Exercise 7: Implementing Constructor and Setter Injection

Scenario:

The library management application requires both constructor and setter injection for better control over bean initialization.

**1. Introduction**

This document explains the implementation of both constructor and setter injection in the Library Management Application using Spring Framework. It also includes an explanation of the application flow and the class structure, as depicted by a flowchart and a class diagram

**2. Updates Made**

**2.1 Configuring Constructor Injection**

Constructor injection is configured in applicationContext.xml for the BookService class.

**Changes Made:**

* The applicationContext.xml was updated to ensure that the BookService bean is instantiated using constructor injection.

**2.2 Configuring Setter Injection**

The BookService class already contains a setter method for BookRepository. This setter method is used for setter injection, which is also configured in the applicationContext.xml.

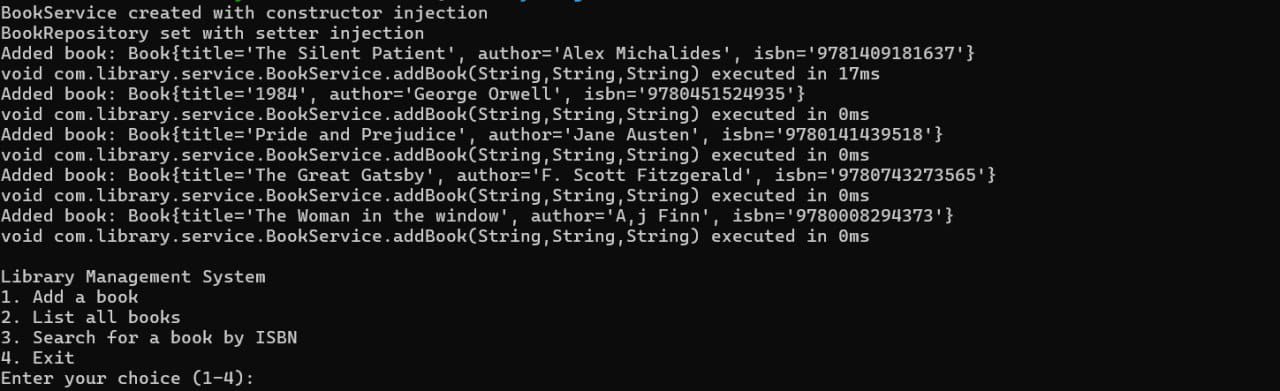
**Changes Made:**

* The applicationContext.xml was updated to include a property element that configures the setter injection for BookService.

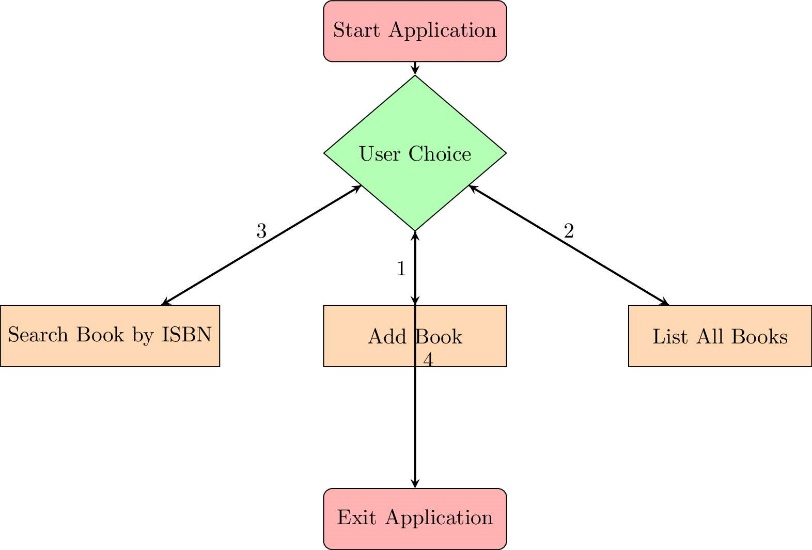
**2.3 Testing the Injection**

* After making the above changes, the application was tested by running the LibraryManagementApplication main class. It run without any error.

**OUTPUT :**



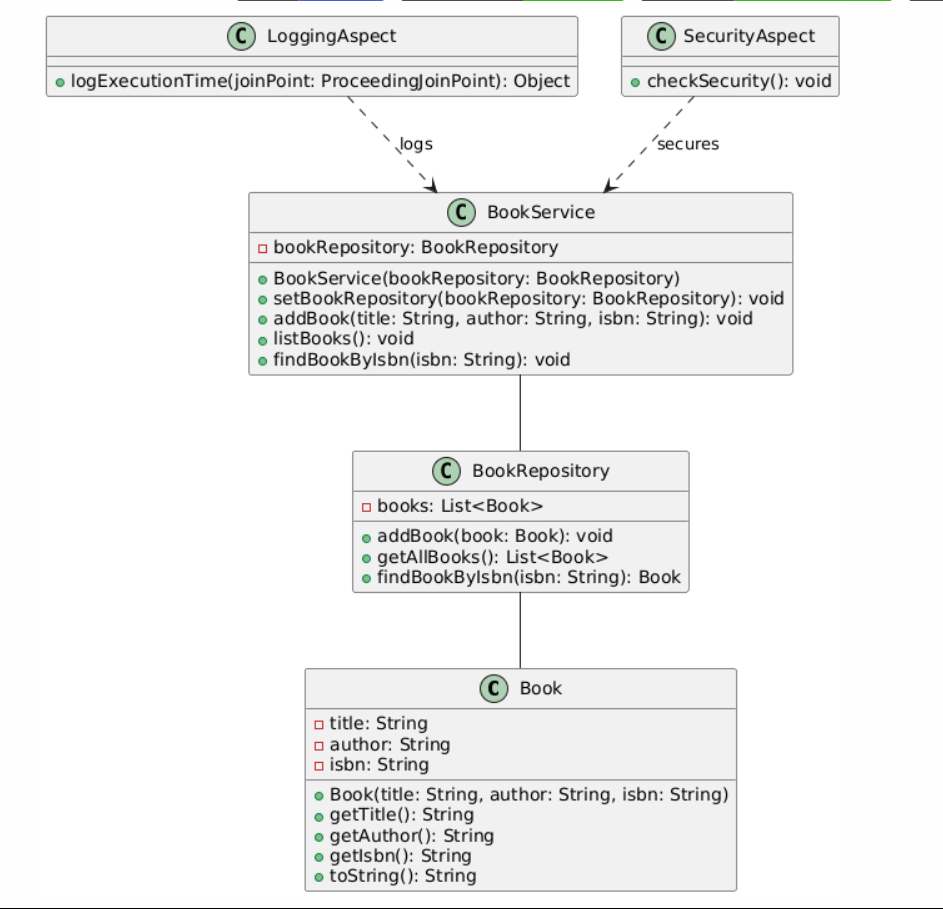
**FLOWCHART of the program :**



The flowchart represents the sequence of operations in the LibraryManagementApplication:

* **Start Application:** The application starts, and the user is prompted to make a choice.
* **User Choice:** The user can choose to add a book, list all books, search for a book by ISBN, or exit the application.
* **Action Based on Choice:** Depending on the user’s choice, one of the following operations occurs:
  + **Add Book:** The user enters book details, and the book is added to the repository.
  + **List All Books:** All books currently in the repository are listed.
  + **Search Book by ISBN:** The user searches for a book by its ISBN.
  + **Exit Application:** The application terminates.

**CLASS DIAGRAM :**



The class diagram represents the structure and relationships between the main classes in the LibraryManagementApplication:

* **Book Class:** Represents the entity that stores information about each book, including title, author, and ISBN.
* **BookRepository Class:** Manages a collection of books, allowing operations like adding books, listing all books, and finding books by ISBN.
* **BookService Class:** Contains business logic to interact with BookRepository. It uses both constructor and setter injection to manage its dependency on BookRepository.
* **LoggingAspect:** Intercepts methods in BookService to log execution times.
* **SecurityAspect:** Intercepts the addBook method in BookService to perform security checks before allowing the addition of a book.

The diagram shows the dependencies and interactions between these components, emphasizing the use of dependency injection.