```
import sys
class BookNode:
def __init__(self, book_id, title, author, status="Available"):
self.book_id = book_id
self.title = title
self.author = author
self.status = status
self.next = None
class BookList:
def __init__(self):
self.head = None
def insert_book(self, book_id, title, author):
new_book = BookNode(book_id, title, author)
if not self.head:
self.head = new_book
else:
current = self.head
while current.next:
current = current.next
current.next = new_book
print(f"Book '{title}' inserted.")
def delete_book(self, book_id):
current = self.head
prev = None
while current and current.book_id != book_id:
prev = current
current = current.next
if not current:
print("Book not found.")
return
```

```
if prev:

prev.next = current.next

else:

self.head = current.next

print(f"Book ID {book_id} deleted.")

def search_book(self, book_id):

current = self.head

while current:

if current.book_id == book_id:

print(f"Book Found: ID: {current.book_id}, Title: {current.title}, Author: {current.author}, Status:
{current.status}")

return

current = current.next

print("Book not found.")

def display_books(self):

if not self.head:

print("No books available.")

return

current = self.head

print("Books in Library:")

while current:

print(f"ID: {current.book_id}, Title: {current.title}, Author: {current.author}, Status: {current.status}")

current = current.next

class TransactionStack:

def __init__(self):

self.stack = []

def push(self, transaction):

self.stack.append(transaction)

def pop(self):

if not self.stack:

return None
```

```
return self.stack.pop()
def view_transactions(self):
if not self.stack:
print("No transactions available.")
return
print("Recent Transactions:")
for transaction in reversed(self.stack):
print(transaction)
def is_empty(self):
return len(self.stack) == 0
book_list = BookList()
transaction_stack = TransactionStack()
def main():
while True:
print("\nLibrary Management System Menu:")
print("1. Insert Book")
print("2. Delete Book")
print("3. Search Book")
print("4. Display Books")
print("5. Issue Book")
print("6. Return Book")
print("7. Undo Last Transaction")
print("8. View Transactions")
print("9. Exit")
choice = input("Enter your choice: ")
if choice == '1':
book_id = int(input("Enter Book ID: "))
title = input("Enter Book Title: ")
author = input("Enter Author Name: ")
book_list.insert_book(book_id, title, author)
```

```
elif choice == '2':
book_id = int(input("Enter Book ID to delete: "))
book_list.delete_book(book_id)
elif choice == '3':
book_id = int(input("Enter Book ID to search: "))
book_list.search_book(book_id)
elif choice == '4':
book_list.display_books()
elif choice == '5':
book_id = int(input("Enter Book ID to issue: "))
current = book_list.head
while current:
if current.book_id == book_id:
if current.status == "Available":
current.status = "Issued"
transaction_stack.push(f"Issued Book ID {book_id}")
print(f"Book ID {book_id} issued.")
else:
print("Book is already issued.")
break
current = current.next
else:
print("Book not found.")
elif choice == '6':
book_id = int(input("Enter Book ID to return: "))
current = book_list.head
while current:
if current.book_id == book_id:
if current.status == "Issued":
current.status = "Available"
```

```
transaction_stack.push(f"Returned Book ID {book_id}")
print(f"Book ID {book_id} returned.")
else:
print("Book is not issued.")
break
current = current.next
else:
print("Book not found.")
elif choice == '7':
last_transaction = transaction_stack.pop()
if not last_transaction:
print("No transactions to undo.")
else:
action, _, book_id_str = last_transaction.partition(' ')
book_id = int(book_id_str.split()[-1])
current = book_list.head
while current:
if current.book_id == book_id:
if action == "Issued":
current.status = "Available"
elif action == "Returned":
current.status = "Issued"
break
current = current.next
print(f"Undo: {last_transaction}")
elif choice == '8':
transaction_stack.view_transactions()
elif choice == '9':
sys.exit()
else:
```

```
print("Invalid choice. Please try again.")

if __name__ == "__main__":

main()
```