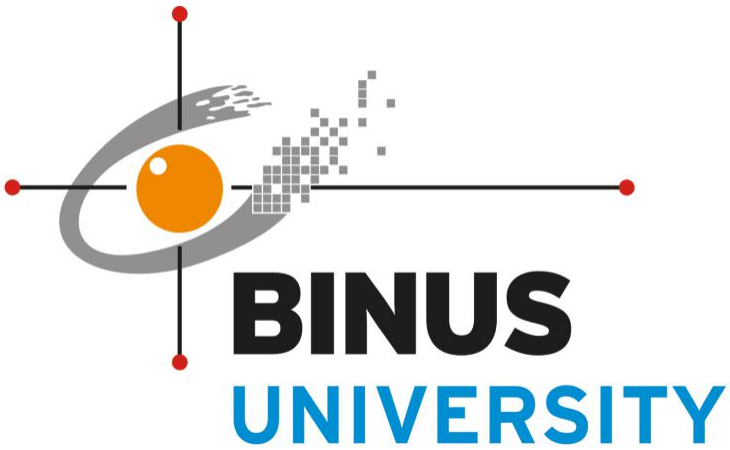
Final Java Project Report
(Library System Management)



By: Calvin Scorpiano Halim
Student ID : 2301915374

Project Specification

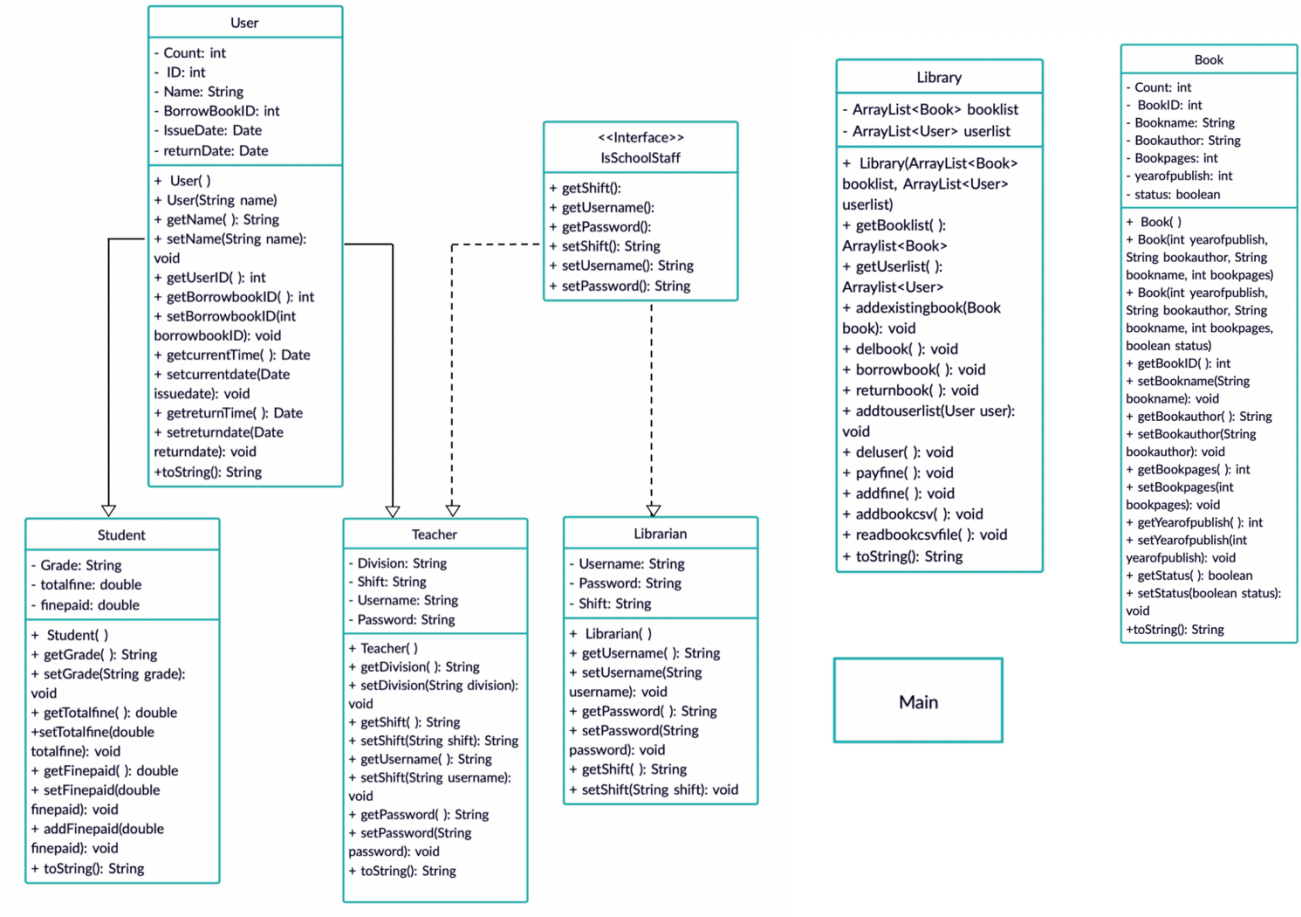
The last project of this semester requires the understudies to make their very own program dependent on the data and information that was gained all through the second term. The requirements for this project was set by the lecture, these prerequisites are set with the goal that understudies can show what they have realized and their comprehension of the subject.

In order to choose one of many possible project ideas, one must look into personal or daily problems that offers many possible ideas. The problem that I choose is to create a program where it will be easier to manage a library for schools where teacher and especially students can borrow and return books more efficient. Plus, I have been interested on how management works from behind the computer and thus this is the program that I chose to create.

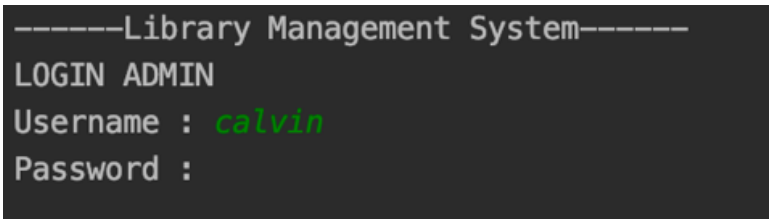
Solution Design

In order to fulfils the requirement based on the project specification , the library management system was intended to help students and teachers to borrow books from the library more convenient and with ease. The library will be able to distinguish between a student and teacher whereas student can be fined if they return the books past the due date while teachers can't. For the admin, they can add and delete the users (student and teacher), add or delete a book if the book is no longer available, and add fine based on the student who return the book past the due date. The book data itself will be saved in a CSV file, so you don't have to input a new book again and again every time you re-run the program.
I intended to put the user data in CSV file also, but since the teacher and the student inherits from the class and I haven't figured out how to make a program to read and distinguish the difference between teacher and students. So that is why I will be using CSV file for the book data only.

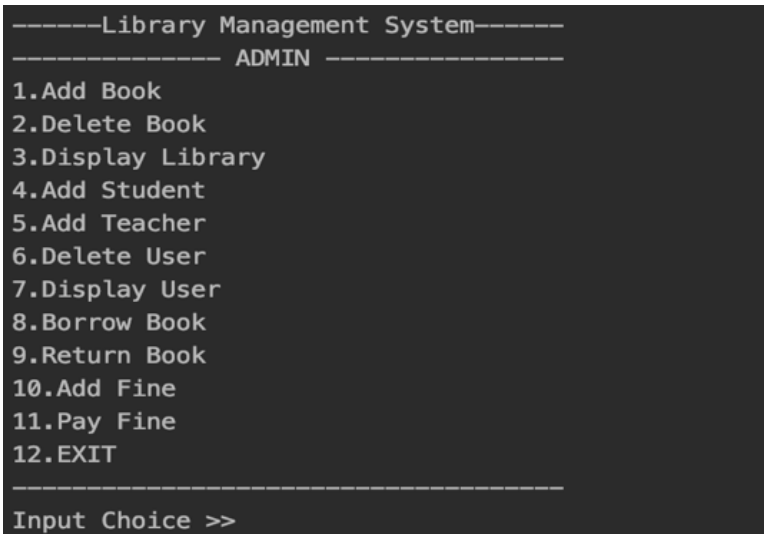
UML Diagram



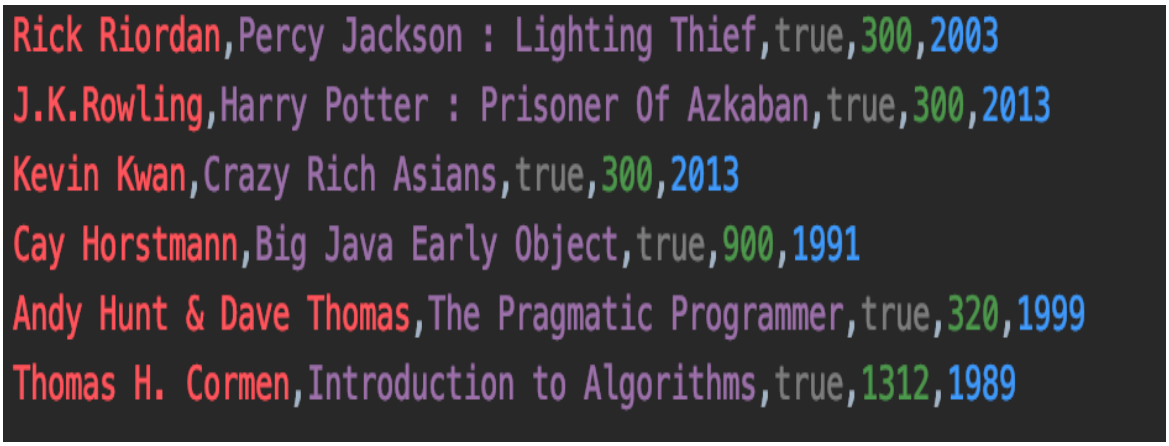
Working Program Images



When u first run the program, it will show u this login as an admin. You have to enter the username and the password. You can obtain the username and the password from the "Librarian.java" class. If it matches, you will be directed into the menu.



This is the Menu for the program. As you can see, there are 12 functions you can do as an admin. I will be showing you the screenshots to some of the functions.



This is the CSV file where it contains all the Book data.
Starting from the book's author, book's name, the status of the book (true means its available and false means it's not available or being borrowed), number of pages, and the year of publish.

```
Input Choice >> 3
[Book { ID= 1, Book= 'Percy Jackson : Lighting Thief', Author= 'Rick Riordan', Pages= 300 Pages, published= 2003, status= 'true'}
, Book { ID= 2, Book= 'Harry Potter : Prisoner Of Azkaban', Author= 'J.K.Rowling', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 3, Book= 'Crazy Rich Asians', Author= 'Kevin Kwan', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 5, Book= 'Big Java Early Objects', Author= 'Cay Horstmann', Pages= 900 Pages, published= 1991, status= 'true'}
, Book { ID= 7, Book= 'The Pragmatic Programmer', Author= 'Andy Hunt & Dave Thomas', Pages= 320 Pages, published= 1999, status= 'true'}
, Book { ID= 9, Book= 'Introductions To Algorithm', Author= 'Thomas H. Cormen', Pages= 1312 Pages, published= 1989, status= 'true'}
]
```

The photo above is a function to display all the books, now the ID here are generated automatically. So every time the admin adds a new book, the ID will be add by one. You can see that the ID number 4 and 6 is missing, this is because I deleted the book with the ID = 4 and 6. As mentioned above, the book status means whether the book is available to be borrow or not.

```
Input Choice >> 8
[Book { ID= 1, Book= 'Percy Jackson : Lighting Thief', Author= 'Rick Riordan', Pages= 300 Pages, published= 2003, status= 'true'}
, Book { ID= 2, Book= 'Harry Potter : Prisoner Of Azkaban', Author= 'J.K.Rowling', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 3, Book= 'Crazy Rich Asians', Author= 'Kevin Kwan', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 5, Book= 'Big Java Early Objects', Author= 'Cay Horstmann', Pages= 900 Pages, published= 1991, status= 'true'}
, Book { ID= 7, Book= 'The Pragmatic Programmer', Author= 'Andy Hunt & Dave Thomas', Pages= 320 Pages, published= 1999, status= 'true'}
, Book { ID= 9, Book= 'Introductions To Algorithm', Author= 'Thomas H. Cormen', Pages= 1312 Pages, published= 1989, status= 'true'}
]
Enter book ID to borrow :
5
[Student { User { ID= '1, Name= Calvin , Borrow Book ID= 0, Issued Date= null,Due Date= null, Grade= 12, total fine= $0.0, fine paid= $0.0}
]
Enter student ID to borrow :
1
```

The photo above is a function the “Borrow Book” function. It will ask the admin to enter the book ID that wants to be borrowed and the ID of the student and teacher that wants to borrow it (there is only 1 student right now because I only add one user). After the book has been borrow and the function is done, the book’s status will be changed to “false”, thus the book cant be borrow anymore.

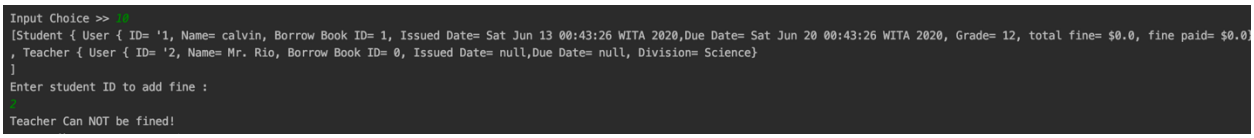
```
Input Choice >> 3
[Book { ID= 1, Book= 'Percy Jackson : Lighting Thief', Author= 'Rick Riordan', Pages= 300 Pages, published= 2003, status= 'true'}
, Book { ID= 2, Book= 'Harry Potter : Prisoner Of Azkaban', Author= 'J.K.Rowling', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 3, Book= 'Crazy Rich Asians', Author= 'Kevin Kwan', Pages= 300 Pages, published= 2013, status= 'true'}
, Book { ID= 5, Book= 'Big Java Early Objects', Author= 'Cay Horstmann', Pages= 900 Pages, published= 1991, status= 'true'}
, Book { ID= 7, Book= 'The Pragmatic Programmer', Author= 'Andy Hunt & Dave Thomas', Pages= 320 Pages, published= 1999, status= 'false'}
, Book { ID= 9, Book= 'Introductions To Algorithm', Author= 'Thomas H. Cormen', Pages= 1312 Pages, published= 1989, status= 'true'}
]
```

After you borrow the book and you choose the “Display Book” function, you can see that one of the book’s status is “false”, that is because its been borrowed.

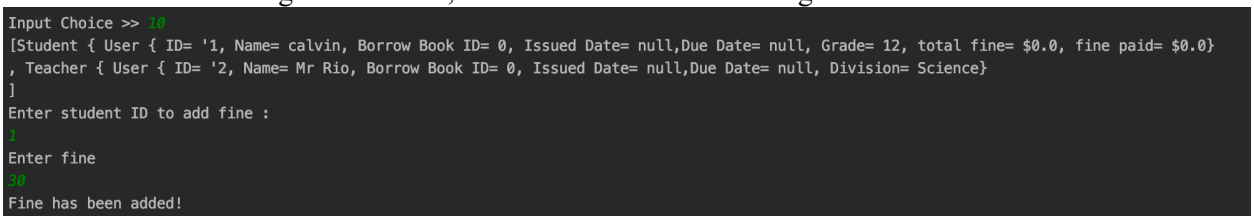
```
Input Choice >> 7
[Student { User { ID= '1, Name= calvin, Borrow Book ID= 1, Issued Date= Sat Jun 13 00:43:26 WITA 2020,Due Date= Sat Jun 20 00:43:26 WITA 2020, Grade= 12, total fine= $0.0, fine paid= $0.0}
, Teacher { User { ID= '2, Name= Mr. Rio, Borrow Book ID= 0, Issued Date= null,Due Date= null, Division= Science}
]
```

This is the “Display User” function. The photo will show you the difference between the student and teacher value and the difference between the user who borrow a book and one who doesn’t. I use the Java.util.Date and java.util.Calendar to work on the issued date and the return date, so the user can borrow a week only. If they want to extends, they have to reapply again. If student returns the book past the due date, the student will be fined. The teacher issue date and return date is null is because the teacher hasn’t

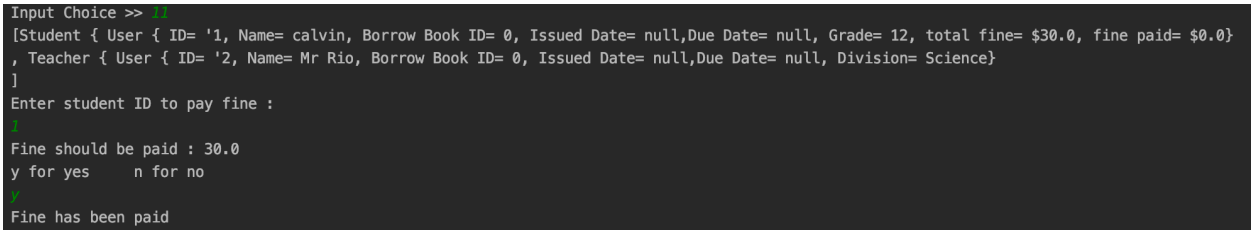
borrow a book yet. The teacher also doesn't have the total fine and fine paid, that is because as I mentioned earlier in the solution design that teacher can't be fined. The student and teacher also have a difference where student have grade and teacher have division.



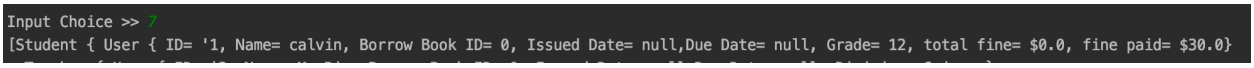
This is the "Add fine" function. On this picture, the program asks for the "Student" ID. If the admin enters the ID that belongs to a teacher, it will shows the error message.



If you enter the ID that belongs to a student, the function will proceed. And it will ask to enter the amount of fine that wants to be added. If it's successful, the total fine of the student will be added.



For the function "Pay fine", it only allows you to pay the whole fine meaning you have to pay fully. Same as previous, if you enter a teacher's ID it will produce an error message. But, if you enter student's ID, it will ask for a y or n input to pay the whole fine.



After you pay the fine, the total fine will be turned to zero, and the amount of fine paid will be increased by the amount of the amount of fine paid previously.

Conclusion

The Library Management System works without bugs or error. But, it still far from perfect. Doing this project made me understand more about OOP in Java and I am able to use the Date function and so does the Calendar. Although is not much, but for me it is already quite an achievement. After testing all the functions, they all works without difficulties. Hopefully for next time, I will be able to put more functions, clean my code, and use better design with the implementation of database and GUI.

Evaluation

The program works, but it still lacks in many aspects. In the future, I will look into more about GUI and Databases so I can implement it in the program to make it easier for user to use it and hopefully I can develop it more into an actual downloadable program.