# Spring Data Exam – 13 August 2023

# Library Management System

*Dive into the fascinating realm of books, readers, and the organized world of libraries that have shaped our understanding of literature and information. Expand your knowledge of library resources, discover hidden connections between various genres, and embark on a journey through the vast sea of books and records. Embrace the joy of reading with just a few clicks. The "Library Management System" empowers you to delve into the world of literature, connect with fellow book enthusiasts, and indulge in the pleasure of exploring new literary horizons. Join us on this exciting journey through the pages of knowledge, and let the Library Management System enrich your reading adventures like never before.*

## Functionality Overview

The application should be able to easily **import** hard-formatted data and **support functionalities** for also **exporting** the imported data. The application is called – **Library Management System**.

Look at the pictures below to see what must happen:

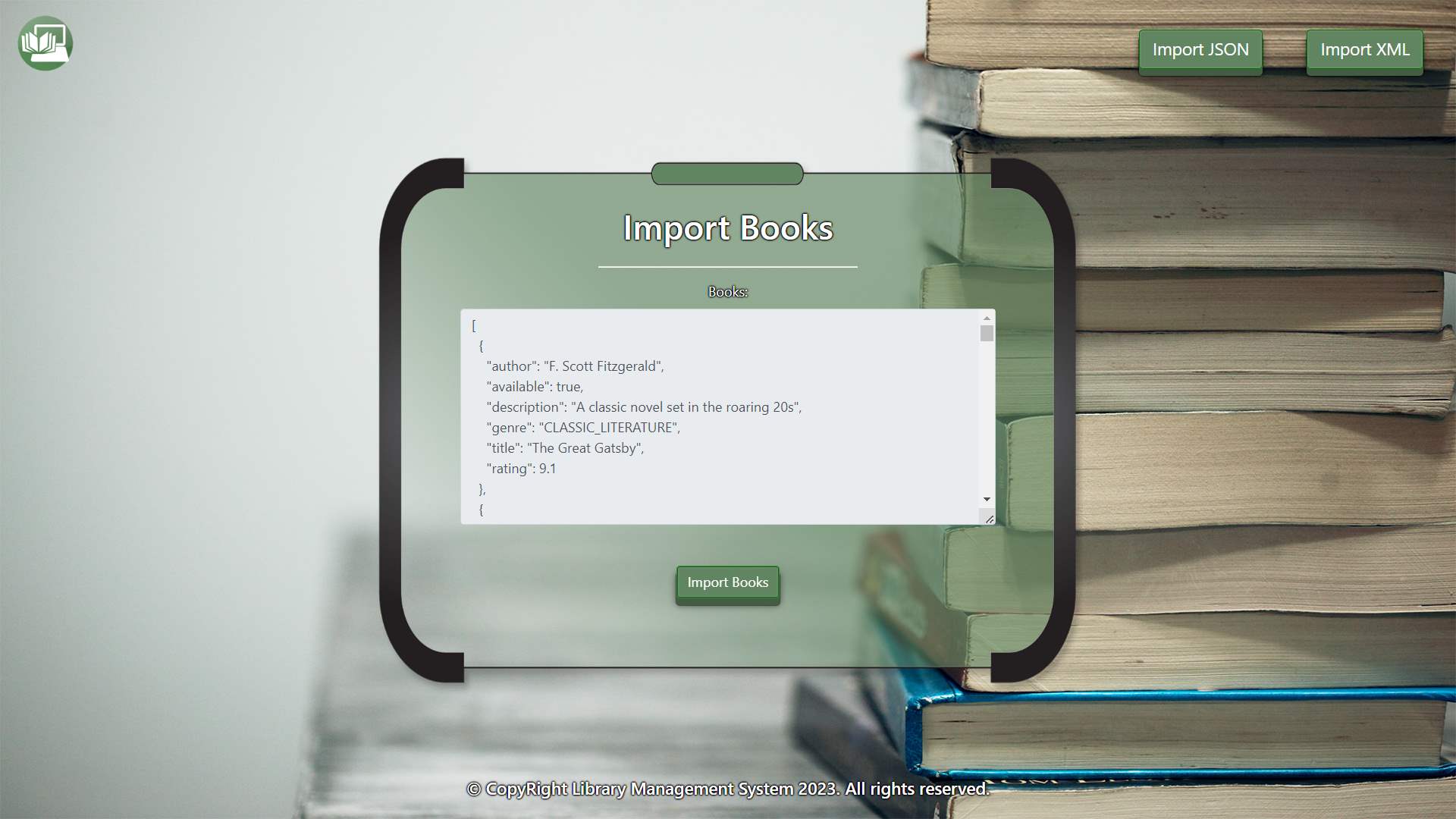
* The home page before importing anything:



* The import JSON page before importing anything:



* Import the books first:



* Import the library members second:



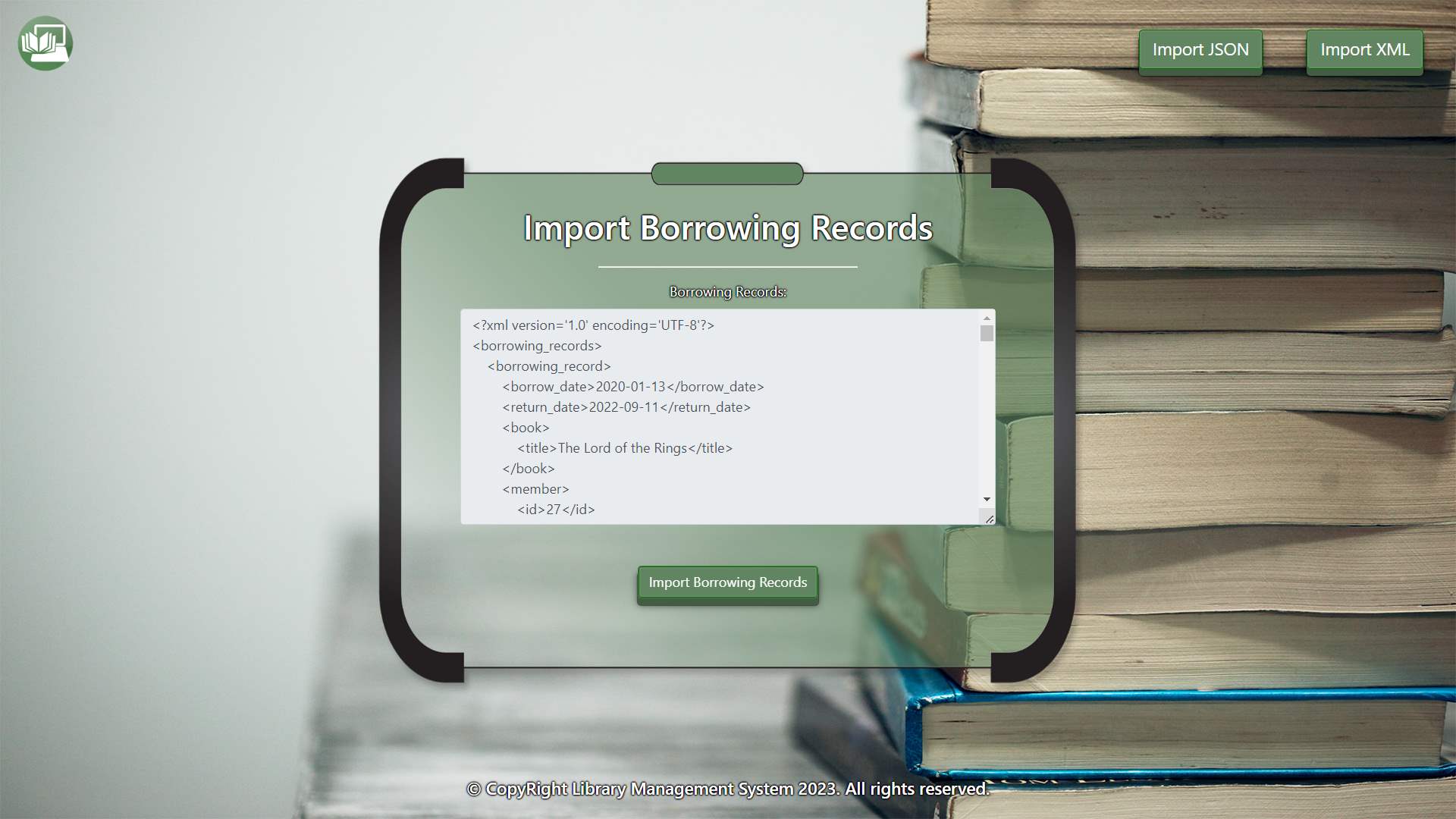
* The import JSON page after importing both files:



* The import XML page before importing the given data:



* Import the borrowing records data:



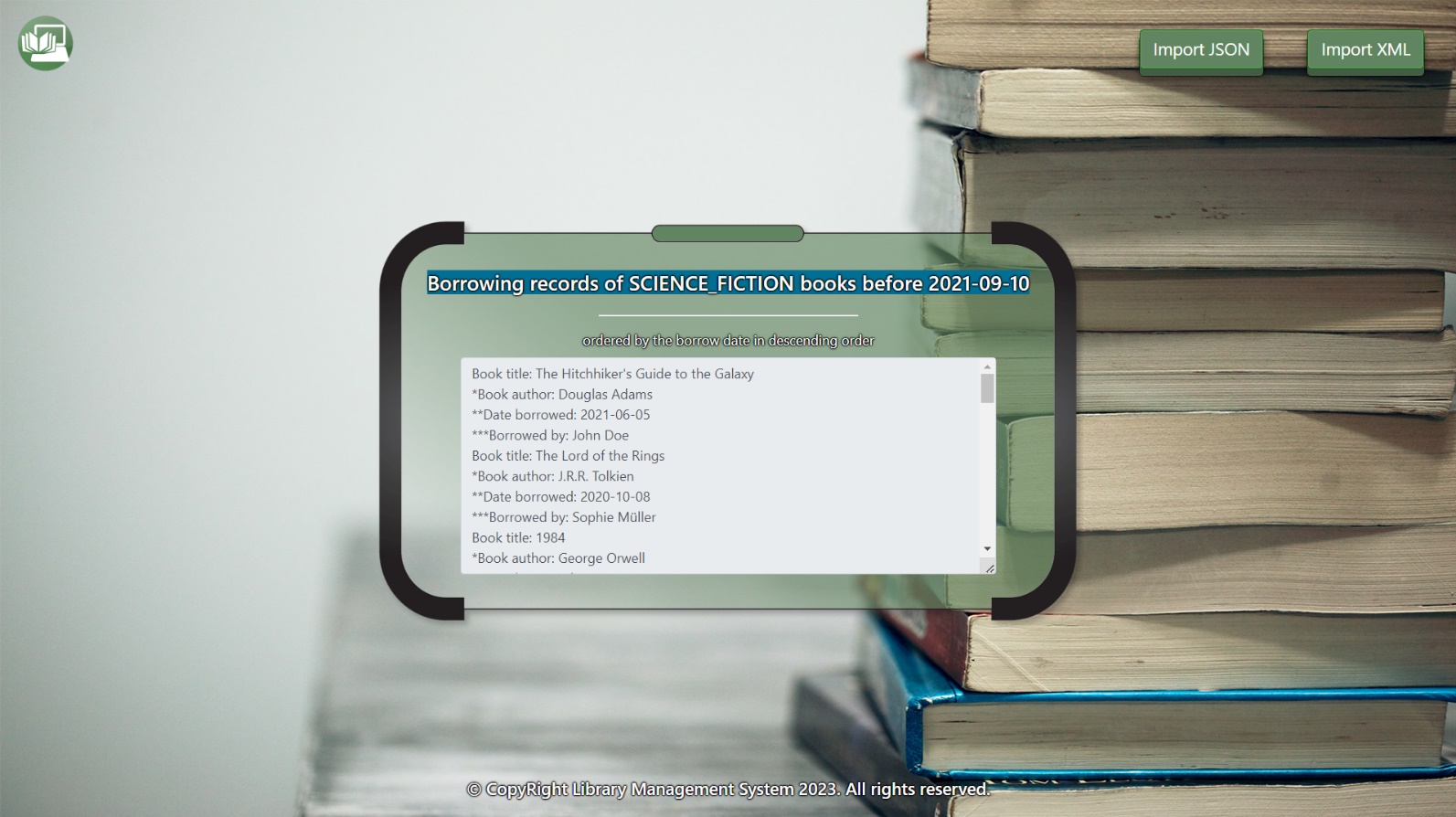
* The import XML page after importing the data:



* The home page after the data is imported:



* Export the borrowing records before 2021-09-10:



## Project Skeleton Overview

You will be given a **skeleton**, containing a **certain architecture (MVC)** with **several classes**, some of which are completely empty. The **Skeleton** will include the **files** with which you will **seed** the **database**.

**Don’t change the skeleton or pom.xml files.**

When submitting your project in Judge please make a **.zip** file only with **src** folder and **pom.xml** file.  
There are four problems in **Judge**:

**1. Database** will test the structure of your database.

**2. Service** will test the two methods **areImported()** and **readClassFromFile()** in every service class.  
**3. Import JSON** will test the logic of importing **JSON** files.

**4. Import XML** will test the logic of importing **XML** files.

**5. Export** will test the logic of the final **export** button which is in the **ExportController**.

Please note that the automated system Judge may take up to **5 minutes** to show results for each problem. We kindly ask you to be patient.

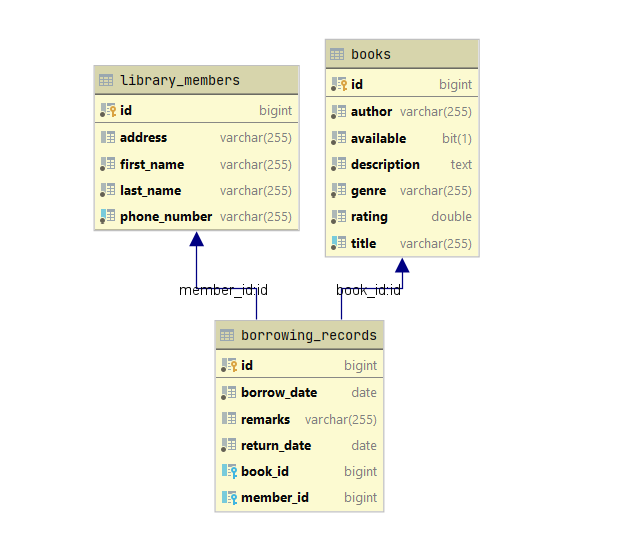
Be aware that Jude might **not** recognize the **var** keyword, leading to compilation errors.

To help you ensure the correctness of your solutions and **gain immediate feedback**, we have included a set of **commented-out tests** within the provided skeleton project. These tests are designed to verify the functionality of the tasks you are required to implement.

## Model Definition

There are 3 main models that the **Library Management System database** application should contain in its functionality.

We have provided you with an Entity-Relationship (ER) diagram that represents the database schema. The ER diagram displays the tables and their relationships, along with the field types for each table.



Name the entities and their class members **exactly** in the **format stated** above.

All fields are **NOT NULL** unless explicitly stated to be nullable.

Ensure that the **Java** code **validates** data against the specified constraints before persisting it into the database. Handle any validation errors gracefully and provide meaningful feedback to the users.

Design them in the **most appropriate** way, considering the following **data constraints**:

### Book

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* title – accepts **char sequence** (between **3** to **40** inclusive). The values are **unique in the database**.
* author - accepts **char sequence** (between **3** to **40** inclusive).
* **description** - a long and detailed description about the book with a character length value higher than or equal to **5**.
* **available** – accepts a true or false, representing the availability status of the book.
* **genre** – String enumeration, one of the following – **CLASSIC\_LITERATURE, SCIENCE\_FICTION, FANTASY**
* **rating** – accepts number values that are positive.

### Library Member

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* first name - accepts **char** **sequence** (between **2** to **30** inclusive).
* **last name** -accepts **char** **sequence** (between **2** to **30** inclusive).
* address - accepts **char** **sequence** (between **2** to **40** inclusive). Can be nullable.
* **phone number** - accepts **char** **sequence** (between **2** to **20** inclusive). The values are **unique in the database**.

### Borrowing Record

* id - accepts **integer** values, a **primary identification field, an auto incremented field**.
* borrow date - a date in the "**yyyy-MM-dd**" format.
* return date - a date in the "**yyyy-MM-dd**" format.
* remarks - can be used to store any relevant information and might be helpful for tracking and managing the borrowing records. Accepts char sequence (between 3 to 100 inclusive). Can be nullable.
* Constraint: The borrowing\_records table has a relation with books table.
* Constraint: The borrowing\_records table has a relation with library\_members table.

#### Relationships

Your partners gave you a little hint about the more complex relationships in the database, so that you can implement it correctly.

One **Borrowing Record** may have only one **Book**, but one **Book** may be in many **Borrowing Records**.

One **Borrowing Record** may have only one **Library Member**, but one **Library Member** can be in many **Borrowing Records**.

## Data Import

Use the provided files to populate the database with data. Import all the information from those files into the database.

**You are not allowed to modify the provided files.**

**ANY INCORRECT** data should be **ignored** and a message:

"**Invalid {book / library member/ borrowing record}** should be printed.

**When the import is finished:**

"**Successfully imported {book / library member/ borrowing record} {author – title/ first name – last name/ title – borrow date}**"

**Judge** will only accept **file paths** in a specific format. When dealing with file paths for files, please adhere to the following format: "src/main/resources/files/xml/format-example.xml"

### JSON Import

Your new colleagues have prepared some JSON data for you to import.

**Books (books.json)**

**Constraint:**

* **If a book with the same title already exists in the DB return "Invalid book".**

|  |
| --- |
| **Books (books.json)** |
| [  {  "author": "F. Scott Fitzgerald",  "available": true,  "description": "A classic novel set in the roaring 20s",  "genre": "CLASSIC\_LITERATURE",  "title": "The Great Gatsby",  "rating": 9.1  },  {  "author": "F. Scott Fitzgerald",  "available": true,  "description": "Romantic bolevard literature",  "genre": "CLASSIC\_LITERATURE",  "title": "Fo",  "rating": 9.1  },  {  "author": "Harper Lee",  "available": true,  "description": "A powerful story addressing racial injustice",  "genre": "CLASSIC\_LITERATURE",  "title": "To Kill a Mockingbird",  "rating": 5.6  },  {  "author": "Toni Morrison",  "available": true,  "description": "Tells the story of a dysfunctional family of formerly enslaved people whose Cincinnati home is haunted by a malevolent spirit.",  "genre": "CLASSIC\_LITERATURE",  "title": "Beloved",  "rating": -3.1  },  {  "author": "Douglas Adams",  "available": true,  "description": "The Hitchhiker's Guide to the Galaxy is a comedy science fiction franchise created by Douglas Adams.",  "genre": "SCIENCE\_FICTION",  "title": "The Hitchhiker's Guide to the Galaxy",  "rating": 4.2  },  {  "author": "Don Williams",  "available": true,  "description": "Some kind of copy of a famous book",  "genre": "SCIENCE\_FICTION",  "title": "The Hitchhiker's Guide to the Galaxy",  "rating": 3.2  },  ... |
| Successfully imported book F. Scott Fitzgerald - The Great Gatsby  Invalid book  Successfully imported book Harper Lee - To Kill a Mockingbird  Invalid book  Successfully imported book Douglas Adams - The Hitchhiker's Guide to the Galaxy  Invalid book  … |

Please be aware that due to variations in local settings on different computers, the representation of decimal numbers (Double) may differ. In some regions, the decimal separator is a comma (,), while in others, it is a dot (.).

Judge local settings represent all decimal numbers using a dot (.) as the decimal separator. *Locale.US*

**Library Members (library-members.json)**

**Constraint:**

* **If a library member with the same phone number already exists in the DB return "Invalid library member".**

|  |
| --- |
| **Library Members (library-members.json)** |
| [  {  "address": "123 Main St",  "firstName": "John",  "lastName": "Doe",  "phoneNumber": "555-1111"  },  {  "address": "13 Main St",  "firstName": "Joah",  "lastName": "Doevill",  "phoneNumber": "555-1111"  },  {  "address": "23 Einsteinufer str",  "firstName": null,  "lastName": "Schmidt",  "phoneNumber": "525-22"  },  {  "address": "456 Elm St",  "firstName": "Jane",  "lastName": "Smith",  "phoneNumber": "555-2222"  },  {  "address": null,  "firstName": "Michael",  "lastName": "Johnson",  "phoneNumber": "555-3333"  },  {  "address": "321 Maple Rd",  "firstName": "Emily",  "lastName": "Williams",  "phoneNumber": "5555-4444-2213-594810"  },  ... |
| Successfully imported library member John - Doe  Invalid library member  Invalid library member  Successfully imported library member Jane - Smith  Successfully imported library member Michael - Johnson  Invalid library member  … |

### XML Import

Your new colleagues have prepared some XML data for you to import.

**Borrowing Records (borrowing-records.xml)**

**Constraint:**

* **If a book with the given title doesn't exist in the DB return "Invalid borrowing record".**
* **If a library member with the given id doesn't exist in the DB return "Invalid borrowing record".**

|  |
| --- |
| **Borrowing Records (borrowing-records.xml)** |
| *<?*xml version='1.0' encoding='UTF-8'*?>* <borrowing\_records>  <borrowing\_record>  <borrow\_date>2020-01-13</borrow\_date>  <return\_date>2022-09-11</return\_date>  <book>  <title>The Lord of the Rings</title>  </book>  <member>  <id>27</id>  </member>  <remarks>Handle with care, fragile book.</remarks>  </borrowing\_record>  <borrowing\_record>  <borrow\_date>2020-01-13</borrow\_date>  <return\_date>2022-09-11</return\_date>  <book>  <title>The Lord of the RingsX</title>  </book>  <member>  <id>27</id>  </member>  <remarks>Extended borrowing period approved by librarian.</remarks>  </borrowing\_record>  <borrowing\_record>  <borrow\_date>2020-01-13</borrow\_date>  <return\_date>2022-09-11</return\_date>  <book>  <title>The Lord of the Rings</title>  </book>  <member>  <id>272</id>  </member>  <remarks>Member provided student ID for verification.</remarks>  </borrowing\_record>  <borrowing\_record>  <borrow\_date>2020-06-01</borrow\_date>  <return\_date>2023-01-07</return\_date>  <book>  <title>The Great Gatsby</title>  </book>  <member>  <id>14</id>  </member>  <remarks>Returned with minor damage, to be repaired.</remarks>  </borrowing\_record>  <borrowing\_record>  <borrow\_date>2020-03-09</borrow\_date>  <return\_date>2023-06-18</return\_date>  <book>  <title>Pride and Prejudice</title>  </book>  <member>  <id>29</id>  </member>  <remarks>Special discount applied for senior citizen member.</remarks>  </borrowing\_record>  <borrowing\_record>  <borrow\_date>2019-09-29</borrow\_date>  <return\_date>2023-04-10</return\_date>  <book>  <title>The Hobbit</title>  </book>  <member>  <id>25</id>  </member>  <remarks>Book in great condition, due back 14 days from borrow date. Contact if extensions needed. Be careful.</remarks>  </borrowing\_record>  ... |
| Successfully imported borrowing record The Lord of the Rings - 2020-01-13  Invalid borrowing record  Invalid borrowing record  Successfully imported borrowing record The Great Gatsby - 2020-06-01  Invalid borrowing record  Invalid borrowing record  … |

## Data Export

Get ready to export the data you have imported in the previous task. Here you will have some complex database querying. Export the data in the formats specified below.

#### Export the Borrowing records before 2021-09-10 from the Database

* Extract from the database the **book title, book author, date borrowed and the full name(first name and last name) of the library member**.
* **Filter only books which are SCIENCE\_FICTION and order them by the borrow date in descending order.**
* Return the information in this format:

**"Book title: {bookTitle}**

**"\*Book author: {bookAuthor}**

**"\*\*Date borrowed: {dateBorrowed}**

**"\*\*\*Borrowed by: {firstName} {lastName}**

**. . ."**

