



PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2021-22

# SOFTWARE TESTING AND QUALITY ASSURANCE MINI PROJECT REPORT ON

## LIBRARY MANAGEMENT SYSTEM

*Performed by*

Aakash Patil – 41247

Saket Gupta – 41256

Under the guidance of  
Prof. Manish Jansari

## Table of Contents:

1	OBJECTIVE	
2	PROBLEM STATEMENT	
3	OUTCOME	
4	REQUIREMENT: SOFTWARE REQUIREMENT HARDWARE REQUIREMENT	
5	THEORY	
6	CODE SCREENSHOTS	
7	OUTPUT SCREENSHOTS	
8	TEST CASES	
9	APPLICATIONS	
10	CONCLUSION	

### OBJECTIVE:

The objective of this mini project is to

- Develop a java application with GUI for Library Management System
- To prepare suitable test cases for developed application
- Prepare a test report for the application

### PROBLEM STATEMENT:

Create a Medical System / relevant system by selecting relevant system environment / platform and programming languages.

Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass / Fail Criteria and judge the acceptance of application developed.

## **OUTCOME:**

Be able to,

- Understand testing tools like Junit and Maven
- Differentiate between different types of testing
- Write tests for different types of applications

## **4.1 SOFTWARE REQUIREMENTS:**

- Ubuntu 18 OS/Windows OS
- MySQL
- JUnit4
- JDK 1.8
- Eclipse IDE
- Maven Surefire Report Plugin

## **4.2 HARDWARE REQUIREMENTS:**

64-bit machine with 8GB RAM, i5 or greater processor, 128GB SSD / 1TB HDD

## **THEORY:**

- **JUnit:**

JUnit is a unit testing framework for Java programming language. JUnit has been important in the development of test-driven development and is one of a family of unit testing frameworks collectively known as xUnit, that originated with Junit.

Features of Junit:

- JUnit is an open-source framework, which is used for writing

and running tests.

- Provides annotations to identify test methods.
- Provides assertions for testing expected results.
- Provides test runners for running tests.
- JUnit tests allow you to write codes faster, which increases quality.
- JUnit is elegantly simple. It is less complex and takes less time.
- JUnit tests can be run automatically, and they check their own results and provide immediate feedback.
- JUnit tests can be organized into test suites containing test cases and even other test suites.
- JUnit shows test progress in a bar that is green if the test is running smoothly, and it turns red when a test fails.

- **Unit Test Case:**

A Unit Test Case is a part of code, which ensures that another part of code (method) works as expected. To achieve the desired results quickly, a test framework is required. JUnit is a perfect unit test framework for Java programming language.

A formal written unit test case is characterized by a known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition.

- **Java Swing:**

Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

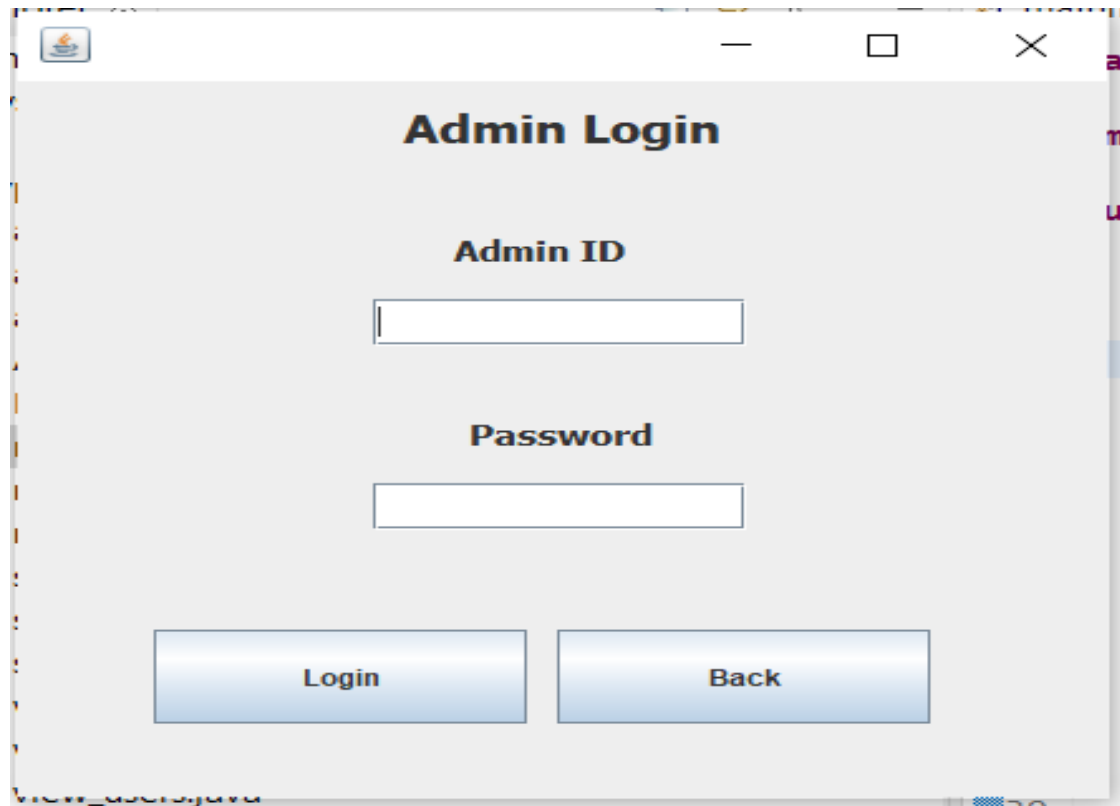
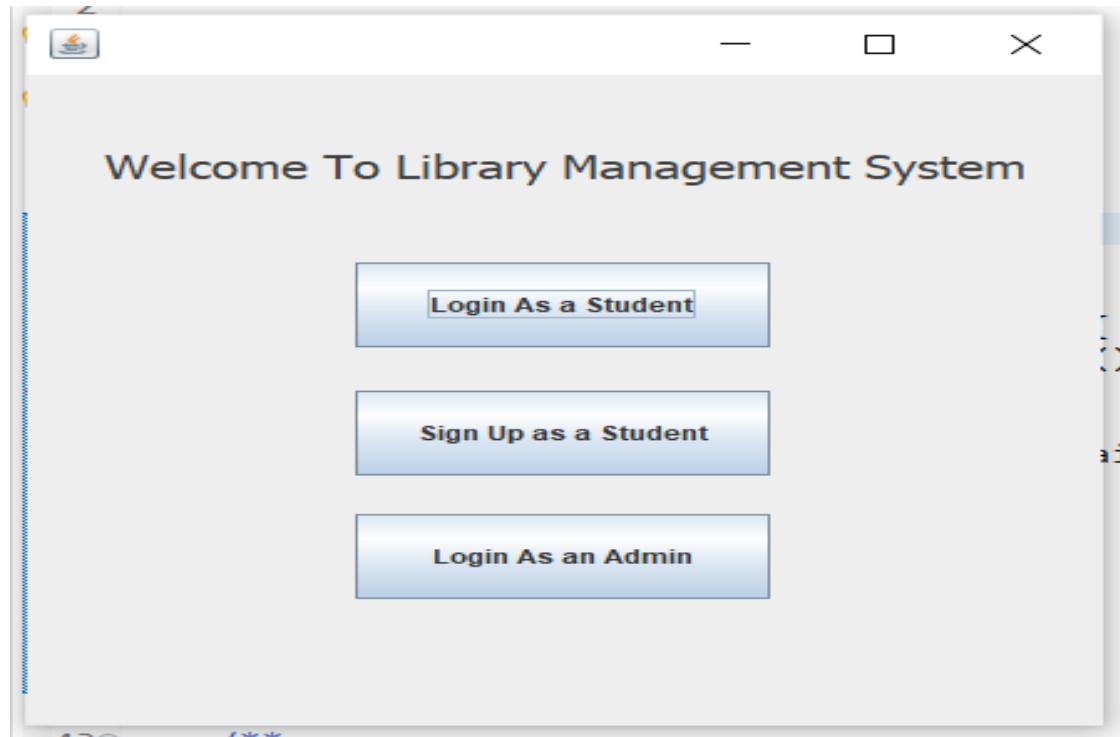
- **Types of Testing:**

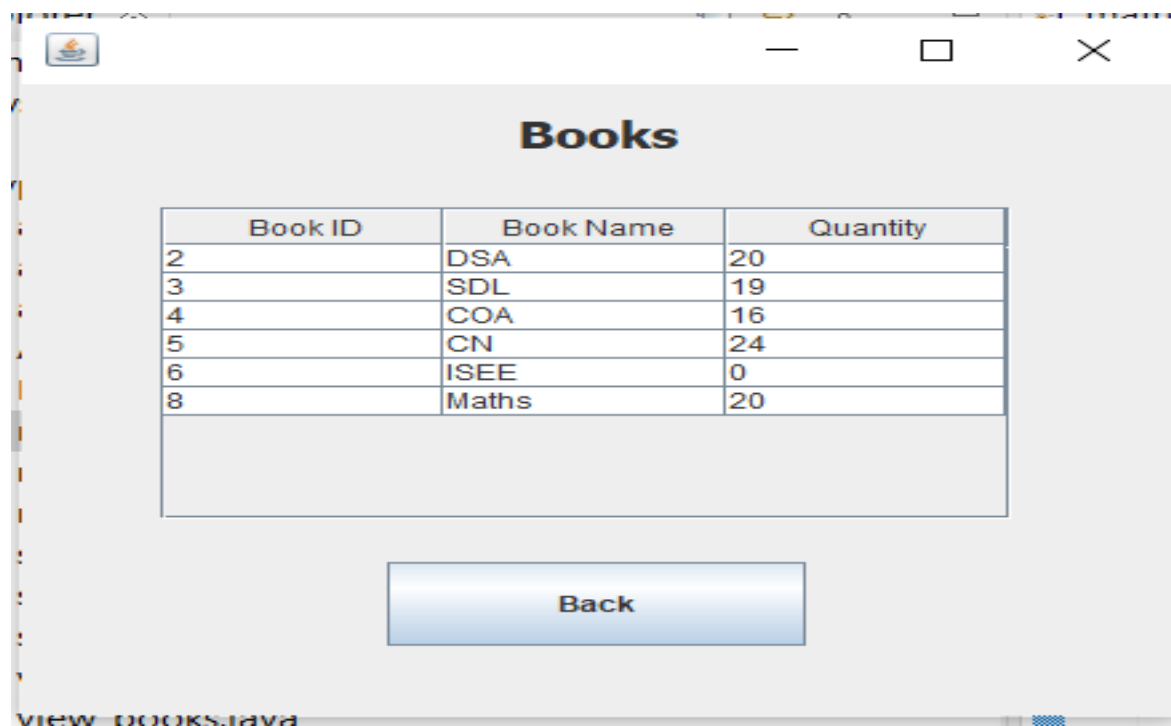
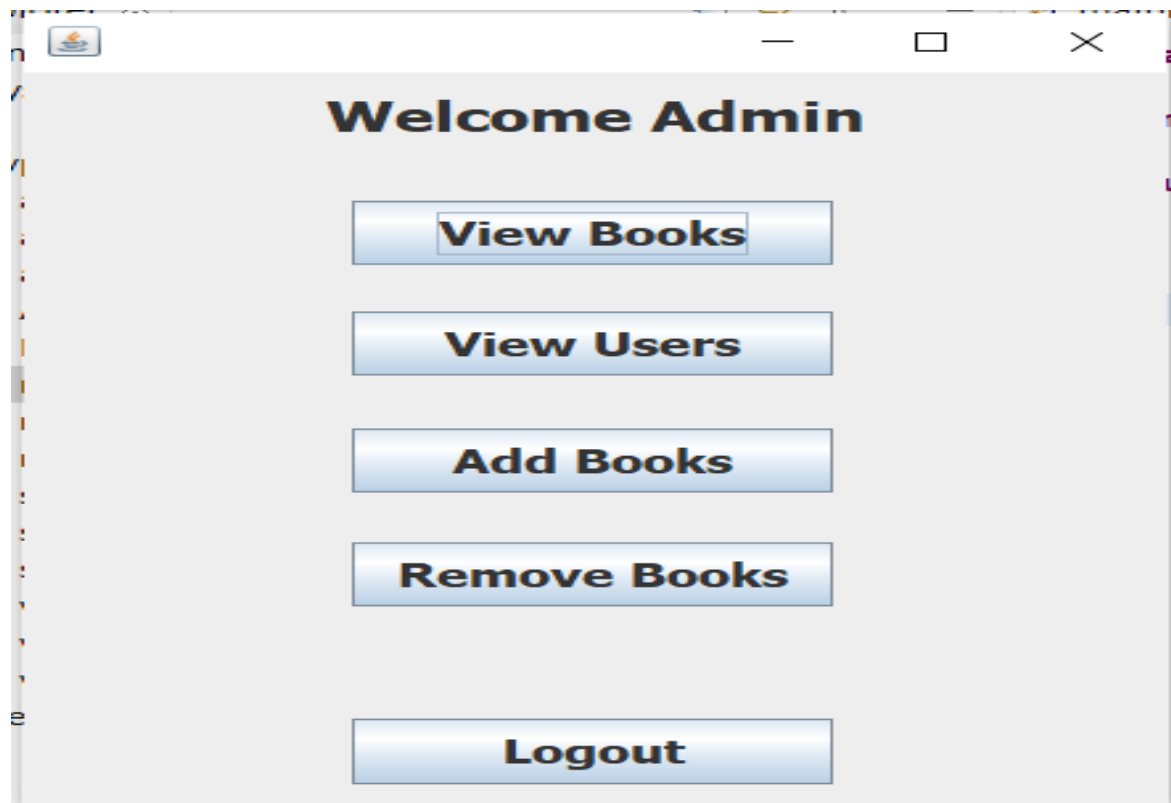
1. Exploratory (Black Box Testing): The purpose of this test is to make sure critical defects are removed before the next levels of testing can start.
2. Functional Test (White Box Testing): Functional testing will be performed to check the functions of the application. The input is fed and the output from the application is validated.
3. User Acceptance Test (Integration Testing): This test focusses on validating the business logic. It allows the end user to complete one final review of the system prior to deployment.


- **Library Management System:**

A Library management system is a software that uses to maintain the record of the library. It contains work like the number of available books in the library, the number of books is issued or returning or renewing a book or late fine charge record, etc. Library Management Systems is software that helps to maintain a database that is useful to enter new books & record books borrowed by the members, with the respective submission dates. Moreover, it also reduces the manual record burden of the librarian.

- **CODE SCREENSHOTS:**






— □ ×

## Users

Roll No.	Name	Books
1	Aakash	
2	Sarish	CN
3	Onkar	
4	Hemant	DSA
5	Kunal	
7	Vaibhav	

Back

— □ ×

## Add Book

Book ID

8

Book Name

maths

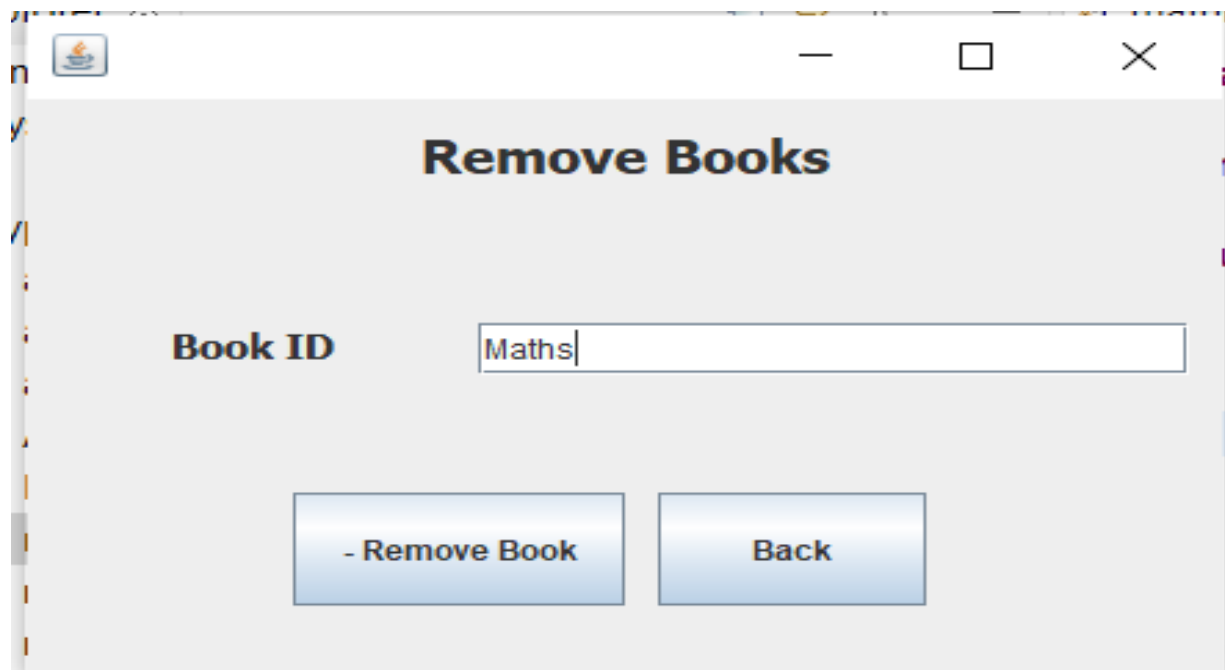
Quantity

20

+ Add Book

Back

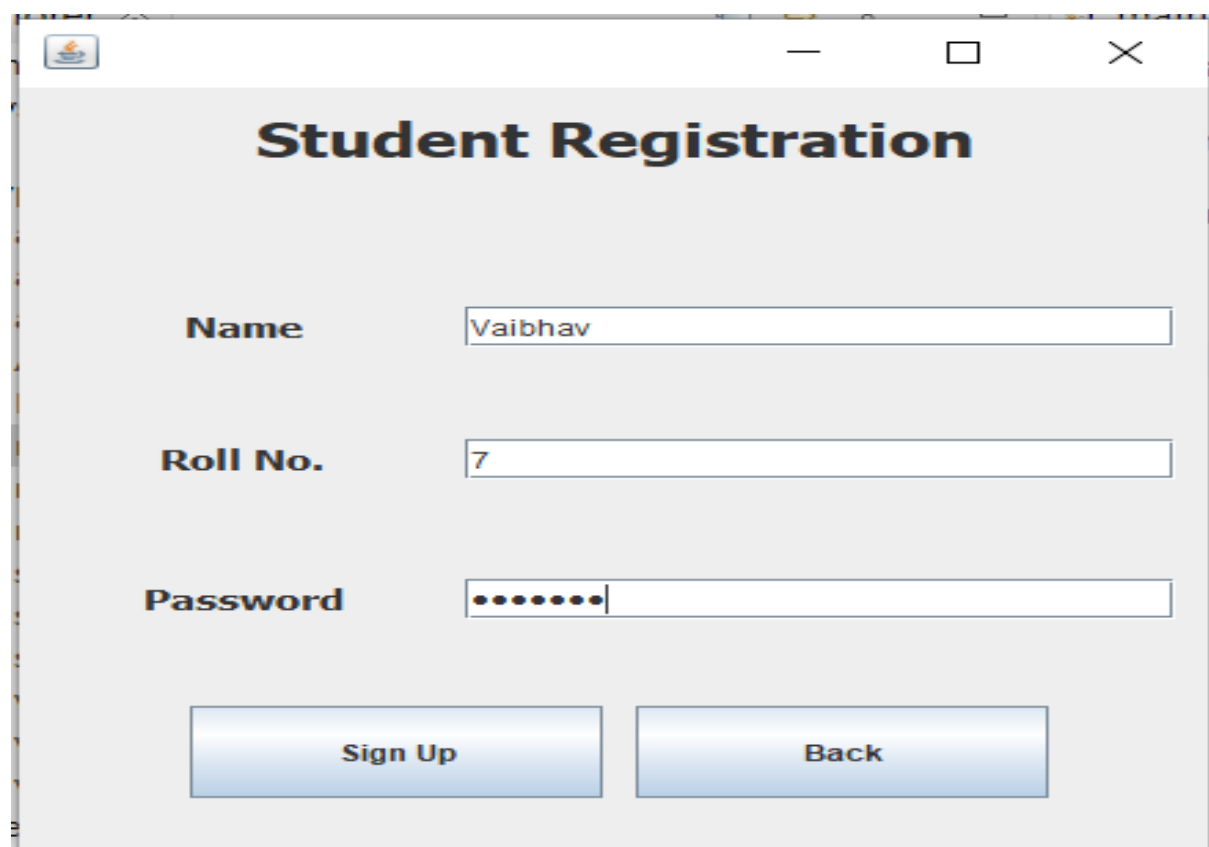




A screenshot of a software window titled "Remove Books". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area has a light gray background. At the top, the title "Remove Books" is centered in a bold, black font. Below the title, there is a label "Book ID" on the left and a text input field on the right containing the text "Maths". At the bottom of the window, there are two buttons: "- Remove Book" on the left and "Back" on the right. Both buttons have a light blue gradient and a thin black border.

**Remove Books**

**Book ID**



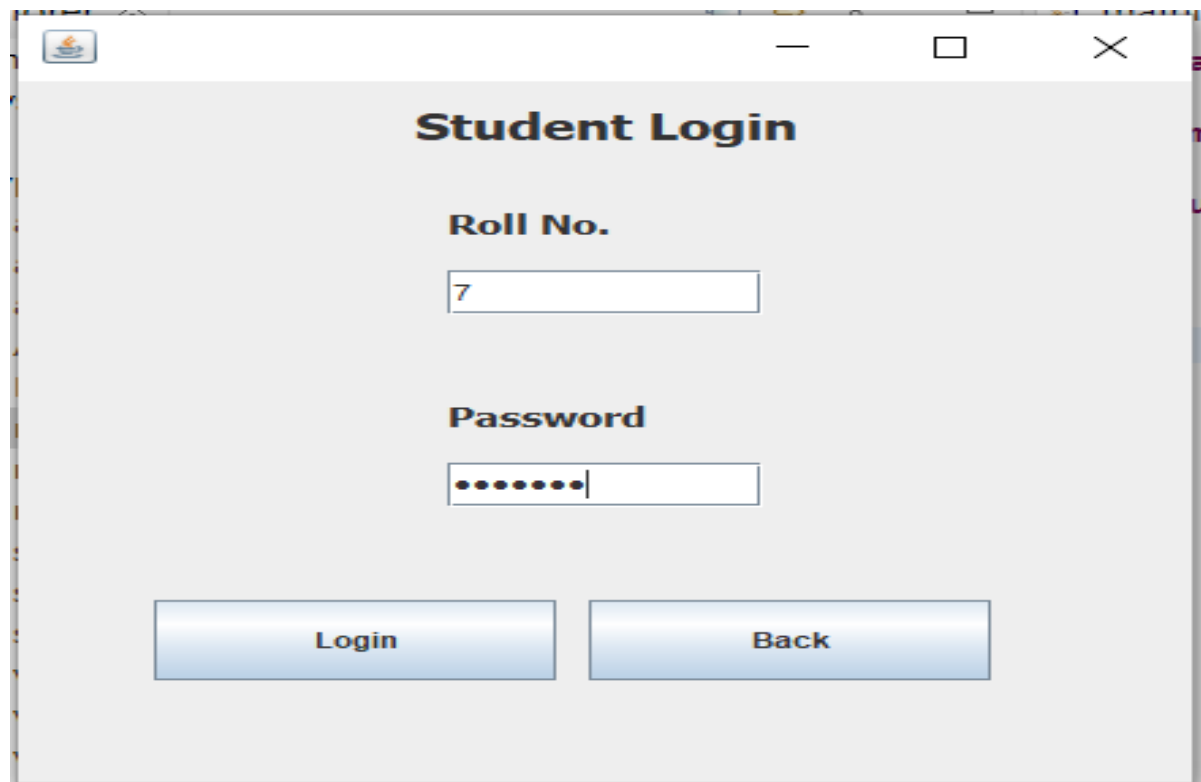
A screenshot of a software window titled "Student Registration". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area has a light gray background. At the top, the title "Student Registration" is centered in a bold, black font. Below the title, there are three labels and their corresponding input fields: "Name" with a text input field containing "Vaibhav", "Roll No." with a text input field containing "7", and "Password" with a text input field containing seven dots. At the bottom of the window, there are two buttons: "Sign Up" on the left and "Back" on the right. Both buttons have a light blue gradient and a thin black border.

**Student Registration**

**Name**

**Roll No.**

**Password**



A screenshot of a Java Swing window titled "Student Login". The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area has a light gray background. At the top, the text "Student Login" is displayed in a bold, black, sans-serif font. Below this, the label "Roll No." is followed by a text input field containing the number "7". Further down, the label "Password" is followed by a password input field represented by a series of dots. At the bottom of the window, there are two rectangular buttons with a blue gradient and black text: "Login" on the left and "Back" on the right.

**Student Login**

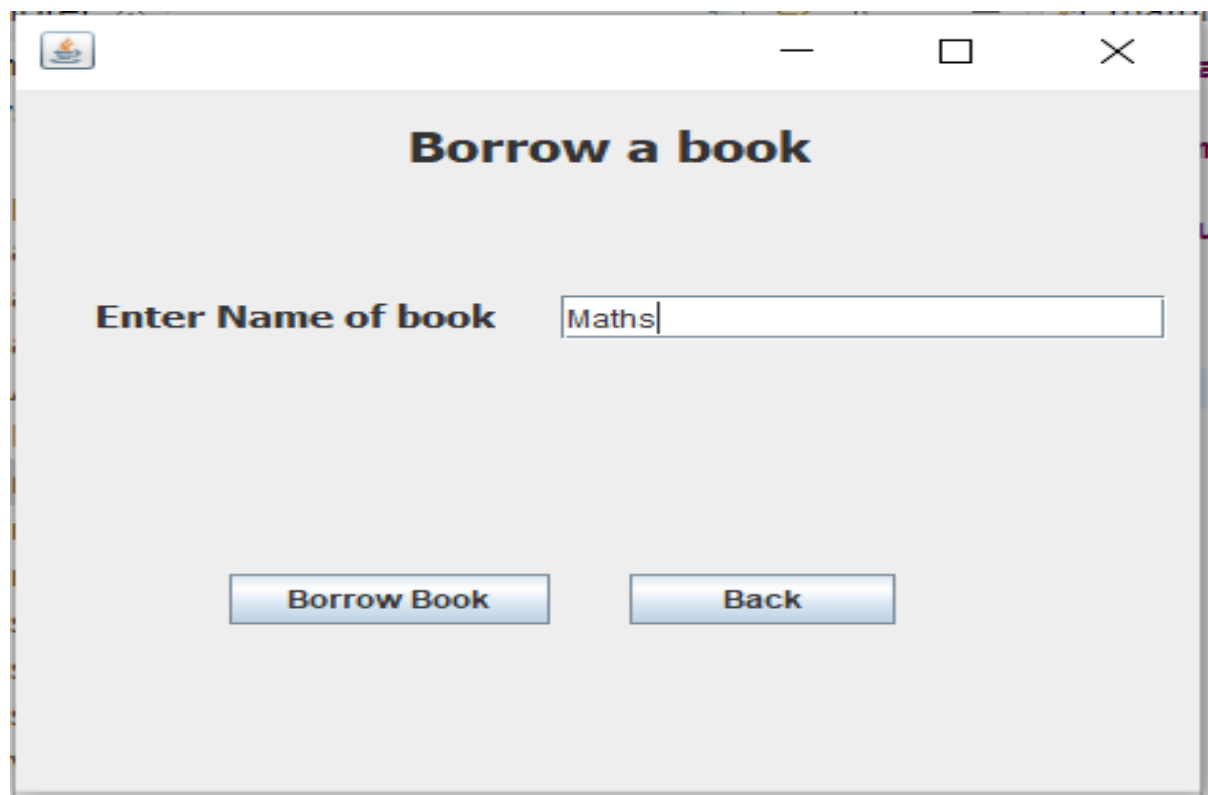
Roll No.

7

Password

.....

Login Back



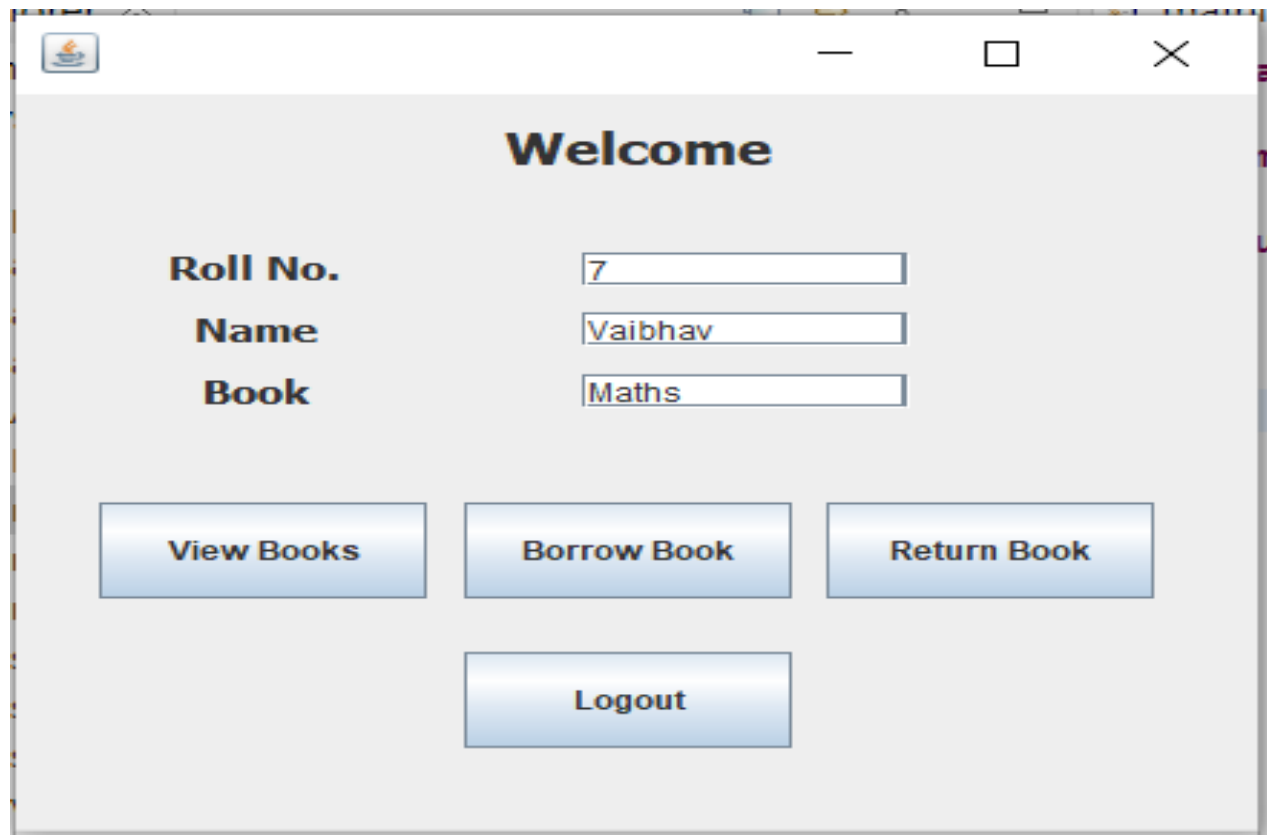
A screenshot of a Java Swing window titled "Borrow a book". The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area has a light gray background. At the top, the text "Borrow a book" is displayed in a bold, black, sans-serif font. Below this, the label "Enter Name of book" is followed by a text input field containing the word "Maths". At the bottom of the window, there are two rectangular buttons with a blue gradient and black text: "Borrow Book" on the left and "Back" on the right.

**Borrow a book**

Enter Name of book

Maths

Borrow Book Back



A Java Swing window titled "Welcome" with a light gray background. It features three input fields for "Roll No.", "Name", and "Book", each with a blue button below it. The "Roll No." field contains "7", "Name" contains "Vaibhav", and "Book" contains "Maths". There are also three buttons: "View Books", "Borrow Book", and "Return Book", and a "Logout" button at the bottom.

**Welcome**

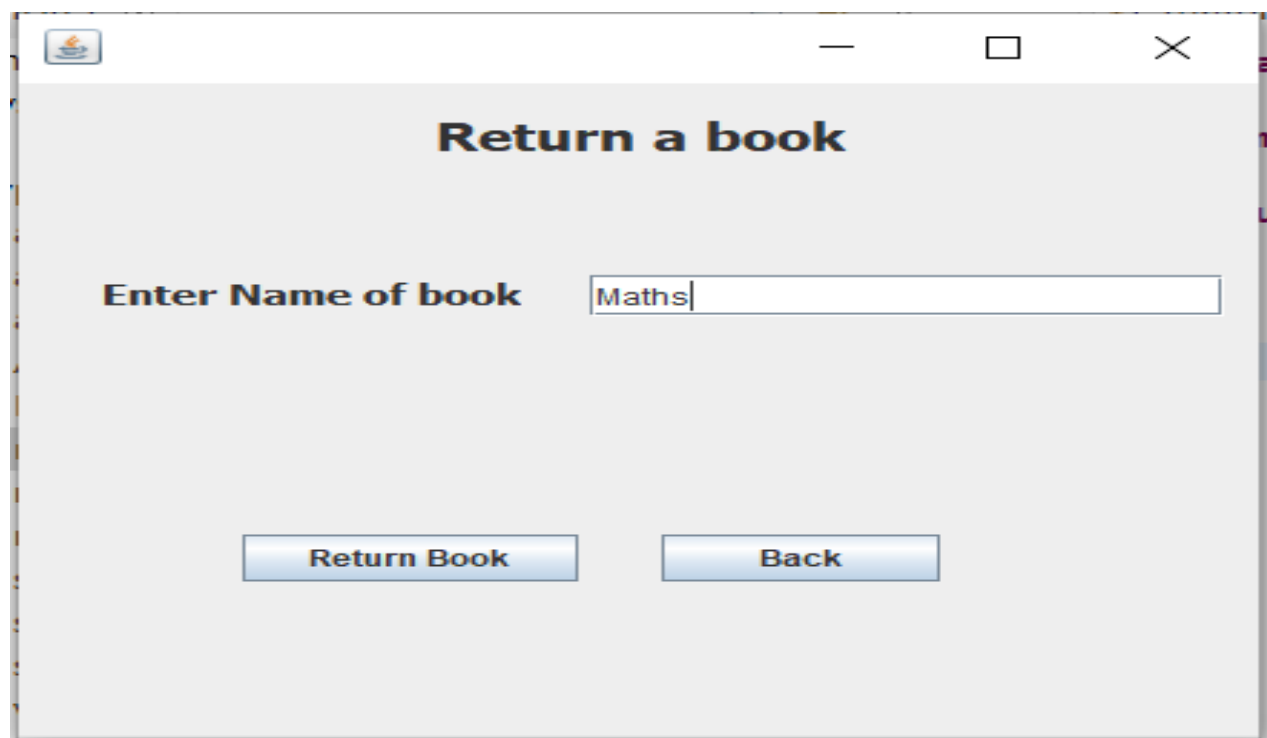
**Roll No.**

**Name**

**Book**

**View Books** **Borrow Book** **Return Book**

**Logout**



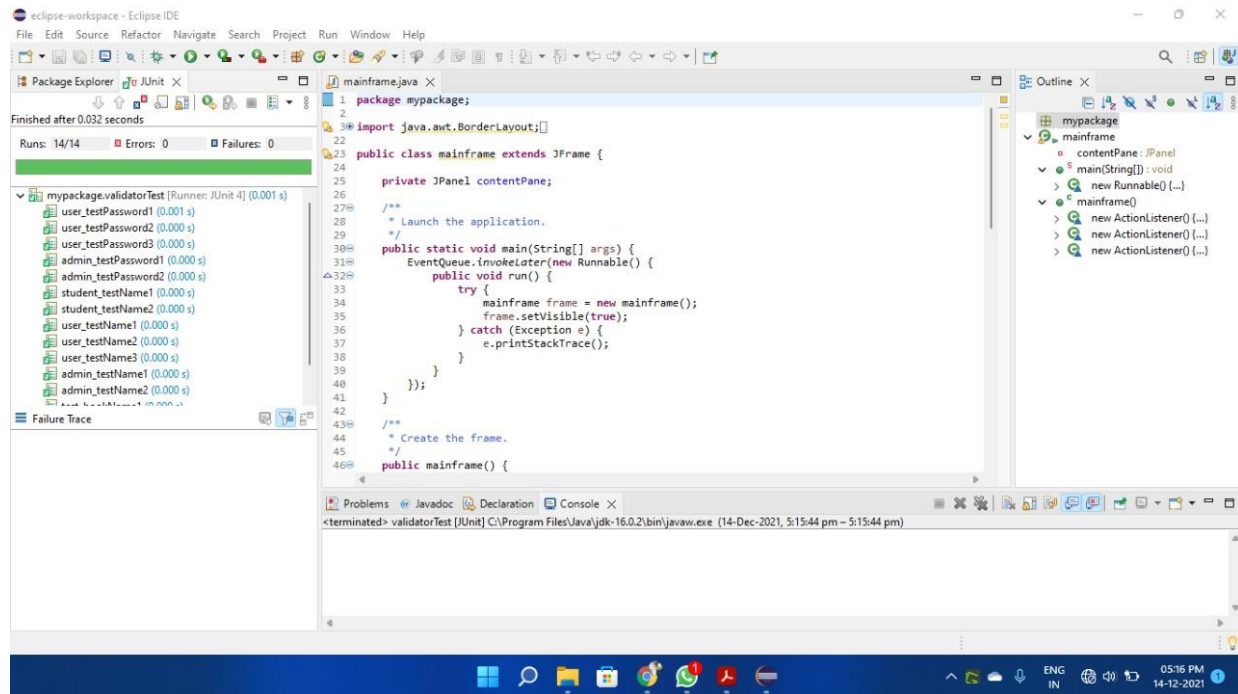
A Java Swing window titled "Return a book" with a light gray background. It features a single input field for "Enter Name of book" containing "Maths". There are two buttons: "Return Book" and "Back".

**Return a book**

**Enter Name of book**

**Return Book** **Back**

- Output of Testing and Testing Report:



```
C:\Users\meena\Documents\Aakash\College\Third Year\First Sem\Practicals\
[INFO] Scanning for projects...
[INFO]
[INFO] -----< MavenTestReportDemo:MavenTestProject >-----
[INFO] Building MavenTestProject 0.0.1-SNAPSHOT
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MavenTestProject ---
[WARNING] Using platform encoding <Cp1252 actually> to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MavenTestProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ MavenTestProject ---
[WARNING] Using platform encoding <Cp1252 actually> to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ MavenTestProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ MavenTestProject ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.848 s
[INFO] Finished at: 2018-09-20T15:53:21+05:30
[INFO]
'cmd' is not recognized as an internal or external command,
operable program or batch file.
E:\MavenTestProject>mvn test -DTest-AllTest
[INFO] Scanning for projects...
[INFO]
[INFO] -----< MavenTestReportDemo:MavenTestProject >-----
[INFO] Building MavenTestProject 0.0.1-SNAPSHOT
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MavenTestProject ---
[WARNING] Using platform encoding <Cp1252 actually> to copy filtered resources,
i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MavenTestProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ MavenTestProject ---
```

## TEST CASES:

Name	Input	Expected output	Description
admin_testname1()	admin	false	Invalid Admin Name
admin_testname2()	12345	true	Valid Name
Admin_testpassword1()	admin	true	Valid Password
Admin_testpassword2()	""	false	Invalid Password
User_testname1()	41247	true	Valid user name
User_testname2()	""	false	Invalid user name
User_testname3()	41256	true	Valid user name
User_testpassword1()	""	false	Invalid format
User_testpassword2()	Saket@1234	true	Valid format
User_testpassword3()	Saket1234	false	Invalid format
Test_bookName1()	""	false	Invalid
Test_bookName2()	DSA Fundamentals	true	Valid- Any valid String accepted
Student_testName1()	41247	true	Valid – Roll number in Valid Range
Student_testName2()	01	false	Invalid input

- **APPLICATIONS:**

1. Automated Library Management System.
2. Does not require any human involvement.
3. Does not require any paperwork to maintain records.

**CONCLUSION:**

Successfully implemented Library Management System and performed the testing using Junit and Maven. Generated a test case report using maven-surefire plugin.