

directed by: Bouziane Imane

Summary:

Team organization

(Distribution of tasks the duration of the project …)

Program organization

(role of the functions & explanations of the program)

Qualitative assessment of the work

(difficulties, screenshots, execution scenarios …)

Conclusion

(overview)

Team organization

Since my friend is my partner in the project, the communication between us was easy and effective and the work was comfortable, despite that, we created some rules to work on the project together fairly

⇒ **A project is "work"**

⇒ **Friendship is "personal"**

So, when "personal" meets "work", "professional" plays a key role. At work, you'd have to be very professional. You could try your best way to influence your friend for being professional and team up for the project.

The distribution of tasks was the easiest part of the project, the tasks were as follows:

1. **Work together on each line of the project and discuss it simultaneously.**

2. **(Nouhaila) corrected errors and improved the final form of the project.**

3. **(Yassin) wrote the project report.**

Since the delivery duration of the project was very sufficient, we didn't press ourselves with a specific schedule, and we used to work whenever the opportunity permitted. After 20 days we had finished the project completely.

Program organization

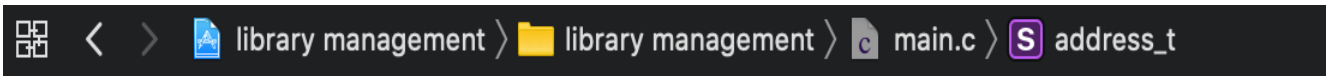
Library Management System

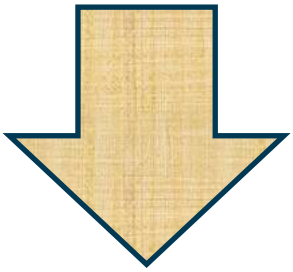
is an application created by Yassen & Nouhaila, which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used. In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

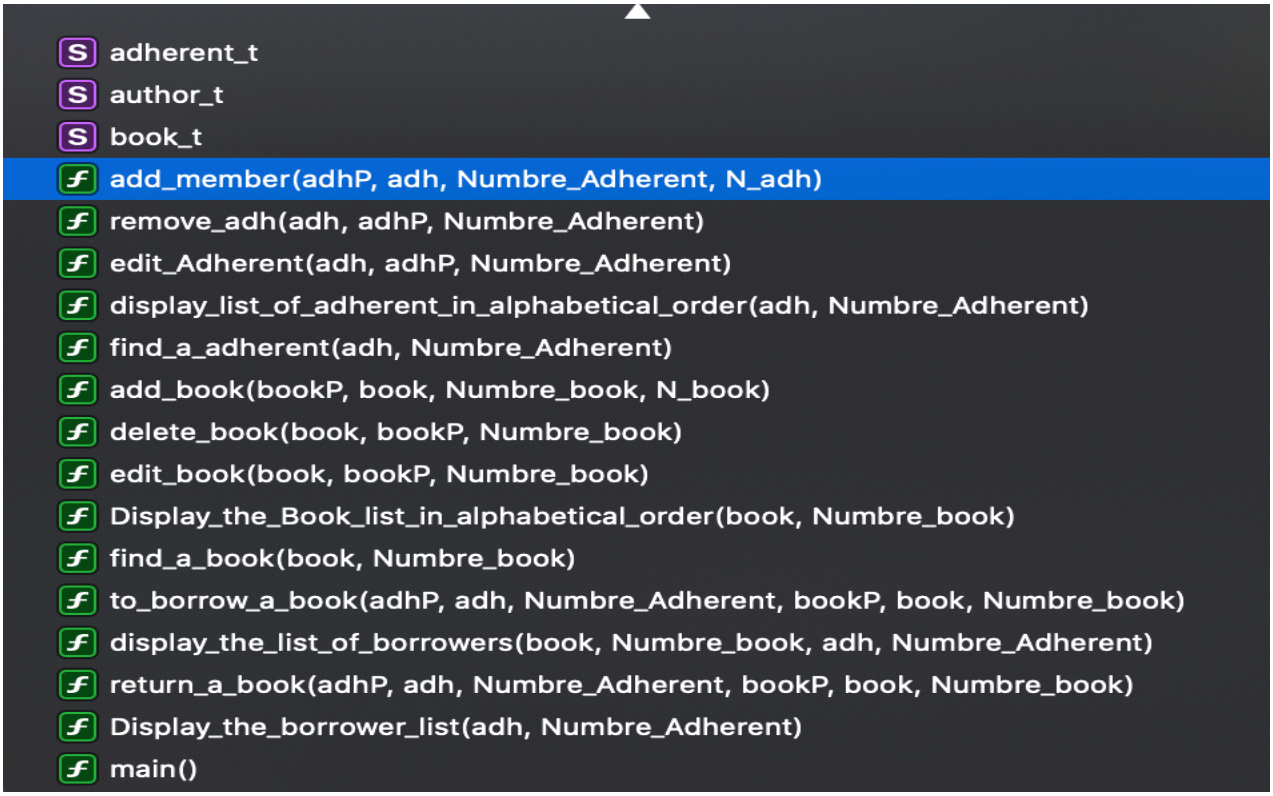
---

functions:

To find the required function easily, its recommended to use the tool bar:







⇒ add_member

This function opens the binary file in append mode and writes the adherents and the details.

```
52  void add_member (struct adherent_t *adhP, struct adherent_t adh [MAX_ADH], int *Numbre_Adherent, int *N_adh)
```

⇒ remove_adh

This function asks the adherent ID from the user to delete it.

```
99    void remove_adh (struct adherent_t adh[MAX_ADH], struct adherent_t *adhP, int *Numbre_Adherent)
```

⇒ edit_adherent

This function allows to edit the adherent's info

```
152    void edit_Adherent ( struct adherent_t adh [MAX_ADH] , struct adherent_t *adhP , int *Numbre_Adherent)
```

⇒ display_list_adh_alphabitical_order

This function shows the list of adherents by alphabetic order.

```
269 void  display_list_of_adherent_in_alphabetical_order ( struct adherent_t adh[MAX_ADH], int *Numbre_Adherent )
```

⇒ find_adherent

This function asks the user to enter the adherent name which one to search.

```
297  void find_a_adherent ( struct adherent_t adh[MAX_ADH] , int *Numbre_Adherent)
```

⇒ add_book

This function opens the binary file in append mode and writes the books and the details.

```
357  void add_book ( struct book_t *bookP , struct book_t book [MAX_BOOK] , int *Numbre_book , int *N_book)
```

⇒ delete_book

This function asks the book ID from the user to delete it.

```
397  void delete_book ( struct book_t book [MAX_BOOK] , struct book_t *bookP , int *Numbre_book )
```

⇒ edit_book

This function allows to edit the books info.

```
451  void edit_book ( struct book_t book [MAX_BOOK] , struct book_t *bookP , int *Numbre_book )
```

⇒ display_list_book_alphabitical_order

This function shows the list of books by alphabetic order.

```
546  void  Display_the_Book_list_in_alphabetical_order ( struct book_t book [MAX_BOOK] , int *Numbre_book )
```

⇒ find_book

This function asks the user to enter the book name which one to search, if the book not available in the list it shows the message <<book not found>>.

```
572  void find_a_book ( struct book_t book [MAX_BOOK] , int *Numbre_book )
```

⇒ borrow_book

This function suggests the available books to borrow

```
627  void to_borrow_a_book ( struct adherent_t *adhP, struct adherent_t adh[MAX_ADH], int *Numbre_Adherent, struct book_t *bookP, struct book_t
       book[MAX_BOOK], int *Numbre_book){
```

⇒ display_list_of_brrowers

This function display only the borrowed book

```
722  void display_the_list_of_borrowers(struct book_t book[MAX_BOOK] , int *Numbre_book , struct adherent_t adh[MAX_ADH] , int *Numbre_Adherent)
```

⇒ return_a_book

This function made to return the borrowed books.

```
744  void return_a_book( struct adherent_t *adhP, struct adherent_t adh[MAX_ADH], int *Numbre_Adherent, struct book_t *bookP, struct book_t
       book[MAX_BOOK], int *Numbre_book){
```

⇒ display_brrower_list

This function displays the borrowers list.

```
839  void Display_the_borrower_list ( struct adherent_t adh [MAX_ADH] , int *Numbre_Adherent )
```

Qualitative assessment of the work

- **interesting points:**

* **Pointers** in **C** are one of the strongest and sometimes dangerous features of **C**. there is some reasons why we find **pointers** so interesting & helpful.

a) **Pointers reduce the length and complexity of the program.**

b) **They increase execution speed.**

c) **A pointer enables us to access a variable that is defined outside the function.**

d) **Pointers are more efficient in handling the data tables.**

e) **The use of a pointer array of character strings results in saving of data storage space in memory.**

* **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case.

f) **Switch reduce the length and complexity of the program.**

g) **They increase execution speed.**

h) **They make the menu better.**

- **Difficulties:**

*even if the work together was really fun there where some difficulties that kept us moving slowly:

⇒ **Online communication:**

Because of the quarantine we didn't have the chance to met face to face, so we were satisfied with the online communication only (WhatsApp, skype...).

⇒ **Different compilers:**

Nouhaila was working on **DEVPLUS** compiler and i (**Yassen**) was working with **XCode**.

⇒ **Not going in the same direction:**

At some point of the project, we disagreed a lot about how to formulate some functions, which led to wasting of time and effort.

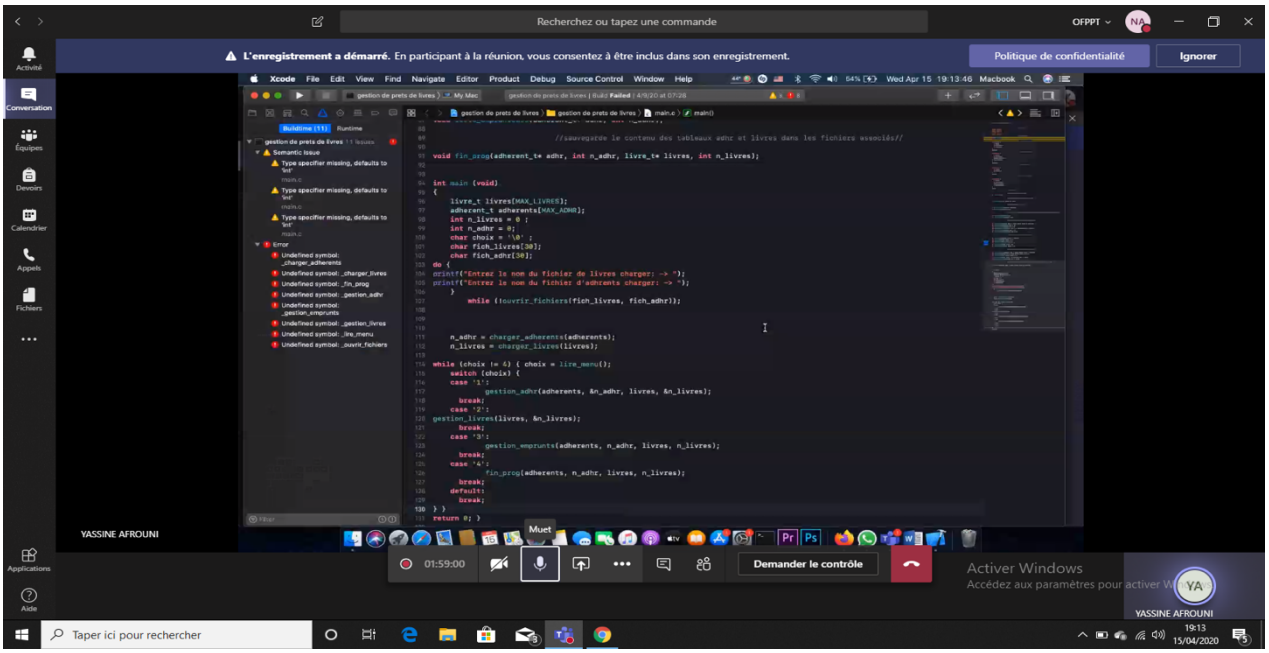
⇒ **Debugging**

It was the worst part, picture this scenario. After working for days to perfect a program, you go to take rest satisfied that it will work like it's supposed to. When you come back in the next day, your colleague gives you a long list of bugs to work through.

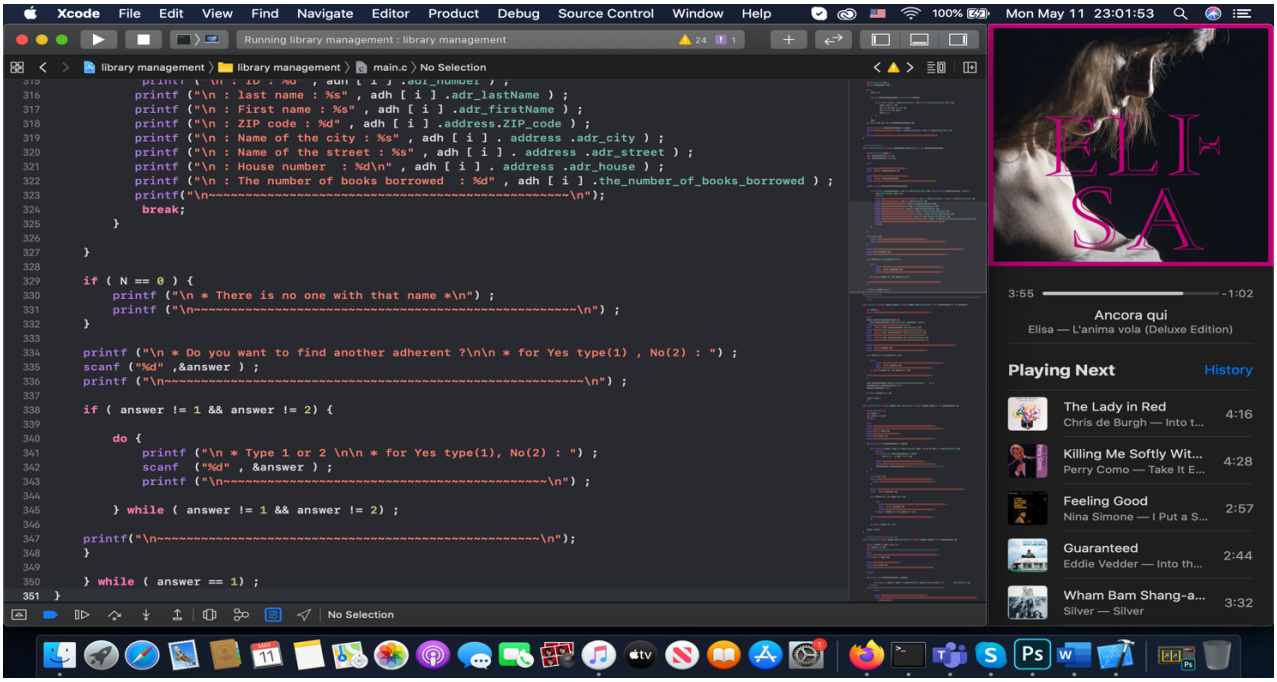
- **Screenshots:**

WhatsApp call:

Skype call:



Desktop screen:



Conclusion

Certainly, you remember some of my sarcastic comments about the simplicity of some exercises that involve manipulating numbers or characters ... I would like to apologize about that, by working on the project, I found out how these little details are helpful & important.

Like most of my colleagues at the TRI105, before going through this module, I had a very limited view about programming in general and didn't know what to expect from an internship.

SO

We want to thank you Mms IMANE, you've been so much more than a teacher for us. You've been our mentor, our support and our guide. Thank you for everything you have done for us!!