**WEEK 3 – SPRING CORE AND MAVEN**

**EXERCISE 1: CONFIGURING A BASIC SPRING APPLICATION**

**STEP 1:** Download a SpringBoot Project from Spring Initializer

**STEP 2**: Import into Eclipse

**STEP 3:** Add Repository and Service Classes

LibraryManagementApplication.java

package com.library.LibraryManagement;

import com.library.LibraryManagement.service.BookService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class LibraryManagementApplication implements CommandLineRunner {

*@Autowired*

private BookService bookService;

public static void main(String[] args) {

SpringApplication.*run*(LibraryManagementApplication.class, args);

}

*@Override*

public void run(String... args) {

bookService.addBook("The Great Gatsby");

}

}

BookService.java:

package com.library.LibraryManagement.service;

import com.library.LibraryManagement.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

*@Service*

public class BookService {

*@Autowired*

private BookRepository bookRepository;

public void addBook(String bookName) {

System.***out***.println("Adding book: " + bookName);

bookRepository.saveBook(bookName);

}

}

BookRepository.java:

package com.library.LibraryManagement.repository;

import org.springframework.stereotype.Repository;

*@Repository*

public class BookRepository {

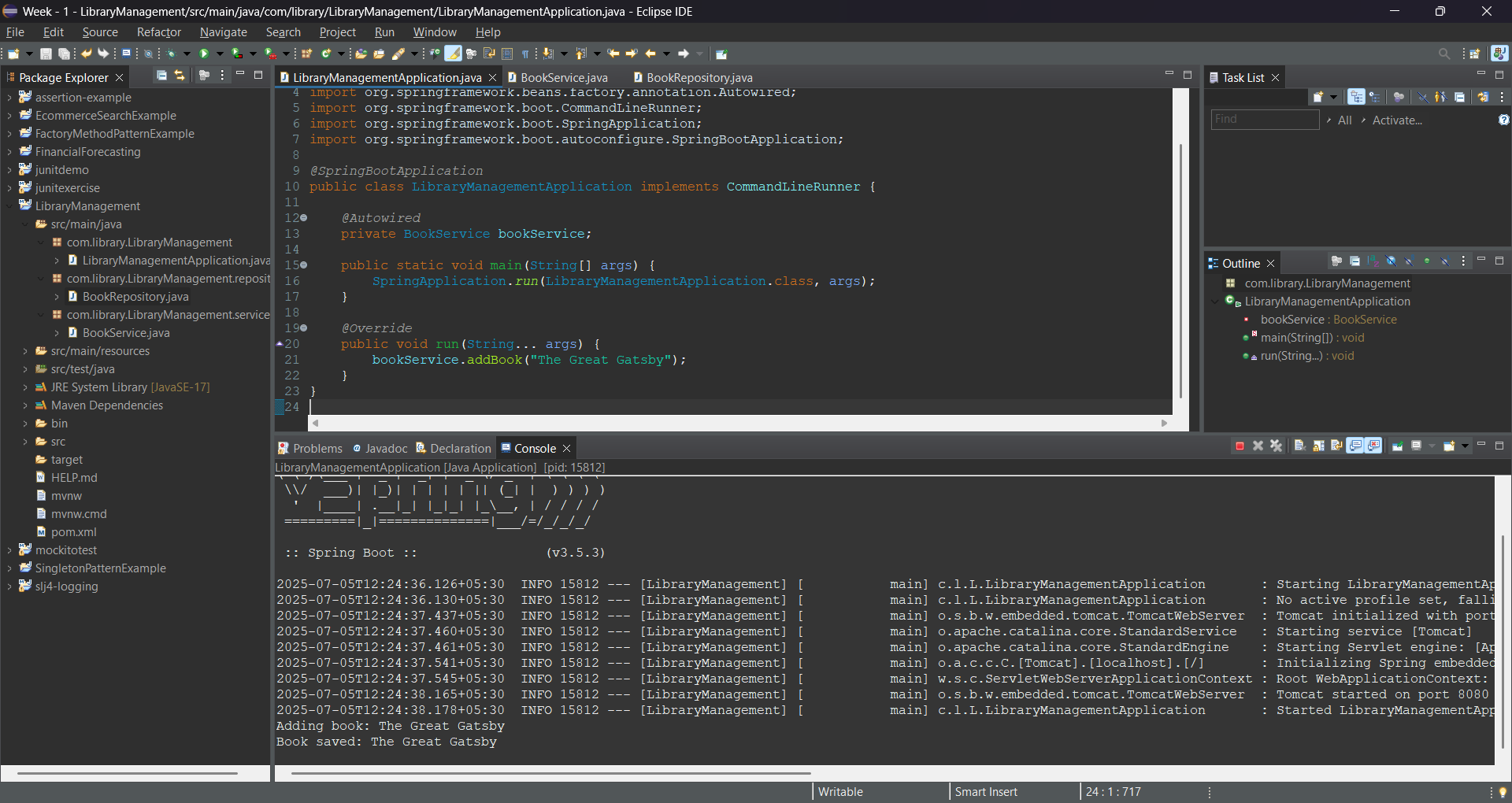
public void saveBook(String bookName) {

System.***out***.println("Book saved: " + bookName);

}

}

OUTPUT:



**EXERCISE 2 :** **IMPLEMENTING DEPENDENCY INJECTION**

**STEP 1:** Download a SpringBoot Project from Spring Initializer

**STEP 2**: Import into Eclipse

**STEP 3:** Add applcationContext.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- BookRepository Bean -->

<bean id="bookRepository" class="com.library.LibraryManagement.repository.BookRepository"/>

<!-- BookService Bean with DI -->

<bean id="bookService" class="com.library.LibraryManagement.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

LibraryManagementApplication.java

package com.library.LibraryManagement;

import com.library.LibraryManagement.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = (BookService) context.getBean("bookService");

bookService.listSampleBooks();

((ClassPathXmlApplicationContext) context).close();

}

}

BookService.java

package com.library.LibraryManagement.service;

import com.library.LibraryManagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.***out***.println("Request to add book: " + bookName);

bookRepository.saveBook(bookName);

}

public void listSampleBooks() {

addBook("The Alchemist");

addBook("Harry Potter and the Sorcerer's Stone");

addBook("Clean Code");

addBook("Wings of Fire");

addBook("You Can Win");

}

}

BookRepository.java

package com.library.LibraryManagement.repository;

import org.springframework.stereotype.Repository;

*@Repository*

public class BookRepository {

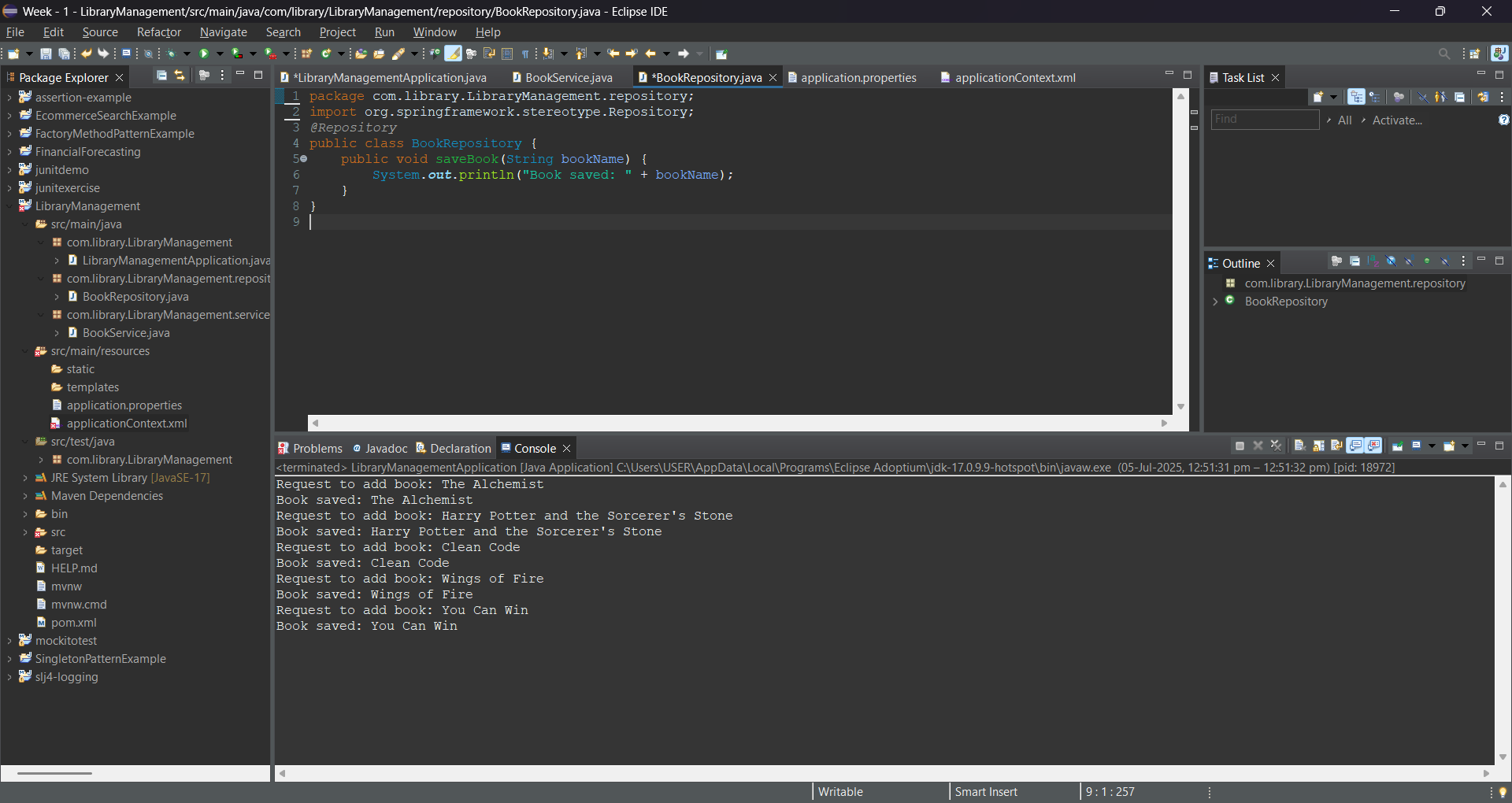
public void saveBook(String bookName) {

System.***out***.println("Book saved: " + bookName);

}

}

Output:



**EXERCISE 4: CREATING AND CONFIGURING A MAVEN PROJECT**

**STEP 1:** Download a SpringBoot Project from Spring Initializer

**STEP 2**: Import into Eclipse

**STEP 3:** Update pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<!-- Spring Core / Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Spring WebMVC -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Servlet API (needed for WebMVC, scope = provided) -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Compiler Plugin for Java 1.8 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**LibraryManagementApplication.java**

package com.library.LibraryManagement;

import com.library.LibraryManagement.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = (BookService) context.getBean("bookService");

bookService.listSampleBooks();

((ClassPathXmlApplicationContext) context).close();

}

}

**BookService.java**

package com.library.LibraryManagement.service;

import com.library.LibraryManagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.***out***.println("Request to add book: " + bookName);

bookRepository.saveBook(bookName);

}

public void listSampleBooks() {

addBook("The Alchemist");

addBook("Harry Potter and the Sorcerer's Stone");

addBook("Clean Code");

}

public void logActivity(String user) {

System.***out***.println("User '" + user + "' is accessing the book service.");

}

}

OUTPUT:

