Bulacan State University
City of Malolos, Bulacan

College of Information and Communications Technology
Information Technology Department

Library Management System

Project Title:

FINAL PROJECT

in

IT 203

OBJECT-ORIENTED PROGRAMMING 1

Submitted by:

DIMLA, Tyron B.

MENDOZA, Reymark M.

OXALES, Emmanuel D.G.

QUIÑONES JR, Romeo M.

TAGANAS, Lorenz Romeo O.

BSIT 2G – G1

Submitted to:

ANIAG, Jennylyn L.

Library Management System

Our Library Management System (LMS) is a comprehensive and user-friendly platform designed to streamline the management of library resources and facilitate easy access for students, teachers, staff, and librarians. The system is built with a user-centric approach, ensuring a seamless experience for all users.

Key Functionalities:

Sign Up:

- **New Member Registration:** Prospective users, including students, teachers, and staff, can easily register for library access by providing essential details such as name, email, and a unique username/password combination.

Login:

- **Secure Access:** Registered members can log in securely using their unique username and password, ensuring that only authorized individuals can access their library accounts.
- **Dashboard Access:** Upon successful login, users are directed to a personalized dashboard where they can explore the library catalog, manage borrowed books, and access other relevant features.

Librarian:

- **Manage Books:** Librarians can efficiently manage the library's collection, including adding new books, updating existing book information, and deleting/removing book records.
- **Manage Members:** Librarians have the ability to manage the library's user database. This includes adding new members, updating member information, and deleting or removing members when necessary.
- **Manage Borrowed Books:** Librarians can oversee and track borrowed books, handle book returns..

Student/Teacher/Staff:

- **View Books:** Users can browse the library catalog, search for specific books, and view detailed information about each book, including availability status.
- **Borrow Books:** Members can easily borrow books by double-clicking on a specific book. After clicking the "Borrow" option, the borrowing transaction will be stored in the database.
- **View Borrowed Books:** Users have access to a personalized dashboard where they can view a list of books they have borrowed, check due dates.

Key Features:

Search and Filters:

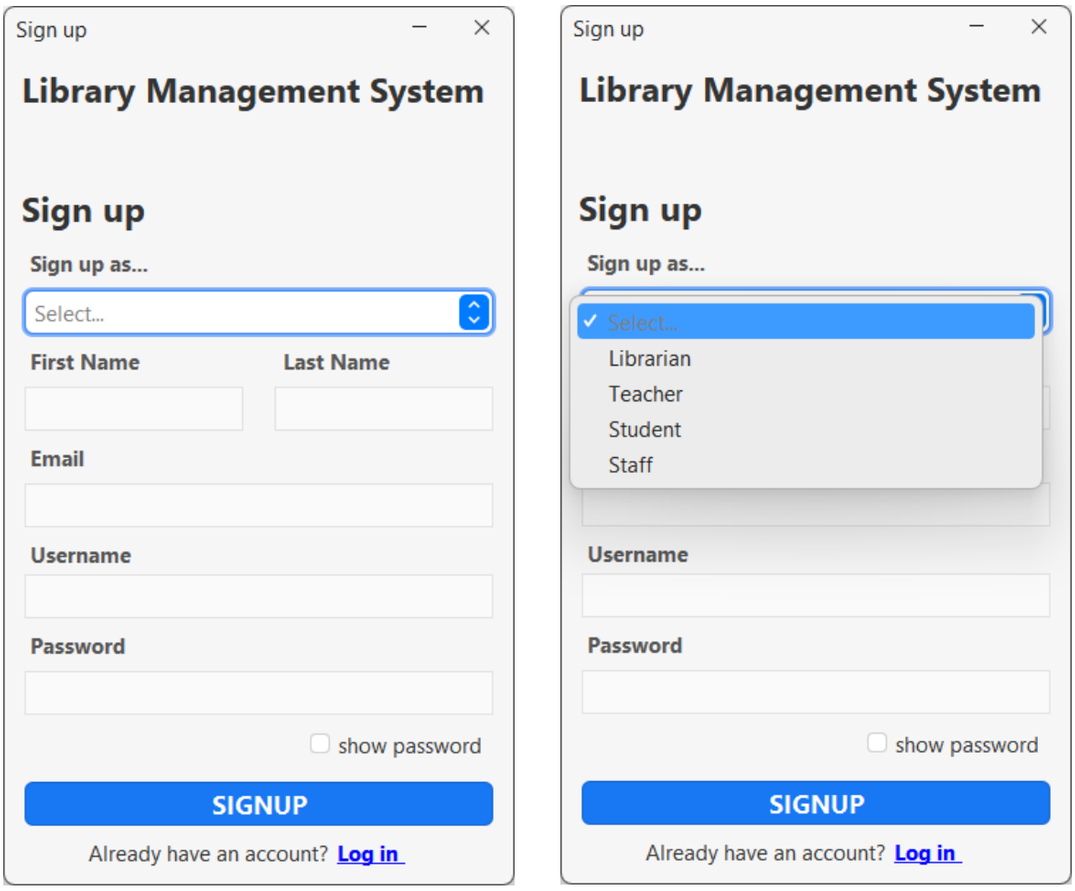
- **Search and Filters:** A robust search functionality allows users to quickly find books based on titles, authors, genres, or keywords. Filters enable precise searches, making it convenient for users to locate specific materials.
- **User-Friendly Interface:** The system boasts an intuitive and user-friendly interface, ensuring a smooth navigation experience for both librarians and library members.

Benefits:

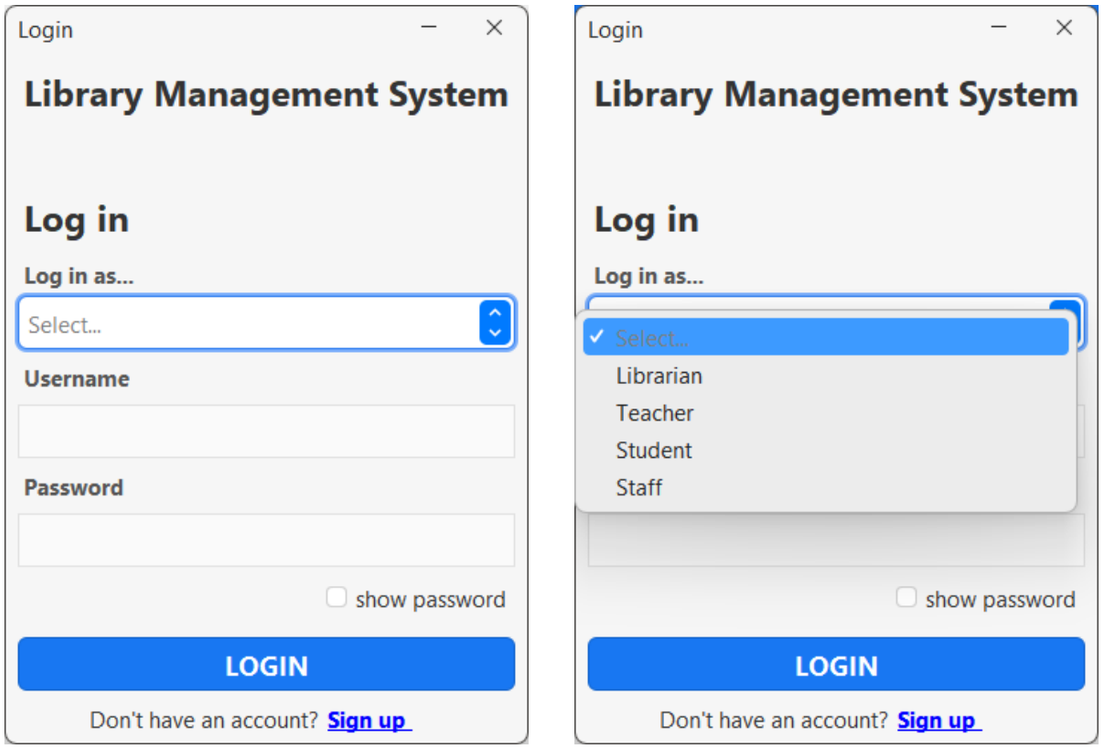
- **Efficiency:** Streamlines library operations, reducing manual effort in book management.
- **Accessibility:** Enhances accessibility to the library's resources for all users, promoting a culture of learning.
- **Transparency:** Provides transparency in borrowing processes, allowing users to monitor their activity and librarians to track library usage.

Our Library Management System is a powerful tool designed to create a harmonious environment for both librarians and library users, fostering a culture of knowledge sharing and exploration.

Log in/Sing up

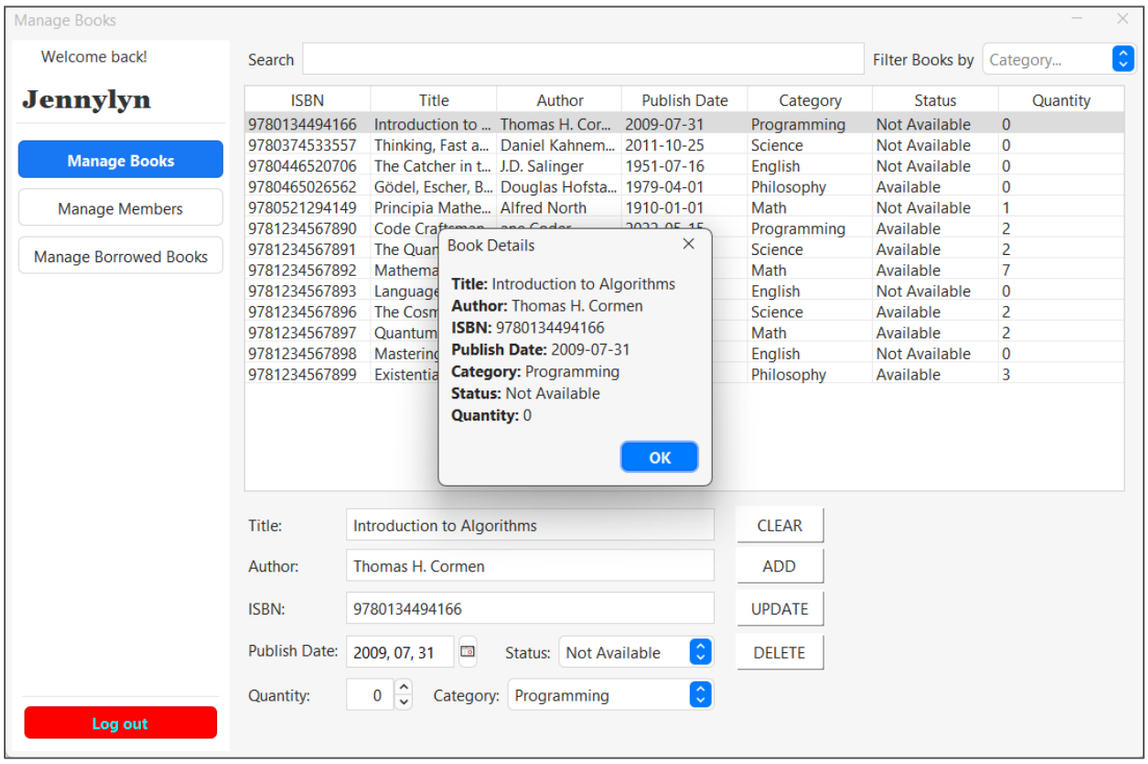


Our Library Management System offers a user-friendly **Sign Up** feature, allowing individuals to register. Users can sign up based on their roles, choosing from options such as Librarian, Teacher, Student or Staff.

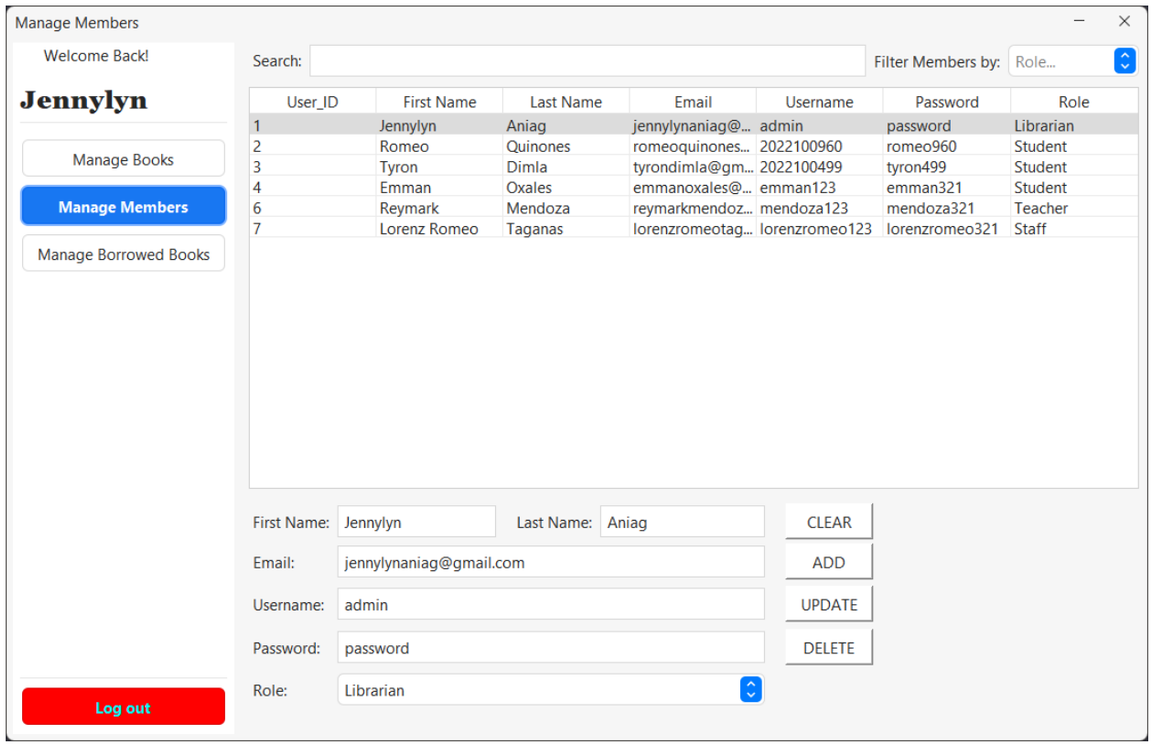


Our Library Management System provides a straightforward and secure **Log in** feature, ensuring seamless access for users based on their roles: Librarian, Student, Teacher, or Staff.

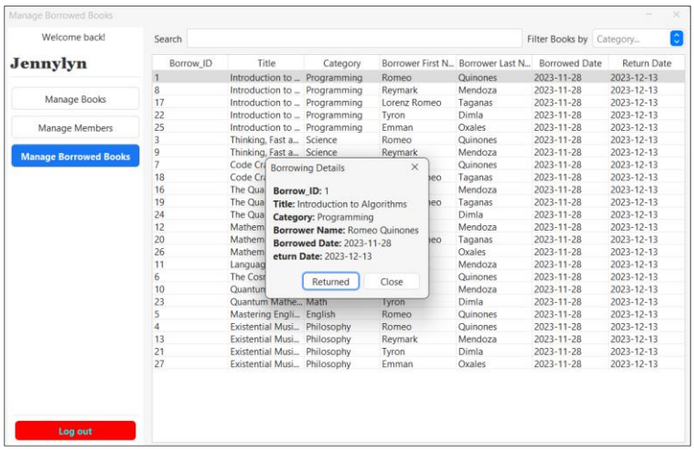
Librarian



On the **"Manage Books"**, the role of the librarian encompasses efficient control and organization of the library's collection. Librarians have the capability to **add, update, and delete book records**, ensuring a well-maintained and up-to-date catalog for the library's users.



On the **"Manage Members"** empowers librarians with comprehensive tools to oversee and maintain member records within the Library Management System. **Librarians have the capability to add, update, and delete member records.**



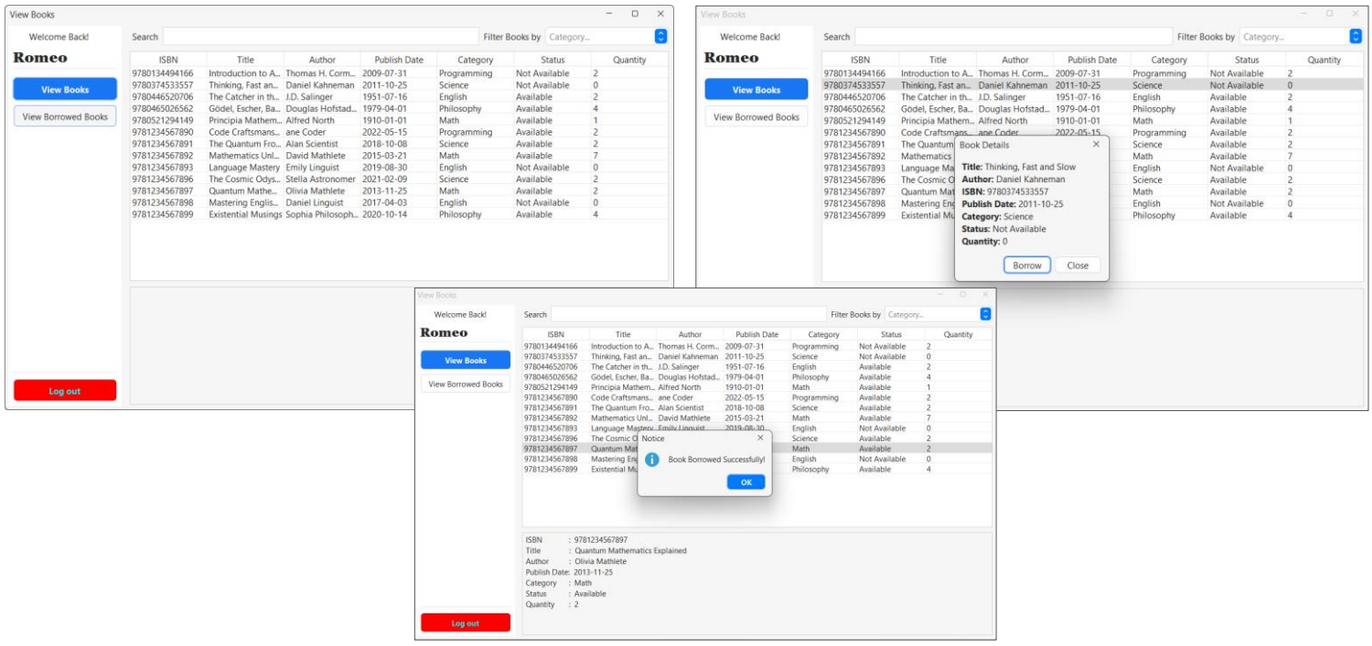
On the **"Manage Borrowed Books", librarians** can click on specific borrowing records within the system to access detailed information about the borrowed book and borrower.

Upon selecting a record, an option dialog promptly appears, presenting the librarian with choices, including the option to mark the book as **"Returned"**.

Choose to mark a book as "Returned" triggers a secondary confirmation dialog designed to verify if the book has indeed been returned. This dialog includes a question: "Has this book already been returned?".

The Librarian can choose **"Yes" to confirm that the book has been returned**, initiating the removal of the borrowing record from the database.

User (Teacher, Student, Staff)

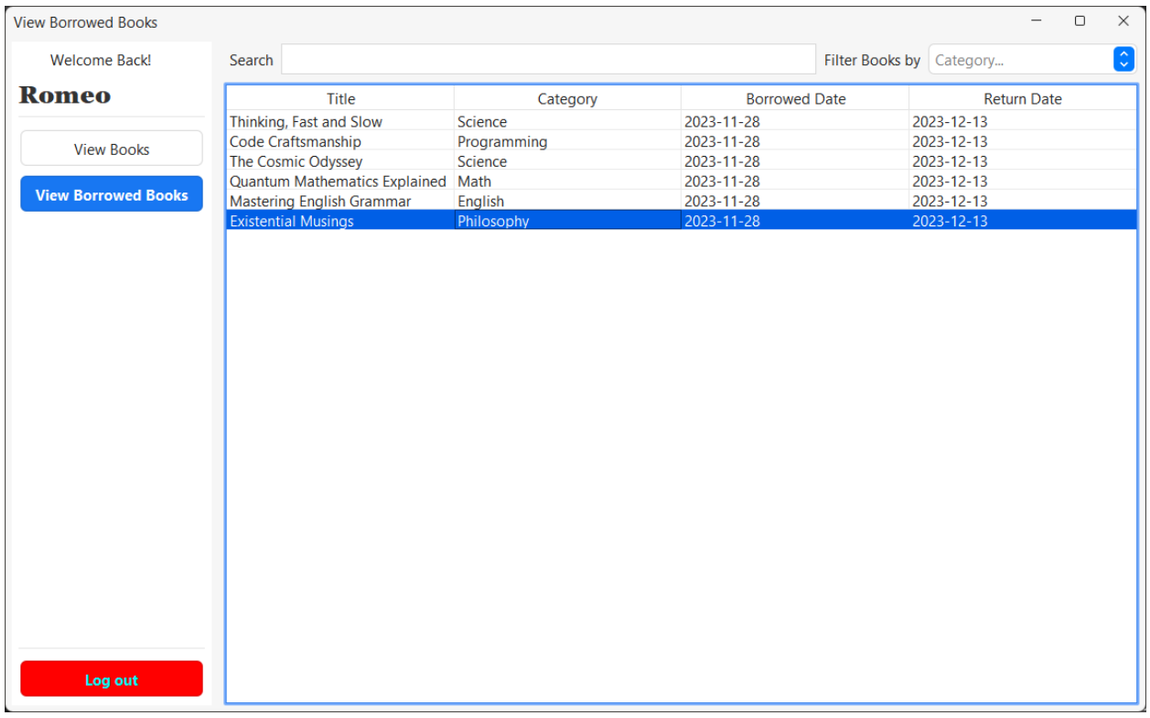


On the **"View Books ", users (Teacher, Student, Staff)** can effortlessly navigate through a user-friendly interface to view a comprehensive list of books categorized based on subjects like Programming, Science, Math, English, and Philosophy.

By clicking on a specific book record, users trigger an interactive option dialog that provides details about the selected book, including the title, author, ISBN, publish date, category, availability status, and quantity.

Upon deciding to borrow a book, users can choose the "Borrow" option from the interactive dialog. A message dialog promptly appears, confirming the successful borrowing of the selected book.

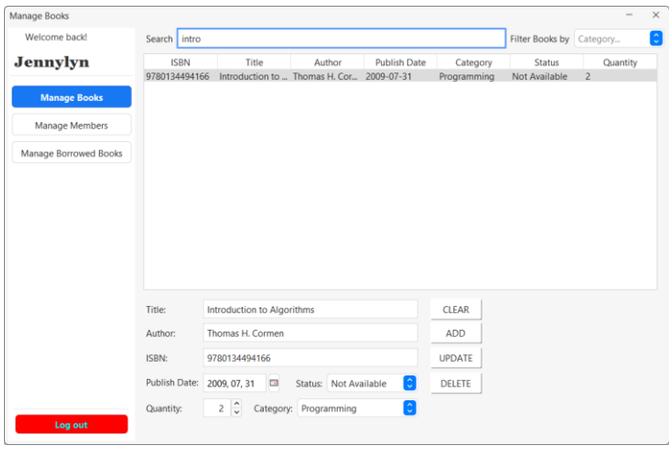
The system dynamically stores the borrowing transaction in the database, associating it with the borrower's unique ID. This ensures accurate and real-time tracking of borrowed books per user.

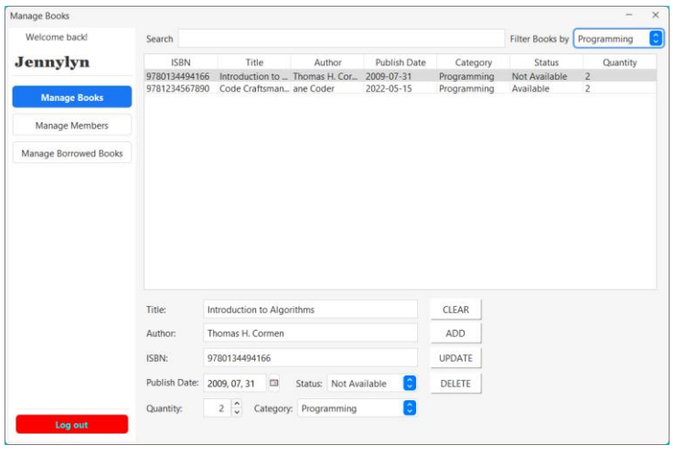


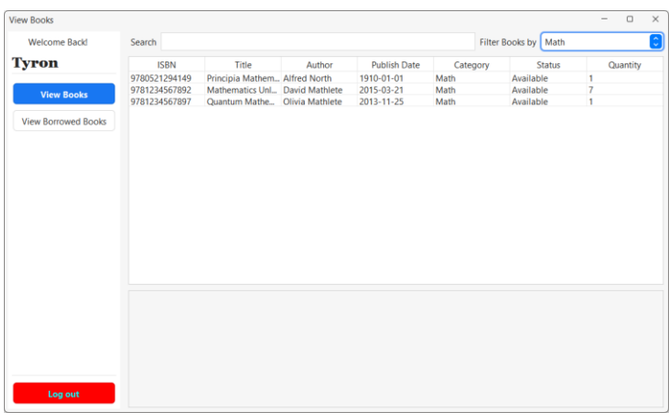
For Users, the **"View Borrowed Books"** feature provides a convenient way to manage and monitor their borrowed book transactions within the Library Management System.

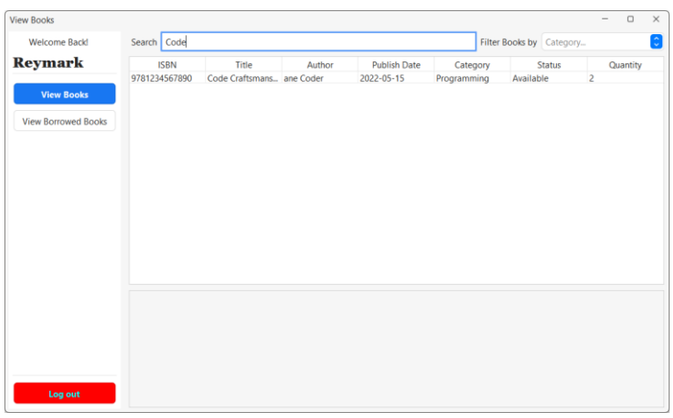
Users can access comprehensive details for each borrowed book, including title, category, and borrowing date. **The return date is set to 15 days** from the borrowing date, offering a clear timeframe for returning the book.

Search and Filter









Source Code

DatabaseConnection Class

```
import java.sql.*;
public class DatabaseConnection {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/library_management_system";
    private static final String DB_USERNAME = "root";
    private static final String DB_PASSWORD = "2022100960Romeo";
    private static Connection connection;

    public static Connection getConnection(){
        try {
            connection = DriverManager.getConnection(DB_URL, DB_USERNAME, DB_PASSWORD);
        } catch (SQLException ex) {
            System.err.println("Database Connection: " + ex.getMessage());
        }
        return connection;
    }
}
```

SignUpFrame Class

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import java.awt.event.ItemEvent;
import javax.swing.*;
import java.sql.*;
public class SignUpFrame extends javax.swing.JFrame {
    Connection connection = DatabaseConnection.getConnection();

    public SignUpFrame() {
        initComponents();
        all_InputFieldSetEnabled(false);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        firstNameTextField = new javax.swing.JTextField();
        signUpButton = new javax.swing.JButton();
        showPasswordCheckBox = new javax.swing.JCheckBox();
        logInLink = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        accountRoleComboBox = new javax.swing.JComboBox<>();
        jLabel7 = new javax.swing.JLabel();
        lastNameTextField = new javax.swing.JTextField();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        passwordField = new javax.swing.JPasswordField();
        jLabel6 = new javax.swing.JLabel();
        usernameTextField = new javax.swing.JTextField();
        emailTextField = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Sign up");
        setLocation(new java.awt.Point(800, 200));
        setResizable(false);

        jPanel1.setForeground(new java.awt.Color(249, 249, 249));
```

```
        jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(51, 51, 51));
        jLabel1.setText("Library Management System");

        jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(51, 51, 51));
        jLabel2.setText("Sign up");

        signUpButton.setBackground(new java.awt.Color(24, 119, 242));
        signUpButton.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
        signUpButton.setForeground(new java.awt.Color(255, 255, 255));
        signUpButton.setText("SIGNUP");
        signUpButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                signUpButtonActionPerformed(evt);
            }
        });

        showPasswordCheckBox.setForeground(new java.awt.Color(51, 51, 51));
        showPasswordCheckBox.setText("show password");
        showPasswordCheckBox.setVerticalAlignment(javax.swing.SwingConstants.BOTTOM);
        showPasswordCheckBox.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                showPasswordCheckBoxItemStateChanged(evt);
            }
        });

        logInLink.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logInLink.setForeground(new java.awt.Color(0, 0, 255));
        logInLink.setText("<html><u> Log in </u><html>");
        logInLink.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                logInLinkMouseClicked(evt);
            }
        });

        jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(85, 85, 85));
        jLabel3.setText("Email");

        jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(85, 85, 85));
        jLabel4.setText("First Name");

        accountRoleComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Select...</font></html>", "Librarian", "Teacher", "Student", "Staff"
}));
        accountRoleComboBox.setOpaque(true);
        accountRoleComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                accountRoleComboBoxActionPerformed(evt);
            }
        });

        jLabel7.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel7.setForeground(new java.awt.Color(85, 85, 85));
        jLabel7.setText("Sign up as...");

        jLabel8.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel8.setForeground(new java.awt.Color(85, 85, 85));
        jLabel8.setText("Last Name");

        jLabel9.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel9.setForeground(new java.awt.Color(85, 85, 85));
        jLabel9.setText("Password");

        jLabel10.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
```

```
        jLabel10.setForeground(new java.awt.Color(85, 85, 85));
        jLabel10.setText("Username");

        jLabel6.setForeground(new java.awt.Color(51, 51, 51));
        jLabel6.setText("Already have an account?");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(passwordField,
javax.swing.GroupLayout.Alignment.TRAILING)
                            .addComponent(signUpButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(accountRoleComboBox, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(firstNameTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 158, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(6, 6, 6)
                                        .addComponent(jLabel4)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(lastNameTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 158, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                                        .addComponent(jLabel8)
                                        .addGap(75, 75, 75))))
                            .addComponent(usernameTextField)
                            .addComponent(emailTextField)
                            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel1)
                                    .addComponent(jLabel2)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(6, 6, 6)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .addComponent(jLabel3)
                                            .addComponent(jLabel7))))
                                .addGap(0, 8, Short.MAX_VALUE)))
                        .addContainerGap())
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(6, 6, 6)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel9)
                            .addComponent(jLabel10))
                        .addGap(0, 0, Short.MAX_VALUE))))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
```

```
                .addGap(0, 0, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel6)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(logInLink, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(63, 63, 63))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(showPasswordCheckBox)
                        .addGap(14, 14, 14))))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(52, 52, 52)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel7)
                .addGap(7, 7, 7)
                .addComponent(accountRoleComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel8)
                    .addComponent(jLabel4))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(lastNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(firstNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(emailTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel10)
                .addGap(2, 2, 2)
                .addComponent(usernameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel9)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(passwordField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(showPasswordCheckBox)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(signUpButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(logInLink, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
```

```
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void showPasswordCheckBoxItemStateChanged(java.awt.event.ItemEvent evt) {
        int state = evt.getStateChange();

        if (state == ItemEvent.SELECTED){
            passwordField.setEchoChar((char) 0);
        }
        else {
            passwordField.setEchoChar('•');
        }
    }

    private void logInLinkMouseClicked(java.awt.event.MouseEvent evt) {
        new LogInFrame().setVisible(true);
        dispose();
    }

    private void signUpButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String accountRole = accountRoleComboBox.getSelectedItem().toString();
        String firstName = firstNameTextField.getText();
        String lastName = lastNameTextField.getText();
        String email = emailTextField.getText();
        String username = usernameTextField.getText();
        String password = String.valueOf(passwordField.getPassword());

        int userId = 0;
        if (firstName.isBlank() ||
                lastName.isBlank() ||
                email.isBlank() ||
                username.isBlank() ||
                password.isBlank()) {

            JOptionPane.showMessageDialog(this, "Fill out needed information", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        String insertQuery = "INSERT INTO account (first_name, last_name, email, username,
password, role, user_id) " +
                "VALUES(?, ?, ?, ?, ?, ?, ?)";

        try (PreparedStatement account = connection.prepareStatement(insertQuery)) {
            account.setString(1, firstName);
            account.setString(2, lastName);
            account.setString(3, email);
```

```
            account.setString(4, username);
            account.setString(5, password);
            account.setString(6, accountRole);
            account.setInt(7, userId);
            account.executeUpdate();

            JOptionPane.showMessageDialog(this, "Sign up Succesful", "Notice",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException ex) {
            System.out.println("Sign up Operation: " + ex.getMessage());
        }

        clearUserInput();
    }

    private void clearUserInput() {
        accountRoleComboBox.setSelectedIndex(0);
        firstNameTextField.setText("");
        lastNameTextField.setText("");
        emailTextField.setText("");
        usernameTextField.setText("");
        passwordField.setText("");
    }

    private void accountRoleComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        final int SELECT = 0;
        final int ROLE = accountRoleComboBox.getSelectedIndex();
        if (ROLE == SELECT) {
            all_InputFieldSetEnabled(false);
        }
        else {
            all_InputFieldSetEnabled(true);
        }
    }

    private void all_InputFieldSetEnabled(boolean decision){
        firstNameTextField.setEnabled(decision);
        lastNameTextField.setEnabled(decision);
        emailTextField.setEnabled(decision);
        usernameTextField.setEnabled(decision);
        passwordField.setEnabled(decision);
    }

    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new SignUpFrame().setVisible(true);
            }
        });
    }


    // Variables declaration - do not modify
    private javax.swing.JComboBox<String> accountRoleComboBox;
    private javax.swing.JTextField emailTextField;
    private javax.swing.JTextField firstNameTextField;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
```

```
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField lastNameTextField;
    private javax.swing.JLabel logInLink;
    private javax.swing.JPasswordField passwordField;
    private javax.swing.JCheckBox showPasswordCheckBox;
    private javax.swing.JButton signUpButton;
    private javax.swing.JTextField usernameTextField;
    // End of variables declaration
}
```

LogInFrame Class

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import java.awt.event.ItemEvent;
import javax.swing.*;
import java.sql.*;
public class LogInFrame extends javax.swing.JFrame {
    private Connection connection =  DatabaseConnection.getConnection();
    public String _firstName;
    public int _userId;

    public LogInFrame() {
        initComponents();
        userNameTextField.setEnabled(false);
        passwordField.setEnabled(false);
    }


    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        java.awt.GridBagConstraints gridBagConstraints;

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        userNameTextField = new javax.swing.JTextField();
        logInButton = new javax.swing.JButton();
        showPasswordCheckBox = new javax.swing.JCheckBox();
        jLabel5 = new javax.swing.JLabel();
        signUpLink = new javax.swing.JLabel();
        passwordField = new javax.swing.JPasswordField();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        accountRoleComboBox = new javax.swing.JComboBox<>();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("<html><b><font color='#333333'>Log in</font></b></html>");
        setLocation(new java.awt.Point(800, 250));
        setResizable(false);
        getContentPane().setLayout(new java.awt.GridBagLayout());

        jPanel1.setForeground(new java.awt.Color(249, 249, 249));

        jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(51, 51, 51));
        jLabel1.setText("Library Management System");

        jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(51, 51, 51));
        jLabel2.setText("Log in");

        userNameTextField.setForeground(new java.awt.Color(51, 51, 51));

        logInButton.setBackground(new java.awt.Color(24, 119, 242));
        logInButton.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
        logInButton.setForeground(new java.awt.Color(255, 255, 255));
        logInButton.setText("LOGIN");
        logInButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logInButtonActionPerformed(evt);
            }
        });
```

```
        showPasswordCheckBox.setForeground(new java.awt.Color(51, 51, 51));
        showPasswordCheckBox.setText("show password");
        showPasswordCheckBox.setVerticalAlignment(javax.swing.SwingConstants.BOTTOM);
        showPasswordCheckBox.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                showPasswordCheckBoxItemStateChanged(evt);
            }
        });

        jLabel5.setForeground(new java.awt.Color(51, 51, 51));
        jLabel5.setText("Don't have an account?");

        signUpLink.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        signUpLink.setForeground(new java.awt.Color(0, 0, 255));
        signUpLink.setText("<html><u> Sign up </u><html>");
        signUpLink.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                signUpLinkMouseClicked(evt);
            }
        });

        passwordField.setForeground(new java.awt.Color(51, 51, 51));

        jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(85, 85, 85));
        jLabel3.setText("Password");

        jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(85, 85, 85));
        jLabel4.setText("Username");

        jLabel6.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        jLabel6.setForeground(new java.awt.Color(85, 85, 85));
        jLabel6.setText("Log in as...");

        accountRoleComboBox.setForeground(new java.awt.Color(51, 51, 51));
        accountRoleComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Select...</font></html>", "Librarian", "Teacher", "Student", "Staff"
}));
        accountRoleComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                accountRoleComboBoxActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(logInButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(userNameTextField)
            .addComponent(passwordField)
            .addComponent(accountRoleComboBox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel2)
                                    .addComponent(jLabel1)))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGap(50, 50, 50)
                                .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(signUpLink, javax.swing.GroupLayout.PREFERRED_SIZE,
74, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()
```

```
                                .addComponent(jLabel6))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(jLabel4))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(jLabel3)))
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(showPasswordCheckBox)))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(52, 52, 52)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel6)
                .addGap(2, 2, 2)
                .addComponent(accountRoleComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel4)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(userNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(passwordField, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(showPasswordCheckBox)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(logInButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(signUpLink, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
        );

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
        gridBagConstraints.insets = new java.awt.Insets(6, 6, 0, 6);
        getContentPane().add(jPanel1, gridBagConstraints);

        pack();
    }// </editor-fold>

    private void showPasswordCheckBoxItemStateChanged(java.awt.event.ItemEvent evt) {
        int state = evt.getStateChange();

        if (state == ItemEvent.SELECTED){
            passwordField.setEchoChar((char) 0);
        }
        else {
            passwordField.setEchoChar('•');
        }
    }

    private void signUpLinkMouseClicked(java.awt.event.MouseEvent evt) {
        new SignUpFrame().setVisible(true);
        dispose();
    }
```

```
    private void accountRoleComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        final int SELECT = 0;
        final int SELECTED_ROLE = accountRoleComboBox.getSelectedIndex();

        if (SELECTED_ROLE != SELECT) {
            userNameTextField.setEnabled(true);
            passwordField.setEnabled(true);
        }
        else {
            userNameTextField.setEnabled(false);
            passwordField.setEnabled(false);
        }
    }

    private void logInButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String username = userNameTextField.getText();
        String password = String.valueOf(passwordField.getPassword());
        String role = accountRoleComboBox.getSelectedItem().toString();

        if (username.isBlank() ||
                password.isBlank() ||
                role.equals("Select")) {

            JOptionPane.showMessageDialog(this, "Fill out needed information", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        String selectQuery = "SELECT * FROM account WHERE username = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setString(1, username);
            ResultSet resultSet = preparedStatement.executeQuery();

            while (resultSet.next()) {
                int dbUserId = resultSet.getInt("user_id");
                String dbFirstName = resultSet.getString("first_name");
                String dbUsername = resultSet.getString("username");
                String dbPassword = resultSet.getString("password");
                String dbRole = resultSet.getString("role");

                if (dbUsername.equals(username) &&
                        dbPassword.equals(password) &&
                        dbRole.equals(role)) {

                    _userId = dbUserId;
                    _firstName = dbFirstName;

                    JOptionPane.showMessageDialog(this, "Log in succesful", "Notice",
JOptionPane.INFORMATION_MESSAGE);
                    identifyAccountRole(role);
                    return;
                }
            }

            JOptionPane.showMessageDialog(this, "This credential doesn't match our record", "Error
Occured", JOptionPane.ERROR_MESSAGE);
        } catch (SQLException ex) {
            System.err.println("Log in Operation: " + ex.getMessage());
        }
    }

    private void identifyAccountRole(String role) {
        switch (role) {
            case "Librarian": {
                LibrarianBookManagementFrame.librarianFirstName = _firstName;
                LibrarianBookManagementFrame.librarianUserId = _userId;

                LibrarianMemberManagementFrame.librarianFirstName = _firstName;
                LibrarianMemberManagementFrame.librarianUserId = _userId;

                LibrarianBorrowedBookManagementFrame.librarianFirstName = _firstName;
                LibrarianBorrowedBookManagementFrame.librarianUserId = _userId;

                new LibrarianBookManagementFrame().setVisible(true);
                dispose();
                break;
```

```
            }
            case "Student": {
                UserViewBooksFrame.firstName = _firstName;
                UserViewBooksFrame.userFrameUserId = _userId;

                UserViewBorrowedBooksFrame.firstName = _firstName;
                UserViewBorrowedBooksFrame.userFrameUserId = _userId;

                new UserViewBooksFrame().setVisible(true);
                dispose();
                break;
            }
            case "Staff": {
                UserViewBooksFrame.firstName = _firstName;
                UserViewBooksFrame.userFrameUserId = _userId;

                UserViewBorrowedBooksFrame.firstName = _firstName;
                UserViewBorrowedBooksFrame.userFrameUserId = _userId;

                new UserViewBooksFrame().setVisible(true);
                dispose();
                break;
            }
            case "Teacher": {
                UserViewBooksFrame.firstName = _firstName;
                UserViewBooksFrame.userFrameUserId = _userId;

                UserViewBorrowedBooksFrame.firstName = _firstName;
                UserViewBorrowedBooksFrame.userFrameUserId = _userId;

                new UserViewBooksFrame().setVisible(true);
                dispose();
                break;
            }
        }
    }


    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LogInFrame().setVisible(true);
            }
        });
    }


    // Variables declaration - do not modify
    private javax.swing.JComboBox<String> accountRoleComboBox;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton logInButton;
    private javax.swing.JPasswordField passwordField;
    private javax.swing.JCheckBox showPasswordCheckBox;
    private javax.swing.JLabel signUpLink;
    private javax.swing.JTextField userNameTextField;
    // End of variables declaration
}
```

LibrarianBookManagementFrame Class

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import javax.swing.*;
import java.sql.*;
import java.time.*;
import javax.swing.table.DefaultTableModel;
public class LibrarianBookManagementFrame extends javax.swing.JFrame {
    Connection connection = DatabaseConnection.getConnection();
```

```
    DefaultTableModel booksTableModel;
    public static int librarianUserId;
    public static String librarianFirstName;

    public LibrarianBookManagementFrame() {
        initComponents();
        librarianFirstNameLabel.setText(librarianFirstName);
        LibrarianMemberManagementFrame.librarianUserId = librarianUserId;
        populateBooksTable();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        manageBooksButton = new javax.swing.JButton();
        manageMembersButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        logOutButton = new javax.swing.JButton();
        librarianFirstNameLabel = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        manageBooksButton1 = new javax.swing.JButton();
        jSeparator2 = new javax.swing.JSeparator();
        jPanel2 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        booksTable = new javax.swing.JTable();
        titleTextField = new javax.swing.JTextField();
        authorTextField = new javax.swing.JTextField();
        isbnTextField = new javax.swing.JTextField();
        searchTextField = new javax.swing.JTextField();
        publishDateChooser = new com.toedter.calendar.JDateChooser();
        quantitySpinner = new javax.swing.JSpinner();
        clearButton = new javax.swing.JButton();
        addButton = new javax.swing.JButton();
        updateButton = new javax.swing.JButton();
        deleteButton = new javax.swing.JButton();
        categoryComboBox = new javax.swing.JComboBox<>();
        statusComboBox = new javax.swing.JComboBox<>();
        booksFilterComboBox = new javax.swing.JComboBox<>();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Manage Books");
        setLocation(new java.awt.Point(450, 150));
        setResizable(false);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(255, 255, 255), 4,
true));

        manageBooksButton.setBackground(new java.awt.Color(24, 119, 242));
        manageBooksButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        manageBooksButton.setForeground(new java.awt.Color(255, 255, 255));
        manageBooksButton.setText("Manage Books");

        manageMembersButton.setText("Manage Members");
        manageMembersButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageMembersButtonActionPerformed(evt);
            }
        });

        jLabel1.setForeground(new java.awt.Color(51, 51, 51));
        jLabel1.setText("Welcome back!");

        logOutButton.setBackground(new java.awt.Color(255, 0, 0));
        logOutButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logOutButton.setForeground(new java.awt.Color(0, 255, 255));
```

```
        logOutButton.setText("Log out");
        logOutButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logOutButtonActionPerformed(evt);
            }
        });

        librarianFirstNameLabel.setFont(new java.awt.Font("Georgia Pro Black", 0, 24)); // NOI18N
        librarianFirstNameLabel.setForeground(new java.awt.Color(51, 51, 51));
        librarianFirstNameLabel.setText("First Name");

        manageBooksButton1.setText("Manage Borrowed Books");
        manageBooksButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageBooksButton1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(librarianFirstNameLabel))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(24, 24, 24)
                        .addComponent(jLabel1)))
                .addGap(0, 0, Short.MAX_VALUE))
            .addComponent(manageBooksButton1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(manageMembersButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(manageBooksButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jSeparator1)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(logOutButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jSeparator2))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(librarianFirstNameLabel)
                .addGap(7, 7, 7)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageMembersButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(logOutButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        booksTable.setAutoCreateRowSorter(true);
```

```
        booksTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
```

```
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null}
            },
            new String [] {
                "ISBN", "Title", "Author", "Publish Date", "Category", "Status", "Quantity"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        booksTable.setFillsViewportHeight(true);
        booksTable.setShowGrid(true);
        booksTable.getTableHeader().setResizingAllowed(false);
        booksTable.getTableHeader().setReorderingAllowed(false);
        booksTable.setUpdateSelectionOnSort(false);
        booksTable.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                booksTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(booksTable);

        isbnTextField.setToolTipText("");

        searchTextField.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                searchTextFieldKeyReleased(evt);
            }
        });

        publishDateChooser.setDateFormatString("yyyy, MM, dd");

        clearButton.setText("CLEAR");
        clearButton.setToolTipText("Remove all inputs");
        clearButton.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        clearButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                clearButtonActionPerformed(evt);
            }
        });

        addButton.setText("ADD");
        addButton.setToolTipText("Add book on the list");
        addButton.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        addButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                addButtonActionPerformed(evt);
```

```
            }
        });

        updateButton.setText("UPDATE");
        updateButton.setToolTipText("Update book on the list");
        updateButton.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        updateButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                updateButtonActionPerformed(evt);
            }
        });

        deleteButton.setText("DELETE");
        deleteButton.setToolTipText("Delete book on the list");
        deleteButton.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        deleteButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                deleteButtonActionPerformed(evt);
            }
        });

        categoryComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Select...</font></html>", "Programming", "Science", "Math",
"Philosophy", "English" }));

        statusComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "<html><font
color='gray'>Select...</font></html>", "Available", "Not Available" }));

        booksFilterComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Category...</font></html>", "Programming", "Science", "Math",
"Philosophy", "English" }));
        booksFilterComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                booksFilterComboBoxActionPerformed(evt);
            }
        });

        jLabel3.setText("Author:");

        jLabel4.setText("ISBN:");

        jLabel5.setText("Quantity:");

        jLabel7.setText("Title:");

        jLabel8.setText("Category:");

        jLabel2.setText("Status:");

        jLabel6.setText("Publish Date:");

        jLabel9.setText("Search");

        jLabel10.setText("Filter Books by");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 850,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel2Layout.createSequentialGroup()
                                .addComponent(jLabel9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(searchTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 544, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(booksFilterComboBox, 0, 153, Short.MAX_VALUE))
                            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel6)
                                    .addComponent(jLabel7)
                                    .addComponent(jLabel3)
                                    .addComponent(jLabel4)
                                    .addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                                    .addComponent(isbnTextField,
javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(authorTextField,
javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(titleTextField,
javax.swing.GroupLayout.Alignment.LEADING)
                                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(jPanel2Layout.createSequentialGroup()
                                            .addComponent(quantitySpinner,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addGap(18, 18, 18)
                                            .addComponent(jLabel8))
                                        .addComponent(publishDateChooser,
javax.swing.GroupLayout.PREFERRED_SIZE, 130, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addComponent(categoryComboBox, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                        .addGroup(jPanel2Layout.createSequentialGroup()
                                            .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addComponent(statusComboBox,
javax.swing.GroupLayout.PREFERRED_SIZE, 154, javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addGap(0, 0, Short.MAX_VALUE)))))
                                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(addButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                    .addComponent(updateButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 85, Short.MAX_VALUE)
                                    .addComponent(deleteButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                    .addComponent(clearButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel9)
                    .addComponent(jLabel10)
                    .addComponent(booksFilterComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 394,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(titleTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel7)
                    .addComponent(clearButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(4, 4, 4)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(authorTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3)
                    .addComponent(addButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(isbnTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel4)
                    .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                .addComponent(statusComboBox,
javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jLabel2)
                                .addComponent(deleteButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addGroup(jPanel2Layout.createSequentialGroup()
                            .addGap(7, 7, 7)
                            .addComponent(publishDateChooser,
javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jLabel6)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel8)
                        .addComponent(categoryComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(quantitySpinner, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel5)))
                .addContainerGap(37, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String isbn = isbnTextField.getText().trim();
        String title = titleTextField.getText();
        String author = authorTextField.getText();
        String date = "";
        try {
            date =
publishDateChooser.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate().toString();
        } catch (NullPointerException ex) {
            JOptionPane.showMessageDialog(this, "Fill out needed information", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }
        String category = categoryComboBox.getSelectedItem().toString();
        String status = statusComboBox.getSelectedItem().toString();
        String quantity = quantitySpinner.getValue().toString();

        if (isbn.isBlank() ||
                title.isBlank() ||
                author.isBlank() ||
                date.isBlank() ||
                status.isBlank() ||
                quantity.isBlank()) {

            JOptionPane.showMessageDialog(this, "Fill out needed information.", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        String insertQuery = "INSERT INTO books " +
                "VALUES(?, ?, ?, ?, ?, ?, ?)";

        try (PreparedStatement preparedStatement = connection.prepareStatement(insertQuery)){
            preparedStatement.setLong(1, Long.parseLong(isbn));
            preparedStatement.setString(2, title);
            preparedStatement.setString(3, author);
            preparedStatement.setString(4, date);
            preparedStatement.setString(5, category);
            preparedStatement.setString(6, status);
            preparedStatement.setInt(7, Integer.parseInt(quantity));
            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Book Added Successfully", "Notice",
JOptionPane.INFORMATION_MESSAGE);

            clearUserInput();
        } catch (SQLException ex) {
            System.err.println("Add Operation: " + ex.getMessage());
            JOptionPane.showMessageDialog(this, "This record already exist", "Notice",
JOptionPane.WARNING_MESSAGE);
        } catch (NumberFormatException ex) {
            System.err.println("Add Operation: " + ex.getMessage());
            JOptionPane.showMessageDialog(this, "ISBN Must Contain Only Numbers", "Error",
JOptionPane.ERROR_MESSAGE);
        }

        populateBooksTable();
    }

    private void checkUserInput() {
        String isbn = isbnTextField.getText().trim();
        String title = titleTextField.getText();
        String author = authorTextField.getText();
        String date = "";
        try {
```

```
            date =
publishDateChooser.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate().toString();
        } catch (NullPointerException ex) {
            JOptionPane.showMessageDialog(this, "Fill out needed information", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }
        String category = categoryComboBox.getSelectedItem().toString();
        String status = statusComboBox.getSelectedItem().toString();
        String quantity = quantitySpinner.getValue().toString();

        if (isbn.isBlank() ||
                title.isBlank() ||
                author.isBlank() ||
                date.isBlank() ||
                category.equals("Select...") ||
                status.equals("Select...") ||
                status.isBlank() ||
                quantity.isBlank()) {
            JOptionPane.showMessageDialog(this, "Fill out needed information.", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }
    }

    private void clearButtonActionPerformed(java.awt.event.ActionEvent evt) {
        clearUserInput();
    }

    private void clearUserInput() {
        authorTextField.setText("");
        titleTextField.setText("");
        isbnTextField.setText("");
        publishDateChooser.setDate(null);
        statusComboBox.setSelectedIndex(0);
        quantitySpinner.setValue(0);
        categoryComboBox.setSelectedIndex(0);

        booksTable.clearSelection();
    }

    private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int selectedRow = booksTable.getSelectedRow();

        if (selectedRow != -1) {
            final int ISBN_COLUMN = 0;

            long deletedBook = (long) booksTable.getValueAt(selectedRow, ISBN_COLUMN);
            String deleteQuery = "DELETE FROM books WHERE isbn = ?";

            try (PreparedStatement preparedStatement = connection.prepareStatement(deleteQuery)) {
                preparedStatement.setLong(1, deletedBook);
                preparedStatement.executeUpdate();

                JOptionPane.showMessageDialog(this, "Book deleted Succesfully.", "Notice",
JOptionPane.INFORMATION_MESSAGE);

                booksTableModel.removeRow(selectedRow);
                booksTableModel.fireTableDataChanged();
            } catch (SQLException ex) {
                System.out.println(ex.getMessage());
            }
        }
        else {
            JOptionPane.showMessageDialog(this, "Select a row in the Table you want to delete.",
"Reminder", JOptionPane.WARNING_MESSAGE);
        }
    }

    private void updateButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String isbn = isbnTextField.getText();
        String title = titleTextField.getText();
        String author = authorTextField.getText();
        String date = "";
        try {
            date =
publishDateChooser.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate().toString();
```

```
        } catch (NullPointerException ex) {
            JOptionPane.showMessageDialog(this, "Fill out needed information", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }
        String category = categoryComboBox.getSelectedItem().toString();
        String status = statusComboBox.getSelectedItem().toString();
        String quantity = quantitySpinner.getValue().toString();

        String updateQuery = "UPDATE books " +
                "SET isbn = ?, title = ?, author = ?, publish_date = ?, category = ?, status = ?,
quantity = ? " +
                "WHERE isbn = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {
            preparedStatement.setLong(1, Long.parseLong(isbn));
            preparedStatement.setString(2, title);
            preparedStatement.setString(3, author);
            preparedStatement.setString(4, date);
            preparedStatement.setString(5, category);
            preparedStatement.setString(6, status);
            preparedStatement.setInt(7, Integer.parseInt(quantity));
            preparedStatement.setLong(8, Long.parseLong(isbn));
            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Record Succesfully Updated", "Notice",
JOptionPane.INFORMATION_MESSAGE);

            clearUserInput();
        } catch (SQLException ex) {
            System.err.println("Update Operation: " + ex.getMessage());
        }

        populateBooksTable();
    }

    private void booksTableMouseClicked(java.awt.event.MouseEvent evt) {
        int selectedRow = 0;

        try {
            selectedRow = booksTable.getSelectedRow();
        } catch (IndexOutOfBoundsException ex) {
            System.err.println("SelectedRow: " + ex.getMessage());
        }

        String isbn = String.valueOf(booksTable.getValueAt(selectedRow, 0));
        String title = String.valueOf(booksTable.getValueAt(selectedRow, 1));
        String author = String.valueOf(booksTable.getValueAt(selectedRow, 2));
        String publishDate = String.valueOf(booksTable.getValueAt(selectedRow, 3));
        String category = String.valueOf(booksTable.getValueAt(selectedRow, 4));
        String status = String.valueOf(booksTable.getValueAt(selectedRow, 5));
        String quantity = String.valueOf(booksTable.getValueAt(selectedRow, 6));

        final int Clicked_2_TIMES = 2;
        if (evt.getClickCount() == Clicked_2_TIMES) {
            String bookDetails = "<html><b>Title:</b> " + title + "<br>" +
                    "<b>Author:</b> " + author + "<br>" +
                    "<b>ISBN:</b> " + isbn + "<br>" +
                    "<b>Publish Date:</b> " + publishDate + "<br>" +
                    "<b>Category:</b> " + category + "<br>" +
                    "<b>Status:</b> " + status + "<br>" +
                    "<b>Quantity:</b> " + quantity + "</html>";

            JOptionPane.showMessageDialog(this, bookDetails, "Book Details",
JOptionPane.PLAIN_MESSAGE);
        }
        else {
            titleTextField.setText(title);
            authorTextField.setText(author);
            isbnTextField.setText(isbn);
            publishDateChooser.setDate(Date.valueOf(LocalDate.parse((CharSequence)publishDate)));
            categoryComboBox.setSelectedItem(category);
            statusComboBox.setSelectedItem(status);
            quantitySpinner.setValue(Integer.valueOf(quantity));
        }
    }
```

```
    private void manageMembersButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new LibrarianMemberManagementFrame().setVisible(true);
        dispose();
    }

    private void logOutButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this, "Are you sure you want to log out?",
"Log out Confirmation", JOptionPane.YES_NO_OPTION);
        final int YES = 0;

        if (response == YES) {
            new LogInFrame().setVisible(true);
            dispose();
        }
    }

    private void searchTextFieldKeyReleased(java.awt.event.KeyEvent evt) {
        booksTableModel.setRowCount(0);
        String userInput = searchTextField.getText();

        String searchQuery = "SELECT * FROM books " +
                "WHERE title LIKE ? OR author LIKE ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(searchQuery)) {
            preparedStatement.setString(1, "%" + userInput + "%");
            preparedStatement.setString(2, "%" + userInput + "%");
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.err.println("Search Operation: " + ex.getMessage());
        }
    }

    private void booksFilterComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        String selectedCategory = booksFilterComboBox.getSelectedItem().toString();
        booksTableModel.setRowCount(0);

        String selectQuery = "SELECT * FROM books " +
                "WHERE category = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setString(1, selectedCategory);
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Books Filtering Operation: " + ex.getMessage());
        }

        final int SELECT = 0;
        if (booksFilterComboBox.getSelectedIndex() == SELECT) {
           populateBooksTable();
        }
    }

    private void manageBooksButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        new LibrarianBorrowedBookManagementFrame().setVisible(true);
        dispose();
    }

    private void populateBooksTable() {
        booksTableModel = (DefaultTableModel) booksTable.getModel();
        booksTableModel.setRowCount(0);

        String selectQuery = "SELECT * FROM books";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject( i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Pupulate Books Table Operation: " + ex.getMessage());
        }
    }

    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LibrarianBookManagementFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton addButton;
    private javax.swing.JTextField authorTextField;
    private javax.swing.JComboBox<String> booksFilterComboBox;
    private javax.swing.JTable booksTable;
    private javax.swing.JComboBox<String> categoryComboBox;
    private javax.swing.JButton clearButton;
    private javax.swing.JButton deleteButton;
    private javax.swing.JTextField isbnTextField;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    public javax.swing.JLabel librarianFirstNameLabel;
    private javax.swing.JButton logOutButton;
    private javax.swing.JButton manageBooksButton;
    private javax.swing.JButton manageBooksButton1;
    private javax.swing.JButton manageMembersButton;
    private com.toedter.calendar.JDateChooser publishDateChooser;
    private javax.swing.JSpinner quantitySpinner;
    private javax.swing.JTextField searchTextField;
    private javax.swing.JComboBox<String> statusComboBox;
    private javax.swing.JTextField titleTextField;
    private javax.swing.JButton updateButton;
```

```
    // End of variables declaration
}
```

LibrarianMemberManagementFrame

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import javax.swing.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class LibrarianMemberManagementFrame extends javax.swing.JFrame {
    private Connection connection = DatabaseConnection.getConnection();
    private DefaultTableModel membersTableModel;
    public static int librarianUserId;
    public static String librarianFirstName;

    public LibrarianMemberManagementFrame() {
        initComponents();
        LibrarianBookManagementFrame.librarianUserId = librarianUserId;
        librarianFirstNameLabel.setText(librarianFirstName);
        userIDTextField.setVisible(false);
        populateMembersTable();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        manageBooksButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        manageMembersButton = new javax.swing.JButton();
        logOutButton = new javax.swing.JButton();
        librarianFirstNameLabel = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        manageBooksButton1 = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        membersTable = new javax.swing.JTable();
        firstNameTextField = new javax.swing.JTextField();
        emailTextField = new javax.swing.JTextField();
        userNameTextField = new javax.swing.JTextField();
        roleComboBox = new javax.swing.JComboBox<>();
        clearButton = new javax.swing.JButton();
        addButton = new javax.swing.JButton();
        updateButton = new javax.swing.JButton();
        deleteButton = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        searchTextField = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();
        accountRoleComboBox = new javax.swing.JComboBox<>();
        passwordTextField = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        lastNameTextField = new javax.swing.JTextField();
        userIDTextField = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Manage Members");
        setLocation(new java.awt.Point(450, 150));
        setResizable(false);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(255, 255, 255), 1,
true));

        manageBooksButton.setText("Manage Books");
        manageBooksButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageBooksButtonActionPerformed(evt);
            }
        });
```

```
        jLabel1.setText("Welcome Back!");

        manageMembersButton.setBackground(new java.awt.Color(24, 119, 242));
        manageMembersButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        manageMembersButton.setForeground(new java.awt.Color(255, 255, 255));
        manageMembersButton.setText("Manage Members");

        logOutButton.setBackground(new java.awt.Color(255, 0, 0));
        logOutButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logOutButton.setForeground(new java.awt.Color(0, 255, 255));
        logOutButton.setText("Log out");
        logOutButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logOutButtonActionPerformed(evt);
            }
        });

        librarianFirstNameLabel.setFont(new java.awt.Font("Georgia Pro Black", 0, 24)); // NOI18N
        librarianFirstNameLabel.setText("First Name");

        manageBooksButton1.setText("Manage Borrowed Books");
        manageBooksButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageBooksButton1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(29, 29, 29)
                .addComponent(jLabel1)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jSeparator2)
                    .addComponent(manageBooksButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jSeparator1)
                    .addComponent(logOutButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(librarianFirstNameLabel)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addComponent(manageBooksButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(manageMembersButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addGap(20, 20, 20)
                .addComponent(librarianFirstNameLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageMembersButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
                .addComponent(logOutButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        membersTable.setAutoCreateRowSorter(true);
        membersTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null}
            },
            new String [] {
                "User_ID", "First Name", "Last Name", "Email", "Username", "Password", "Role"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        membersTable.setFillsViewportHeight(true);
        membersTable.setShowGrid(true);
        membersTable.getTableHeader().setResizingAllowed(false);
        membersTable.getTableHeader().setReorderingAllowed(false);
        membersTable.setUpdateSelectionOnSort(false);
        membersTable.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                membersTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(membersTable);

        roleComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "<html><font
color='gray'>Select...</font></html>", "Librarian", "Teacher", "Student", "Staff" }));

        clearButton.setText("CLEAR");

clearButton.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.R
AISED));
        clearButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                clearButtonActionPerformed(evt);
            }
        });

        addButton.setText("ADD");

addButton.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAI
SED));
        addButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                addButtonActionPerformed(evt);
            }
        });

        updateButton.setText("UPDATE");

updateButton.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.
RAISED));
        updateButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                updateButtonActionPerformed(evt);
            }
        });

        deleteButton.setText("DELETE");

deleteButton.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.
RAISED));
        deleteButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                deleteButtonActionPerformed(evt);
            }
        });

        jLabel3.setText("Email:");

        jLabel4.setText("Password:");

        jLabel7.setText("First Name:");

        jLabel8.setText("Role:");

        jLabel9.setText("Search:");

        searchTextField.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                searchTextFieldKeyReleased(evt);
            }
        });

        jLabel10.setText("Filter Members by:");

        accountRoleComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Role...</font></html>", "Librarian", "Teacher", "Student", "Staff", " ",
" " }));
        accountRoleComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                accountRoleComboBoxActionPerformed(evt);
            }
        });

        jLabel6.setText("Username:");
```

```
        jLabel11.setText("Last Name:");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel2Layout.createSequentialGroup()
                                .addComponent(jLabel9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(searchTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 544, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(accountRoleComboBox, 0, 131, Short.MAX_VALUE))
                            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel7)
                                    .addComponent(jLabel3)
                                    .addComponent(jLabel6)
                                    .addComponent(jLabel4)
                                    .addComponent(jLabel8))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(passwordTextField)
                                    .addComponent(userNameTextField)
                                    .addComponent(emailTextField)
                                    .addGroup(jPanel2Layout.createSequentialGroup()
                                        .addComponent(firstNameTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 158, javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addGap(18, 18, 18)
                                        .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                        .addComponent(lastNameTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 164, javax.swing.GroupLayout.PREFERRED_SIZE))
                                    .addComponent(roleComboBox, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(addButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(clearButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(updateButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(deleteButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(userIDTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 85, javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGap(0, 0, Short.MAX_VALUE))))
                    .addComponent(jScrollPane1))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel9))
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel10)
                        .addComponent(accountRoleComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 394,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(firstNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel7)
                    .addComponent(clearButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel11)
                    .addComponent(lastNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(4, 4, 4)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(emailTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3)
                    .addComponent(addButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(userNameTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6)
                    .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(passwordTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel4)
                    .addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(roleComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel8)
                    .addComponent(userIDTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(26, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String firstName = firstNameTextField.getText();
        String lastName = lastNameTextField.getText();
        String email = emailTextField.getText();
        String username = userNameTextField.getText();
        String password = passwordTextField.getText();
        String role = roleComboBox.getSelectedItem().toString();
        int user_id = 0;
        try {
            int selectedRow = membersTable.getSelectedRow();
            final int USER_ID_COLUMN = 0;
            user_id = Integer.parseInt(membersTable.getValueAt(selectedRow,
USER_ID_COLUMN).toString());
        } catch (IndexOutOfBoundsException ex) {
            System.out.println("Row Selection: " + ex.getMessage());
        }

        if (firstName.isBlank() ||
                lastName.isBlank() ||
                email.isBlank() ||
                username.isBlank() ||
                password.isBlank() ||
                role.isBlank()) {
            JOptionPane.showMessageDialog(this, "Fill out needed information.", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        String insertQuery = "INSERT INTO account (first_name, last_name, email, username,
password, role, user_id) " +
                "VALUES (?, ?, ?, ?, ?, ?, ?)";

        try (PreparedStatement preparedStatement = connection.prepareStatement(insertQuery)) {
            preparedStatement.setString(1, firstName);
            preparedStatement.setString(2, lastName);
            preparedStatement.setString(3, email);
            preparedStatement.setString(4, username);
            preparedStatement.setString(5, password);
            preparedStatement.setString(6, role);
            preparedStatement.setInt(7, user_id);
            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Member Added Successfully.", "Notice",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException ex) {
            System.err.println("Add Operation: " + ex.getMessage());
            JOptionPane.showMessageDialog(this, "This record already exist", "Notice",
JOptionPane.WARNING_MESSAGE);
        }

        populateMembersTable();
    }

    private void populateMembersTable() {
        membersTableModel = (DefaultTableModel) membersTable.getModel();
        membersTableModel.setRowCount(0);

        String selectQuery = "SELECT * FROM account";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            ResultSet resultSet = preparedStatement.executeQuery();
            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }
```

```
                membersTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.err.println("Populate Members Table Operation: " + ex.getMessage());
        }

    }

    private void updateButtonActionPerformed(java.awt.event.ActionEvent evt) {
        String firstName = firstNameTextField.getText();
        String lastName = lastNameTextField.getText();
        String email = emailTextField.getText();
        String username = userNameTextField.getText();
        String password = passwordTextField.getText();
        String role = roleComboBox.getSelectedItem().toString();
        int user_id = 0;
        try {
            int selectedRow = membersTable.getSelectedRow();
            final int USER_ID_COLUMN = 0;
            user_id = Integer.parseInt(membersTable.getValueAt(selectedRow,
USER_ID_COLUMN).toString());
        } catch (IndexOutOfBoundsException ex) {
            System.out.println("Row Selection: " + ex.getMessage());
        }

        if (firstName.isBlank() ||
                lastName.isBlank() ||
                email.isBlank() ||
                username.isBlank() ||
                password.isBlank() ||
                role.isBlank()) {

            JOptionPane.showMessageDialog(this, "Fill out needed information.", "Reminder",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        String updateQuery = "UPDATE account " +
                "SET first_name = ?, last_name = ?, email = ?, username = ?, password = ?, role =
? " +
                "WHERE user_id = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {
            preparedStatement.setString(1, firstName);
            preparedStatement.setString(2, lastName);
            preparedStatement.setString(3, email);
            preparedStatement.setString(4, username);
            preparedStatement.setString(5, password);
            preparedStatement.setString(6, role);
            preparedStatement.setInt(7, user_id);
            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Record Succesfully Updated", "Notice",
JOptionPane.INFORMATION_MESSAGE);

            clearUserInputs();
            membersTable.clearSelection();
        } catch (SQLException ex) {
            System.err.println("Update Operation: " + ex.getMessage());
        }

        populateMembersTable();
    }

    private void membersTableMouseClicked(java.awt.event.MouseEvent evt) {
        final int CLICKED_2_TIMES = 2;
        if (evt.getClickCount() == CLICKED_2_TIMES) {
            try {
                int selectedRow = membersTable.getSelectedRow();
                String userID = String.valueOf(membersTable.getValueAt(selectedRow, 0));
                String firstName = String.valueOf(membersTable.getValueAt(selectedRow, 1));
                String lastName = String.valueOf(membersTable.getValueAt(selectedRow, 2));
                String email = String.valueOf(membersTable.getValueAt(selectedRow, 3));
                String username = String.valueOf(membersTable.getValueAt(selectedRow, 4));
                String password = String.valueOf(membersTable.getValueAt(selectedRow, 5));
                String role = String.valueOf(membersTable.getValueAt(selectedRow, 6));
```

```
                userIDTextField.setText(userID);
                firstNameTextField.setText(firstName);
                lastNameTextField.setText(lastName);
                emailTextField.setText(email);
                userNameTextField.setText(username);
                passwordTextField.setText(password);
                roleComboBox.setSelectedItem(role);
            } catch (IndexOutOfBoundsException ex) {
                System.err.println("membersTableMouseClicked Operation: " + ex.getMessage());
            }
        }
    }

    private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int selectedRow = membersTable.getSelectedRow();

        if (selectedRow != -1) {
            final int USER_ID_COLUMN = 0;
            int deletedMember = (int) membersTable.getValueAt(selectedRow, USER_ID_COLUMN);

            String deleteQuery = "DELETE FROM account WHERE user_id = ?";

            try (PreparedStatement preparedStatement = connection.prepareStatement(deleteQuery)) {
                preparedStatement.setInt(1, deletedMember);
                preparedStatement.executeUpdate();

                JOptionPane.showMessageDialog(this, "Book deleted Succesfully.", "Notice",
JOptionPane.INFORMATION_MESSAGE);

                membersTableModel.removeRow(selectedRow);
                membersTableModel.fireTableDataChanged();
            } catch (SQLException ex) {
                System.err.println("Delete Operation: " + ex.getMessage());
            }
        }
        else {
            JOptionPane.showMessageDialog(this, "Select a row in the Table you want to delete.",
"Reminder", JOptionPane.WARNING_MESSAGE);
        }
    }

    private void clearButtonActionPerformed(java.awt.event.ActionEvent evt) {
        clearUserInputs();
    }

    private void manageBooksButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new LibrarianBookManagementFrame().setVisible(true);
        dispose();
    }

    private void logOutButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this, "Are you sure you want to log out?",
"Log out Confirmation", JOptionPane.YES_NO_OPTION);

        final int YES = 0;
        if (response == YES) {
            new LogInFrame().setVisible(true);
            dispose();
        }
    }

    private void searchTextFieldKeyReleased(java.awt.event.KeyEvent evt) {
        membersTableModel.setRowCount(0);

        String userInput = searchTextField.getText();
        String searchQuery = "SELECT * FROM account " +
                "WHERE first_name LIKE ? OR last_name LIKE ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(searchQuery)) {
            preparedStatement.setString(1, "%" + userInput + "%");
            preparedStatement.setString(2, "%" + userInput + "%");
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
```

```
            Object[] row = new Object[columnCount];

            for (int i = 1; i <= columnCount; i++) {
                row[i-1] = resultSet.getObject(i);
            }

            membersTableModel.addRow(row);
        }
    } catch (SQLException ex) {
        System.out.println("Search Operation: " + ex.getMessage());
    }
}

private void accountRoleComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    membersTableModel.setRowCount(0);

    String selectedRole = accountRoleComboBox.getSelectedItem().toString();

    String selectQuery = "SELECT * FROM account " +
            "WHERE role = ?";

    try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
        preparedStatement.setString(1, selectedRole);
        ResultSet resultSet = preparedStatement.executeQuery();

        int columnCount = resultSet.getMetaData().getColumnCount();

        while (resultSet.next()) {
            Object[] row = new Object[columnCount];

            for (int i = 1; i <= columnCount; i++) {
                row[i-1] = resultSet.getObject(i);
            }

            membersTableModel.addRow(row);
        }
    } catch (SQLException ex) {
        System.out.println("Selecting Account Role Operation: " + ex.getMessage());
    }

    final int SELECT = 0;
    if (accountRoleComboBox.getSelectedIndex() == SELECT) {
        populateMembersTable();
    }
}

private void manageBooksButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    new LibrarianBorrowedBookManagementFrame().setVisible(true);
    dispose();
}

private void clearUserInputs() {
    firstNameTextField.setText("");
    lastNameTextField.setText("");
    emailTextField.setText("");
    userNameTextField.setText("");
    passwordTextField.setText("");
    roleComboBox.setSelectedIndex(0);
    userIDTextField.setText("");

    membersTable.clearSelection();
}

public static void main(String args[]) {
    try {
        UIManager.setLookAndFeel(new FlatMacLightLaf());
    } catch (UnsupportedLookAndFeelException ex) {
         System.out.println(ex.getMessage());
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new LibrarianMemberManagementFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
```

```
    private javax.swing.JComboBox<String> accountRoleComboBox;
    private javax.swing.JButton addButton;
    private javax.swing.JButton clearButton;
    private javax.swing.JButton deleteButton;
    private javax.swing.JTextField emailTextField;
    private javax.swing.JTextField firstNameTextField;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JTextField lastNameTextField;
    public javax.swing.JLabel librarianFirstNameLabel;
    private javax.swing.JButton logOutButton;
    private javax.swing.JButton manageBooksButton;
    private javax.swing.JButton manageBooksButton1;
    private javax.swing.JButton manageMembersButton;
    private javax.swing.JTable membersTable;
    private javax.swing.JTextField passwordTextField;
    private javax.swing.JComboBox<String> roleComboBox;
    private javax.swing.JTextField searchTextField;
    private javax.swing.JButton updateButton;
    private javax.swing.JTextField userIDTextField;
    private javax.swing.JTextField userNameTextField;
    // End of variables declaration
}
```

LibrarianBorrowedBookManagementFrame

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import javax.swing.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
public class LibrarianBorrowedBookManagementFrame extends javax.swing.JFrame {
    Connection connection = DatabaseConnection.getConnection();
    DefaultTableModel booksTableModel;
    public static int librarianUserId;
    public static String librarianFirstName;

    public LibrarianBorrowedBookManagementFrame() {
        initComponents();
        librarianFirstNameLabel.setText(librarianFirstName);
        LibrarianMemberManagementFrame.librarianUserId = librarianUserId;
        populateBooksTable();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        manageBooksButton = new javax.swing.JButton();
        manageMembersButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        logOutButton = new javax.swing.JButton();
        librarianFirstNameLabel = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        manageBooksButton1 = new javax.swing.JButton();
        jSeparator2 = new javax.swing.JSeparator();
        jPanel2 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        booksTable = new javax.swing.JTable();
        searchTextField = new javax.swing.JTextField();
        booksFilterComboBox = new javax.swing.JComboBox<>();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Manage Borrowed Books");
```

```
        setLocation(new java.awt.Point(450, 150));
        setResizable(false);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(255, 255, 255), 4,
true));

        manageBooksButton.setText("Manage Books");
        manageBooksButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageBooksButtonActionPerformed(evt);
            }
        });

        manageMembersButton.setText("Manage Members");
        manageMembersButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                manageMembersButtonActionPerformed(evt);
            }
        });

        jLabel1.setForeground(new java.awt.Color(51, 51, 51));
        jLabel1.setText("Welcome back!");

        logOutButton.setBackground(new java.awt.Color(255, 0, 0));
        logOutButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logOutButton.setForeground(new java.awt.Color(0, 255, 255));
        logOutButton.setText("Log out");
        logOutButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logOutButtonActionPerformed(evt);
            }
        });

        librarianFirstNameLabel.setFont(new java.awt.Font("Georgia Pro Black", 0, 24)); // NOI18N
        librarianFirstNameLabel.setForeground(new java.awt.Color(51, 51, 51));
        librarianFirstNameLabel.setText("First Name");

        manageBooksButton1.setBackground(new java.awt.Color(24, 119, 242));
        manageBooksButton1.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        manageBooksButton1.setForeground(new java.awt.Color(255, 255, 255));
        manageBooksButton1.setText("Manage Borrowed Books");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(manageBooksButton1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(manageMembersButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(manageBooksButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jSeparator1)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(logOutButton,
javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jSeparator2))
                        .addContainerGap())
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(45, 45, 45)
                        .addComponent(jLabel1)
                        .addGap(0, 0, Short.MAX_VALUE))))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(librarianFirstNameLabel)
                .addGap(0, 0, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                .addComponent(jLabel1)
                .addGap(19, 19, 19)
                .addComponent(librarianFirstNameLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageMembersButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(manageBooksButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 395,
Short.MAX_VALUE)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(logOutButton, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        booksTable.setAutoCreateRowSorter(true);
        booksTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
```

```
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null}
            },
            new String [] {
                "Borrow_ID", "Title", "Category", "Borrower First Name", "Borrower Last Name",
"Borrowed Date", "Return Date"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        booksTable.setFillsViewportHeight(true);
        booksTable.setShowGrid(true);
        booksTable.getTableHeader().setResizingAllowed(false);
        booksTable.getTableHeader().setReorderingAllowed(false);
        booksTable.setUpdateSelectionOnSort(false);
        booksTable.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                booksTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(booksTable);
```

```
        searchTextField.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                searchTextFieldKeyReleased(evt);
            }
        });

        booksFilterComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Category...</font></html>", "Programming", "Science", "Math",
"Philosophy", "English" }));
        booksFilterComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                booksFilterComboBoxActionPerformed(evt);
            }
        });

        jLabel9.setText("Search");

        jLabel10.setText("Filter Books by");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel9)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
544, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel10)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(booksFilterComboBox, 0, 153, Short.MAX_VALUE))
                    .addComponent(jScrollPane1))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel9)
                    .addComponent(jLabel10)
                    .addComponent(booksFilterComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 628,
Short.MAX_VALUE)
                .addContainerGap())
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void manageMembersButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new LibrarianMemberManagementFrame().setVisible(true);
        dispose();
    }

    private void logOutButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this,
                "Are you sure you want to log out?",
                "Log out Confirmation",
                JOptionPane.YES_NO_OPTION);

        final int YES = 0;
        if (response == YES) {
            new LogInFrame().setVisible(true);
            dispose();
        }
    }

    private void booksFilterComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        booksTableModel.setRowCount(0);

        String selectedCategory = booksFilterComboBox.getSelectedItem().toString();

        String selectQuery = "SELECT borrowed_books.borrow_id, "
                + "books.title, books.category, "
                + "account.first_name, account.last_name, "
                + "borrowed_books.borrowed_date, "
                + "borrowed_books.returned_date " +
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id " +
                "WHERE category = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setString(1, selectedCategory);
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Books Filtering Operation: " + ex.getMessage());
        }

        final int SELECT = 0;
        if (booksFilterComboBox.getSelectedIndex() == SELECT) {
            populateBooksTable();
        }
    }

    private void searchTextFieldKeyReleased(java.awt.event.KeyEvent evt) {
        booksTableModel.setRowCount(0);

        String userInputTiTle = searchTextField.getText();
        String userInputFirstName = searchTextField.getText();
        String userInputLastName = searchTextField.getText();

        String searchQuery = "SELECT borrowed_books.borrow_id, "
                + "books.title, "
                + "books.category, account.first_name, "
                + "account.last_name, "
                + "borrowed_books.borrowed_date, "
                + "borrowed_books.returned_date " +
```

```
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id " +
                "WHERE books.title LIKE ? OR account.first_name LIKE ? OR account.last_name LIKE
?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(searchQuery)) {
            preparedStatement.setString(1, "%" + userInputTiTle + "%");
            preparedStatement.setString(2, "%" + userInputFirstName);
            preparedStatement.setString(3, userInputLastName + "%");
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Search Operation: " + ex.getMessage());
        }
    }

    private void booksTableMouseClicked(java.awt.event.MouseEvent evt) {
        int selectedRow = 0;

        try {
            selectedRow = booksTable.getSelectedRow();
        } catch (IndexOutOfBoundsException ex) {
            System.out.println("SelectedRow: " + ex.getMessage());
        }

        int borrow_id = Integer.parseInt(booksTable.getValueAt(selectedRow, 0).toString());
        String title = String.valueOf(booksTable.getValueAt(selectedRow, 1));
        String category = String.valueOf(booksTable.getValueAt(selectedRow, 2));
        String borrowerFirstName = String.valueOf(booksTable.getValueAt(selectedRow, 3));
        String borrowerLastName = String.valueOf(booksTable.getValueAt(selectedRow, 4));
        String borrowedDate = String.valueOf(booksTable.getValueAt(selectedRow, 5));
        String returnDate = String.valueOf(booksTable.getValueAt(selectedRow, 6));

        final int CLICKED_2_TIMES = 2;
        if (evt.getClickCount() == CLICKED_2_TIMES) {

            String borrowingDetails = "<html><b>Borrow_ID:</b> " + borrow_id + "<br>" +
                    "<b>Title:</b> " + title + "<br>" +
                    "<b>Category:</b> " + category + "<br>" +
                    "<b>Borrower Name:</b> " + borrowerFirstName + " " + borrowerLastName +"<br>"
+
                    "<b>Borrowed Date:</b> " + borrowedDate + "<br>" +
                    "<b>eturn Date:</b> " + returnDate + "</html>";

            String[] options = {"Returned", "Close"};

            int response = JOptionPane.showOptionDialog(this, borrowingDetails, "Borrowing
Details", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE, null, options, 1);

            final int RETURNED = 0;
            if (response == RETURNED) {
                int returnConfirmation = JOptionPane.showConfirmDialog(this, "Has this book
already been returned?", "Returning Book Confirmation", JOptionPane.YES_NO_OPTION);

                final int YES = 0;
                if (returnConfirmation == YES) {
                    deleteReturnedBooks(borrow_id);
                    incrementBookQuantity(title);

                    populateBooksTable();
                }
            }
        }
    }
```

```
    private void deleteReturnedBooks(int borrowId) {
        String deleteQuery = "DELETE FROM borrowed_books " +
                "WHERE borrow_id = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(deleteQuery)) {
            preparedStatement.setInt(1, borrowId);
            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(this, "Returned Successful!", "Notice",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException ex) {
            System.out.println("Delete Return Books Method: " + ex.getMessage());
        }
    }

    private void incrementBookQuantity(String title) {
        String updateQuery = "UPDATE books "
                + "SET quantity = quantity + 1 "
                + "WHERE title = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {
            preparedStatement.setString(1, title);
            preparedStatement.executeUpdate();
        } catch (SQLException ex) {
            System.out.println("Increment Book Quantity Method: " + ex.getMessage());
        }
    }

    private void manageBooksButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new LibrarianBookManagementFrame().setVisible(true);
        dispose();
    }

    private void populateBooksTable() {
        booksTableModel = (DefaultTableModel) booksTable.getModel();
        booksTableModel.setRowCount(0);

        String selectQuery = "SELECT borrowed_books.borrow_id, "
                + "books.title, books.category, "
                + "account.first_name, "
                + "account.last_name, "
                + "borrowed_books.borrowed_date, "
                + "borrowed_books.returned_date " +
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject( i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Populate Table Operation: " + ex.getMessage());
        }
    }

    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LibrarianBorrowedBookManagementFrame().setVisible(true);
            }
        });
```

```
    }

    // Variables declaration - do not modify
    private javax.swing.JComboBox<String> booksFilterComboBox;
    private javax.swing.JTable booksTable;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    public javax.swing.JLabel librarianFirstNameLabel;
    private javax.swing.JButton logOutButton;
    private javax.swing.JButton manageBooksButton;
    private javax.swing.JButton manageBooksButton1;
    private javax.swing.JButton manageMembersButton;
    private javax.swing.JTextField searchTextField;
    // End of variables declaration
}
```

UserViewBooksFrame

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import javax.swing.table.DefaultTableModel;
import javax.swing.*;
import java.sql.*;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
public class UserViewBooksFrame extends javax.swing.JFrame {
    private Connection connection = DatabaseConnection.getConnection();
    private DefaultTableModel booksTableModel;
    public static int userFrameUserId;
    public static String firstName;

    public UserViewBooksFrame() {
        initComponents();
        UserViewBorrowedBooksFrame.userFrameUserId = userFrameUserId;
        userFirstNameLabel.setText(firstName);
        populateBookTable();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        viewBooksButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        viewBorrowedBooksButton = new javax.swing.JButton();
        logOutButton = new javax.swing.JButton();
        userFirstNameLabel = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        jPanel2 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        booksTable = new javax.swing.JTable();
        jLabel9 = new javax.swing.JLabel();
        searchTextField = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();
        booksFilterComboBox = new javax.swing.JComboBox<>();
        jScrollPane2 = new javax.swing.JScrollPane();
        bookDetailsTextArea = new javax.swing.JTextArea();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("View Books");
        setLocation(new java.awt.Point(380, 200));

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(255, 255, 255), 1,
true));

        viewBooksButton.setBackground(new java.awt.Color(24, 119, 242));
        viewBooksButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        viewBooksButton.setForeground(new java.awt.Color(255, 255, 255));
        viewBooksButton.setText("View Books");
```

```
        jLabel1.setText("Welcome Back!");

        viewBorrowedBooksButton.setText("View Borrowed Books");
        viewBorrowedBooksButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                viewBorrowedBooksButtonActionPerformed(evt);
            }
        });

        logOutButton.setBackground(new java.awt.Color(255, 0, 0));
        logOutButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logOutButton.setForeground(new java.awt.Color(0, 255, 255));
        logOutButton.setText("Log out");
        logOutButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logOutButtonActionPerformed(evt);
            }
        });

        userFirstNameLabel.setFont(new java.awt.Font("Georgia Pro Black", 0, 24)); // NOI18N
        userFirstNameLabel.setText("First Name");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(viewBorrowedBooksButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                    .addComponent(viewBooksButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGap(34, 34, 34)
                                .addComponent(jLabel1))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(userFirstNameLabel)))
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(6, 6, 6)
                        .addComponent(logOutButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jSeparator1))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jSeparator2)))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(userFirstNameLabel)
                .addGap(3, 3, 3)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 4,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(viewBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(viewBorrowedBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 6,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(logOutButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        booksTable.setAutoCreateRowSorter(true);
        booksTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
```

```
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null},
                {null, null, null, null, null, null, null}
            },
            new String [] {
                "ISBN", "Title", "Author", "Publish Date", "Category", "Status", "Quantity"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        booksTable.setFillsViewportHeight(true);
        booksTable.setShowGrid(true);
        booksTable.getTableHeader().setResizingAllowed(false);
        booksTable.getTableHeader().setReorderingAllowed(false);
        booksTable.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                booksTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(booksTable);

        jLabel9.setText("Search");

        searchTextField.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                searchTextFieldKeyReleased(evt);
            }
        });

        jLabel10.setText("Filter Books by");

        booksFilterComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Category...</font></html>", "Programming", "Science", "Math",
"Philosophy", "English" }));
        booksFilterComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                booksFilterComboBoxActionPerformed(evt);
            }
```

```
        });

        bookDetailsTextArea.setEditable(false);
        bookDetailsTextArea.setColumns(20);
        bookDetailsTextArea.setRows(5);
        jScrollPane2.setViewportView(bookDetailsTextArea);

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel9)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
544, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel10)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(booksFilterComboBox, 0, 215, Short.MAX_VALUE))
                    .addComponent(jScrollPane2)
                    .addComponent(jScrollPane1))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel9)
                    .addComponent(jLabel10)
                    .addComponent(booksFilterComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 395,
Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 205,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void booksTableMouseClicked(java.awt.event.MouseEvent evt) {
```

```
        int selectedRow = 0;
        try {
            if (selectedRow != -1)
                selectedRow = booksTable.getSelectedRow();
        } catch (IndexOutOfBoundsException ex) {
            System.out.println("SelectedRow: " + ex.getMessage());
            return;
        }

        String isbn = String.valueOf(booksTable.getValueAt(selectedRow, 0));
        String title = String.valueOf(booksTable.getValueAt(selectedRow, 1));
        String author = String.valueOf(booksTable.getValueAt(selectedRow, 2));
        String publishDate = String.valueOf(booksTable.getValueAt(selectedRow, 3));
        String category = String.valueOf(booksTable.getValueAt(selectedRow, 4));
        String status = String.valueOf(booksTable.getValueAt(selectedRow, 5));
        String quantity = String.valueOf(booksTable.getValueAt(selectedRow, 6));

        final int CLICKED_2_TIMES = 2;
        if (evt.getClickCount() == CLICKED_2_TIMES) {
            String bookDetails = "<html><b>Title:</b> " + title + "<br>" +
                    "<b>Author:</b> " + author + "<br>" +
                    "<b>ISBN:</b> " + isbn + "<br>" +
                    "<b>Publish Date:</b> " + publishDate + "<br>" +
                    "<b>Category:</b> " + category + "<br>" +
                    "<b>Status:</b> " + status + "<br>" +
                    "<b>Quantity:</b> " + quantity + "</html>";

            String[] options = {"Borrow", "Close"};

            int response = JOptionPane.showOptionDialog(this, bookDetails, "Book Details",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE, null, options, 1);

            final int BORROW = 0;
            if (response == BORROW) {
                Date borrowDate = new Date(Calendar.getInstance().getTimeInMillis());

                Calendar returnDateCalendar = Calendar.getInstance();
                returnDateCalendar.setTime(borrowDate);
                final int BORROWED_DAYS_LIMT = 15;
                returnDateCalendar.add(Calendar.DAY_OF_YEAR, BORROWED_DAYS_LIMT);

                Date returnDate = new Date(returnDateCalendar.getTimeInMillis());

                insertNewBorrowedRecord(borrowDate, returnDate, isbn);

                populateBookTable();
            }
        }

        StringBuilder bookDetails = new StringBuilder();

        bookDetails.append("ISBN             :  ").append(isbn).append("\n");
        bookDetails.append("Title              :  ").append(title).append("\n");
        bookDetails.append("Author          :  ").append(author).append("\n");
        bookDetails.append("Publish Date:  ").append(publishDate).append("\n");
        bookDetails.append("Category      :  ").append(category).append("\n");
        bookDetails.append("Status           :  ").append(status).append("\n");
        bookDetails.append("Quantity       :  ").append(quantity).append("\n");

        bookDetailsTextArea.setText(bookDetails.toString());

    }

    private void insertNewBorrowedRecord(Date borrowDate, Date returnDate, String isbn) {
        String insertQuery = "INSERT INTO borrowed_books (user_id, book_isbn, borrowed_date,
returned_date) "
                + "VALUES (?, ?, ?, ?)";

        String selectQuery =  "SELECT quantity, status FROM books WHERE isbn = ?";

        try (PreparedStatement bookQuantityAndStatus = connection.prepareStatement(selectQuery)) {
            PreparedStatement borrowedBooks = connection.prepareStatement(insertQuery);

            bookQuantityAndStatus.setLong(1, Long.parseLong(isbn));
            ResultSet bookResultSet = bookQuantityAndStatus.executeQuery();

            if (bookResultSet.next()) {
```

```
                int bookQuantity = bookResultSet.getInt("quantity");
                String bookStatus = bookResultSet.getString("status");

                if (bookQuantity > 0 && bookStatus.equals("Available")) {
                    borrowedBooks.setInt(1, userFrameUserId);
                    borrowedBooks.setLong(2, Long.parseLong(isbn));
                    borrowedBooks.setString(3, borrowDate.toString());
                    borrowedBooks.setString(4, returnDate.toString());
                    borrowedBooks.executeUpdate();

                    JOptionPane.showMessageDialog(this, "Book Borrowed Successfully!", "Notice",
JOptionPane.INFORMATION_MESSAGE);

                }
                else {
                    JOptionPane.showMessageDialog(this, "This book is currently not available/out
of stock", "Reminder", JOptionPane.WARNING_MESSAGE);
                    return;
                }
            }

        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "You cannot borrow the same book twice!",
"Reminder", JOptionPane.ERROR_MESSAGE);
            System.out.println("Insert New Borrowed Record: " + ex.getMessage());
            return;
        }

        decrementBookQuantity(isbn);
        updateBookStatusWhenQuantityIsZero(isbn);
    }

    private void decrementBookQuantity(String isbn) {
        String updateQuery = "UPDATE books "
                + "SET quantity = quantity - 1 "
                + "WHERE isbn = ? AND quantity > 0";

        try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {
            preparedStatement.setString(1, isbn);
            preparedStatement.executeUpdate();

        } catch (SQLException ex) {
            System.out.println("Decrement Book Quantity Operation: " + ex.getMessage());
        }
    }

    private void updateBookStatusWhenQuantityIsZero(String isbn) {
        String updateQuery = "UPDATE books "
                + "SET status = 'Not Available' "
                + "WHERE isbn = ? AND quantity = 0";

        try (PreparedStatement preparedStatement = connection.prepareStatement(updateQuery)) {
            preparedStatement.setString(1, isbn);
            preparedStatement.executeUpdate();

        } catch (SQLException ex) {
            System.out.println("Update Book Status Operation: " + ex.getMessage());
        }
    }

    private void logOutButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this,
                "Are you sure you want to log out?",
                "Log out Confirmation",
                JOptionPane.YES_NO_OPTION);

        final int YES = 0;
        if (response == YES) {
            new LogInFrame().setVisible(true);
            dispose();
        }
    }

    private void viewBorrowedBooksButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new UserViewBorrowedBooksFrame().setVisible(true);
        dispose();
    }
```

```
private void booksFilterComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    booksTableModel.setRowCount(0);

    String selectedCategory = booksFilterComboBox.getSelectedItem().toString();
    String selectQuery = "SELECT * FROM books " +
           "WHERE category = ?";

    try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
        preparedStatement.setString(1, selectedCategory);
        ResultSet resultSet = preparedStatement.executeQuery();

        int columnCount = resultSet.getMetaData().getColumnCount();

        while (resultSet.next()) {
            Object[] row = new Object[columnCount];

            for (int i = 1; i <= columnCount; i++) {
                row[i-1] = resultSet.getObject(i);
            }

            booksTableModel.addRow(row);
        }
    } catch (SQLException ex) {
        System.out.println("Books Filtering Operation: " + ex.getMessage());
    }

    final int SELECT = 0;
    if (booksFilterComboBox.getSelectedIndex() == SELECT) {
        populateBookTable();
    }
}

private void searchTextFieldKeyReleased(java.awt.event.KeyEvent evt) {
    booksTableModel.setRowCount(0);

    String userInput = searchTextField.getText();
    String searchQuery = "SELECT * FROM books "
            + "WHERE title LIKE ? OR author LIKE ?";

    try (PreparedStatement preparedStatement = connection.prepareStatement(searchQuery)) {
        preparedStatement.setString(1, "%" + userInput + "%");
        preparedStatement.setString(2, "%" + userInput + "%");
        ResultSet resultSet = preparedStatement.executeQuery();

        int columnCount = resultSet.getMetaData().getColumnCount();

        while (resultSet.next()) {
            Object[] row = new Object[columnCount];

            for (int i = 1; i <= columnCount; i++) {
                row[i-1] = resultSet.getObject(i);
            }

            booksTableModel.addRow(row);
        }
    } catch (SQLException ex) {
        System.out.println("Search Operation: " + ex.getMessage());
    }
}

private void populateBookTable() {
    booksTableModel = (DefaultTableModel) booksTable.getModel();
    booksTableModel.setRowCount(0);

    String selectQuery = "SELECT * FROM books";

    try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
        ResultSet resultSet = preparedStatement.executeQuery();
        int columnCount = resultSet.getMetaData().getColumnCount();

        while (resultSet.next()) {
            Object[] row = new Object[columnCount];

            for (int i = 1; i <= columnCount; i++) {
                row[i-1] = resultSet.getObject(i);
            }
```

```
                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Populate Book Table Operation: " + ex.getMessage());
        }
    }



    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new UserViewBooksFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextArea bookDetailsTextArea;
    private javax.swing.JComboBox<String> booksFilterComboBox;
    private javax.swing.JTable booksTable;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JButton logOutButton;
    private javax.swing.JTextField searchTextField;
    private javax.swing.JLabel userFirstNameLabel;
    private javax.swing.JButton viewBooksButton;
    private javax.swing.JButton viewBorrowedBooksButton;
    // End of variables declaration
}
```

UserViewBorrowedBooksFrame

```
import com.formdev.flatlaf.themes.FlatMacLightLaf;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import javax.swing.JOptionPane;
public class UserViewBorrowedBooksFrame extends javax.swing.JFrame {
    private Connection connection = DatabaseConnection.getConnection();
    private DefaultTableModel booksTableModel;
    public static int userFrameUserId;
    public static String firstName;

    public UserViewBorrowedBooksFrame() {
        initComponents();
        UserViewBooksFrame.userFrameUserId = userFrameUserId;
        userFirstNameLabel.setText(firstName);
        populateBookTable();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        ViewBooksButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        burrowBooksButton = new javax.swing.JButton();
        logOutButton = new javax.swing.JButton();
        userFirstNameLabel = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        jPanel2 = new javax.swing.JPanel();
```

```
        jScrollPane1 = new javax.swing.JScrollPane();
        booksTable = new javax.swing.JTable();
        jLabel9 = new javax.swing.JLabel();
        searchTextField = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();
        booksFilterComboBox = new javax.swing.JComboBox<>();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("View Borrowed Books");
        setLocation(new java.awt.Point(380, 200));

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(255, 255,
255)));

        ViewBooksButton.setText("View Books");
        ViewBooksButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ViewBooksButtonActionPerformed(evt);
            }
        });

        jLabel1.setText("Welcome Back!");

        burrowBooksButton.setBackground(new java.awt.Color(24, 119, 242));
        burrowBooksButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        burrowBooksButton.setForeground(new java.awt.Color(255, 255, 255));
        burrowBooksButton.setText("View Borrowed Books");

        logOutButton.setBackground(new java.awt.Color(255, 0, 0));
        logOutButton.setFont(new java.awt.Font("Segoe UI", 1, 15)); // NOI18N
        logOutButton.setForeground(new java.awt.Color(0, 255, 255));
        logOutButton.setText("Log out");
        logOutButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logOutButtonActionPerformed(evt);
            }
        });

        userFirstNameLabel.setFont(new java.awt.Font("Georgia Pro Black", 0, 24)); // NOI18N
        userFirstNameLabel.setForeground(new java.awt.Color(51, 51, 51));
        userFirstNameLabel.setText("First Name");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jSeparator1, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(burrowBooksButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(ViewBooksButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGap(32, 32, 32)
                                .addComponent(jLabel1))
                            .addComponent(userFirstNameLabel)
                            .addComponent(logOutButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jSeparator2))
                        .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(userFirstNameLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(3, 3, 3)
                .addComponent(ViewBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(burrowBooksButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 6,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(logOutButton, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        booksTable.setAutoCreateRowSorter(true);
        booksTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
```

```
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Title", "Category", "Borrowed Date", "Return Date"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, true, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    booksTable.setFillsViewportHeight(true);
    booksTable.setShowGrid(true);
    booksTable.getTableHeader().setResizingAllowed(false);
    booksTable.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(booksTable);

    jLabel9.setText("Search");

    searchTextField.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            searchTextFieldKeyReleased(evt);
        }
    });

    jLabel10.setText("Filter Books by");

    booksFilterComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"<html><font color='gray'>Category...</font></html>", "Programming", "Science", "Math",
"Philosophy", "English" }));
    booksFilterComboBox.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                booksFilterComboBoxActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel9)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
544, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel10)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(booksFilterComboBox, 0, 215, Short.MAX_VALUE))
                    .addComponent(jScrollPane1))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel9)
                    .addComponent(jLabel10)
                    .addComponent(booksFilterComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 627,
Short.MAX_VALUE)
                .addContainerGap())
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addContainerGap())
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        );

        pack();
    }// </editor-fold>

    private void logOutButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int response = JOptionPane.showConfirmDialog(this,
                "Are you sure you want to log out?",
                "Log out Confirmation",
                JOptionPane.YES_NO_OPTION);
```

```
        final int YES = 0;
        if (response == YES) {
            new LogInFrame().setVisible(true);
            dispose();
        }
    }

    private void ViewBooksButtonActionPerformed(java.awt.event.ActionEvent evt) {
        new UserViewBooksFrame().setVisible(true);
        dispose();
    }

    private void booksFilterComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
        booksTableModel.setRowCount(0);

        String selectedCategory = booksFilterComboBox.getSelectedItem().toString();

        String selectQuery = "SELECT books.title, books.category, borrowed_books.borrowed_date,
borrowed_books.returned_date " +
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id " +
                "WHERE books.category LIKE ? AND account.user_id = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setString(1, selectedCategory);
            preparedStatement.setInt(2, userFrameUserId);
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Books Filtering Operation: " + ex.getMessage());
        }

        final int SELECT = 0;
        if (booksFilterComboBox.getSelectedIndex() == SELECT) {
            populateBookTable();
        }
    }

    private void searchTextFieldKeyReleased(java.awt.event.KeyEvent evt) {
        booksTableModel.setRowCount(0);

        String userInput = searchTextField.getText();

        String selectQuery = "SELECT books.title, books.category, borrowed_books.borrowed_date,
borrowed_books.returned_date " +
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id " +
                "WHERE books.title LIKE ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setString(1, "%" + userInput + "%");
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
```

```
        } catch (SQLException ex) {
            System.out.println("Search Operation: " + ex.getMessage());
        }
    }

    private void populateBookTable() {
        booksTableModel = (DefaultTableModel) booksTable.getModel();
        booksTableModel.setRowCount(0);

        String selectQuery = "SELECT books.title, books.category, borrowed_books.borrowed_date,
borrowed_books.returned_date " +
                "FROM borrowed_books " +
                "JOIN books ON borrowed_books.book_isbn = books.isbn " +
                "JOIN account ON borrowed_books.user_id = account.user_id " +
                "WHERE account.user_id = ?";

        try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {
            preparedStatement.setInt(1, userFrameUserId);
            ResultSet resultSet = preparedStatement.executeQuery();

            int columnCount = resultSet.getMetaData().getColumnCount();

            while (resultSet.next()) {
                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {
                    row[i-1] = resultSet.getObject(i);
                }

                booksTableModel.addRow(row);
            }
        } catch (SQLException ex) {
            System.out.println("Populate Table Operation: " + ex.getMessage());
        }
    }


    public static void main(String args[]) {
        try {
            UIManager.setLookAndFeel(new FlatMacLightLaf());
        } catch (UnsupportedLookAndFeelException ex) {
             System.out.println(ex.getMessage());
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new UserViewBorrowedBooksFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton ViewBooksButton;
    private javax.swing.JComboBox<String> booksFilterComboBox;
    private javax.swing.JTable booksTable;
    private javax.swing.JButton burrowBooksButton;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JButton logOutButton;
    private javax.swing.JTextField searchTextField;
    private javax.swing.JLabel userFirstNameLabel;
    // End of variables declaration
}
```