## Description of Object Oriented Design:

Sure, here's the description of each class in the library management system code along with their attributes, methods, and relationships:

1. LibraryItem Class:
   - Attributes:
     - title: Stores the title of the library item.
   - Methods:
     - __init__(self, title): Initializes a new library item with its title.

2. Book Class (Inherits from LibraryItem):
   - Attributes:
     - id: A unique identifier for the book.
     - author: The author of the book.
     - available: A boolean indicating whether the book is currently available for borrowing.
   - Methods:
     - __init__(self, id, title, author): Initializes the book with a unique id, title, author, and sets availability to True.

3. Member Class:
   - Attributes:
     - member_id: A unique identifier for the library member.
     - name: The name of the library member.
     - borrowed_books: A list to keep track of the books borrowed by the member.
   - Methods:
     - __init__(self, member_id, name): Initializes the member with a member_id, name, and an empty list for borrowed_books.

4. Library Class:
   - Attributes:

  - books: A list to store all books in the library.
  - members: A list to store all members of the library.
  - next_book_id: An integer to keep track of the next unique book ID for new books.

- Methods:
  - add_book(self, title, author): Adds a new book to the library with the given title and author.
  - display_books(self): Displays the entire book collection along with their availability.
  - search_book_by_name(self, name): Searches for books by name and displays matching results.
  - display_members(self): Displays the list of current members in the library.
  - add_member(self, name): Adds a new member to the library with the given name.
  - search_member_by_id(self, member_id): Searches for a member by their ID and returns the member object if found.
  - borrow_book(self, member_id, book_id): Allows a member to borrow a book from the library.
  - return_book(self, member_id, book_id): Allows a member to return a borrowed book to the library.
  - find_book_by_id(self, book_id): Helper method to find a book by its ID.

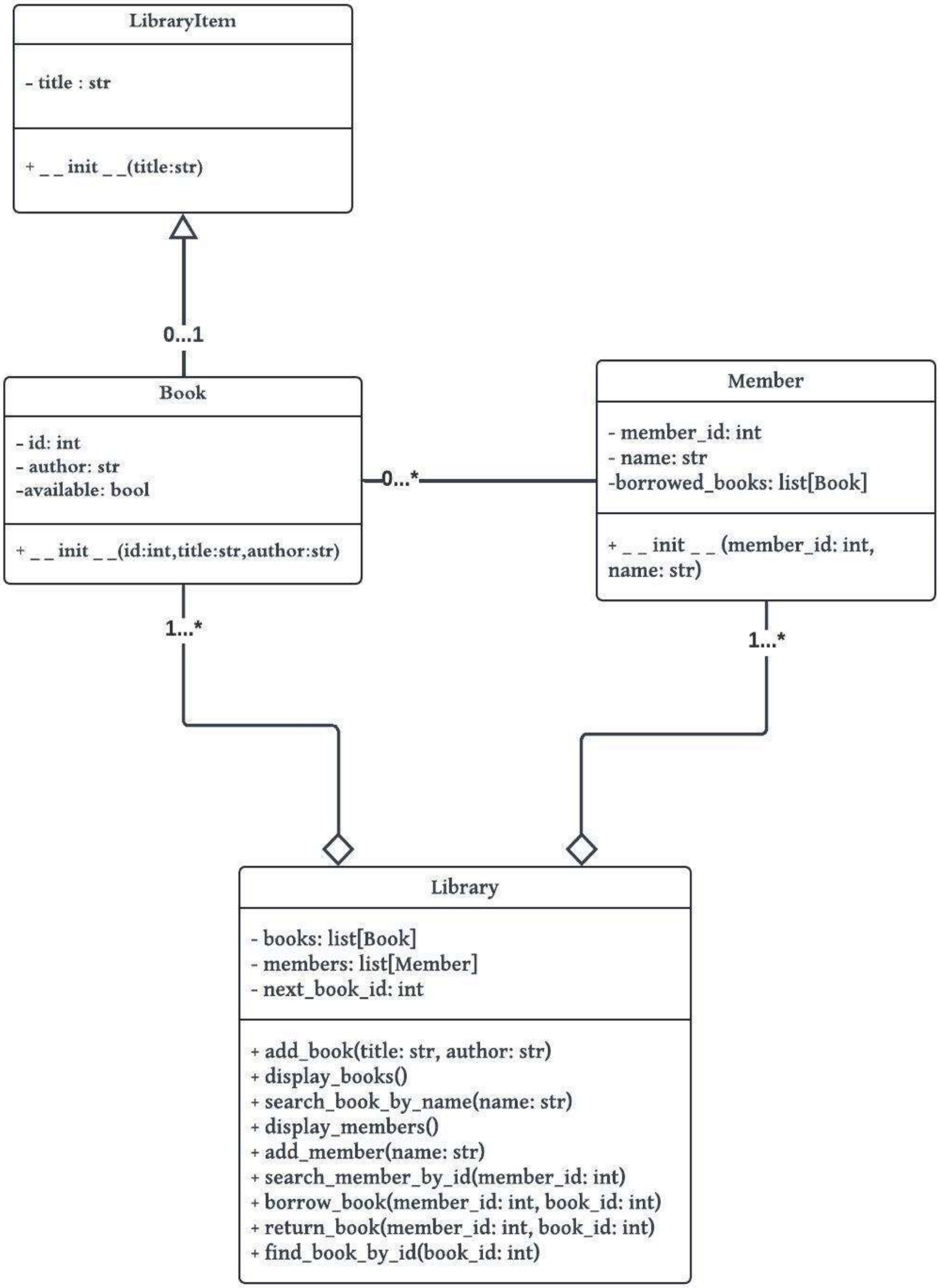
Relationships:

- The Book class inherits from the LibraryItem class, indicating an "is-a" relationship, where a book "is-a" library item.

- The Library class has compositions of Book and Member classes, meaning it contains instances of these classes as part of its own instance. This represents a "has-a" relationship, where a library "has" books and members.

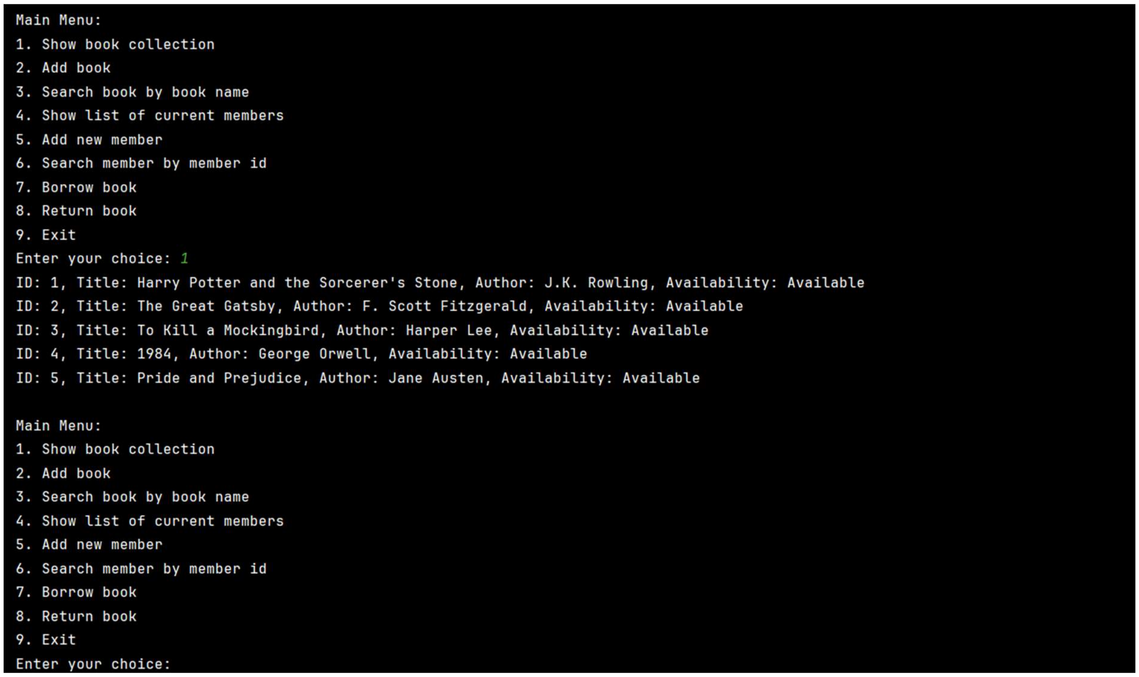
- The Member class has an association with the Book class through the borrowed_books attribute, indicating that a member can borrow multiple books.

UML Class Diagram:


LibraryItem
- title : str
+ _ _ init _ _(title:str)
0...1
Book
- id: int
- author: str
-available: bool
+ _ _ init _ _(id:int,title:str,author:str)
0...*
Member
- member_id: int
- name: str
-borrowed_books: list[Book]
+ _ _ init _ _ (member_id: int, name: str)
1...*
1...*
Library
- books: list[Book]
- members: list[Member]
- next_book_id: int
+ add_book(title: str, author: str)
+ display_books()
+ search_book_by_name(name: str)
+ display_members()
+ add_member(name: str)
+ search_member_by_id(member_id: int)
+ borrow_book(member_id: int, book_id: int)
+ return_book(member_id: int, book_id: int)
+ find_book_by_id(book_id: int)

## Screenshots of Running Results:

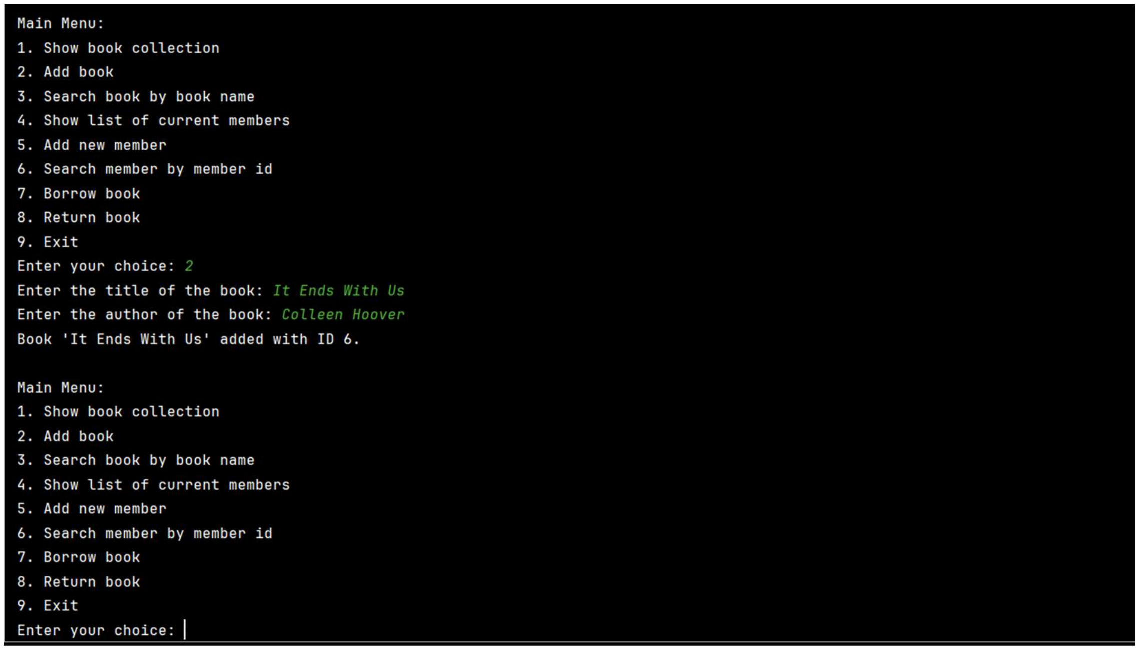
### Show book collection:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 1
ID: 1, Title: Harry Potter and the Sorcerer's Stone, Author: J.K. Rowling, Availability: Available
ID: 2, Title: The Great Gatsby, Author: F. Scott Fitzgerald, Availability: Available
ID: 3, Title: To Kill a Mockingbird, Author: Harper Lee, Availability: Available
ID: 4, Title: 1984, Author: George Orwell, Availability: Available
ID: 5, Title: Pride and Prejudice, Author: Jane Austen, Availability: Available

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

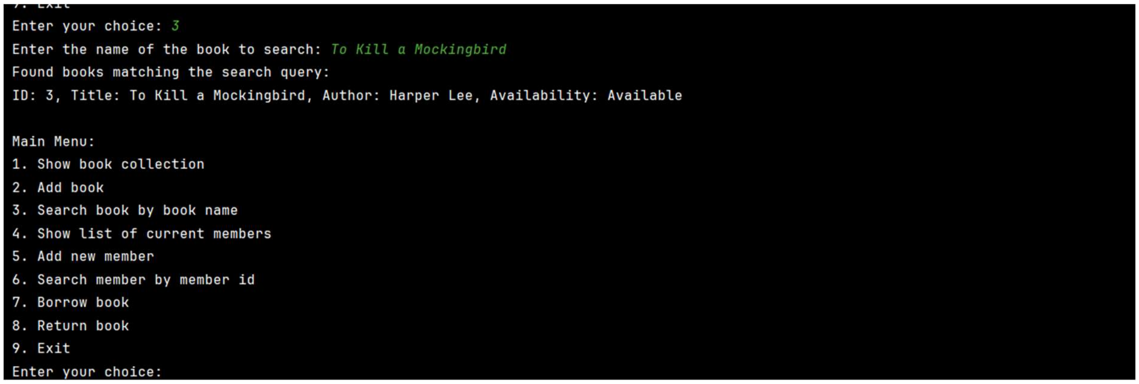
### Add book:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 2
Enter the title of the book: It Ends With Us
Enter the author of the book: Colleen Hoover
Book 'It Ends With Us' added with ID 6.

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

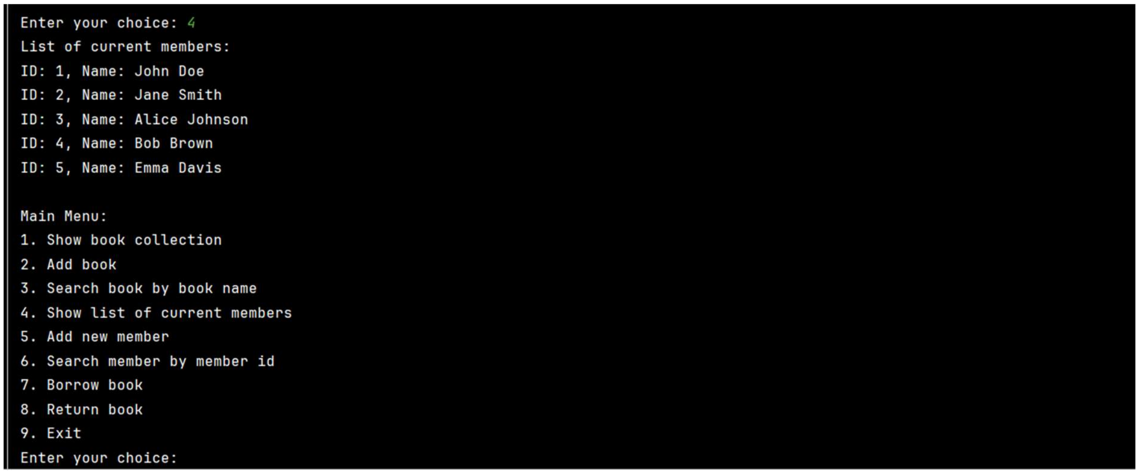
Search book by book name:



```
9. Exit
Enter your choice: 3
Enter the name of the book to search: To Kill a Mockingbird
Found books matching the search query:
ID: 3, Title: To Kill a Mockingbird, Author: Harper Lee, Availability: Available

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

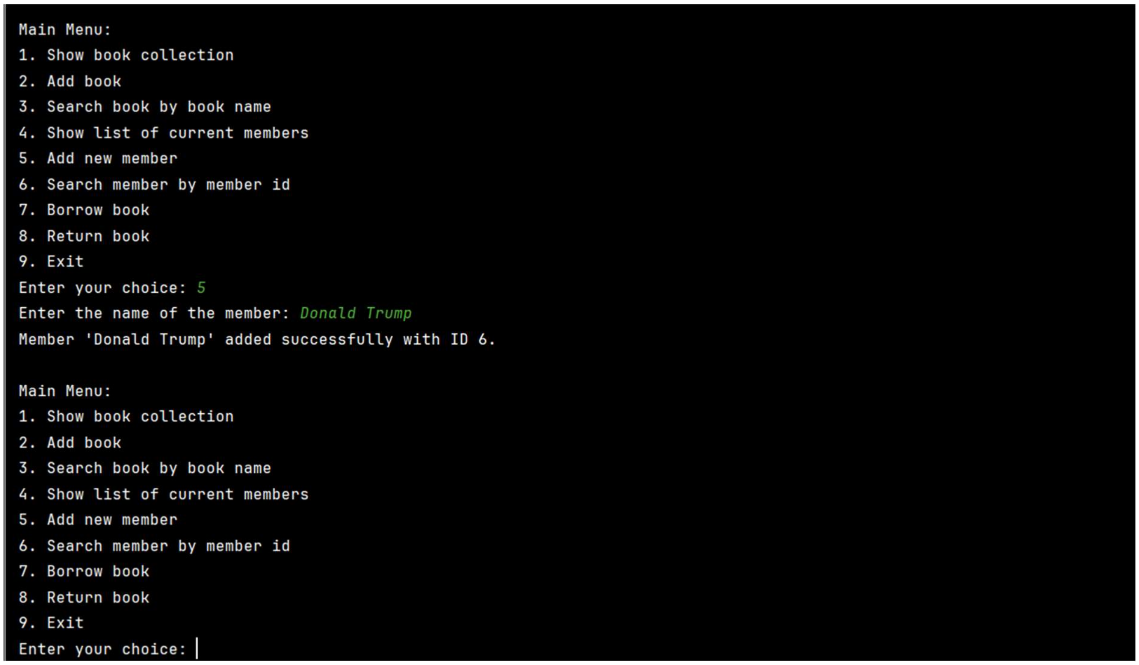
Show list of current members:



```
Enter your choice: 4
List of current members:
ID: 1, Name: John Doe
ID: 2, Name: Jane Smith
ID: 3, Name: Alice Johnson
ID: 4, Name: Bob Brown
ID: 5, Name: Emma Davis

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

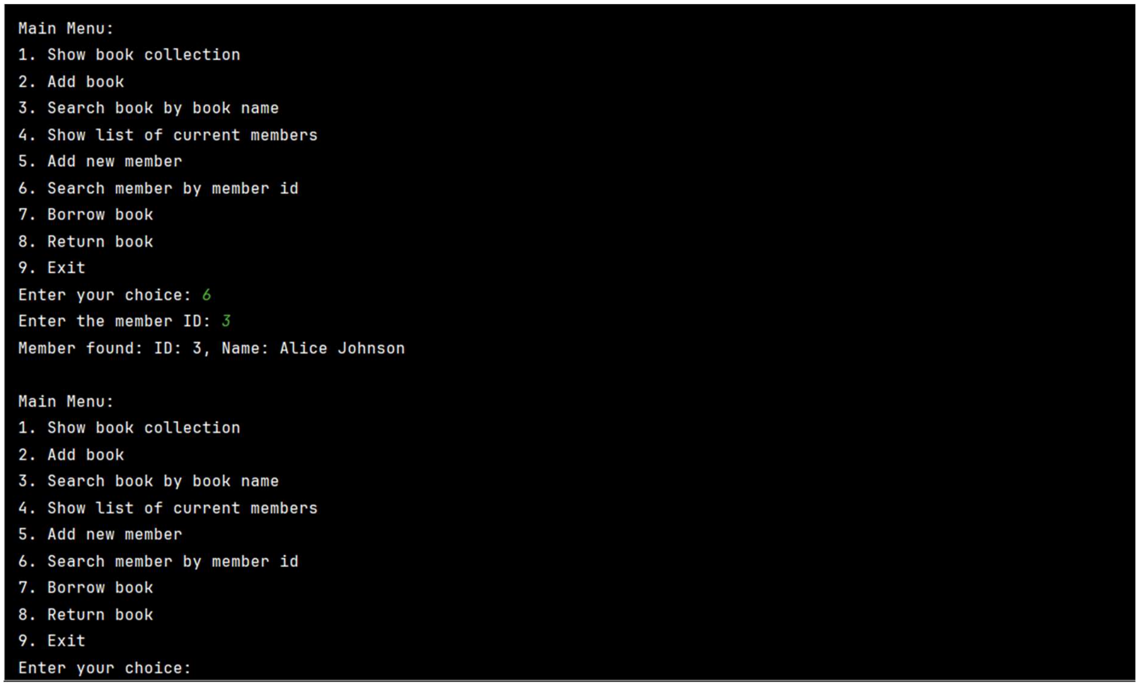
Add new member:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 5
Enter the name of the member: Donald Trump
Member 'Donald Trump' added successfully with ID 6.

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

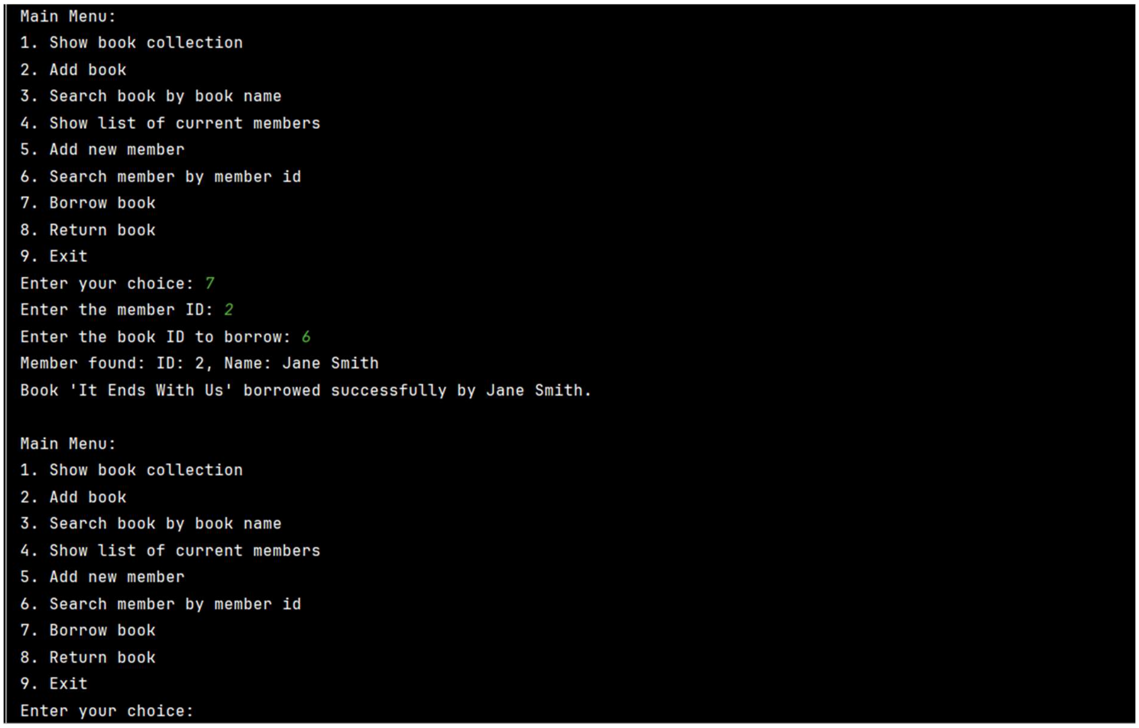
Search member by member id:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 6
Enter the member ID: 3
Member found: ID: 3, Name: Alice Johnson

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

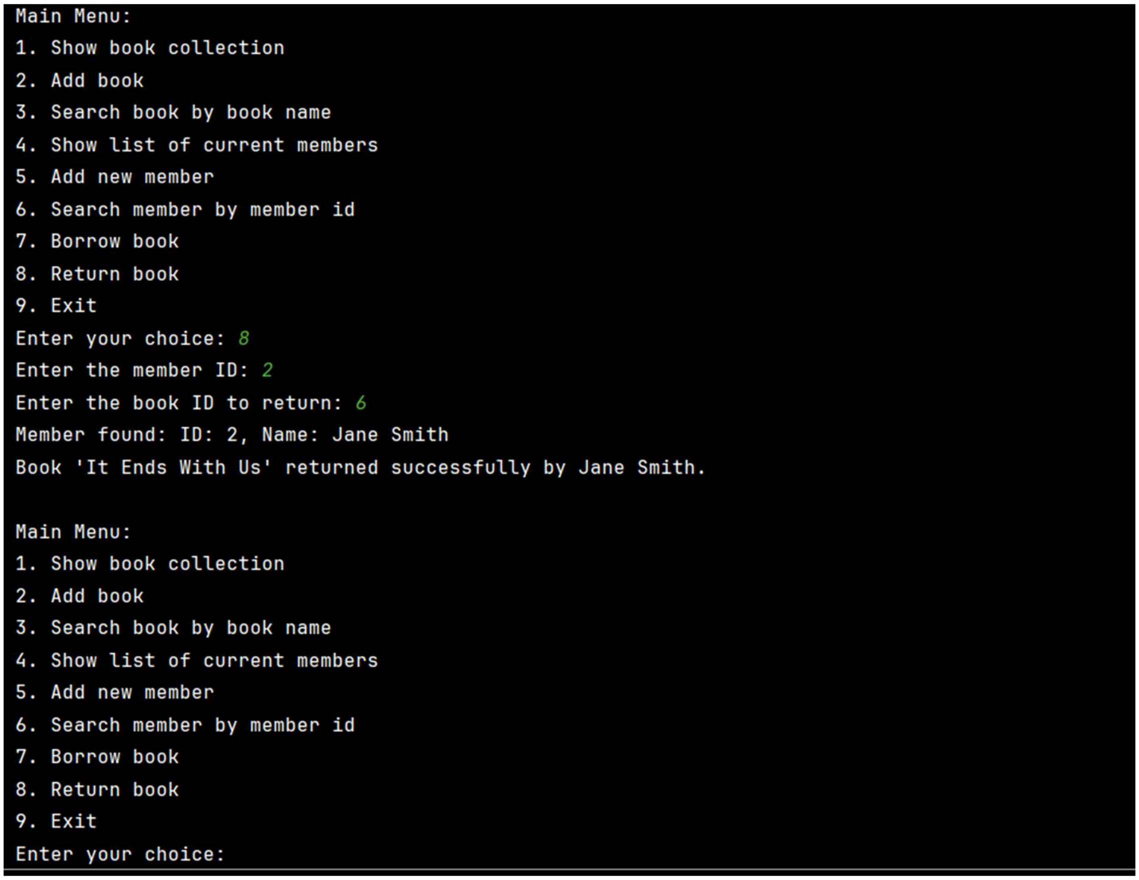
Borrow book:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 7
Enter the member ID: 2
Enter the book ID to borrow: 6
Member found: ID: 2, Name: Jane Smith
Book 'It Ends With Us' borrowed successfully by Jane Smith.

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

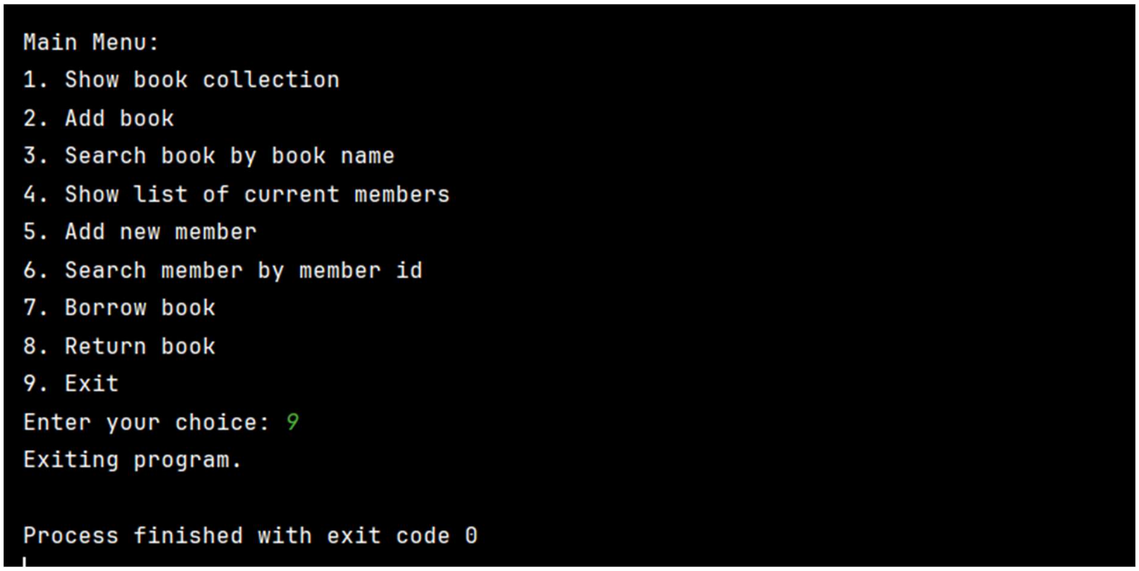
Return book:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 8
Enter the member ID: 2
Enter the book ID to return: 6
Member found: ID: 2, Name: Jane Smith
Book 'It Ends With Us' returned successfully by Jane Smith.

Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice:
```

Exit:



```
Main Menu:
1. Show book collection
2. Add book
3. Search book by book name
4. Show list of current members
5. Add new member
6. Search member by member id
7. Borrow book
8. Return book
9. Exit
Enter your choice: 9
Exiting program.

Process finished with exit code 0
```

## Source Code:

```
class LibraryItem:
    def __init__(self, title):
        self.title = title


class Book(LibraryItem):
    def __init__(self, id, title, author):
        super().__init__(title)
        self.id = id
        self.author = author
        self.available = True


class Member:
    def __init__(self, member_id, name):
        self.member_id = member_id
        self.name = name
        self.borrowed_books = []


class Library:
    def __init__(self):
        self.books = []
        self.members = []
        self.next_book_id = 1

    def add_book(self, title, author):
        book = Book(self.next_book_id, title, author)
        self.books.append(book)
        self.next_book_id += 1
        return book

    def display_books(self):
        if not self.books:
            print("No books available in the library.")
            return
        for book in self.books:
            availability = "Available" if book.available else "Not Available"
            print(f"ID: {book.id}, Title: {book.title}, Author: {book.author}, Availability: {availability}")

    def search_book_by_name(self, name):
        found_books = [book for book in self.books if name.lower() in book.title.lower()]
        if found_books:
            print("Found books matching the search query:")
            for book in found_books:
                availability = "Available" if book.available else "Not Available"
                print(f"ID: {book.id}, Title: {book.title}, Author: {book.author}, Availability: {availability}")
        else:
            print("No books found with the given name.")

    def display_members(self):
        if not self.members:
            print("No members in the library.")
            return
        print("List of current members:")
        for member in self.members:
            print(f"ID: {member.member_id}, Name: {member.name}")
    def add_member(self, name):
        member_id = len(self.members) + 1  # Generating unique member ID
        member = Member(member_id, name)
        self.members.append(member)
        return member

    def search_member_by_id(self, member_id):
        for member in self.members:
            if member.member_id == member_id:
                print(f"Member found: ID: {member.member_id}, Name: {member.name}")
                return member
        print("Member not found.")
        return None

    def borrow_book(self, member_id, book_id):
        member = self.search_member_by_id(member_id)
        if not member:
            print("Cannot borrow book. Member not found.")
            return
        book = self.find_book_by_id(book_id)
        if not book:
            print("Cannot borrow book. Book not found.")
            return
        if not book.available:
            print("Sorry, the book is not available for borrowing.")
            return
        member.borrowed_books.append(book)
        book.available = False
        print(f"Book '{book.title}' borrowed successfully by {member.name}.")

    def return_book(self, member_id, book_id):
        member = self.search_member_by_id(member_id)
        if not member:
            print("Cannot return book. Member not found.")
            return
        for book in member.borrowed_books:
            if book.id == book_id:
                book.available = True
                member.borrowed_books.remove(book)
                print(f"Book '{book.title}' returned successfully by {member.name}.")
                return
        print("Cannot return book. Book not found in the member's borrowed list.")

    def find_book_by_id(self, book_id):
        for book in self.books:
            if book.id == book_id:
                return book
        return None


def main():
    library = Library()

    # Adding some initial book and member data for testing
    library.add_book("Harry Potter and the Sorcerer's Stone", "J.K. Rowling")
    library.add_book("The Great Gatsby", "F. Scott Fitzgerald")
    library.add_book("To Kill a Mockingbird", "Harper Lee")
    library.add_book("1984", "George Orwell")
    library.add_book("Pride and Prejudice", "Jane Austen")

    library.add_member("John Doe")
    library.add_member("Jane Smith")
    library.add_member("Alice Johnson")
    library.add_member("Bob Brown")
    library.add_member("Emma Davis")

    while True:
        print("\nMain Menu:")
        print("1. Show book collection")
        print("2. Add book")
        print("3. Search book by book name")
        print("4. Show list of current members")
        print("5. Add new member")
        print("6. Search member by member id")
        print("7. Borrow book")
        print("8. Return book")
        print("9. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            library.display_books()
        elif choice == "2":
            title = input("Enter the title of the book: ")
            author = input("Enter the author of the book: ")
            new_book = library.add_book(title, author)
            print(f"Book '{new_book.title}' added with ID {new_book.id}.")
        elif choice == "3":
            name = input("Enter the name of the book to search: ")
            library.search_book_by_name(name)
        elif choice == "4":
            library.display_members()
        elif choice == "5":
            name = input("Enter the name of the member: ")
            new_member = library.add_member(name)
            print(f"Member '{new_member.name}' added successfully with ID {new_member.member_id}.")
        elif choice == "6":
            member_id = int(input("Enter the member ID: "))
            library.search_member_by_id(member_id)
        elif choice == "7":
            member_id = int(input("Enter the member ID: "))
            book_id = int(input("Enter the book ID to borrow: "))
            library.borrow_book(member_id, book_id)
        elif choice == "8":
            member_id = int(input("Enter the member ID: "))
            book_id = int(input("Enter the book ID to return: "))
            library.return_book(member_id, book_id)
        elif choice == "9":
            print("Exiting program.")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 9.")


if __name__ == "__main__":
    main()
```