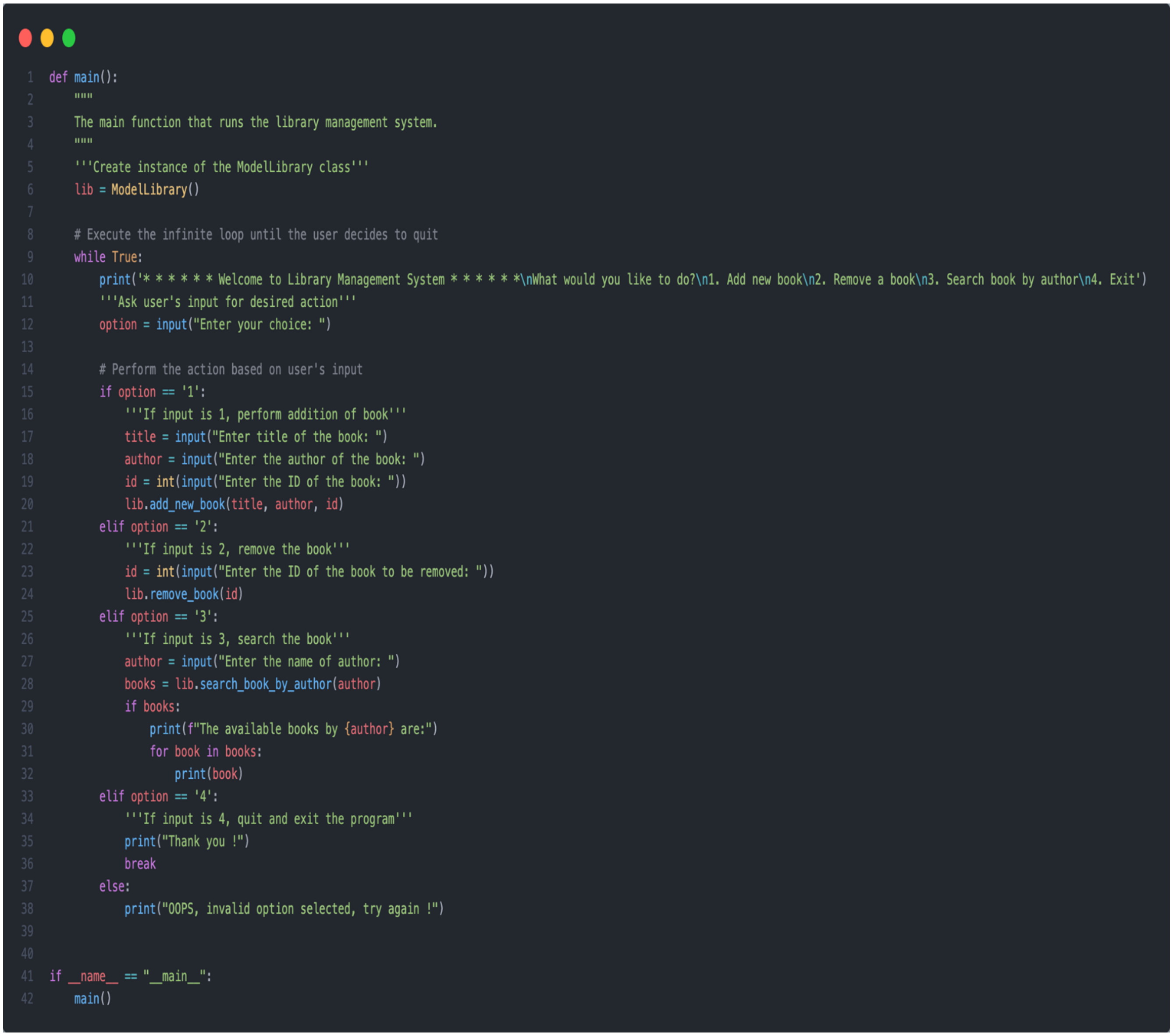
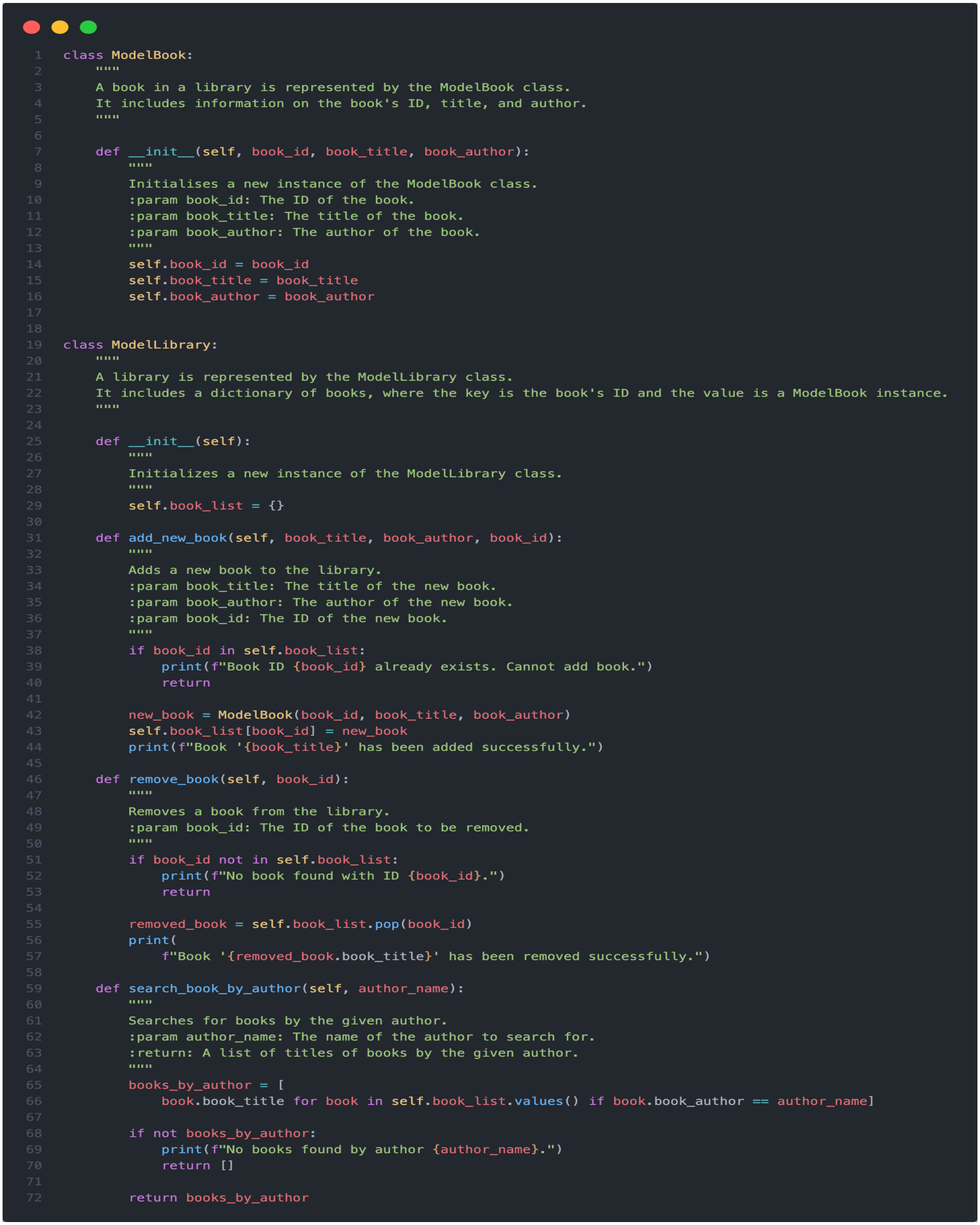
Assignment 7
Student ID: 901142
Student Name: Roshan Shrestha

Source code:



```python
def main():
    """
    The main function that runs the library management system.
    """
    '''Create instance of the ModelLibrary class'''
    lib = ModelLibrary()

    # Execute the infinite loop until the user decides to quit
    while True:
        print('* * * * * * Welcome to Library Management System * * * * * *\nWhat would you like to do?\n1. Add new book\n2. Remove a book\n3. Search book by author\n4. Exit')
        '''Ask user's input for desired action'''
        option = input("Enter your choice: ")

        # Perform the action based on user's input
        if option == '1':
            '''If input is 1, perform addition of book'''
            title = input("Enter title of the book: ")
            author = input("Enter the author of the book: ")
            id = int(input("Enter the ID of the book: "))
            lib.add_new_book(title, author, id)
        elif option == '2':
            '''If input is 2, remove the book'''
            id = int(input("Enter the ID of the book to be removed: "))
            lib.remove_book(id)
        elif option == '3':
            '''If input is 3, search the book'''
            author = input("Enter the name of author: ")
            books = lib.search_book_by_author(author)
            if books:
                print(f"The available books by {author} are:")
                for book in books:
                    print(book)
        elif option == '4':
            '''If input is 4, quit and exit the program'''
            print("Thank you !")
            break
        else:
            print("OOPS, invalid option selected, try again !")


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

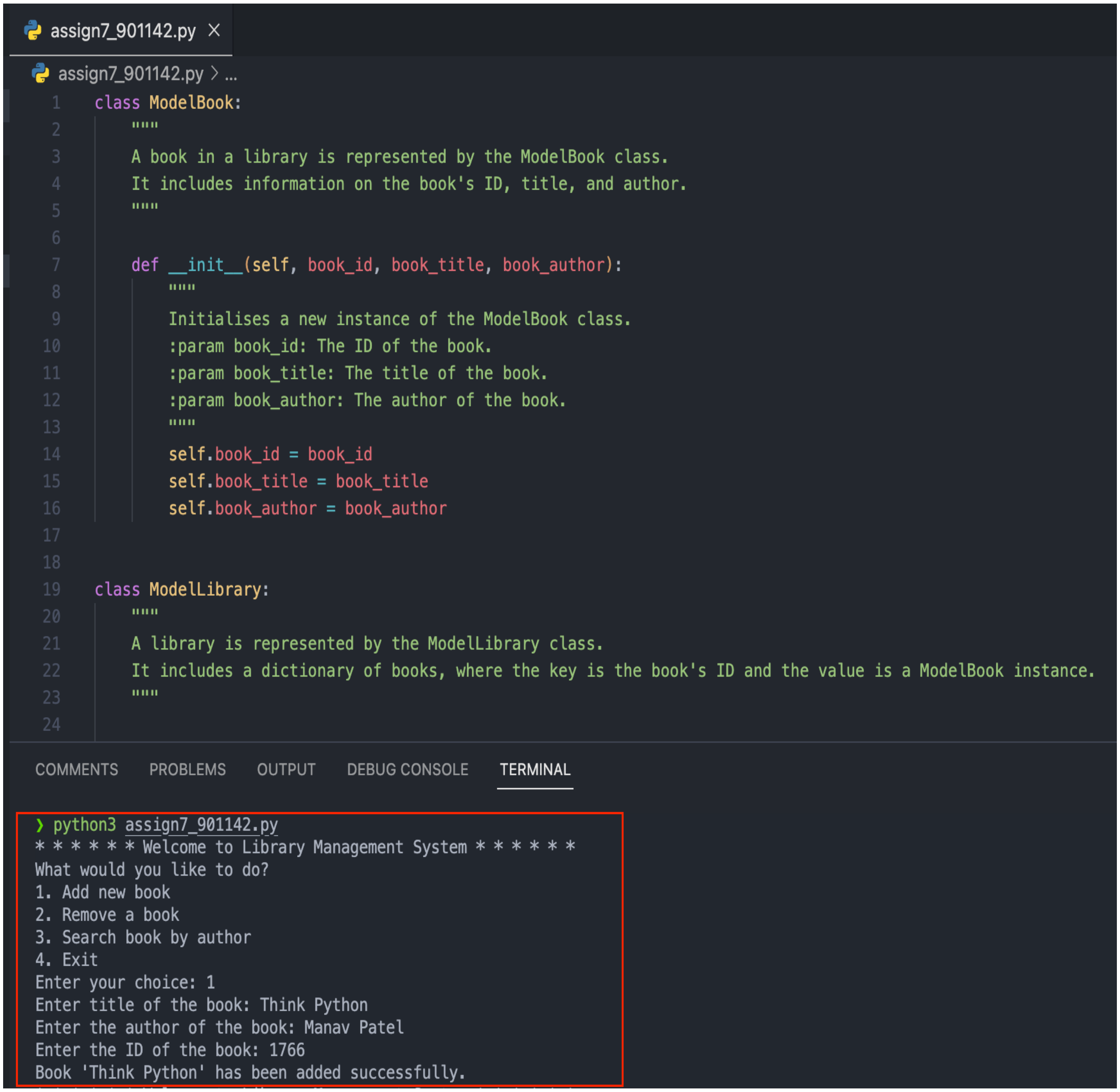
Figure 1: Source code for main function and performed actions.



```python
class ModelBook:
    """
    A book in a library is represented by the ModelBook class.
    It includes information on the book's ID, title, and author.
    """

    def __init__(self, book_id, book_title, book_author):
        """
        Initialises a new instance of the ModelBook class.
        :param book_id: The ID of the book.
        :param book_title: The title of the book.
        :param book_author: The author of the book.
        """
        self.book_id = book_id
        self.book_title = book_title
        self.book_author = book_author


class ModelLibrary:
    """
    A library is represented by the ModelLibrary class.
    It includes a dictionary of books, where the key is the book's ID and the value is a ModelBook instance.
    """

    def __init__(self):
        """
        Initializes a new instance of the ModelLibrary class.
        """
        self.book_list = {}

    def add_new_book(self, book_title, book_author, book_id):
        """
        Adds a new book to the library.
        :param book_title: The title of the new book.
        :param book_author: The author of the new book.
        :param book_id: The ID of the new book.
        """
        if book_id in self.book_list:
            print(f"Book ID {book_id} already exists. Cannot add book.")
            return

        new_book = ModelBook(book_id, book_title, book_author)
        self.book_list[book_id] = new_book
        print(f"Book '{book_title}' has been added successfully.")

    def remove_book(self, book_id):
        """
        Removes a book from the library.
        :param book_id: The ID of the book to be removed.
        """
        if book_id not in self.book_list:
            print(f"No book found with ID {book_id}.")
            return

        removed_book = self.book_list.pop(book_id)
        print(
            f"Book '{removed_book.book_title}' has been removed successfully.")

    def search_book_by_author(self, author_name):
        """
        Searches for books by the given author.
        :param author_name: The name of the author to search for.
        :return: A list of titles of books by the given author.
        """
        books_by_author = [
            book.book_title for book in self.book_list.values() if book.book_author == author_name]

        if not books_by_author:
            print(f"No books found by author {author_name}.")
            return []

        return books_by_author
```

Figure 2: Source code for Classes for book and Library with function to add, remove and search book.

Test case for adding book:

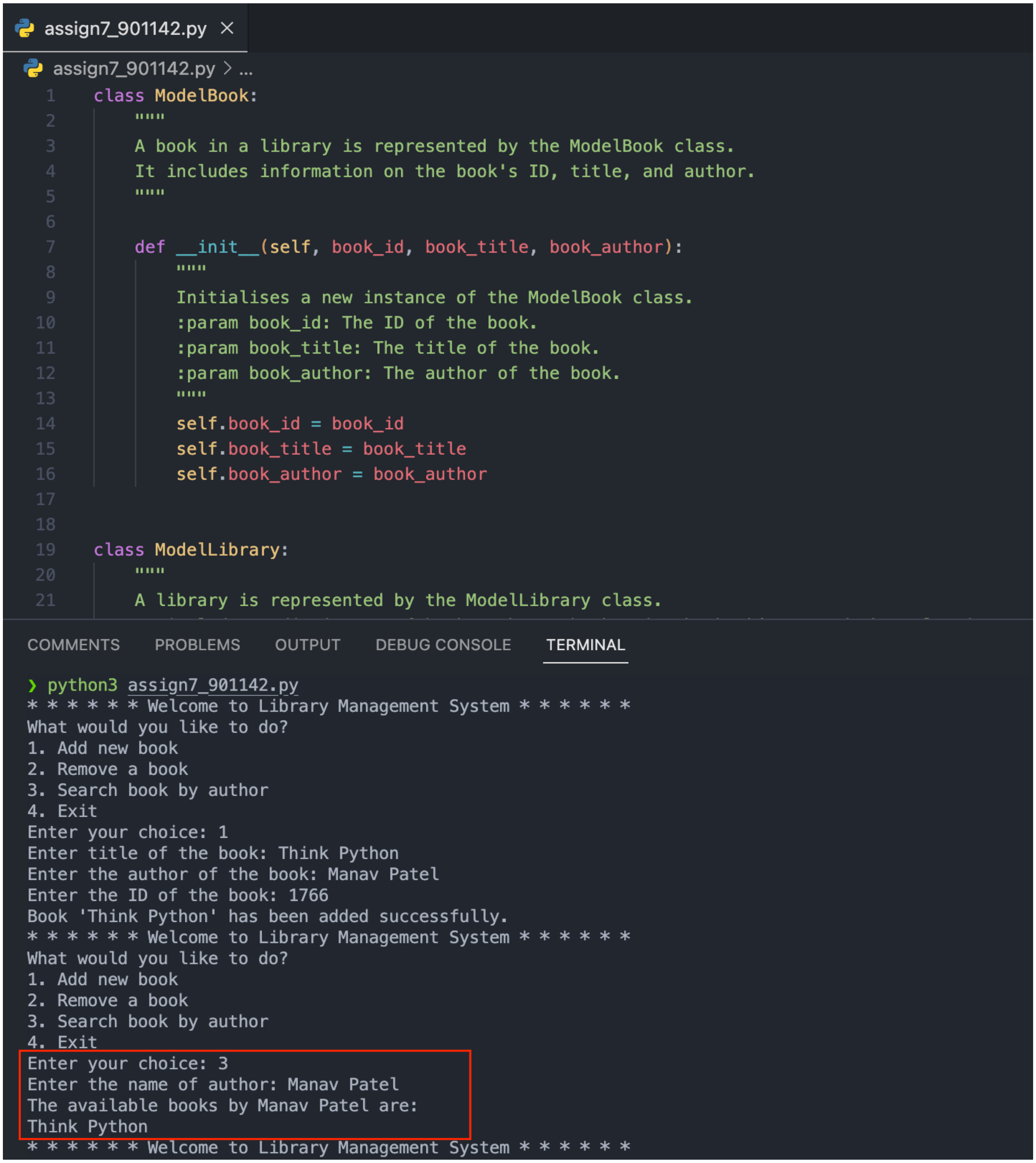


```python
class ModelBook:
    """
    A book in a library is represented by the ModelBook class.
    It includes information on the book's ID, title, and author.
    """

    def __init__(self, book_id, book_title, book_author):
        """
        Initialises a new instance of the ModelBook class.
        :param book_id: The ID of the book.
        :param book_title: The title of the book.
        :param book_author: The author of the book.
        """
        self.book_id = book_id
        self.book_title = book_title
        self.book_author = book_author


class ModelLibrary:
    """
    A library is represented by the ModelLibrary class.
    It includes a dictionary of books, where the key is the book's ID and the value is a ModelBook instance.
    """

```

```
❯ python3 assign7_901142.py
* * * * * * Welcome to Library Management System * * * * * *
What would you like to do?
1. Add new book
2. Remove a book
3. Search book by author
4. Exit
Enter your choice: 1
Enter title of the book: Think Python
Enter the author of the book: Manav Patel
Enter the ID of the book: 1766
Book 'Think Python' has been added successfully.
```

Figure 3: Adding a book.

Test case for searching book:

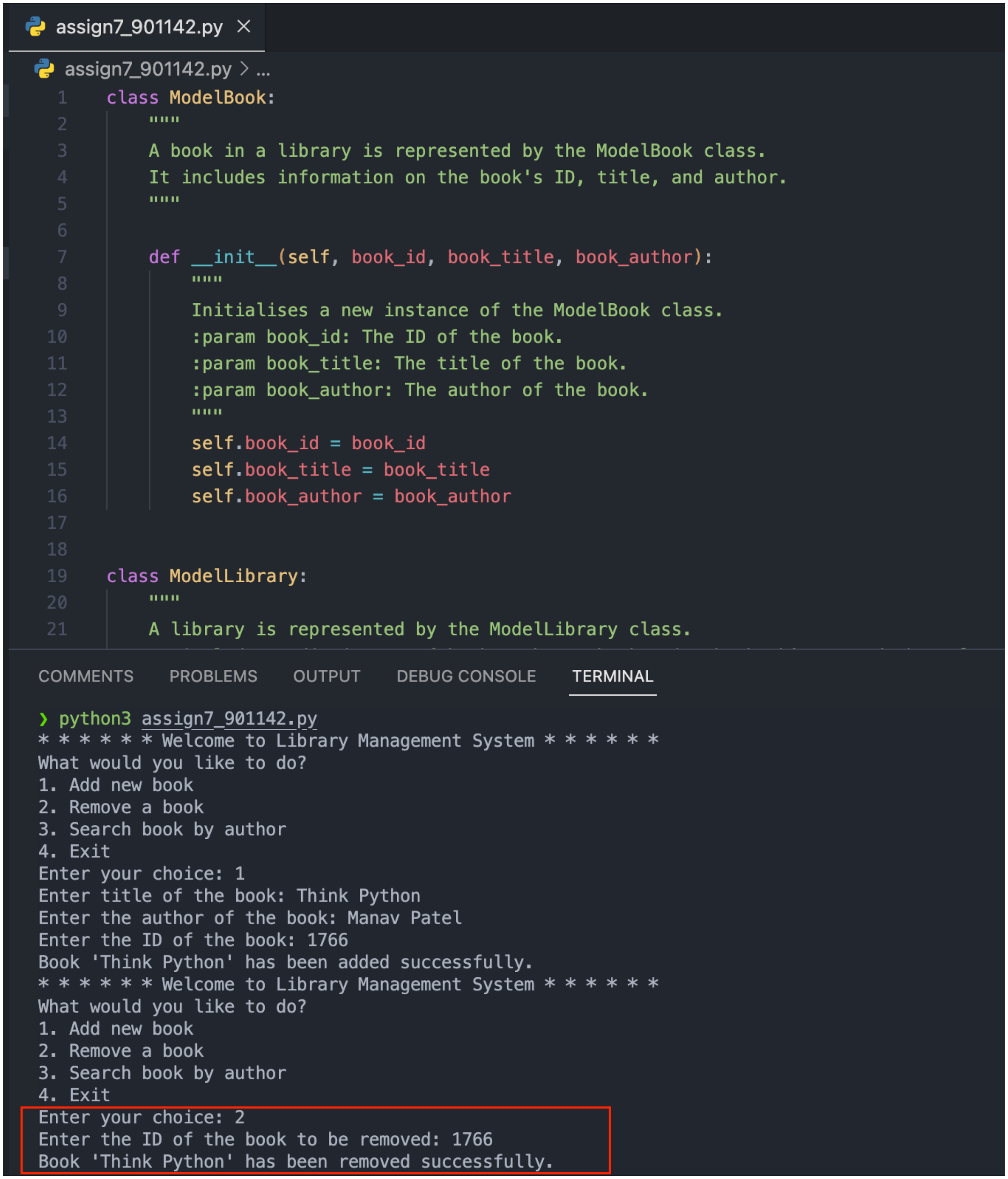


```python
class ModelBook:
    """
    A book in a library is represented by the ModelBook class.
    It includes information on the book's ID, title, and author.
    """

    def __init__(self, book_id, book_title, book_author):
        """
        Initialises a new instance of the ModelBook class.
        :param book_id: The ID of the book.
        :param book_title: The title of the book.
        :param book_author: The author of the book.
        """
        self.book_id = book_id
        self.book_title = book_title
        self.book_author = book_author


class ModelLibrary:
    """
    A library is represented by the ModelLibrary class.
```

```
❯ python3 assign7_901142.py
* * * * * * Welcome to Library Management System * * * * * *
What would you like to do?
1. Add new book
2. Remove a book
3. Search book by author
4. Exit
Enter your choice: 1
Enter title of the book: Think Python
Enter the author of the book: Manav Patel
Enter the ID of the book: 1766
Book 'Think Python' has been added successfully.
* * * * * * Welcome to Library Management System * * * * * *
What would you like to do?
1. Add new book
2. Remove a book
3. Search book by author
4. Exit
Enter your choice: 3
Enter the name of author: Manav Patel
The available books by Manav Patel are:
Think Python
* * * * * * Welcome to Library Management System * * * * * *
```

Figure 4: Search for book.

Test case for removing book:



```python
class ModelBook:
    """
    A book in a library is represented by the ModelBook class.
    It includes information on the book's ID, title, and author.
    """

    def __init__(self, book_id, book_title, book_author):
        """
        Initialises a new instance of the ModelBook class.
        :param book_id: The ID of the book.
        :param book_title: The title of the book.
        :param book_author: The author of the book.
        """
        self.book_id = book_id
        self.book_title = book_title
        self.book_author = book_author


class ModelLibrary:
    """
    A library is represented by the ModelLibrary class.
```

```
❯ python3 assign7_901142.py
* * * * * * Welcome to Library Management System * * * * * *
What would you like to do?
1. Add new book
2. Remove a book
3. Search book by author
4. Exit
Enter your choice: 1
Enter title of the book: Think Python
Enter the author of the book: Manav Patel
Enter the ID of the book: 1766
Book 'Think Python' has been added successfully.
* * * * * * Welcome to Library Management System * * * * * *
What would you like to do?
1. Add new book
2. Remove a book
3. Search book by author
4. Exit
Enter your choice: 2
Enter the ID of the book to be removed: 1766
Book 'Think Python' has been removed successfully.
```

Figure 5: Removing a book.