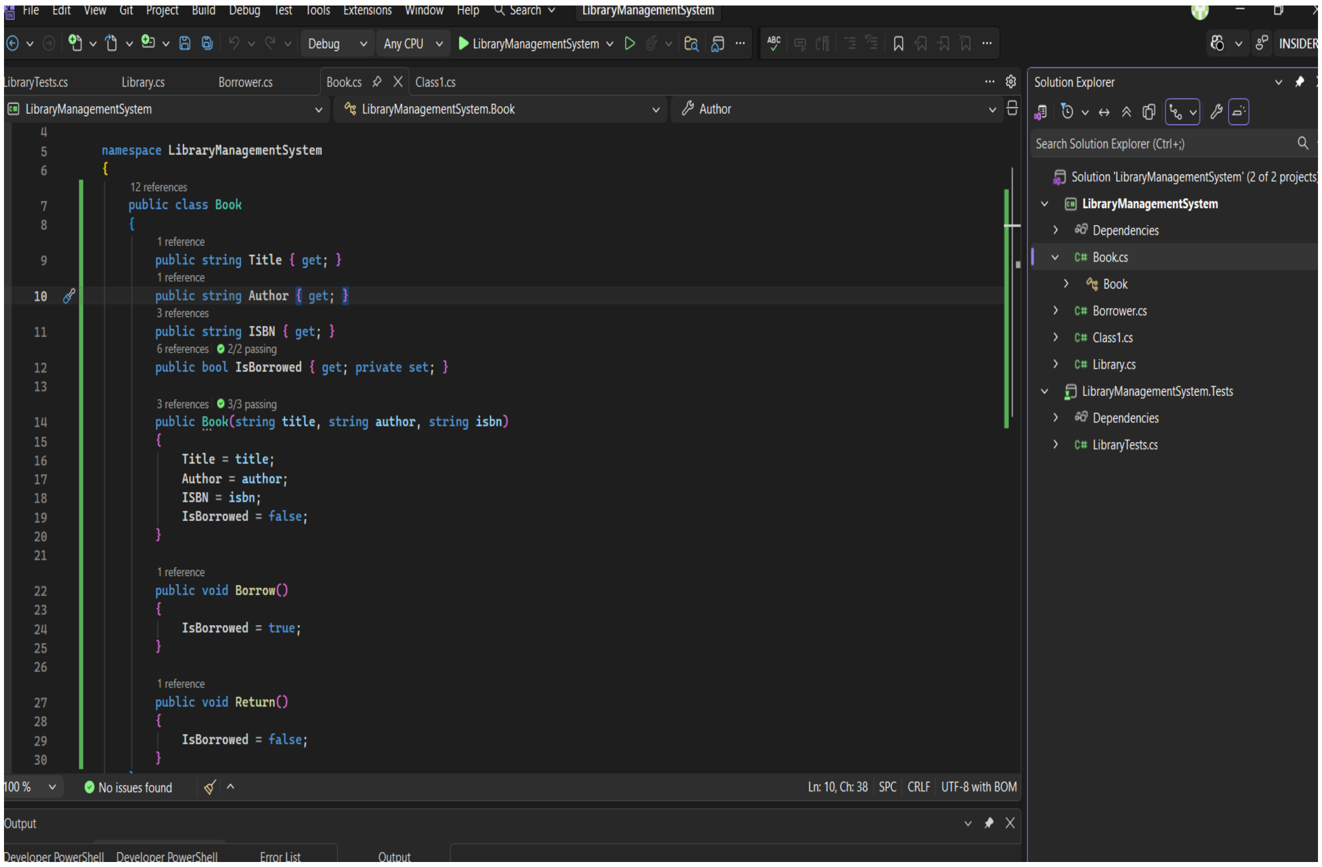
STEP 1: Create the Solution

1. Click Create a new project
2. Select Class Library (.NET)
3. Project Name: LibraryManagementSystem
4. This project will contain Book, Borrower, and Library classes

STEP 2: Create Required Classes

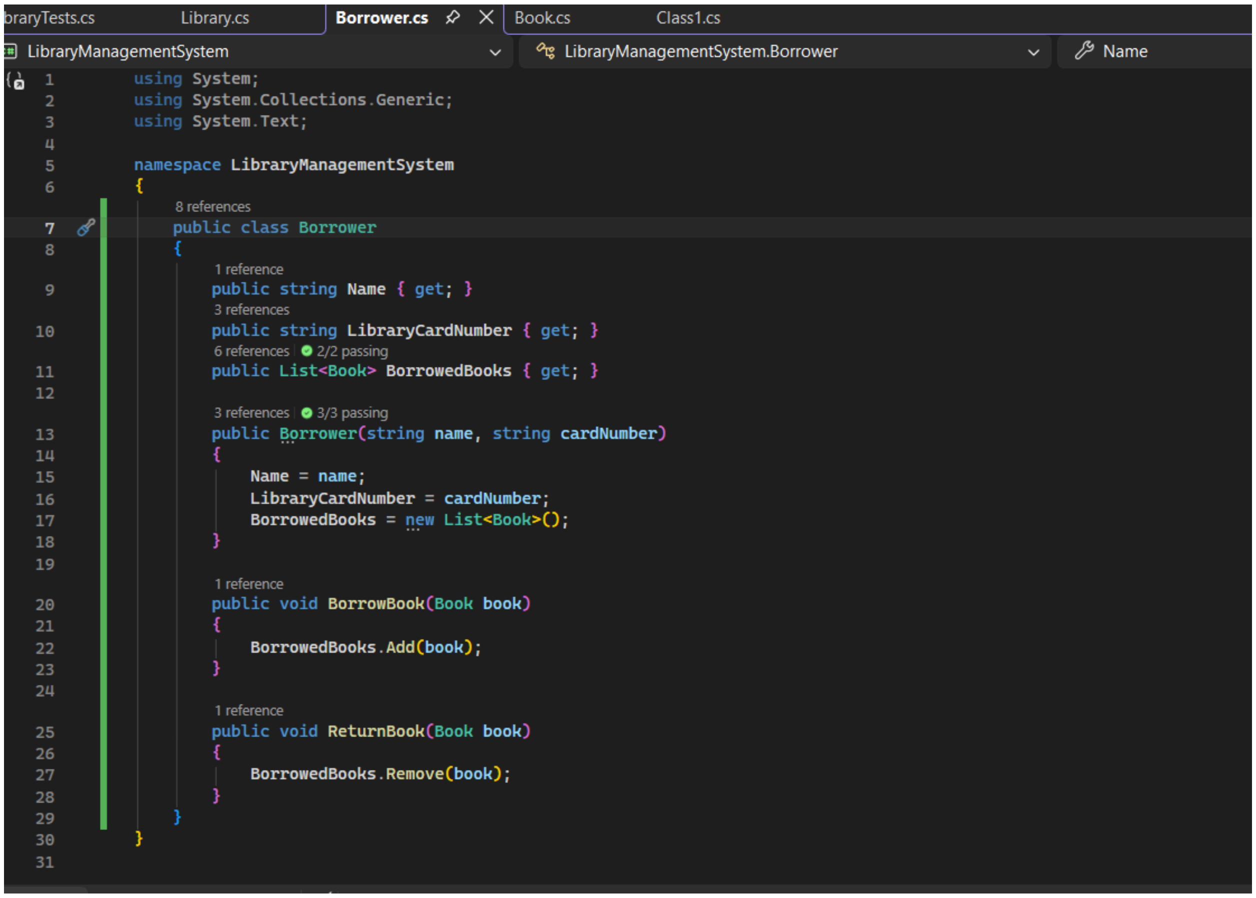


```
namespace LibraryManagementSystem
{
    12 references
    public class Book
    {
        1 reference
        public string Title { get; }
        1 reference
        public string Author { get; }
        3 references
        public string ISBN { get; }
        6 references | 2/2 passing
        public bool IsBorrowed { get; private set; }

        3 references | 3/3 passing
        public Book(string title, string author, string isbn)
        {
            Title = title;
            Author = author;
            ISBN = isbn;
            IsBorrowed = false;
        }

        1 reference
        public void Borrow()
        {
            IsBorrowed = true;
        }

        1 reference
        public void Return()
        {
            IsBorrowed = false;
        }
```

Book. cs Code :

```
using System;

using System.Collections.Generic;

using System.Text;

namespace LibraryManagementSystem

{

    public class Book

    {

        public string Title { get; }

        public string Author { get; }

        public string ISBN { get; }

        public bool IsBorrowed { get; private set; }


        public Book(string title, string author, string isbn)

        {

            Title = title;

            Author = author;

            ISBN = isbn;

            IsBorrowed = false;

        }


        public void Borrow()

        {

            IsBorrowed = true;

        }
        public void Return()
        {
            IsBorrowed = false;
        }
    }
}
```

Borrower. cs

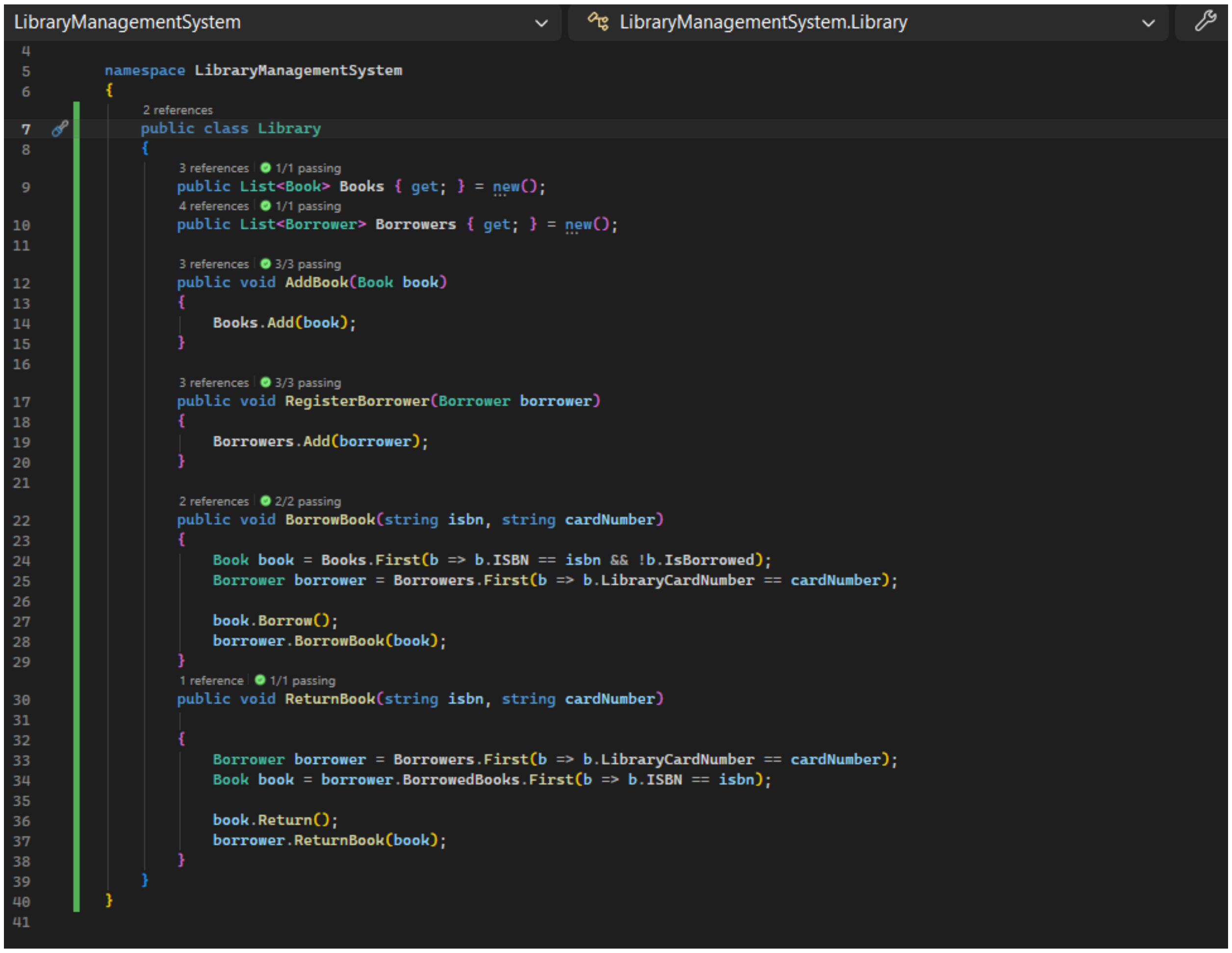


```
using System;
using System.Collections.Generic;
using System.Text;

namespace LibraryManagementSystem
{
    8 references
    public class Borrower
    {
        1 reference
        public string Name { get; }
        3 references
        public string LibraryCardNumber { get; }
        6 references | 2/2 passing
        public List<Book> BorrowedBooks { get; }

        3 references | 3/3 passing
        public Borrower(string name, string cardNumber)
        {
            Name = name;
            LibraryCardNumber = cardNumber;
            BorrowedBooks = new List<Book>();
        }

        1 reference
        public void BorrowBook(Book book)
        {
            BorrowedBooks.Add(book);
        }

        1 reference
        public void ReturnBook(Book book)
        {
            BorrowedBooks.Remove(book);
        }
    }
}
```

Borrower. cs Code:

```
using System;

using System.Collections.Generic;

using System.Text;

namespace LibraryManagementSystem

{

    public class Borrower

    {

        public string Name { get; }

        public string LibraryCardNumber { get; }

        public List<Book> BorrowedBooks { get; }


        public Borrower(string name, string cardNumber)

        {

            Name = name;

            LibraryCardNumber = cardNumber;

            BorrowedBooks = new List<Book>();

        }


        public void BorrowBook(Book book)

        {

            BorrowedBooks.Add(book);

        }


        public void ReturnBook(Book book)

        {
            BorrowedBooks.Remove(book);
        }
    }
}
```

Library. cs



```
namespace LibraryManagementSystem
{
    2 references
    public class Library
    {
        3 references | 1/1 passing
        public List<Book> Books { get; } = new();
        4 references | 1/1 passing
        public List<Borrower> Borrowers { get; } = new();

        3 references | 3/3 passing
        public void AddBook(Book book)
        {
            Books.Add(book);
        }

        3 references | 3/3 passing
        public void RegisterBorrower(Borrower borrower)
        {
            Borrowers.Add(borrower);
        }

        2 references | 2/2 passing
        public void BorrowBook(string isbn, string cardNumber)
        {
            Book book = Books.First(b => b.ISBN == isbn && !b.IsBorrowed);
            Borrower borrower = Borrowers.First(b => b.LibraryCardNumber == cardNumber);

            book.Borrow();
            borrower.BorrowBook(book);
        }
        1 reference | 1/1 passing
        public void ReturnBook(string isbn, string cardNumber)

        {
            Borrower borrower = Borrowers.First(b => b.LibraryCardNumber == cardNumber);
            Book book = borrower.BorrowedBooks.First(b => b.ISBN == isbn);

            book.Return();
            borrower.ReturnBook(book);
        }
    }
}
```

Library .cs Code

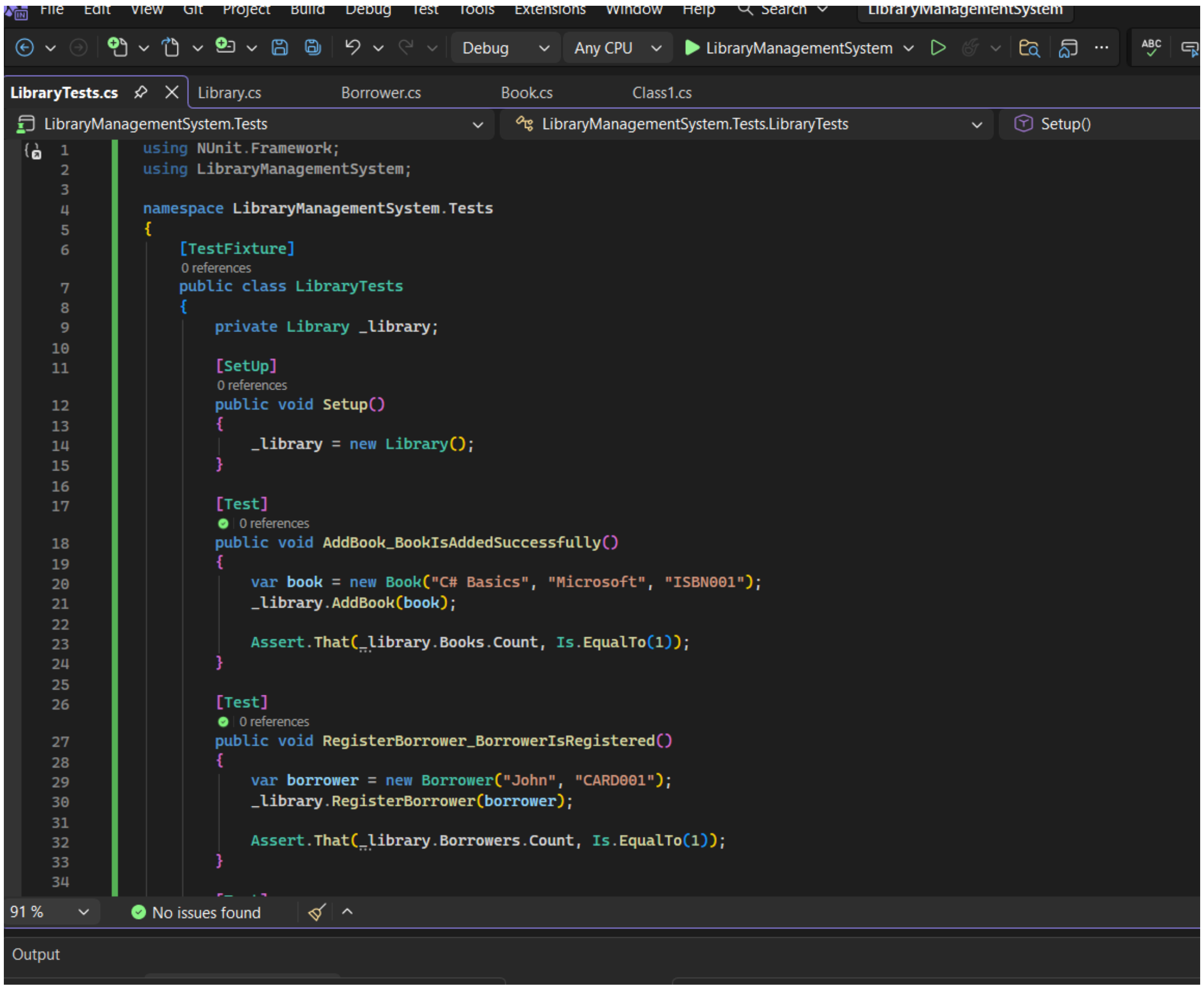
```
namespace LibraryManagementSystem
{
    public class Library
    {
        public List<Book> Books { get; } = new();
        public List<Borrower> Borrowers { get; } = new();

        public void AddBook(Book book)
        {
            Books.Add(book);
        }

        public void RegisterBorrower(Borrower borrower)
        {
            Borrowers.Add(borrower);
        }

        public void BorrowBook(string isbn, string cardNumber)
        {
            Book book = Books.First(b => b.ISBN == isbn && !b.IsBorrowed);
            Borrower borrower = Borrowers.First(b => b.LibraryCardNumber == cardNumber);

            book.Borrow();
            borrower.BorrowBook(book);
        }
        public void ReturnBook(string isbn, string cardNumber)

        {
            Borrower borrower = Borrowers.First(b => b.LibraryCardNumber == cardNumber);
            Book book = borrower.BorrowedBooks.First(b => b.ISBN == isbn);

            book.Return();
            borrower.ReturnBook(book);
        }
    }
}
```

STEP 3: Create NUnit Test Project

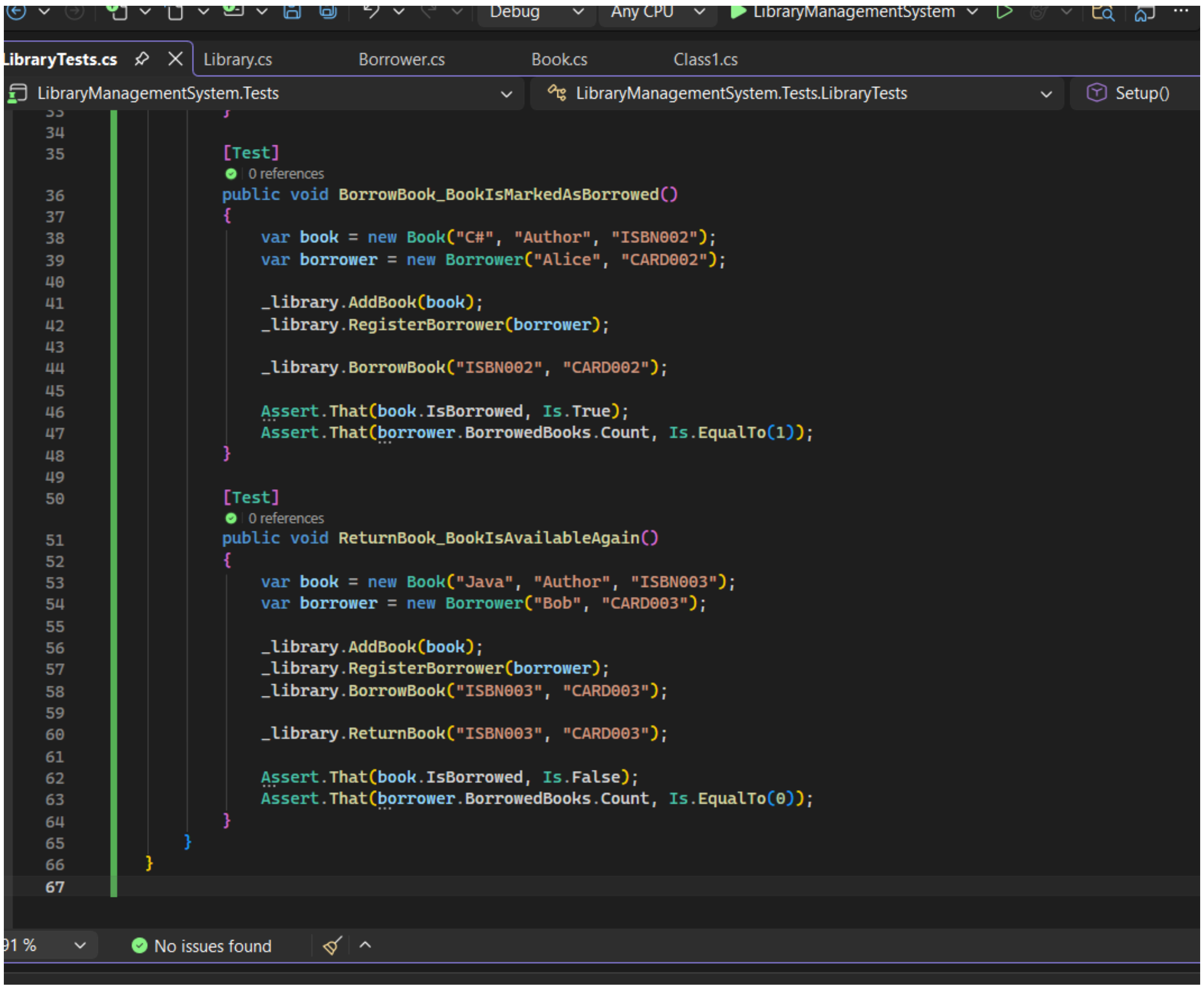
1. Right-click Solution → Add → New Project
2. Select NUnit Test Project
3. Project Name: LibraryManagementSystem.Tests

STEP 4: Add Reference to Class Library

1. Right-click Test Project
2. Click Add Project Reference
3. Select LibraryManagementSystem


File Edit View Git Project Build Debug Test Tools Extensions Window Help Search
LibraryManagementSystem
Debug
Any CPU
LibraryManagementSystem
LibraryTests.cs
Library.cs
Borrower.cs
Book.cs
Class1.cs
LibraryManagementSystem.Tests
LibraryManagementSystem.Tests.LibraryTests
Setup()
91 %
No issues found
Output

```csharp
using NUnit.Framework;
using LibraryManagementSystem;

namespace LibraryManagementSystem.Tests
{
    [TestFixture]
    0 references
    public class LibraryTests
    {
        private Library _library;

        [SetUp]
        0 references
        public void Setup()
        {
            _library = new Library();
        }

        [Test]
        0 references
        public void AddBook_BookIsAddedSuccessfully()
        {
            var book = new Book("C# Basics", "Microsoft", "ISBN001");
            _library.AddBook(book);

            Assert.That(_library.Books.Count, Is.EqualTo(1));
        }

        [Test]
        0 references
        public void RegisterBorrower_BorrowerIsRegistered()
        {
            var borrower = new Borrower("John", "CARD001");
            _library.RegisterBorrower(borrower);

            Assert.That(_library.Borrowers.Count, Is.EqualTo(1));
        }
```


```
        }

        [Test]
        0 references
        public void BorrowBook_BookIsMarkedAsBorrowed()
        {
            var book = new Book("C#", "Author", "ISBN002");
            var borrower = new Borrower("Alice", "CARD002");

            _library.AddBook(book);
            _library.RegisterBorrower(borrower);

            _library.BorrowBook("ISBN002", "CARD002");

            Assert.That(book.IsBorrowed, Is.True);
            Assert.That(borrower.BorrowedBooks.Count, Is.EqualTo(1));
        }

        [Test]
        0 references
        public void ReturnBook_BookIsAvailableAgain()
        {
            var book = new Book("Java", "Author", "ISBN003");
            var borrower = new Borrower("Bob", "CARD003");

            _library.AddBook(book);
            _library.RegisterBorrower(borrower);
            _library.BorrowBook("ISBN003", "CARD003");

            _library.ReturnBook("ISBN003", "CARD003");

            Assert.That(book.IsBorrowed, Is.False);
            Assert.That(borrower.BorrowedBooks.Count, Is.EqualTo(0));
        }
    }
}
```

LibraryTests. cs Code

```
using NUnit.Framework;

using LibraryManagementSystem;

namespace LibraryManagementSystem.Tests

{

    [TestFixture]

    public class LibraryTests

    {

        private Library _library;
```

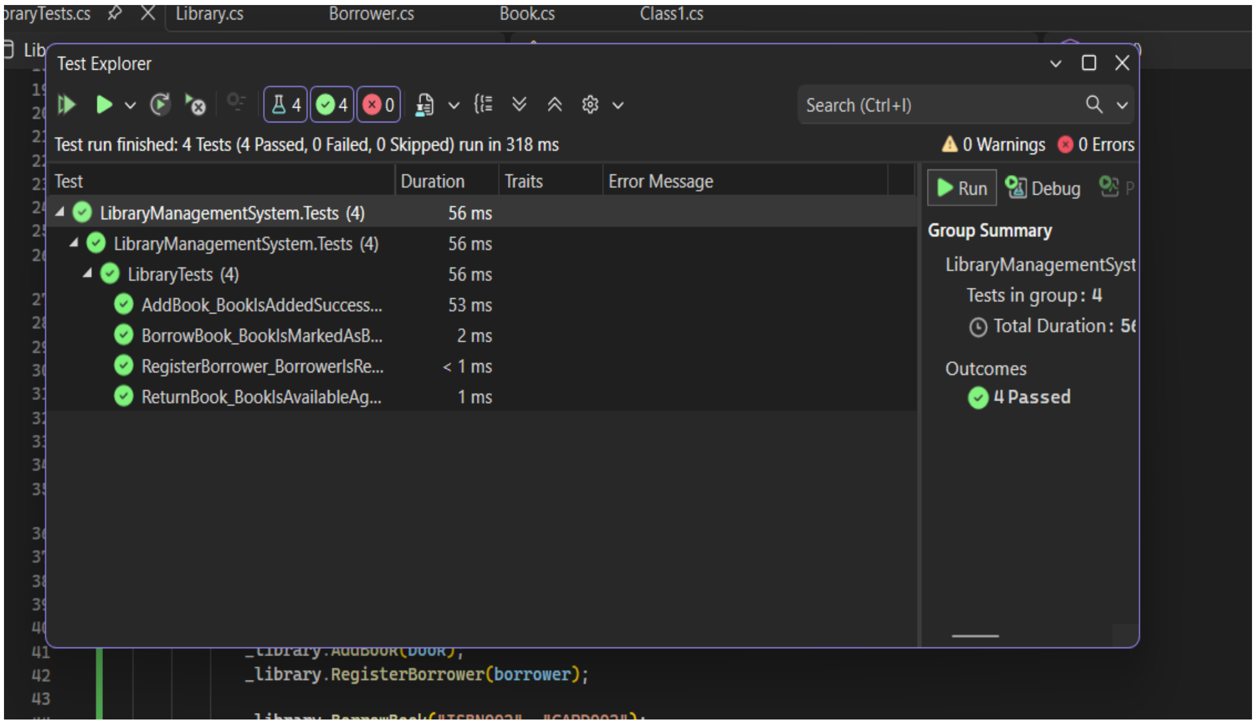
```
[SetUp]
public void Setup()
{
    _library = new Library();
}

[Test]
public void AddBook_BookIsAddedSuccessfully()
{
    var book = new Book("C# Basics", "Microsoft", "ISBN001");
    _library.AddBook(book);

    Assert.That(_library.Books.Count, Is.EqualTo(1));
}

[Test]
public void RegisterBorrower_BorrowerIsRegistered()
{
    var borrower = new Borrower("John", "CARD001");
    _library.RegisterBorrower(borrower);

    Assert.That(_library.Borrowers.Count, Is.EqualTo(1));
}

[Test]
```

```
public void BorrowBook_BookIsMarkedAsBorrowed()
{
    var book = new Book("C#", "Author", "ISBN002");
    var borrower = new Borrower("Alice", "CARD002");

    _library.AddBook(book);
    _library.RegisterBorrower(borrower);

    _library.BorrowBook("ISBN002", "CARD002");

    Assert.That(book.IsBorrowed, Is.True);
    Assert.That(borrower.BorrowedBooks.Count, Is.EqualTo(1));
}

[Test]
public void ReturnBook_BookIsAvailableAgain()
{
    var book = new Book("Java", "Author", "ISBN003");
    var borrower = new Borrower("Bob", "CARD003");

    _library.AddBook(book);
    _library.RegisterBorrower(borrower);
    _library.BorrowBook("ISBN003", "CARD003");

    _library.ReturnBook("ISBN003", "CARD003");
```

```
            Assert.That(book.IsBorrowed, Is.False);

            Assert.That(borrower.BorrowedBooks.Count, Is.EqualTo(0));

        }

    }

}
```

STEP 5: Run Unit Tests

1. Open Test Test Explorer
2. Click Run All



Test run finished: 4 Tests (4 Passed, 0 Failed, 0 Skipped) run in 318 ms

0 Warnings 0 Errors

Test	Duration	Traits	Error Message
LibraryManagementSystem.Tests (4)	56 ms		
LibraryManagementSystem.Tests (4)	56 ms		
LibraryTests (4)	56 ms		
AddBook_BookIsAddedSuccess...	53 ms		
BorrowBook_BookIsMarkedAsB...	2 ms		
RegisterBorrower_BorrowerIsRe...	< 1 ms		
ReturnBook_BookIsAvailableAg...	1 ms		

Run Debug

Group Summary

LibraryManagementSyst

Tests in group: 4

Total Duration: 56

Outcomes

4 Passed

```
_library.AddBook(book);
_library.RegisterBorrower(borrower);
```