```
import java.awt.*;
import java.util.List;
import java.util.ArrayList;
import javax.swing.*;

// Book class
class Book {
    String title;
    String author;
    String isbn;
    String genre;
    boolean isAvailable;

    Book(String title, String author, String isbn, String genre) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.genre = genre;
        this.isAvailable = true;
    }

    public String toString() {
        return "[" + isbn + "] " + title + " by " + author + " (" + genre + ") - "
                + (isAvailable ? "✅ Available" : "❌ Borrowed");
    }
}
```

```
// User class
class User {
    String userId;
    String name;
    String contact;
    List<Book> borrowedBooks = new ArrayList<>();

    User(String userId, String name, String contact) {
        this.userId = userId;
        this.name = name;
        this.contact = contact;
    }

    boolean canBorrow() {
        return borrowedBooks.size() < 5;
    }

    void borrowBook(Book book) {
        borrowedBooks.add(book);
    }

    void returnBook(Book book) {
        borrowedBooks.remove(book);
    }
```

```
    public String toString() {
        return name + " (User ID: " + userId + ")";
    }
}

// Librarian class
class Librarian extends User {
    Librarian(String userId, String name, String contact) {
        super(userId, name, contact);
    }

    void addBook(List<Book> books, Book book) {
        books.add(book);
    }

    void registerUser(List<User> users, User user) {
        users.add(user);
    }
}

// GUI-based Library Management System
public class LibraryManagementSystemGUI extends JFrame {
    private final List<Book> books = new ArrayList<>();
    private final List<User> users = new ArrayList<>();
    private final Librarian librarian = new Librarian("L001", "ANN", "ann@library.com");
```

```
private final JTextArea outputArea = new JTextArea(10, 40);

public LibraryManagementSystemGUI() {
    super("📚 Library Management System");

    setLayout(new BorderLayout());
    outputArea.setEditable(false);
    add(new JScrollPane(outputArea), BorderLayout.CENTER);

    // Buttons
    JPanel buttonPanel = new JPanel(new GridLayout(3, 2, 10, 10));
    JButton addBookBtn = new JButton("Add Book");
    JButton registerUserBtn = new JButton("Register User");
    JButton borrowBookBtn = new JButton("Borrow Book");
    JButton returnBookBtn = new JButton("Return Book");
    JButton viewBooksBtn = new JButton("View All Books");
    JButton exitBtn = new JButton("Exit");

    buttonPanel.add(addBookBtn);
    buttonPanel.add(registerUserBtn);
    buttonPanel.add(borrowBookBtn);
    buttonPanel.add(returnBookBtn);
    buttonPanel.add(viewBooksBtn);
    buttonPanel.add(exitBtn);

    add(buttonPanel, BorderLayout.SOUTH);
```

```
    // Event Listeners
    addBookBtn.addActionListener(_ -> addBook());
    registerUserBtn.addActionListener(_ -> registerUser());
    borrowBookBtn.addActionListener(_ -> borrowBook());
    returnBookBtn.addActionListener(_ -> returnBook());
    viewBooksBtn.addActionListener(_ -> viewBooks());
    exitBtn.addActionListener(_ -> System.exit(0));

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(650, 420);
    setLocationRelativeTo(null);
    setVisible(true);
}

private void addBook() {
    String title = JOptionPane.showInputDialog(this, "Enter Book Title:");
    String author = JOptionPane.showInputDialog(this, "Enter Author:");
    String isbn = JOptionPane.showInputDialog(this, "Enter ISBN:");
    String genre = JOptionPane.showInputDialog(this, "Enter Genre:");

    if (title != null && author != null && isbn != null && genre != null) {
        librarian.addBook(books, new Book(title, author, isbn, genre));
        outputArea.append("✅ Book Added: " + title + " (ISBN: " + isbn + ")\n");
    }
}
```

```
    private void registerUser() {
        String userId = JOptionPane.showInputDialog(this, "Enter User ID:");
        String name = JOptionPane.showInputDialog(this, "Enter Name:");
        String contact = JOptionPane.showInputDialog(this, "Enter Contact Info:");

        if (userId != null && name != null && contact != null) {
            librarian.registerUser(users, new User(userId, name, contact));
            outputArea.append(" 👤 User Registered: " + name + " (User ID: " + userId + ")\n");
        }
    }

    private void borrowBook() {
        String userId = JOptionPane.showInputDialog(this, "Enter User ID:");
        String isbn = JOptionPane.showInputDialog(this, "Enter Book ISBN:");
        User borrower = findUser(users, userId);
        Book bookToBorrow = findBook(books, isbn);

        if (borrower != null && bookToBorrow != null && bookToBorrow.isAvailable &&
borrower.canBorrow()) {
            borrower.borrowBook(bookToBorrow);
            bookToBorrow.isAvailable = false;
            outputArea.append(" 📘 Borrowed: " + bookToBorrow.title + " by "
                + borrower.name + " (User ID: " + borrower.userId + ")\n");
        } else {
```

```
            JOptionPane.showMessageDialog(this, "Borrow failed! Check User ID or Book
Availability.");
        }
    }

    private void returnBook() {
        String userId = JOptionPane.showInputDialog(this, "Enter User ID:");
        String isbn = JOptionPane.showInputDialog(this, "Enter Book ISBN:");
        User returner = findUser(users, userId);
        Book bookToReturn = findBook(books, isbn);

        if (returner != null && bookToReturn != null &&
returner.borrowedBooks.contains(bookToReturn)) {
            returner.returnBook(bookToReturn);
            bookToReturn.isAvailable = true;
            outputArea.append(" 📗 Returned: " + bookToReturn.title + " by "
                    + returner.name + " (User ID: " + returner.userId + ")\n");
        } else {
            JOptionPane.showMessageDialog(this, "Return failed! Check User ID or ISBN.");
        }
    }

    private void viewBooks() {
        outputArea.append("\n 📚 All Books:\n");
        if (books.isEmpty()) {
            outputArea.append("No books available.\n");
```

```
    } else {
        for (Book b : books) {
            outputArea.append(b.toString() + "\n");
        }
    }
    outputArea.append("\n");
}

// Helper methods
static Book findBook(List<Book> books, String isbn) {
    for (Book b : books) {
        if (b.isbn.equals(isbn))
            return b;
    }
    return null;
}

static User findUser(List<User> users, String userId) {
    for (User u : users) {
        if (u.userId.equals(userId))
            return u;
    }
    return null;
}

public static void main(String[] args) {
```

```
        SwingUtilities.invokeLater(LibraryManagementSystemGUI::new);
    }
}
```