Ministry of Higher Education and Scientific Research

Mustansiriyah University

College of Science

Department of Computer Science

By

Hawra Hussein Abdel

Farah Abdel Salam

First Stage (B1)

Library Catalog System Using C#

Introduction

In today's fast-paced world, technology has become an integral part of our daily lives, simplifying and automating traditional processes. The "Library Catalog System" project is a simple application developed using the C# programming language within a console-based environment. This project aims to manage books in a library, allowing users to add books, search for them, check their availability, and display all books with ease.

This project serves as a practical example of implementing Object-Oriented Programming (OOP) concepts and demonstrates the basic principles of software development for small-scale applications.

Objectives of the Project

1. Provide a basic system for managing books in a library.
2. Teach essential Object-Oriented Programming (OOP) concepts such as classes, objects, and collections.
3. Implement efficient methods for searching and displaying data.
4. Enhance understanding of software project structures using C# and Console applications.

Project Description

The project is a console-based program where users interact with the system by entering specific commands. The system includes the following functionalities:

1. Add a New Book: Users can input the title, author, and availability status of a book to add it to the library.
2. Search for a Book: Users can search for books by title or author, and the system will display matching results.
3. Display All Books: All books stored in the library are displayed in an organized format.
4. Check Book Availability: The system allows users to verify if a specific book is available.

5. Exit the Program: The application terminates when the user selects the exit command.

Core Functionalities

1. **Add a Book:**
    - Users input the book's details such as title, author, and availability.
    - The book is stored as an object in a list.
2. **Search for a Book:**
    - The user enters a query (title or author), and the system searches for matching entries in the book list.
    - If the book is found, its details are displayed; otherwise, an appropriate message is shown.
3. **Display All Books:**
    - The system lists all books stored in the library, displaying their titles, authors, and availability.
    - If no books are present, a message notifies the user.
4. **Check Availability:**
    - Users can check whether a specific book is marked as available.
5. **Exit**:
    - Ends the program with a farewell message.

System Structure

The project comprises three main classes:

1. **The Book Class:**
    - **Represents individual books in the system.**
    - **Contains the following attributes:**
        - **Title (Book title)**
        - **Author (Book author)**
        - **IsAvailable (Availability status)**

csharp

copy code

```
public class Book
{
    public string Title { get; set; }
    public string Author { get; set; }
    public bool IsAvailable { get; set; }
}
```

2. **The Library Class:**
    - **Manages a collection of books and provides methods for interacting with the data.**
    - **Includes methods like:**
        - **Adding a book (AddBook)**
        - **Searching for a book (SearchBook)**
        - **Displaying all books (DisplayAllBooks)**
        - **Checking book availability (CheckAvailability)**

csharp

Copy code

```
public class Library
{
    private List<Book> books = new List<Book>();

    public void AddBook(Book book) { ... }
    public Book SearchBook(string query) { ... }
    public void DisplayAllBooks() { ... }
    public bool CheckAvailability(string title) { ... }
}
```

3. **The Program Class:**
    - **Acts as the entry point of the application.**

- **Manages the user interface and connects user commands to the library's functionality.**

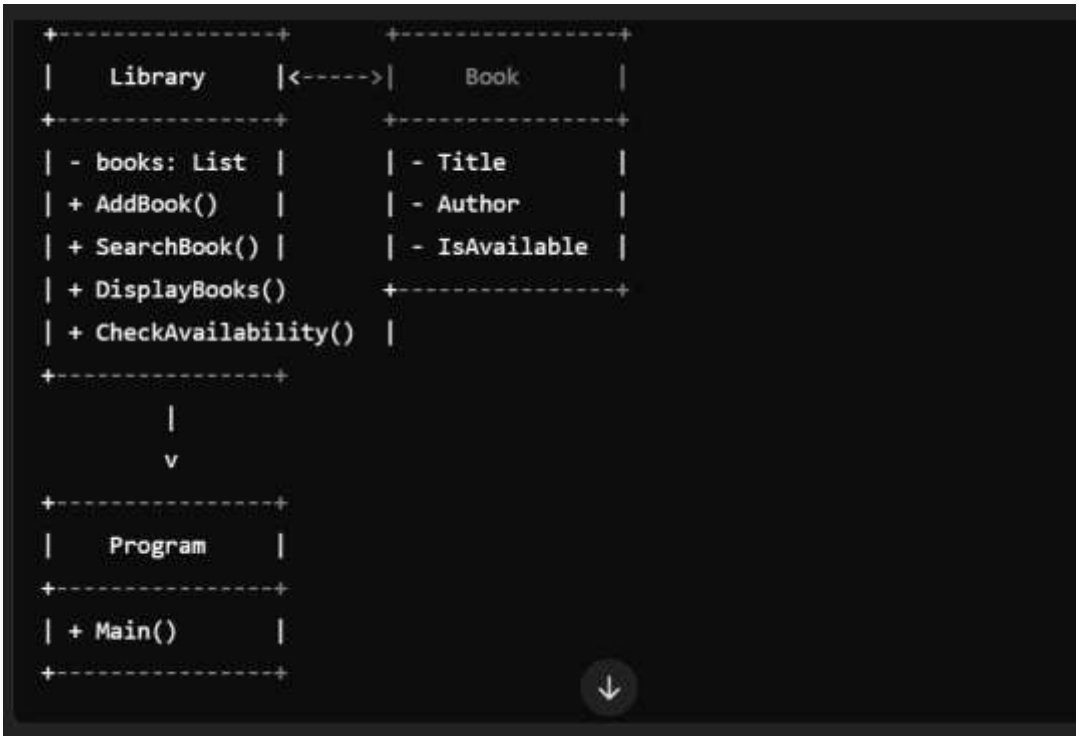
csharp

Copy code

```
class Program
{
    static void Main(string[] args)
    {
        Library library = new Library();
        // Code for user interaction goes here.
    }
}
```

UML Diagram

The following UML Class Diagram represents the relationships and structure of the project:



Challenges and Opportunities

Challenges:

1. Validating user input to ensure data integrity.
2. Handling errors to prevent unexpected crashes.
3. Optimizing the search functionality when dealing with large datasets.

Opportunities:

1. Expanding the system to include features like book deletion or data modification.
2. Integrating the application with a database (e.g., SQL) for persistent data storage.
3. Upgrading the console-based interface to a graphical user interface (GUI).

Summary

The "Library Catalog System" project is a simple yet effective application designed to organize and manage library books using C#. It features a user-friendly interface and demonstrates the foundational principles of Object-Oriented Programming (OOP). With further improvements, this project can evolve into a comprehensive library management system.

Code

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

class Program
{
    static void Main(string[] args)
    {
        Library library = new Library();
        int choice;

        do
        {
            Console.WriteLine("\n--- Library Catalogue ---");
            Console.WriteLine("1. Add a Book");
            Console.WriteLine("2. Search for a Book");
            Console.WriteLine("3. Display All Books");
            Console.WriteLine("4. Check Availability");
            Console.WriteLine("5. Exit");
            Console.Write("Enter your choice: ");
            choice = int.Parse(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    library.AddBook();
                    break;
                case 2:
                    library.SearchBook();
                    break;
                case 3:
```

```
                library.DisplayBooks();
                break;
            case 4:
                library.CheckAvailability();
                break;
            case 5:
                Console.WriteLine("Exiting... Goodbye!");
                break;
            default:
                Console.WriteLine("Invalid choice. Please try
again.");
                break;
        }
    } while (choice != 5);
  }
}

class Book
{
  public string Title { get; set; }
  public string Author { get; set; }
  public bool IsAvailable { get; set; }

  public Book(string title, string author, bool isAvailable)
  {
    Title = title;
    Author = author;
    IsAvailable = isAvailable;
  }
}
```

```
class Library
{
    private List<Book> books = new List<Book>();

    public void AddBook()
    {
        Console.Write("Enter the book title: ");
        string title = Console.ReadLine();

        Console.Write("Enter the book author: ");
        string author = Console.ReadLine();

        Console.Write("Is the book available? (yes/no): ");
        bool isAvailable = Console.ReadLine().ToLower() ==
"yes";

        books.Add(new Book(title, author, isAvailable));
        Console.WriteLine("Book added successfully!");
    }

    public void SearchBook()
    {
        Console.Write("Enter the title or author to search: ");
        string query = Console.ReadLine().ToLower();

        foreach (var book in books)
        {
            if (book.Title.ToLower().Contains(query) ||
book.Author.ToLower().Contains(query))
            {
```

```
            Console.WriteLine($"Found: {book.Title} by
{book.Author} - {(book.IsAvailable ? "Available" : "Not
Available")}");
            return;
        }
    }
    Console.WriteLine("Book not found.");
}

public void DisplayBooks()
{
    if (books.Count == 0)
    {
        Console.WriteLine("No books in the library.");
        return;
    }

    Console.WriteLine("\n--- Library Books ---");
    foreach (var book in books)
    {
        Console.WriteLine($"{book.Title} by
{book.Author} - {(book.IsAvailable ? "Available" : "Not
Available")}");
    }
}

public void CheckAvailability()
{
    Console.Write("Enter the book title to check
availability: ");
    string title = Console.ReadLine().ToLower();
```

```
        foreach (var book in books)
        {
            if (book.Title.ToLower() == title)
            {
                Console.WriteLine(book.IsAvailable ? "The book is available." : "The book is not available.");
                return;
            }
        }
        Console.WriteLine("Book not found.");
    }
}
```