ASSIGNMENT 1

LIBRARY MANAGEMENT SYSTEM

NAME: Ushba Fatima

CMS ID: 467212

Class: BESE 14-B

INTRODUCTION

1. Project Overview:

The Library Management System is a Java-based application developed to streamline library operations and enhance the management of student and book records. This system provides a graphical user interface (GUI) for librarians to efficiently manage library resources. It facilitates tasks such as adding students, adding books, searching books, displaying books, issuing books, and returning books. The application is integrated with a MySQL database to store and retrieve data, ensuring data consistency and reliability.

2. Purpose and Objectives:

The Library Management System is designed to modernize and optimize the operations within a library environment. Its primary purpose is to simplify the management of student and book records while providing librarians with a user-friendly interface. By automating various tasks, this application aims to enhance the efficiency and effectiveness of library operations, ultimately improving the overall library experience for both librarians and patrons.

- a. **Automation:** Automate tasks related to adding, searching, issuing, and returning books, reducing manual effort and time consumption.
- b. **Streamlined Operations**: Provide a streamlined process for managing student and book records, ensuring data accuracy and consistency.
- c. **User-Friendly Interface**: Offer a user-friendly interface for librarians to interact with the system easily and efficiently.

SYSTEM DESIGN

1. Classes:

The following major classes are utilized in the Library Management System:

a. Book:

Represents a book entity with attributes such as book ID, title, author, genre, availability status, issue date, due date, and user ID. The private attributes are accessed through getters/setters.

Attributes:

• **bookID**: Identifier for the book.

title: Title of the book.

• author: Author of the book.

• genre: Genre of the book.

available: Availability status of the book.

• issueDate: Date when the book was issued.

dueDate: Due date for returning the book.

userID: ID of the user who borrowed the book.

Book

bookID: Stringtitle: Stringauthor: Stringgenre: Stringavailable: String

issueDate: StringdueDate: StringuserID: String

+ getBookID(): String

+ getBookTitle(): String

+ getAuthor(): String

+ getGenre(): String

+ getIssueDate(): String

+ getDueDate(): String

+ getUserID(): String

+ getAvailability(): String

+ setAvailability(available: String): void

+ setBookID(bookID: String): void

+ setBookTitle(title: String): void

+ setAuthor(author: String): void

+ setGenre(genre: String): void

+ setUserID(userID: String): void

+ setDueDate(dueDate: String): void

+ setIssueDate(issueDate: String):

void

b. User:

Represents a user entity, typically a student, with attributes including user ID, name, contact number, and course name. The private attributes are accessed through getters/setters.

Attributes:

- **userID:** Unique identifier for the user.
- name: Name of the user.
- **contactNo:** Contact number of the user.
- **courseName:** Course in which the user is enrolled.

User

userID: Stringname: StringcontactNo: StringcourseName: String

+ getUserID(): String + getUserName(): String + getContactNo(): String + getCourseName(): String

+ setUserID(userID: String): void + setUserName(name: String): void + setContactNo(contactNo: String): void

+ setCourseName(courseName: String): void

c. Library:

Serves as the core class responsible for implementing various functionalities of the library management system.

Contains methods for adding users and books to the database, finding books and users by their attributes, issuing and returning books, and validating IDs.

Methods:

- addUserToDB(User user): Adds a new user to the database. Returns a boolean indicating success status of adding a user.
- addBookToDB(Book book): Adds a new book to the database. Returns a boolean indicating success status of adding a book.
- **findBookbyTitle(String title):** Finds a book in the database by its title and returns it.
- **findBookbyID(String ID):** Finds a book in the database by its ID and returns it.
- **findUserbyID(String ID):** Finds a user in the database by their ID and returns it.
- **issueBook(Book book):** Issues a book to a user. Returns Boolean indicating the success status of issuing a book.
- **returnBook(Book book):** Returns a book to the library that was issued earlier. Returns Boolean indicating the success status of returning a book.
- isValidID(String ID): Validates the format of an ID. Returns a Boolean to validate the ID

Library

- + addUserToDB(User): Boolean
- + addBookToDB(Book): Boolean
- + findBookbyTitle(String): Book
- + findBookbyID(String): Book
- + findUserbyID(String): User
- + issueBook(Book): Boolean
- + returnBook(Book): Boolean
- + isValidID(String): Boolean

2. Database:

MySQL serves as the backend database management system for the library management application. The database is named "library" and consists of two tables: "student" and "book".

a. Student Table:

Manages student data.

Columns:

- **studentName**: Name of the student (varchar(30)).
- **studentID**: Unique identifier for the student (varchar(20), Primary Key).
- **contact**: Contact number of the student (varchar(20)).
- **course**: Course enrolled by the student (varchar(10)).

b. Book Table:

Handles book-related information.

Columns:

- **bookTitle**: Title of the book (varchar(50)).
- **bookID**: Unique identifier for the book (varchar(20), Primary Key).
- author: Author of the book (varchar(50)).
- **genre**: Genre of the book (varchar(30)).
- availability: Availability status of the book (varchar(20)).
- **issueDate**: Date when the book was issued (varchar(60)).
- **dueDate**: Due date for returning the book (varchar(60)).
- studentID: ID of the student who borrowed the book (varchar(20)).

c. Database Connectivity:

JDBC (Java Database Connectivity) is utilized to establish a connection to the MySQL database. The Connect class contains the logic for establishing this connection. This configuration ensures seamless interaction between the application and the MySQL database for effective data storage and retrieval. The following command is ru for establishing a connection.

Connectioncon=DriverManager.getConnection("jdbc:mysql://localhost:3306/lib
rary", "root", "MySQL@2773778");

IMPLEMENTATION

1. <u>Software Requirements:</u>

- a. Java Development Kit (JDK)
- b. Eclipse IDE with WindowBuilder plugin

Windows Builder 1.13.0: Eclipse downloads - Select a mirror | The Eclipse Foundation

c. MySQL database

MySQL :: MySQL Community Downloads (My SQL Installer for Windows)

d. MySQL Connector/J 8.3.0

MySQL :: Download Connector/J (Platform Independent)

e. rs2xml.jar file for populating JTables

rs2xml.jar - Google Drive

2. Key Features:

- a. User Authentication: Admins can log in using predefined credentials to access the application.
- **b. User Management:** Add new users to the database, including their name, ID, contact number, and course.
- **c. Book Management:** Add new books to the database, including details like title, ID, author, genre, and availability.
- **d. Search Functionality:** Search for books by title.
- e. Display Books: View a list of all books stored in the database.
- **f. Book Issuing:** Issue books to users by updating their due dates, issue dates, and availability status.
- g. Book Returning: Return books by updating their availability status and removing user details.
- **h. GUI Interface:** Utilize a user-friendly graphical interface created using the WindowBuilder plugin for Eclipse, featuring drag-and-drop tools and action buttons for seamless interaction.
- **i. Login Page Integration:** Begin the program execution with a login page to authenticate users and grant access to the entire application.

GRAPHICAL USER INTERFACE

The graphical user interface (GUI) of this application is developed using the Windows Builder tool in Eclipse, which offers a convenient drag-and-drop interface for designing user interfaces. The GUI consists of various windows, each serving a specific purpose in the library management system.

The program starts with the loginPage, the entry point to the system. Users log in here to access the library management features. From the homePage, users can navigate to different windows and perform tasks. The loginPage controls access to the entire program, ensuring security.

Here's an overview of the GUI components and their functionality:

3. Login Page:

This window prompts the user to enter their credentials to log in as an administrator. It typically includes fields for entering the username and password, along with a "Login" button to authenticate.

Admin credentials:

a. Username: Admin1 Password: library123b. Username: Admin2 Password: library456c. Username: Admin3 Password: library789

Screenshot:



4. Home Page:

Upon successful login, the user is directed to the homePage, which serves as the main interface for accessing different functionalities of the library management system.

It contains buttons representing various actions or tasks that the user can perform, such as adding a new user, adding a new book, searching for books, displaying books, issuing books, and returning books. Each button or tab is associated with a specific window. The logout button leads back to the login window

5. Add User Window:

This window allows the librarian to add a new user to the library database. It typically includes fields for entering the user's name, ID, contact number, and course details. A "Add User" button is provided to add the user to the database.

6. Add Book Window:

Similar to the Add User window, this window enables the librarian to add a new book to the library database. It includes fields for entering the book's title, ID, author, genre, and availability status. A "Add Book" button allows the librarian to add the book to the database.

7. Search Book Window:

This window allows the librarian to search for books in the database based on their title. It typically includes a search field where the librarian can enter the title of the book they want to search for. A "Search" button triggers the search operation, displaying the matching results.

8. Display Books Window:

This window displays a list of all the books available in the library database in table format.

9. <u>Issue Book Window:</u>

This window facilitates the process of issuing a book to a library user. It includes fields for entering the book ID and the user ID to whom the book is being issued. A "Issue" button allows the librarian to complete the book issuing process.

10. Return Book Window:

Similar to the Issue Book window, this window is used to process the return of a book by a library user. It includes fields for entering the book ID and the user ID returning the book. A "Return" button triggers the book return process.

Overall, the GUI is designed to provide user-friendly interface so that the librarian can navigate through different operations easily.

TESTING

During the testing phase of the library management system, evaluations are conducted to assess its functionality and performance. Various scenarios are explored to ensure the reliability and effectiveness of the software. These tests cover aspects such as login functionality, addition of users and books, search capabilities, and book issuing and return processes. Additionally, error handling mechanisms are thoroughly tested to ensure proper handling of unexpected situations.

1. Login Page

a. Entering correct credentials leads to the home page.



b. Entering wring credentials displays error message.



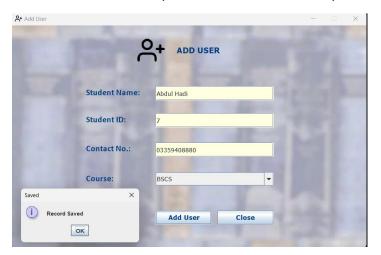
2. Home Page:

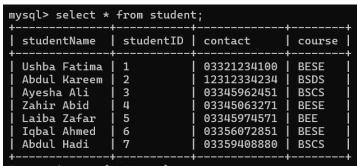
It is ensured that all buttons on the homepage lead to the correct windows and functionalities and the logout button leads back to the login page.



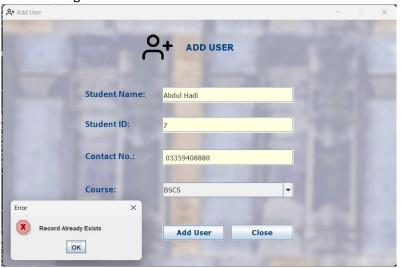
3. Add User Window:

a. Verify that users can be added successfully in the database without any errors.

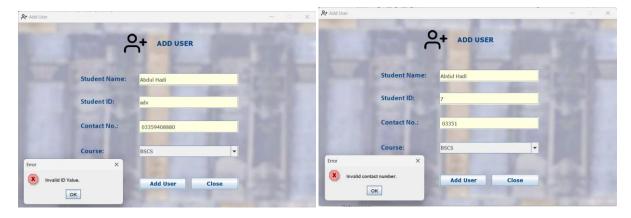




b. Check if the system properly handles cases where a user with the same ID already exists, displaying an appropriate error message.

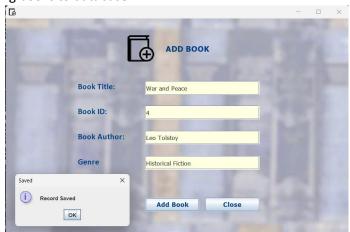


c. Validate the handling of invalid inputs such as an invalid ID, name, or contact number, ensuring that the system displays relevant error messages.



4. Add Book Window

a. Successfully adding books to database





b. Checking if the record already exists based on book ID.

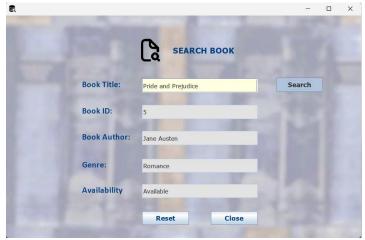


c. Checking for valid ID value:

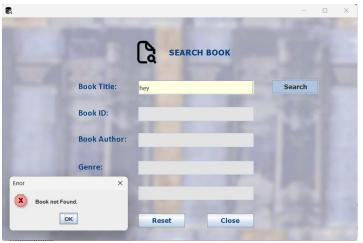


5. <u>Searching Book Window:</u>

a. Successful searching of book by title from database on pressing the search button.



b. Error message if book not found in the database.



6. <u>Display Books:</u>

Simply displays the table of books stored in the database.



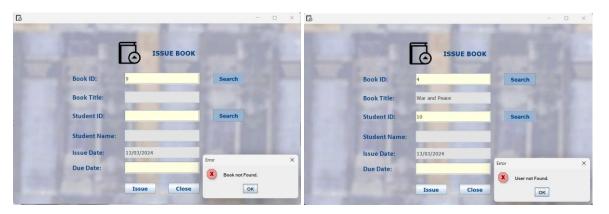
7. <u>Issue Book:</u>

a. Successful issuing of a book.



bookTitle	bookID	author	genre	availability	issueDate	dueDate	studentID
Java- How to Program	1	Paul & Dietel	Education	Available	NULL	NULL	NULL
Harry Potter	2	JK Rowling	Fantasy	Available	NULL	NULL	NULL
The Hunger Games	3	Suzanne Collins	Fiction	Available	NULL	NULL	NULL
War and Peace	4	Leo Tolstoy	Historical Fiction	Available	NULL	NULL	NULL
Pride and Prejudice	5	Jane Austen	Romance	Not Available	13/03/2024	13/05/2024	2

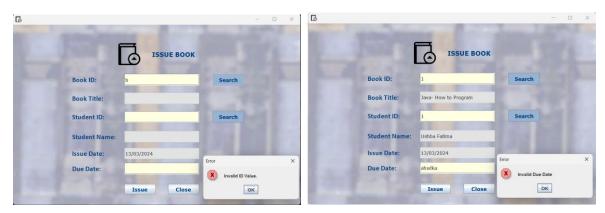
b. Error if book/user data is not found in database



c. Error is the book has already been issued



d. Input Validation:

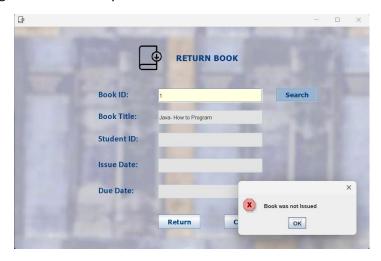


8. Return Book Window:

a. Successful returning of book



b. Error message if book is already available.



c. Input Validation:



CODE

1. Book Class:

```
package libraryManagementSystem;
                                                            return userID;
                                                        }
// Class representing a Book object
class Book {
                                                        public String getAvailability() {
    // Private fields to store book
                                                            return available;
attributes
    private String bookID;
                                                        // Setter methods to modify the values
    private String title;
    private String author;
                                                   of private fields
    private String genre;
    private String available;
                                                        public void setAvailability(String
    private String issueDate;
                                                    available) {
    private String dueDate;
                                                            this.available = available;
    private String userID;
                                                        }
    // Default constructor
                                                        public void setBookID(String bookID) {
    public Book() {
                                                            this.bookID = bookID;
    // Getter methods to retrieve the
                                                        public void setBookTitle(String title)
values of private fields
                                                            this.title = title;
    public String getBookID() {
        return bookID;
    }
                                                        public void setAuthor(String author) {
                                                            this.author = author;
    public String getBookTitle() {
        return title;
                                                        public void setGenre(String genre) {
                                                            this.genre = genre;
    public String getAuthor() {
        return author;
                                                        public void setUserID(String userID) {
    }
                                                            this.userID = userID;
    public String getGenre() {
                                                        }
        return genre;
                                                        public void setDueDate(String dueDate)
    }
    public String getIssueDate() {
                                                            this.dueDate = dueDate;
        return issueDate;
                                                        }
    }
                                                        public void setIssueDate(String
    public String getDueDate() {
                                                    issueDate) {
        return dueDate;
                                                            this.issueDate = issueDate;
                                                    }
    public String getUserID() {
```

2. User Class

```
package libraryManagementSystem;
// Class representing a User object
public class User {
    // User Attributes
    private String userID;
    private String name;
    private String contactNo;
    private String courseName;
    // Constructor
    public User() {
    }
    // Getter methods to retrieve the values of private fields
    public String getUserID() {
        return userID;
    }
    public String getUserName() {
        return name;
    public String getContactNo() {
        return contactNo;
    public String getCourseName() {
       return courseName;
    }
    // Setter methods to modify the values of private fields
    public void setUserID(String userID) {
        this.userID = userID;
    public void setUserName(String name) {
        this.name = name;
    }
    public void setContactNo(String contactNo) {
        this.contactNo = contactNo;
    }
    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }
}
```

3. Library Class:

```
package libraryManagementSystem;
                                                  // Return false to indicate unsuccessful
                                                  addition of user to the database
                                                                    return false;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
                                                      }
import java.sql.SQLException;
                                                      /***** ADD USER
public class Library {
                                                  **************/
                                                      /***** ADD BOOK
   static Connection con =
Connect.ConnectToDB();
                                                  **************/
   static PreparedStatement pst = null;
                                                      public static Boolean
   static ResultSet rs = null;
                                                  addBookToDB(Book book) {
                                                             try {
                                                                    // Prepare SQL
   public Library() {
                                                  statement to insert book details into
                                                  the database
                                                                    pst =
   /***** ADD USER ******/
                                                  con.prepareStatement(
   public static Boolean
                                                                                   "INSERT
                                                  INTO BOOK
addUserToDB(User user) {
                                                  (bookTitle,bookID,author,genre,
          try {
                  // Prepare SQL
                                                  availability) VALUES (?,?,?,?,?)");
statement to insert user details into
the database
                                                                    // Set values for the
                                                  parameters in the SQL statement using
con.prepareStatement("INSERT INTO
                                                  book details
                                                                    pst.setString(1,
(studentName, studentID, contact, course)
                                                  book.getBookTitle());
VALUES (?,?,?,?)");
                                                                    pst.setString(2,
                                                  book.getBookID());
                  // Set values for the
                                                                    pst.setString(3,
                                                  book.getAuthor());
parameters in the SQL statement using
user details
                                                                    pst.setString(4,
                  pst.setString(1,
                                                  book.getGenre());
user.getUserName());
                                                                    pst.setString(5,
                  pst.setString(2,
                                                  book.getAvailability());
user.getUserID());
                                                                    // Execute the SOL
                  pst.setString(3,
user.getContactNo());
                                                  statement to insert the book into the
                  pst.setString(4,
                                                  database
user.getCourseName());
                                                                    pst.executeUpdate();
                  // Execute the SQL
                                                                    // Return true to
statement to insert the user into the
                                                  indicate successful addition of book to
database
                                                  the database
                  pst.executeUpdate();
                                                                    return true;
                                                             } catch (SQLException e1) {
                  // Return true to
                                                                    e1.printStackTrace();
indicate successful addition of user to
                                                                    return false;
the database
                                                             }
                  return true;
                                                      }
                                                      /***** ADD BOOK
          } catch (SQLException e1) {
                                                  **************/
                  e1.printStackTrace();
```

```
/***** FIND BOOK
***************/
                                                              // Return the Book object
   // Find Book by Title
                                                   with the retrieved details
   public static Book
                                                              return book;
findBookbyTitle(String title) {
                                                       }
           // Create a new Book object
                                                       // Find book by name
           Book book = new Book();
                                                       public static Book
           try {
                                                   findBookbyID(String ID) {
                  // Prepare SQL
                                                              // Create a new Book object
statement to select book from database
                                                              Book book = new Book();
using book title
                  pst =
                                                              try {
con.prepareStatement("SELECT * FROM BOOK
                                                                      // Prepare SQL
WHERE bookTitle=?");
                                                   statement to select book from database
                  pst.setString(1,
                                                   using book ID
title);
                                                                      pst =
                                                   con.prepareStatement("SELECT * FROM BOOK
                  rs =
pst.executeQuery();
                                                   WHERE bookID=?");
                                                                      pst.setString(1, ID);
                  // If a book with the
specified title exists in the database
                                                   pst.executeQuery();
                  if (rs.next()) {
                         // Set the
                                                                      // If a book with the
book details retrieved from the database
                                                   specified ID exists in the database
                                                                      if (rs.next()) {
                                                                             // Set the
   book.setBookTitle(rs.getString("book
                                                   book details retrieved from the database
Title"));
   book.setBookID(rs.getString("bookID"
                                                       book.setBookTitle(rs.getString("book
                                                   Title"));
));
                                                       book.setBookID(rs.getString("bookID"
   book.setAuthor(rs.getString("author"
));
                                                   ));
   book.setGenre(rs.getString("genre"))
                                                       book.setAuthor(rs.getString("author"
                                                   ));
   book.setAvailability(rs.getString("a
                                                       book.setGenre(rs.getString("genre"))
vailability"));
                                                   ;
   book.setDueDate(rs.getString("dueDat
                                                       book.setUserID(rs.getString("student
e"));
                                                   ID")); // Assuming you have a method to
                                                   set the user ID who borrowed
   book.setIssueDate(rs.getString("issu
eDate"));
                                                                      // the book
                  } else {
                         // If no book
with the specified title is found,
                                                       book.setAvailability(rs.getString("a
return null
                                                   vailability"));
                         return null;
                                                       book.setDueDate(rs.getString("dueDat
                  }
                                                   e"));
           } catch (SQLException e1) {
                  e1.printStackTrace();
                                                       book.setIssueDate(rs.getString("issu
                  return null;
                                                   eDate"));
```

}

```
} else {
                         return null;
                                                             } catch (SQLException e1) {
                                                                     e1.printStackTrace();
                                                                     return null;
          } catch (SQLException e1) {
                                                             }
                  e1.printStackTrace();
                  return null;
                                                             // Return the User object
          }
                                                  with the retrieved details
                                                             return user;
          // Return the Book object
                                                      /***** FIND USER
with the retrieved details
                                                   **********/
          return book;
   /***** FIND BOOK
                                                      /****** ISSUE BOOK
****************
                                                   ****************
                                                      public static Boolean issueBook(Book
   /***** FIND USER
                                                  book) {
**********/
                                                             try {
   public static User
                                                                     // Updates book
findUserbyID(String ID) {
                                                  details in the database based on the
                                                  provided information that
           // Create a new User object
          User user = new User();
                                                                     // has been populated
                                                   in the object book
                                                                     pst =
          trv {
                                                  con.prepareStatement("UPDATE BOOK SET
                  // Prepare SQL
statement to select user from database
                                                   ISSUEDATE=?,DUEDATE=?,STUDENTID=? WHERE
                                                  BOOKID=?");
using user ID
                  pst =
                                                                     pst.setString(1,
con.prepareStatement("SELECT * FROM
                                                  book.getIssueDate());
STUDENT WHERE studentID=?");
                                                                     pst.setString(2,
                  pst.setString(1, ID);
                                                  book.getDueDate());
                                                                     pst.setString(3,
pst.executeQuery();
                                                  book.getUserID());
                                                                     pst.setString(4,
                  // If the user with
                                                  book.getBookID());
the specified ID exists in the database
                  if (rs.next()) {
                                                                     int rowsAffected =
                         // Set the
                                                  pst.executeUpdate(); // Use
user details retrieved from the database
                                                  executeUpdate() for UPDATE queries
                                                                     // If update was
   user.setUserName(rs.getString("stude
                                                  successful
ntName"));
                                                                     if (rowsAffected > 0)
                                                   {
   user.setUserID(rs.getString("student
ID"));
                                                      book.setAvailability("Not
                                                  Available");
   user.setContactNo(rs.getString("cont
act"));
                                                                            // Update the
                                                  availability of the book
   user.setCourseName(rs.getString("cou
                                                                            pst =
rse"));
                                                  con.prepareStatement("UPDATE BOOK SET
                                                  AVAILABILITY=? WHERE BOOKID=?");
                  } else {
                         // If no user
                                                      pst.setString(1, "Not Available");
with the specified ID is found, return
null
                         return null;
                                                      pst.setString(2, book.getBookID());
                  }
```

```
pst.executeUpdate(); // Execute the
                                                                             // Prepare SQL
update query for availability
                                                   statement to update book details in the
                                                   database
                         // Update the
                                                                             pst =
book object's availability
                                                   con.prepareStatement(
   book.setAvailability("Not
                                                       "UPDATE BOOK SET AVAILABILITY=?,
Available");
                                                   ISSUEDATE=?, DUEDATE=?,STUDENTID=? WHERE
                                                   BOOKID=?");
                         return true;
// Indicate that the book was
                                                       pst.setString(1, "Available");
successfully issued
                                                       pst.setString(2, null);
                  } else
                         // Book could
                                                       pst.setString(3, null);
not be issued
                                                       pst.setString(4, null);
                         return false;
                                                       pst.setString(5, bookID);
           } catch (SQLException e1) {
                  // TODO Auto-
generated catch block\
                                                                             // Execute the
                                                   update query and get the number of rows
                  e1.printStackTrace();
                                                   affected
                  return null;
                                                                             int
                                                   rowsAffected = pst.executeUpdate(); //
                                                   Use executeUpdate() for UPDATE queries
        ****** ISSUE BOOK
                                                                             // If the
                                                   update was successful
   /***** RETURN BOOK
                                                                             if
*************/
                                                   (rowsAffected > 0) {
   public static Boolean
returnBook(Book book) {
                                                   Update the book object's availability
           // Get the book ID
          String bookID =
                                                       book.setAvailability("Available");
book.getBookID();
                                                   Return true to indicate successful book
                                                   return
           try {
                  // Prepare SQL
                                                                                    return
statement to select book from database
                                                   true;
using book ID
                                                                             } else
                  pst =
                                                                                    //
con.prepareStatement("SELECT * FROM BOOK
                                                   Return false to indicate unsuccessful
WHERE bookID=?");
                                                   book return
                  pst.setString(1,
                                                                                    return
bookID);
                                                   false;
                                                                     }
pst.executeQuery();
                                                              } catch (SQLException e1) {
                  // If the book exists
in the database
                  if (rs.next()) {
                                                                     // Return false to
                         // Find the
                                                   indicate unsuccessful book return
book by its ID
                                                                     e1.printStackTrace();
                         book =
                                                                     return false;
findBookbyID(book.getBookID());
                                                              }
```

```
return false;
             return null;
                                                   // Return false method if validation
                                                   fails
                                                                    }
       /***** RETURN BOOK
                                                             } catch
   ****************
                                                   (NumberFormatException ex) {
                                                                    // Handle invalid ID
      // Validate ID Value
                                                   format
      public static Boolean
   isValidID(String ID) {
                                                                    return false; //
             try {
                                                   Return false if validation fails
                    int idValue =
   Integer.parseInt(ID);
                                                             return true;
                    // Check if ID is a
                                                      }
   positive number
                    if (idValue <= 0) {</pre>
                                                   }
4. Connect Class for JDBC:
   package libraryManagementSystem;
   import java.sql.Connection;
   import java.sql.DriverManager;
   import java.sql.SQLException;
   import java.util.logging.Level;
   import java.util.logging.Logger;
   public class Connect {
       static Connection con = null;
       // Method to establish connection to the database
       public static Connection ConnectToDB() {
           try {
                // Establish connection to the MySQL database named "library" on
   localhost at port 3306
               // with username "root" and password "MySQL@2773778"
   DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root",
   "MySQL@2773778");
           } catch (SQLException ex) {
               // Log any SQLException that occurs during the connection attempt
               Logger.getLogger(Connect.class.getName()).log(Level.SEVERE, null,
   ex);
           return con; // Return the established connection (may be null if
   connection fails)
       }
```

}

5. Login Page:

```
package libraryManagementSystem;
                                                                 loginPage frame
                                             = new loginPage();
import java.awt.EventOueue;
import javax.swing.JFrame;
                                             frame.setVisible(true);
import javax.swing.JPanel;
                                                             } catch (Exception
import
                                             e) {
javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
                                             e.printStackTrace();
import javax.swing.JOptionPane;
import javax.swing.JTextField;
                                                         }
import javax.swing.JPasswordField;
                                                     });
import javax.swing.JButton;
                                                 /***************
import
java.awt.event.ActionListener;
                                             MAIN METHOD
import java.awt.event.ActionEvent;
                                             ****************************
import java.awt.GridLayout;
import java.awt.BorderLayout;
                                                  * Create the frame.
import
net.miginfocom.swing.MigLayout;
                                                  */
import java.awt.Font;
                                                 public loginPage() {
import javax.swing.SwingConstants;
import java.awt.SystemColor;
                                                       // GUI SETUP
import javax.swing.ImageIcon;
                                                       setResizable(false);
import java.awt.Color;
                                                       setTitle("LMS");
import java.awt.Toolkit;
                                                setIconImage(Toolkit.getDefaultT
                                             oolkit().getImage("C:\\Users\\Dell\\
public class loginPage extends
                                             Desktop\\book.png"));
JFrame {
    private static final long
                                             setDefaultCloseOperation(JFrame.EXIT
serialVersionUID = 1L;
                                             _ON_CLOSE);
                                                     setBounds(500, 250, 500,
    private JPanel contentPane;
    private JTextField
                                             300);
usernameField;
                                                     contentPane = new JPanel();
    private JPasswordField
                                             contentPane.setBackground(SystemColo
passwordField;
                                             r.inactiveCaption);
    /**
                                                     contentPane.setBorder(new
     * Launch the application.
                                             EmptyBorder(5, 5, 5, 5));
                                                     setContentPane(contentPane);
    contentPane.setLayout(null);
MAIN METHOD
                                                     /***** LABELS
****************************/
                                             *******************
    public static void main(String[]
                                                     JLabel txtLogin = new
args) {
        EventQueue.invokeLater(new
                                             JLabel("L M S");
Runnable() {
                                                     txtLogin.setForeground(new
           public void run() {
                                             Color(19, 79, 151));
               try {
```

```
txtLogin.setHorizontalAlignment(Swin
                                             TEXT/PASSWORD FIELDS
                                             **************/
gConstants.CENTER);
       txtLogin.setFont(new
                                                     usernameField = new
Font("Times New Roman", Font.BOLD,
                                             JTextField();
20));
                                                     usernameField.setBounds(175,
        txtLogin.setBounds(174, 10,
                                             99, 229, 31);
144, 31);
        contentPane.add(txtLogin);
                                             contentPane.add(usernameField);
        JLabel txtUsername = new
                                             usernameField.setColumns(10);
JLabel("Username:");
txtUsername.setForeground(new
                                                     passwordField = new
Color(19, 79, 151));
                                             JPasswordField();
                                                     passwordField.setBounds(175,
       txtUsername.setFont(new
Font("Calibri", Font.BOLD, 19));
                                             144, 229, 31);
       txtUsername.setBounds(49,
100, 116, 31);
                                             contentPane.add(passwordField);
                                                     /*********
contentPane.add(txtUsername);
                                             TEXT/PASSWORD FIELDS
                                             **************/
        JLabel txtPassword = new
                                                     /*************
JLabel("Password:");
                                             BUTTONS *******************/
txtPassword.setForeground(new
                                                     // Close Button
Color(19, 79, 151));
                                                     JButton closeButton = new
       txtPassword.setFont(new
                                             JButton("Close");
Font("Calibri", Font.BOLD, 19));
       txtPassword.setBounds(49,
                                             closeButton.setForeground(new
146, 105, 28);
                                             Color(0, 64, 128));
                                                     closeButton.setFont(new
contentPane.add(txtPassword);
                                             Font("Tahoma", Font.BOLD, 14));
                                                     closeButton.setBounds(295,
        JLabel title = new
                                             194, 110, 25);
JLabel("Library Management System");
                                                     // Program Execution
        title.setForeground(new
                                             terminates on close button
Color(19, 79, 151));
                                             closeButton.addActionListener(new
title.setHorizontalAlignment(SwingCo
                                             ActionListener() {
nstants.CENTER);
                                                       public void
        title.setFont(new
                                             actionPerformed(ActionEvent e) {
Font("Times New Roman", Font.BOLD |
                                                             System.exit(ABORT);
Font.ITALIC, 19));
                                                       }
       title.setBounds(113, 41,
                                                     });
256, 32);
        contentPane.add(title);
                                             contentPane.add(closeButton);
        /***** LABELS
********************/
                                                     // Login Button
                                                     JButton loginButton = new
                                             JButton("Login");
```

/*********

```
loginButton.setIcon(new
                                                                // If the
ImageIcon("C:\\Users\\Dell\\Desktop\
                                             credentials don't match, display an
\login.png"));
                                             error message
loginButton.setForeground(new
                                              JOptionPane.showMessageDialog(null,
Color(0, 64, 128));
                                              "Invalid Username or Password",
        loginButton.setFont(new
                                              "Error", JOptionPane.ERROR_MESSAGE);
Font("Tahoma", Font.BOLD, 14));
        loginButton.setBounds(175,
194, 110, 25);
                                                      });
loginButton.addActionListener(new
                                             contentPane.add(loginButton);
ActionListener() {
         public void
                                                      /***********
                                             BUTTONS **********************/
actionPerformed(ActionEvent e) {
             // Get the entered
username and password
             String username =
                                                      /***********
usernameField.getText();
             char[] passwordChars =
                                             BACKGROUND IMAGE
                                              ***************/
passwordField.getPassword();
             String password = new
                                                      JLabel lblNewLabel = new
String(passwordChars);
                                              JLabel("");
             // Perform
                                             lblNewLabel.setHorizontalAlignment(S
authentication or validation here
                                             wingConstants.CENTER);
             // Check if the
                                                      lblNewLabel.setIcon(new
entered username and password match
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
any of the predefined credentials
                                              \book.png"));
             if
                                                      lblNewLabel.setBounds(164, -
(username.equals("Admin1") &&
                                             21, 70, 87);
password.equals("library123") ||
                                             contentPane.add(lblNewLabel);
username.equals("Admin2") &&
password.equals("library456") ||
                                                      JLabel background = new
                                              JLabel("");
username.equals("Admin3") &&
                                                      background.setIcon(new
password.equals("library789")) {
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
                                              \R (1).jpg"));
                                                      background.setBounds(-13, -
                 // If the
credentials match, open the home
                                              18, 522, 302);
                                                      contentPane.add(background);
page
                                                      /**********
                 setVisible(false);
                                             BACKGROUND IMAGE
                                              ******************
homePage().setVisible(true);
             } else {
                                              }
```

6. Home Page:

```
package libraryManagementSystem;
                                                       });
                                                 }
import java.awt.Dimension;
import java.awt.EventQueue;
                                                 /****************
import java.awt.Toolkit;
                                             MAIN METHOD
                                             ****************************
import javax.swing.JFrame;
                                                 /**
import javax.swing.JPanel;
import
                                                  * Create the frame.
javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
                                                 public homePage() {
                                                       // GUI SETUP
import javax.swing.JButton;
import
                                                       setTitle("LMS");
java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
                                                 setIconImage(Toolkit.getDefaultT
import javax.swing.ImageIcon;
                                             oolkit()
import java.awt.Font;
import java.awt.Color;
                                                 .getImage("C:\\Users\\Dell\\Desk
import javax.swing.SwingConstants;
                                             top\\Library Management
import java.awt.SystemColor;
                                             System\\Images\\book.png"));
public class homePage extends JFrame
                                                 setDefaultCloseOperation(JFrame.
                                             EXIT ON CLOSE);
   private static final long
                                                 setExtendedState(JFrame.MAXIMIZE
serialVersionUID = 1L;
                                             D BOTH);
   private JPanel contentPane;
                                                       Dimension screenSize =
                                             Toolkit.getDefaultToolkit().getScree
   /**
                                             nSize();
    * Launch the application.
                                                       int screenWidth =
                                             screenSize.width;
   int screenHeight =
MAIN METHOD
                                             screenSize.height;
****************************/
   public static void main(String[]
                                                       int width = 1100;
args) {
                                                       int height = 700;
         EventQueue.invokeLater(new
Runnable() {
                                                       int x = (screenWidth -
                public void run() {
                                             width) / 2;
                      try {
                                                       int y = (screenHeight -
                                             height) / 2;
   homePage frame = new homePage();
                                                       setBounds(1, 1,
   frame.setVisible(true);
                                             screenWidth, screenHeight);
                       } catch
                                                       contentPane = new
(Exception e) {
                                             JPanel();
                                                       contentPane.setBorder(new
   e.printStackTrace();
                                             EmptyBorder(5, 5, 5, 5));
                      }
```

}

```
setContentPane(contentPane);
                                                 searchBookButton.addActionListen
                                             er(new ActionListener() {
   contentPane.setLayout(null);
                                                              public void
                                             actionPerformed(ActionEvent e) {
searchBookWindow().setVisible(true);
         // Each button opens
                                                       });
corresponding window when pressed
         JButton addUserButton =
                                                 searchBookButton.setBounds(575,
new JButton("Add User");
                                             48, 154, 40);
         addUserButton
                                                 contentPane.add(searchBookButton
                       .setIcon(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                             );
\Library Management
System\\Images\\adduser.png"));
                                                       JButton addBookButton =
                                             new JButton("Add Book");
   addUserButton.setForeground(new
Color(0, 64, 128));
                                                 addBookButton.setForeground(new
                                             Color(0, 64, 128));
   addUserButton.addActionListener(
                                                       addBookButton
new ActionListener() {
                                                                    .setIcon(new
                public void
                                             ImageIcon("C:\\Users\\Dell\\Desktop\
actionPerformed(ActionEvent e) {
                                             \Library Management
                                             System\\Images\\addbook.png"));
addUserWindow().setVisible(true);
                                                 addBookButton.addActionListener(
                }
                                             new ActionListener() {
         });
         addUserButton.setFont(new
                                                              public void
Font("Tahoma", Font.BOLD, 14));
                                             actionPerformed(ActionEvent e) {
   addUserButton.setBounds(95, 48,
                                             addBookWindow().setVisible(true);
146, 40);
                                                       });
   contentPane.add(addUserButton);
                                                       addBookButton.setFont(new
                                             Font("Tahoma", Font.BOLD, 14));
         JButton searchBookButton =
new JButton("Search Book");
                                                 addBookButton.setBounds(338, 48,
                                             146, 40);
   searchBookButton.setForeground(n
ew Color(0, 64, 128));
                                                 contentPane.add(addBookButton);
         searchBookButton
                                                       JButton displayBooksButton
                       .setIcon(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                             = new JButton("Display Books");
                                                       displayBooksButton
\Library Management
System\\Images\\datamining.png"));
                                                                    .setIcon(new
                                             ImageIcon("C:\\Users\\Dell\\Desktop\
   searchBookButton.setFont(new
                                             \Library Management
Font("Tahoma", Font.BOLD, 14));
                                             System\\Images\\display.png"));
```

```
displayBooksButton.setForeground
                                                        JButton returnBookButton =
(new Color(0, 64, 128));
                                              new JButton("Return Book");
   displayBooksButton.addActionList
                                                  returnBookButton.setForeground(n
ener(new ActionListener() {
                                              ew Color(0, 64, 128));
                public void
actionPerformed(ActionEvent e) {
                                                  returnBookButton.addActionListen
                                              er(new ActionListener() {
displayBooksWindow().setVisible(true
                                                               public void
                                              actionPerformed(ActionEvent e) {
);
                }
          });
                                              returnBookWindow().setVisible(true);
   displayBooksButton.setFont(new
                                                        });
Font("Tahoma", Font.BOLD, 14));
                                                        returnBookButton
                                                                      .setIcon(new
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
   displayBooksButton.setBounds(808
, 48, 165, 40);
                                              \Library Management
                                              System\\Images\\return.png"));
   contentPane.add(displayBooksButt
                                                  returnBookButton.setFont(new
on);
                                              Font("Tahoma", Font.BOLD, 14));
          JButton issueBookButton =
new JButton("Issue Book");
                                                  returnBookButton.setBounds(1293,
          issueBookButton
                                              48, 154, 40);
                       .setIcon(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                                  contentPane.add(returnBookButton
\Library Management
                                              );
System\\Images\\issuebook.png"));
                                                        JButton logoutButton = new
                                              JButton("Logout");
   issueBookButton.setForeground(ne
w Color(0, 64, 128));
                                                  logoutButton.setForeground(new
   issueBookButton.addActionListene
                                              Color(0, 64, 128));
r(new ActionListener() {
                                                        logoutButton.setFont(new
                                              Font("Tahoma", Font.BOLD, 14));
                public void
                                                        logoutButton.setIcon(new
actionPerformed(ActionEvent e) {
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
issueBookWindow().setVisible(true);
                                               \Library Management
                                              System\\Images\\logout.png"));
                }
          });
                                                  logoutButton.addActionListener(n
   issueBookButton.setFont(new
                                              ew ActionListener() {
Font("Tahoma", Font.BOLD, 14));
                                                               public void
                                              actionPerformed(ActionEvent e) {
   issueBookButton.setBounds(1057,
48, 146, 40);
                                                  setVisible(false);
   contentPane.add(issueBookButton)
                                              loginPage().setVisible(true);
;
                                                               }
```

```
});
                                               lblTitle1.setHorizontalAlignment
   logoutButton.setBounds(1322,
                                            (SwingConstants.CENTER);
696, 146, 40);
                                                      lblTitle1.setFont(new
                                            Font("Times New Roman", Font.BOLD
   contentPane.add(logoutButton);
                                            Font.ITALIC, 30));
         /**********
                                                      lblTitle1.setBounds(510,
BUTTONS *********************/
                                            296, 546, 46);
         contentPane.add(lblTitle1);
JLabel lblTItle = new
                                                      JLabel lblQuote = new
JLabel("L M S");
                                            JLabel("There's no friend as loyal
                                            as a book!");
         lblTItle.setForeground(new
Color(0, 64, 128));
                                               lblQuote.setForeground(SystemCol
   lblTItle.setHorizontalAlignment(
                                            or.inactiveCaption);
SwingConstants.CENTER);
                                                      lblQuote.setFont(new
                                            Font("Times New Roman", Font.BOLD |
         lblTItle.setFont(new
Font("Times New Roman", Font.BOLD,
                                            Font.ITALIC, 20));
40));
         lblTItle.setBounds(492,
                                               lblQuote.setHorizontalAlignment(
210, 574, 90);
                                            SwingConstants.CENTER);
         contentPane.add(lblTItle);
                                                      lblQuote.setBounds(510,
                                            696, 555, 40);
         JLabel bookimg = new
                                                      contentPane.add(lblQuote);
JLabel("");
                                                      JLabel background = new
                                            JLabel("");
   bookimg.setHorizontalAlignment(S
wingConstants.CENTER);
                                                     background.setIcon(
         bookimg.setIcon(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                            ImageIcon("C:\\Users\\Dell\\Desktop\
\Library Management
                                            \Library Management
System\\Images\\LMSbook.png"));
                                            System\\Images\\bluebackground.jpg")
         bookimg.setBounds(703,
                                            );
114, 146, 110);
                                                      background.setBounds(0,
         contentPane.add(bookimg);
                                            10, 1565, 893);
         JLabel lblTitle1 = new
                                               contentPane.add(background);
JLabel("Library Management System");
                                               }
                                            }
   lblTitle1.setForeground(new
```

Color(0, 64, 128));

7. Add User Page:

```
package libraryManagementSystem;
                                                addUserWindow frame = new
import java.awt.EventQueue;
                                          addUserWindow();
import javax.swing.JFrame;
import javax.swing.JPanel;
                                                frame.setVisible(true);
import javax.swing.border.EmptyBorder;
                                                                   } catch
import java.awt.Toolkit;
                                          (Exception e) {
import javax.swing.JLabel;
import javax.swing.JOptionPane;
                                                e.printStackTrace();
import javax.swing.JTextField;
                                                                   }
import javax.swing.JComboBox;
                                                             }
import java.awt.Font;
                                                      });
                                                }
import
javax.swing.DefaultComboBoxModel;
                                                 /****** MAIN
import javax.swing.ImageIcon;
                                          import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Color;
                                                 * Create the frame.
import javax.swing.SwingConstants;
import java.awt.SystemColor;
                                                public addUserWindow() {
                                                       // Window GUI
public class addUserWindow extends
                                                       setResizable(false);
JFrame {
                                                       setTitle("Add User");
      private static final long
                                                setIconImage(Toolkit.getDefaultT
serialVersionUID = 1L;
                                          oolkit()
      private JPanel contentPane;
      private JTextField
                                                 .getImage("C:\\Users\\Dell\\Desk
studentNameField;
                                          top\\Library Management
                                          System\\Images\\adduser.png"));
      private JTextField
studentIDField;
                                                setDefaultCloseOperation(JFrame.
      private JTextField contactField;
                                          DISPOSE_ON_CLOSE);
                                                       setBounds(350, 150, 800,
       * Launch the application.
                                          530);
                                                      contentPane = new
                                          JPanel();
      /***** MAIN
                                                       contentPane.setBorder(new
EmptyBorder(5, 5, 5, 5));
      public static void main(String[]
args) {
            EventQueue.invokeLater(new
                                                setContentPane(contentPane);
Runnable() {
                  public void run() {
                                                contentPane.setLayout(null);
                        try {
                                                      /************
```

```
JLabel lblID = new
JLabel("Student ID:");
                                                    lblAddUser.setForeground(new
             lblID.setForeground(new
                                             Color(0, 64, 128));
Color(0, 64, 128));
                                                          lblAddUser.setFont(new
             lblID.setFont(new
                                             Font("Tahoma", Font.BOLD, 18));
Font("Calibri", Font.BOLD, 19));
             lblID.setBounds(165, 191,
                                                    lblAddUser.setHorizontalAlignmen
169, 43);
                                             t(SwingConstants.CENTER);
                                                           lblAddUser.setIcon(new
             contentPane.add(lblID);
                                             ImageIcon("C:\\Users\\Dell\\Desktop\\Li
             JLabel lblStudentName =
                                             brary Management
new JLabel("Student Name:");
                                             System\\Images\\adduser1.png"));
                                                          lblAddUser.setBounds(160,
      lblStudentName.setForeground(new
                                             10, 420, 90);
Color(0, 64, 128));
             lblStudentName.setFont(new
                                                    contentPane.add(lblAddUser);
Font("Calibri", Font.BOLD, 19));
                                                           /***********
                                             LABELS ****************/
      lblStudentName.setBounds(165,
                                                          /***** TEXT
131, 158, 43);
                                             FIELDS ***************/
      contentPane.add(lblStudentName);
                                                           studentNameField = new
                                             JTextField();
             JLabel lblcontact = new
JLabel("Contact No.:");
                                                    studentNameField.setBackground(S
                                             ystemColor.info);
      lblcontact.setForeground(new
Color(0, 64, 128));
                                                    studentNameField.setFont(new
             lblcontact.setFont(new
                                             Font("Tahoma", Font.PLAIN, 14));
Font("Calibri", Font.BOLD, 19));
             lblcontact.setBounds(165,
                                                    studentNameField.setBounds(319,
255, 130, 43);
                                             135, 261, 31);
      contentPane.add(lblcontact);
                                                    contentPane.add(studentNameField
                                             );
             JLabel lblcourse = new
JLabel("Course:");
                                                    studentNameField.setColumns(10);
      lblcourse.setForeground(new
                                                          studentIDField = new
Color(0, 64, 128));
                                             JTextField();
             lblcourse.setFont(new
Font("Calibri", Font.BOLD, 19));
                                                    studentIDField.setBackground(Sys
             lblcourse.setBounds(165,
                                             temColor.info);
321, 90, 43);
                                                           studentIDField.setFont(new
                                             Font("Tahoma", Font.PLAIN, 14));
      contentPane.add(lblcourse);
                                                    studentIDField.setBounds(319,
                                             195, 261, 31);
             JLabel lblAddUser = new
JLabel("
            ADD USER");
                                                    contentPane.add(studentIDField);
```

```
// Giving action to add
      studentIDField.setColumns(10);
                                            user button
            contactField = new
                                                   btnAddUser.addActionListener(new
JTextField();
                                            ActionListener() {
                                                                public void
      contactField.setBackground(Syste
                                            actionPerformed(ActionEvent e) {
mColor.info);
            contactField.setFont(new
                                                                      // Retrieving
Font("Tahoma", Font.PLAIN, 14));
                                            the user information from the text
                                            fields
      contactField.setBounds(319, 260,
                                                                      String
261, 31);
                                            studentName =
                                            studentNameField.getText();
      contentPane.add(contactField);
                                                                      String
                                            studentID = studentIDField.getText();
      contactField.setColumns(10);
                                                                      String
            /***** TEXT
                                            contactNo = contactField.getText();
FIELDS *************/
                                                                      String course
            /***** COMBO
                                            courseBox.getSelectedItem().toString();
BOX ************/
            JComboBox courseBox = new
                                                                      // Removing
                                            nay spaces in the contact number or ID.
JComboBox();
                                                                      contactNo =
      \verb|courseBox.setBackground(SystemCo|\\
                                            contactNo.replace(" ", "");
lor.control);
                                                                      studentID =
                                            studentID.replace(" ", "");
            courseBox.setFont(new
Font("Tahoma", Font.PLAIN, 14));
            courseBox.setModel(new
                                                   /***** ERROR
DefaultComboBoxModel(new String[] {
"BESE", "BSDS", "BSCS", "BEE" }));
                                            HANDLING ******************************/
      courseBox.setMaximumRowCount(4);
                                                                      // Checking
            courseBox.setBounds(319,
                                            if all fields have been filled
325, 261, 30);
                                            (studentName.isEmpty() ||
                                            studentID.isEmpty() ||
      contentPane.add(courseBox);
            /****** COMBO
                                            contactNo.isEmpty() ||
BOX *************/
                                            course.isEmpty()) {
            /***** BUTTONS
                                                   JOptionPane.showMessageDialog(nu
****************/
                                            ll, "Please Fill in All FIelds",
            JButton btnAddUser = new
                                            "Error",
JButton("Add User");
                                                   JOptionPane.ERROR MESSAGE);
      btnAddUser.setForeground(new
                                                                      } else {
Color(0, 64, 128));
                                                                            //
                                            Perform additional validations
            btnAddUser.setFont(new
Font("Tahoma", Font.BOLD, 15));
```

```
11, "Contact number must be a numeric
Validate ID (it should be numeric)
                                              value.", "Error",
(!Library.isValidID(studentID)) {
                                                    JOptionPane.ERROR MESSAGE);
      JOptionPane.showMessageDialog(nu
11, "Invalid ID Value.", "Error",
                                                    return; // Exit method if
JOptionPane.ERROR MESSAGE);
                                              validation fails
                                                                               }
      return;
                                                                               //
                                 }
                                              Validation of Student name
                                                                              try {
Validate contact number (it should be
numeric)
                                try {
                                                    // Check if students name
                                              contains any numbers
      // Converts the contact number
                                                    if
string to a long value
                                              (studentName.matches(".*\\d.*")) {
      long contactValue =
                                                    throw new
Long.parseLong(contactNo);
                                              IllegalArgumentException();
                                                    }
      // Check if contact number is a
positive number and has a valid length
                                              catch (IllegalArgumentException e1) {
      if (contactValue <= 0 ||</pre>
contactNo.length() != 11) {
                                                    // Display error message dialog
                                              box
      JOptionPane.showMessageDialog(nu
11, "Invalid contact number.", "Error",
                                                    JOptionPane.showMessageDialog(nu
                                              11,
      JOptionPane.ERROR_MESSAGE);
                                                           "Error: Username is not
                                              valid. Please provide a valid
      return; // Exit method if
                                              username.", "Error",
validation fails
                                                    JOptionPane.ERROR MESSAGE);
catch (NumberFormatException ex) {
                                                    return;
                                                                              }
      // Handle invalid contact
                                                                        }
number, if it has characters other than
                                                    /***** ERROR
digits
                                             HANDLING ***********************/
      JOptionPane.showMessageDialog(nu
```

```
// Creating
an Object of the User as per the
                                                   btnClose.addActionListener(new
provided details
                                             ActionListener() {
                         User userX =
                                                                public void
                                             actionPerformed(ActionEvent e) {
new User();
                                                                       dispose();
      userX.setUserName(studentName);
                                                                }
                                                          });
      userX.setUserID(studentID);
                                                          btnClose.setForeground(new
                                             Color(0, 64, 128));
      userX.setContactNo(contactNo);
                                                          btnClose.setFont(new
                                             Font("Tahoma", Font.BOLD, 15));
      userX.setCourseName(course);
                                                          btnClose.setBounds(456,
                                             410, 124, 31);
                         // Adding
                                                          contentPane.add(btnClose);
                                                          /***** BUTTONS
user to the database
                                             ********/
                         if
(Library.addUserToDB(userX)) {
                                                          /*******
                                             BACKGROUND ***************/
      JOptionPane.showMessageDialog(ro
otPane, "Record Saved", "Saved",
                                                          JLabel background = new
JOptionPane.INFORMATION MESSAGE);
                                             JLabel("");
                                                          background.setIcon(
      clear();
                         } else
                                             ImageIcon("C:\\Users\\Dell\\Desktop\\Li
                                             brary Management
                                //
Error message if the student already
                                             System\\Images\\windowbackground.jpg"))
exists based in the Student ID
                                                          background.setBounds(0, 0,
      JOptionPane.showMessageDialog(nu
                                             801, 505);
11, "Record Already Exists", "Error",
JOptionPane.ERROR MESSAGE);
                                                   contentPane.add(background);
                                                          /*******
                                             BACKGROUND **************/
                   }
                                                   }
            });
             btnAddUser.setBounds(319,
                                                   // Method to clear all fields
                                                   public void clear() {
410, 124, 31);
      contentPane.add(btnAddUser);
                                                   studentIDField.setText("");
            // Close button to dispose
                                                   studentNameField.setText("");
                                                          contactField.setText("");
the window
            JButton btnClose = new
JButton("Close");
                                                   }
                                             }
```

8. Add Book Page:

```
public void run() {
package libraryManagementSystem;
                                                                   try {
import java.awt.EventQueue;
                                                addBookWindow frame = new
import javax.swing.ImageIcon;
                                             addBookWindow();
import javax.swing.JFrame;
import javax.swing.JLabel;
                                                frame.setVisible(true);
import javax.swing.JOptionPane;
                                                                   } catch
import javax.swing.JPanel;
                                             (Exception e) {
import
javax.swing.border.EmptyBorder;
                                                e.printStackTrace();
import java.awt.Toolkit;
                                                                   }
import javax.swing.JTextField;
                                                             }
import javax.swing.JButton;
                                                      });
import javax.swing.SwingConstants;
                                                /*********
import java.awt.Font;
import java.awt.Color;
                                             MAIN METHOD
                                             *****************************
import
java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
                                                 * Create the frame.
import java.sql.SQLException;
import java.awt.event.ActionEvent;
                                                public addBookWindow() {
import java.awt.SystemColor;
                                                      // GUI Setup
public class addBookWindow extends
                                                setIconImage(Toolkit.getDefaultT
                                             oolkit().getImage("C:\\Users\\Dell\\
JFrame {
                                             Desktop\\Library Management
   private static final long
                                             System\\Images\\addbook.png"));
serialVersionUID = 1L;
   private JPanel contentPane;
                                                setDefaultCloseOperation(JFrame.
   private JTextField bookIDField;
                                             DISPOSE ON CLOSE);
   private JTextField authorField;
                                                      setBounds(350, 150, 800,
   private JTextField genreField;
                                             530);
   private JLabel lblBookTItle;
                                                      contentPane = new
   private JTextField
                                             JPanel();
bookNameField;
                                                      contentPane.setBorder(new
   private JLabel lblBackground;
                                             EmptyBorder(5, 5, 5, 5));
   /**
    * Launch the application.
                                                setContentPane(contentPane);
                                                contentPane.setLayout(null);
   /*******
                                             LABELS ****************/
****************************/
   public static void main(String[]
                                                      JLabel lblTitle = new
                                             JLabel("
args) {
                                                        ADD BOOK");
         EventQueue.invokeLater(new
                                                      lblTitle.setIcon(new
Runnable() {
                                             ImageIcon("C:\\Users\\Dell\\Desktop\
```

```
\Library Management
                                                       lblGenre.setForeground(new
                                             Color(0, 64, 128));
System\\Images\\addbook1.png"));
         lblTitle.setForeground(new
                                                       lblGenre.setFont(new
Color(0, 64, 128));
                                             Font("Calibri", Font.BOLD, 19));
         lblTitle.setFont(new
                                                       lblGenre.setBounds(165,
Font("Tahoma", Font.BOLD, 18));
                                             316, 129, 31);
                                                       contentPane.add(lblGenre);
                                                       /************
   lblTitle.setHorizontalAlignment(
                                             LABELS ****************/
SwingConstants.CENTER);
         lblTitle.setBounds(165,
                                                       /***** TEXT
35, 415, 69);
                                             contentPane.add(lblTitle);
                                                       bookIDField = new
         lblBookTItle = new
                                             JTextField();
JLabel("Book Title:");
                                                       bookIDField.setFont(new
         lblBookTItle.setFont(new
                                             Font("Tahoma", Font.PLAIN, 14));
Font("Calibri", Font.BOLD, 19));
                                                bookIDField.setBackground(System
   lblBookTItle.setForeground(new
                                             Color.info);
Color(0, 64, 128));
                                                       bookIDField.setBounds(319,
                                             200, 261, 31);
   lblBookTItle.setBounds(165, 138,
158, 43);
                                                contentPane.add(bookIDField);
   contentPane.add(lblBookTItle);
                                                bookIDField.setColumns(10);
         JLabel lblID = new
                                                       authorField = new
JLabel("Book ID:");
                                             JTextField();
         lblID.setForeground(new
                                                       authorField.setFont(new
Color(0, 64, 128));
                                             Font("Tahoma", Font.PLAIN, 14));
         lblID.setFont(new
Font("Calibri", Font.BOLD, 19));
                                                authorField.setBackground(System
         lblID.setBounds(165, 201,
                                             Color.info);
90, 31);
                                                       authorField.setBounds(319,
         contentPane.add(lblID);
                                             257, 261,31);
         JLabel lblAuthor = new
                                                contentPane.add(authorField);
JLabel("Book Author:");
                                                authorField.setColumns(10);
   lblAuthor.setForeground(new
Color(0, 64, 128));
                                                       genreField = new
         lblAuthor.setFont(new
                                             JTextField();
Font("Calibri", Font.BOLD, 19));
                                                       genreField.setFont(new
         lblAuthor.setBounds(165,
                                             Font("Tahoma", Font.PLAIN, 14));
254, 129, 38);
                                                genreField.setBackground(SystemC
   contentPane.add(lblAuthor);
                                             olor.info);
                                                       genreField.setBounds(319,
         JLabel lblGenre = new
                                             315, 261, 31);
JLabel("Genre");
                                                contentPane.add(genreField);
```

```
genreField.setColumns(10);
                                             ERROR HANDLING
                                             ***********
         bookNameField = new
JTextField();
                                             (bookName.isEmpty() ||
         bookNameField.setFont(new
                                             bookID.isEmpty() || author.isEmpty()
Font("Tahoma", Font.PLAIN, 14));
                                             || genre.isEmpty()) {
   bookNameField.setBackground(Syst
                                                JOptionPane.showMessageDialog(nu
                                             ll, "Please Fill in All FIelds",
emColor.info);
                                             "Error",
   bookNameField.setBounds(319,
                                                JOptionPane.ERROR_MESSAGE);
144, 261, 31);
                                                                   } else {
   contentPane.add(bookNameField);
                                             Perform additional validations
   bookNameField.setColumns(10);
         /***** TEXT
                                                                          //
FIELDS ***************/
                                             Validate ID (it should be numeric)
         /***** BUTTONS
                                                if(!Library.isValidID(bookID)) {
*******************
         // Add Book Button
                                                JOptionPane.showMessageDialog(nu
                                             11, "Invalid ID Value.", "Error",
         JButton btnAdddBook = new
JButton("Add Book");
         // Button Actions
                                                JOptionPane.ERROR MESSAGE);
   btnAdddBook.addActionListener(ne
                                                return;
w ActionListener() {
                                                                          }
                public void
actionPerformed(ActionEvent e) {
                                                                          try {
                      // Retrieving
the user information from the text
                                             Check if genre and author's name
fields
                                             contains any numbers
                      String
                                             (author.matches(".*\\d.*")||genre.ma
bookName = bookNameField.getText();
                                             tches(".*\\d.*")) {
                      String bookID
= bookIDField.getText();
                      String author
                                             throw new
= authorField.getText();
                                             IllegalArgumentException();
                      String genre=
                                                                         }
genreField.getText();
                      // Removing
                                                                     } catch (
nay spaces in the book ID.
                                             IllegalArgumentException e1) {
   bookID=bookID.replace(" ", "");
                                             Display error message dialog box for
                                             invalid author's name
   /**********
                                             JOptionPane.showMessageDialog(null,
```

```
"Error: Genre/Author Name is not
                                                        btnAdddBook.setFont(new
valid.", "Error",
                                              Font("Tahoma", Font.BOLD, 15));
JOptionPane.ERROR_MESSAGE);
                                                       btnAdddBook.setBounds(319,
                            return;
                                              410, 124, 31);
                        }
                                                 contentPane.add(btnAdddBook);
Creating an Object of the Book as
                                                        // Dispose Window on close
per the provided details
                                              button
                                                        JButton btnClose = new
                             Book
                                              JButton("Close");
bookX = new Book();
   bookX.setBookTitle(bookName);
                                                 btnClose.addActionListener(new
                                              ActionListener() {
   bookX.setBookID(bookID);
                                                              public void
                                              actionPerformed(ActionEvent e) {
   bookX.setAuthor(author);
                                                                     dispose();
   bookX.setGenre(genre);
                                                 setVisible(false);
                                                              }
   bookX.setAvailability("Available
                                                        });
");
                                                        btnClose.setFont(new
                                              Font("Tahoma", Font.BOLD, 15));
                                                       btnClose.setForeground(new
                             //
Adding book to the database
                                              Color(0, 64, 128));
                                                       btnClose.setBounds(456,
                             if
(Library.addBookToDB(bookX)) {
                                              410, 124, 31);
                                                        contentPane.add(btnClose);
                                                        /****** BUTTONS
   JOptionPane.showMessageDialog(ro
otPane, "Record Saved", "Saved",
                                              ******************
JOptionPane.INFORMATION_MESSAGE);
                                                        /*********
                                              BACKGROUND ***************/
   clear();
                             }
                                                       lblBackground = new
                                              JLabel("");
Error message if the Book already
                                                       lblBackground.setIcon(new
exists based on the Book ID
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
                                              \Library Management
                             else
                                              System\\Images\\windowbackground.jpg
   JOptionPane.showMessageDialog(nu
                                              "));
11, "Record Already Exists",
                                                        lblBackground.setBounds(0,
"Error", JOptionPane.ERROR MESSAGE);
                                             0, 786, 493);
                                                 contentPane.add(lblBackground);
                       }
                                                        /*********
                                              BACKGROUND ***************/
                }
         });
                                                 }
   btnAdddBook.setForeground(new
Color(0, 64, 128));
                                                 public void clear() {
                                                       bookIDField.setText("");
```

```
bookNameField.setText("");
             authorField.setText("");
                                                    }
             genreField.setText("");
9. Search Book Page:
   package libraryManagementSystem;
                                                     * Launch the application.
                                                     */
   import java.awt.EventQueue;
                                                     /***** MAIN METHOD
                                                 *******************/
   import javax.swing.ImageIcon;
   import javax.swing.JFrame;
                                                    public static void main(String[]
   import javax.swing.JLabel;
                                                 args) {
   import javax.swing.JOptionPane;
                                                           EventQueue.invokeLater(new
   import javax.swing.JPanel;
                                                 Runnable() {
   import
                                                                  public void run() {
   javax.swing.border.EmptyBorder;
                                                                        try {
   import java.awt.Toolkit;
   import javax.swing.SwingConstants;
                                                    searchBookWindow frame = new
   import java.awt.Font;
                                                 searchBookWindow();
   import java.awt.Color;
   import javax.swing.JToggleButton;
                                                    frame.setVisible(true);
   import javax.swing.JTable;
                                                                        } catch
   import javax.swing.JTextField;
                                                 (Exception e) {
   import javax.swing.JButton;
   import javax.swing.JCheckBox;
                                                    e.printStackTrace();
   import
   java.awt.event.ActionListener;
                                                                  }
   import java.sql.Connection;
                                                           });
   import java.sql.PreparedStatement;
                                                    /***** MAIN METHOD
   import java.sql.ResultSet;
                                                 ***************/
   import java.sql.SQLException;
   import java.awt.event.ActionEvent;
   import java.awt.SystemColor;
                                                     * Create the frame.
   public class searchBookWindow
                                                     */
   extends JFrame {
                                                    public searchBookWindow() {
                                                           // GUI SETUP
      private static final long
   serialVersionUID = 1L;
      private JPanel contentPane;
                                                    setIconImage(Toolkit.getDefaultT
      private JTextField
                                                 oolkit()
   bookTitleField;
      private JTextField bookIDField;
                                                     .getImage("C:\\Users\\Dell\\Desk
      private JTextField
                                                 top\\Library Management
   bookAuthorField;
                                                 System\\Images\\datamining.png"));
      private JTextField
   bookGenreField;
                                                    setDefaultCloseOperation(JFrame.
      private JTextField
                                                 DISPOSE ON CLOSE);
   availabilityField;
                                                           setBounds(350, 150, 800,
                                                 530);
```

/**

```
contentPane = new
                                                       lblBookID.setFont(new
                                             Font("Calibri", Font.BOLD, 19));
JPanel();
         contentPane.setBorder(new
                                                       lblBookID.setBounds(169,
EmptyBorder(5, 5, 5, 5));
                                             202, 103, 24);
                                                contentPane.add(lblBookID);
   setContentPane(contentPane);
                                                       JLabel lblAuthor = new
                                             JLabel("Book Author:");
   contentPane.setLayout(null);
         /****** LABELS
                                                lblAuthor.setForeground(new
**************/
                                             Color(0, 64, 128));
         JLabel lblSearchBook = new
                                                       lblAuthor.setFont(new
JLabel("
                                             Font("Calibri", Font.BOLD, 19));
          SEARCH BOOK");
         lblSearchBook.setIcon(new
                                                       lblAuthor.setBounds(169,
ImageIcon("C:\\Users\\Dell\\Desktop\
                                             260, 123, 24);
\Library Management
System\\Images\\search.png"));
                                                contentPane.add(lblAuthor);
   lblSearchBook.setForeground(new
                                                       JLabel lblGenre = new
                                             JLabel("Genre:");
Color(0, 64, 128));
         lblSearchBook.setFont(new
                                                       lblGenre.setForeground(new
Font("Tahoma", Font.BOLD, 18));
                                             Color(0, 64, 128));
                                                       lblGenre.setFont(new
                                             Font("Calibri", Font.BOLD, 19));
   lblSearchBook.setHorizontalAlign
ment(SwingConstants.CENTER);
                                                       lblGenre.setBounds(169,
                                             316, 123, 35);
   lblSearchBook.setBounds(278, 44,
                                                       contentPane.add(lblGenre);
239, 66);
                                                       JLabel lblAvailability =
   contentPane.add(lblSearchBook);
                                             new JLabel("Availability");
         JLabel lblBookTitle = new
                                                lblAvailability.setForeground(ne
JLabel("Book Title:");
                                             w Color(0, 64, 128));
   lblBookTitle.setForeground(new
                                                lblAvailability.setFont(new
Color(0, 64, 128));
                                             Font("Calibri", Font.BOLD, 19));
         lblBookTitle.setFont(new
Font("Calibri", Font.BOLD, 19));
                                                lblAvailability.setBounds(169,
                                             372, 103, 35);
   lblBookTitle.setBounds(169, 144,
123, 24);
                                                contentPane.add(lblAvailability)
                                                       /****** LABELS
   contentPane.add(lblBookTitle);
                                             **************/
         JLabel lblBookID = new
                                                       /***** TEXT
JLabel("Book ID:");
                                             FIELDS ************/
   lblBookID.setForeground(new
                                                       bookTitleField = new
Color(0, 64, 128));
                                             JTextField();
```

```
bookTitleField.setFont(new
                                                       bookGenreField.setFont(new
Font("Tahoma", Font.PLAIN, 14));
                                             Font("Tahoma", Font.PLAIN, 14));
   bookTitleField.setBackground(Sys
                                                 bookGenreField.setBackground(Sys
                                             temColor.controlHighlight);
temColor.info);
   bookTitleField.setBounds(302,
                                                 bookGenreField.setEditable(false
140, 255, 31);
                                             );
   contentPane.add(bookTitleField);
                                                 bookGenreField.setBounds(302,
                                             317, 255, 31);
   bookTitleField.setColumns(10);
                                                 contentPane.add(bookGenreField);
         bookIDField = new
JTextField();
                                                 bookGenreField.setColumns(10);
         bookIDField.setFont(new
Font("Tahoma", Font.PLAIN, 14));
                                                       availabilityField = new
                                             JTextField();
   bookIDField.setBackground(System
Color.controlHighlight);
                                                 availabilityField.setFont(new
                                             Font("Tahoma", Font.PLAIN, 14));
   bookIDField.setEditable(false);
         bookIDField.setBounds(302,
                                                 availabilityField.setBackground(
198, 255, 31);
                                             SystemColor.controlHighlight);
   contentPane.add(bookIDField);
                                                 availabilityField.setEditable(fa
                                             lse);
   bookIDField.setColumns(10);
                                                 availabilityField.setBounds(302,
         bookAuthorField = new
                                             373, 255, 31);
JTextField();
                                                 contentPane.add(availabilityFiel
   bookAuthorField.setFont(new
                                             d);
Font("Tahoma", Font.PLAIN, 14));
                                                 availabilityField.setColumns(10)
   bookAuthorField.setBackground(Sy
                                                       /***** TEXT
stemColor.controlHighlight);
                                             FIELDS ***********/
   bookAuthorField.setEditable(fals
e);
                                                       /***** BUTTONS
                                             ********
   bookAuthorField.setBounds(302,
                                                       JButton btnSearch = new
256, 255, 31);
                                             JButton("Search");
   contentPane.add(bookAuthorField)
                                                 btnSearch.setBackground(new
                                             Color(176, 196, 222));
;
                                                       btnSearch.setFont(new
   bookAuthorField.setColumns(10);
                                             Font("Tahoma", Font.BOLD, 15));
         bookGenreField = new
                                                 btnSearch.addActionListener(new
JTextField();
                                             ActionListener() {
```

```
public void
actionPerformed(ActionEvent e) {
                                                  JOptionPane.showMessageDialog(nu
                       String
                                              11, "Book not Found.", "Error",
bookName = bookTitleField.getText();
                                               JOptionPane.ERROR_MESSAGE);
                                                                             }
                       // Checks if
                                                                      }
book name has been entered or not
                                                               }
                       if
                                                         });
(bookName.isEmpty()) {
                                                         btnSearch.setBounds(598,
                                              138, 103, 31);
   JOptionPane.showMessageDialog(nu
11, "Please Add Book Name", "Error",
                                                         // Reset button is used to
JOptionPane.ERROR MESSAGE);
                                               clear all text fields
                                                         JButton btnReset = new
                       } else {
                              // If
                                               JButton("Reset");
the book name field is not empty,
                                                         btnReset.setForeground(new
proceed to search for the book
                                              Color(0, 64, 128));
                                                         btnReset.setFont(new
Search for the book by its title
                                              Font("Tahoma", Font.BOLD, 15));
                              Book
foundBook = new Book();
                                                  btnReset.addActionListener(new
                                              ActionListener() {
   foundBook =
                                                               public void
Library.findBookbyTitle(bookName);
                                              actionPerformed(ActionEvent e) {
                                                                      clear();
                              //
                                                               }
Check if the book is found
                                                         });
                              if
                                                         btnReset.setBounds(302,
                                              433, 103, 31);
(foundBook != null) {
                                                         contentPane.add(btnReset);
   // If the book is found,
populate the fields with the book
                                                         // CLose button to dispose
details
                                              the window
                                                         JButton btnClose = new
   bookIDField.setText(foundBook.ge
                                               JButton("Close");
tBookID());
                                                         btnClose.setForeground(new
                                              Color(0, 64, 128));
   bookAuthorField.setText(foundBoo
                                                         btnClose.setFont(new
k.getAuthor());
                                              Font("Tahoma", Font.BOLD, 15));
   bookGenreField.setText(foundBook
                                                  btnClose.addActionListener(new
                                              ActionListener() {
.getGenre());
                                                               public void
   availabilityField.setText(foundB
                                              actionPerformed(ActionEvent e) {
ook.getAvailability());
                                                                      dispose();
                              } else
                                                               }
{
                                                         });
                                                         btnClose.setBounds(454,
                                              433, 103, 31);
   // If the book is not found,
display an error message
                                                         contentPane.add(btnClose);
```

```
BACKGROUND ********/
      contentPane.add(btnSearch);
            /***** BUTTONS
                                                    }
   **************
                                                    // Method to Clear all Fields
             /**********
                                                    public void clear() {
   BACKGROUND *********/
            JLabel background = new
                                                    bookTitleField.setText("");
   JLabel("");
                                                          bookIDField.setText("");
            background.setIcon(
                                                    bookAuthorField.setText("");
   ImageIcon("C:\\Users\\Dell\\Desktop\
   \Library Management
                                                    bookGenreField.setText("");
   System\\Images\\windowbackground.jpg
   "));
                                                    availabilityField.setText("");
             background.setBounds(0, 0,
   786, 493);
                                                }
      contentPane.add(background);
10.
      Display Books Page:
   package libraryManagementSystem;
                                                    private static final long
                                                serialVersionUID = 1L;
                                                    private JPanel contentPane;
   import java.awt.EventQueue;
   import javax.swing.ImageIcon;
                                                    // Establish connection with
   import javax.swing.JFrame;
   import javax.swing.JLabel;
                                                database
   import javax.swing.JOptionPane;
                                                    Connection con =
   import javax.swing.JPanel;
                                                Connect.ConnectToDB();
   import javax.swing.JScrollPane;
                                                    PreparedStatement pst=null;
                                                    ResultSet rs=null;
   import
   javax.swing.border.EmptyBorder;
                                                    private JTable table book;
   import java.awt.Toolkit;
                                                    private JScrollPane scrollPane;
   import java.sql.Connection;
   import java.sql.PreparedStatement;
   import java.sql.ResultSet;
   import java.sql.SQLException;
                                                     * Launch the application.
   import javax.swing.SwingConstants;
   import java.awt.Font;
                                                    /************ MAIN METHOD
   import java.awt.Color;
                                                 *************/
   import javax.swing.JTable;
                                                    public static void main(String[]
   javax.swing.table.DefaultTableModel;
                                                args) {
                                                          EventQueue.invokeLater(new
   import net.proteanit.sql.DbUtils;
                                                Runnable() {
                                                                 public void run() {
   public class displayBooksWindow
                                                                       try {
   extends JFrame {
                                                    displayBooksWindow frame = new
                                                displayBooksWindow();
```

```
// GUI SETUP
```

```
frame.setVisible(true);
                       } catch
                                                 setIconImage(Toolkit.getDefaultT
(Exception e) {
                                              oolkit().getImage("C:\\Users\\Dell\\
                                              Desktop\\Library Management
   e.printStackTrace();
                                              System\\Images\\display.png"));
                                                 setDefaultCloseOperation(JFrame.
                }
          });
                                              DISPOSE_ON_CLOSE);
                                                        setBounds(350, 150, 800,
                    MAIN METHOD
                                              530);
**************/
                                                        contentPane = new
                                              JPanel();
   // Function to update the table
                                                        contentPane.setBorder(new
by taking information from the
                                              EmptyBorder(5, 5, 5, 5));
database book
   private void UpdateTable() {
                                                 setContentPane(contentPane);
           // Prepare the SQL query
to select data from the book table
                                                 contentPane.setLayout(null);
           pst =
                                                        /****** LABELS
con.prepareStatement("SELECT
                                              *********
bookTitle, bookID, author, genre,
availability FROM book");
                                                        JLabel lblTitle = new
                                              JLabel("
                                                          DISPLAY BOOKS");
                                                        lblTitle.setForeground(new
           // Execute the query and
retrieve the ResultSet
                                              Color(0, 64, 128));
           rs = pst.executeQuery();
                                                        lblTitle.setFont(new
                                              Font("Tahoma", Font.BOLD, 18));
           // Update the table_book
                                                        lblTitle.setIcon(new
JTable with the fetched data
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
                                              \Library Management
table book.setModel(DbUtils.resultSe
                                              System\\Images\\displaybook1.png"));
tToTableModel(rs));
                                                 lblTitle.setHorizontalAlignment(
       } catch (SQLException e) {
                                              SwingConstants.CENTER);
           // Handle SQLException
                                                        lblTitle.setBounds(243,
                                              39, 284, 71);
JOptionPane.showMessageDialog(rootPa
                                                        contentPane.add(lblTitle);
ne, "", "",
JOptionPane.ERROR MESSAGE);
                                                        JLabel lblBookTitle = new
           e.printStackTrace();
                                              JLabel("Book Title");
       }
                                                        lblBookTitle.setFont(new
   }
                                              Font("Tahoma", Font.BOLD, 12));
                                                 lblBookTitle.setBounds(107, 125,
   /**
                                              68, 13);
    * Create the frame.
                                                 contentPane.add(lblBookTitle);
   public displayBooksWindow() {
```

```
/***** TABLE
           JLabel lblBookID = new
                                                      ***************/
JLabel("Book ID");
           lblBookID.setFont(new
                                                                  table_book = new JTable();
Font("Tahoma", Font.BOLD, 12));
           lblBookID.setBounds(230,
                                                          scrollPane.setColumnHeaderView(t
125, 63, 13);
                                                      able book);
                                                                  table_book.setModel(new
   contentPane.add(lblBookID);
                                                      DefaultTableModel(
                                                                          new Object[][] {
           JLabel lblAuthor = new
                                                                                  {"Java- How
JLabel("Author");
                                                      to Program", "1", "Paul & Dietel",
           lblAuthor.setFont(new
                                                       "Education", "Available"},
Font("Tahoma", Font.BOLD, 12));
                                                                                  {"Harry
                                                      Potter", "2", "JK Rowling",
           lblAuthor.setBounds(369,
125, 62, 13);
                                                      "Fantasy", "Available"},
                                                                                  {"The Hunger
                                                      Games", "3", "Suzanne Collins",
    contentPane.add(lblAuthor);
                                                      "Fiction", "Available"},
           JLabel lblGenre = new
                                                                                  {"Pride and
JLabel("Genre");
                                                      Prejudice", "5", "Jane Austen",
           lblGenre.setFont(new
                                                      "Romance", "Available"},
                                                                                  {null, null, null,
Font("Tahoma", Font.BOLD, 12));
                                                      null, null},
           lblGenre.setBounds(493,
                                                                                  {null, null, null,
                                                      null, null},
125, 68, 13);
                                                                                  {null, null, null,
                                                      null, null},
           contentPane.add(lblGenre);
                                                                                  {null, null, null,
                                                      null, null},
                                                                                  {null, null, null,
           JLabel lblAvailability =
                                                      null, null},
                                                                                  {null, null, null,
new JLabel("Availability");
                                                      null, null},
                                                                                  {null, null, null,
                                                      null, null},
    lblAvailability.setFont(new
                                                                                  {null, null, null,
                                                      null, null},
Font("Tahoma", Font.BOLD, 12));
                                                                                  {null, null, null,
                                                      null, null},
                                                                                  {null, null, null,
    lblAvailability.setHorizontalAli
                                                      null, null},
                                                                                  {null, null, null,
gnment(SwingConstants.CENTER);
                                                      null, null},
                                                                                  {null, null, null,
                                                      null, null},
    lblAvailability.setBounds(598,
                                                                                  {null, null, null,
                                                      null, null},
125, 76, 13);
                                                                                  {null, null, null,
                                                      null, null},
                                                                                  {null, null, null,
    contentPane.add(lblAvailability)
                                                      null, null},
                                                                                  {null, null, null,
                                                      null, null},
           /***** LABELS
                                                                                  {null, null, null,
                                                      null, null},
*************/
                                                                                  {null, null, null,
                                                      null, null},
                                                                                  {null, null, null,
           //SCROLL PANE
                                                      null, null},
                                                                                  {null, null, null,
           scrollPane = new
                                                      null, null},
                                                                                  {null, null, null,
JScrollPane();
                                                      null, null},
           scrollPane.setBounds(76,
                                                                                  {null, null, null,
                                                      null, null},
148, 628, 221);
                                                                                  {null, null, null,
                                                      null, null},
                                                                                  {null, null, null,
    contentPane.add(scrollPane);
                                                      null, null},
                                                                                  {null, null, null,
                                                      null, null},
```

```
{null, null, null,
null, null},
                                                                table_book.getColumnModel().getC
                              {null, null, null,
null, null},
                                                           olumn(0).setPreferredWidth(125);
                              {null, null, null,
null, null},
                              {null, null, null,
                                                                table book.getColumnModel().getC
null, null},
                              {null, null, null,
                                                           olumn(1).setPreferredWidth(55);
null, null},
                              {null, null, null,
null, null},
                              {null, null, null,
                                                                table_book.getColumnModel().getC
null, null},
                                                            olumn(2).setPreferredWidth(100);
                              {null, null, null,
null, null},
                                                                         /***** TABLE
                              {null, null, null,
null, null},
                              {null, null, null,
null, null},
                              {null, null, null,
null, null},
                                                                         /**********
                              {null, null, null,
null, null},
                                                            BACKGROUND *************/
                              {null, null, null,
null, null},
                                                                         JLabel lblBackground = new
                              {null, null, null,
                                                            JLabel("");
null, null},
                                                                         lblBackground.setIcon(new
                              {null, null, null,
null, null},
                                                            ImageIcon("C:\\Users\\Dell\\Desktop\\Library
                              {null, null, null,
                                                           Management
null, null},
                                                            System\\Images\\windowbackground.jpg"));
                              {null, null, null,
                                                                         lblBackground.setBounds(0, 0,
null, null},
                                                            786, 512);
                              {null, null, null,
null, null},
                                                                         contentPane.add(lblBackground);
                              {null, null, null,
                                                                         /***** BACKGROUND
null, null},
                                                            **************/
                              {null, null, null,
null, null},
                                                                         // Call the update table method
                              {null, null, null,
                                                            to fill the rows of the table
null, null},
                                                                        UpdateTable();
                     },
                     new String[] {
                                                                }
                              "bookTitle",
"bookID", "author", "genre",
"availability"
                     }
             ));
```

11. <u>Issue Book Page:</u>

```
/****** MAIN METHOD
package libraryManagementSystem;
                                              ***********/
                                                 public static void main(String[]
import java.awt.EventQueue;
                                              args) {
import javax.swing.ImageIcon;
                                                        EventQueue.invokeLater(new
import javax.swing.JFrame;
                                              Runnable() {
import javax.swing.JLabel;
import javax.swing.JOptionPane;
                                                               public void run() {
import javax.swing.JPanel;
                                                                     try {
import
javax.swing.border.EmptyBorder;
                                                 issueBookWindow frame = new
                                              issueBookWindow();
import java.awt.Toolkit;
import java.awt.Font;
                                                 frame.setVisible(true);
import java.awt.Color;
                                                                     } catch
import javax.swing.SwingConstants;
                                              (Exception e) {
import javax.swing.JTextField;
import javax.swing.JButton;
                                                 e.printStackTrace();
import
                                                                     }
java.awt.event.ActionListener;
                                                               }
import java.sql.Connection;
                                                        });
import java.sql.PreparedStatement;
                                                 }
import java.sql.ResultSet;
import java.sql.SQLException;
                                                 /***** MAIN METHOD
                                              ************/
import java.text.SimpleDateFormat;
                                                 /**
import java.util.Date;
                                                  * Create the frame.
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
public class issueBookWindow extends
                                                 public issueBookWindow() {
JFrame {
                                                        // GUI SETUP
   private static final long
                                                 setIconImage(Toolkit.getDefaultT
serialVersionUID = 1L;
                                              oolkit()
   private JPanel contentPane;
   private JTextField
                                                 .getImage("C:\\Users\\Dell\\Desk
bookNameField;
                                              top\\Library Management
   private JTextField bookIDField;
                                              System\\Images\\issuebook.png"));
   private JTextField
studentIDField;
                                                 setDefaultCloseOperation(JFrame.
   private JTextField
                                              DISPOSE ON CLOSE);
                                                        setBounds(350, 150, 800,
issueDateField;
   private JTextField dueDateField;
                                              530);
   private JTextField
                                                        contentPane = new
studentNameField;
                                              JPanel();
                                                        contentPane.setBorder(new
                                              EmptyBorder(5, 5, 5, 5));
    * Launch the application.
    */
                                                 setContentPane(contentPane);
```

```
lblStudentID.setForeground(new
   contentPane.setLayout(null);
                                             Color(0, 64, 128));
         /***** LABELS
                                                       lblStudentID.setFont(new
*************/
                                             Font("Calibri", Font.BOLD, 19));
         JLabel lblTitle = new
JLabel("
            ISSUE BOOK");
                                                 lblStudentID.setBounds(162, 240,
                                             102, 32);
   lblTitle.setHorizontalAlignment(
SwingConstants.CENTER);
                                                 contentPane.add(lblStudentID);
         lblTitle.setForeground(new
Color(0, 64, 128));
                                                       JLabel lblIssueDate = new
         lblTitle.setFont(new
                                             JLabel("Issue Date:");
Font("Tahoma", Font.BOLD, 18));
         lblTitle.setIcon(new
                                                 lblIssueDate.setForeground(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                             Color(0, 64, 128));
\Library Management
                                                       lblIssueDate.setFont(new
                                             Font("Calibri", Font.BOLD, 19));
System\\Images\\issuebook1.png"));
         lblTitle.setBounds(178,
40, 431, 83);
                                                 lblIssueDate.setBounds(162, 344,
         contentPane.add(lblTitle);
                                             102, 24);
         JLabel lblBookTitle = new
                                                 contentPane.add(lblIssueDate);
JLabel("Book Title:");
                                                       JLabel lblDueDate = new
   lblBookTitle.setForeground(new
                                             JLabel("Due Date:");
Color(0, 64, 128));
         lblBookTitle.setFont(new
                                                 lblDueDate.setForeground(new
Font("Calibri", Font.BOLD, 19));
                                             Color(0, 64, 128));
                                                       lblDueDate.setFont(new
   lblBookTitle.setBounds(162, 193,
                                             Font("Calibri", Font.BOLD, 19));
                                                       lblDueDate.setBounds(162,
102, 24);
                                             385, 102, 24);
   contentPane.add(lblBookTitle);
                                                 contentPane.add(lblDueDate);
         JLabel lblBookID = new
JLabel("Book ID:");
                                                       JLabel lblStudentName =
                                             new JLabel("Student Name:");
   lblBookID.setForeground(new
Color(0, 64, 128));
                                                 lblStudentName.setForeground(new
         lblBookID.setFont(new
                                             Color(0, 64, 128));
Font("Calibri", Font.BOLD, 19));
                                                       lblStudentName.setFont(new
         lblBookID.setBounds(162,
                                             Font("Calibri", Font.BOLD, 19));
142, 102, 24);
                                                 lblStudentName.setBounds(162,
   contentPane.add(lblBookID);
                                             294, 133, 32);
         JLabel lblStudentID = new
                                                 contentPane.add(lblStudentName);
JLabel("Student ID:");
                                                       /****** LABELS
                                             ************/
```

```
/***** TEXT
FIELDS **********/
                                                 studentNameField.setEditable(fal
         bookNameField = new
                                             se);
JTextField();
                                                 studentNameField.setFont(new
                                             Font("Tahoma", Font.PLAIN, 14));
   bookNameField.setEditable(false)
;
         bookNameField.setFont(new
                                                 studentNameField.setColumns(10);
Font("Tahoma", Font.PLAIN, 14));
                                                 studentNameField.setBackground(S
   bookNameField.setBackground(Syst
                                             ystemColor.controlHighlight);
emColor.controlHighlight);
                                                 studentNameField.setBounds(305,
   bookNameField.setBounds(305,
                                              291, 204, 31);
186, 204, 31);
                                                 contentPane.add(studentNameField
   contentPane.add(bookNameField);
                                             );
   bookNameField.setColumns(10);
                                                        issueDateField = new
                                              JTextField();
         bookIDField = new
JTextField():
                                                 issueDateField.setBackground(Sys
         bookIDField.setFont(new
                                              temColor.controlHighlight);
Font("Tahoma", Font.PLAIN, 14));
                                                        issueDateField.setFont(new
                                              Font("Tahoma", Font.PLAIN, 14));
   bookIDField.setBackground(System
Color.info);
                                                 issueDateField.setEditable(false
         bookIDField.setBounds(305,
                                             );
135, 204, 31);
                                                 issueDateField.setBounds(305,
   contentPane.add(bookIDField);
                                             337, 204, 31);
   bookIDField.setColumns(10);
                                                 contentPane.add(issueDateField);
         studentIDField = new
                                                 issueDateField.setColumns(10);
JTextField():
                                                        // Gets current date in
          studentIDField.setFont(new
Font("Tahoma", Font.PLAIN, 14));
                                             the format dd/MM/yyyy
                                                       SimpleDateFormat date =
   studentIDField.setBackground(Sys
                                             new SimpleDateFormat("dd/MM/yyyy");
temColor.info);
                                                       Date d = new Date();
   studentIDField.setBounds(305,
                                                 issueDateField.setText(date.form
237, 204, 31);
                                              at(d));
   contentPane.add(studentIDField);
                                                        dueDateField = new
                                              JTextField();
   studentIDField.setColumns(10);
                                                        dueDateField.setFont(new
                                              Font("Tahoma", Font.PLAIN, 14));
          studentNameField = new
JTextField();
```

```
dueDateField.setBackground(Syste
                                                 btnSearch1.addActionListener(new
mColor.info);
                                              ActionListener() {
                                                              public void
   dueDateField.setBounds(305, 378,
                                              actionPerformed(ActionEvent e) {
                                                                     String bookID
204, 31);
                                              = bookIDField.getText();
   contentPane.add(dueDateField);
                                                                     // Checks if
                                              ID has been provided
                                                                     if
   dueDateField.setColumns(10);
         /***** TEXT
                                              (bookID.isEmpty()) {
FIELDS ***********/
                                                 JOptionPane.showMessageDialog(nu
         /***** BUTTONS
                                              ll, "Please Add Book ID", "Error",
**************/
                                              JOptionPane.ERROR_MESSAGE);
         // Close button to dispose
window
                                                 return;
         JButton btnClose = new
                                                                     }
                                                                     // Input
JButton("Close");
         btnClose.setFont(new
                                              validation
Font("Tahoma", Font.BOLD, 15));
                                                                     if
         btnClose.setForeground(new
                                              (!Library.isValidID(bookID)) {
Color(0, 64, 128));
                                                 JOptionPane.showMessageDialog(nu
   btnClose.addActionListener(new
                                              11, "Invalid ID Value.", "Error",
                                              JOptionPane.ERROR MESSAGE);
ActionListener() {
                public void
actionPerformed(ActionEvent e) {
                                                 return;
                                                                     }
                       dispose();
                }
                                                                     // Create a
                                              new Book object to store the found
         });
         btnClose.setBounds(424,
                                              book
436, 85, 28);
                                                                     Book
         contentPane.add(btnClose);
                                              foundBook = new Book();
                                                                     // Search for
         // Search Button 1 is used
to get Book name when Book ID is
                                              the book by its ID using the
entered
                                              Library.findBookbyID() method
         JButton btnSearch1 = new
                                                                     foundBook =
JButton("Search");
                                              Library.findBookbyID(bookID);
   btnSearch1.setBackground(SystemC
                                                                     // Check if
olor.activeCaption);
                                              the book is found
                                                                     if (foundBook
   btnSearch1.setForeground(new
                                              != null) {
Color(0, 64, 128));
                                                                           // If
         btnSearch1.setFont(new
                                             the book is found, set the text of
Font("Tahoma", Font.BOLD, 15));
                                              the bookNameField to the book's
                                              title
```

```
bookNameField.setText(foundBook.
                                                  JOptionPane.showMessageDialog(nu
                                              11, "Invalid ID Value.", "Error",
getBookTitle());
                       } else {
                                               JOptionPane.ERROR_MESSAGE);
                              // If
the book is not found, display an
                                                  return;
error message
                                                                             }
   JOptionPane.showMessageDialog(nu
                                                                             //
11, "Book not Found.", "Error",
                                              Creates a User object and searches
JOptionPane.ERROR MESSAGE);
                                              for it in the Database
                                                                             User
                                              foundUser = new User();
                }
                                                  foundUser =
                                              Library.findUserbyID(studentID);
          });
          btnSearch1.setBounds(546,
135, 85, 32);
                                                                             //
                                              Check if the user is found
   contentPane.add(btnSearch1);
                                                                             if
                                               (foundUser != null) {
          // Search Button 2 is used
to get Student name when student id
                                                  // If the user is found, set the
is entered
                                              text of the studentNameField to the
          JButton btnSearch2 = new
                                              user's name
JButton("Search");
                                                  studentNameField.setText(foundUs
   btnSearch2.setBackground(SystemC
                                              er.getUserName());
olor.activeCaption);
                                                                             } else
                                              {
   btnSearch2.addActionListener(new
                                                  // If the user is not found,
ActionListener() {
                public void
                                              display an error message
actionPerformed(ActionEvent e) {
                                                  JOptionPane.showMessageDialog(nu
                       String
studentID =
                                               11, "User not Found.", "Error",
studentIDField.getText();
                                               JOptionPane.ERROR_MESSAGE);
                       // Checks if
                                                                             }
student ID has been entered or not
                                                                      }
                       if
                                                               }
(studentID.isEmpty()) {
                                                         });
   JOptionPane.showMessageDialog(nu
                                                  btnSearch2.setForeground(new
11, "Please Add Student ID",
                                              Color(0, 64, 128));
"Error", JOptionPane.ERROR MESSAGE);
                                                        btnSearch2.setFont(new
                       } else {
                                               Font("Tahoma", Font.BOLD, 15));
                              //
                                                        btnSearch2.setBounds(546,
Validates Student ID
                                              236, 85, 32);
(!Library.isValidID(studentID)) {
                                                  contentPane.add(btnSearch2);
```

```
// Issue Book Button
          JButton btnIssue = new
                                                  if (dueDate.matches(".*[a-zA-
JButton("Issue");
                                               Z].*")) {
          btnIssue.setForeground(new
                                                  JOptionPane.showMessageDialog(nu
Color(0, 64, 128));
          btnIssue.setFont(new
                                               11, "Invalid Due Date", "Error",
Font("Tahoma", Font.BOLD, 15));
                                               JOptionPane.ERROR_MESSAGE);
   btnIssue.addActionListener(new
                                                  return;
ActionListener() {
                public void
                                                  }
actionPerformed(ActionEvent e) {
                       String
bookName = bookNameField.getText();
                                               catch (IllegalArgumentException e2)
                       String bookID
= bookIDField.getText();
                                                  JOptionPane.showMessageDialog(nu
                       String
                                               ll, "", "Error",
studentName =
studentNameField.getText();
                                               JOptionPane.ERROR MESSAGE);
                       String
studentID =
                                                  return;
studentIDField.getText();
                                                                             }
                       String
                                                                      }
issueDate =
issueDateField.getText();
                                                                      // Checking
                                               if the book is available or not
                       String
dueDate = dueDateField.getText();
                                                                      boolean
                                               available = false;
                       // Checking
                                                                      Book
if all fields have been filled
                                               foundBook = new Book();
                                                                      foundBook =
                                               Library.findBookbyID(bookIDField.get
(bookName.isEmpty() ||
issueDate.isEmpty() ||
                                               Text());
studentName.isEmpty() ||
                                                                      if (foundBook
dueDate.isEmpty()) {
                                               != null) {
   JOptionPane.showMessageDialog(nu
                                                                             String
ll, "Please Fill in All Fields",
                                               availability =
                                               foundBook.getAvailability();
"Error",
   JOptionPane.ERROR MESSAGE);
                                                  available =
                                               (availability.equals("Available")) ?
                                               true : false;
   return;
                       } else {
                                                                      } else {
Checking if correct due date has
been entered
                                                  JOptionPane.showMessageDialog(nu
                                               11, "Book Availability not Found.",
                              try {
                                               "Error",
   // Checks if due date has any
alphabets
                                                  JOptionPane.ERROR_MESSAGE);
```

```
return;
                                                 return;
                       }
                       // Issue book
                                                                     } else {
if book is avaialable
                       if
                                              Display a message indicating that
(available) {
                                              the book is not available
the book is available, create a new
                                                  JOptionPane.showMessageDialog(ro
Book object to issue
                                              otPane, "Book Not Available", "Not
                              Book
                                              Available",
bookToIssue = new Book();
                                                 JOptionPane.ERROR MESSAGE);
                             // Set
the details of the book to be issued
                                                               }
   bookToIssue.setBookTitle(bookNam
e);
                                                        });
                                                        btnIssue.setBounds(305,
   bookToIssue.setBookID(bookID);
                                              436, 85, 28);
                                                        contentPane.add(btnIssue);
   bookToIssue.setUserID(studentID)
                                                        /****** BUTTONS
                                              ***************/
;
   bookToIssue.setIssueDate(issueDa
                                                        JLabel lblBackGround = new
                                              JLabel("");
te);
                                                        lblBackGround.setIcon(
   bookToIssue.setDueDate(dueDate);
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
                                              \Library Management
                              //
Attempt to issue the book using the
                                              System\\Images\\windowbackground.jpg
Library class method
                                              "));
                                                        lblBackGround.setBounds(0,
(Library.issueBook(bookToIssue)) {
                                              0, 786, 493);
   // Display a success message
                                                 contentPane.add(lblBackGround);
                                                 }
   JOptionPane.showMessageDialog(ro
otPane, "Book Issued", "",
                                                 // Method to clear all fields
JOptionPane.INFORMATION MESSAGE);
                                                 public void clear() {
   clear();
                                                        bookIDField.setText("");
                                                        bookNameField.setText("");
                             } else
{
                                                 studentIDField.setText("");
                                                        dueDateField.setText("");
   // Display an error message
   JOptionPane.showMessageDialog(ro
                                                 studentNameField.setText("");
otPane, "Failed to issue book", "",
JOptionPane.ERROR_MESSAGE);
                                                 }
                                              }
```

```
12.
      Return Book Page:
   package libraryManagementSystem;
                                                          EventQueue.invokeLater(new
                                                Runnable() {
   import java.awt.EventQueue;
                                                                 public void run() {
                                                                       try {
   import javax.swing.ImageIcon;
   import javax.swing.JFrame;
                                                    returnBookWindow frame = new
   import javax.swing.JLabel;
                                                returnBookWindow();
   import javax.swing.JOptionPane;
   import javax.swing.JPanel;
                                                    frame.setVisible(true);
   import
                                                                       } catch
   javax.swing.border.EmptyBorder;
                                                (Exception e) {
   import java.awt.Toolkit;
   import javax.swing.SwingConstants;
                                                    e.printStackTrace();
   import java.awt.Font;
                                                                       }
   import java.awt.Color;
                                                                 }
   import javax.swing.JTextField;
                                                          });
   import javax.swing.JButton;
                                                    /***** MAIN METHOD
   import
                                                 ************/
   java.awt.event.ActionListener;
   import java.sql.Connection;
   import java.sql.PreparedStatement;
                                                    /**
   import java.sql.ResultSet;
                                                     * Create the frame.
   import java.sql.SQLException;
   import java.awt.event.ActionEvent;
                                                    public returnBookWindow() {
   import java.awt.SystemColor;
                                                          // GUI SETUP
   public class returnBookWindow
                                                    setIconImage(Toolkit.getDefaultT
   extends JFrame {
                                                oolkit().getImage("C:\\Users\\Dell\\
                                                Desktop\\Library Management
                                                System\\Images\\return.png"));
      private static final long
   serialVersionUID = 1L;
      private JPanel contentPane;
                                                    setDefaultCloseOperation(JFrame.
      private JTextField bookIDField;
                                                DISPOSE_ON_CLOSE);
      private JTextField
                                                          setBounds(350, 150, 800,
   bookNameField;
                                                530);
      private JTextField
                                                          contentPane = new
   studentIDField;
                                                 JPanel();
      private JTextField dueDateField;
                                                          contentPane.setBorder(new
      private JTextField
                                                EmptyBorder(5, 5, 5, 5));
   issueDateField;
                                                    setContentPane(contentPane);
       * Launch the application.
       */
                                                    contentPane.setLayout(null);
      /***** MAIN METHOD
                                                          /***** LABELS
   *******/
                                                 ************/
      public static void main(String[]
                                                          JLabel lblTitle = new
                                                 JLabel("
                                                            RETURN BOOK");
   args) {
```

```
lblTitle.setIcon(new
ImageIcon("C:\\Users\\Dell\\Desktop\
                                                contentPane.add(lblDueDate);
\Library Management
System\\Images\\returnbook1.png"));
                                                       JLabel lblBookID = new
                                             JLabel("Book ID:");
         lblTitle.setForeground(new
Color(0, 64, 128));
         lblTitle.setFont(new
                                                lblBookID.setForeground(new
Font("Tahoma", Font.BOLD, 18));
                                             Color(0, 64, 128));
                                                       lblBookID.setFont(new
   lblTitle.setHorizontalAlignment(
                                             Font("Calibri", Font.BOLD, 19));
SwingConstants.CENTER);
                                                       lblBookID.setBounds(186,
         lblTitle.setBounds(186,
                                             141, 102, 24);
33, 395, 73);
         contentPane.add(lblTitle);
                                                contentPane.add(lblBookID);
         JLabel lblStudentName =
                                                       JLabel lblBookTitle = new
new JLabel("Student ID:");
                                             JLabel("Book Title:");
   lblStudentName.setForeground(new
                                                lblBookTitle.setForeground(new
Color(0, 64, 128));
                                             Color(0, 64, 128));
         lblStudentName.setFont(new
                                                       lblBookTitle.setFont(new
                                             Font("Calibri", Font.BOLD, 19));
Font("Calibri", Font.BOLD, 19));
   lblStudentName.setBounds(186,
                                                lblBookTitle.setBounds(186, 189,
241, 128, 21);
                                             102, 24);
   contentPane.add(lblStudentName);
                                                contentPane.add(lblBookTitle);
                                                       /***** LABELS
         JLabel lblIssueDate = new
                                             ************/
JLabel("Issue Date:");
                                                       /***** TEXT
   lblIssueDate.setForeground(new
                                             FIELDS ***********/
Color(0, 64, 128));
         lblIssueDate.setFont(new
                                                       bookIDField = new
Font("Calibri", Font.BOLD, 19));
                                             JTextField();
   lblIssueDate.setBounds(186, 298,
                                                bookIDField.setBackground(System
113, 21);
                                             Color.info);
                                                       bookIDField.setBounds(319,
   contentPane.add(lblIssueDate);
                                             137, 230, 31);
         JLabel lblDueDate = new
                                                contentPane.add(bookIDField);
JLabel("Due Date:");
                                                bookIDField.setColumns(10);
   lblDueDate.setForeground(new
                                                       bookNameField = new
Color(0, 64, 128));
         lblDueDate.setFont(new
                                             JTextField();
Font("Calibri", Font.BOLD, 19));
         lblDueDate.setBounds(186,
                                                bookNameField.setBackground(Syst
355, 113, 21);
                                             emColor.controlHighlight);
```

```
bookNameField.setEditable(false)
                                                 issueDateField.setBounds(319,
                                             292, 230, 31);
   bookNameField.setBounds(319,
                                                 contentPane.add(issueDateField);
185, 230, 31);
                                                 issueDateField.setColumns(10);
                                                       /***** TEXT
   contentPane.add(bookNameField);
                                             FIELDS **********/
   bookNameField.setColumns(10);
                                                       /***** BUTTONS
                                              **************/
         studentIDField = new
JTextField();
                                                       JButton btnReturn = new
                                              JButton("Return");
   studentIDField.setBackground(Sys
temColor.controlHighlight);
                                                 btnReturn.setForeground(new
                                             Color(0, 64, 128));
   studentIDField.setEditable(false
                                                       btnReturn.setFont(new
                                             Font("Tahoma", Font.BOLD, 15));
);
   studentIDField.setBounds(319,
                                                 btnReturn.addActionListener(new
                                             ActionListener() {
235, 230, 31);
                                                           public void
                                             actionPerformed(ActionEvent e) {
   contentPane.add(studentIDField);
                                                               // Get the book ID
   studentIDField.setColumns(10);
                                             entered by the user
                                                               String bookID =
         dueDateField = new
                                             bookIDField.getText();
JTextField();
                                                               // Check if the
   dueDateField.setBackground(Syste
                                             book ID is empty
mColor.controlHighlight);
                                              (bookID.isEmpty()) {
   dueDateField.setEditable(false);
                                              JOptionPane.showMessageDialog(rootPa
   dueDateField.setBounds(319, 349,
                                              ne, "Please Enter Book ID", "Error",
230, 31);
                                              JOptionPane.ERROR_MESSAGE);
                                                               }
   contentPane.add(dueDateField);
                                                               // Create a new
   dueDateField.setColumns(10);
                                             Book object to represent the book to
                                             be returned
         issueDateField = new
                                                               Book bookToReturn
JTextField();
                                             = new Book();
   issueDateField.setBackground(Sys
                                             bookToReturn.setBookID(bookID);
temColor.controlHighlight);
                                                               // Attempt to
   issueDateField.setEditable(false
                                             return the book using the Library
);
                                             class method
```

```
if
(Library.returnBook(bookToReturn)) {
                                                  btnSearch.setBackground(SystemCo
                      // Display a
                                              lor.activeCaption);
success message
                                                         btnSearch.setFont(new
                                              Font("Tahoma", Font.BOLD, 15));
JOptionPane.showMessageDialog(rootPa
ne, "Book Returned", "",
                                                  btnSearch.setForeground(new
JOptionPane.INFORMATION_MESSAGE);
                                              Color(0, 64, 128));
                      clear();
                                                  btnSearch.addActionListener(new
                  } else {
                      // Display an
                                              ActionListener() {
                                                               public void
error message
                                              actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(rootPa
                                                                   // Get the book
                                              ID entered by the user
ne, "Failed to return book", "",
JOptionPane.ERROR_MESSAGE);
                                                                   String bookID =
                                              bookIDField.getText();
                      return;
                  }
              }
                                                                   // Check if the
          });
                                              book ID is empty
                                                                   if
                                               (bookID.isEmpty()) {
          btnReturn.setBounds(319,
416, 93, 31);
                                               JOptionPane.showMessageDialog(null,
   contentPane.add(btnReturn);
                                               "Please Add Book ID", "Error",
                                               JOptionPane.ERROR MESSAGE);
          // Dispose Window on Close
                                                                        return;
button
          JButton btnClose = new
JButton("Close");
                                                                   // Validate the
          btnClose.setFont(new
                                              book ID (ensure it is numeric)
Font("Tahoma", Font.BOLD, 15));
          btnClose.setForeground(new
                                               (!Library.isValidID(bookID)) {
Color(0, 64, 128));
                                               JOptionPane.showMessageDialog(null,
   btnClose.addActionListener(new
                                               "Invalid ID Value.", "Error",
                                               JOptionPane.ERROR_MESSAGE);
ActionListener() {
                public void
                                                                       clear();
actionPerformed(ActionEvent e) {
                                                                       return;
                       dispose();
                                                                   }
          });
                                                                   Boolean
         btnClose.setBounds(464,
                                              available = false;
416, 85, 31);
          contentPane.add(btnClose);
                                                                   // Create a new
                                              Book object to store the found book
                                              details
          // Search Button
          JButton btnSearch = new
                                                                   Book foundBook
JButton("Search");
                                              = new Book();
                                              foundBook.setBookID(bookID);
```

```
JOptionPane.showMessageDialog(rootPa
                    // Retrieve
book details from the Library using
                                              ne, "Book was not Issued", "",
the book ID
                                              JOptionPane.ERROR_MESSAGE);
                    foundBook =
                                                                      clear();
Library.findBookbyID(bookIDField.get
                                                                      return;
Text());
                                                                  }
                                                              }
                    // Check if the
book was found
                    if (foundBook
                                                       btnSearch.setBounds(579,
!= null) {
                                              135, 93, 31);
                        // If
found, populate the fields with the
                                                 contentPane.add(btnSearch);
                                                        /***** BUTTONS
book details
bookNameField.setText(foundBook.getB
                                                       /**********
ookTitle());
                                              BACKGROUND **********/
studentIDField.setText(foundBook.get
                                                       JLabel background = new
                                              JLabel("");
UserID());
                                                       background.setIcon(
issueDateField.setText(foundBook.get
                                              ImageIcon("C:\\Users\\Dell\\Desktop\
IssueDate());
                                              \Library Management
dueDateField.setText(foundBook.getDu
                                              System\\Images\\windowbackground.jpg
eDate());
                                              "));
                                                       background.setBounds(0, 0,
                        // Check if
                                              786, 493);
the book is available for issuance
                        available =
                                                 contentPane.add(background);
                                                       /***********
foundBook.getAvailability().equals("
                                              BACKGROUND **********/
Available");
                    } else {
                                                 }
                        // If the
book was not found, display an error
                                                 // Clear all Fields
message
                                                 public void clear() {
                                                        bookIDField.setText("");
JOptionPane.showMessageDialog(null,
"Book not Found.", "Error",
                                                 studentIDField.setText("");
JOptionPane.ERROR MESSAGE);
                                                       bookNameField.setText("");
                                                       dueDateField.setText("");
                        clear();
                        return;
                                                 issueDateField.setText("");
                                                 }
                    // If the book
                                              }
is available for issuance
                    if (available)
{
```

CONCLUSION

The Library Management System, developed in Java with Eclipse and MySQL integration, offers an efficient solution for librarians. With its user-friendly GUI and essential functionalities like adding users and books, searching, issuing, and returning books, it streamlines library operations. Testing ensures reliability, and error handling notifies users of any issues.

1. References:

- a. Utilize resources like Geek for Geeks for comprehensive study materials.
- b. Explore YouTube tutorials for step-by-step guidance on setting up libraries and learning SQL basics.
- c. Leverage GitHub for a wide range of ideas and solutions to enhance and expand the functionality of the system.