

Case Study
On
Used book selling website

INTRODUCTION

- The Used Book Selling Web Application is designed to provide an online platform for buying and selling second-hand books. The project aims to reduce the cost of educational and non-academic books by encouraging reuse. It allows sellers to list used books and buyers to browse available books easily. The application simplifies the process of book exchange by directly connecting buyers and sellers. It also helps reduce wastage and promotes sustainable learning practices.

ABSTRACT

- The Used Book Selling Web Application is a full-stack system developed using Java, Spring Boot, MySQL, HTML, CSS, and Thymeleaf.
The system supports CRUD operations for managing book listings. A secure login system allows only authenticated sellers to add or delete books.
Buyers can directly contact sellers using WhatsApp integration.
The application follows the MVC architecture to ensure scalability and maintainability.

What we are going to build: Client Requirement

- The client requires a system where sellers can sell used books online.
- Buyers should be able to view available books without logging in.
- Sellers must be able to register, login, and manage their book listings.
- Each book should display seller contact details.
- The system should support secure data storage and easy navigation.

What we are going to build: Some technical terms.

- Seller should be able to register and login
- Seller should be able to add, update, and delete books
- Buyer should be able to view all available books
- Buyer should be able to contact seller via WhatsApp
- System should display book status (Available / Sold)

What are the technologies and tools we are going to use ?

- Backend: Java, Spring Boot
- Frontend: HTML, CSS, Bootstrap, Thymeleaf
- Database: MySQL
- ORM: Spring Data JPA (Hibernate)
- IDE: Spring Tool Suite (STS)
- Web Server: Apache Tomcat
- Browser: Google Chrome

SYSTEM REQUIREMENTS

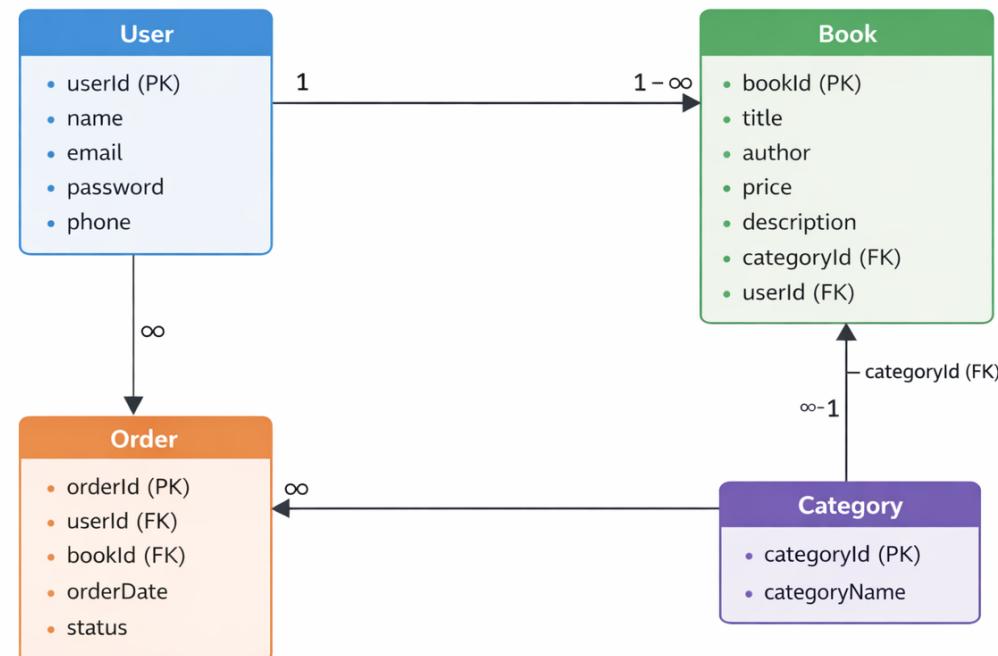
- **Software Requirements**
- • Operating System: Windows / Linux
 - IDE: Spring Tool Suite (STS)
 - Database Server: MySQL 8.0
 - Web Server: Apache Tomcat
 - Browser: Google Chrome

PROJECT MODULE

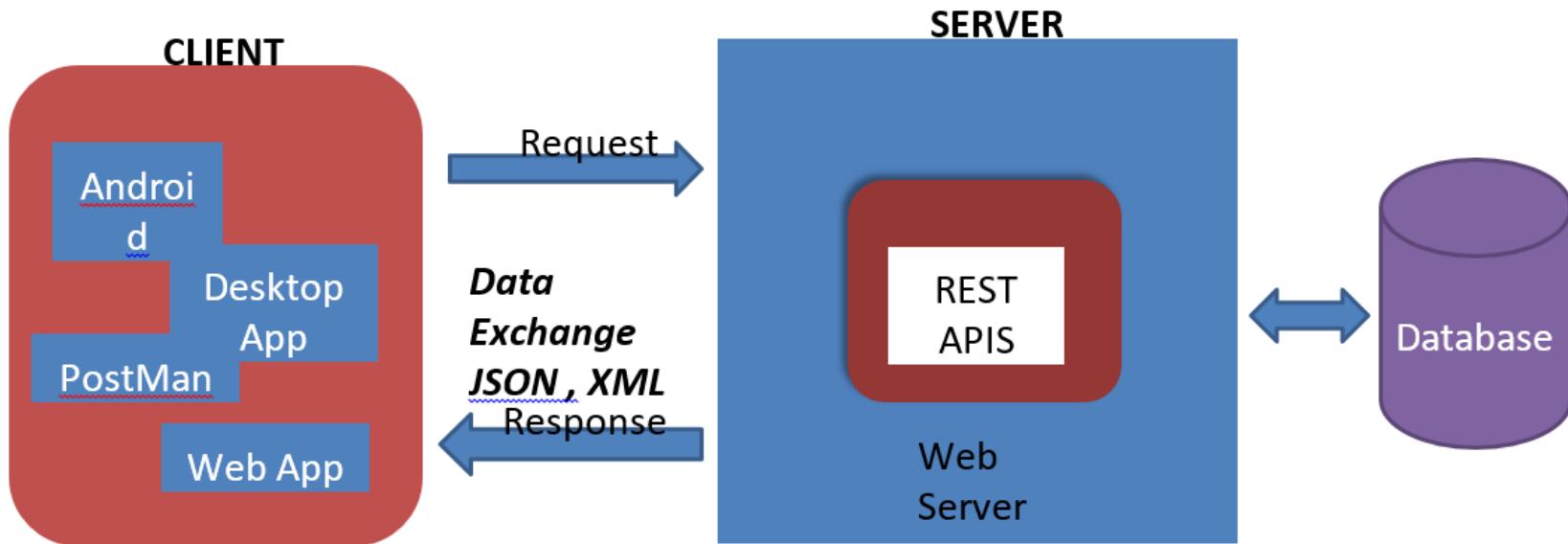
- Seller Module
 - Book Module
 - Authentication Module
 - Buyer Interaction Module

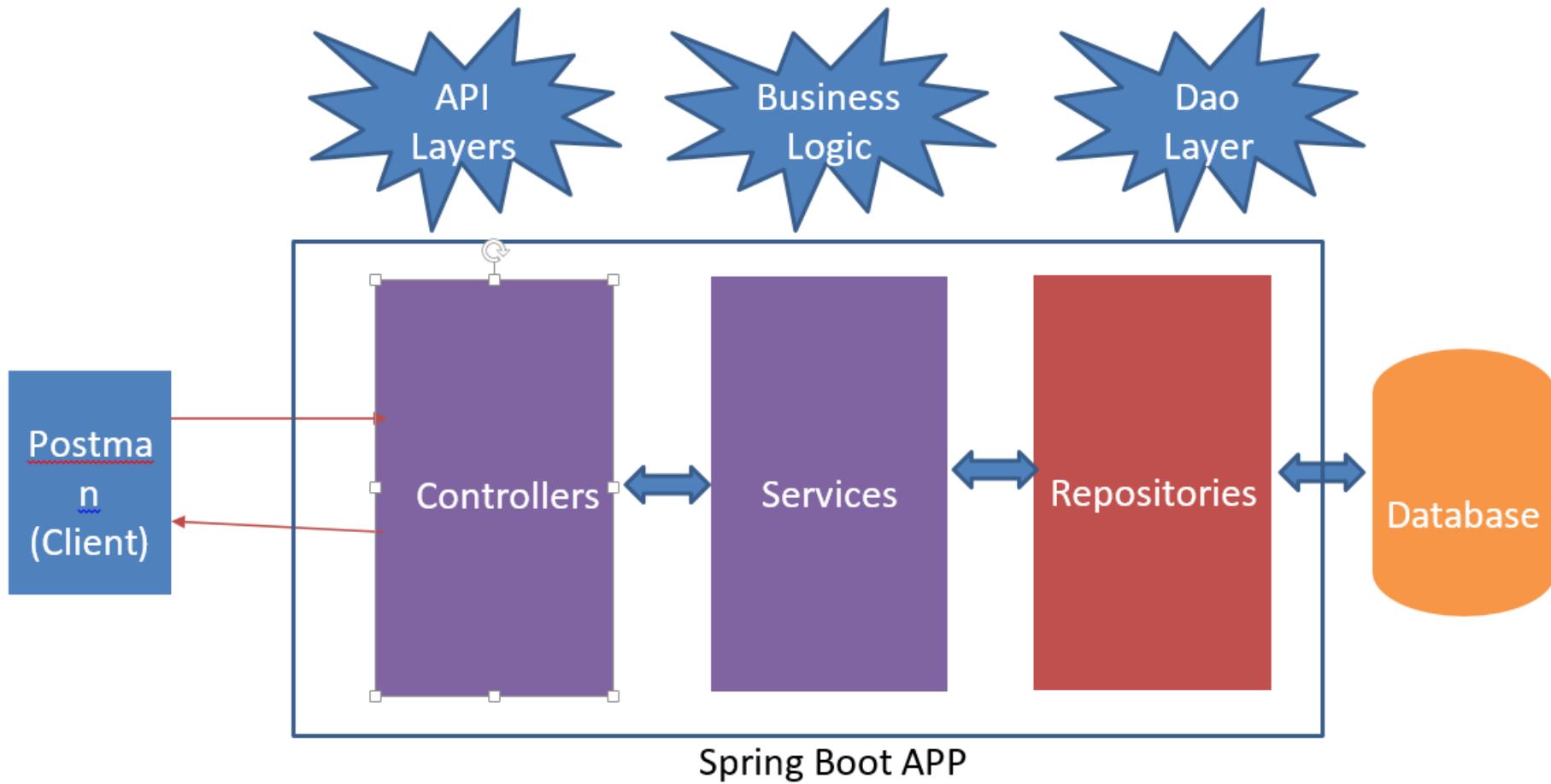
ER DIAGRAM

Used Book Selling App ER Diagram



Client – Server Architecture





SELLER MODULE

- The Seller Module allows users to register and login into the system.
- Only authenticated sellers can add or delete book listings.
- Seller details such as name and phone number are stored securely.
- Seller information is linked to the books they upload.

BOOK MODULE

- The Book Module manages all book-related operations.
- Sellers can add new books with title, author, price, and description.
- Books can be updated or deleted by the seller.
- All available books are displayed on the home page.

BUYER INTERACTION MODULE

- Buyers can view all available books without logging in.
- Each book listing displays seller contact details.
- Buyers can contact sellers directly using WhatsApp integration.
- This ensures fast and direct communication between users.

HTTP REQUEST METHODS USED

- GET – To retrieve book listings
- POST – To add new books
- PUT – To update existing book details
- DELETE – To remove books from the system

ARCHITECTURE USED

- The application follows Client–Server Architecture.
- The frontend sends requests to the backend server.
- The backend processes the request and interacts with the database.
- Data is exchanged in structured format between layers.
- The MVC pattern ensures separation of concerns.

DATA DICTIONARY

- TABLES OF DATABASE

The screenshot shows the MySQL Workbench interface. In the top-left corner, the connection status is "unconnected". The main window has a toolbar with various icons. On the left, there's a "Navigator" pane with sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, etc.), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports). The "Administration" tab is selected. Below the tabs, it says "No object selected". The central area contains a "Query 1" editor with the following SQL code:

```
CREATE DATABASE used_book_db;
show tables;
use used_book_db;
```

Below the code, a "Result Grid" shows the results of the "show tables" command:

Tables_in_used_book_db
book
sellers
user

On the right side of the interface, there's a "SQLAdditions" panel with a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." At the bottom, there's a "Result 1" tab showing the execution history:

Action Output	Time	Action	Message	Duration / Fetch
1	19:50:46	CREATE DATABASE used_book_db	Error Code: 1007. Can't create database 'used_book_db'; database exists	0.031 sec
2	21:34:12	show tables	Error Code: 1046. No database selected Select the default DB to be used ...	0.032 sec
3	21:34:35	use used_book_db	0 row(s) affected	0.000 sec
4	21:34:38	show tables	3 row(s) returned	0.000 sec / 0.000 sec

Book database

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
1 • CREATE DATABASE used_book_db;
2 • show tables;
3 • use used_book_db;
4 • select * from book;
```
- Result Grid:** Displays the results of the last query, showing data for books. The columns are: id, author, condition_desc, price, seller_name, seller_phone, status, title, and image_nar. The data is as follows:

#	id	author	condition_desc	price	seller_name	seller_phone	status	title	image_nar
▶	3	sarah j mass	new	344	Avanthica	7305372597	AVAILABLE	inheritance games	NULL
▶	4	winona ryder	good	3422	Avanthica	7305372597	AVAILABLE	acotar	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
- Action History:** Shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	19:50:46	CREATE DATABASE used_book_db	Error Code: 1007. Can't create database 'used_book_db'; database exists	0.031 sec
2	21:34:12	show tables	Error Code: 1046. No database selected Select the default DB to be used ...	0.032 sec
3	21:34:35	use used_book_db	0 row(s) affected	0.000 sec
4	21:34:38	show tables	3 row(s) returned	0.000 sec / 0.000 sec
5	21:39:57	select * from book LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Seller database

The screenshot shows the MySQL Workbench interface with the following details:

- File Edit View Query Database Server Tools Scripting Help**
- Navigator**: MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports), Administration, Schemas, Information.
- Query 1**:

```
1 • CREATE DATABASE used_book_db;
2 • show tables;
3 • use used_book_db;
4 • select * from sellers;
```
- Result Grid**:

	id	email	name	password	phone
▶	1	user	Avanthica	5df6a08-9add-411b-b221-3449af714b16	7305372597
*	NULL	NULL	NULL	NULL	NULL
- sellers 4**:

#	Time	Action	Message	Duration / Fetch
2	21:34:12	show tables	Error Code: 1046. No database selected Select the default DB to be used...	0.032 sec
3	21:34:35	use used_book_db	0 row(s) affected	0.000 sec
4	21:34:38	show tables	3 row(s) returned	0.000 sec / 0.000 sec
5	21:39:57	select * from book LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
6	21:41:11	show tables	3 row(s) returned	0.000 sec / 0.000 sec
7	21:41:24	select * from sellers LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
- SQLAdditions**: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
- Object Info Session**

User database

The screenshot shows the MySQL Workbench interface. In the top-left corner, there's a navigation pane with sections like MANAGEMENT, INSTANCE, and PERFORMANCE. The central part of the screen has a 'Query 1' tab open with the following SQL code:

```
1 • CREATE DATABASE used_book_db;
2 • show tables;
3 • use used_book_db;
4 • select * from user;
```

Below the code, there's a 'Result Grid' showing the following data:

	id	email	name	password	phone
*	HULL	HULL	HULL	HULL	HULL

On the right side of the interface, there's a message about context help being disabled:

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

At the bottom, there's an 'Output' window titled 'user 6' showing the execution history:

#	Time	Action	Message	Duration / Fetch
4	21:34:38	show tables	3 row(s) returned	0.000 sec / 0.000 sec
5	21:39:57	select * from book LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
6	21:41:11	show tables	3 row(s) returned	0.000 sec / 0.000 sec
7	21:41:24	select * from sellers LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	21:42:11	show tables	3 row(s) returned	0.000 sec / 0.000 sec

Main program

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure for "used-book-appused-book-app". It includes the `src/main/java` directory containing `UsedBookAppusedBookAppApplication.java`, and `src/main/resources` containing `application.properties` and several HTML files (add-book.html, book-list.html, index.html, login.html, new_book.html, register.html).
- Code Editor:** Displays the `UsedBookAppusedBookAppApplication.java` file, which contains the main application class definition.
- Console:** Shows the output of the application's execution. The log starts with the Spring Boot logo and proceeds with various INFO-level logs from different components like `UsedBookAppusedBookAppApplication`, `RepositoryConfigurationDelegate`, `TomcatWebServer`, and `Hibernate`. The log ends with a warning about failed connections.
- Outline:** Shows the class hierarchy and methods for `UsedBookAppusedBookAppApplication`.
- Task List:** Shows a single task related to the main method.
- Workspace Log:** Shows a list of log entries, mostly from `java.lang`.
- Message:** Shows a list of messages, mostly from `java.lang`.

```
1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class UsedBookAppusedBookAppApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(UsedBookAppusedBookAppApplication.class, args);
10    }
11 }
12
13 }
```

```
2025-12-18T20:58:31.047+05:30 INFO 25332 --- [           main] c.e.d.UsedBookAppusedBookAppApplication : Starting UsedBookAppusedBookAppApplication
2025-12-18T20:58:31.059+05:30 INFO 25332 --- [           main] c.e.d.UsedBookAppusedBookAppApplication : No active profile present, falling back to default: "default"
2025-12-18T20:58:32.535+05:30 INFO 25332 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories...
2025-12-18T20:58:32.622+05:30 INFO 25332 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning
2025-12-18T20:58:33.616+05:30 INFO 25332 --- [           main] o.s.boot.tomcat.TomcatWebServer        : Tomcat initialized with port(s): 8080 (http)
2025-12-18T20:58:33.652+05:30 INFO 25332 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-12-18T20:58:33.654+05:30 INFO 25332 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet Engine: [Apache Tomcat/9.0.64]
2025-12-18T20:58:34.160+05:30 INFO 25332 --- [           main] b.w.c.s.WebApplicationContextInitializer : Root WebApplicationContext is starting
2025-12-18T20:58:34.620+05:30 INFO 25332 --- [           main] o.h.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo
2025-12-18T20:58:34.997+05:30 INFO 25332 --- [           main] o.h.hibernate.Version                    : HHH000412: Hibernate Core {4.3.11.Final}
2025-12-18T20:58:36.482+05:30 INFO 25332 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver configured - using default
2025-12-18T20:58:36.595+05:30 INFO 25332 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1
```

Video link

- <https://drive.google.com/file/d/1k47wtXYKe4fqcQSa19U9toxMsTGrTRS/view?usp=sharing>