

How To Create Library Management System Project in Java?

In this modern era of the internet, almost all of us rely on web-based applications from small to big tasks. Well, Library management system is one of the most popular use-cases considered by the [professionals](#) while [building applications in Java](#). In this article, I will show you how to create a library management system project in Java.

The following will be the sequence of topics for this article:

1. [What is Java?](#)
2. [What is MySQL?](#)
3. [Library Management System Project](#)
 - - [System Requirements](#)
 - [Tables Considered](#)
 - [Project Code](#)
 - [Output](#)

Let's get started.

What is Java?

Java is an [object-oriented language](#) similar to [C++](#), but with advanced and simplified features. Java is **free to access** and can **run on all platforms**.

The features of Java are as follows:

- **Simple:** Java has made life easier by removing all the complexities such as pointers, operator overloading as you see in C++ or any other programming language.
- **Object-oriented:** Everything is considered to be an “**object**” which possess some state, behavior and all the operations are performed using these objects.
- **Secured:** All the code is converted in **bytecode** after compilation, which is not readable by a human. and java does not use an explicit pointer and run the programs inside the sandbox to prevent any activities from untrusted sources. It enables to develop virus-free, tamper-free systems/applications.

What is MySQL?

MySQL is an open-source relational database management system that works on many platforms. It provides multi-user access to support many storage engines and is backed by Oracle. So, you can buy a commercial license version from Oracle to get premium support services.

The features of MySQL are as follows:

- **Ease of Management** – The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.
- **Robust Transactional Support** – Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-version support.
- **Comprehensive Application Development** – MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, functions, views and many more for application development. Refer to [RDS Tutorial](#) to understand Amazon's RDBMS.

Library Management System Project in Java

Library Management System is one of the most popular projects which is created using Java. So, in this article, I will show you how to create this project using the following system requirements.

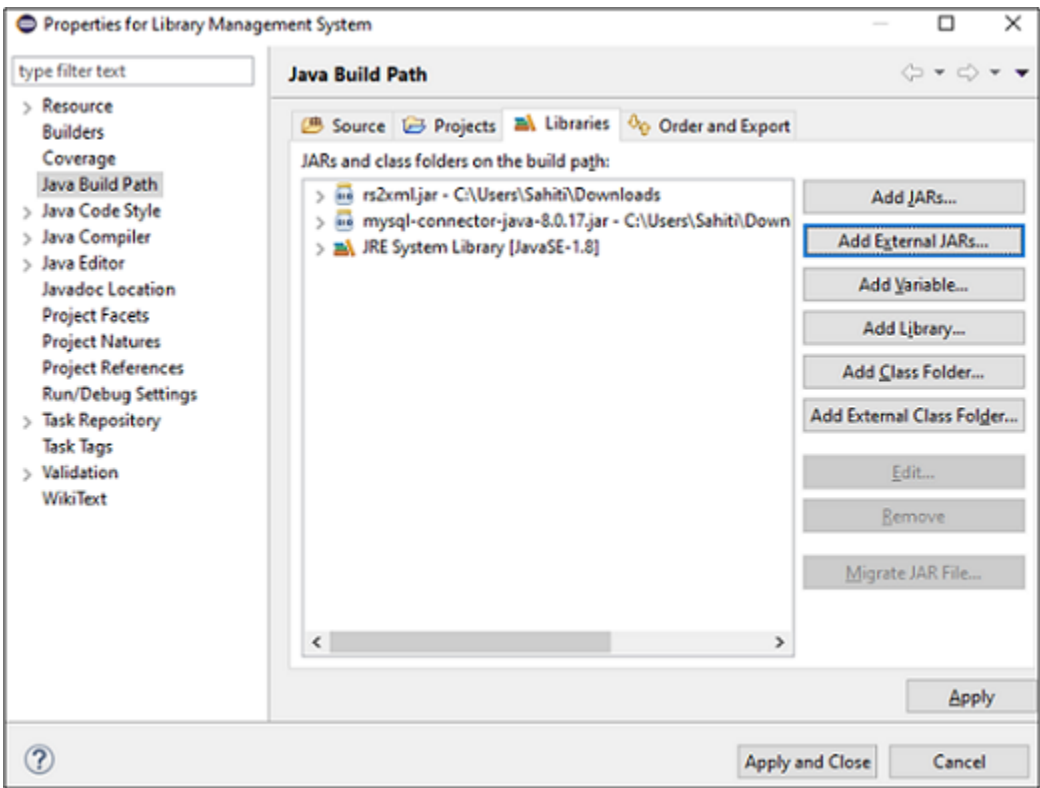
System Requirements

To execute the below project, you will need the following business requirements:

- [MySQL Community Server](#)
- MySQL JDBC Connector
- [Java](#)
- [Eclipse IDE](#)
- rs2xml.jar

The rs2xml jar is used to display the data in a table format. So, once you create a project in Eclipse IDE, you have to import the rs2xml jar and [JDBC connector JAR](#) into the project.

To do that, **right-click on the project**, choose **Build Path -> Configure Build Path**. In the dialog box, which opens up, choose **Add External JARs**, and add the JAR files. Once added, click on **Apply and Close**. Refer below.



Tables Considered

Now, for this particular project, I have considered three tables, which are:

- **Users** -> This table consists of the columns {UID, Username, Password, Admin}
- **Books** -> The book's table consists of the columns {BID, Book name, Price, Genre}
- **Issue** -> This table consists of the columns {IID, UID, BID, IssueDate, Period, ReturnDate, Fine}

Alright, so now that the Initial set is done, and I have told you the schema of tables, let us get started.

Library Management System Project Code

For your better understanding, I have divided the code into the following functions and I will be explaining you function-wise:

- [Login](#)
- [Connect](#)
- [Create/ Reset](#)
- [User Menu](#)
- [Admin Menu](#)

Also, to create a GUI, I will be using Swing. Swing is a library or a set of program components used to create graphical user interface components such as scroll bars, buttons, dialog boxes, etc.

Before I discuss the code of functions with you, let me show you the code for the main class and the libraries to be imported:

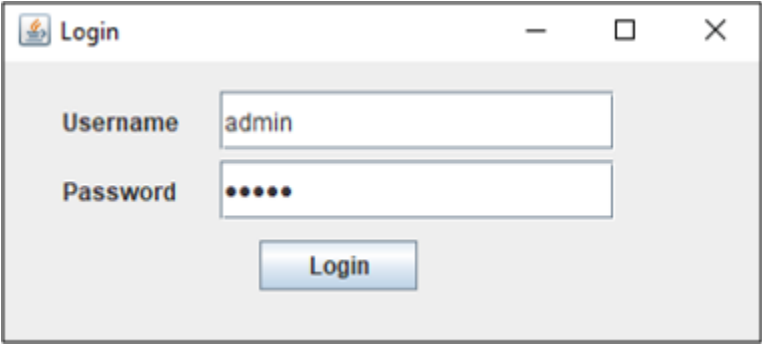
```
1      import java.awt.event.ActionEvent;
2      import java.awt.event.ActionListener;
3          import java.sql.*;
4          import java.text.DateFormat;
5          import java.text.ParseException;
6          import java.text.SimpleDateFormat;
7          import java.util.ArrayList;
8          import java.util.Date;
9          import java.util.Locale;
10         import java.util.concurrent.TimeUnit;
11
12         import javax.swing.*;
13         import net.proteanit.sql.DbUtils;
14
15         public class main {
16
17             public static class ex{
18                 public static int days=0;
19             }
20
21             public static void main(String[] args) {
22
23                 login();
24                 //create();
```

```
21         }
22
23
24
25
```

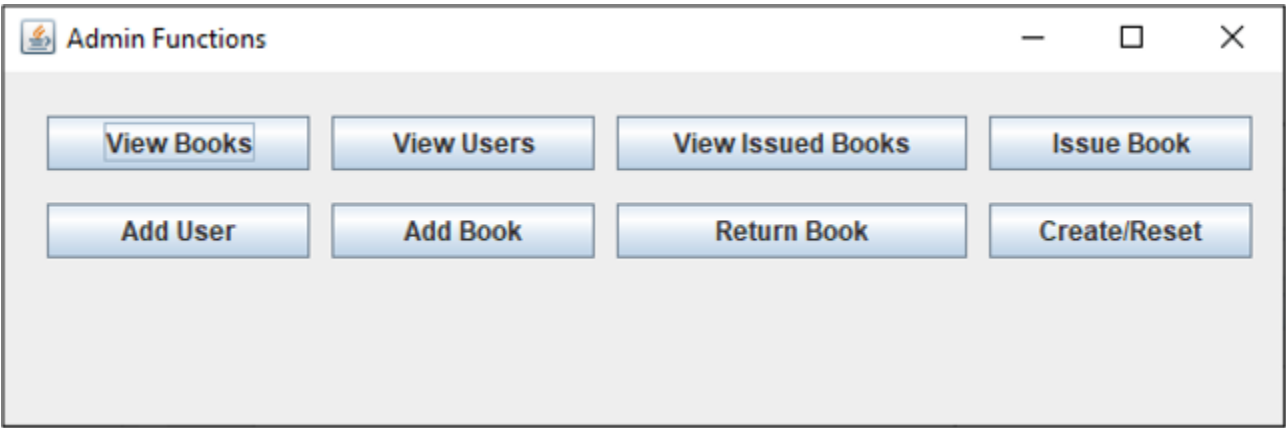
Now in this article on Library Management System in Java, let us understand the code of all the above functions.

Login

I have created this function to enable the user and the admin login. So, initially when a user logs in for the first time, that user will be an admin by default, and the username and password will be {admin, admin}. Refer below.



For this schema, I have considered only one admin. So, once a user logs in as an admin, he or she will be redirected to the admin menu as below. I will discuss the functions of the admin in the [admin menu](#) section.



Coming back to the login functions, refer to the below code:

```
1         public static void login() {
2
3             JFrame f=new JFrame("Login");//creating instance of JFrame
4                 JLabel l1,l2;
5                 l1=new JLabel("Username"); //Create label Username
6                 l1.setBounds(30,15, 100,30); //x axis, y axis, width, height
7
8                 l2=new JLabel("Password"); //Create label Password
9                 l2.setBounds(30,50, 100,30);
10
11                 JTextField F_user = new JTextField(); //Create text field for username
12                 F_user.setBounds(110, 15, 200, 30);
13
14                 JPasswordField F_pass=new JPasswordField(); //Create text field for password
15                 F_pass.setBounds(110, 50, 200, 30);
16
17                 JButton login_but=new JButton("Login");//creating instance of JButton for Login Button
18                 login_but.setBounds(130,90,80,25);//Dimensions for button
19                 login_but.addActionListener(new ActionListener() { //Perform action
20
21                     public void actionPerformed(ActionEvent e){
22
23                         String username = F_user.getText(); //Store username entered by the user in the variable
24                         String password = F_pass.getText(); //Store password entered by the user in the variable
25
26                         if(username.equals("")) //If username is null
27                         {
28                             JOptionPane.showMessageDialog(null,"Please enter username"); //Display dialog box with
29                         }
30                         else if(password.equals("")) //If password is null
31                         {
32                             JOptionPane.showMessageDialog(null,"Please enter password"); //Display dialog box with
33                         }
34                         else{ //If both the fields are present then to login the user, check wether the user exists
35                             //System.out.println("Login connect");
36                             Connection connection=connect(); //Connect to the database
37                             try
38                             {
39                                 Statement stmt = connection.createStatement();
40                                 stmt.executeUpdate("USE LIBRARY"); //Use the database with the name "Library"
41                                 String st = ("SELECT * FROM USERS WHERE USERNAME='"+username+"' AND PASSWORD='"+password+"'"); //Retrieve data
42                                 from users
43                                 ResultSet rs = stmt.executeQuery(st); //Execute query
```

```
38         if(rs.next()==false) { //Move pointer below
39             System.out.print("No user");
40             JOptionPane.showMessageDialog(null,"Wrong Username/Password!"); //Display M
41         }
42         else {
43             f.dispose();
44             rs.beforeFirst(); //Move the pointer above
45             while(rs.next())
46             {
47                 String admin = rs.getString("ADMIN"); //user is admin
48                 //System.out.println(admin);
49                 String UID = rs.getString("UID"); //Get user ID of the user
50                 if(admin.equals("1")) { //If boolean value 1
51                     admin_menu(); //redirect to admin menu
52                 }
53                 else{
54                     user_menu(UID); //redirect to user menu for that user ID
55                 }
56             }
57             catch (Exception ex) {
58                 ex.printStackTrace();
59             }
60         }
61     });
62
63     f.add(F_pass); //add password
64     f.add(login_but);//adding button in JFrame
65     f.add(F_user); //add user
66     f.add(l1); // add label1 i.e. for username
67     f.add(l2); // add label2 i.e. for password
68
69     f.setSize(400,180);//400 width and 500 height
70     f.setLayout(null);//using no layout managers
71     f.setVisible(true);//making the frame visible
72     f.setLocationRelativeTo(null);
73
74
75
76
77
78
79
80
81
82
83
84
```

Connect

The connect function is used to connect the database to the [GUI](#). So, to do that, I have mentioned the below code:

```
1
2         public static Connection connect()
3         {
4             try {
5                 Class.forName("com.mysql.cj.jdbc.Driver");
6                 //System.out.println("Loaded driver");
7                 Connection con = DriverManager.getConnection("jdbc:mysql://localhost/mysql?user=root&password=edureka");
8                 //System.out.println("Connected to MySQL");
9                 return con;
10            }
11            catch (Exception ex) {
12                ex.printStackTrace();
13            }
14            return null;
15        }
```

In the above function, we are connecting our **MySQL database** with the **username “root”** and **password “edureka”** to our application. Now, once the application is connected to the database, our next step is to create or reset the database. So, next in this article on Library Management System Project in Java, let us discuss the Create function..

Create

The create function is used to create the database, tables and add data into these tables. So, to do that, [SQL statements](#) will be used as below.

```
1         public static void create() {
2             try {
```

```

2         Connection connection=connect();
3         ResultSet resultSet = connection.getMetaData().getCatalogs();
4         //iterate each catalog in the ResultSet
5         while (resultSet.next()) {
6             // Get the database name, which is at position 1
7             String databaseName = resultSet.getString(1);
8             if(databaseName.equals("library")) {
9                 //System.out.print("yes");
10                Statement stmt = connection.createStatement();
11                //Drop database if it pre-exists to reset the complete database
12                String sql = "DROP DATABASE library";
13                stmt.executeUpdate(sql);
14            }
15        }
16        Statement stmt = connection.createStatement();
17
18        String sql = "CREATE DATABASE LIBRARY"; //Create Database
19        stmt.executeUpdate(sql);
20        stmt.executeUpdate("USE LIBRARY"); //Use Database
21        //Create Users Table
22        String sql1 = "CREATE TABLE USERS(UID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, USERNAME VARCHAR(30), PASSWORD VARCHAR(30), ADMIN BOOLEAN)";
23        stmt.executeUpdate(sql1);
24        //Insert into users table
25        stmt.executeUpdate("INSERT INTO USERS(USERNAME, PASSWORD, ADMIN) VALUES('admin','admin','admin')");
26        //Create Books table
27        stmt.executeUpdate("CREATE TABLE BOOKS(BID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, BNAME VARCHAR(50), PRICE INT)");
28        //Create Issued Table
29        stmt.executeUpdate("CREATE TABLE ISSUED(IID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, UID INT, BID INT, RETURN_DATE VARCHAR(20), PERIOD INT, FINE INT)");
30        //Insert into books table
31        stmt.executeUpdate("INSERT INTO BOOKS(BNAME, GENRE, PRICE) VALUES ('War and Peace', 'Mystery', 200), ('The Perfect Murder', 'Mystery', 150), ('Accidental Presidents', 'Biography', 250), ('The Wicked King', 300)");
32
33        resultSet.close();
34    }
35    catch (Exception ex) {
36        ex.printStackTrace();
37    }
38
39

```

Now, that we have created the database, connected with GUI and enables the login function, next in this article on Library Management System Project in Java, let us now discuss the functions of the User Menu.

User Menu

The User Menu is designed to show details of all the books present in the library and the books issued by the user.

```

1         public static void user_menu(String UID) {
2
3
4             JFrame f=new JFrame("User Functions"); //Give dialog box name as User functions
5             //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Exit user menu on closing the dial
6             JButton view_but=new JButton("View Books");//creating instance of JButton
7             view_but.setBounds(20,20,120,25);//x axis, y axis, width, height
8             view_but.addActionListener(new ActionListener() {
9                 public void actionPerformed(ActionEvent e){
10
11                     JFrame f = new JFrame("Books Available"); //View books stored in database
12                     //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13
14                     Connection connection = connect();
15                     String sql="select * from BOOKS"; //Retreive data from database
16                     try {
17                         Statement stmt = connection.createStatement(); //connect to database
18                         stmt.executeUpdate("USE LIBRARY"); // use librabry
19                         stmt=connection.createStatement();
20                         ResultSet rs=stmt.executeQuery(sql);
21                         JTable book_list= new JTable(); //show data in table format
22                         book_list.setModel(DbUtils.resultSetToTableModel(rs));
23
24                         JScrollPane scrollPane = new JScrollPane(book_list); //enable scroll bar
25
26                         f.add(scrollPane); //add scroll bar
27                         f.setSize(800, 400); //set dimensions of view books frame
28                         f.setVisible(true);
29                         f.setLocationRelativeTo(null);
30                     } catch (SQLException e1) {
31                         // TODO Auto-generated catch block
32                         JOptionPane.showMessageDialog(null, e1);
33                     }
34
35                 }
36
37             }
38
39

```

```

33                                     );
34
35         JButton my_book=new JButton("My Books");//creating instance of JButton
36         my_book.setBounds(150,20,120,25);//x axis, y axis, width, height
37         my_book.addActionListener(new ActionListener() { //Perform action
38                                     public void actionPerformed(ActionEvent e){
39
40                                     JFrame f = new JFrame("My Books"); //View books issued by user
41                                     //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
42                                     int UID_int = Integer.parseInt(UID); //Pass user ID
43
44                                     //.iid,issued.uid,issued.bid,issued.issued_date,issued.return_date,issued,
45                                     Connection connection = connect(); //connect to database
46                                     //retrieve data
47                                     String sql="select distinct issued.*,books.bname,books.genre,books.price from issued,books " + "where (
48                                     and (books.bid in (select bid from issued where issued.uid="+UID_int+"))) group by iid"
49                                     String sql1 = "select bid from issued where uid="+UID_int;
50                                     try {
51                                     Statement stmt = connection.createStatement();
52                                     //use database
53                                     stmt.executeUpdate("USE LIBRARY");
54                                     stmt=connection.createStatement();
55                                     //store in array
56                                     ArrayList books_list = new ArrayList();
57
58                                     ResultSet rs=stmt.executeQuery(sql);
59                                     JTable book_list= new JTable(); //store data in table format
60                                     book_list.setModel(DbUtils.resultSetToTableModel(rs));
61                                     //enable scroll bar
62                                     JScrollPane scrollPane = new JScrollPane(book_list);
63
64                                     f.add(scrollPane); //add scroll bar
65                                     f.setSize(800, 400); //set dimensions of my books frame
66                                     f.setVisible(true);
67                                     f.setLocationRelativeTo(null);
68                                     } catch (SQLException e1) {
69                                     // TODO Auto-generated catch block
70                                     JOptionPane.showMessageDialog(null, e1);
71                                     }
72                                     }
73
74
75                                     f.add(my_book); //add my books
76                                     f.add(view_but); // add view books
77                                     f.setSize(300,100);//400 width and 500 height
78                                     f.setLayout(null);//using no layout managers
79                                     f.setVisible(true);//making the frame visible
80                                     f.setLocationRelativeTo(null);
81                                     }
82
83
84
85
86
87
88
89
90
91
92

```

Next, in this article on Library Management System Project in Java, let us discuss the code for Admin Menu function.

Admin Menu

The Admin Menu is designed to show details of users, books, issued books, add books, return books, add user, and create or reset the database.

```

1                                     public static void admin_menu() {
2
3
4                                     JFrame f=new JFrame("Admin Functions"); //Give dialog box name as admin functions
5                                     //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //
6
7                                     JButton create_but=new JButton("Create/Reset");//creating instance of JButton to create or re
8                                     create_but.setBounds(450,60,120,25);//x axis, y axis, width, height
9                                     create_but.addActionListener(new ActionListener() { //Perform action
10                                     public void actionPerformed(ActionEvent e){

```

```

11
12         create(); //Call create function
13 JOptionPane.showMessageDialog(null,"Database Created/Reset!"); //Open a dialog box and di
14
15     }
16     });
17
18 JButton view_but=new JButton("View Books");//creating instance of JButton to view bo
19 view_but.setBounds(20,20,120,25);//x axis, y axis, width, height
20 view_but.addActionListener(new ActionListener() {
21     public void actionPerformed(ActionEvent e){
22
23         JFrame f = new JFrame("Books Available");
24         //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25
26         Connection connection = connect(); //connect to database
27         String sql="select * from BOOKS"; //select all books
28         try{
29             Statement stmt = connection.createStatement();
30             stmt.executeUpdate("USE LIBRARY"); //use database
31             stmt=connection.createStatement();
32             ResultSet rs=stmt.executeQuery(sql);
33             JTable book_list= new JTable(); //view data in table format
34             book_list.setModel(DbUtils.resultSetToTableModel(rs));
35             //mention scroll bar
36             JScrollPane scrollPane = new JScrollPane(book_list);
37
38             f.add(scrollPane); //add scrollpane
39             f.setSize(800, 400); //set size for frame
40             f.setVisible(true);
41             f.setLocationRelativeTo(null);
42         } catch (SQLException e1) {
43             // TODO Auto-generated catch block
44             JOptionPane.showMessageDialog(null, e1);
45         }
46     }
47
48 JButton users_but=new JButton("View Users");//creating instance of JButton to view u
49 users_but.setBounds(150,20,120,25);//x axis, y axis, width, height
50 users_but.addActionListener(new ActionListener() { //Perform action on click butto
51     public void actionPerformed(ActionEvent e){
52
53         JFrame f = new JFrame("Users List");
54         //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55
56         Connection connection = connect();
57         String sql="select * from users"; //retrieve all users
58         try{
59             Statement stmt = connection.createStatement();
60             stmt.executeUpdate("USE LIBRARY"); //use database
61             stmt=connection.createStatement();
62             ResultSet rs=stmt.executeQuery(sql);
63             JTable book_list= new JTable();
64             book_list.setModel(DbUtils.resultSetToTableModel(rs));
65             //mention scroll bar
66             JScrollPane scrollPane = new JScrollPane(book_list);
67
68             f.add(scrollPane); //add scrollpane
69             f.setSize(800, 400); //set size for frame
70             f.setVisible(true);
71             f.setLocationRelativeTo(null);
72         } catch (SQLException e1) {
73             // TODO Auto-generated catch block
74             JOptionPane.showMessageDialog(null, e1);
75         }
76     }
77
78 JButton issued_but=new JButton("View Issued Books");//creating instance of JButton to view the
79 issued_but.setBounds(280,20,160,25);//x axis, y axis, width, height
80 issued_but.addActionListener(new ActionListener() {
81     public void actionPerformed(ActionEvent e){
82
83         JFrame f = new JFrame("Users List");
84         //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
85
86         Connection connection = connect();
87         String sql="select * from issued";
88         try{
89             Statement stmt = connection.createStatement();

```



```

89         stmt.executeUpdate("USE LIBRARY");
90         stmt=connection.createStatement();
91         ResultSet rs=stmt.executeQuery(sql);
92         JTable book_list= new JTable();
93         book_list.setModel(DbUtils.resultSetToTableModel(rs));
94
95         JScrollPane scrollPane = new JScrollPane(book_list);
96
97         f.add(scrollPane);
98         f.setSize(800, 400);
99         f.setVisible(true);
100        f.setLocationRelativeTo(null);
101        } catch (SQLException e1) {
102            // TODO Auto-generated catch block
103            JOptionPane.showMessageDialog(null, e1);
104        }
105    }
106
107
108    JButton add_user=new JButton("Add User"); //creating instance of JButton to add user
109    add_user.setBounds(20,60,120,25); //set dimensions for button
110
111    add_user.addActionListener(new ActionListener() {
112        public void actionPerformed(ActionEvent e){
113
114            JFrame g = new JFrame("Enter User Details"); //Frame to enter user details
115            //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
116            //Create label
117            JLabel l1,l2;
118            l1=new JLabel("Username"); //label 1 for username
119            l1.setBounds(30,15, 100,30);
120
121            l2=new JLabel("Password"); //label 2 for password
122            l2.setBounds(30,50, 100,30);
123
124            //set text field for username
125            JPasswordField F_user = new JPasswordField();
126            F_user.setBounds(110, 15, 200, 30);
127
128            //set text field for password
129            JPasswordField F_pass=new JPasswordField();
130            F_pass.setBounds(110, 50, 200, 30);
131            //set radio button for admin
132            JRadioButton a1 = new JRadioButton("Admin");
133            a1.setBounds(55, 80, 200,30);
134            //set radio button for user
135            JRadioButton a2 = new JRadioButton("User");
136            a2.setBounds(130, 80, 200,30);
137            //add radio buttons
138            ButtonGroup bg=new ButtonGroup();
139            bg.add(a1);bg.add(a2);
140
141            JButton create_but=new JButton("Create");//creating instance of JButton for
142            create_but.setBounds(130,130,80,25);//x axis, y axis, width, height
143            create_but.addActionListener(new ActionListener() {
144
145                public void actionPerformed(ActionEvent e){
146
147                    String username = F_user.getText();
148                    String password = F_pass.getText();
149                    Boolean admin = false;
150
151                    if(a1.isSelected()) {
152                        admin=true;
153                    }
154
155                    Connection connection = connect();
156
157                    try {
158                        Statement stmt = connection.createStatement();
159                        stmt.executeUpdate("USE LIBRARY");
160                        stmt.executeUpdate("INSERT INTO USERS (USERNAME,PASSWORD,ADMIN) VALUES ('"+username+"','"+password+"',"+admin+")");
161                        JOptionPane.showMessageDialog(null,"User added!");
162                        g.dispose();
163                    }
164
165                    catch (SQLException e1) {
166                        // TODO Auto-generated catch block
167                        JOptionPane.showMessageDialog(null, e1);
168                    }
169                }
170            }
171        }
172    }
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

167         });
168
169
170         g.add(create_but);
171         g.add(a2);
172         g.add(a1);
173         g.add(l1);
174         g.add(l2);
175         g.add(F_user);
176         g.add(F_pass);
177         g.setSize(350,200);//400 width and 500 height
178         g.setLayout(null);//using no layout managers
179         g.setVisible(true);//making the frame visible
180         g.setLocationRelativeTo(null);
181
182     }
183     });
184
185     JButton add_book=new JButton("Add Book"); //creating instance of JButton for adding b
186     add_book.setBounds(150,60,120,25);
187
188     add_book.addActionListener(new ActionListener() {
189         public void actionPerformed(ActionEvent e){
190             //set frame wot enter book details
191             JFrame g = new JFrame("Enter Book Details");
192             //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
193             // set labels
194             JLabel l1,l2,l3;
195             l1=new JLabel("Book Name"); //lebel 1 for book name
196             l1.setBounds(30,15, 100,30);
197
198
199             l2=new JLabel("Genre"); //label 2 for genre
200             l2.setBounds(30,53, 100,30);
201
202             l3=new JLabel("Price"); //label 2 for price
203             l3.setBounds(30,90, 100,30);
204
205             //set text field for book name
206             JTextField F_bname = new JTextField();
207             F_bname.setBounds(110, 15, 200, 30);
208
209             //set text field for genre
210             JTextField F_genre=new JTextField();
211             F_genre.setBounds(110, 53, 200, 30);
212             //set text field for price
213             JTextField F_price=new JTextField();
214             F_price.setBounds(110, 90, 200, 30);
215
216
217     JButton create_but=new JButton("Submit");//creating instance of JButton to subm
218     create_but.setBounds(130,130,80,25);//x axis, y axis, width, height
219     create_but.addActionListener(new ActionListener() {
220
221         public void actionPerformed(ActionEvent e){
222             // assign the book name, genre, price
223             String bname = F_bname.getText();
224             String genre = F_genre.getText();
225             String price = F_price.getText();
226             //convert price of integer to int
227             int price_int = Integer.parseInt(price);
228
229             Connection connection = connect();
230
231             try {
232                 Statement stmt = connection.createStatement();
233                 stmt.executeUpdate("USE LIBRARY");
234                 stmt.executeUpdate("INSERT INTO BOOKS (BNAME,GENRE,PRICE) VALUES ('"+bname+"','"+genre
235                 JOptionPane.showMessageDialog(null,"Book added!");
236                 g.dispose();
237             }
238
239             catch (SQLException e1) {
240                 // TODO Auto-generated catch block
241                 JOptionPane.showMessageDialog(null, e1);
242             }
243
244         }
245     });
246
247     g.add(l3);
248     g.add(create_but);
249     g.add(l1);
250     g.add(l2);
251     g.add(F_bname);

```

```
245         g.add(F_genre);
246         g.add(F_price);
247         g.setSize(350,200);//400 width and 500 height
248         g.setLayout(null);//using no layout managers
249         g.setVisible(true);//making the frame visible
250         g.setLocationRelativeTo(null);
251     }
252     });
253
254     JButton issue_book=new JButton("Issue Book"); //creating instance of JButton to issue
255     issue_book.setBounds(450,20,120,25);
256
257     issue_book.addActionListener(new ActionListener() {
258         public void actionPerformed(ActionEvent e){
259             //enter details
260             JFrame g = new JFrame("Enter Details");
261             //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
262             //create labels
263             JLabel l1,l2,l3,l4;
264             l1=new JLabel("Book ID(BID)"); // Label 1 for Book ID
265             l1.setBounds(30,15, 100,30);
266
267             l2=new JLabel("User ID(UID)"); //Label 2 for user ID
268             l2.setBounds(30,53, 100,30);
269
270             l3=new JLabel("Period(days)"); //Label 3 for period
271             l3.setBounds(30,90, 100,30);
272
273             l4=new JLabel("Issued Date(DD-MM-YYYY)"); //Label 4 for issue date
274             l4.setBounds(30,127, 150,30);
275
276             JTextField F_bid = new JTextField();
277             F_bid.setBounds(110, 15, 200, 30);
278
279             JTextField F_uid=new JTextField();
280             F_uid.setBounds(110, 53, 200, 30);
281
282             JTextField F_period=new JTextField();
283             F_period.setBounds(110, 90, 200, 30);
284
285             JTextField F_issue=new JTextField();
286             F_issue.setBounds(180, 130, 130, 30);
287
288     JButton create_but=new JButton("Submit");//creating instance of JButton
289     create_but.setBounds(130,170,80,25);//x axis, y axis, width, height
290     create_but.addActionListener(new ActionListener() {
291
292         public void actionPerformed(ActionEvent e){
293
294             String uid = F_uid.getText();
295             String bid = F_bid.getText();
296             String period = F_period.getText();
297             String issued_date = F_issue.getText();
298
299             int period_int = Integer.parseInt(period);
300
301             Connection connection = connect();
302
303             try{
304                 Statement stmt = connection.createStatement();
305                 stmt.executeUpdate("USE LIBRARY");
306                 stmt.executeUpdate("INSERT INTO ISSUED(UID,BID,ISSUED_DATE,PERIOD) VALUES('"+uid+"','"+bid+"','"+issued_date+"','"+period_int+"')");
307                 JOptionPane.showMessageDialog(null,"Book Issued!");
308                 g.dispose();
309             }
310
311             catch (SQLException e1) {
312                 // TODO Auto-generated catch block
313                 JOptionPane.showMessageDialog(null, e1);
314             }
315
316         });
317
318         g.add(l3);
319         g.add(l4);
320         g.add(create_but);
321         g.add(l1);
322         g.add(l2);
```

```

323         g.add(F_uid);
324         g.add(F_bid);
325         g.add(F_period);
326         g.add(F_issue);
327         g.setSize(350,250);//400 width and 500 height
328         g.setLayout(null);//using no layout managers
329         g.setVisible(true);//making the frame visible
330         g.setLocationRelativeTo(null);
331     }
332     });
333
334
335     JButton return_book=new JButton("Return Book"); //creating instance of JButton to return
336     return_book.setBounds(280,60,160,25);
337
338     return_book.addActionListener(new ActionListener() {
339     public void actionPerformed(ActionEvent e){
340
341         JFrame g = new JFrame("Enter Details");
342         //g.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
343         //set labels
344         JLabel l1,l2,l3,l4;
345         l1=new JLabel("Issue ID(IID)"); //Label 1 for Issue ID
346         l1.setBounds(30,15, 100,30);
347
348         l4=new JLabel("Return Date(DD-MM-YYYY)");
349         l4.setBounds(30,50, 150,30);
350
351         JTextField F_iid = new JTextField();
352         F_iid.setBounds(110, 15, 200, 30);
353
354         JTextField F_return=new JTextField();
355         F_return.setBounds(180, 50, 130, 30);
356
357     JButton create_but=new JButton("Return");//creating instance of JButton to mention return d
358     create_but.setBounds(130,170,80,25);//x axis, y axis, width, height
359     create_but.addActionListener(new ActionListener() {
360
361     public void actionPerformed(ActionEvent e){
362
363         String iid = F_iid.getText();
364         String return_date = F_return.getText();
365
366         Connection connection = connect();
367
368         try {
369             Statement stmt = connection.createStatement();
370             stmt.executeUpdate("USE LIBRARY");
371             //Intialize datel with NULL value
372             String datel=null;
373             String date2=return_date; //Intialize date2 with return date
374
375             //select issue date
376             ResultSet rs = stmt.executeQuery("SELECT ISSUED_DATE FROM ISSUED WHERE IID="+iid);
377             while (rs.next()) {
378                 datel = rs.getString(1);
379             }
380
381             try {
382                 Date date_1=new SimpleDateFormat("dd-MM-yyyy").parse(datel);
383                 Date date_2=new SimpleDateFormat("dd-MM-yyyy").parse(date2);
384                 //subtract the dates and store in diff
385                 long diff = date_2.getTime() - date_1.getTime();
386                 //Convert diff from milliseconds to days
387                 ex.days=(int) (TimeUnit.DAYS.convert(diff, TimeUnit.MILLISECON
388             } catch (ParseException e1) {
389                 // TODO Auto-generated catch block
390                 e1.printStackTrace();
391             }
392
393             //update return date
394             stmt.executeUpdate("UPDATE ISSUED SET RETURN_DATE='"+return_date+"' WHERE
395             g.dispose();
396
397             Connection connection1 = connect();
398             Statement stmt1 = connection1.createStatement();
399             stmt1.executeUpdate("USE LIBRARY");
400             ResultSet rs1 = stmt1.executeQuery("SELECT PERIOD FROM ISSUED WHERE IID="+iid)

```

```

401         String diff=null;
402         while (rs1.next()) {
403             diff = rs1.getString(1);
404         }
405         int diff_int = Integer.parseInt(diff);
406         if (ex.days&amp;amp;amp;amp;amp;amp;amp;amp;gt;diff_int) { //If number of days are
407             calculate fine
408
409             //System.out.println(ex.days);
410             int fine = (ex.days-diff_int)*10; //fine for every day after the period
411             //update fine in the system
412             stmt1.executeUpdate("UPDATE ISSUED SET FINE="+fine+" WHERE IID="+id);
413             String fine_str = ("Fine: Rs. "+fine);
414             JOptionPane.showMessageDialog(null,fine_str);
415         }
416         JOptionPane.showMessageDialog(null,"Book Returned!");
417     }
418 }
419
420
421         catch (SQLException e1) {
422             // TODO Auto-generated catch block
423             JOptionPane.showMessageDialog(null, e1);
424         }
425     }
426 }
427
428     });
429     g.add(l4);
430     g.add(create_but);
431     g.add(l1);
432     g.add(F_iid);
433     g.add(F_return);
434     g.setSize(350,250);//400 width and 500 height
435     g.setLayout(null);//using no layout managers
436     g.setVisible(true);//making the frame visible
437     g.setLocationRelativeTo(null);
438 }
439
440     });
441     f.add(create_but);
442     f.add(return_book);
443     f.add(issue_book);
444     f.add(add_book);
445     f.add(issued_but);
446     f.add(users_but);
447     f.add(view_but);
448     f.add(add_user);
449     f.setSize(600,200);//400 width and 500 height
450     f.setLayout(null);//using no layout managers
451     f.setVisible(true);//making the frame visible
452     f.setLocationRelativeTo(null);
453 }
454 }
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478

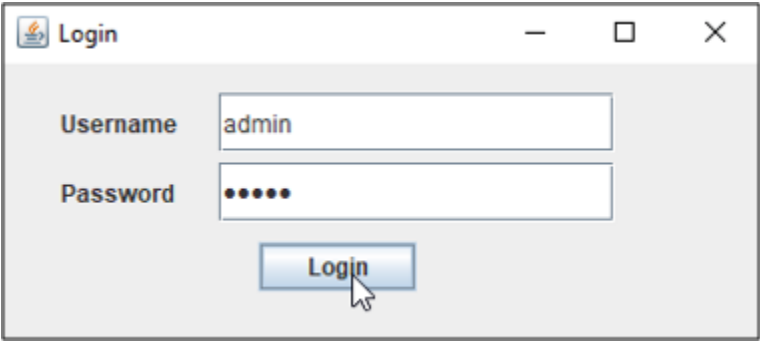
```

479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511

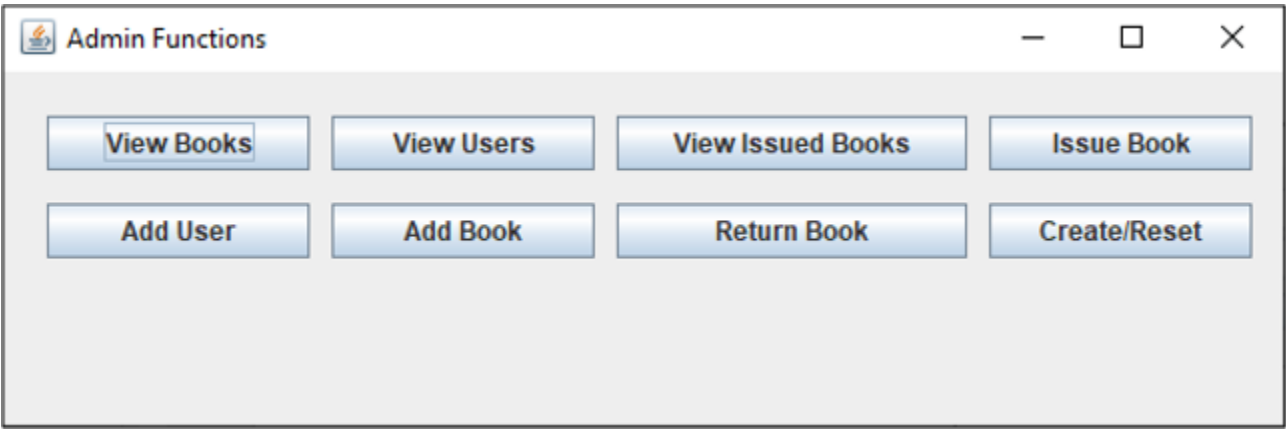
Now that you have understood all the functions, let us execute our library management system project in Java and see the outputs.

Output:

Execute the application by clicking on the run button. Once, you execute you will see the below dialog box. In the below dialog box, mention username and password as {admin, admin}. Then click on the Login button.



Once you click on the **Login button**, you will see the below dialog box opening up.



Here you have various options which you can explore. So, let us start with the first one:

View Books

Once, you click on View Books button, you will see the below frame displaying all the books present in the database, with their details.

Books Available			
BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

View Users


The View Users button is used to view the current users on the system. Since we just have only one user present i.e the admin, it will show you output as below:

Users List			
UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true

Create/Reset

This functionality is used to create or reset a database. So, once you click on the button Create/Rest, you will see the below output:

Message




Database Created/Reset!

OK

Add User

To add a user, click on the option “Add User” and mention details such as **username**, **password** and **choose the radio button user or admin**. By default, it will be the user. Then, click on **Create**.

Enter User Details

—

□

×

Username

sahiti

Password

●●●●●●

☐ Admin

☒ User

Create

Once the user is created, you will see an output as below:


Message

i

User added!

OK

Now, again if you click on **View Users button**, you will see the below output:

Users List

—

□


×

UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true
2	sahiti	sahiti	false

Alright, so now that we have added a user. Let us say, that particular user wants to issue books. To do that, the user has to choose the option of Issue Book.

Issue Book

Suppose, if you are the user, once you click on the **Issue Book button**, you have to mention the **Book ID, User ID, Period(**Number of days for issuing the book**)**, and the **Issue Date** as follows:

 Enter Details

Book ID(BID)

3

User ID(UID)

2

Period(days)

10


Issued Date(DD-MM-YYYY)

02-09-2019

Submit

Then click on **Submit**. Once, you click on **Submit**, you will see the below dialog box:

Message


 Book Issued!

OK

Now, if you want to see the issued books details, you can use the View Issued Books functionality.


View Issued Books

Once you click on this button, you will see the following output:

 Users List

IID	UID	BID	ISSUED_DATE	RETURN_DATE	PERIOD	FINE
1	2	3	02-09-2019		10	

Alright, so, now **if the user logs in to the system**, using the login function, as below:

 Login

Username

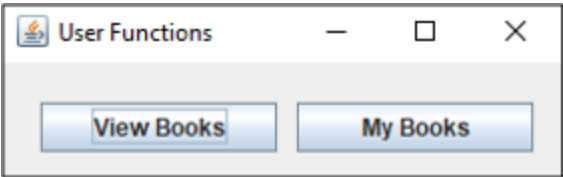
sahiti

Password

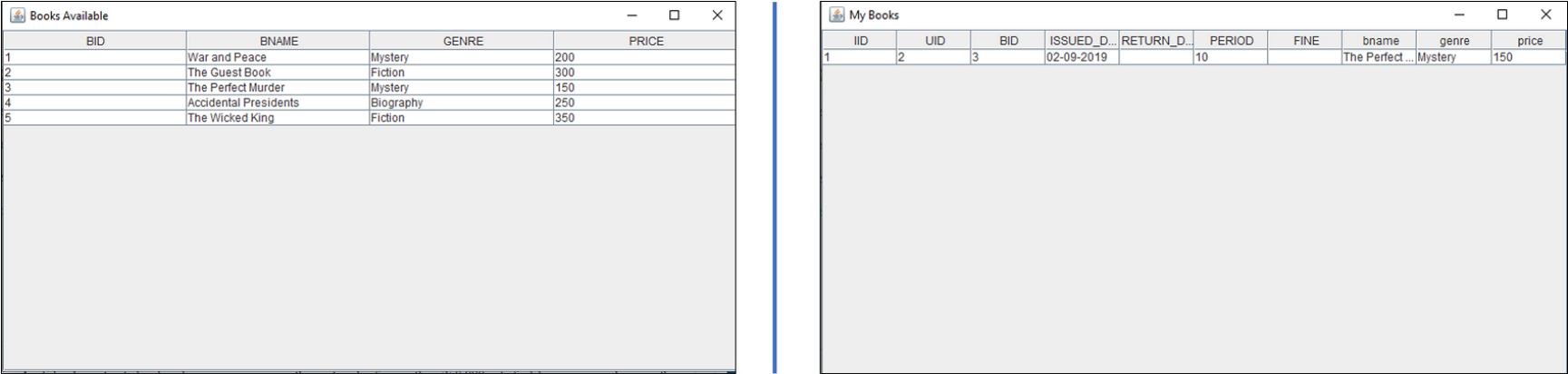
••••••

Login

Then the user will see the below User Menu.



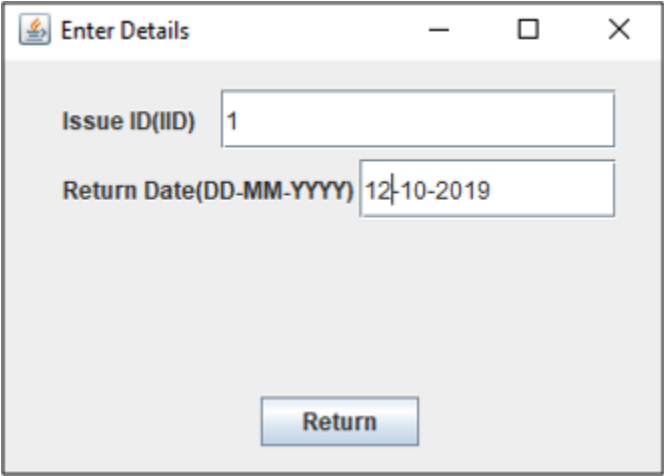
Here, the user can **view all the books** in the database by **using the View Books option** and the **books issued by the user** in the **My Books section** as below:



Now, if you wish to return the book, then you have to choose the option of Return Book.

Return Book

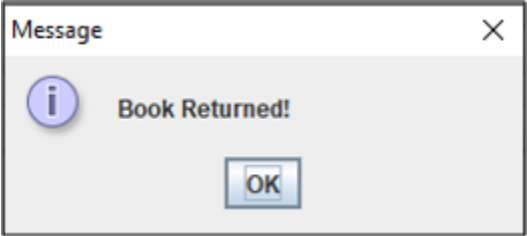
Once, you click on the Return Book, mention the **Issue ID and the return date** as below. Then click on **Return**.



Then, you see a message box displaying the fine.



After that, you again see a dialog box, showing the message "**Book Returned**". Refer below.



Now, if you click on the **View Issued Books**, you will see the below output:

Users List

IID	UID	BID	ISSUED_DATE	RETURN_DATE	PERIOD	FINE
1	2	3	02-09-2019	12-10-2019	10	300

Lastly, if you wish to add a book, you can use the option of Add Book.

Add Book

Click on the **Add Book button**, and mention the **book name, genre and price**. Then, click on the **Submit button**. Refer below.

Enter Book Details

Book Name

Murder on the Orient Express

Genre

Thriller

Price

1000

Submit

You will see a dialog box displaying the below message:

Message

i

Book added!

OK

Apart from this, you can also, see the added books in the **View Books** section as below:

Books Available			
BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350
6	Murder on the Orient Express	Thriller	1000

This brings us to the end of our article on Library Management System Project in Java. I hope you found this article informative and added value to your knowledge.