DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

Author Name	NNM24IS089
Title	JAVA TASK
Paper/Submission ID	4350279
Submitted by	nnm24is076@nmamit.in
Submission Date	2025-09-11 14:33:14
Total Pages, Total Words	8, 615
Document type	Assignment

Result Information

Similarity **0 %**

Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

0	0	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE

Create a Library Management System in Java that allows users to:

- **Add new books (with title, author, and ISBN)**
- **Issue and return books**
- **Track availability**

📋 **Requirements:**

- **Use class hierarchies, inheritance, and interfaces (e.g., interface Issueable)**
- **Write unit tests using JUnits for at least 3 methods (e.g., issueBook(), returnBook(), isAvailable())**

CODE:

```
package libraryManagementSystem;

public interface Issue {

        boolean issueBook();

        boolean returnBook();

        boolean isAvailable();

    }


package libraryManagementSystem;

    public class Book implements Issue {

        private String title;

        private String author;

        private String isbn;

        private boolean available;

        public Book(String title, String author, String isbn) {

            this.title = title;

            this.author = author;

            this.isbn = isbn;

            this.available = true;

        }

        @Override

        public boolean issueBook() {

            if (available) {
```

```
            available = false;
            return true;
        }
        return false;
    }
    @Override
    public boolean returnBook() {
        if (!available) {
            available = true;
            return true;
        }
        return false;
    }
    @Override
    public boolean isAvailable() {
        return available;
    }
    public String getIsbn() {
        return isbn;
    }

    public void display1() {
        System.out.println(title + " by " + author + " | ISBN: " + isbn + " | " + (available ?
"Available" : "Issued"));
    }
    public void display() {
        System.out.println(title + " by " + author + " | ISBN: " + isbn + " | " + (available ?
"Available" : "Issued"));
    }
}
```

```
package libraryManagementSystem;
import java.util.*;
public class Library {
        private Map<String, Book> books = new HashMap<>();
        private Scanner input = new Scanner(System.in);
        public void addBook(Book book) {
           books.put(book.getIsbn(), book);
        }
  public boolean issueBook(String isbn) {
          Book book = books.get(isbn);
           return book != null && book.issueBook();
        }
        public boolean returnBook(String isbn) {
           Book book = books.get(isbn);
           return book != null && book.returnBook();
        }
        public void showAllBooks() {
           for (Book book : books.values()) {
              book.display();
           }
        }
        public void start() {
           while (true) {
              System.out.println("\n=== Library Menu ===");
              System.out.println("1. Add Book");
              System.out.println("2. Issue Book");
              System.out.println("3. Return Book");
              System.out.println("4. Show All Books");
              System.out.println("5. Exit");
              System.out.print("Choose an option: ");
              int option = input.nextInt();
```

```
input.nextLine();
switch (option) {
    case 1:
        System.out.print("Enter title: ");
        String title = input.nextLine();
        System.out.print("Enter author: ");
        String author = input.nextLine();
        System.out.print("Enter ISBN: ");
        String isbn = input.nextLine();
        addBook(new Book(title, author, isbn));
        System.out.println("Book added!");
        break;
    case 2:
        System.out.print("Enter ISBN to issue: ");
        isbn = input.nextLine();
        if (issueBook(isbn)) {
            System.out.println("Book issued.");
        } else {
            System.out.println("Book not available.");
        }
        break;
    case 3:
        System.out.print("Enter ISBN to return: ");
        isbn = input.nextLine();
        if (returnBook(isbn)) {
            System.out.println("Book returned.");
        } else {
            System.out.println("Invalid return.");
        }
        break;
    case 4:
```

```
                    showAllBooks();
                    break;
                case 5:
                    System.out.println("Exiting...");
                    input.close();
                    return;
                default:
                    System.out.println("Invalid option.");
            }
        }
    }
    public static void main(String[] args) {
        new Library().start();
    }
}
```

```
package libraryManagementSystem;
import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;
public class Test {
    Library library;
    Book book;

    @BeforeEach
    void setUp() {
        library = new Library();
        book = new Book("Test Book", "Test Author", "T001");
        library.addBook(book);
    }
    void testIssueBook() {
```

```
        assertTrue(library.issueBook("T001"));
        assertFalse(book.isAvailable());
    }
    private void assertFalse(boolean available) {
                    // TODO Auto-generated method stub

        }
    void testReturnBook() {
        library.issueBook("T001");
        assertTrue(library.returnBook("T001"));
        assertTrue(book.isAvailable());
    }
    void testAvailability() {
        assertTrue(book.isAvailable());
    }
}
```

Output:

```
=== Library Menu ===
1. Add Book
2. Issue Book
3. Return Book
4. Show All Books
5. Exit
Choose an option: 1
Enter title: the jungle book
Enter author: Rudyard Kipling
Enter ISBN: 67
Book added!

=== Library Menu ===
```

1. Add Book

2. Issue Book

3. Return Book

4. Show All Books

5. Exit

Choose an option: 2

Enter ISBN to issue: 67

Book issued.

=== Library Menu ===

1. Add Book

2. Issue Book

3. Return Book

4. Show All Books

5. Exit

Choose an option: 4

the jungle book by Rudyard Kipling | ISBN: 67 | Issued

=== Library Menu ===

1. Add Book

2. Issue Book

3. Return Book

4. Show All Books

5. Exit

Choose an option: 3

Enter ISBN to return: 67

Book returned.

=== Library Menu ===

1. Add Book

2. Issue Book

3. Return Book

4. Show All Books

5. Exit

Choose an option: 5

Exiting...