📚 Digital Library Book Tracking System using HashSet

Problem Description

You are tasked with developing a **Digital Library Book Tracking System** in Java that manages and tracks books available in a digital library.
The system must store **unique book titles** using a `HashSet<String>`, ensuring that duplicate titles are ignored automatically.

The application should support the following operations:

- Registering new book titles
- Removing a book from the library
- Searching for the alphabetically earliest book
- Displaying all books currently in the library

If an attempt is made to remove a book that does not exist in the library, a **custom exception** named `BookNotFoundException` must be thrown with a specified error message.

This project simulates a real-world **library backend system** and must follow a **modular and testable structure** using clearly defined classes and methods.

Project Structure and Files

File Name	Description
LibrarySystem.java	Implements the library logic and manages book records using a `HashSet<String>`.
BookNotFoundException.java	Custom exception class to handle removal of non-existent books.
LibraryManagementApp.java	Contains the `main()` method, handles user input using `Scanner`, and invokes methods from `LibrarySystem`.

Class Responsibilities

1. LibrarySystem.java

Must declare a private field:

```java
private HashSet<String> bookTitles;
```

Must provide the following public methods with **exact names and signatures**:

```java
public void addBook(String title);
public void removeBook(String title) throws BookNotFoundException;
public void findEarliestBook();
public void displayAllBooks();
```

Method Functionalities

Method	Description
`addBook(String title)`	Adds the book title to the HashSet. Duplicate titles are ignored automatically.
`removeBook(String title)`	Removes the given book title from the HashSet. If it doesn't exist, throw `BookNotFoundException`.
`findEarliestBook()`	Finds and displays the alphabetically earliest book in the HashSet. If the library is empty, print a suitable message.
`displayAllBooks()`	Displays all current books in the library after performing the operation.

2. BookNotFoundException.java

- Must extend the `Exception` class.
- Must include a constructor:

```java
public BookNotFoundException(String message)
```

- Must be thrown when a removal attempt is made for a non-existent book title.

---

3. LibraryManagementApp.java

- Reads input using `Scanner`.
- Adds all book titles using the `addBook()` method.
- Based on the operation type (`"REMOVE"` or `"EARLIEST"`), invokes the corresponding method.
- Must handle `BookNotFoundException` gracefully.
- Must call `displayAllBooks()` after performing the requested operation.

Functional Requirements

✅ Book Registration

- Input accepts an integer `n` followed by `n` space-separated book titles.
- All titles are stored in a `HashSet<String>`.
- Duplicate titles are ignored automatically.

✅ Book Removal

- If the given title exists, remove it.
- If it doesn't exist, throw a `BookNotFoundException` with the appropriate message.

✅ Find Earliest Book

- Identify and print the alphabetically earliest book title.
- If the library is empty, print an appropriate message.

✅ Display All Books

- After every operation, display all current books in the library.

Input Format

```pgsql
First line: An integer n – the number of books to register.
Second line: n space-separated strings – the book titles.
Third line: A string operation – either "REMOVE" or "EARLIEST".
If the operation is "REMOVE": the fourth line contains the book title to remove.
```

Output Format

Scenario	Output Message
Successful registration	`Book added to library: [Title]`
Successful removal	`Book removed from library: [Title]`
Invalid removal	`Alert: Book [Title] not found in library records`
Earliest book found	`Alphabetically earliest book: [Title]`
No books recorded	`No books currently in library`
Final list	`Current library status: Books in record: [Title1, Title2, ...]`

Sample Input 2

Copy code

```
5
Networking AI101 CloudComputing DevOps ML101
EARLIEST
```

Sample Output 2

yaml

Copy code

```
Book added to library: Networking
Book added to library: AI101
Book added to library: CloudComputing
Book added to library: DevOps
Book added to library: ML101
Alphabetically earliest book: AI101
Current library status: Books in record: [Networking, AI101, CloudComputing, DevOps, ML101]
```

Sample Input 1

```
4
JavaBasics Python101 DataStructures Algorithms
REMOVE
Python101
```

Sample Output 1

pgsql

```
Book added to library: JavaBasics
Book added to library: Python101
Book added to library: DataStructures
Book added to library: Algorithms
Book removed from library: Python101
Current library status: Books in record: [JavaBasics, DataStructures, Algorithms]
```

Sample Input 3

```
3
CProgramming JavaCore PythonFundamentals
REMOVE
RubyBasics
```

Sample Output 3

```pgsql
Book added to library: CProgramming
Book added to library: JavaCore
Book added to library: PythonFundamentals
Alert: Book RubyBasics not found in library records
Current library status: Books in record: [CProgramming, JavaCore, PythonFundamentals]
```