Report of book administration system

Yiyang Zuo, Zhenghan Wang

Cotent

- 1.Introduction
- 2.Operation
- 3.UML
- 4.Conclusion
- 5.Participation

Introduction

- For administrators

```
public class Admin extends User{
    1 usage
    public Admin(String name){
        super(name);
        this.operations=new operation[5];
        operations[0]=new Exit();
        operations[1]=new Find();
        operations[2]=new Add();
        operations[3]=new Delete();
        operations[4]=new Show();
    }
    1 usage
    public int menu(){
        System.out.println("*****************************");
        System.out.println("Welcome to the Library Administration System.");
        System.out.println("1: Find the book");
        System.out.println("2: Add the book");
        System.out.println("3: Delete the book");
        System.out.println("4: Show the book");
        System.out.println("0: Exit the system");
        System.out.println("*****************************");
```

Introduction

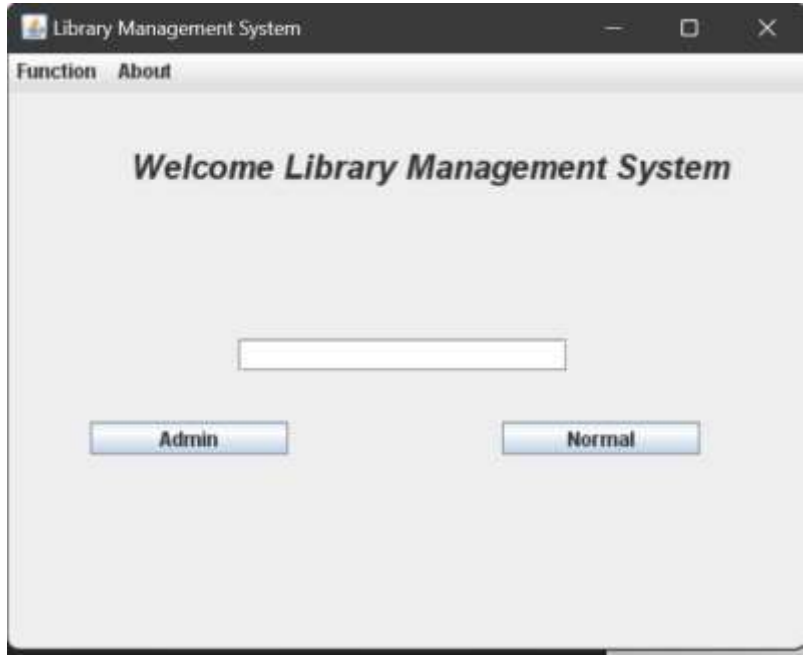
- For normal users

```
public class Normal extends User{
    1 usage
    public Normal(String name){
        super(name);
        this.operations=new operation[4];
        operations[0]=new Exit();
        operations[1]=new Find();
        operations[2]=new Borrow();
        operations[3]=new Return();
    }
    1 usage
    public int menu(){
        System.out.println("*****************************");
        System.out.println("Welcome to the Library Administration System.");
        System.out.println("1: Find the book");
        System.out.println("2: Borrow the book");
        System.out.println("3: Return the book");
        System.out.println("0: Exit the system");
        System.out.println("*****************************");
        System.out.println("Enter your option.");
        Scanner input=new Scanner(System.in);
```

Introduction

- Windows

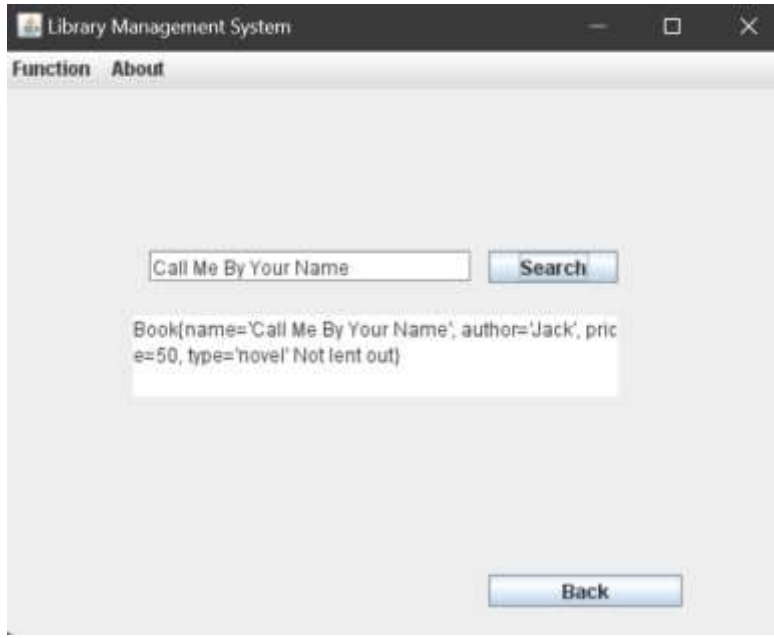
```
public void run() { frame1(); }

3 usages
public void frame1() {//主页面
    window();
    f1.setLocation(x, y);
    f1.setSize( width: 500,  height: 400);
    f1.setLayout(null);
    function.add(search);
    about.add(infor);
    jMenuBar.add(function);
    jMenuBar.add(about);
    f1.setJMenuBar(jMenuBar);
    button_admin_start();
    button_norm_start();
    Text1();
    Label1();
    f1.setVisible(true);
}
```



Operation

- Find

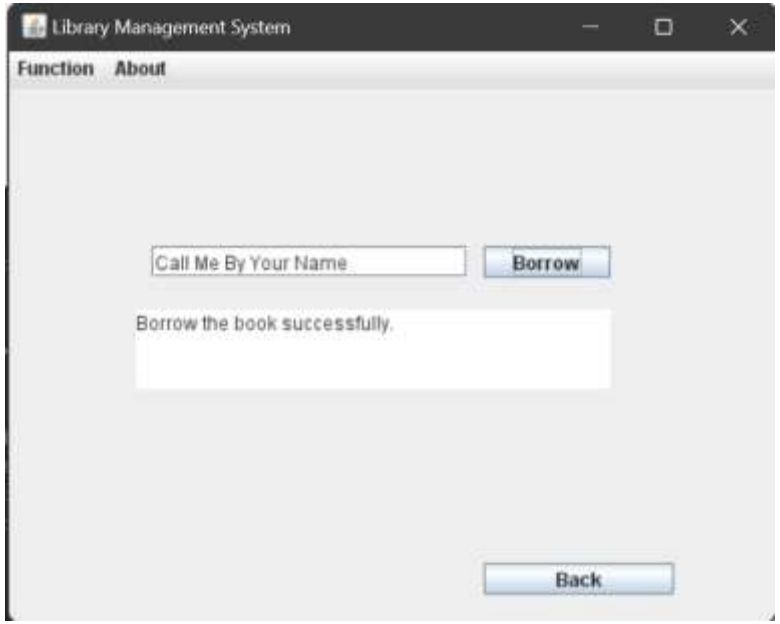
```
public void work(BookList bookList){
    System.out.println("Find the book.");
    System.out.println("Please enter the name of the book you want to find: ");
    String name =input.nextLine();
    int book_stored= bookList.getBook_stored();
    for(int i=0;i<book_stored;i++){
        Book book=bookList.getBooks(i);
        if(book.getName().equals(name)){
            System.out.println("Find this book successfully");
            System.out.println(book);
            return;
        }
    }
    System.out.println("Can't find this book.");
}
public String work(BookList bookList,String name){//窗口用查找
    int book_stored= bookList.getBook_stored();
    for(int i=0;i<book_stored;i++){
        Book book=bookList.getBooks(i);
        if(book.getName().equals(name))
        {
```



Operation

- Borrow

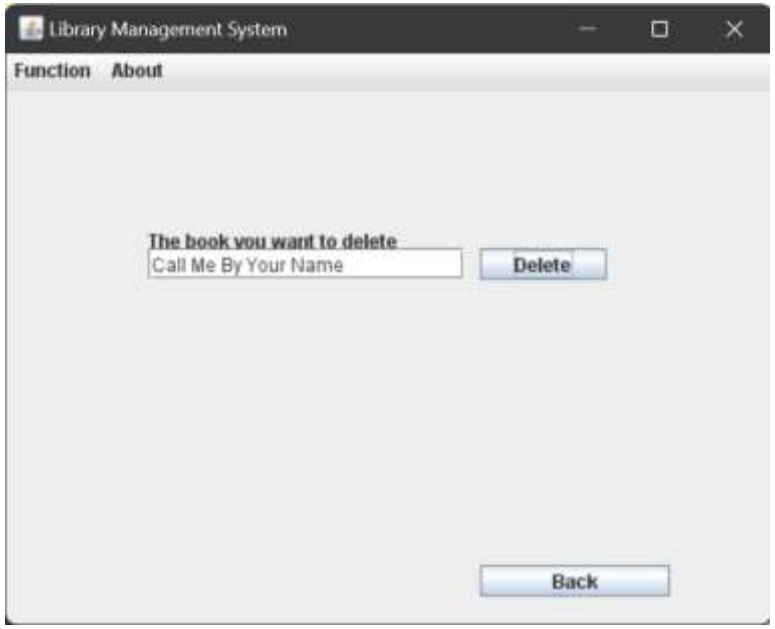
```java
public class Borrow implements operation {
    public void work(BookList bookList){
        System.out.println("Borrow the book.");
        System.out.println("Please enter the name of book you want to borrow: ");
        String name=input.nextLine();
        int book_stored= bookList.getBook_stored();
        int j=0;
        for(int i=0;i<book_stored;i++){
            Book book=bookList.getBooks(i);
            if(book.getName().equals(name) && !book.isborrowed()){
                System.out.println("Borrow the book successfully.");
                book.setIsborrowed(true);
                j++;
            }

        }
        if(j !=1){
            System.out.println("Can't find book.");
        }
    }
```



Operation

- Delete

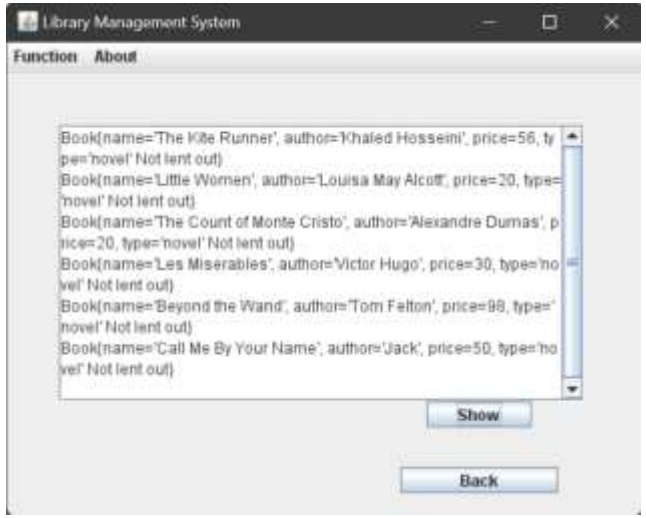
```
public class Delete implements operation{
    public void work(BookList bookList){
        System.out.println("Delete the book.");
        System.out.println("Please enter the name of the book you want to delete: ");
        String name=input.nextLine();
        int book_stored= bookList.getBook_stored();
        int index=-1;
        for(int i=0;i<book_stored;i++){
            Book book=bookList.getBooks(i);
            if(book.getName().equals(name)){
                index=i;
                break;
            }
        }
        for(int i=index;i<book_stored-1;i++){
            Book book=bookList.getBooks( position: i+1);
            bookList.setBooks(i,book);
        }
        bookList.setBook_stored(book_stored-1);
        bookList.setBooks( position: book_stored-1, book: null);
    }
```



Operation

- Show

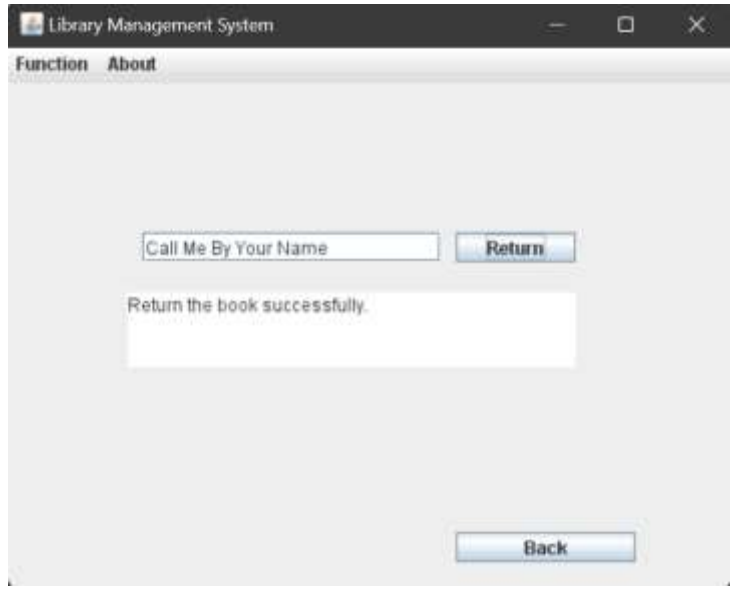
```java
public class Show implements operation {//显示当前图书库中的所有书籍信息

    public void work(BookList bookList) {
        int book_stored = bookList.getBook_stored();
        for (int i = 0; i < book_stored; i++) {
            Book book = bookList.getBooks(i);
            System.out.println(book);
        }
    }

    public String work(BookList bookList, Boolean a) {
        String information = "";
        int book_stored = bookList.getBook_stored();
        for (int i = 0; i < book_stored; i++) {
            Book book = bookList.getBooks(i);
            information += book;
            information += "\n";
        }
        return information;
    }
```



Operation

- Return

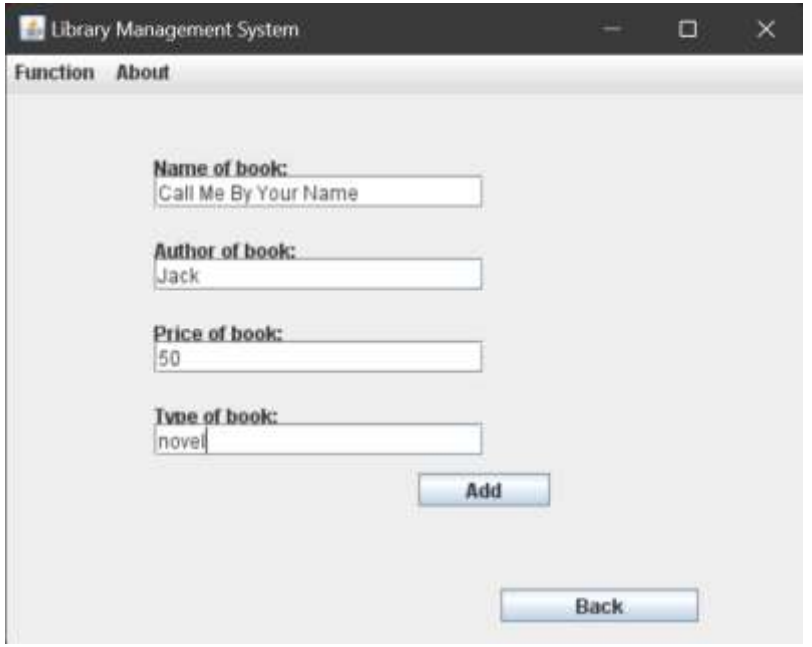
```
ublic class Return implements operation{
    public void work(BookList bookList){
        System.out.println("Return the book.");
        System.out.println("Please enter the name of the book you want to return: ");
        String name=input.nextLine();
        int book_stored= bookList.getBook_stored();
        for(int i=0;i<book_stored;i++){
            Book book=bookList.getBooks(i);
            if(book.getName().equals(name) && book.isborrowed()) {
                book.setIsborrowed(false);
                System.out.println("Return the book successfully.");
                return;
            }
        }
    }
    public String work(BookList bookList,String name){
        int book_stored= bookList.getBook_stored();
        for(int i=0;i<book_stored;i++){
            Book book=bookList.getBooks(i);
            if(book.getName().equals(name) && book.isborrowed()) {
```



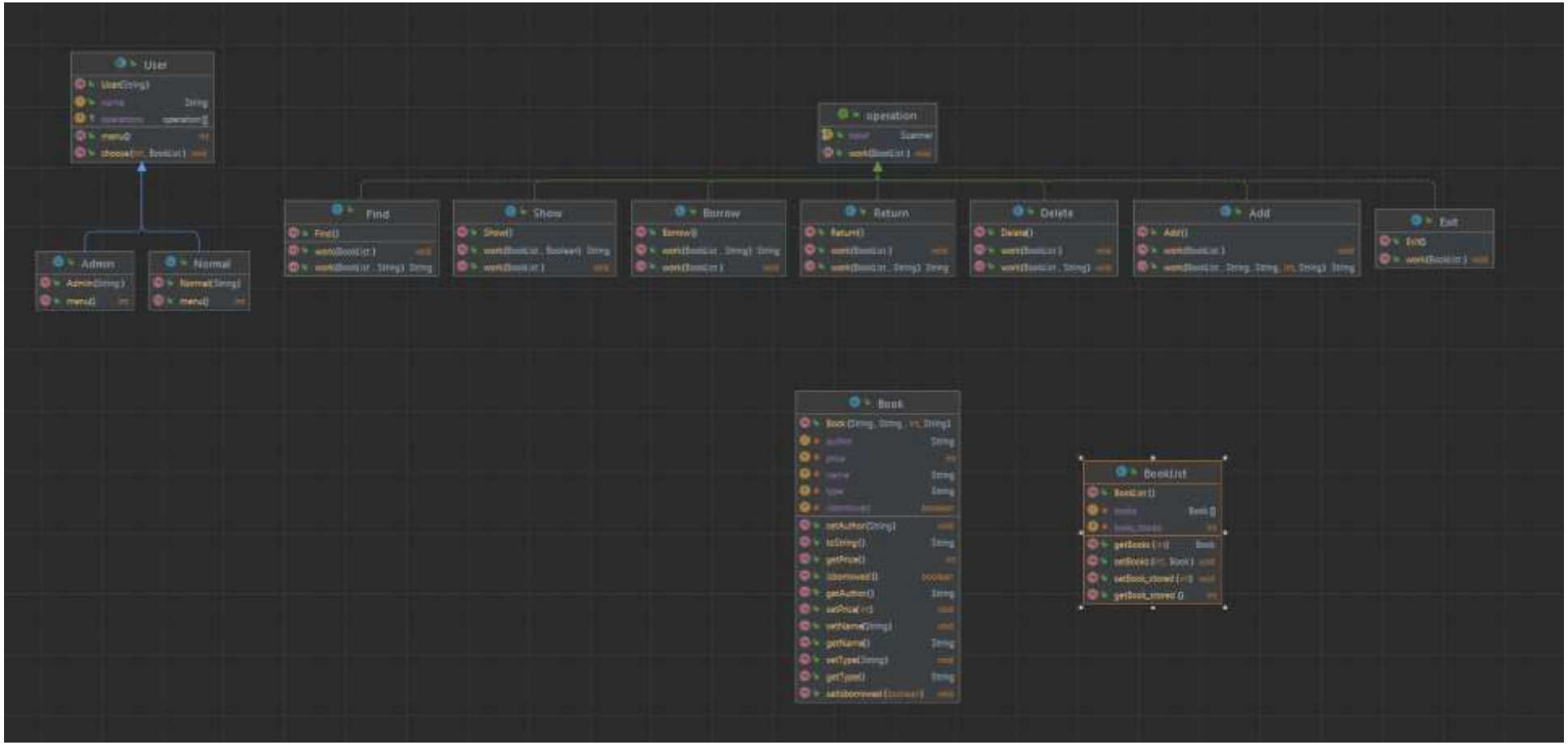
Operation

- Add

```
ublic class Add implements operation {
    public void work(BookList bookList){
        System.out.println("Add a new book.");
        System.out.println("Please enter the name of the book you added: ");
        String name=input.nextLine();
        System.out.println("Please enter the author of the book.");
        String author=input.nextLine();
        System.out.println("Please enter the price of the book.");
        int price=input.nextInt();
        System.out.println("Please enter the type of the book.");
        String type=input.nextLine();
        Book book = new Book(name, author, price, type);//信息录入完成以后创建一本书
        int book_stored=bookList.getBook_stored();
        bookList.setBooks(book_stored, book);
        for(int i=0;i<book_stored;i++){
            Book tmp=bookList.getBooks(i);
            if(book.getName().equals(tmp)){
                System.out.println("The book has existed.");
                return;
            }
        }
```



UML


User
operation
Find
Show
Borrow
Return
Delete
Add
Exit
Admin
Normal
Book
BookList

Conclusion

- The code basically implements the goal of this library management system. This book administration system is a light software and still exists some shortages. For example, the system cannot deal with the situation where the amount of books is enormous.

Participation

- Yiyang Zuo:
- code: Book, Booklist, Add, Borrow, Delete, Exit, Admin, Normal classes & operation interface
- Completing project report & UML Diagram

- Zhenghan Wang
- code: Frame, Find, Return, Show, User classes
- Using Frame class to establish a frame beautifing the system interface

Thank you