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:

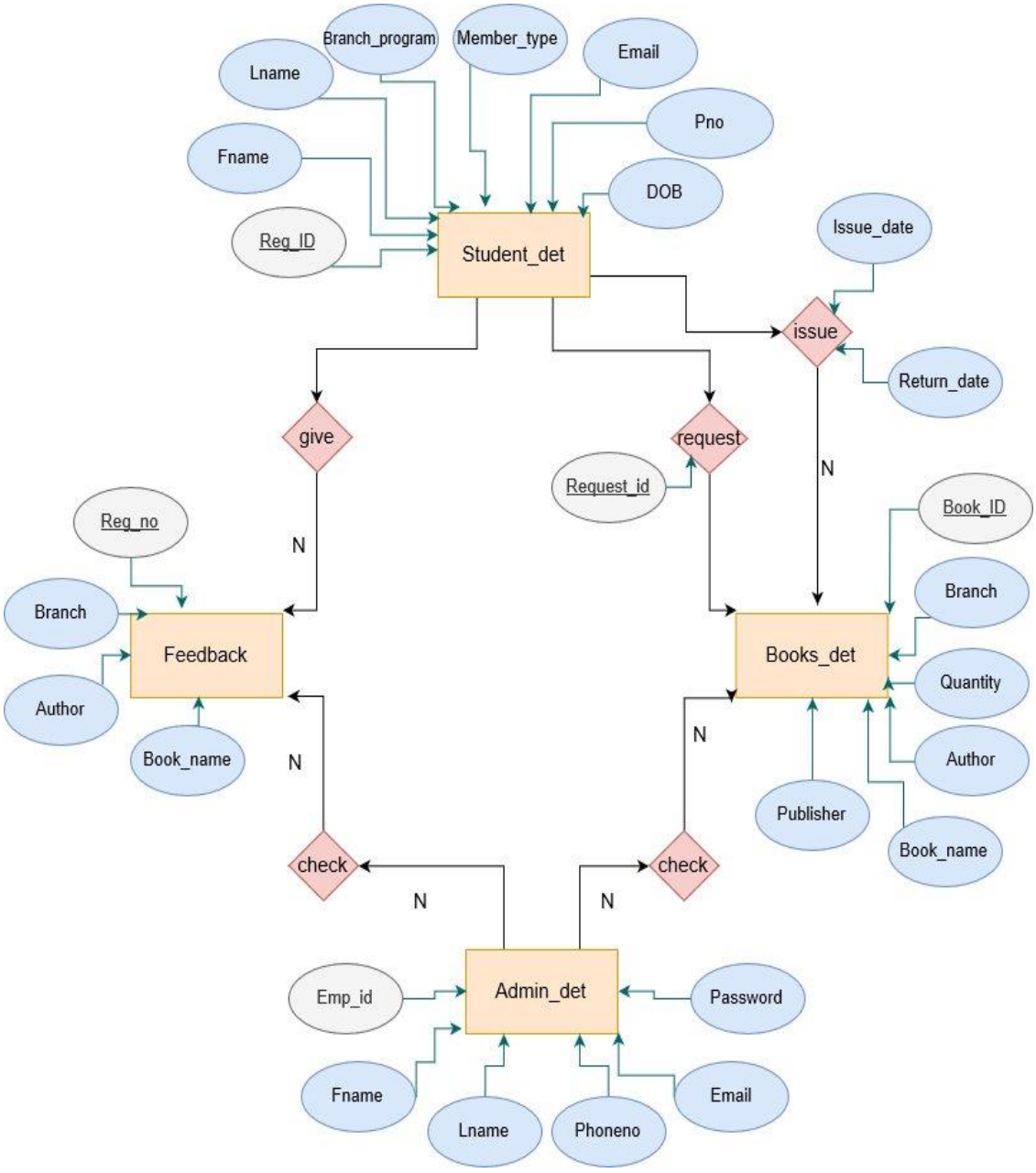
- The followings are the entities in Library Management System:
    1. Books_det
    2. Student_det
    3. Issue_Book_det
    4. Admin_det
    5. feedback
- The followings are the attributes of the entities:
    1. Books_det (Book_ID, Book_name, Author, Branch, Quantity, Publisher)
    2. Student_det (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.

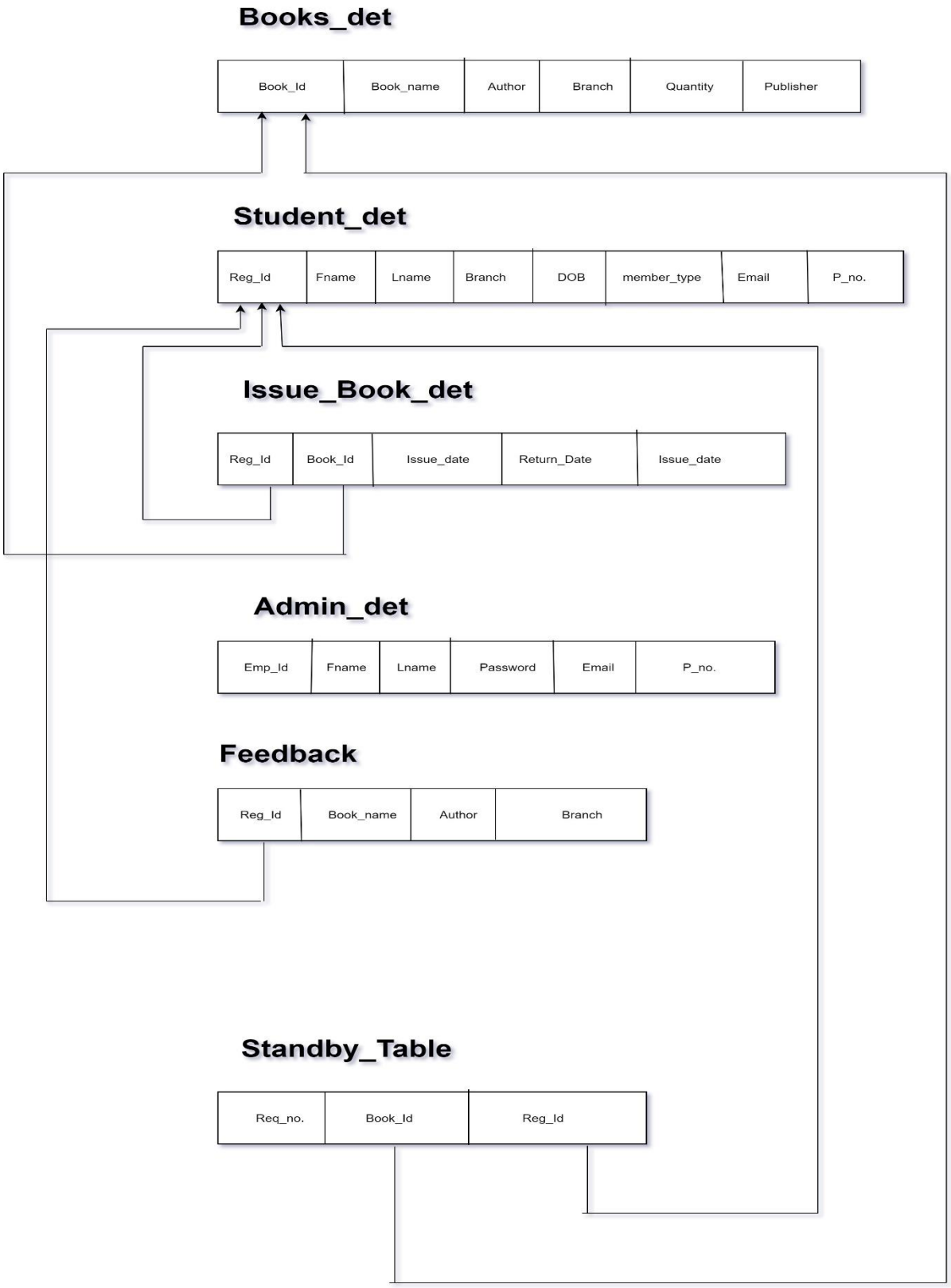
If stock of book is over, student can even send a request for the required book by creating a request id for the book.

- 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:



Functional dependency:

Book_id → book()
Reg_id → Student()

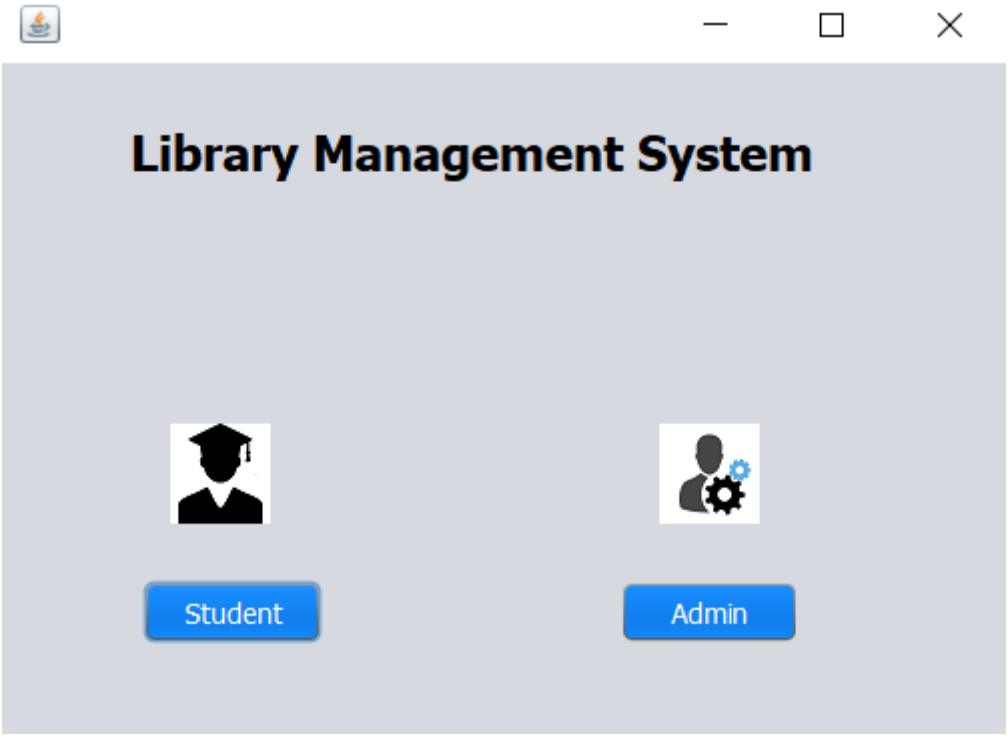
Book_id, Reg_id → Issue()

Book_id, Reg_id → Standby

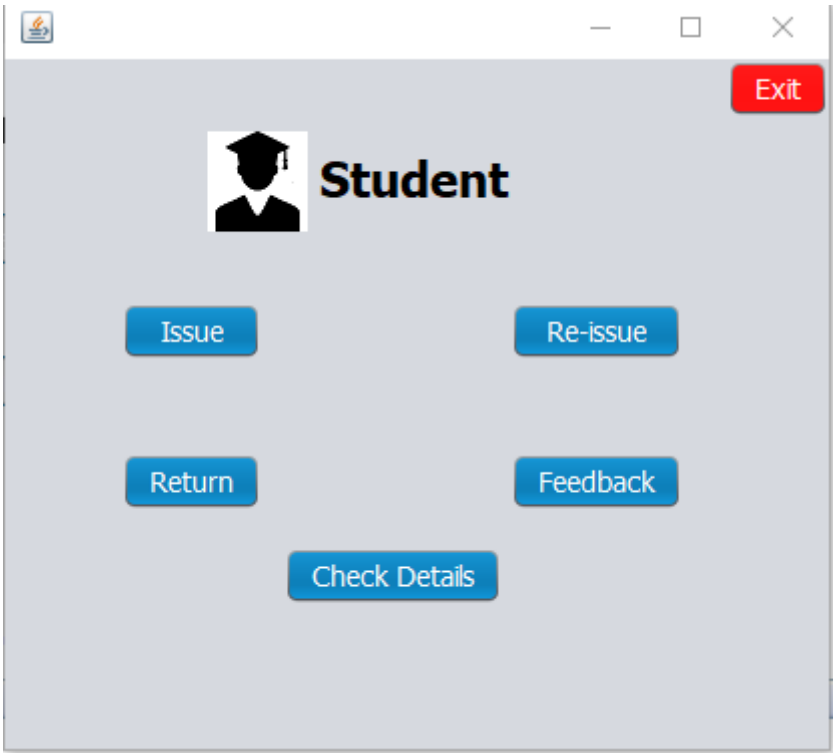
Reg_id → feedback()
Emp_id → Admin()

Admin has authorized permission to books and feedback.

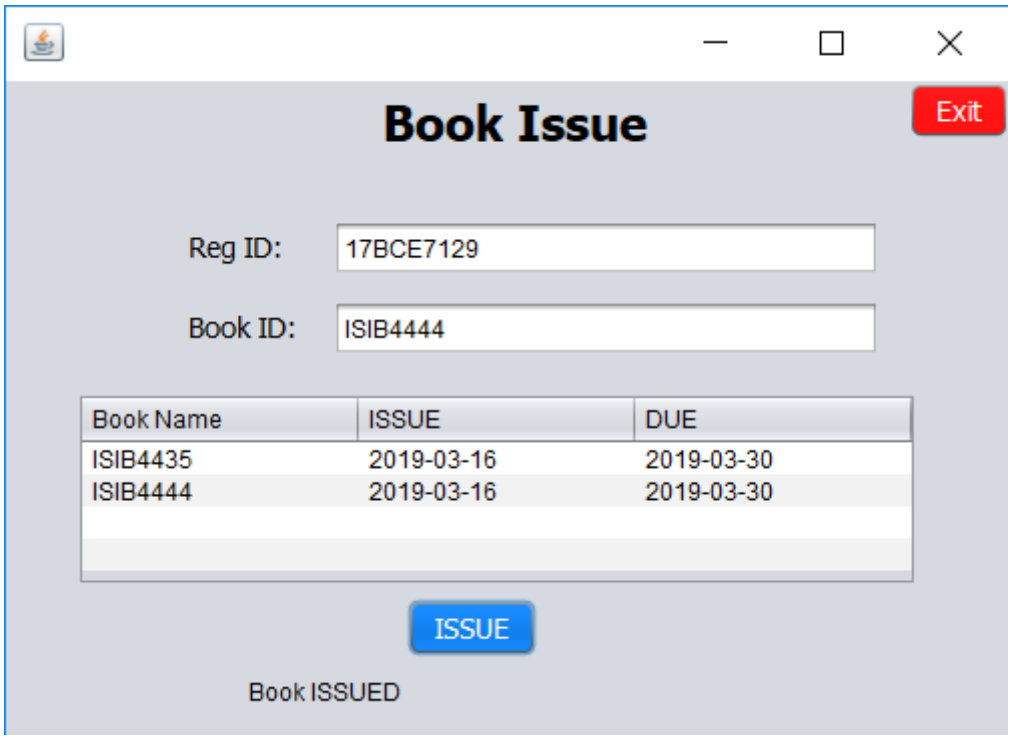
Screenshots:



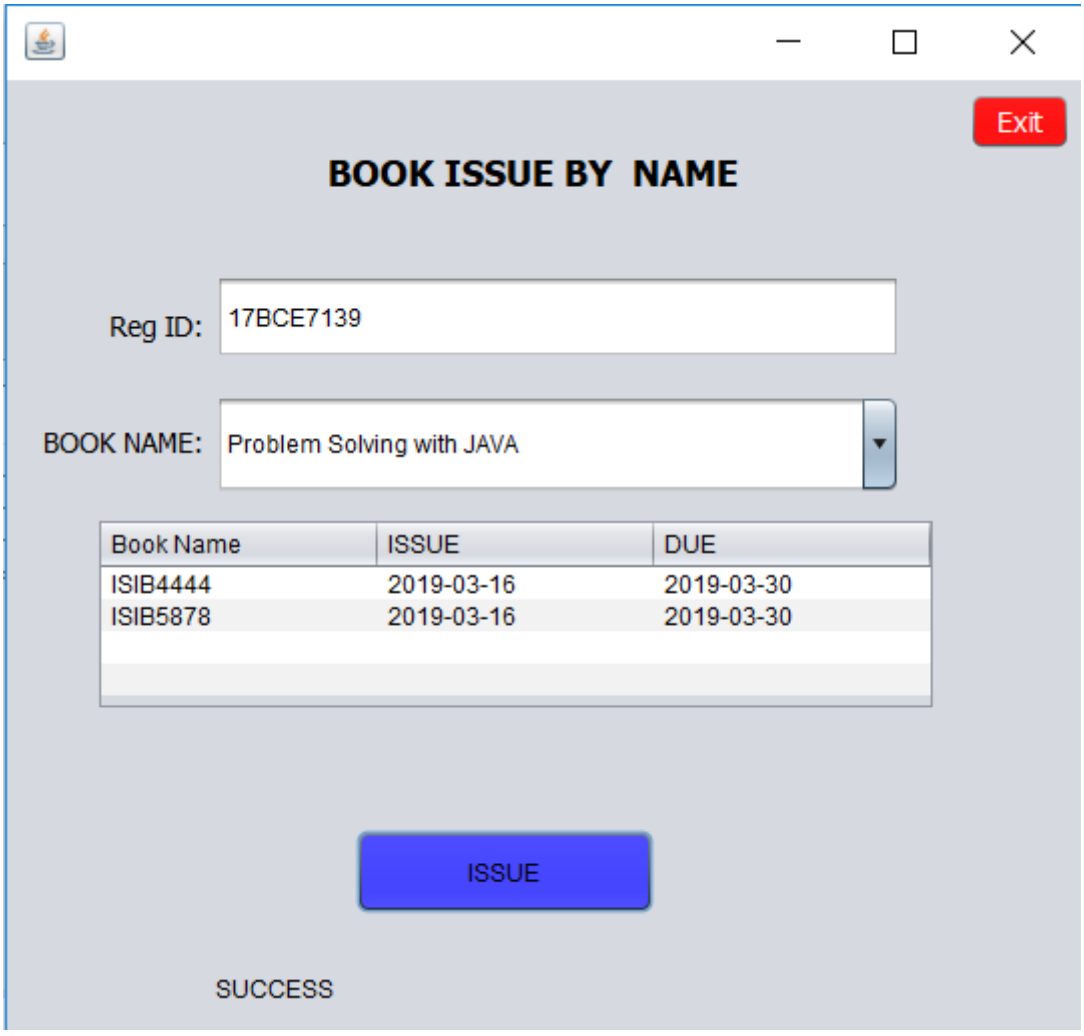
This is the **Main Page** of Library Management System which contains two buttons for Student and Admin.



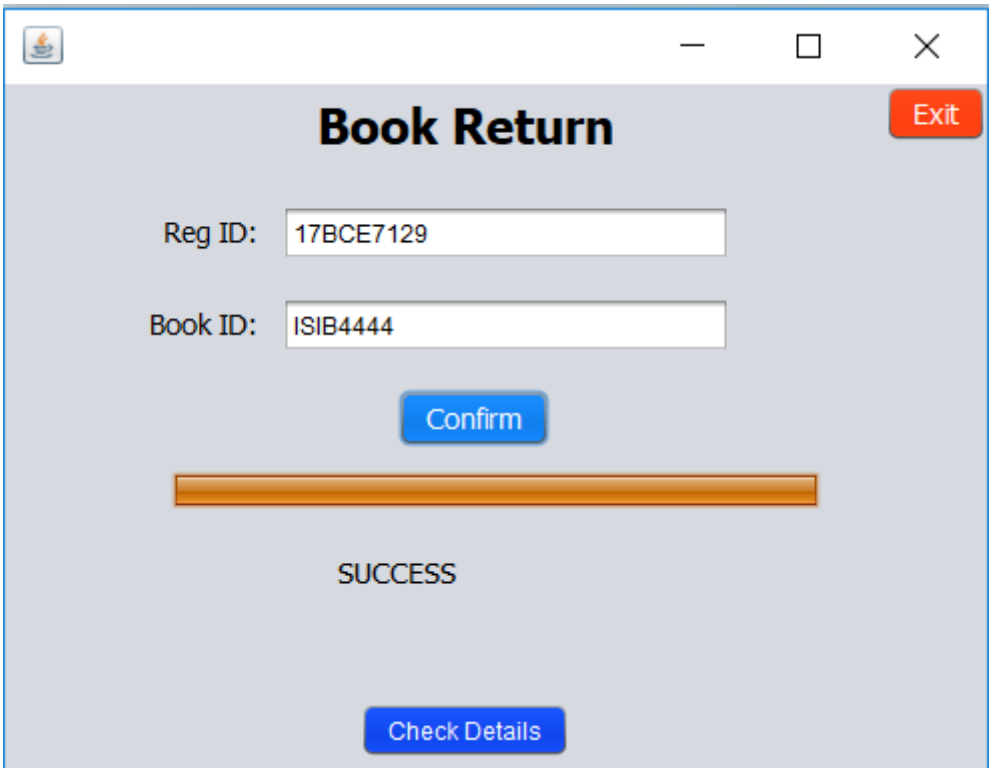
Student button opens a page for Issuing, re issuing, returning of books. Student can check their details and even give feedback for books.



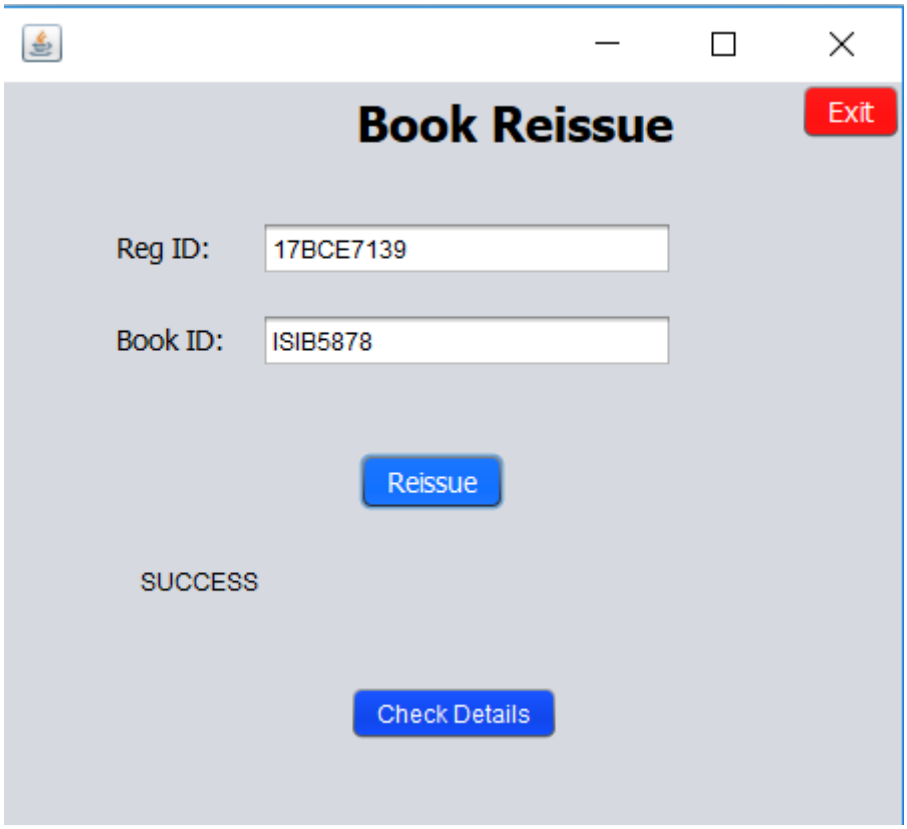
Student can issue NOT more than 2 books at a time.



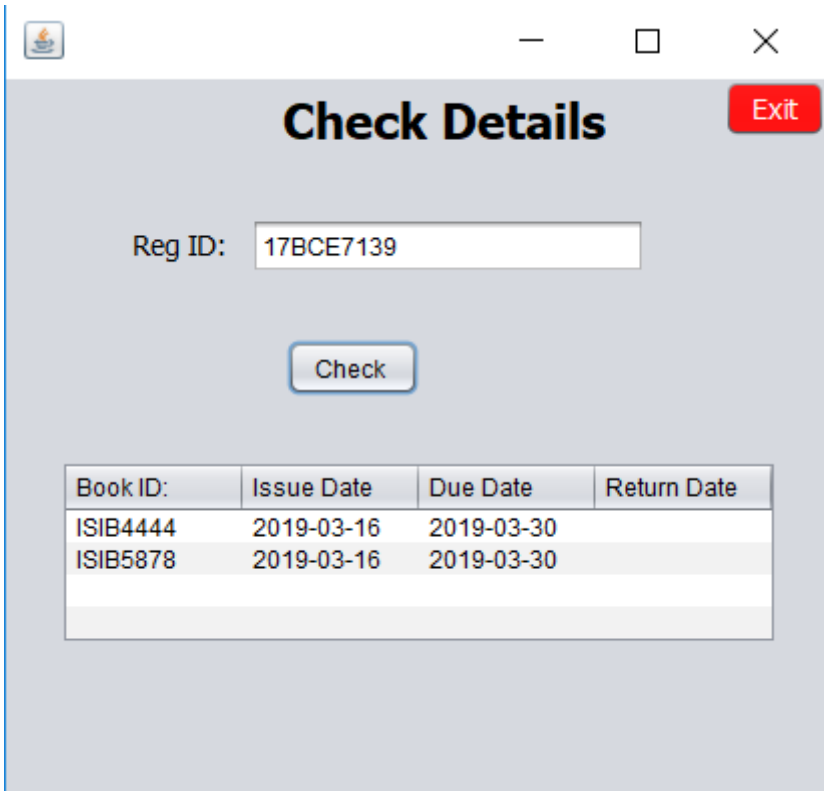
Student can even search book name (using autocomplete dropdownlist in jcombobox) and issue books.



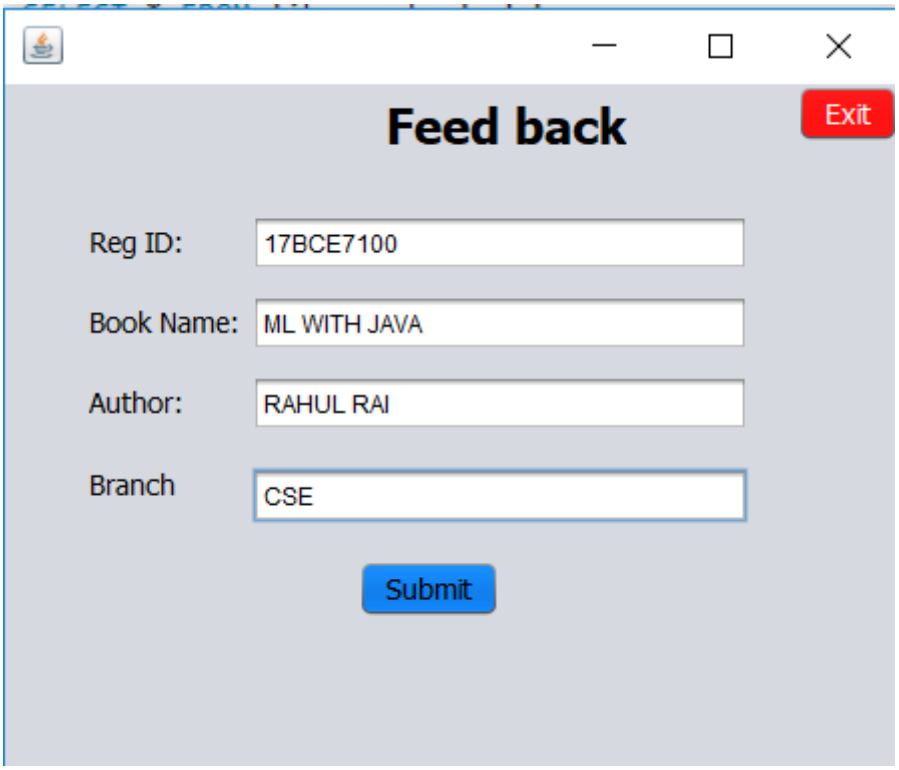
Student's current book ID is matched with the string provided in Book ID and then it is deducted from the issued book of that student's database.



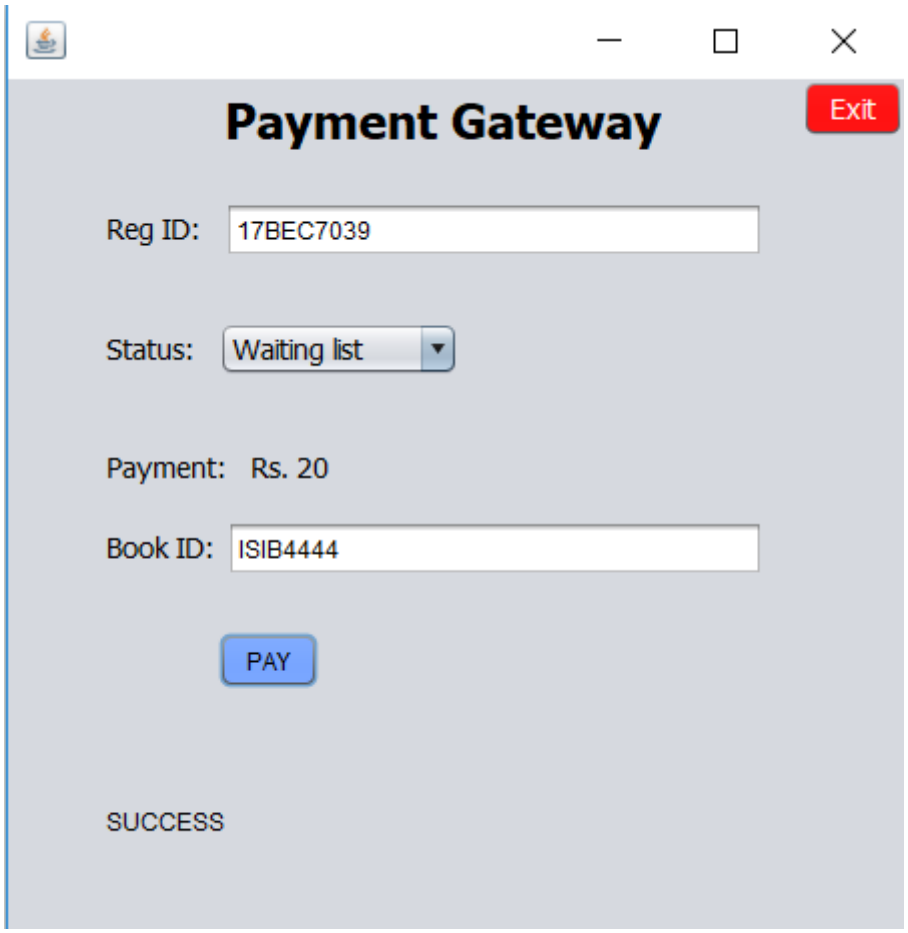
If Student can reissue book if he/she has already issued that book earlier.



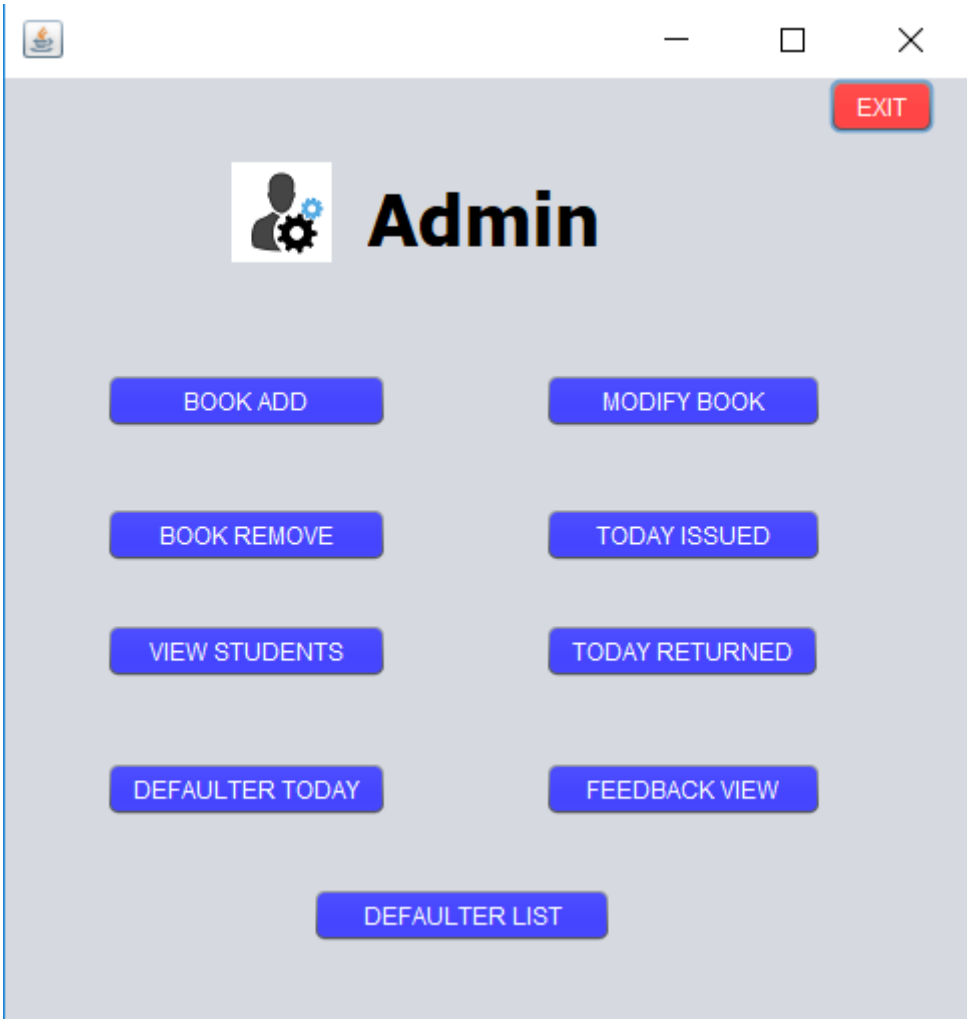
Student can know their issued book ID, issue date and due date of book(s).



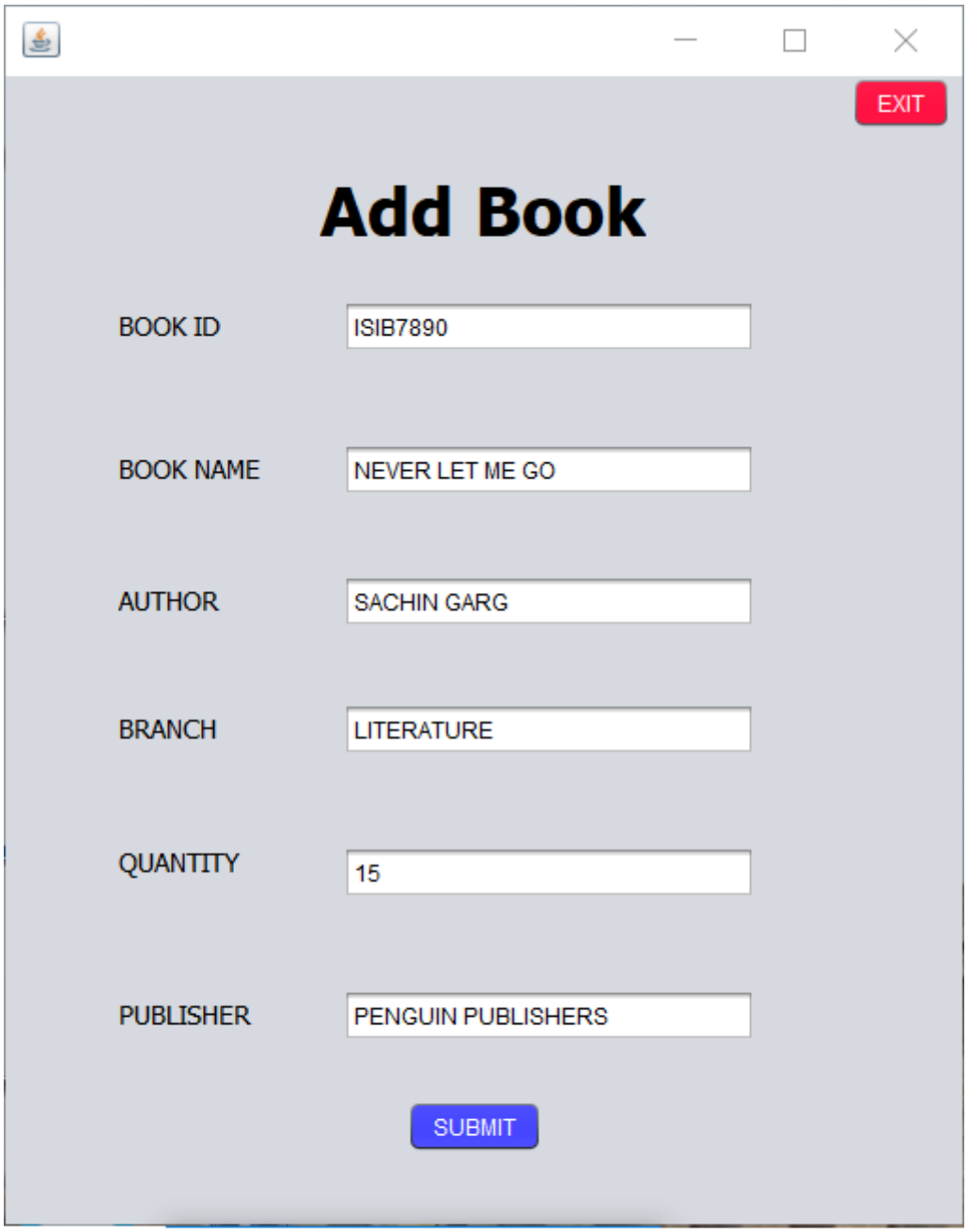
Student can give feedback on any book by mentioning Book and Author's Name and branch that book belongs to.

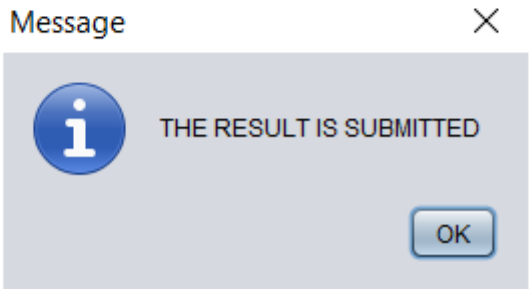


Every EXIT button opens back the Main Page.

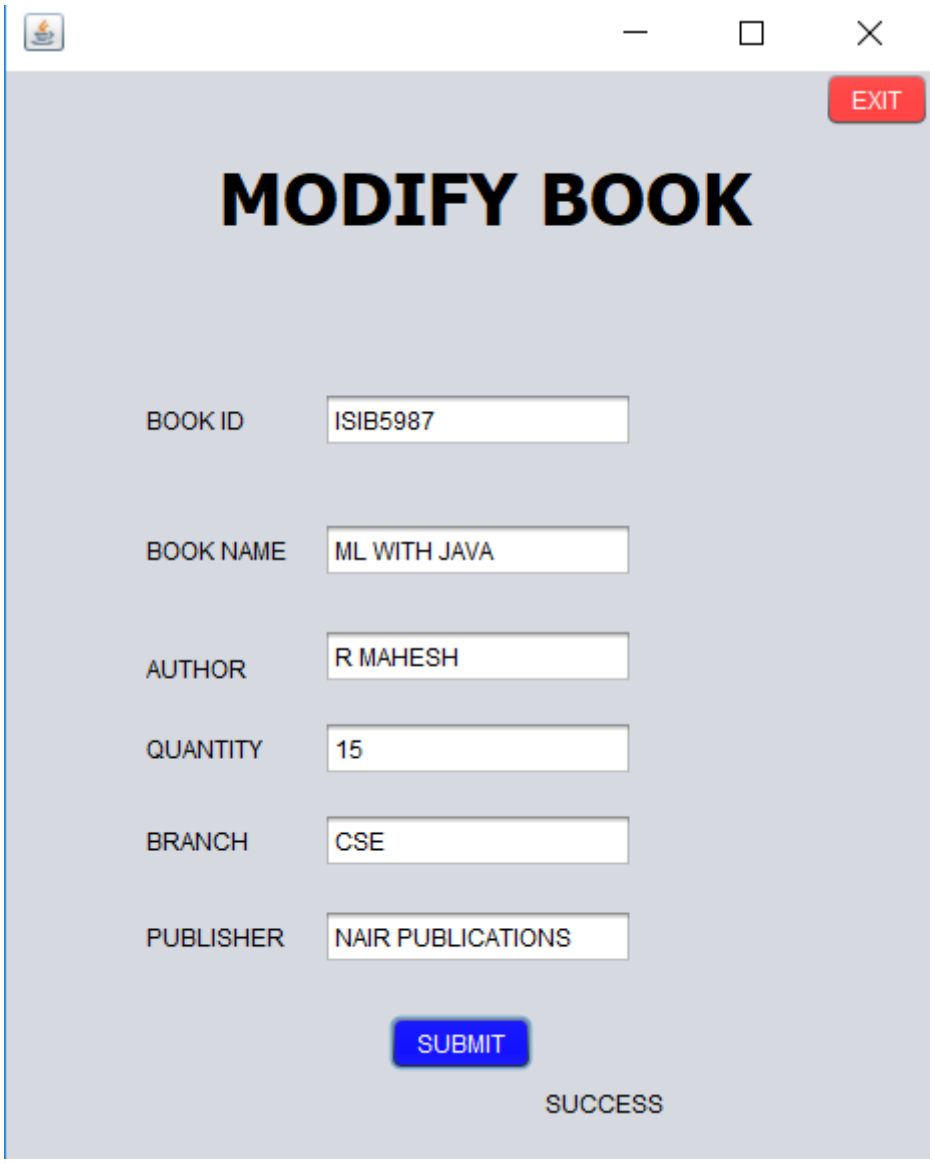


Clicking ADMIN button in Main Page opens this page in which Admin is allowed to add, remove, modify books. He can view the overall review of today issued and returned books and even for defaulter list (even for today).

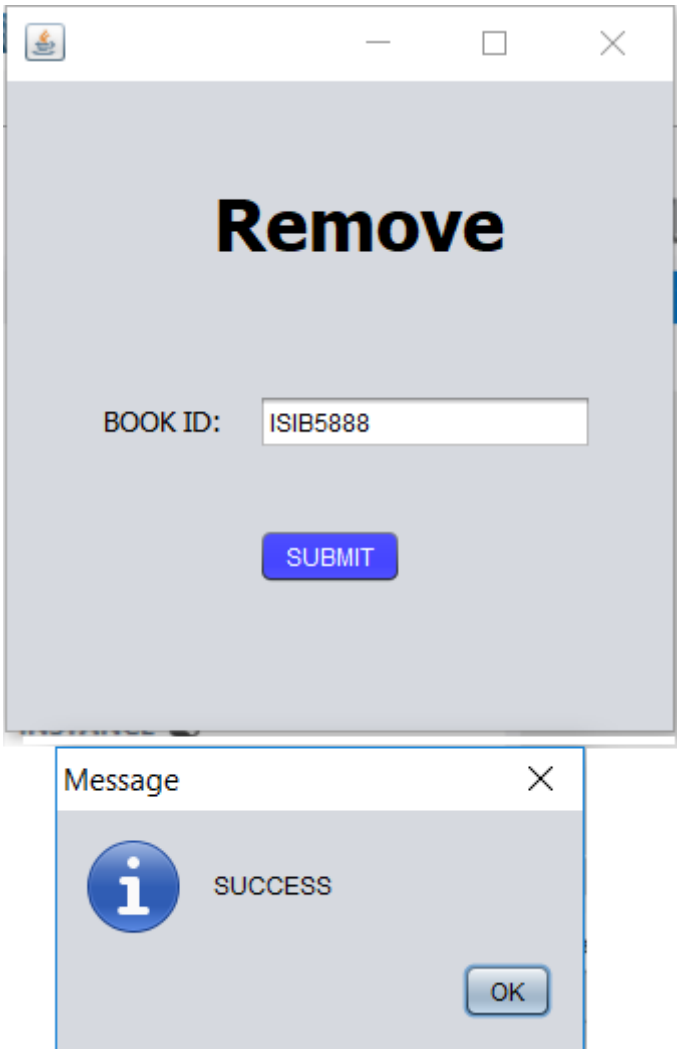




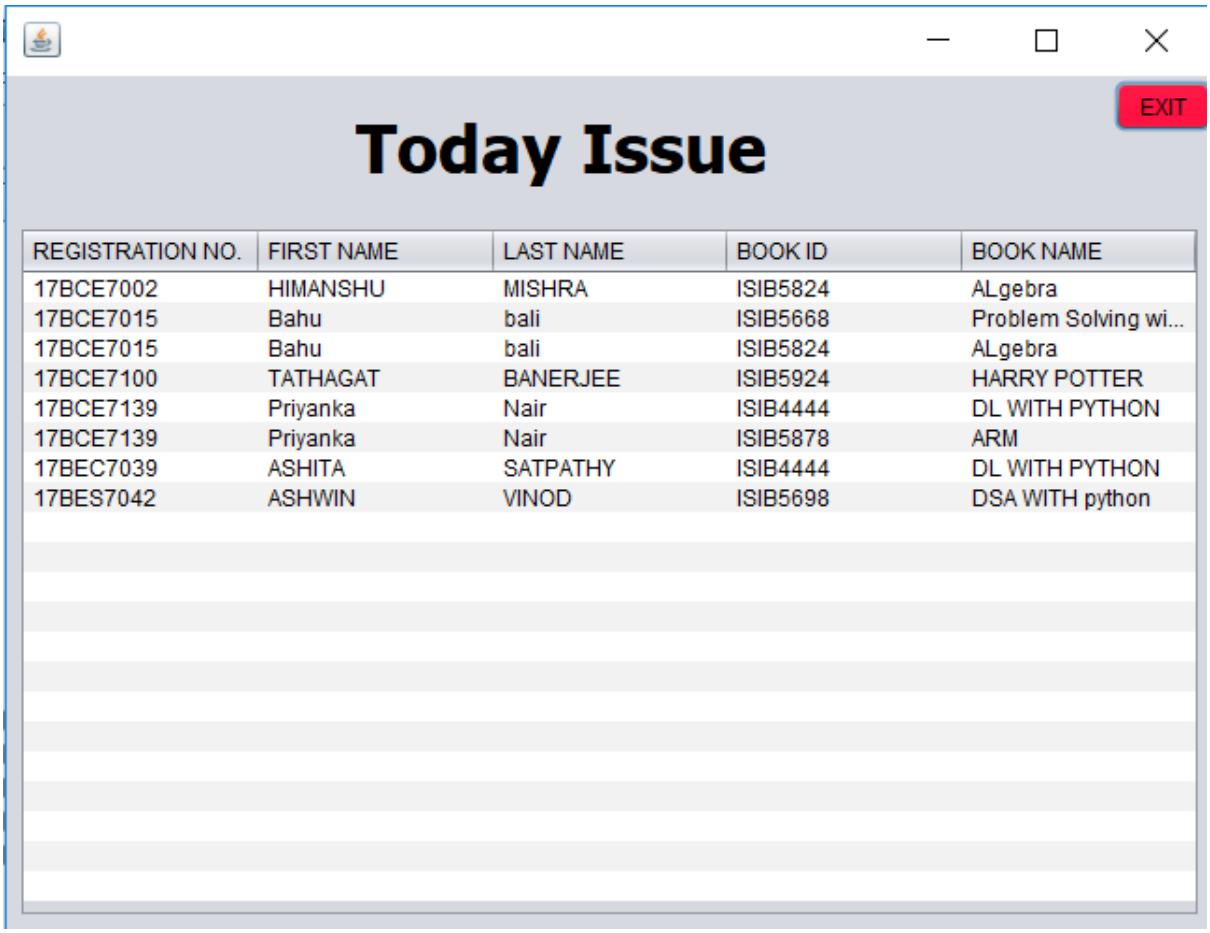
Admin can add books (with all necessary details).



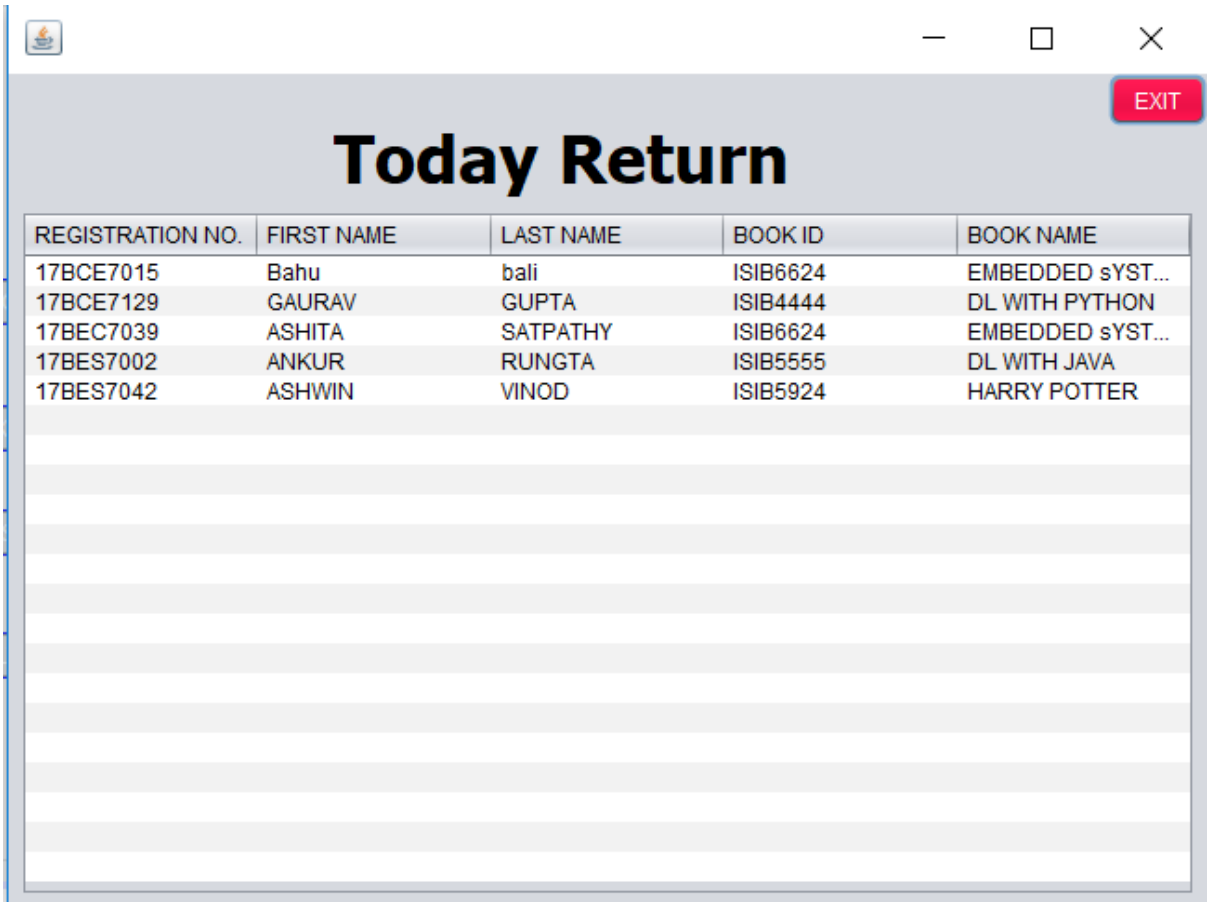
Admin can modify any book details.



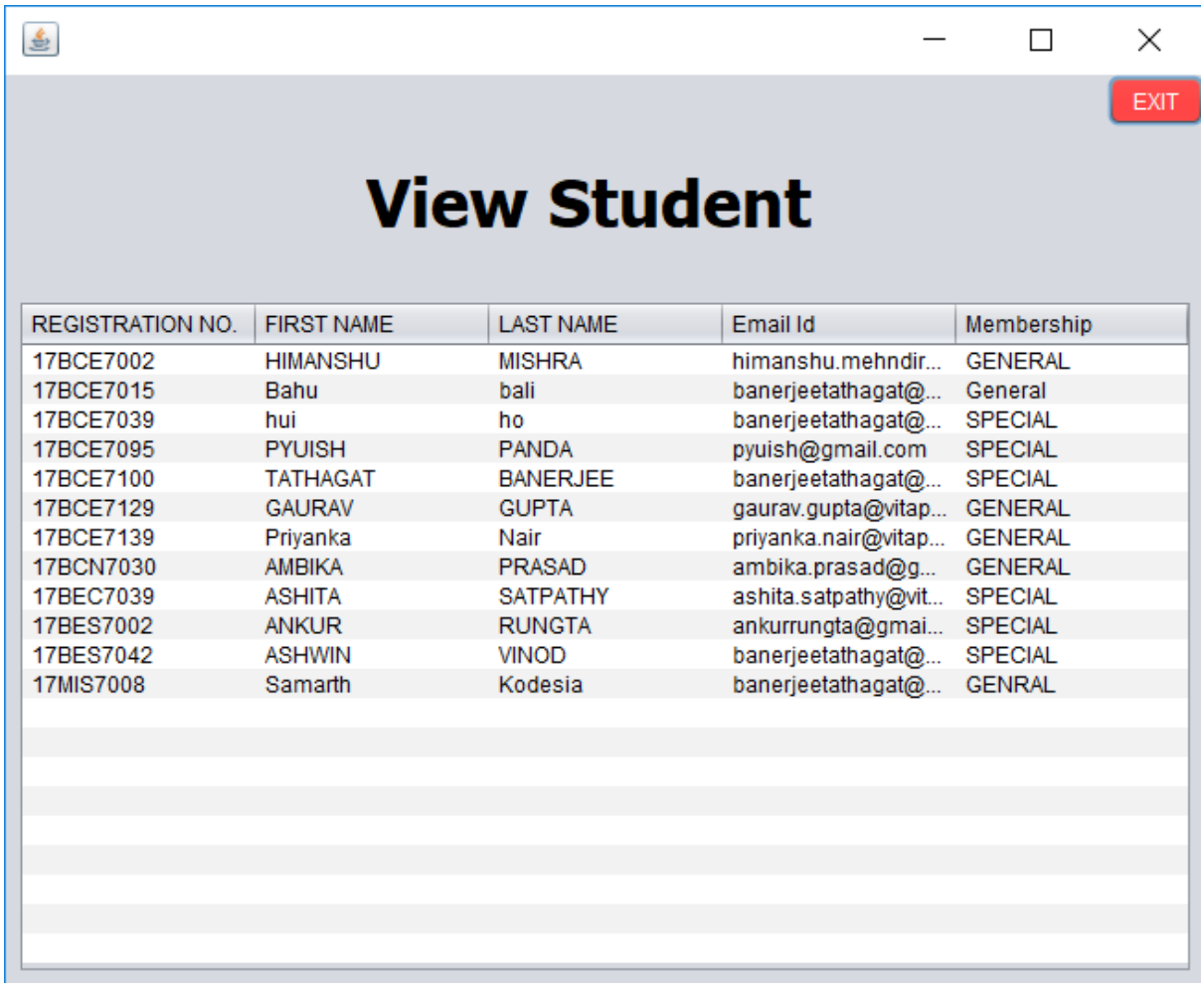
Admin can remove book(s) from his database.



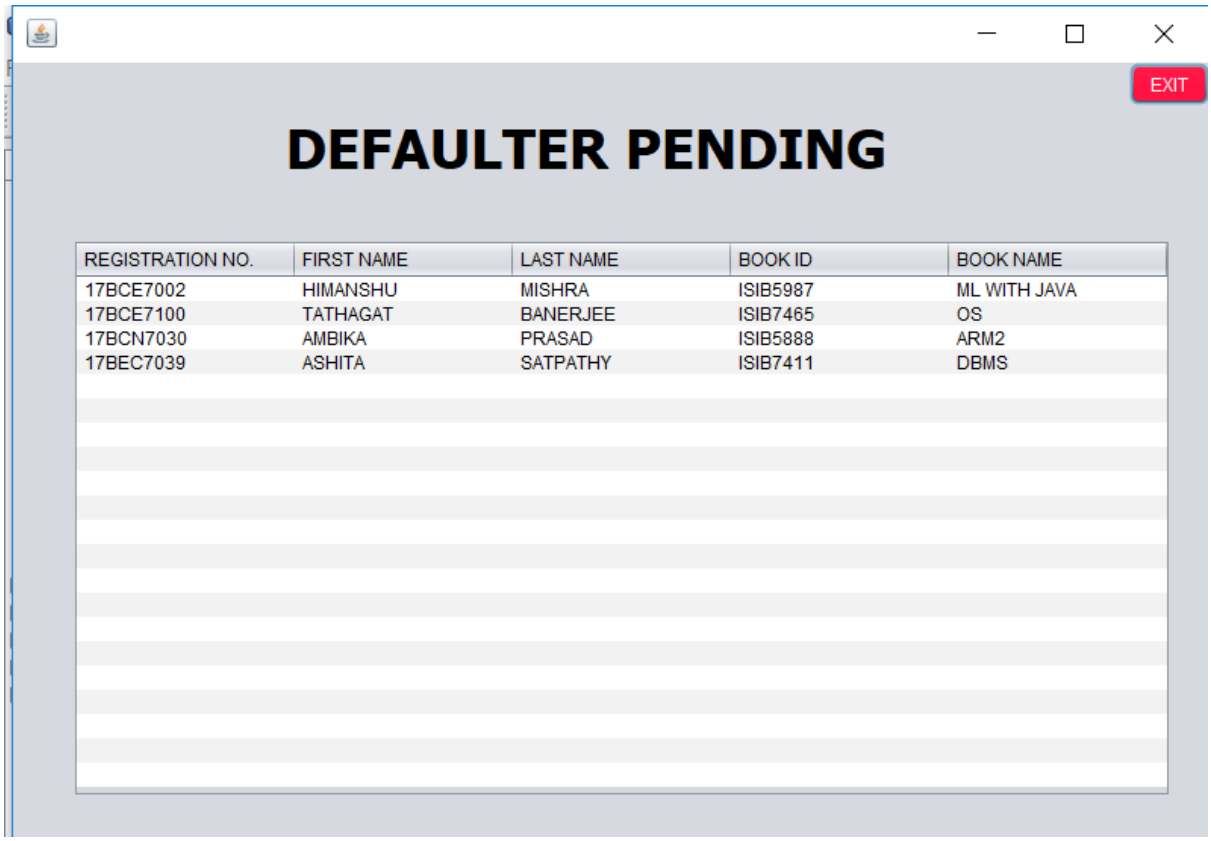
Admin can view no. of students issued book(s) today (that particular day).



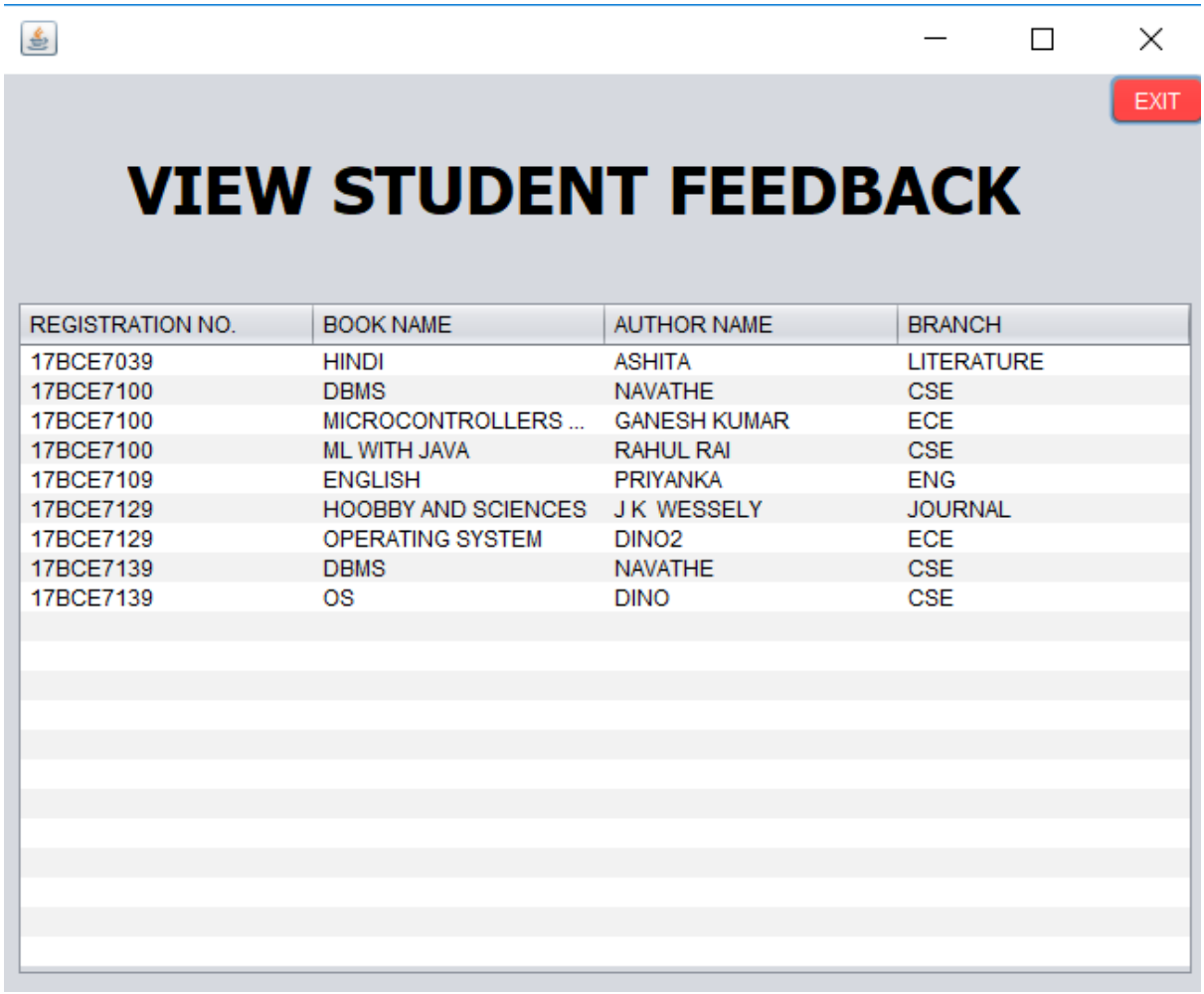
Admin can view no. of students returned book(s) today (that particular day).



Admin can view all students and their information.



Admin can list out number of defaulters.



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;

    }
```