Task 5: Design an Object-Oriented Programming Based Solution

Problem Description: Design a library management system that allows users to borrow and return books, and allows librarians to manage the book inventory.

Solution Design

Objects and Their Responsibilities

1. **Book:**
   - **Attributes:**
     - `book_id`: Unique identifier for the book
     - `title`: Title of the book
     - `author`: Author of the book
     - `genre`: Genre of the book
     - `available`: Boolean indicating if the book is available for borrowing
   - **Methods:**
     - `__init__(self, book_id, title, author, genre)`: Constructor to initialize a book object
     - `__str__(self)`: Method to return a string representation of the book
2. **User:**
   - **Attributes:**
     - `user_id`: Unique identifier for the user
     - `name`: Name of the user
     - `borrowed_books`: List of books borrowed by the user
   - **Methods:**
     - `__init__(self, user_id, name)`: Constructor to initialize a user object
     - `borrow_book(self, book)`: Method to borrow a book
     - `return_book(self, book)`: Method to return a borrowed book
3. **Librarian:**
   - **Attributes:**
     - `librarian_id`: Unique identifier for the librarian
     - `name`: Name of the librarian
   - **Methods:**
     - `__init__(self, librarian_id, name)`: Constructor to initialize a librarian object
     - `add_book(self, book)`: Method to add a book to the library inventory
     - `remove_book(self, book)`: Method to remove a book from the library inventory
4. **Library:**
   - **Attributes:**
     - `books`: List of all books in the library
     - `users`: List of all users registered with the library
   - **Methods:**
     - `__init__(self)`: Constructor to initialize the library object

        - `register_user(self, user)`: Method to register a new user
        - `add_book(self, book)`: Method to add a book to the library
        - `find_book(self, book_id)`: Method to find a book by its ID

Data File Structures (all files supplied)

1. **Books Data File (books.json):**
2. **Users Data File (users.json):**
3. **Librarians Data File (librarians.json):**

Python Implementation:

See Python code files: library.py
See JSON Files: books,json, librarians.json

Conclusion

This object-oriented design for a library management system clearly defines each class's responsibilities and interactions. The use of JSON files for data storage ensures that the system can persist information across sessions. This design allows the system to manage book inventories and user interactions efficiently.

Prepared by:
John Smith
IT Project Manager
john.smith@example.com

Notes:

- Ensure proper error handling and data validation in a production environment.
- Regularly update documentation and training materials to reflect any changes in the system.

This model provides a comprehensive and structured approach to designing an object-oriented programming solution, ensuring all aspects of the problem are addressed effectively