

# ENGR 101 - Introduction to Programming

## Mini Project 3

(Due on May 3, 2020)

In this mini project, you are going to develop a console-based Library Management System! This software will be used by students to borrow/return books. The primary goal of this mini project is to practice using various data structures (i.e., list, dictionary). Hence, please use every opportunity to employ a data structure in your code. At various points, you will be forced to use a particular data structure as explained in the “Implementation Notes” section at the end. The detailed explanations about how you are going to develop this application are provided below. (Note: example console output from the application is written in [Blue](#)).

### Text-based User Interface:

When your program first runs, it will ask for an ID and password to login. Based on the entered user ID, different menus will be displayed (Admin menu or Student menu). In your code, at least the following admin and students should already be defined:

- Admin: ID: ‘admin’ password: ‘123’
- Students: ID: “Ahmet” password: “1234” and ID: “Ayse” password: “4567”

You may define additional admins/students if you like (Please use the most proper data structure to store ID / password information. See implementation notes).

If the user enters an invalid id or password, the program should ask the user to try again and keep asking until valid credentials has been entered. The user successfully logs in when the provided credentials are valid. The following shows the interface for the initial login screen with a scenario where the user first enters incorrect ID/password, and then successfully logs in after the second trial.

\*\*\*\*[Welcome to Library Management System](#)\*\*\*\*  
[Please provide login information](#)

[Id: admin](#)

[Password: 4444](#)

[Invalid id or password please try again](#)

[Id: admin](#)

Password: 123

Successfully logged in!

Once the user enters valid credentials, your program should greet the user and provide admin menu if user is an admin or student menu if the user is a student. The menu displays available operations to the user and the user will be able to select an operation by entering the corresponding menu number. Each operation is discussed in detail below. If the user enters an invalid menu entry, the program should keep asking the user to provide a valid menu number.

Once an operation is done, the user should return back to the menu so that (s)he can perform another operation.

### Admin menu

The following menu will be displayed to the admin:

Welcome Admin! What do you want to do?

1-List books

2-Create a book

3-Clean a book

4-Search for a book

5-Change number of copies of book by id

6-Show students borrowed a book by id

7-List Users by id

8-Create User

9-Delete User

10-Exit

Your choice:

When **main menu item 1** is chosen, the list of available books with their corresponding id, name, and number of copies will be shown as follows.

Your choice: 1

\*\*\* List of Books \*\*\*

1. Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 2

2. Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies: 1

When **main menu item 2** is chosen, the user will be asked to write the name of the book, the author, the id and the number of copies for that book he/she would like to add(ex: Physics), then the program will ask for the id for the added book(ex: 003),then the program will ask about the author(s) for the added book. Multiple authors will be inputted as a comma separated way and input may include white spaces before or after each comma (ex: Einstein, Newton) Lastly it will ask the number of the copies for the added book (ex:4). The program asks the user if he/she is sure about adding this book, input should be either Y/N, otherwise the program should ask for a valid input. Your program should properly save the books name along with its id, author, number of copies, should print the saved info.

id for each book should always be unique. If the admin enters an existing id while creating a book, your program should give an error message and ask for a valid id again.

Welcome Admin! What do you want to do?

1-List books

2-Create a book

3-Clean a book

4-Search for a book

5-Change number of copies of book by id

6-Show students borrowed a book by id

7-List Users by id

8-Create User

9-Delete User

10-ExitYour choice:

Your choice: 2

What is the name of the book that you want to add?: Physics

What is the id that you want to give for Physics book?: 003

What is/are the author(s) of Physics book?: Einstein, Newton

How many copies you have for Physics book?: 4

Are you sure?[Y/N] Y

The following book has been added to your collection:

Book id: 003, Book Name: Physics, Book Authors: ['Einstein', 'Newton'], Number of Copies: 4

When **main menu item 3** is chosen, the user will first be presented with the list of books in the library. Then, the program asks the user which book to delete by its id and print the book that has been deleted. (Note: the program should keep asking for a valid id if id does not exist in the system.)

Search by id should utilize the existing dictionary so that the operation is efficient. (see the implementation notes.) Do not traverse the list of all books to search for an id. Also once a book is removed, you should remove all entries referring to the book from all relevant dictionaries.

Your choice:3

\*\*\* List of Books \*\*\*

1. Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 2
2. Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies: 1

What is the id of the book that you want to delete,(Enter 0 to go to main menu)?:001

The following book has been deleted:

Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 2

When **main menu item 4** is chosen, the admin will be able to search for a book based on its name or author name, and lists the corresponding book name, id, authors and number of copies; then program return to admin menu. (Note: if no book matches the query, the program should ask again for another query.)

Query by author should utilize the existing dictionary so that the operation is efficient. (see the implementation notes.) Do not traverse the list of all books to search for the author name.

Your choice:4

Enter book name or author name to search ,(Enter 0 to go to main menu)?:Chemistry

1. Book id: 002 , Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies: 1

When **main menu item 5** is chosen, the user will be able to choose a book by its id and update the number of copies for that book. **Note** that new value should be consistent with the number of people currently holding the book.

Your choice:5

\*\*\* List of Books \*\*\*

1. Book id : 001 ,Book Name: Biology ,Book Authors: ['Alice', 'Bob'] ,Number of Copies:2

2. Book id : 002 ,Book Name: Chemistry ,Book Authors: ['Alice'] ,Number of Copies:1

What is the id of the book for the change?(Enter 0 to go to main menu):001

Enter the new value for the no of copies:1

2 user(s) is/are holding the book, try a larger number!

Enter the new value for the no of copies:10

The following book has been updated:

Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 10

When **main menu item 6** is chosen, the user should be asked to choose an existing book id to display students that borrowed that book.

Your choice:6

What is the id of the book that you want to show, (Enter 0 to go to main menu):001

Students that borrowed the book:

-Ahmet

-Ayse

When **main menu item 7** is chosen, the current list of students should be shown as follows,and return to main menu

\*\*\* Current Users \*\*\*

1-Ahmet

2-Ayse

When **main menu item 8** is chosen, the user will be asked to create a name and password for a new student as following:

What is the name of the user that you want to create? Ali

What is the password for the new account? 3344

The following user has been added:

Name: Ali, Password: 3344

The name of each student should be unique.

When **main menu item 9** is chosen, current students should be displayed, then the user should be asked which user the program should delete. (Note: the program should keep asking for a valid input)

Current Users:

1-Ahmet

2-Ayse

Which user do you want to delete: Ahmet

Ahmet is deleted!

When **main menu item 10**, the program should go back to the Login page.

### Student menu

When a student logs in, the following menu will be displayed:

\*\*\*\*Welcome to Library Management System\*\*\*\*

Please provide login information

Id:Ahmet

Password:1234

Successfully logged in!

Welcome Ahmet what do you want to do ?

1-Search for a book

2-Add a book to my books list

3-Delete a book from my books list

4-Show my borrowed books

5-Exit

Your choice:

When **main menu item 1** is chosen, the user should be asked to search for a book by its name or author name, and lists the corresponding books. (Note: the program should keep asking for a valid input)

Your choice:1

Enter book name or author name to search ,(Enter 0 to go to main menu?):Chemistry

1. Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies: 1

Welcome Ahmet what do you want to do ?

1-Search for a book

- 2-Add a book to my books list
- 3-Delete a book from my books list
- 4-Show my borrowed books
- 5-Exit

Your choice:

When **main menu item 2** is selected, your program should print the available books, then asks the user which book he/she wants to add based on its id, then return to students menu

If the book is already taken by the user or there is no available copy, your program should tell so and ask the user to choose a different book.

Your choice:2

\*\*\* List of Books \*\*\*

1. Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Available Copies: 2
2. Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Available Copies: 1

What is the id of a book that you want to add (Enter 0 to go to main menu)?:002

The following book has been added to your profile:

Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Available Copies: 1

When **main menu item 3**, the program first will show books borrowed by the student and ask him/her which book to return.

Your choice:3

\*\*\* List of Books \*\*\*

1. Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies:2
2. Book id: 002, Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies:1

What is the id of a book that you want to return (Enter 0 to go to main menu)?: 001

The following book has been selected:

Book id: 001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 2

Are you sure that you want to return this book ? [Y/N]Y

The book has been returned.

When **main menu item 4** is selected, books currently held by the user should be displayed.  
(previously returned books should not be shown).

Your choice: 4

Your Books:

1. Book id:001, Book Name: Biology, Book Authors: ['Alice', 'Bob'], Number of Copies: 2
2. Book id:002, Book Name: Chemistry, Book Authors: ['Alice'], Number of Copies: 1

When **main menu item 5**, the program should go back to Login page.

## Implementation Notes

- Use a dictionary to store users where username is the key, and value is a password. Store admins similarly.
- You need to represent a book as a data structure. A list would be good choice. Note that the book, in our context, consists of an id, a name, a no\_of\_copies, a list of authors.
- A book can have multiple authors and one author can have multiple books.
- Use a dictionary to store books where book id is the key, and the value is the book (e.g., a list mentioned before). Whenever you need to access books by id, you need to use this dictionary for an efficient implementation.
- Use a dictionary that maps authors to books where author name is the key, and value is a list of books written by that author. Whenever you need to access books by author name, you need to use this dictionary for an efficient implementation.
- Use a dictionary to store which users borrowed which books where user is the key, and value is a list of books taken by that user. Whenever you need to access borrowed books by student, you need to use this dictionary for an efficient implementation.
- Use a dictionary to store books borrowed where book id is the key, and value is a list of user names that borrowed that particular book. Whenever you need to access students by id of the book they hold, you need to use this dictionary for an efficient implementation.



## Other Criteria

### Warnings

- **Do not** talk to your classmates on project topics when you are implementing your projects (This is serious). **Do not** show or email your code to others (This is even more serious). If you need help, talk to your TAs or the instructor, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve “**should**”, “**should not**”, “**do not**”, and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.

### How and when do I submit my project?:

- Projects may be done individually or as a small group of two students (doing it individually is recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file. Name your code file with your and your partner's first and last names (see below for naming).
  - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz\_baris\_ahmet\_caliskan.py (Do **not** use any Turkish characters in file name).
  - If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
  - Those who **do not** follow the above naming conventions **will get 5 pts off** of their grade.
- Submit it online on LMS by **5 pm on May 3, 2020**

### Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted.

## Grading Criteria?

Code Organization			Functionality			
Not using meaningful variable names (-5% cut off from your overall grade)	Not properly using functions, code repetitions (-5% cut off from your overall grade)	Insufficient commenting (-5% cut off from your overall grade)	Student functionality with all of its submenus (25 pts)	Admin functionality with all of its submenus (35 pts)	Proper use of data structures to implement the student and admin menu items (30 pts)	Others (10 pts)

### Have further questions?

Please contact your TAs if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not show up randomly (especially on the last day) in your TAs' or the instructor's office hours. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**

Good Luck!