

# Library Management System

---

## REPORT

Submitted by:

Tathagat Banerjee -17BCE7100

Gaurav Gupta – 17BCE7129

Priyanka Nair – 17BCE7139

## Index

SL NO.	TOPICS	PAGE NO.
1	Analysis	2
2	ER diagram	3
3	ER to Relational mapping	4
4	Functional Dependency	5
5	Overview	6-25
5.i	Screenshots	6
5.ii	Sample Codes	18

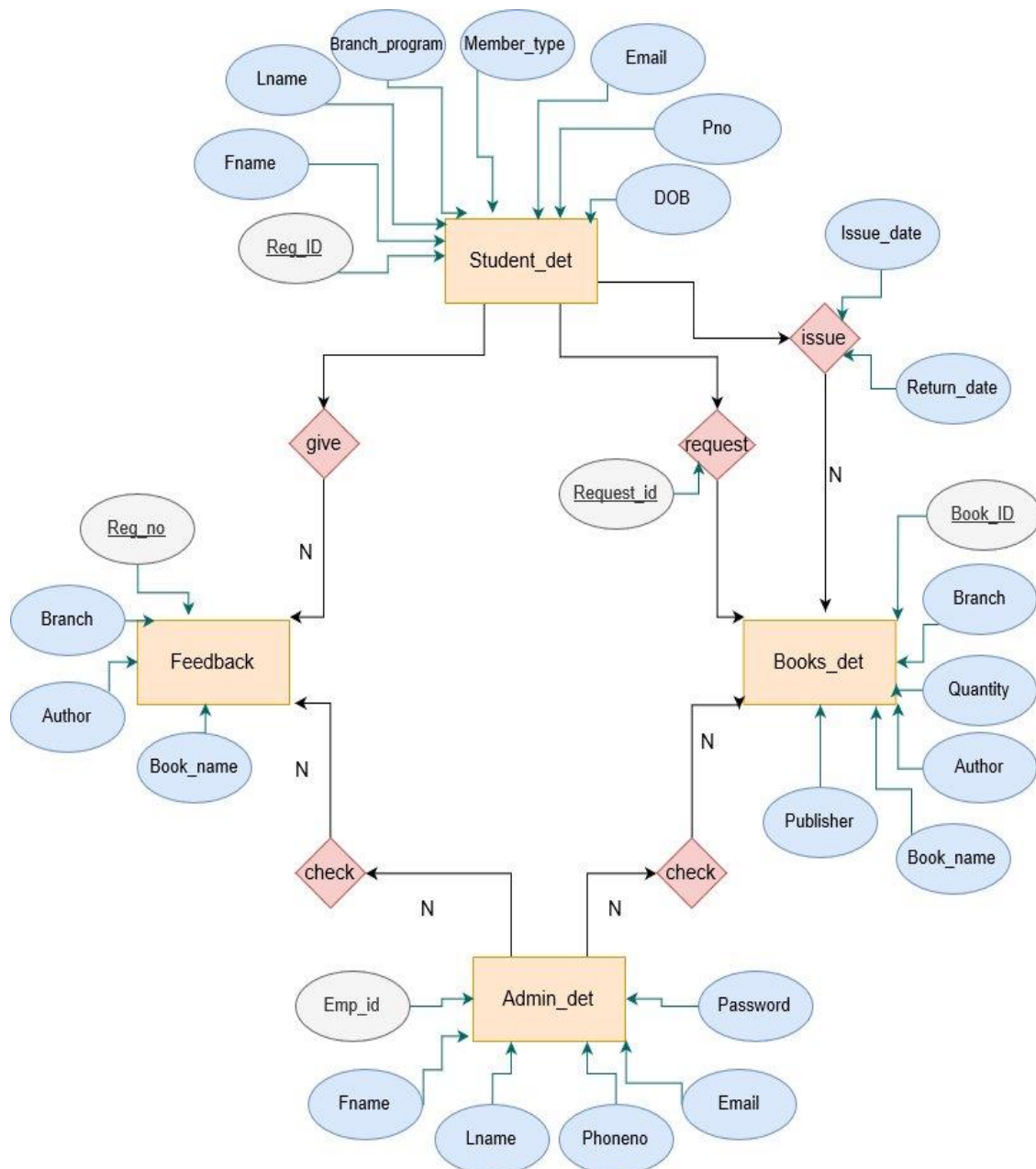
## **Analysis:**

- The followings are the entities in Library Management System:
  1. Books\_det
  2. Student\_det
  3. Issue\_Book\_det
  4. Admin\_det
  5. feedback
- The followings are the attributes of the entities:
  1. Books\_det (Book\_ID, Book\_name, Author, Branch, Quantity, Publisher)
  2. Student\_det (Reg\_ID, Fname, Lname, Branch\_program, DOB, Member\_type, Email, Pno.)
  3. Issue\_Book\_det (Reg\_no, Book\_ID, Issue\_date, Return\_date)
  4. Admin\_det (Reg\_no, Fname, Lname, Password, Email, Phoneno)
  5. Feedback (Reg\_no, Book\_name, Author, Branch)
- In library, the books have book id, branch the book belongs, number of books, name of book, author and publisher.
- Each student has their unique registration number, can issue at most 2 books at a time. Student's first name, last name, branch, type of member, email, contact number and d.o.b stored in student's details.
- Student can issue book which generate issue date and return date of that book.

If stock of book is over, student can even send a request for the required book by creating a request id for the book.

- Student can give feedback about books by mentioning book name, author and branch.
- Admin checks feedback and book details to maintain library management system.

## ER – Diagram:



## ER to Relational Mapping:

### Books\_det

Book_Id	Book_name	Author	Branch	Quantity	Publisher
---------	-----------	--------	--------	----------	-----------

### Student\_det

Reg_Id	Fname	Lname	Branch	DOB	member_type	Email	P_no.
--------	-------	-------	--------	-----	-------------	-------	-------

### Issue\_Book\_det

Reg_Id	Book_Id	Issue_date	Return_Date	Issue_date
--------	---------	------------	-------------	------------

### Admin\_det

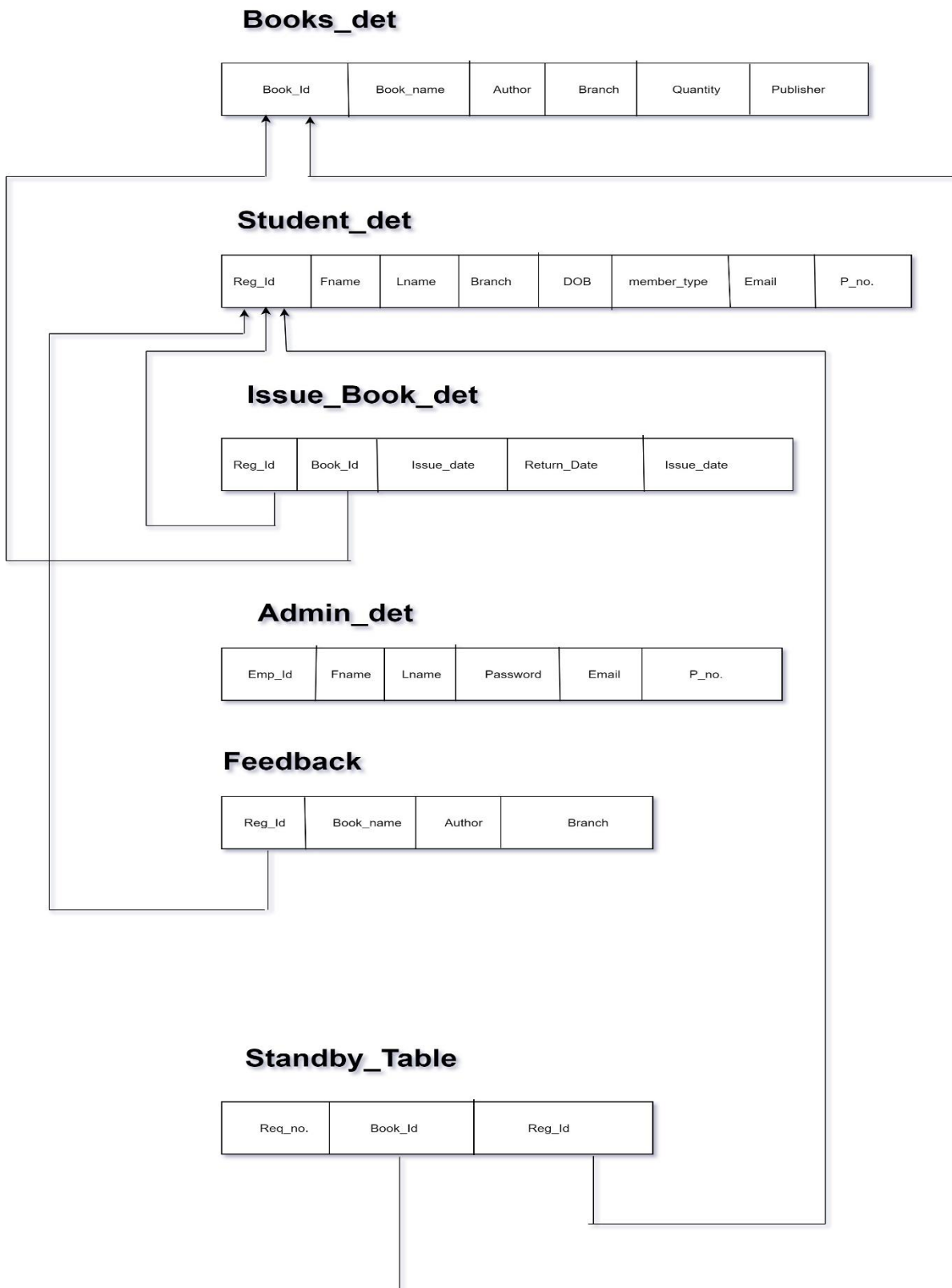
Emp_Id	Fname	Lname	Password	Email	P_no.
--------	-------	-------	----------	-------	-------

### Feedback

Reg_Id	Book_name	Author	Branch
--------	-----------	--------	--------

### Standby\_Table

Req_no.	Book_Id	Reg_Id
---------	---------	--------



**Functional dependency:**

Book\_id  $\rightarrow$  book()

Reg\_id  $\rightarrow$  Student()

Book\_id, Reg\_id  $\rightarrow$  Issue()

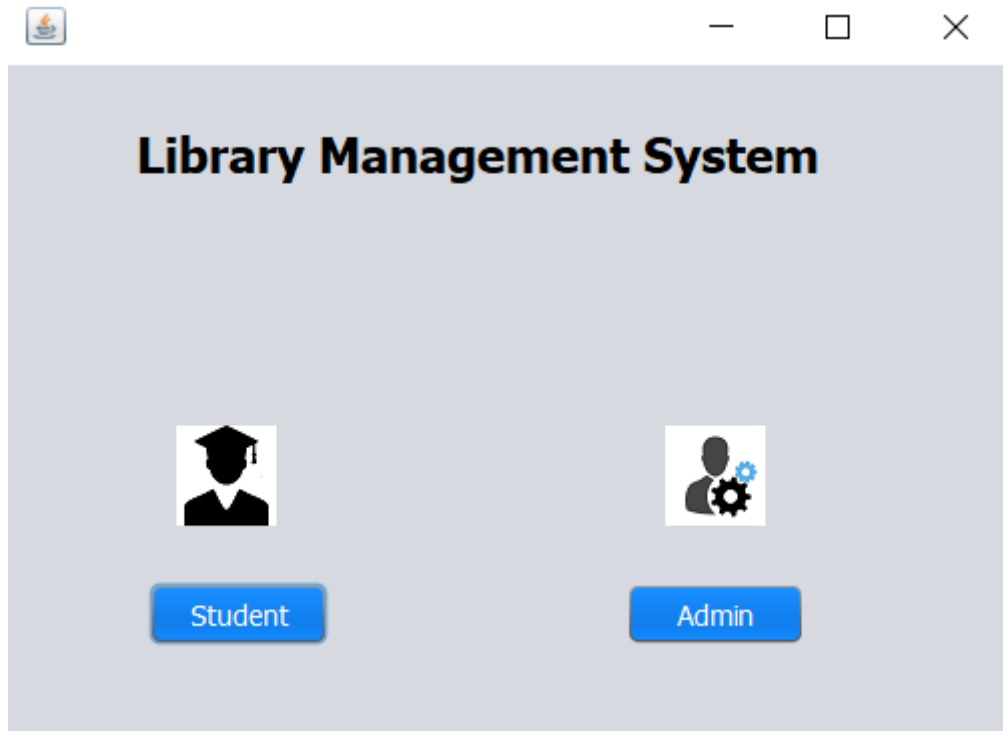
Book\_id, Reg\_id  $\rightarrow$  Standby

Reg\_id  $\rightarrow$  feedback()

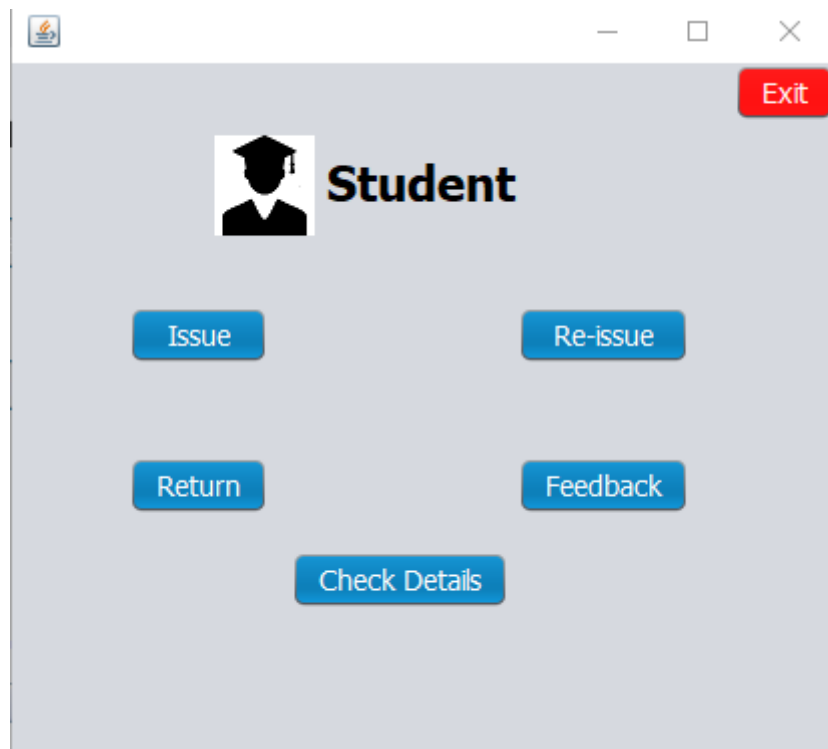
Emp\_id  $\rightarrow$  Admin()

Admin has authorized permission to books and feedback.

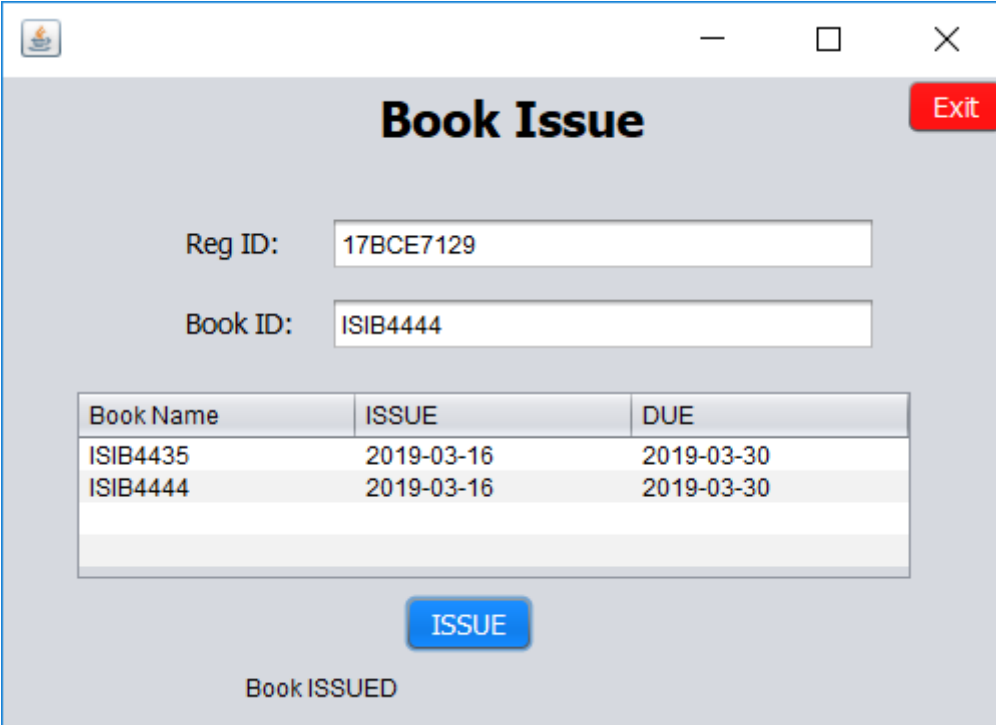
## Screenshots:



This is the **Main Page** of Library Management System which contains two buttons for Student and Admin.



Student button opens a page for Issuing, re issuing, returning of books. Student can check their details and even give feedback for books.



**Book Issue** Exit

Reg ID:

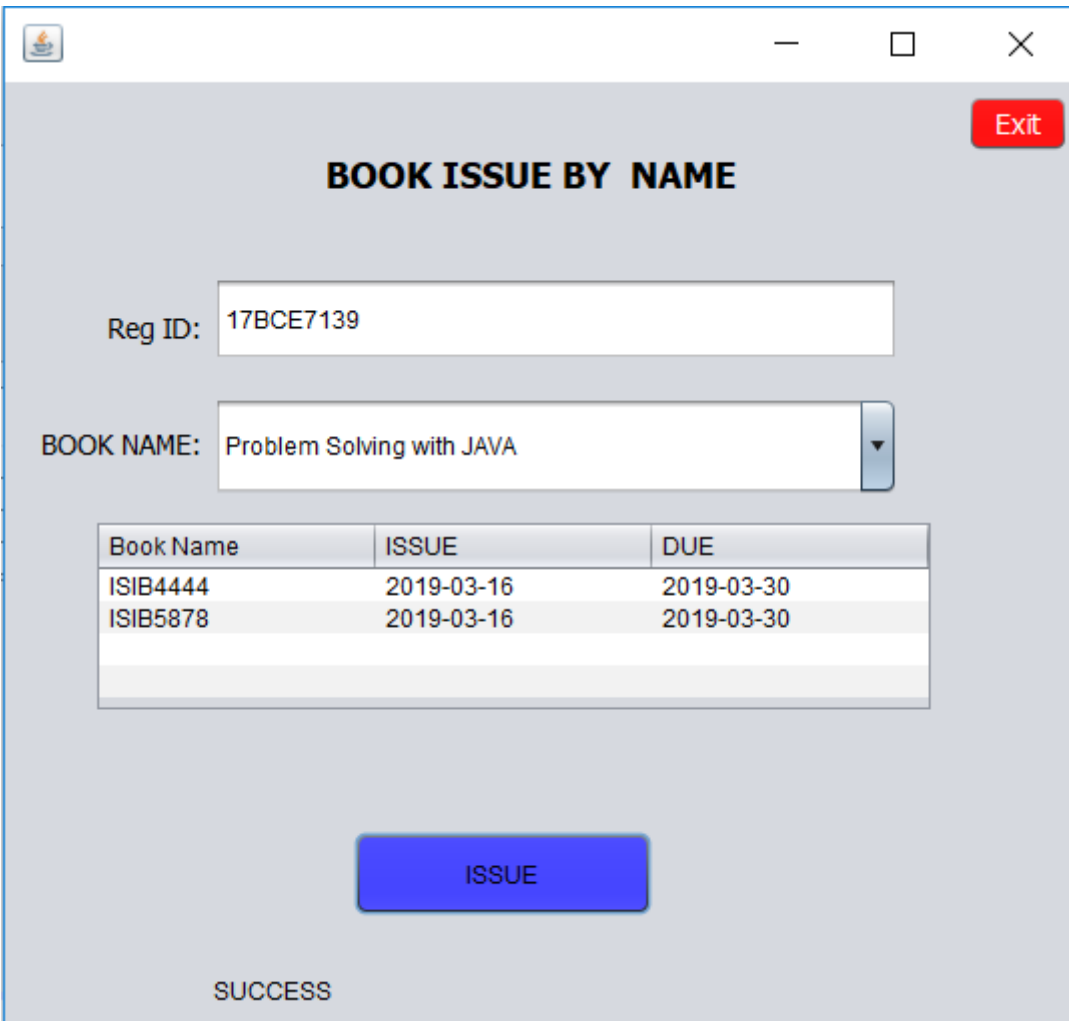
Book ID:

Book Name	ISSUE	DUE
ISIB4435	2019-03-16	2019-03-30
ISIB4444	2019-03-16	2019-03-30

ISSUE

Book ISSUED

Student can issue NOT more than 2 books at a time.



**BOOK ISSUE BY NAME** Exit

Reg ID:

BOOK NAME:

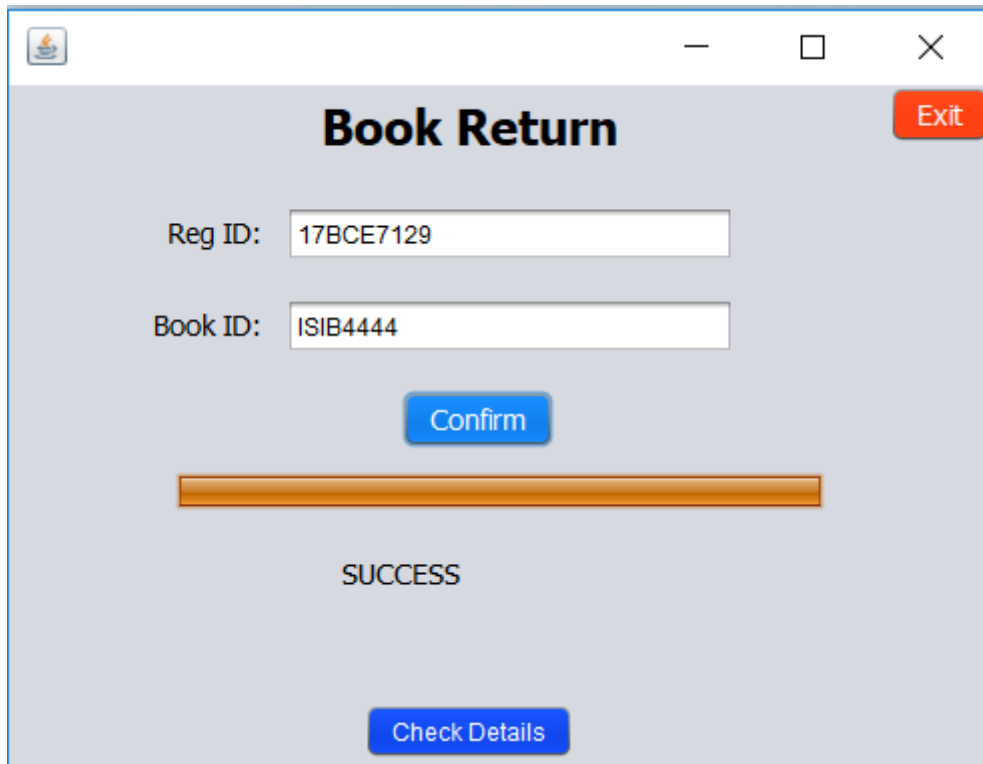
Book Name	ISSUE	DUE
ISIB4444	2019-03-16	2019-03-30
ISIB5878	2019-03-16	2019-03-30

ISSUE

SUCCESS

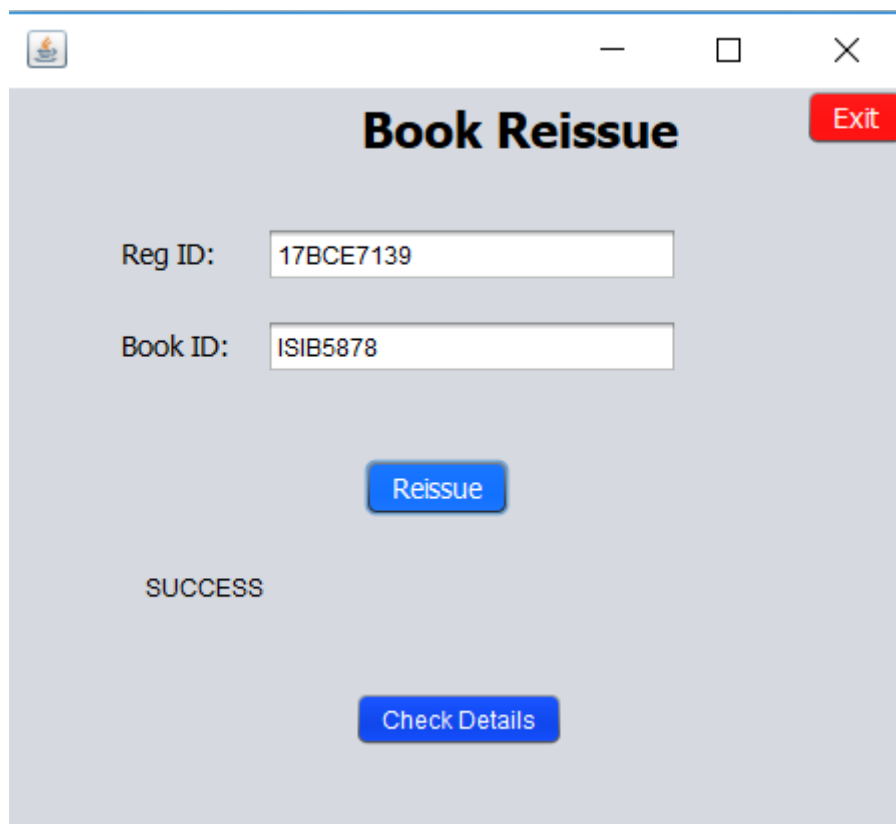
Student can even search book name (using autocomplete dropdownlist in jcombobox) and issue books.





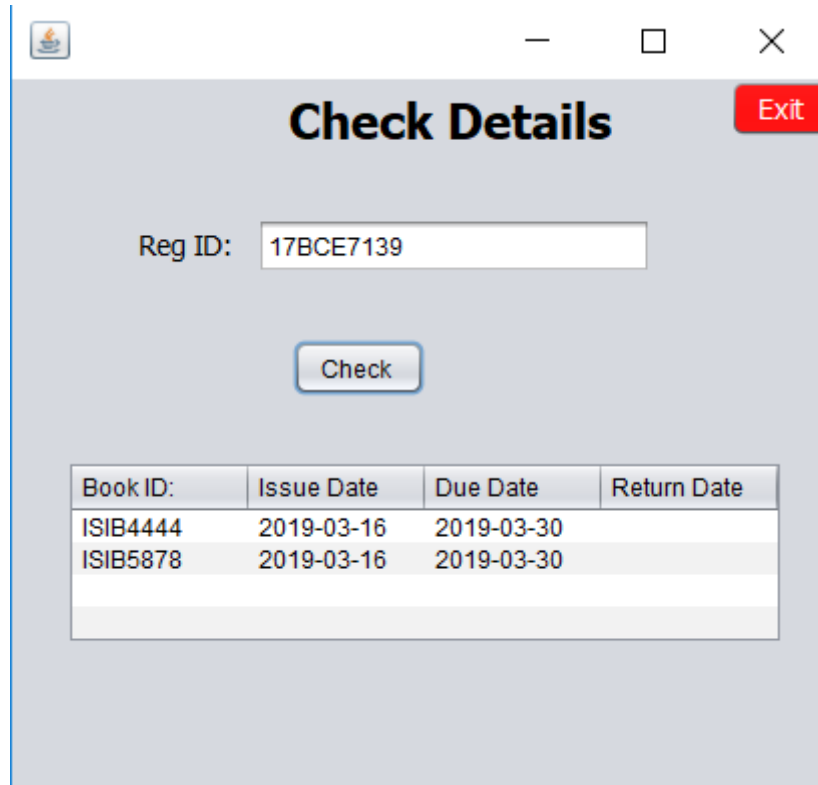
The screenshot shows a window titled "Book Return". It has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The window content has a light gray background. At the top right, there is a red "Exit" button. Below the title, there are two input fields: "Reg ID:" with the value "17BCE7129" and "Book ID:" with the value "ISIB4444". Below these fields is a blue "Confirm" button. Underneath the button is a thick orange progress bar. Below the progress bar, the word "SUCCESS" is displayed in a bold, black, sans-serif font. At the bottom of the window, there is a blue "Check Details" button.

Student's current book ID is matched with the string provided in Book ID and then it is deducted from the issued book of that student's database.



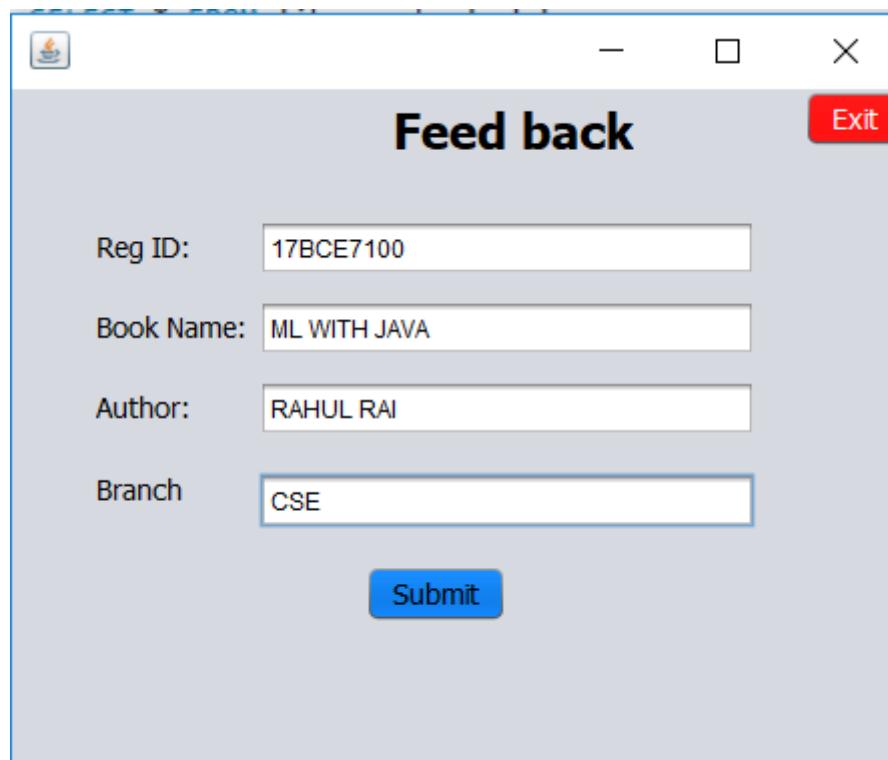
The screenshot shows a window titled "Book Reissue". It has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The window content has a light gray background. At the top right, there is a red "Exit" button. Below the title, there are two input fields: "Reg ID:" with the value "17BCE7139" and "Book ID:" with the value "ISIB5878". Below these fields is a blue "Reissue" button. Underneath the button, the word "SUCCESS" is displayed in a bold, black, sans-serif font. At the bottom of the window, there is a blue "Check Details" button.

If Student can reissue book if he/she has already issued that book earlier.

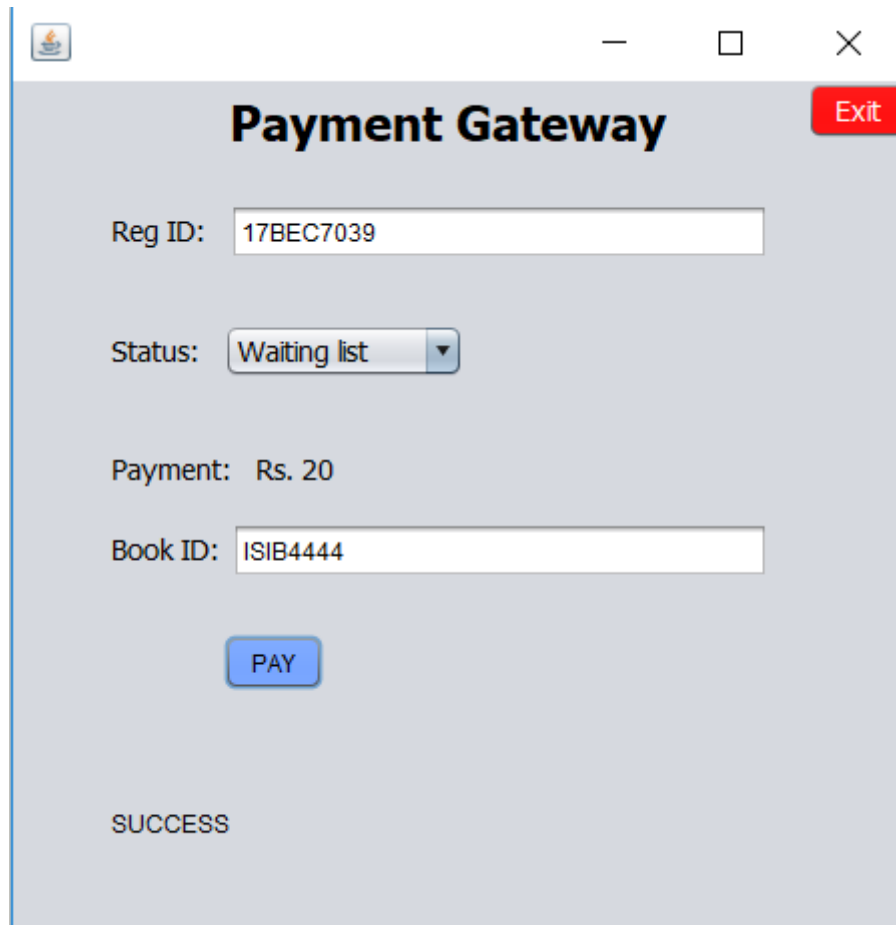


Book ID:	Issue Date	Due Date	Return Date
ISIB4444	2019-03-16	2019-03-30	
ISIB5878	2019-03-16	2019-03-30	

Student can know their issued book ID, issue date and due date of book(s).



Student can give feedback on any book by mentioning Book and Author's Name and branch that book belongs to.



A screenshot of a 'Payment Gateway' window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The title 'Payment Gateway' is centered at the top in a bold black font, with a red 'Exit' button to its right. The main area is light gray and contains the following elements: a 'Reg ID:' label followed by a text box containing '17BEC7039'; a 'Status:' label followed by a dropdown menu showing 'Waiting list'; a 'Payment:' label followed by the text 'Rs. 20'; a 'Book ID:' label followed by a text box containing 'ISIB4444'; a blue 'PAY' button; and the word 'SUCCESS' at the bottom left.

**Payment Gateway** Exit

Reg ID: 17BEC7039

Status: Waiting list ▼

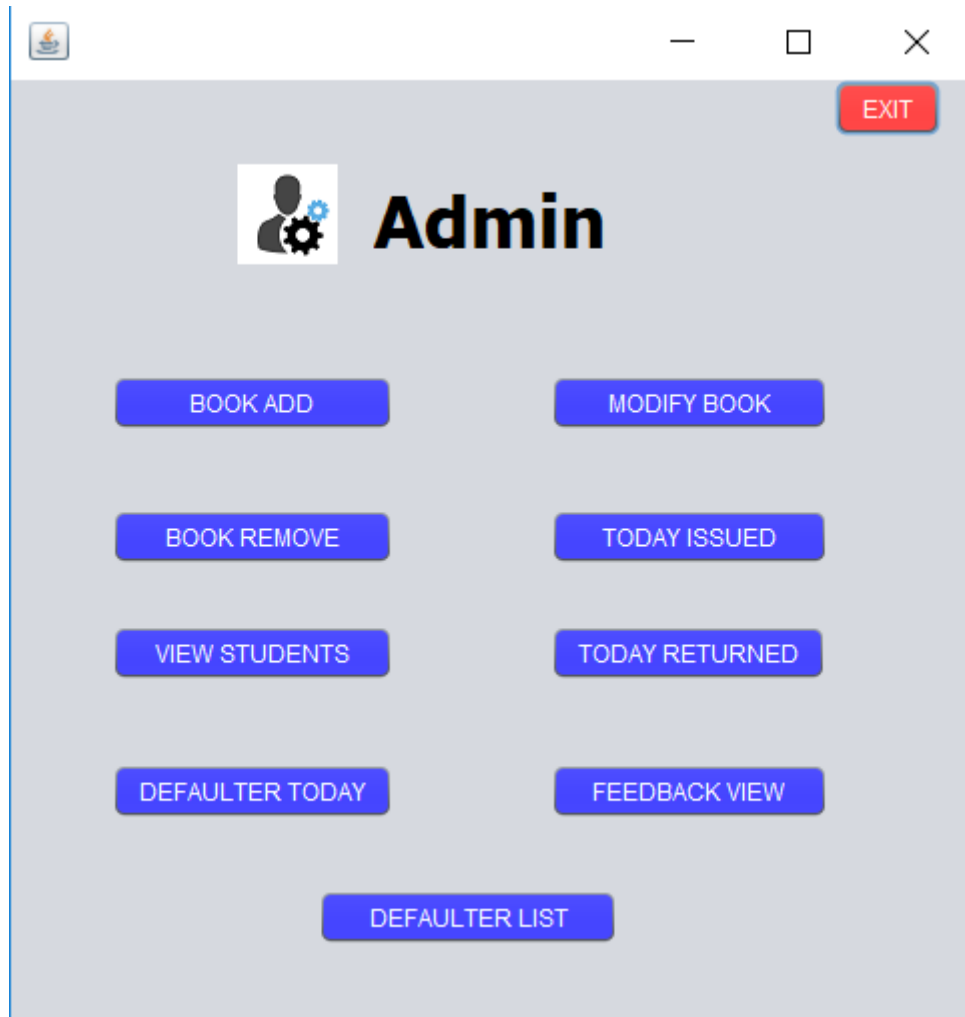
Payment: Rs. 20

Book ID: ISIB4444

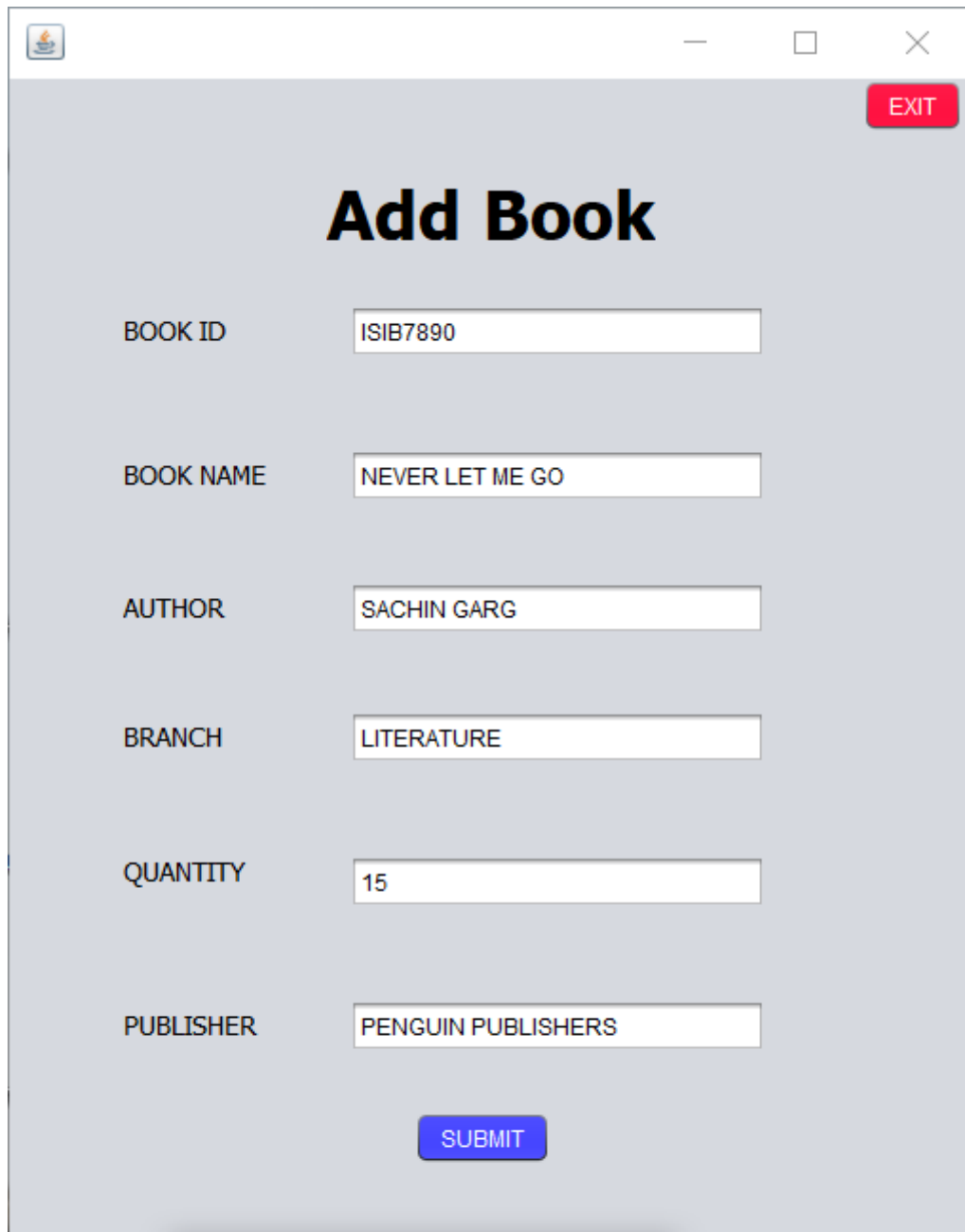
PAY

SUCCESS

**Every EXIT button opens back the Main Page.**

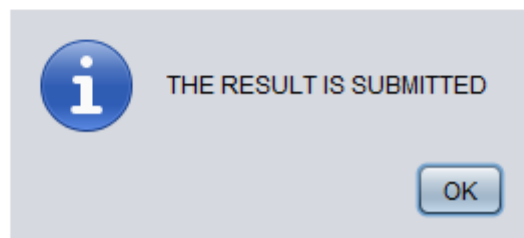


Clicking ADMIN button in Main Page opens this page in which Admin is allowed to add, remove, modify books. He can view the overall review of today issued and returned books and even for defaulter list (even for today).

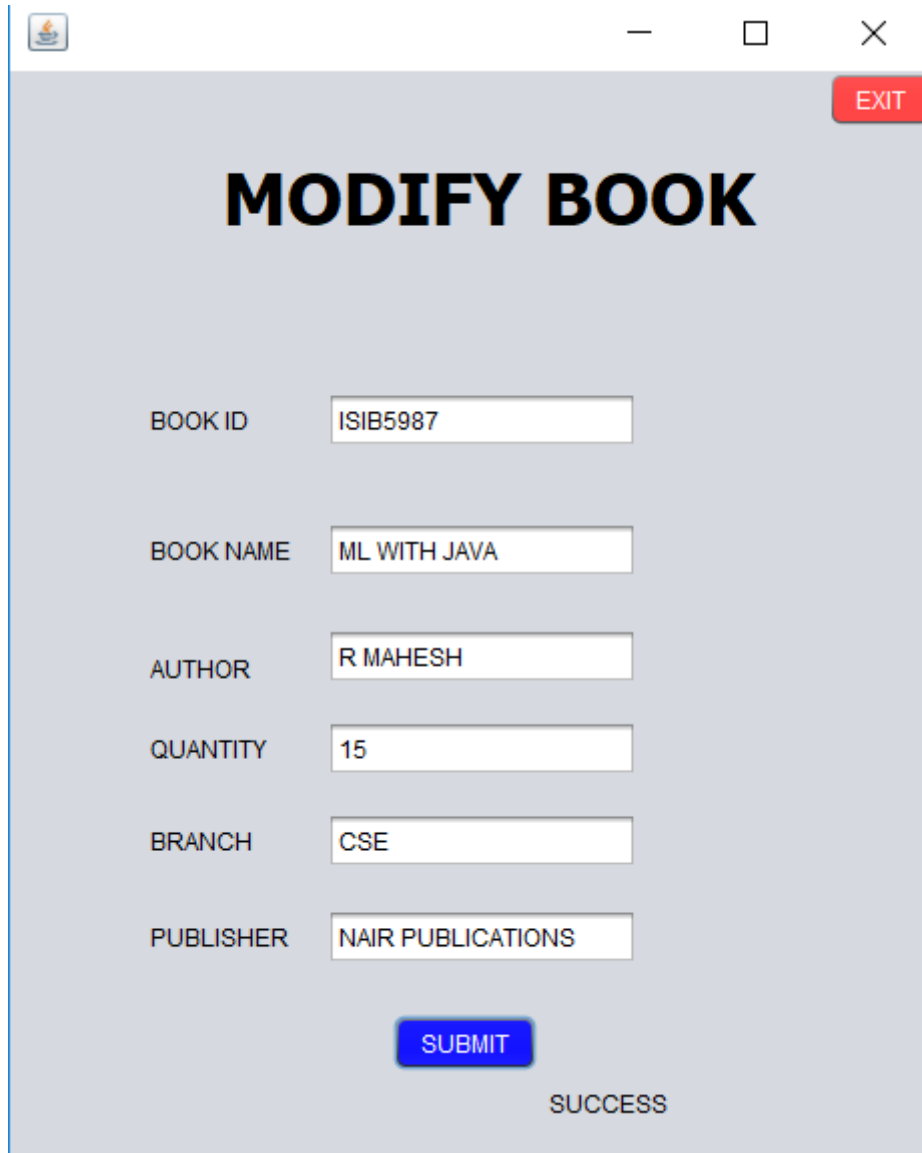


The screenshot shows a web application window with a title bar containing a small icon, a minus sign, a maximize button, and a close button. The main content area has a light gray background. In the top right corner, there is a red button labeled "EXIT". The title "Add Book" is centered in a large, bold, black font. Below the title, there are six input fields, each with a label to its left: "BOOK ID" (containing "ISIB7890"), "BOOK NAME" (containing "NEVER LET ME GO"), "AUTHOR" (containing "SACHIN GARG"), "BRANCH" (containing "LITERATURE"), "QUANTITY" (containing "15"), and "PUBLISHER" (containing "PENGUIN PUBLISHERS"). At the bottom center of the form, there is a blue button labeled "SUBMIT".

Message



Admin can add books (with all necessary details).

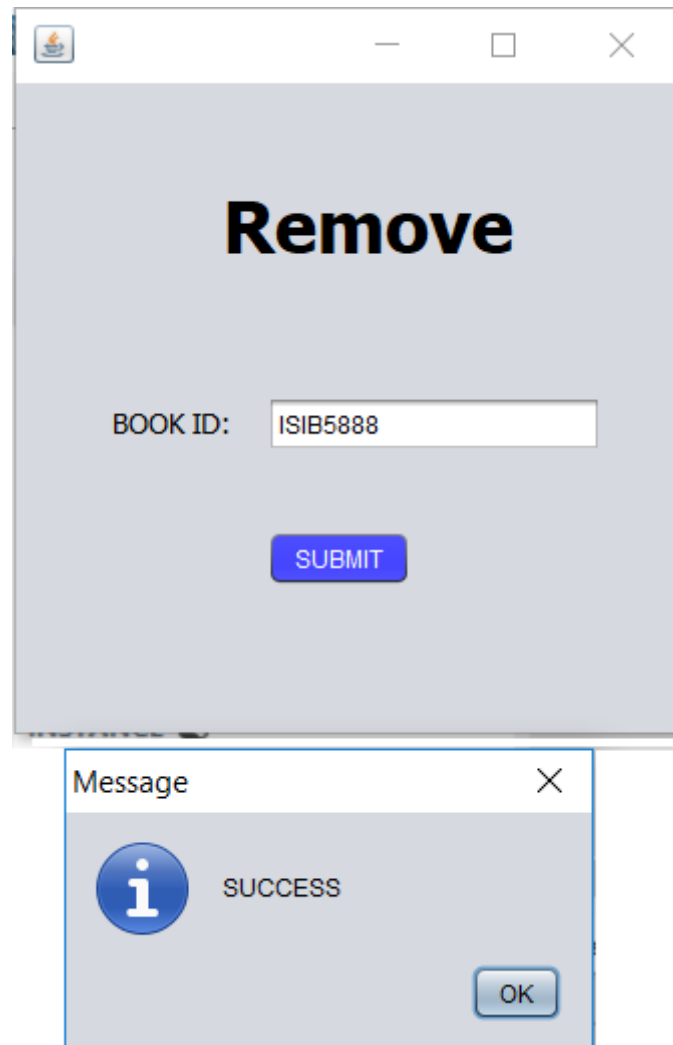


The image shows a web application window titled "MODIFY BOOK". The window has a standard OS-style title bar with a minimize button, a maximize button, and a close button. In the top right corner of the application area, there is a red button labeled "EXIT". The main content area has a light gray background. At the top, the title "MODIFY BOOK" is displayed in large, bold, black letters. Below the title, there are six input fields, each with a label to its left: "BOOK ID" with the value "ISIB5987", "BOOK NAME" with the value "ML WITH JAVA", "AUTHOR" with the value "R MAHESH", "QUANTITY" with the value "15", "BRANCH" with the value "CSE", and "PUBLISHER" with the value "NAIR PUBLICATIONS". Below these fields is a blue button labeled "SUBMIT". At the bottom right of the form area, the word "SUCCESS" is displayed in a smaller font.

Field	Value
BOOK ID	ISIB5987
BOOK NAME	ML WITH JAVA
AUTHOR	R MAHESH
QUANTITY	15
BRANCH	CSE
PUBLISHER	NAIR PUBLICATIONS

**SUCCESS**

Admin can modify any book details.

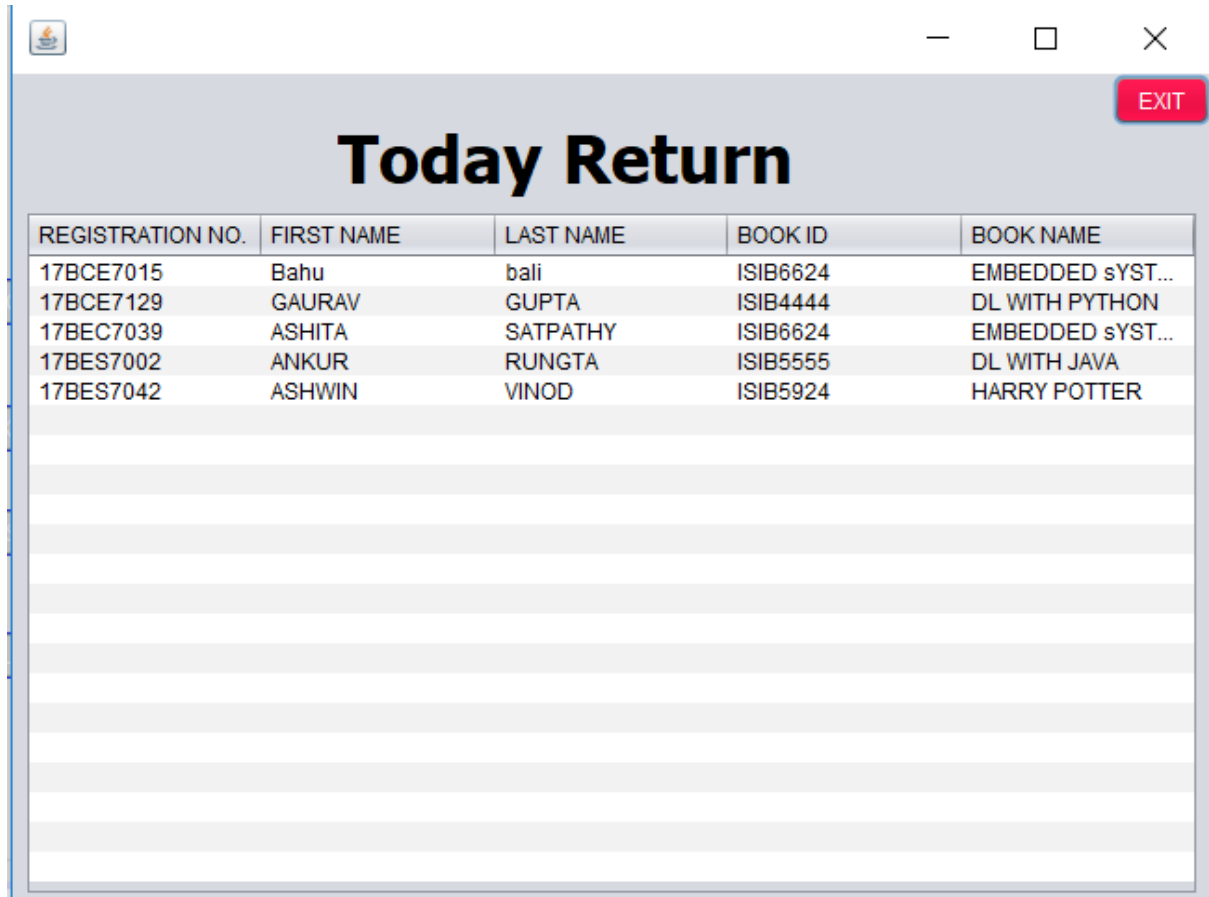


The image shows a web application interface. The main window has a title bar with a small icon and standard window controls. The content area is light gray and features the word **Remove** in large, bold, black text. Below this, there is a label 'BOOK ID:' followed by a text input field containing the value 'ISIB5888'. A blue button with the text 'SUBMIT' is positioned below the input field. In the foreground, a smaller 'Message' dialog box is open. It has a title bar with a close button. The dialog box contains a blue circular icon with a white lowercase 'i' on the left and the word 'SUCCESS' in black text on the right. An 'OK' button is located at the bottom right of the dialog box.

Admin can remove book(s) from his database.

Admin can view no. of students issued book(s) today (that particular day).

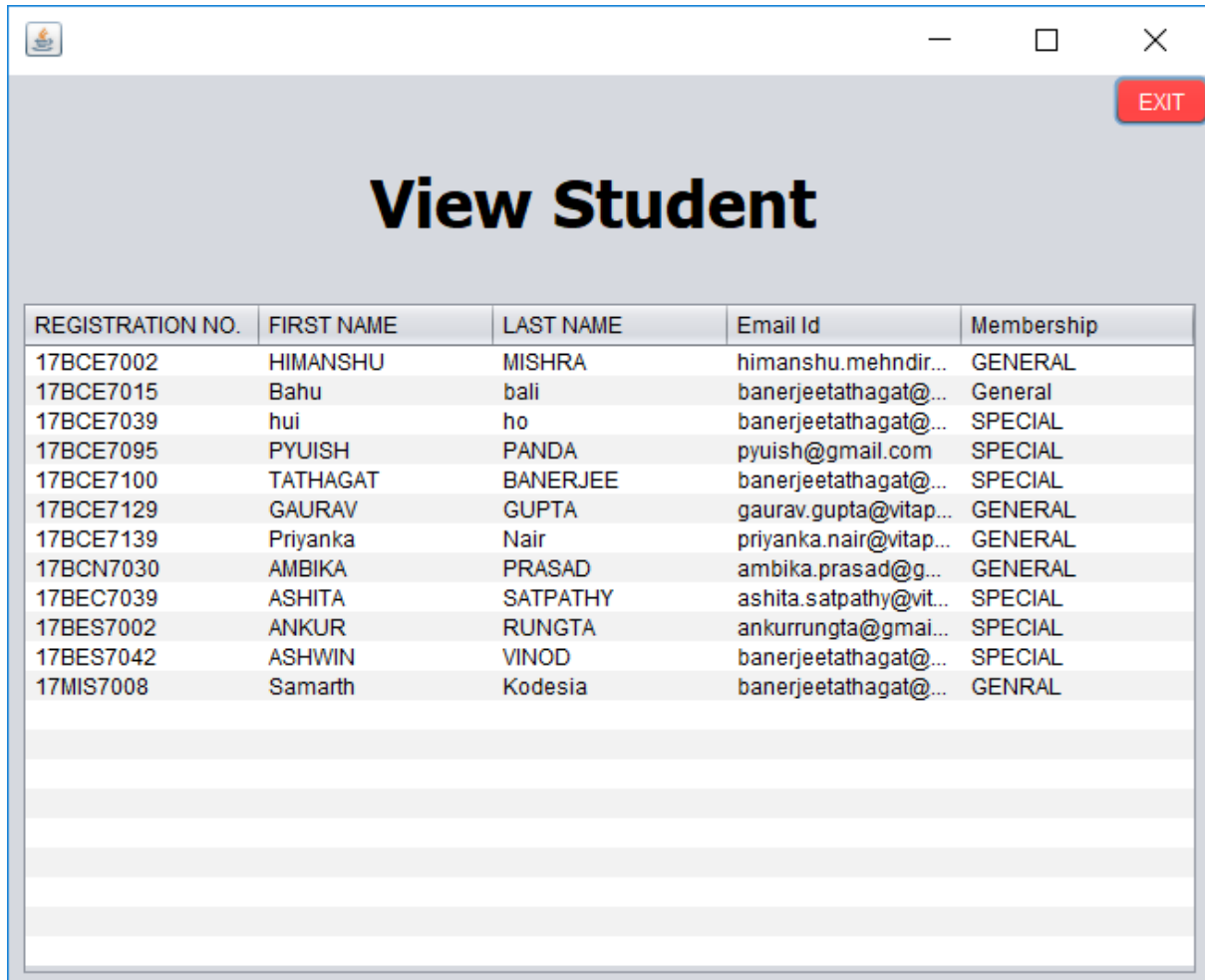




The screenshot shows a web application window with a title bar containing a logo, a minus sign, a square icon, and a close button (X). The window has a light gray header with the title "Today Return" in bold black text and a red "EXIT" button in the top right corner. Below the header is a table with five columns: "REGISTRATION NO.", "FIRST NAME", "LAST NAME", "BOOK ID", and "BOOK NAME". The table contains five rows of data, followed by several empty rows. The data rows are as follows:

REGISTRATION NO.	FIRST NAME	LAST NAME	BOOK ID	BOOK NAME
17BCE7015	Bahu	bali	ISIB6624	EMBEDDED sYST...
17BCE7129	GAURAV	GUPTA	ISIB4444	DL WITH PYTHON
17BEC7039	ASHITA	SATPATHY	ISIB6624	EMBEDDED sYST...
17BES7002	ANKUR	RUNGTA	ISIB5555	DL WITH JAVA
17BES7042	ASHWIN	VINOD	ISIB5924	HARRY POTTER

Admin can view no. of students returned book(s) today (that particular day).



Admin can view all students and their information.

[illegible]

## Sample Code:

- Stand by:

```
public String standby_insert1(String s1,String s2) {  
    String ret="FAILURE";  
    s1=s1.toUpperCase();  
    s2=s2.toUpperCase()  
  
    String sql = "INSERT INTO `library`.`stand_by` (`Reg_no`,  
`book_id`, `Date_applied`) VALUES (?, ?, current_timestamp());";  
    try (Connection conn = this.connect();  
        PreparedStatement pstmt = conn.prepareStatement(sql)) {  
        pstmt.setString(1, s1);  
        pstmt.setString(2, s2);  
        pstmt.executeUpdate();  
        ret="SUCCESS";  
    } catch (SQLException e) {  
        System.out.println(e.getMessage());  
    }  
    return ret;  
}  
  
public void check()  
{  
    Connection s =null;  
    Statement s1=null;  
    ResultSet s2=null;  
  
    String q ="(select Reg_no , book_id , Date_applied ,Issue  
FROM library.stand_by where (adddate(Date_approved , 1) <  
current_timestamp() or Issue = 1 ) and Date_approved is not null  
);";  
  
    try{  
s=DriverManager.getConnection("jdbc:mysql://localhost:3306/library",  
"root", "root");  
        s1=s.createStatement();  
        s2=s1.executeQuery(q);  
    }  
}
```

```
while(s2.next())
{
    String reg1 = s2.getString(1);
    String reg2=s2.getString(2);
    String reg3 = s2.getString(3);
    String reg4 = s2.getString(4);
    //System.out.print(reg1 + " "+reg2+" "+reg3);
    Auto_delete(reg1,reg2,reg3);
    if ( reg4 == (null))
    {
        Quantity_inc(reg2);
        //System.out.println("CLEAR");
        int stand_by =Auto_send(reg2);
        //System.out.println("CLEAR1"+stand_by);
    }
}

}
catch(SQLException e)
{
    e.printStackTrace();
}

}

public void Auto_delete(String s1 , String s2 , String s3 )
{
    s1=s1.toUpperCase();

    String sql = "delete from library.stand_by where Reg_no
='"+s1+"' and book_id ='"+s2+"' and Date_applied ='"+s3+"'";

    try (Connection conn = this.connect();
    PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.executeUpdate();
    }
}
```

```
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
    public void update_standby(String s1,String s2)
    {
        s1=s1.toUpperCase();
        s2=s2.toUpperCase();
        if(id_verification_student(s1)){
            String sql = "update library.stand_by set Issue = 1 where
Reg_no = '"+s1+"' and book_id = '"+s2+"' and date_approved >
adddate(current_timestamp(),-1)";
            try (Connection conn = this.connect();
                PreparedStatement pstmt =
conn.prepareStatement(sql)) {
                pstmt.executeUpdate();
            } catch (SQLException e) {
                System.out.println(e.getMessage());
            }
        }
        else
        {
            System.out.println("Id do not exists");
        }
    }
}
```

- Admin viewing today's issue

```
public String[] todays_issue()
{
    String ret[] = new String[100] ;
    Connection s =null;
    Statement s1=null;
    ResultSet s2=null;

String q ="SELECT s.Reg_no , s.fname ,s.lname ,b.Book_id
,b.Book_name FROM library.issue_det as i join library.book_det as b
join library.student_det as s where i.Registration_id = s.Reg_no
and i.Book_id = b.Book_id and issue_date =curdate()";

    try{
s=DriverManager.getConnection("jdbc:mysql://localhost:3306/library",
"root", "root");

        s1=s.createStatement();
        s2=s1.executeQuery(q);
        int i=0;
        while(s2.next())
        {
            String reg1 = s2.getString(1);
            String reg2=s2.getString(2);
            String reg3=s2.getString(3);
            String reg4=s2.getString(4);
            String reg5=s2.getString(5);

            ret[i]=(reg1+", "+reg2+", "+reg3+", "+reg4+", "+reg5);
            //System.out.print(ret[i]);
            i++;
        }

    }
    catch(SQLException e)
    {
```

```
        e.printStackTrace();
    }
    return ret;
}
```

- **Admin viewing today's return**

```
public String[] todays_return()
{
    String ret[] = new String[100] ;
    Connection s =null;
    Statement s1=null;
    ResultSet s2=null;
    String q ="SELECT s.Reg_no , s.fname ,s.lname ,b.Book_id
,b.Book_name FROM library.transaction as i join library.book_det
as b join library.student_det as s where i.Reg_no = s.Reg_no  and
i.Book_id = b.Book_id and reissue_date =curdate()";
    try{

s=DriverManager.getConnection("jdbc:mysql://localhost:3306/librar
y", "root", "root");
        s1=s.createStatement();
        s2=s1.executeQuery(q);
        int i=0;
        while(s2.next())
        {
            String reg1 = s2.getString(1);
            String reg2=s2.getString(2);
            String reg3=s2.getString(3);
            String reg4=s2.getString(4);
            String reg5=s2.getString(5);
            ret[i]=(reg1+", "+reg2+", "+reg3+", "+reg4+", "+reg5);
            //System.out.print(ret[i]);
            i++;
        }

    }
    catch(SQLException e)
    {
        e.printStackTrace();
    }
    return ret;
}
```



- Admin viewing defaulters

```
public String[] defaulter_students_today()
{
    String ret[] = new String[100] ;
    Connection s =null;
    Statement s1=null;
    ResultSet s2=null;
    String q ="SELECT s.Reg_no , s.fname ,s.lname ,b.Book_id
,b.Book_name FROM library.issue_det as i join library.book_det as
b join library.student_det as s where i.Registration_id =
s.Reg_no and i.Book_id = b.Book_id and due_date =curdate() and
return_date is null";
    try{

s=DriverManager.getConnection("jdbc:mysql://localhost:3306/librar
y", "root", "root");
        s1=s.createStatement();
        s2=s1.executeQuery(q);
        int i=0;
        while(s2.next())
        {
            String reg1 = s2.getString(1);
            String reg2=s2.getString(2);
            String reg3=s2.getString(3);
            String reg4=s2.getString(4);
            String reg5=s2.getString(5);
            ret[i]=(reg1+","+reg2+","+reg3+","+reg4+","+reg5);
            //System.out.print(ret[i]);
            i++;
        }

    }
    catch(SQLException e)
    {
        e.printStackTrace();
    }
    return ret;
}

public String[] defaulter_students_pending()
{
    String ret[] = new String[100] ;
    Connection s =null;
    Statement s1=null;
    ResultSet s2=null;
    String q ="SELECT s.Reg_no , s.fname ,s.lname ,b.Book_id
,b.Book_name ,i.due_date FROM library.issue_det as i join
library.book_det as b join library.student_det as s where
```

```
i.Registration_id = s.Reg_no  and i.Book_id = b.Book_id and
due_date <= curdate() and return_date is null";
    try{

s=DriverManager.getConnection("jdbc:mysql://localhost:3306/librar
y", "root", "root");
    s1=s.createStatement();
    s2=s1.executeQuery(q);
    int i=0;
    while(s2.next())
    {
        String reg1 = s2.getString(1);
        String reg2=s2.getString(2);
        String reg3=s2.getString(3);
        String reg4=s2.getString(4);
        String reg5=s2.getString(5);
        String reg6 =s2.getString(6);
        ret[i]=(reg1+", "+reg2+", "+reg3+", "+reg4+", "+reg5);
        //System.out.print(ret[i]);
        i++;
    }

    }
    catch(SQLException e)
    {
        e.printStackTrace();
    }
    return ret;

}
```