Theme Based Project Report

On

LIBRARY MANGEMENT SYSTEM

Submitted by,

(Students of BE (UG) 3rd Year 6th Semester ECE – Section A)

Abhigna Ogirala

(1602-17-735-001)

A. Madhulika

(1602-17-735-025)

Rohith Kumar Jerripothula

(1602-17-735-038)

Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2019-20

ACKNOWLEDGMENT

We take this opportunity to express our gratitude and heartfelt thanks to our esteemed guide, Ms.S.Aruna Deepthi, Department of Electronics and Communication Engineering, whose guidance and support has been invaluable and helped us in reaching the final outcome of the project.

We are also thankful to Dr.E.Sreenivasa Rao, Head of Department (ECE), Vasavi College of Engineering for the motivation and guidance throughout our course work.

Finally, we also extend our sincere gratitude to our Theme Based Project coordinator Mr. N Abid Ali Khan, Department of Electronics and Communication Engineering for his constant support and encouragement during the entire process.

INDEX

INTRODUCTION

The Library Management System is a CRUD (Create, Read, Update and Delete) application intended for handling and maintaining the daily library transactions in a trouble-free way. This project is being implemented on Java Platform which enables operations like adding new books and adding new students which are updated to the database. This project also facilitates the user to issue books and consequently appends all pertinent issue and return record details to the MySQL database connection with which is employed using the WAMP server.

These records, along with all entered student and book details can be viewed by the user using this application. In addition, it also supports searching operation that permits the user to search for available books, students in the database and also provides features like updating copies, maintain student book count, deleting the existing books etc.

The System mentioned above, is a Windows Desktop Application which is user-friendly and is very easy to operate on, for both naïve as well advanced users. This project has been developed to ease the maintenance of library and to preserve transactions digitally instead of manually writing them down thereby, supplanting the antiquated method with a more effective and uncomplicated one.

LITERATURE SURVEY

Applications similar to the one stated above are there in existence wherein the developers have made an attempt to digitize the existing library systems. Most of those applications enable basic operations of adding books and issuing them. In our application we have gone a step ahead and made an effort to create more attractive and interactive user interface besides adding some more features.

PROBLEM DEFINITION

The project stated earlier proposes a solution to the existing tedious system of physically maintain written records of all details regarding the books and students along with other information relating to the daily library accounts. The primary problem with the existing system is the issues concerning the record maintenance wherein the librarian has to keep manual registers to maintain a record of the books being borrowed by the students and ensuring their return to the library on a timely basis. Keeping in mind the above stated issues we have incorporated the below stated features in our application.

I. Features:

- Add book/student details into the database.
- Delete book/student details from the database.
- Update the number of book copies in the database.
- Add records upon issue of a book.
- Record issue date, return date, and fine.
- Keep count of number of books borrowed by each student
- Search student/book details.

II. Software Requirements:

- Operating System: Windows 10
- Database: MySQL
- Connector: MySQL JDBC connector
- Programming Language: Java (Swing)
- Development Tool: NetBeans IDE
- Server: WAMP

SYSTEM DESIGN & IMPLEMENTATION

The above mentioned features have been implemented in our project under these five modules:

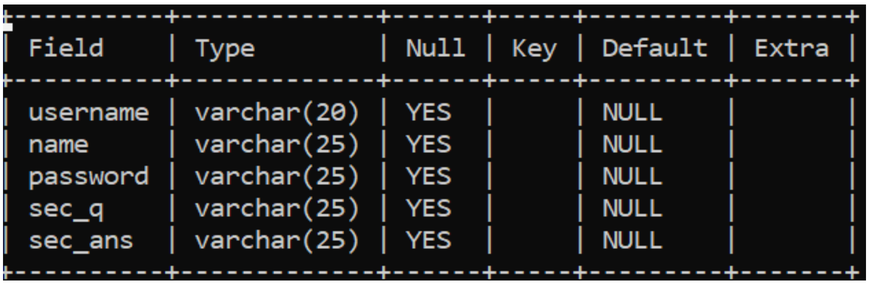
- Book Module
- Student Module
- Issue/Return Module
- Search Module
- Reports Module

Operations such as adding book details, student details, searching, issuing and returning of books are implemented in these modules. Some constraints also have been added so as to avoid repeated entry of the same students or books including the limitation of issuing only three books per student and no issuing of books whose copies are zero. These have been achieved by maintain a copies filed in the book table and a count field in the student table. Every time a student is issued a copy, the count is incremented in the student table and the copies are decremented in the book table. Also, copies of the book already stored in the database can be updated and a record of those can be seen in the book details frame. A fine field s also added in the database wherein if the student fails to return the book with thirty days a fine of hundred rupees is added to his/her account.

These operations along with the login and signup frame have been implemented by creating tables in MySQL database whose connection to the Java application has been established using the WAMP server and JDBC (Java Database Connectivity) by importing the sql package in Java.

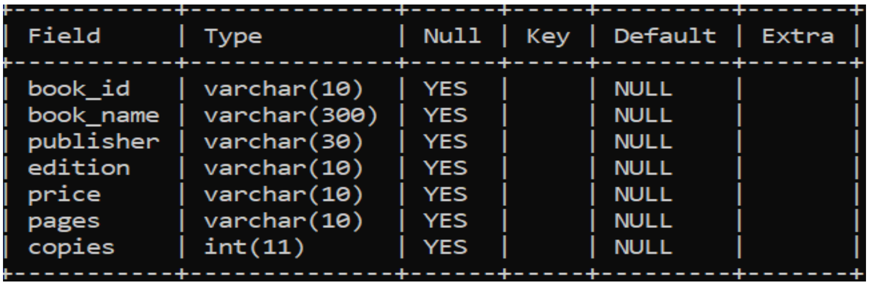
The tables created in MySQL are as follows:

- Account Table:



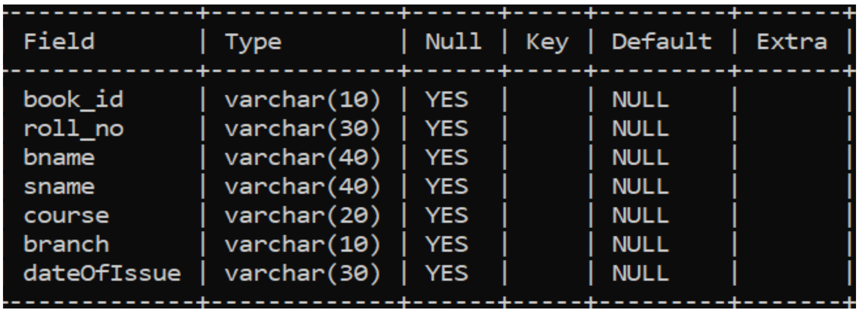
```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | YES  |     | NULL    |       |
| name     | varchar(25) | YES  |     | NULL    |       |
| password | varchar(25) | YES  |     | NULL    |       |
| sec_q    | varchar(25) | YES  |     | NULL    |       |
| sec_ans  | varchar(25) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

- Book Table:



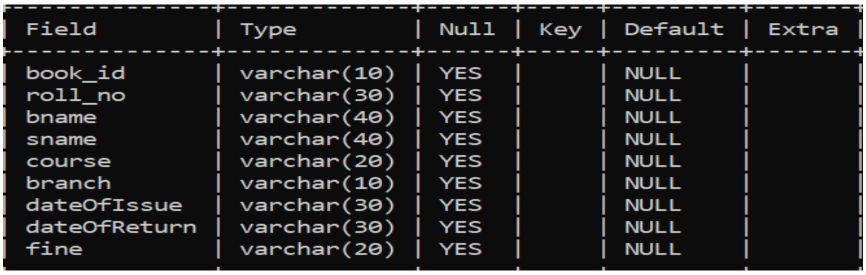
```
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| book_id   | varchar(10)  | YES  |     | NULL    |       |
| book_name | varchar(300) | YES  |     | NULL    |       |
| publisher | varchar(30)  | YES  |     | NULL    |       |
| edition   | varchar(10)  | YES  |     | NULL    |       |
| price     | varchar(10)  | YES  |     | NULL    |       |
| pages     | varchar(10)  | YES  |     | NULL    |       |
| copies    | int(11)      | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

- Issue Book Table:



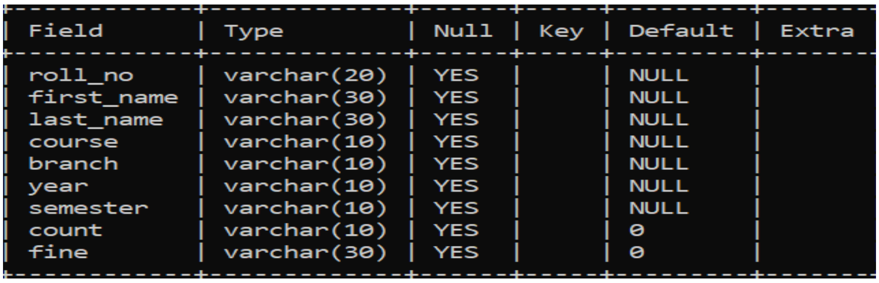
```
-------------+-------------+------+-----+---------+-------+
 Field       | Type        | Null | Key | Default | Extra |
-------------+-------------+------+-----+---------+-------+
 book_id     | varchar(10) | YES  |     | NULL    |       |
 roll_no     | varchar(30) | YES  |     | NULL    |       |
 bname       | varchar(40) | YES  |     | NULL    |       |
 sname       | varchar(40) | YES  |     | NULL    |       |
 course      | varchar(20) | YES  |     | NULL    |       |
 branch      | varchar(10) | YES  |     | NULL    |       |
 dateOfIssue | varchar(30) | YES  |     | NULL    |       |
-------------+-------------+------+-----+---------+-------+
```

- Return Book Table:



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| book_id      | varchar(10) | YES  |     | NULL    |       |
| roll_no      | varchar(30) | YES  |     | NULL    |       |
| bname        | varchar(40) | YES  |     | NULL    |       |
| sname        | varchar(40) | YES  |     | NULL    |       |
| course       | varchar(20) | YES  |     | NULL    |       |
| branch       | varchar(10) | YES  |     | NULL    |       |
| dateOfIssue  | varchar(30) | YES  |     | NULL    |       |
| dateOfReturn | varchar(30) | YES  |     | NULL    |       |
| fine         | varchar(20) | YES  |     | NULL    |       |
```

- Student Table:



```
+------------+-------------+------+-----+---------+-------
| Field      | Type        | Null | Key | Default | Extra
+------------+-------------+------+-----+---------+-------
| roll_no    | varchar(20) | YES  |     | NULL    |
| first_name | varchar(30) | YES  |     | NULL    |
| last_name  | varchar(30) | YES  |     | NULL    |
| course     | varchar(10) | YES  |     | NULL    |
| branch     | varchar(10) | YES  |     | NULL    |
| year       | varchar(10) | YES  |     | NULL    |
| semester   | varchar(10) | YES  |     | NULL    |
| count      | varchar(10) | YES  |     | 0       |
| fine       | varchar(30) | YES  |     | 0       |
+------------+-------------+------+-----+---------+-------
```

These tables as mentioned earlier are connected to the Java application using JDBC by application of the following five steps:

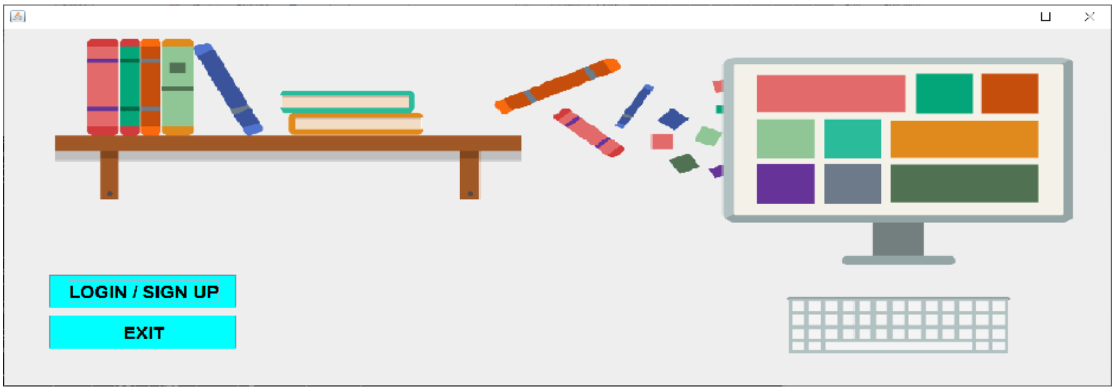
- Step 1: Registering the Driver class for MySQL database using Class.forName(“com.mysql.jdbc.Driver”) which causes the JVM to load the desired driver implementation into memory so that the JDBC requests can be fulfilled.
- Step 2: Creating connection using DriverManager.getConnection(“jdbc:mysql://localhost:3308/projectLib”, “root”, “”) which establishes the connection with the database using the URL string provided.
- Step 3: Create statement using the createStatement() method of Connection interface that will help us in executing the queries.
- Step 4: Execute queries using the executeQuery() method or executeUpdate() method of the PreparedStatement interface.
- Step 5: Close the established connection using close() method.

After establishing the connection we moved on to create the User Interface. This has been implemented completely using the Java Swing concepts by extending the JFrame class. We created frames and panels that act like containers and then created the required labels, buttons, text fields, password fields, menu items, image icons, tables and scroll panes which were then added to the panels on the frame.

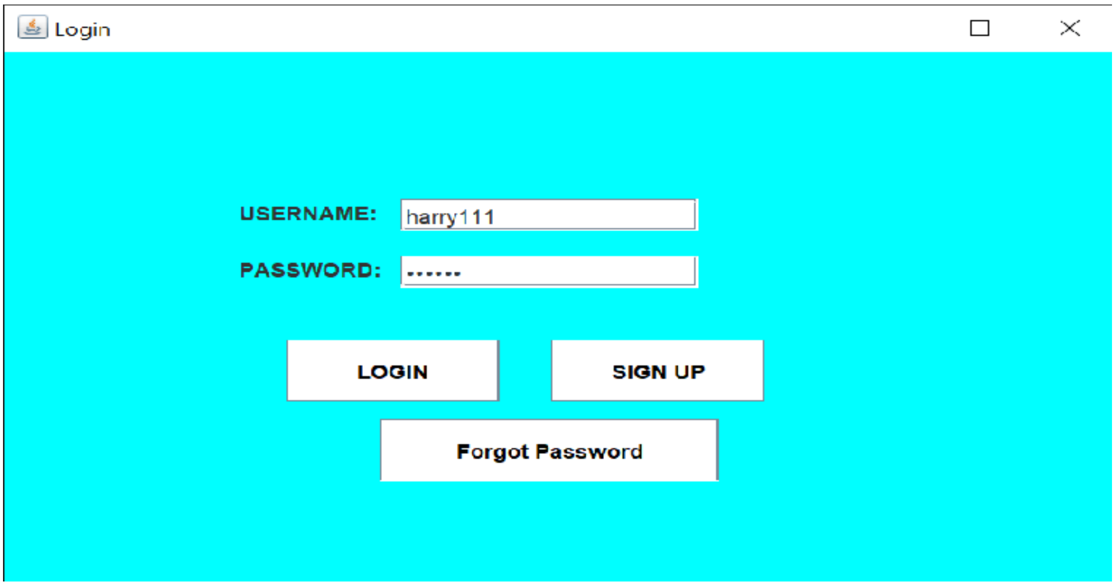
All this frame coding was done by importing the swing and awt (abstract window toolkit) packages in java. Apart from this, different events have to be performed on clicking the buttons or menu items, which has been achieved using the in-built ActionListener Interface and MouseMotionListener Class present in the awt.event package in Java.

RESULTS

- Start Frame:



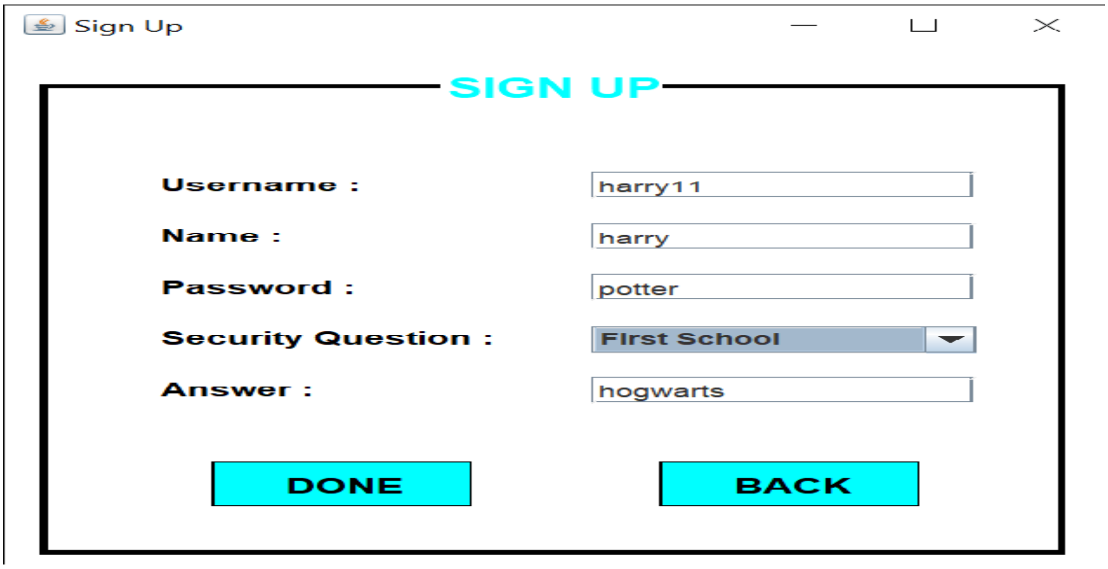
- Login Frame:



USERNAME: harry111

PASSWORD: ......

- Sign Up Frame:



Username : harry11

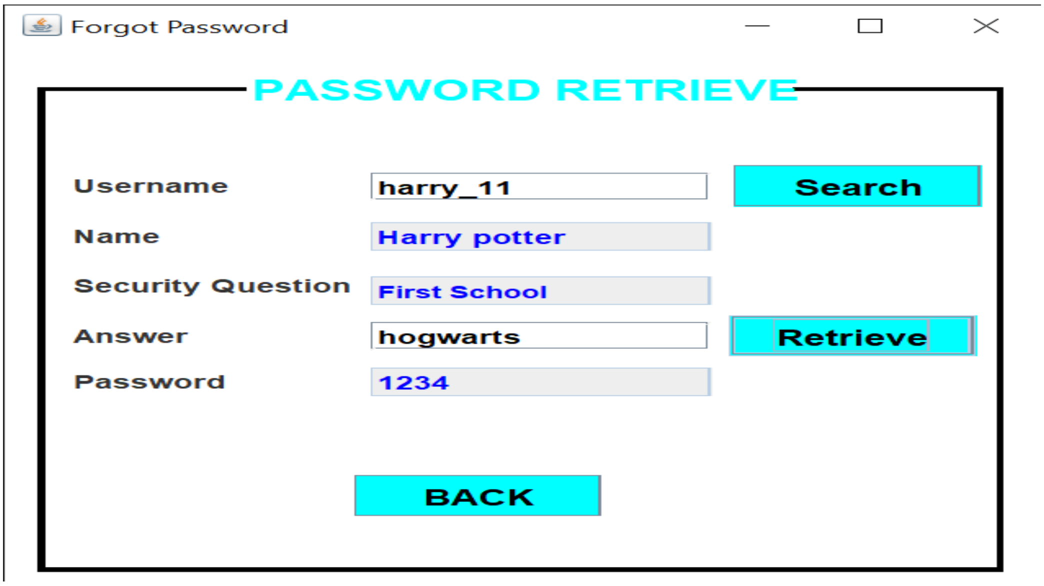
Name : harry

Password : potter

Security Question : First School

Answer : hogwarts

- Forgot Password Frame:



Forgot Password

PASSWORD RETRIEVE

Username: harry_11

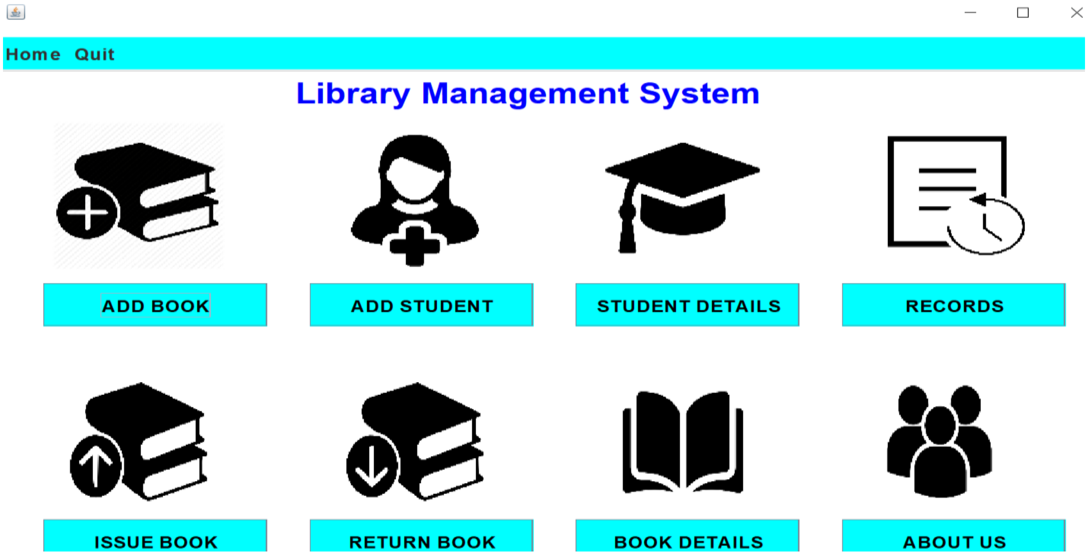
Name: Harry potter

Security Question: First School

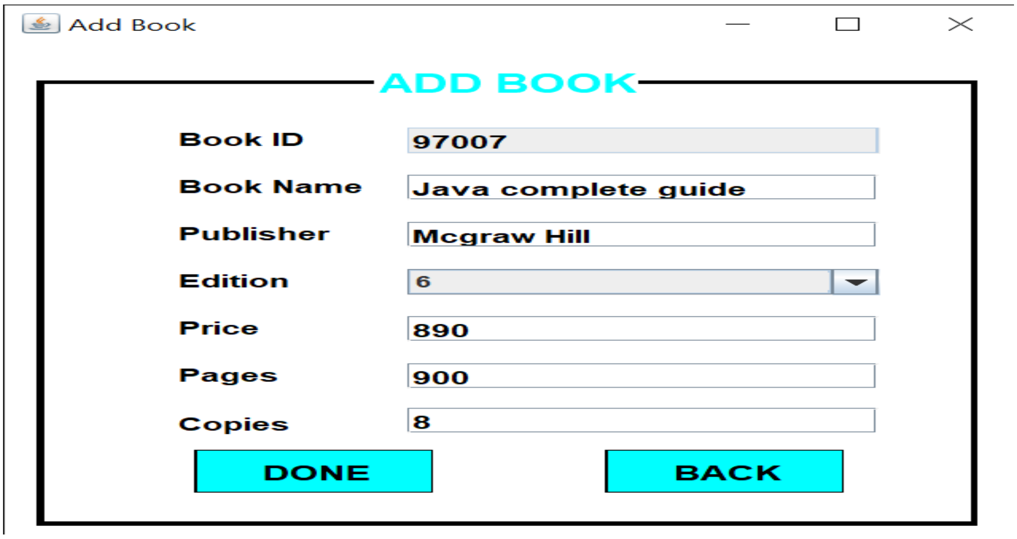
Answer: hogwarts

Password: 1234

- Home Frame:



- Add Book Frame:



Add Book

ADD BOOK

Book ID: 97007

Book Name: Java complete guide

Publisher: Mcgraw Hill

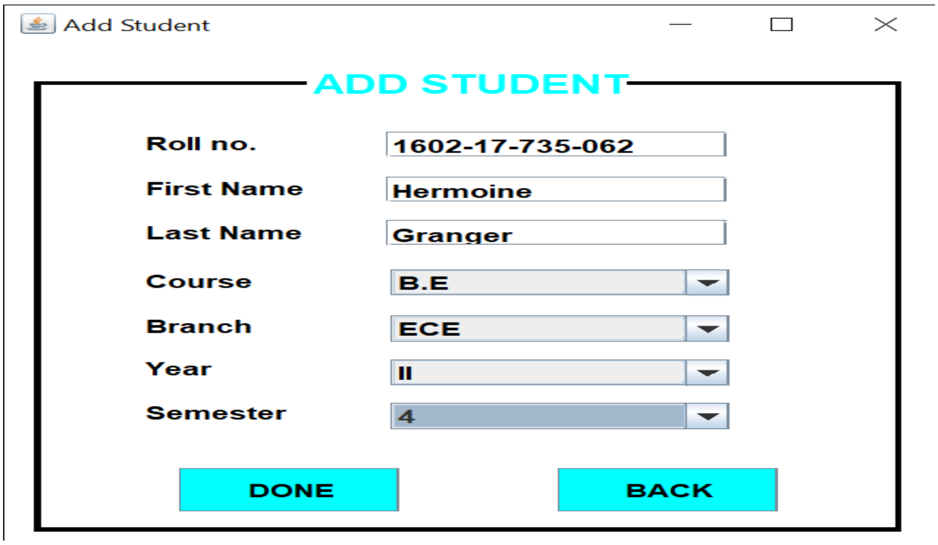
Edition: 6

Price: 890

Pages: 900

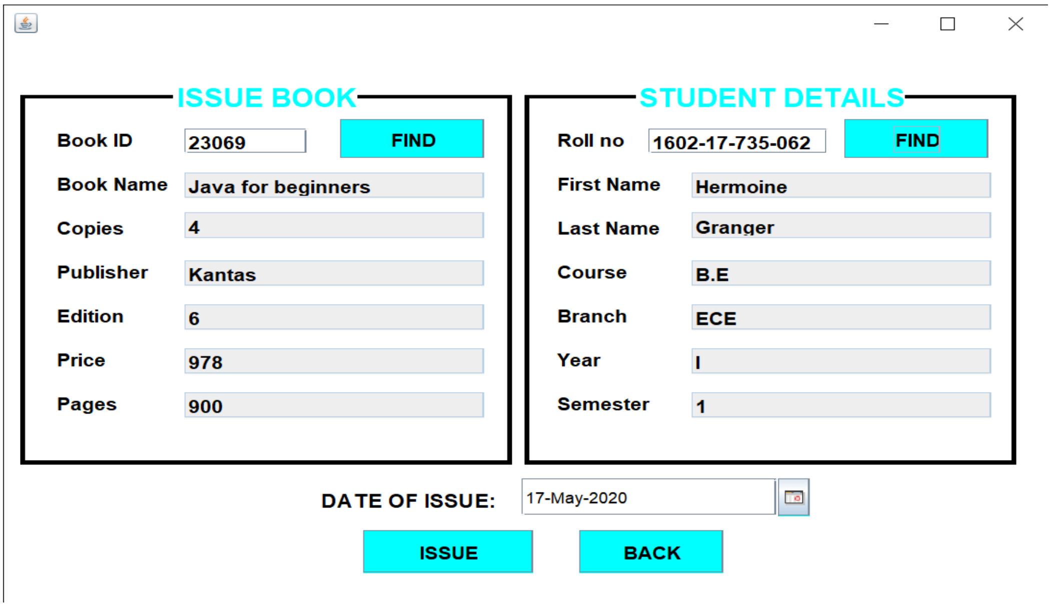
Copies: 8

- Add Student Frame:



Roll no.	1602-17-735-062
First Name	Hermoine
Last Name	Granger
Course	B.E
Branch	ECE
Year	II
Semester	4

- Issue Book Frame:

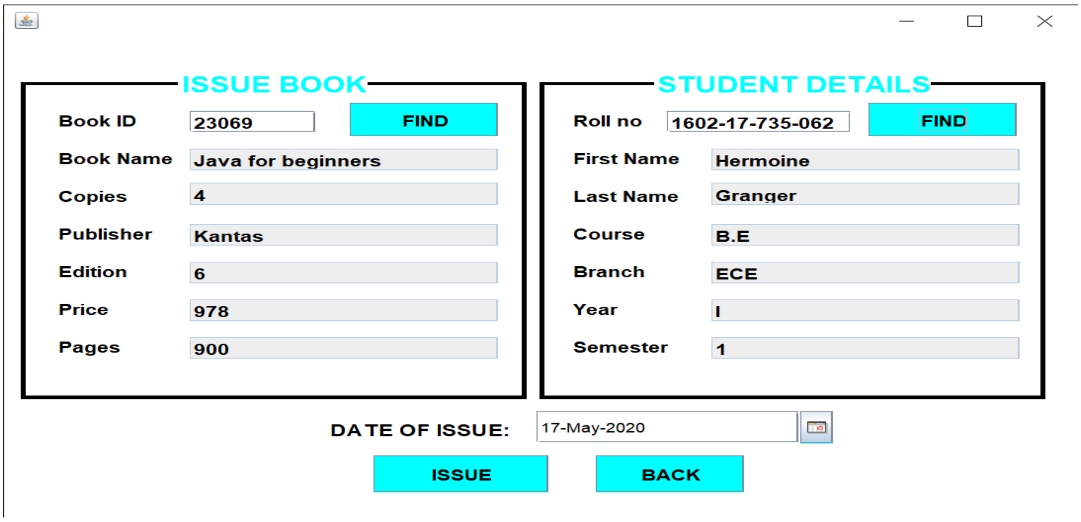


Book ID	23069
Book Name	Java for beginners
Copies	4
Publisher	Kantas
Edition	6
Price	978
Pages	900

Roll no	1602-17-735-062
First Name	Hermoine
Last Name	Granger
Course	B.E
Branch	ECE
Year	I
Semester	1

DATE OF ISSUE: 17-May-2020

- Return Book Frame:

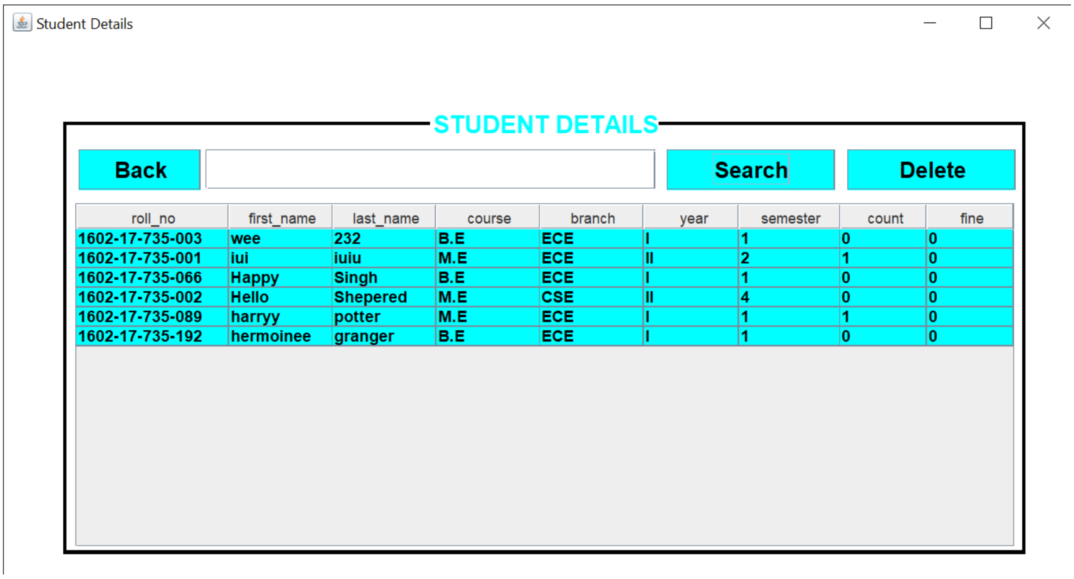


Book ID	23069
Book Name	Java for beginners
Copies	4
Publisher	Kantas
Edition	6
Price	978
Pages	900

Roll no	1602-17-735-062
First Name	Hermoine
Last Name	Granger
Course	B.E
Branch	ECE
Year	I
Semester	1

DATE OF ISSUE: 17-May-2020

- Student Details Frame:

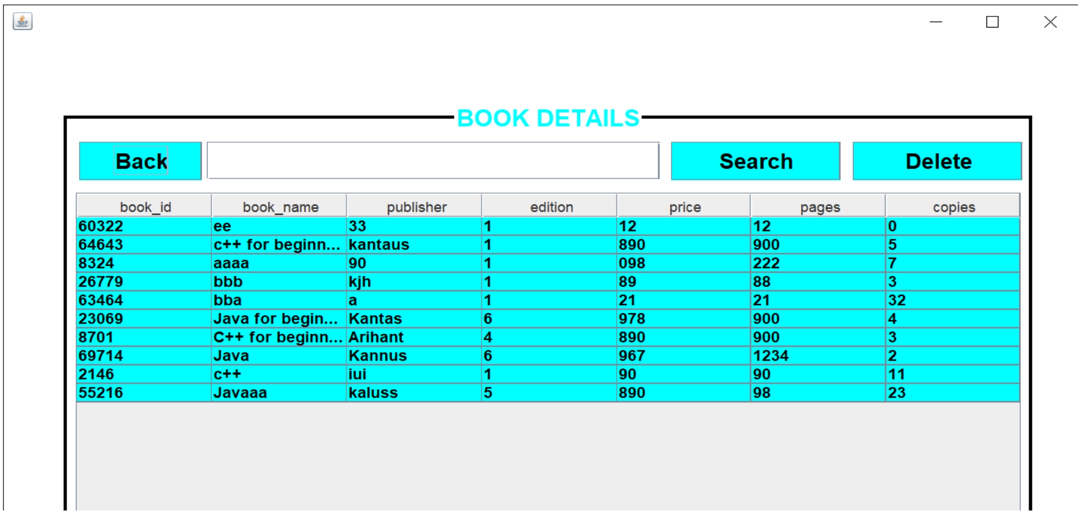


Student Details

STUDENT DETAILS

roll_no	first_name	last_name	course	branch	year	semester	count	fine
1602-17-735-003	wee	232	B.E	ECE	I	1	0	0
1602-17-735-001	iui	iuiu	M.E	ECE	II	2	1	0
1602-17-735-066	Happy	Singh	B.E	ECE	I	1	0	0
1602-17-735-002	Hello	Shepered	M.E	CSE	II	4	0	0
1602-17-735-089	harryy	potter	M.E	ECE	I	1	1	0
1602-17-735-192	hermoinee	granger	B.E	ECE	I	1	0	0

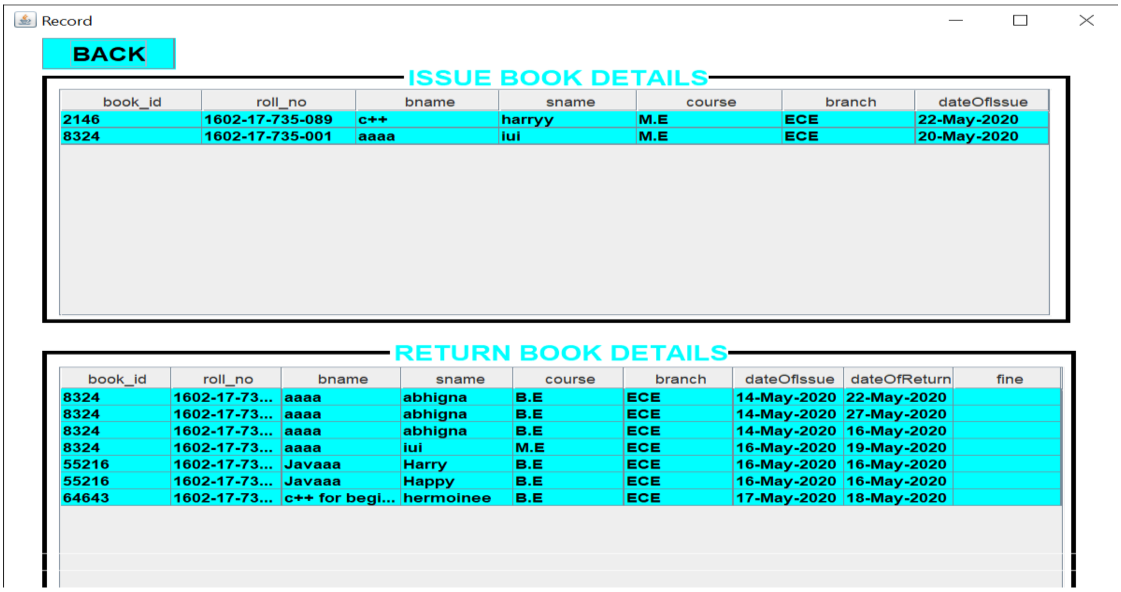
- Book Details Frame:



BOOK DETAILS

book_id	book_name	publisher	edition	price	pages	copies
60322	ee	33	1	12	12	0
64643	c++ for beginn...	kantaus	1	890	900	5
8324	aaaa	90	1	098	222	7
26779	bbb	kjh	1	89	88	3
63464	bba	a	1	21	21	32
23069	Java for begin...	Kantas	6	978	900	4
8701	C++ for beginn...	Arihant	4	890	900	3
69714	Java	Kannus	6	967	1234	2
2146	c++	iui	1	90	90	11
55216	Javaaa	kaluss	5	890	98	23

- Records Frame:



Record

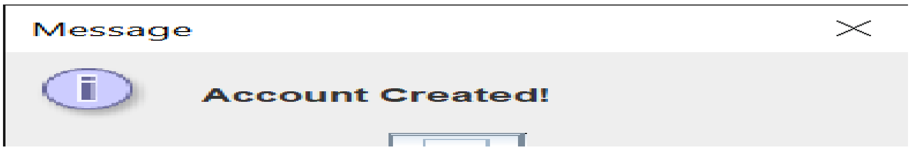
ISSUE BOOK DETAILS

book_id	roll_no	bname	sname	course	branch	dateOfIssue
2146	1602-17-735-089	c++	harryy	M.E	ECE	22-May-2020
8324	1602-17-735-001	aaaa	iui	M.E	ECE	20-May-2020

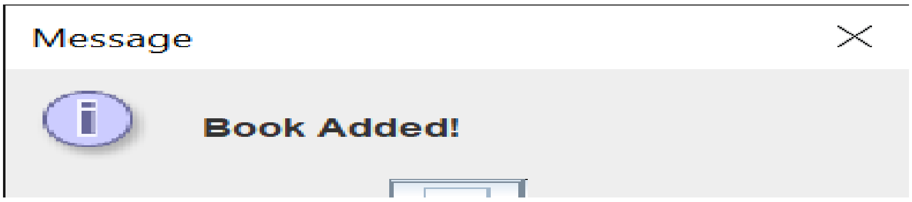
RETURN BOOK DETAILS

book_id	roll_no	bname	sname	course	branch	dateOfIssue	dateOfReturn	fine
8324	1602-17-73...	aaaa	abhigna	B.E	ECE	14-May-2020	22-May-2020	
8324	1602-17-73...	aaaa	abhigna	B.E	ECE	14-May-2020	27-May-2020	
8324	1602-17-73...	aaaa	abhigna	B.E	ECE	14-May-2020	16-May-2020	
8324	1602-17-73...	aaaa	iui	M.E	ECE	16-May-2020	19-May-2020	
55216	1602-17-73...	Javaaa	Harry	B.E	ECE	16-May-2020	16-May-2020	
55216	1602-17-73...	Javaaa	Happy	B.E	ECE	16-May-2020	16-May-2020	
64643	1602-17-73...	c++ for begi...	hermoinee	B.E	ECE	17-May-2020	18-May-2020	

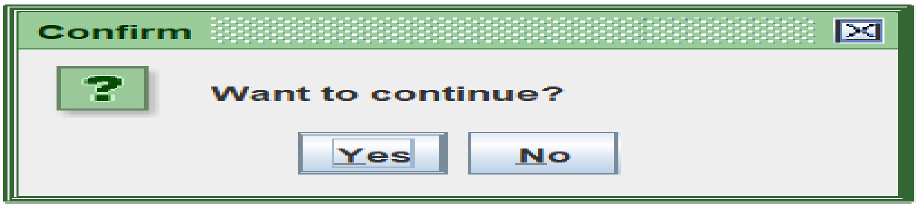
The above are the final outputs of our project. In addition to these we have also generated relevant pop-up messages for every operation performed using showMessageDialog() and showConfirmDialog() methods of the JOptionPane class.



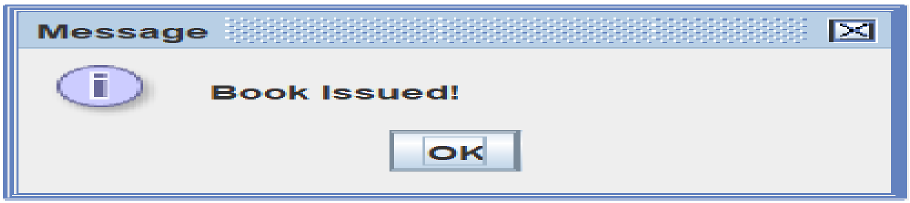
Account Created!



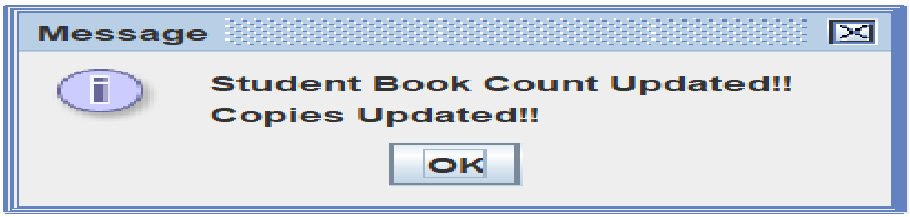
Book Added!



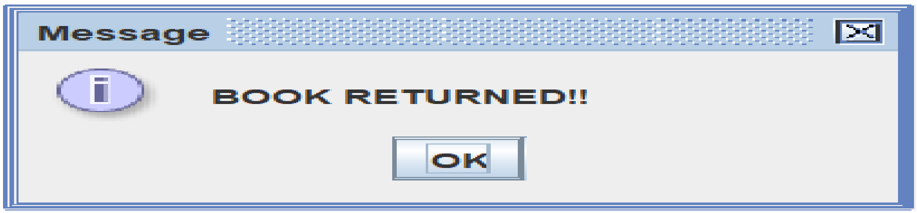
Want to continue?



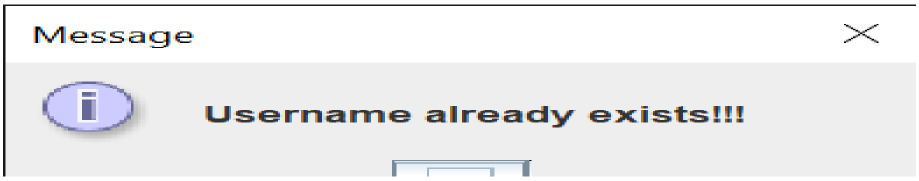
Book Issued!



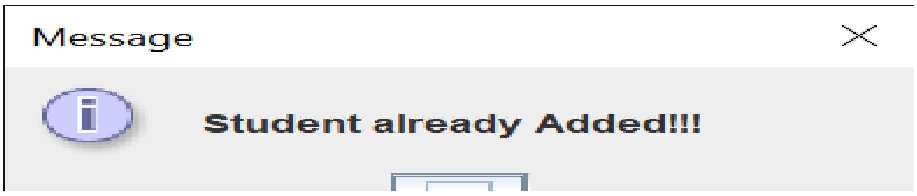
Student Book Count Updated!!
Copies Updated!!



BOOK RETURNED!!



Username already exists!!!



Student already Added!!!

FUTURE ENHANCEMENTS

More advanced features can be added to this application to make it more effective and that which improve its functionality. Like for example, a request module can be added wherein the students can make requests to the admin for including certain books and a book segregation system can be created where the books are sorted according to the subject that they are based on instead of displaying all the books randomly. Few such additions can be made to improve the usefulness of the application and to increase the purposes it can serve.

CONCLUSION

Our project, the Library Management System, therefore intends to solve the problems related to the management of books and student details in a library. With the help of this CRUD application that we created, the user can keep track of books and its whereabouts. All records of issuing and returning of the books are stored in the database and can be viewed anytime the user wishes. We hence believe that this application will serve the purpose of easing the day-to-day library transactions and ensure a hassle free record maintenance of all relevant details.

REFERENCES

- https://www.javatpoint.com/java-swing
- https://www.oracle.com/technical-resources/articles/javase/new-tech.html
- https://www.tutorialspoint.com/jdbc/index.htm