

Case Study 3: Annotation-Based Configuration

//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.example</groupId>
    <artifactId>library-annotation-config</artifactId>
    <version>1.0-SNAPSHOT</version>
    <dependencies>
        <!-- Spring Core -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.33</version>
        </dependency>
    </dependencies>
</project>
```

//Book.java

```
package com.example.library;

import org.springframework.stereotype.Component;
import java.util.ArrayList;
import java.util.List;

@Component
public class Book {
    private List<String> books = new ArrayList<>();

    public void addBook(String title) {
        books.add(title);
    }
}
```

```

        System.out.println("Book added: " + title);
    }
}

public void searchBook(String keyword) {
    System.out.println("Searching books with keyword: " + keyword);
    for (String book : books) {
        if (book.toLowerCase().contains(keyword.toLowerCase())) {
            System.out.println("Found: " + book);
        }
    }
}
}

```

//Member.java

```

package com.example.library;

import org.springframework.stereotype.Component;
import java.util.ArrayList;
import java.util.List;

@Component
public class Member {
    private List<String> members = new ArrayList<>();

    public void registerMember(String name) {
        members.add(name);
        System.out.println("Member registered: " + name);
    }

    public void viewMembers() {
        System.out.println("Registered Members:");
        for (String member : members) {
            System.out.println("- " + member);
        }
    }
}

```

//Loan.java

```
package com.example.library;

import org.springframework.stereotype.Component;

@Component

public class Loan {

    public void issueBook(String memberName, String bookTitle) {

        System.out.println("Issued '" + bookTitle + "' to " + memberName);

    }

    public void returnBook(String memberName, String bookTitle) {

        System.out.println(memberName + " returned '" + bookTitle + "'");

    }

}
```

//LibraryService.java

```
package com.example.library;

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

@Component

public class LibraryService {

    @Autowired

    private Book book;

    @Autowired

    private Member member;

    @Autowired

    private Loan loan;

    public void performOperations() {

        member.registerMember("Lalitha");

        member.registerMember("Pavan");

        book.addBook("Java Basics");

        book.addBook("Spring Framework");

        book.searchBook("spring");

        member.viewMembers();

    }

}
```

```
        loan.issueBook("Lalitha", "Spring Framework");

        loan.returnBook("Lalitha", "Spring Framework");
    }
}

//MainApp.java

package com.example.library;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
@ComponentScan(basePackages = "com.example.library")

public class MainApp {

    public static void main(String[] args) {

        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(MainApp.class);

        LibraryService libraryService = context.getBean(LibraryService.class);

        libraryService.performOperations();

        context.close();
    }
}
```


