

```

import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
public class LibraryManagementSystem extends Frame implements ActionListener
{
    // Components of the GUI
    Label titleLabel, authorLabel, searchLabel;
    TextField titleField, authorField, searchField;
    TextArea outputArea;
    Button addButton, searchButton, viewButton;
    // List to store books
    ArrayList<Book>bookList;
    LibraryManagementSystem() {
        // Initialize the book list
        bookList = new ArrayList<>();
        // Create the GUI components
        titleLabel = new Label("Book Title:");
        authorLabel = new Label("Author:");
        searchLabel = new Label("Search Book:");
        titleField = new TextField();
        authorField = new TextField()

        searchField = new TextField();
        outputArea = new TextArea();
        addButton = new Button("Add Book");
        searchButton = new Button("Search Book");
        viewButton = new Button("View All Books");
        // Set layout
        setLayout(new FlowLayout());
        // Add components
        add(titleLabel);
        add(titleField);
        add(authorLabel);
        add(authorField);
        add(addButton);
        add(searchLabel);
        add(searchField);
        add(searchButton);
        add(viewButton);
        add(outputArea);
        // Set dimensions of the Frame and TextArea
        setSize(450, 400);
        outputArea.setSize(350, 200); // Set size for the output area
        outputArea.setEditable(false); // Prevent the output area from direct editing
        setTitle("Library Management System");
        setVisible(true);
        // Add action listeners
        addButton.addActionListener(this);
    }
}

```

```

searchButton.addActionListener(this);
viewButton.addActionListener(this);
// Close the application
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
});
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == addButton) {
        String title = titleField.getText();
        String author = authorField.getText();
        if (!title.isEmpty() && !author.isEmpty()) {
            bookList.add(new Book(title, author));
            outputArea.append("=== Book Added ===\n");
            outputArea.append("Title: " + title + "\n");
            outputArea.append("Author: " + author + "\n");
            outputArea.append("-----\n\n");
            titleField.setText("");
            authorField.setText("");
        } else {
            outputArea.append("Please enter both title and author.\n\n");
        }
    } else if (e.getSource() == searchButton) {
        String searchTitle = searchField.getText();
        boolean found = false;
        outputArea.append("=== Search Results ===\n");
        for (Book book : bookList) {
            if (book.getTitle().equalsIgnoreCase(searchTitle)) {
                outputArea.append("Found: " + book.getTitle() + " by " +
                    book.getAuthor() + "\n");
                found = true;
                break;
            }
        }
        if (!found) {
            outputArea.append("Book not found.\n");
        }
        outputArea.append("-----\n\n");
        searchField.setText("");
    } else if (e.getSource() == viewButton) {
        outputArea.append("=== All Books ===\n");
        if (bookList.isEmpty()) {
            outputArea.append("No books available in the library.\n");
        } else {
            for (Book book : bookList) {
                outputArea.append(book.getTitle() + " by " + book.getAuthor() + "\n");
            }
        }
    }
}

```

```
}  
}  
outputArea.append("-----\n\n");  
}  
}  
public static void main(String[] args) {  
    new LibraryManagementSystem();  
}  
}  
class Book {  
    private String title;  
    private String author;  
    public Book(String title, String author) {  
        this.title = title;  
        this.author = author;  
    }  
    public String getTitle() {  
        return title;  
    }  
    public String getAuthor() {  
        return author;  
    }  
}
```