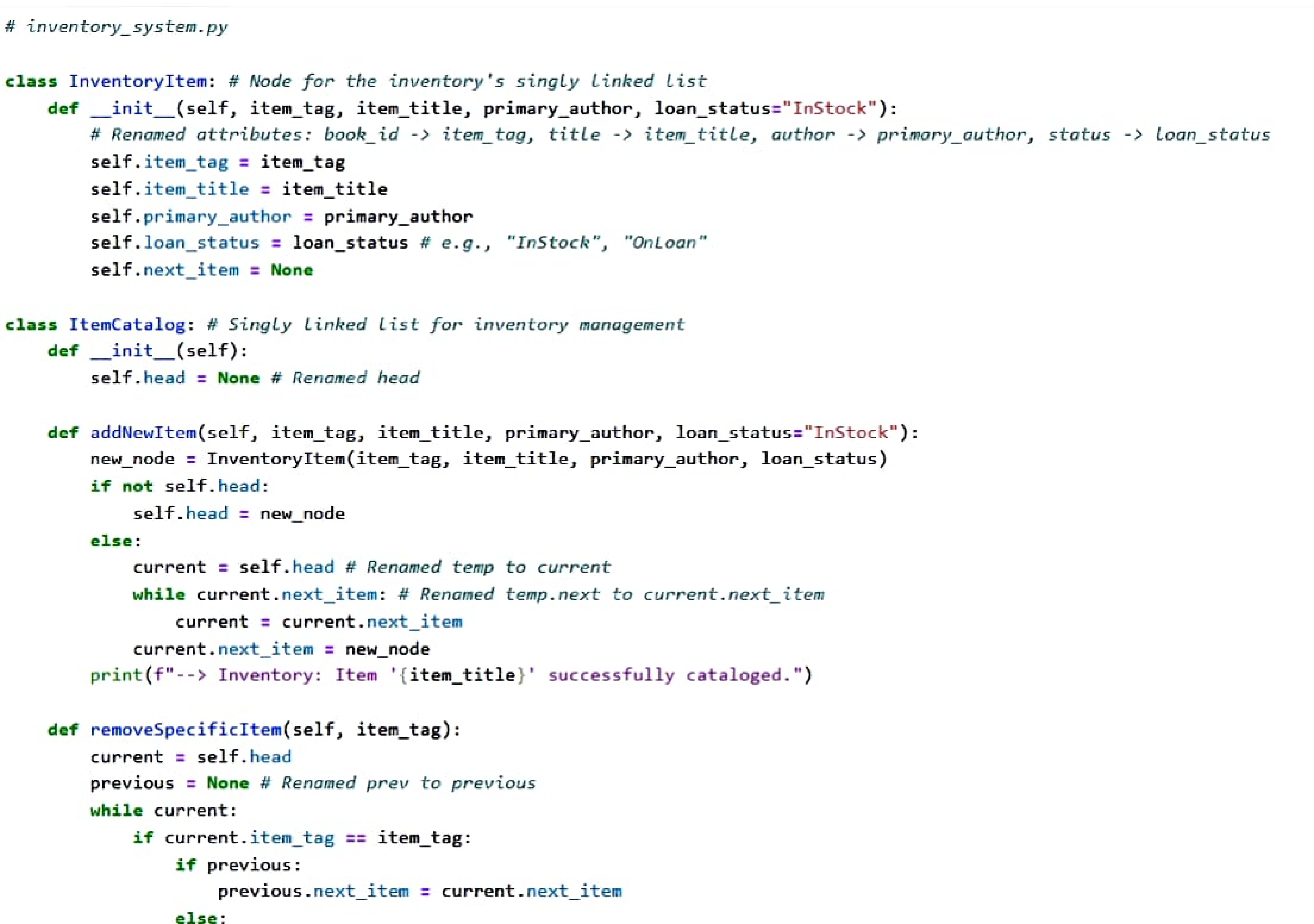
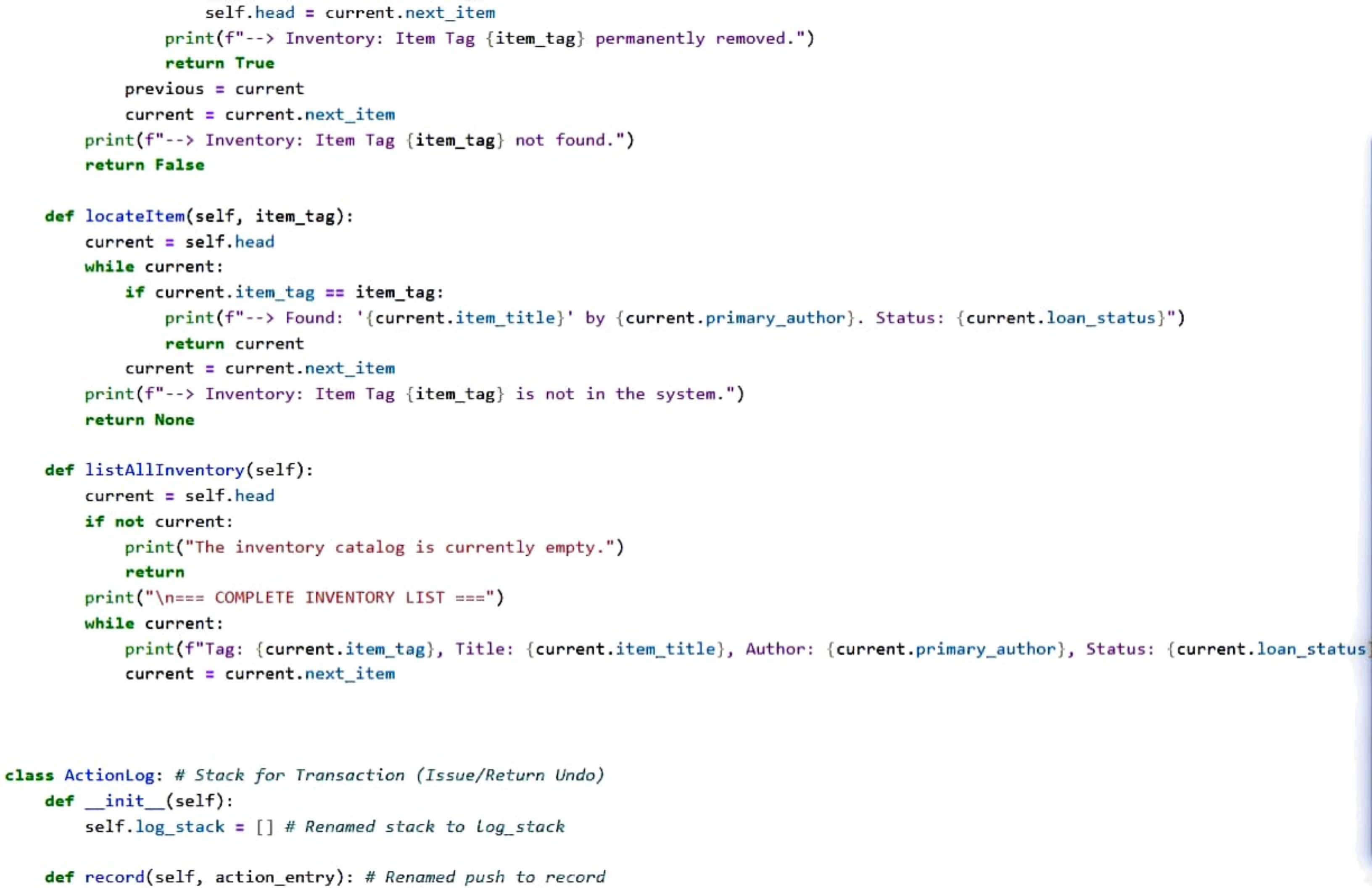
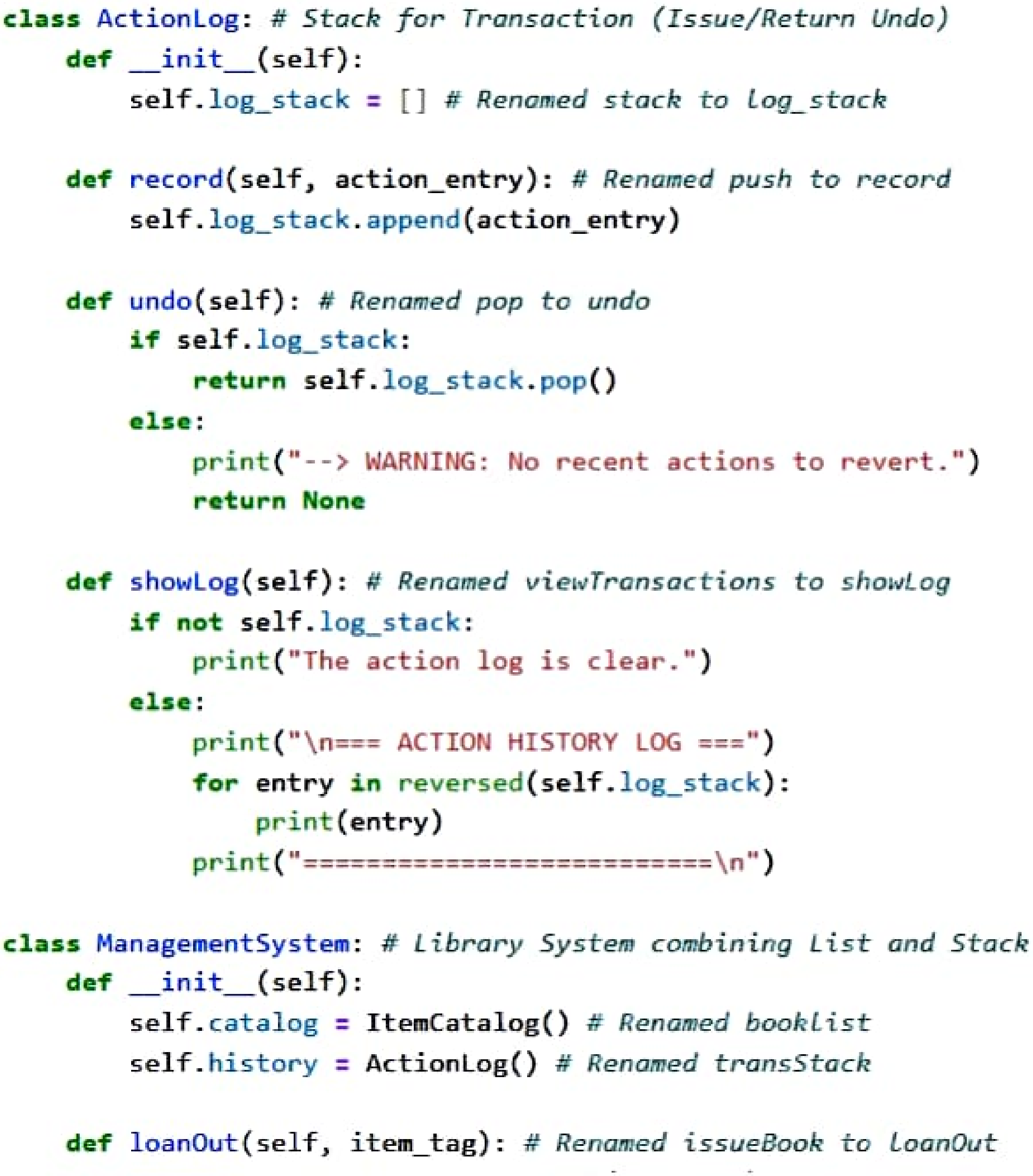


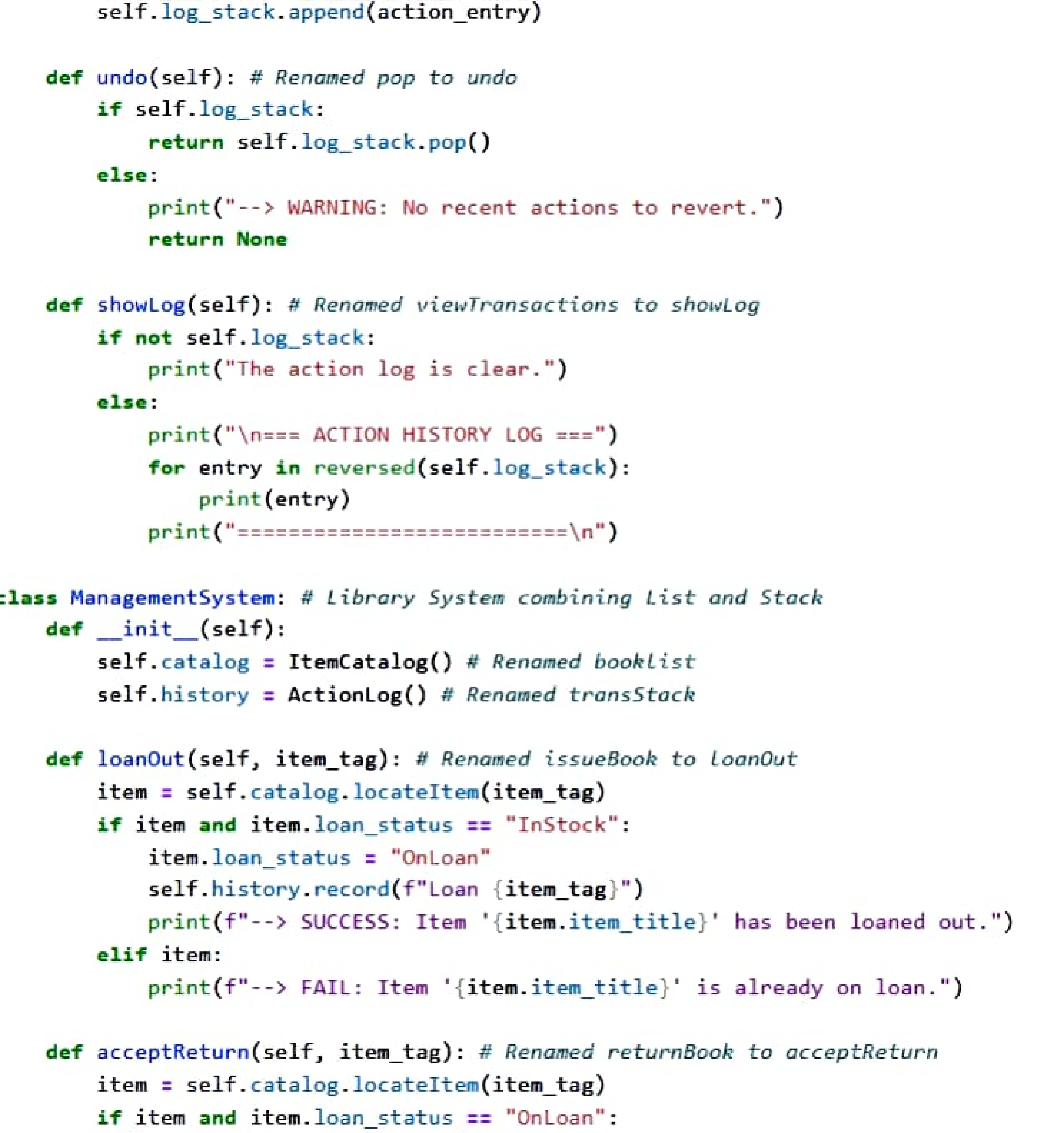
```python
# inventory_system.py

class InventoryItem: # Node for the inventory's singly linked list
    def __init__(self, item_tag, item_title, primary_author, loan_status="InStock"):
        # Renamed attributes: book_id -> item_tag, title -> item_title, author -> primary_author, status -> loan_status
        self.item_tag = item_tag
        self.item_title = item_title
        self.primary_author = primary_author
        self.loan_status = loan_status # e.g., "InStock", "OnLoan"
        self.next_item = None

class ItemCatalog: # Singly linked list for inventory management
    def __init__(self):
        self.head = None # Renamed head

    def addNewItem(self, item_tag, item_title, primary_author, loan_status="InStock"):
        new_node = InventoryItem(item_tag, item_title, primary_author, loan_status)
        if not self.head:
            self.head = new_node
        else:
            current = self.head # Renamed temp to current
            while current.next_item: # Renamed temp.next to current.next_item
                current = current.next_item
            current.next_item = new_node
        print(f"--> Inventory: Item '{item_title}' successfully cataloged.")

    def removeSpecificItem(self, item_tag):
        current = self.head
        previous = None # Renamed prev to previous
        while current:
            if current.item_tag == item_tag:
                if previous:
                    previous.next_item = current.next_item
                else:
```



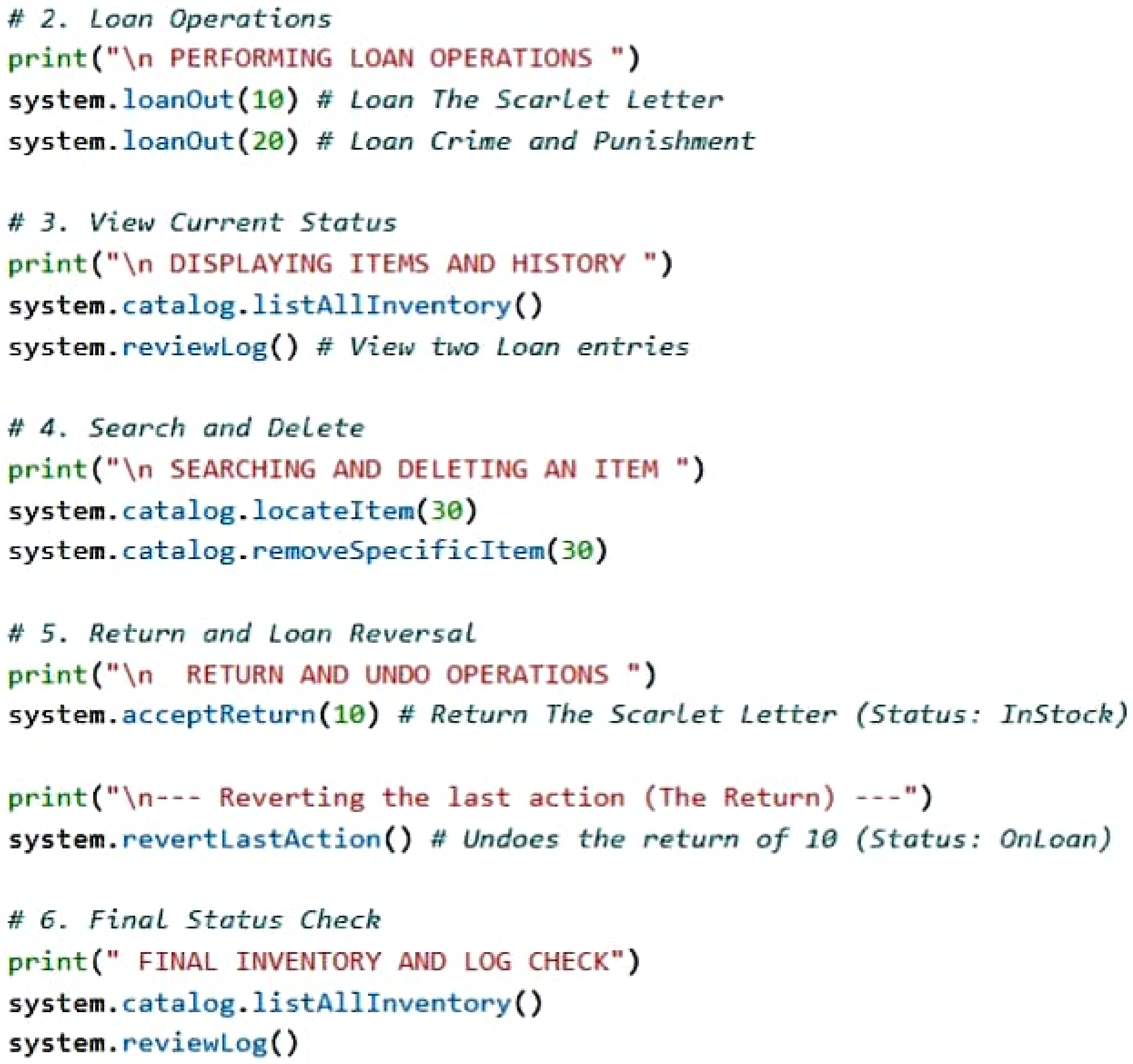
```
                    self.head = current.next_item
                print(f"--> Inventory: Item Tag {item_tag} permanently removed.")
                return True
            previous = current
            current = current.next_item
        print(f"--> Inventory: Item Tag {item_tag} not found.")
        return False

    def locateItem(self, item_tag):
        current = self.head
        while current:
            if current.item_tag == item_tag:
                print(f"--> Found: '{current.item_title}' by {current.primary_author}. Status: {current.loan_status}")
                return current
            current = current.next_item
        print(f"--> Inventory: Item Tag {item_tag} is not in the system.")
        return None

    def listAllInventory(self):
        current = self.head
        if not current:
            print("The inventory catalog is currently empty.")
            return
        print("\n=== COMPLETE INVENTORY LIST ===")
        while current:
            print(f"Tag: {current.item_tag}, Title: {current.item_title}, Author: {current.primary_author}, Status: {current.loan_status
            current = current.next_item


class ActionLog: # Stack for Transaction (Issue/Return Undo)
    def __init__(self):
        self.log_stack = [] # Renamed stack to log_stack

    def record(self, action_entry): # Renamed push to record
```



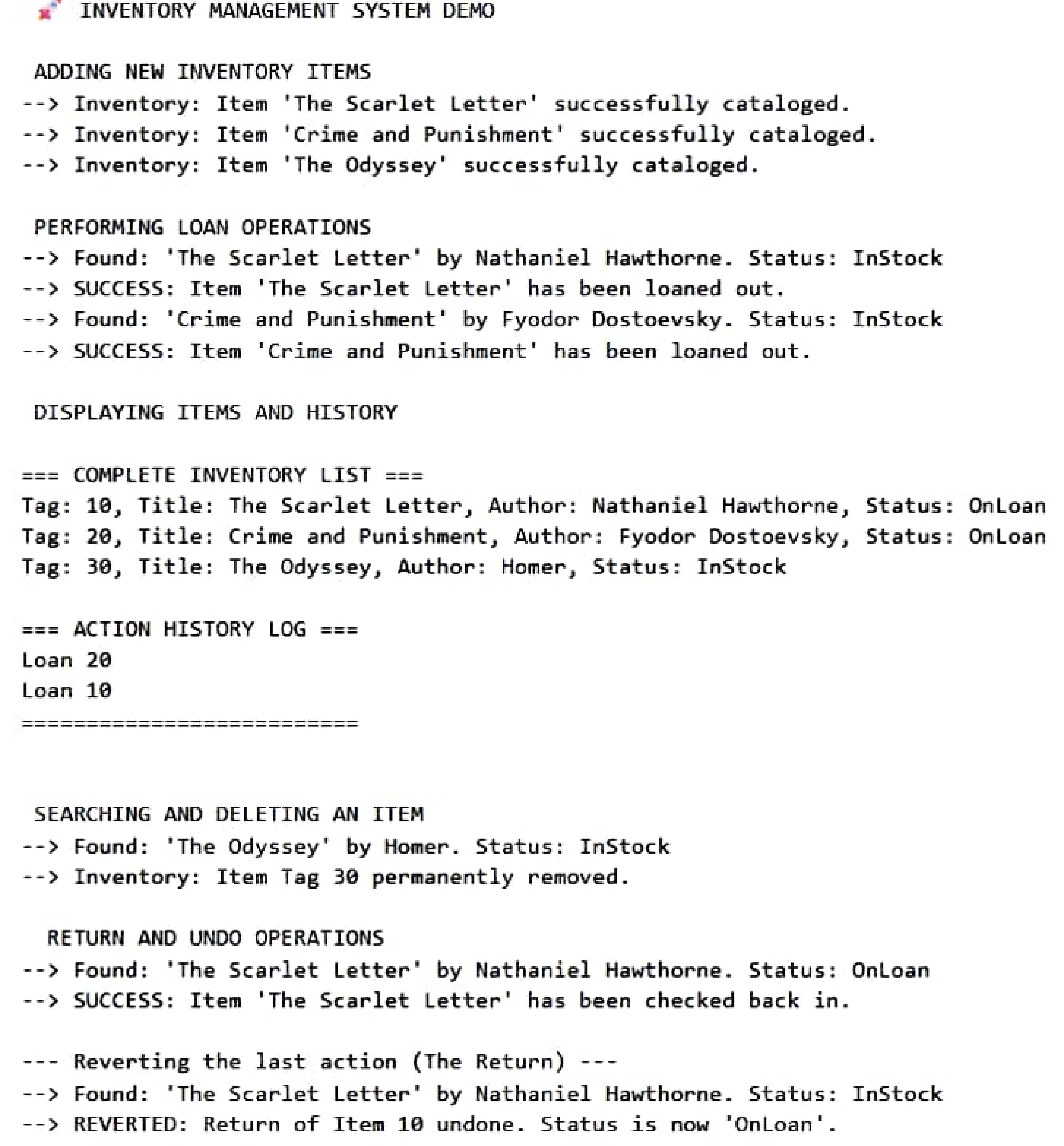
```
class ActionLog: # Stack for Transaction (Issue/Return Undo)
    def __init__(self):
        self.log_stack = [] # Renamed stack to log_stack

    def record(self, action_entry): # Renamed push to record
        self.log_stack.append(action_entry)

    def undo(self): # Renamed pop to undo
        if self.log_stack:
            return self.log_stack.pop()
        else:
            print("--> WARNING: No recent actions to revert.")
            return None

    def showLog(self): # Renamed viewTransactions to showLog
        if not self.log_stack:
            print("The action log is clear.")
        else:
            print("\n=== ACTION HISTORY LOG ===")
            for entry in reversed(self.log_stack):
                print(entry)
            print("==========================\n")

class ManagementSystem: # Library System combining List and Stack
    def __init__(self):
        self.catalog = ItemCatalog() # Renamed bookList
        self.history = ActionLog() # Renamed transStack

    def loanOut(self, item_tag): # Renamed issueBook to loanOut
```



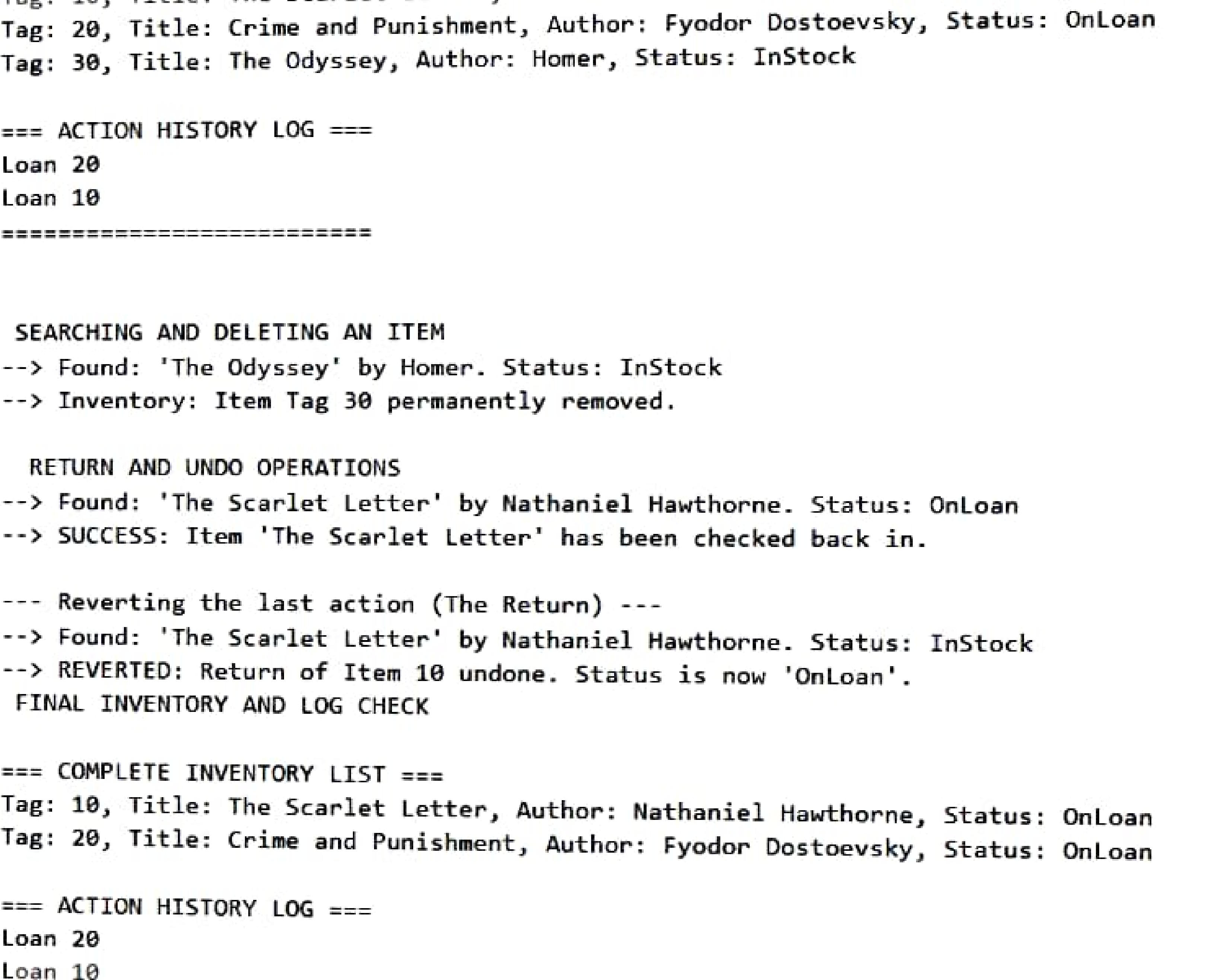
```
        self.log_stack.append(action_entry)

    def undo(self): # Renamed pop to undo
        if self.log_stack:
            return self.log_stack.pop()
        else:
            print("--> WARNING: No recent actions to revert.")
            return None

    def showLog(self): # Renamed viewTransactions to showLog
        if not self.log_stack:
            print("The action log is clear.")
        else:
            print("\n=== ACTION HISTORY LOG ===")
            for entry in reversed(self.log_stack):
                print(entry)
            print("==========================\n")

class ManagementSystem: # Library System combining List and Stack
    def __init__(self):
        self.catalog = ItemCatalog() # Renamed bookList
        self.history = ActionLog() # Renamed transStack

    def loanOut(self, item_tag): # Renamed issueBook to loanOut
        item = self.catalog.locateItem(item_tag)
        if item and item.loan_status == "InStock":
            item.loan_status = "OnLoan"
            self.history.record(f"Loan {item_tag}")
            print(f"--> SUCCESS: Item '{item.item_title}' has been loaned out.")
        elif item:
            print(f"--> FAIL: Item '{item.item_title}' is already on loan.")

    def acceptReturn(self, item_tag): # Renamed returnBook to acceptReturn
        item = self.catalog.locateItem(item_tag)
        if item and item.loan_status == "OnLoan":
```



```
# 2. Loan Operations
print("\n PERFORMING LOAN OPERATIONS ")
system.loanOut(10) # Loan The Scarlet Letter
system.loanOut(20) # Loan Crime and Punishment

# 3. View Current Status
print("\n DISPLAYING ITEMS AND HISTORY ")
system.catalog.listAllInventory()
system.reviewLog() # View two Loan entries

# 4. Search and Delete
print("\n SEARCHING AND DELETING AN ITEM ")
system.catalog.locateItem(30)
system.catalog.removeSpecificItem(30)

# 5. Return and Loan Reversal
print("\n  RETURN AND UNDO OPERATIONS ")
system.acceptReturn(10) # Return The Scarlet Letter (Status: InStock)

print("\n--- Reverting the last action (The Return) ---")
system.revertLastAction() # Undoes the return of 10 (Status: OnLoan)

# 6. Final Status Check
print(" FINAL INVENTORY AND LOG CHECK")
system.catalog.listAllInventory()
system.reviewLog()
```



```
🚀 INVENTORY MANAGEMENT SYSTEM DEMO

 ADDING NEW INVENTORY ITEMS
--> Inventory: Item 'The Scarlet Letter' successfully cataloged.
--> Inventory: Item 'Crime and Punishment' successfully cataloged.
--> Inventory: Item 'The Odyssey' successfully cataloged.

 PERFORMING LOAN OPERATIONS
--> Found: 'The Scarlet Letter' by Nathaniel Hawthorne. Status: InStock
--> SUCCESS: Item 'The Scarlet Letter' has been loaned out.
--> Found: 'Crime and Punishment' by Fyodor Dostoevsky. Status: InStock
--> SUCCESS: Item 'Crime and Punishment' has been loaned out.

 DISPLAYING ITEMS AND HISTORY

=== COMPLETE INVENTORY LIST ===
Tag: 10, Title: The Scarlet Letter, Author: Nathaniel Hawthorne, Status: OnLoan
Tag: 20, Title: Crime and Punishment, Author: Fyodor Dostoevsky, Status: OnLoan
Tag: 30, Title: The Odyssey, Author: Homer, Status: InStock

=== ACTION HISTORY LOG ===
Loan 20
Loan 10
==========================



 SEARCHING AND DELETING AN ITEM
--> Found: 'The Odyssey' by Homer. Status: InStock
--> Inventory: Item Tag 30 permanently removed.

  RETURN AND UNDO OPERATIONS
--> Found: 'The Scarlet Letter' by Nathaniel Hawthorne. Status: OnLoan
--> SUCCESS: Item 'The Scarlet Letter' has been checked back in.

--- Reverting the last action (The Return) ---
--> Found: 'The Scarlet Letter' by Nathaniel Hawthorne. Status: InStock
--> REVERTED: Return of Item 10 undone. Status is now 'OnLoan'.
```



```
Tag: 10, Title: The Scarlet Letter,
Tag: 20, Title: Crime and Punishment, Author: Fyodor Dostoevsky, Status: OnLoan
Tag: 30, Title: The Odyssey, Author: Homer, Status: InStock

=== ACTION HISTORY LOG ===
Loan 20
Loan 10
==========================



 SEARCHING AND DELETING AN ITEM
--> Found: 'The Odyssey' by Homer. Status: InStock
--> Inventory: Item Tag 30 permanently removed.

  RETURN AND UNDO OPERATIONS
--> Found: 'The Scarlet Letter' by Nathaniel Hawthorne. Status: OnLoan
--> SUCCESS: Item 'The Scarlet Letter' has been checked back in.

--- Reverting the last action (The Return) ---
--> Found: 'The Scarlet Letter' by Nathaniel Hawthorne. Status: InStock
--> REVERTED: Return of Item 10 undone. Status is now 'OnLoan'.
 FINAL INVENTORY AND LOG CHECK

=== COMPLETE INVENTORY LIST ===
Tag: 10, Title: The Scarlet Letter, Author: Nathaniel Hawthorne, Status: OnLoan
Tag: 20, Title: Crime and Punishment, Author: Fyodor Dostoevsky, Status: OnLoan

=== ACTION HISTORY LOG ===
Loan 20
Loan 10
```