## SOFTWARE ENGINEERING I

## Library Management System(Loaning)

**Supervised By:**

1. Dr. Marwa Nabil
2. T.A: Mina Essam
3. T.A: Reham Fares

**Submitted By:**

1.Mahmoud Mohamed Samy Abdallah (G10) CS

(Use Case, Class Diagram, Design & Implementation, ProtoType, Testing)

2.Muhamed Abuelkhir Abdelslam Muhamed (G7) CS

(Use Case, Sequential Diagram, ProtoType, Testing)

3.Mahmoud Mohamed Abdallah abdel moaty (G10) CS

(Use Case, Class Diagram, ProtoType, Testing)

4.Mohamed Mahmoud Sedik Sayed (G9) CS

(Use Case, Activity Diagram, ProtoType, Testing)

5.Karim Roshdy Mohamed Ghars El Deen (G3) IS

(Use Case , Sequential Diagram, ProtoType, Testing)

6. Mohamed Nasser Ali Ahmed (G4) IS

(Use Case, Class Diagram, ProtoType, Testing)

## Table Of Contents:

## 1.1Introduction

**Library Administration Software is essential in all schools and colleges; without it, no educational institution could survive. It is a vital feature of any school and college library, and it assists the librarian in keeping track of both accessible and issued volumes. Library Management System software assists students in a variety of ways by allowing them to study, gather resources, facilitate group learning, and enhance their knowledge and abilities.**

## 1.2 HOW DOES APPLICATION WORKS

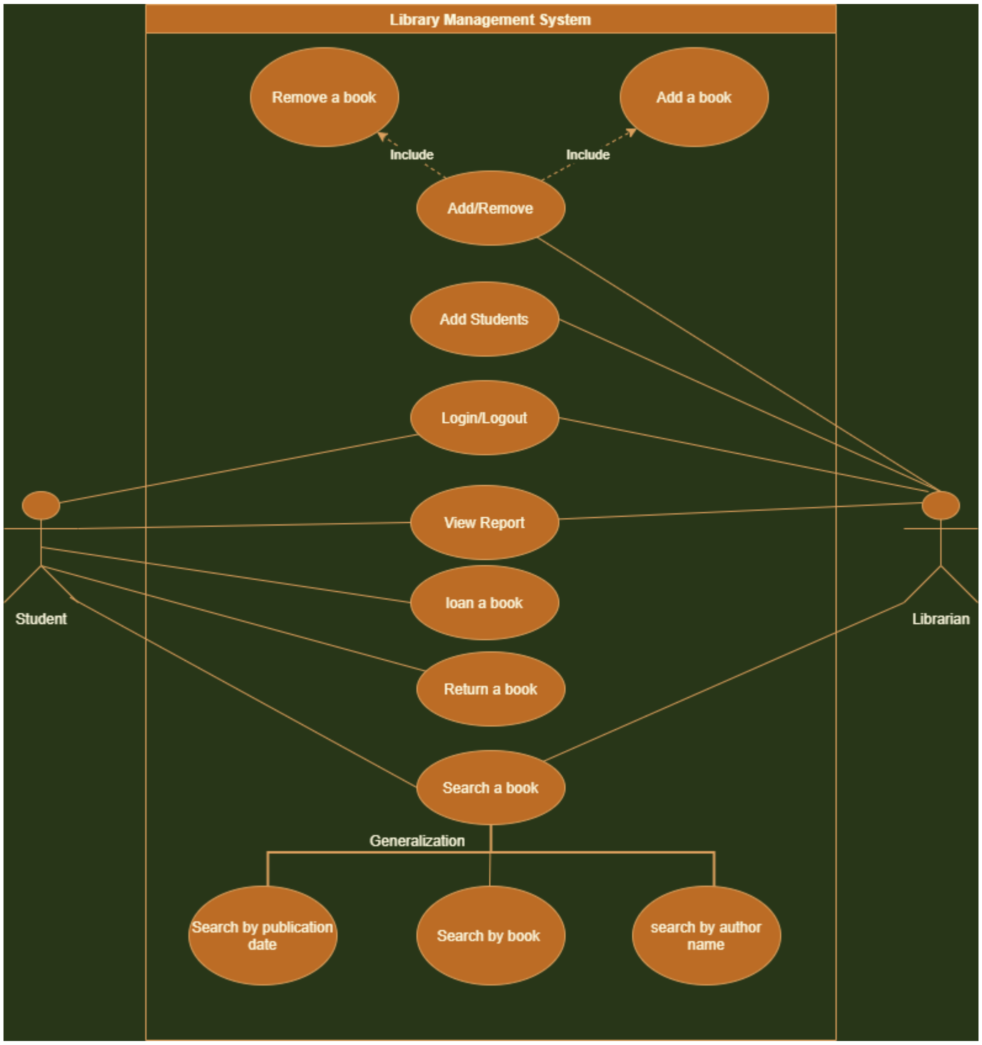
**First of all that’s a representation for our Library management system which allow students to loan and return books and search for books via signing in while the admin is responsible of adding any new students to the library , student can view his report which contains the username and password to memorize them to sign in ,admin can search books too and he can manage books and its’ categories he can add or remove students too and can view his report which contains the username and password to memorize them to sign in**

## 2.Use Case Diagram

### Use Case Diagram Definition

**It's interactions between a system and it's environment. Which used to support requirements elicitation and gives a simple overview of an interaction. Shown as an ellipse with the actors involved in the use case.**

### Design of Use Case:

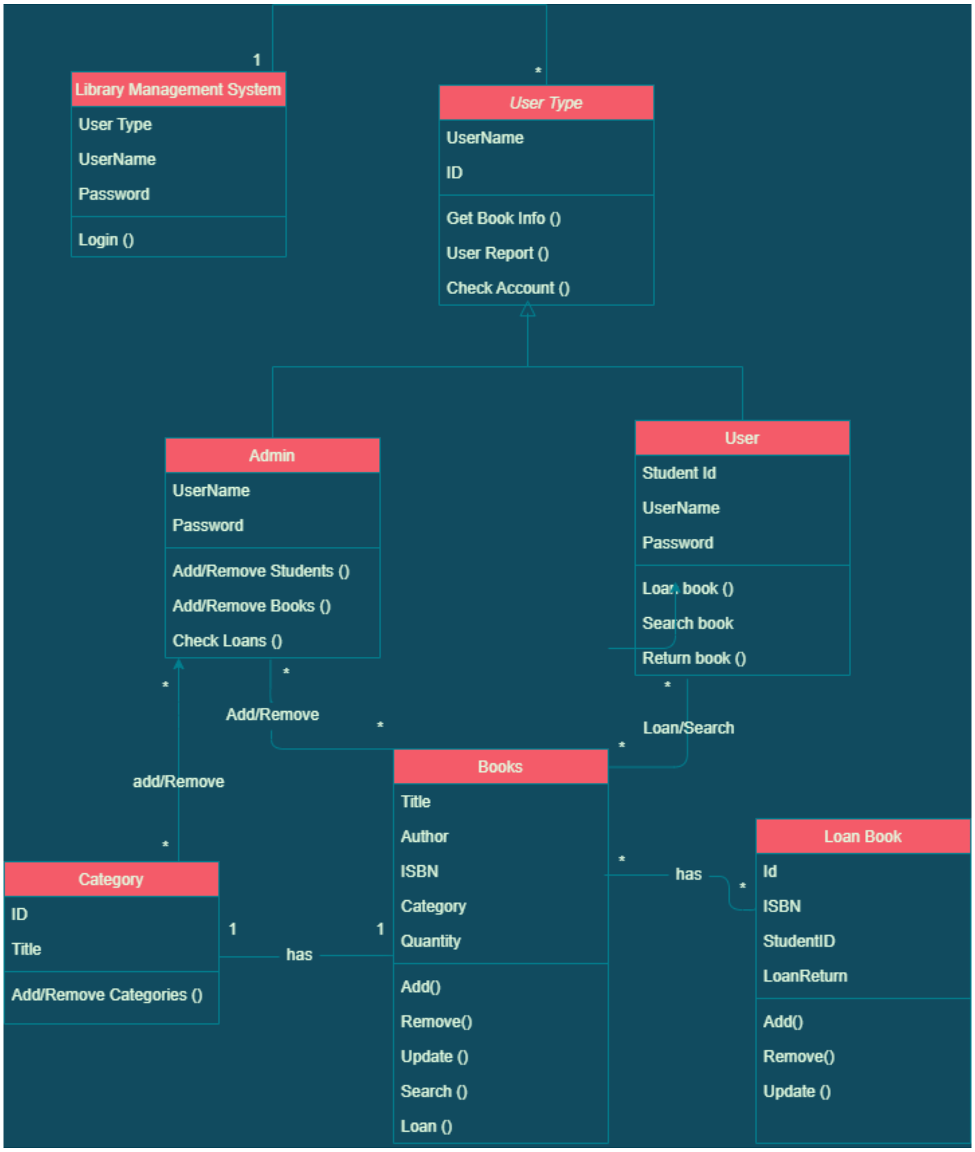


# 3.Class Diagram

## Class Diagram Definition

**It’s an object classes in the system and the associations between these classes, Which link between classes that indicates that there is a relationship using attributes. Each class may have some knowledge of it’s associated class and expressed at different levels of details.**

## Class Diagram Design

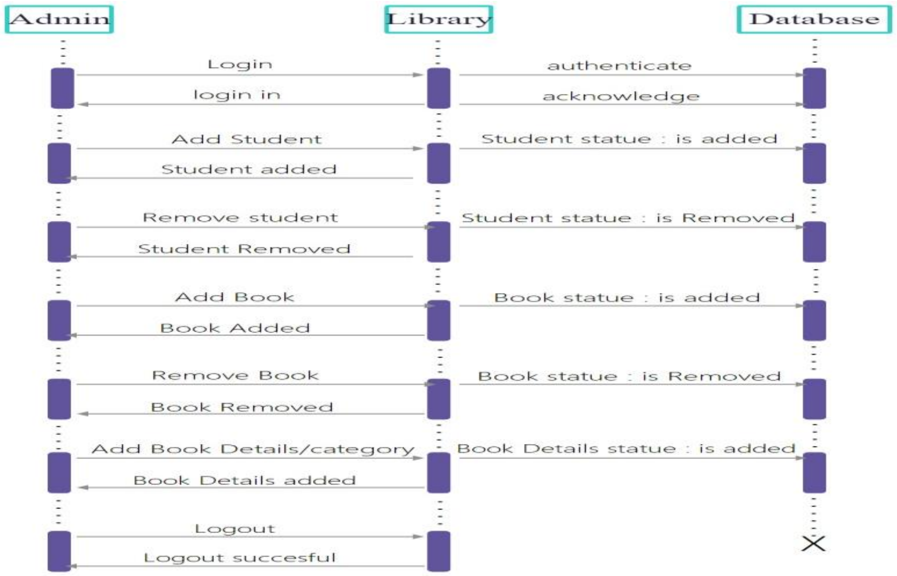


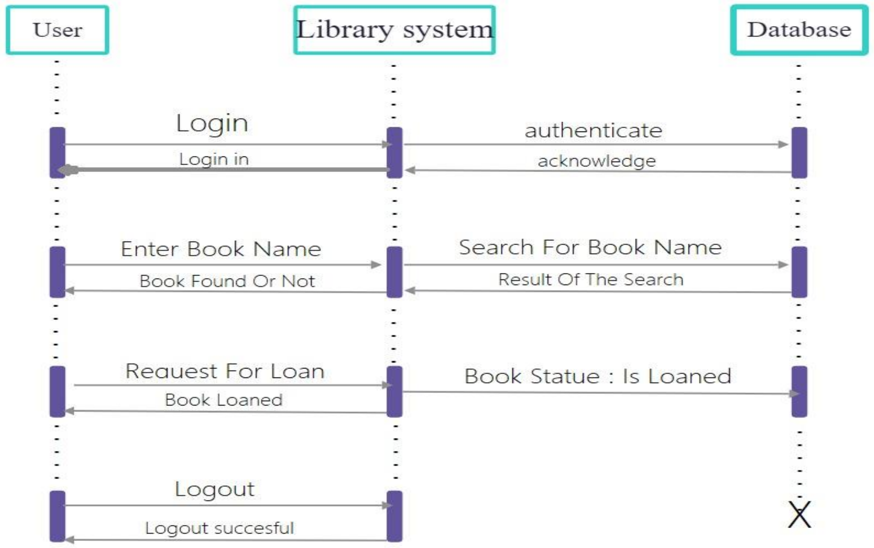
# 4.Sequence Diagram

## Sequence Diagram Definition

**It’s interactions between actors and the system and between system component, That shows the sequence of interactions that take place during a particular use case. Objects and actors involved are listed along the top of the diagram, Interactions between objects are indicated by annotated arrows.**

## Sequence Diagram Design





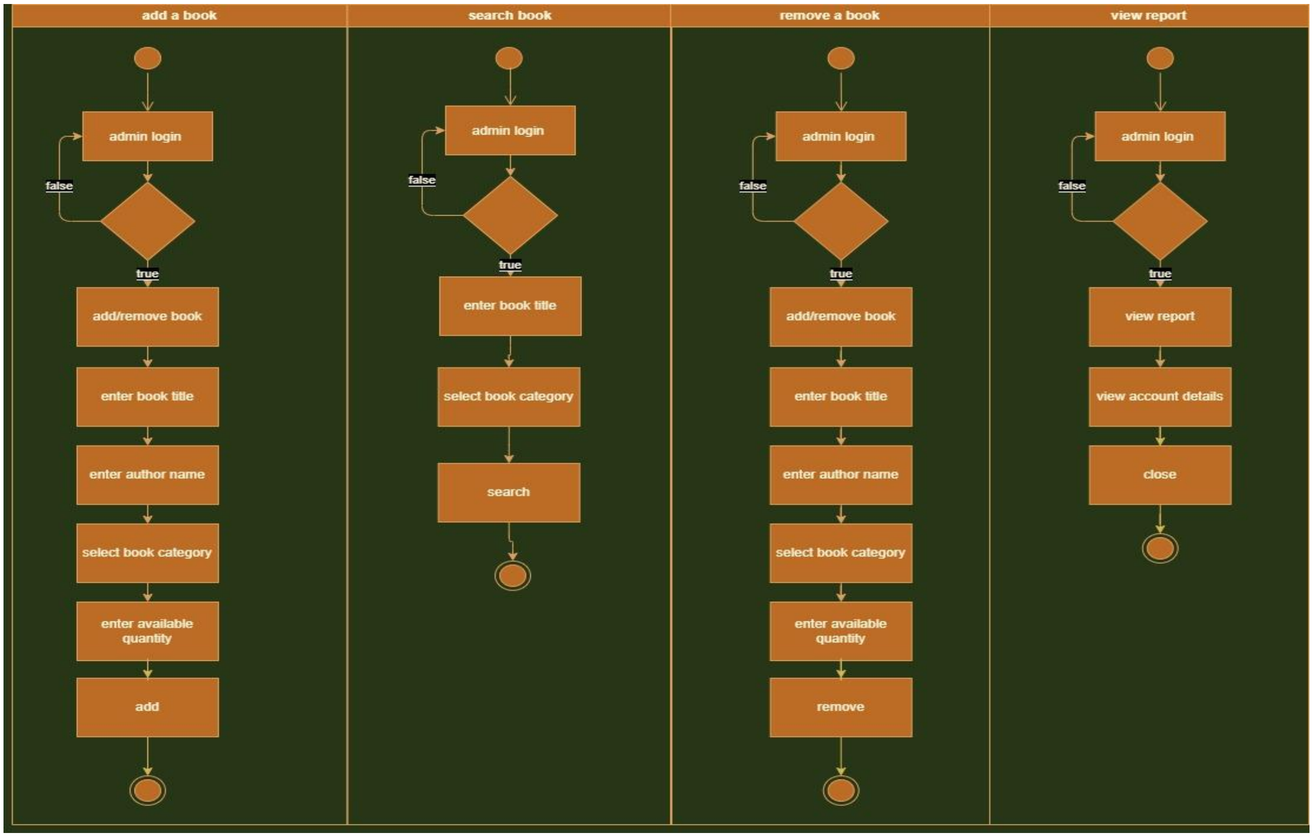
# 5.Activity Diagram

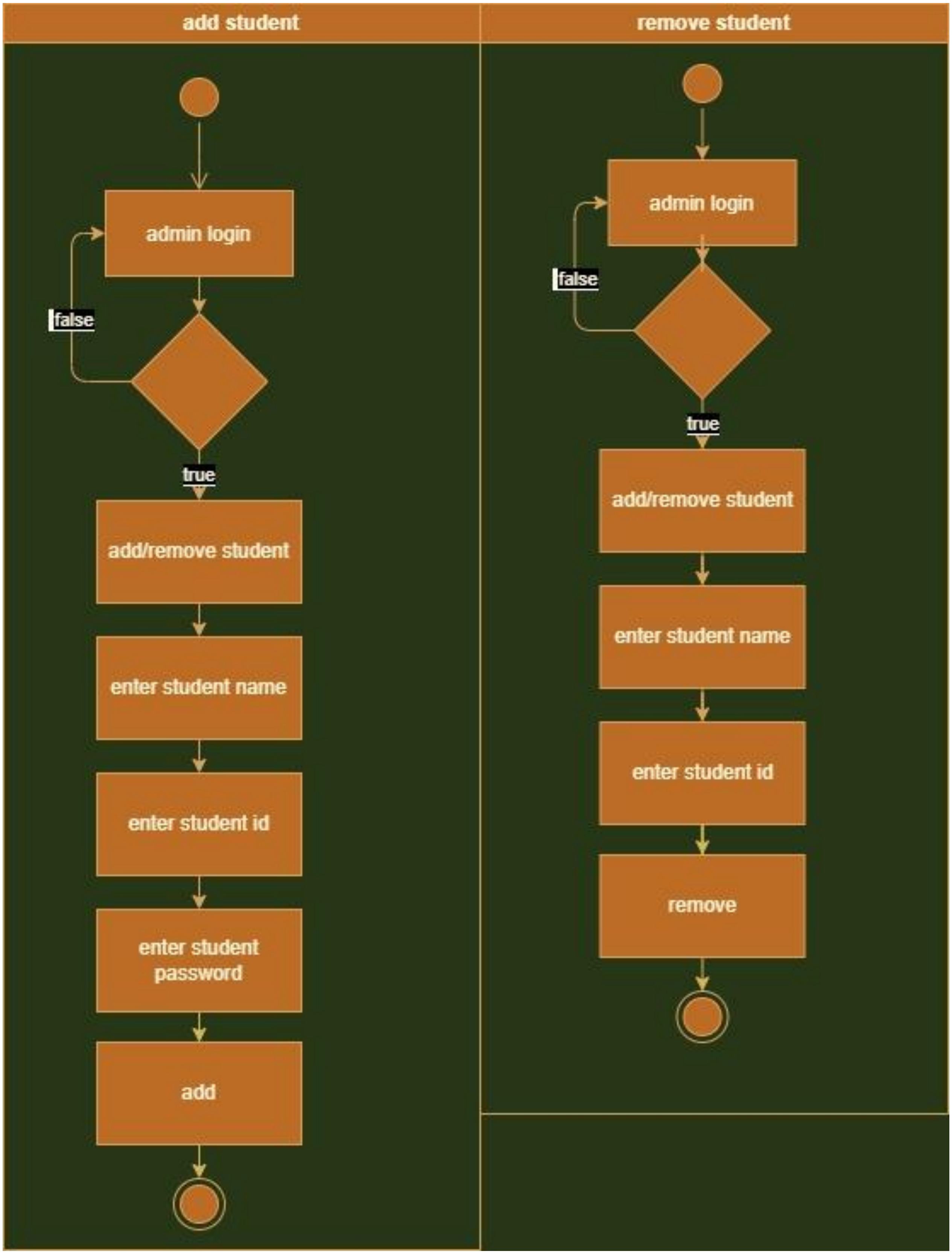
## Activity Diagram Definition

**It's the activities involved in a process or in data processing. It's portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. A UML activity diagram represents the actions and decisions that occur as some function is performed.**

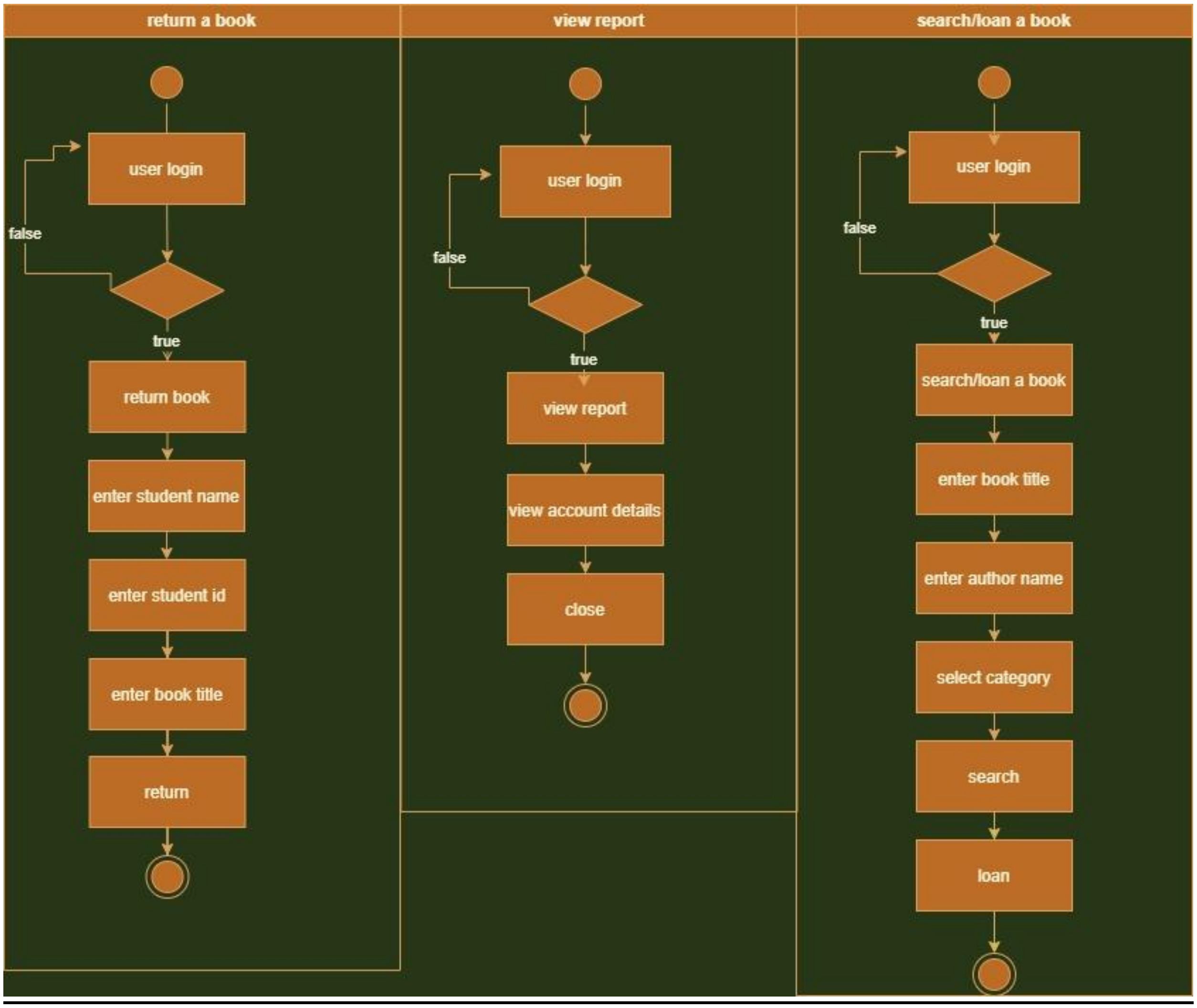
## Activity Diagram Design

## Admin Activities:

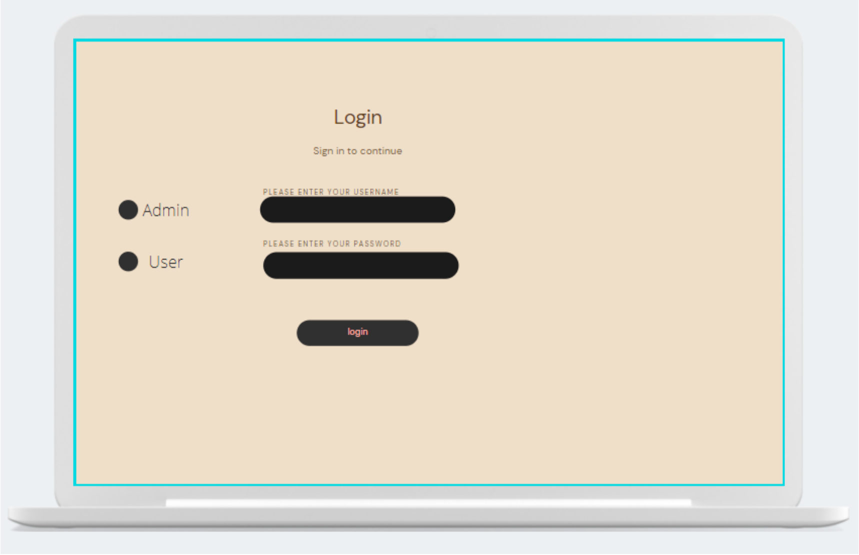


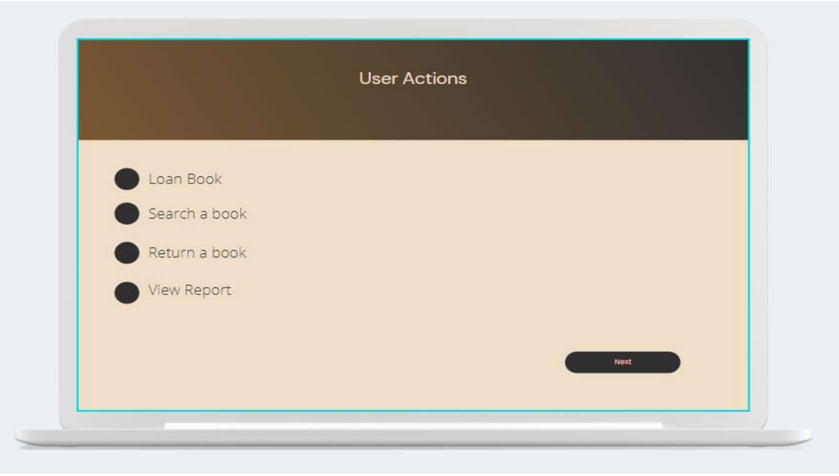

add student
remove student
admin login
false
true
add/remove student
enter student name
enter student id
enter student password
add
admin login
false
true
add/remove student
enter student name
enter student id
remove

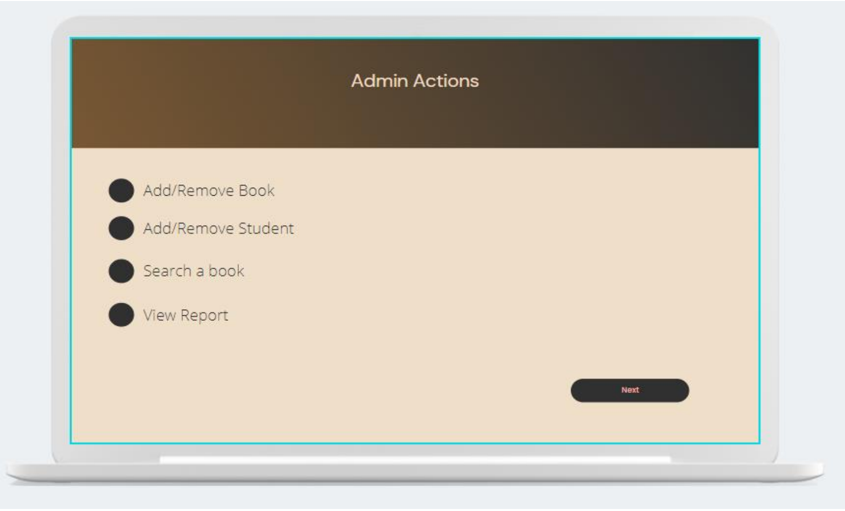
## User Activities:



# 6.Prototype

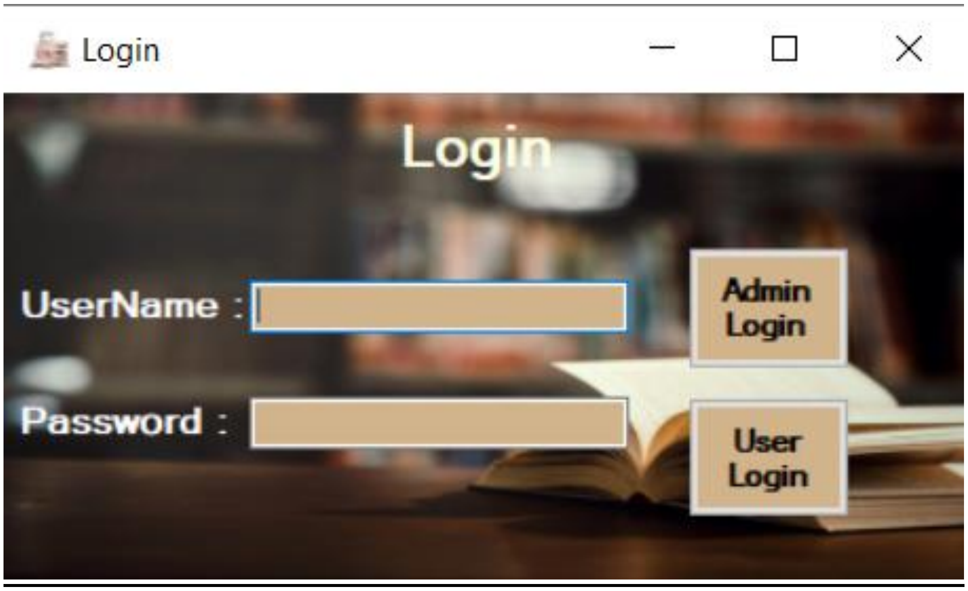

Login
Sign in to continue
● Admin
● User
PLEASE ENTER YOUR USERNAME
PLEASE ENTER YOUR PASSWORD
login


User Actions
● Loan Book
● Search a book
● Return a book
● View Report
Next

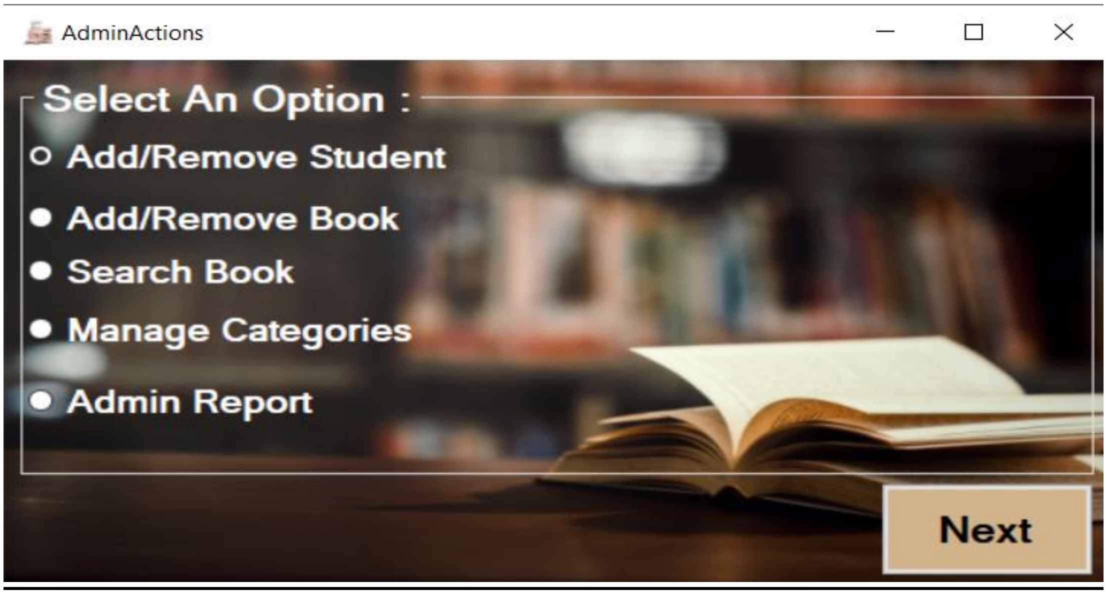

Admin Actions
● Add/Remove Book
● Add/Remove Student
● Search a book
● View Report
Next

## 7. Interface Design:

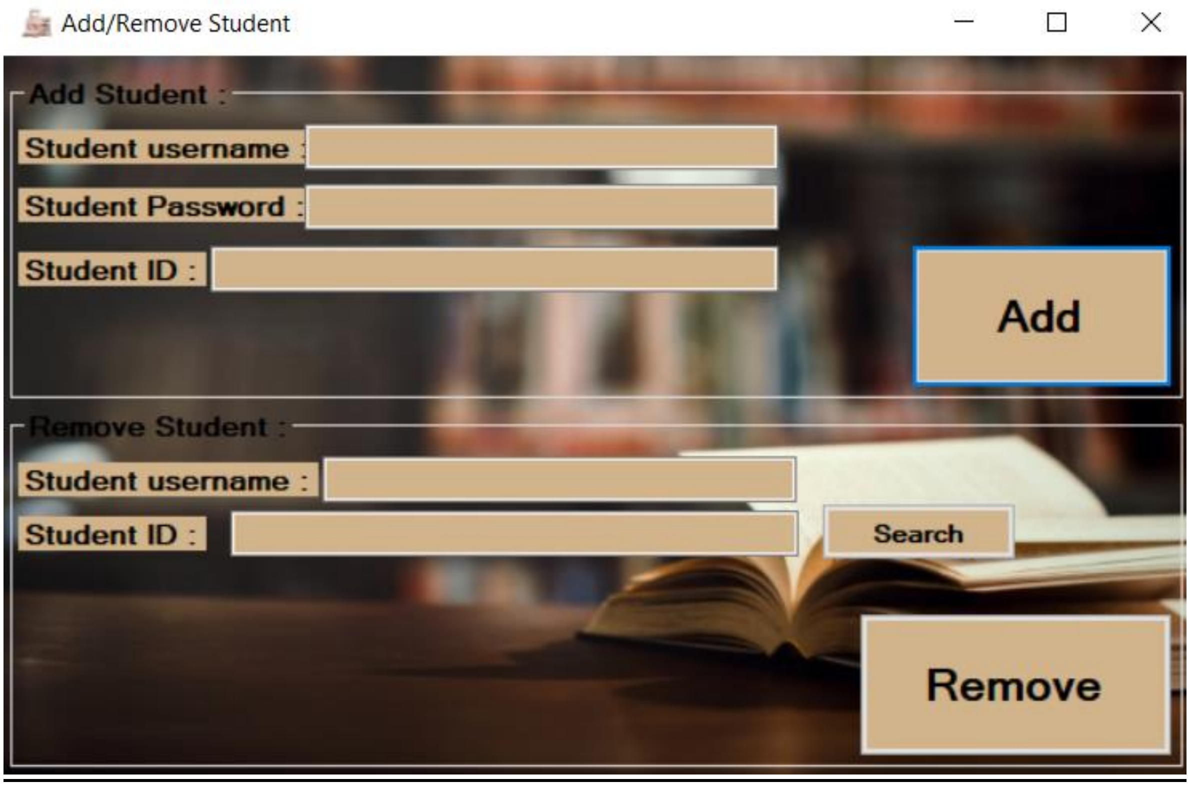
### 1.Login



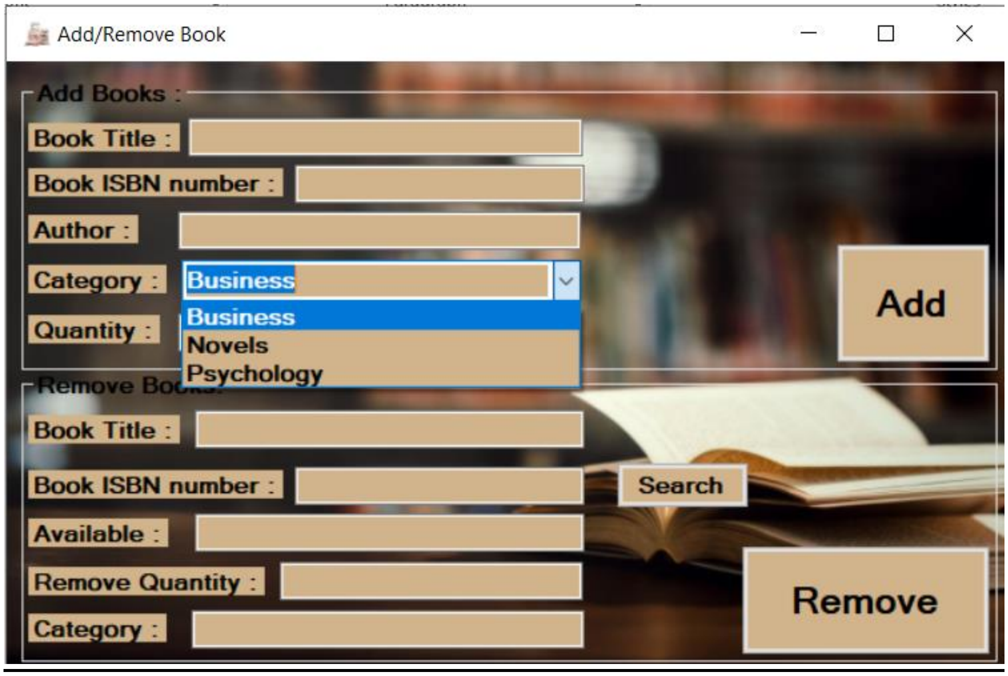
### 2.Admin Actions



## Add/Remove Student

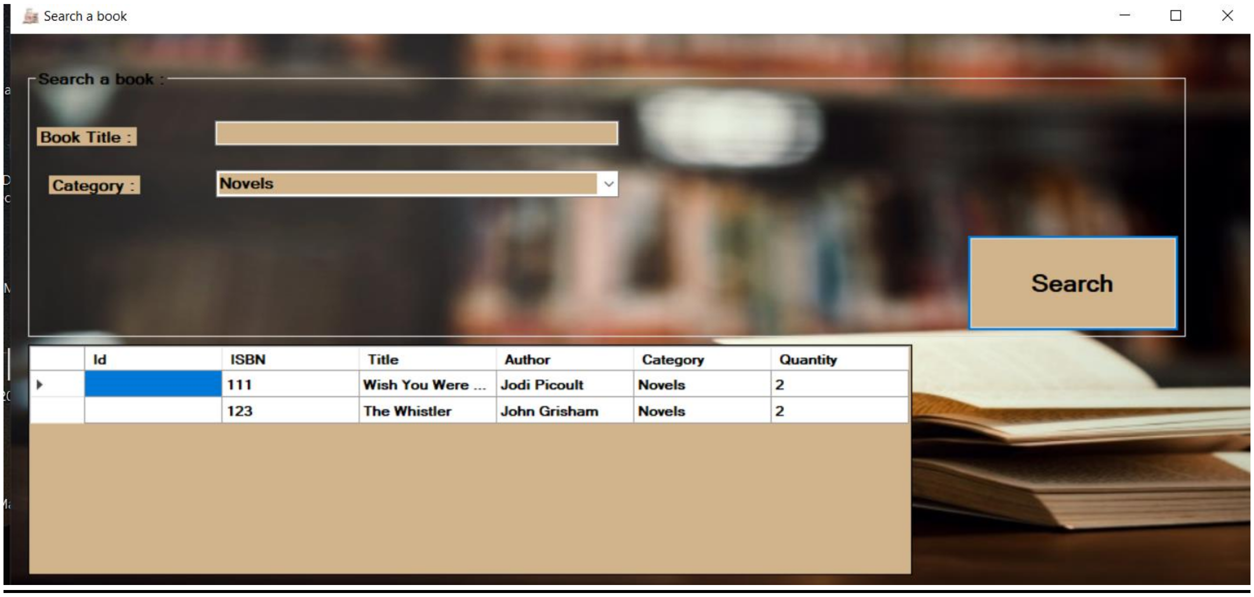


## Add/Remove Books



Category : Business

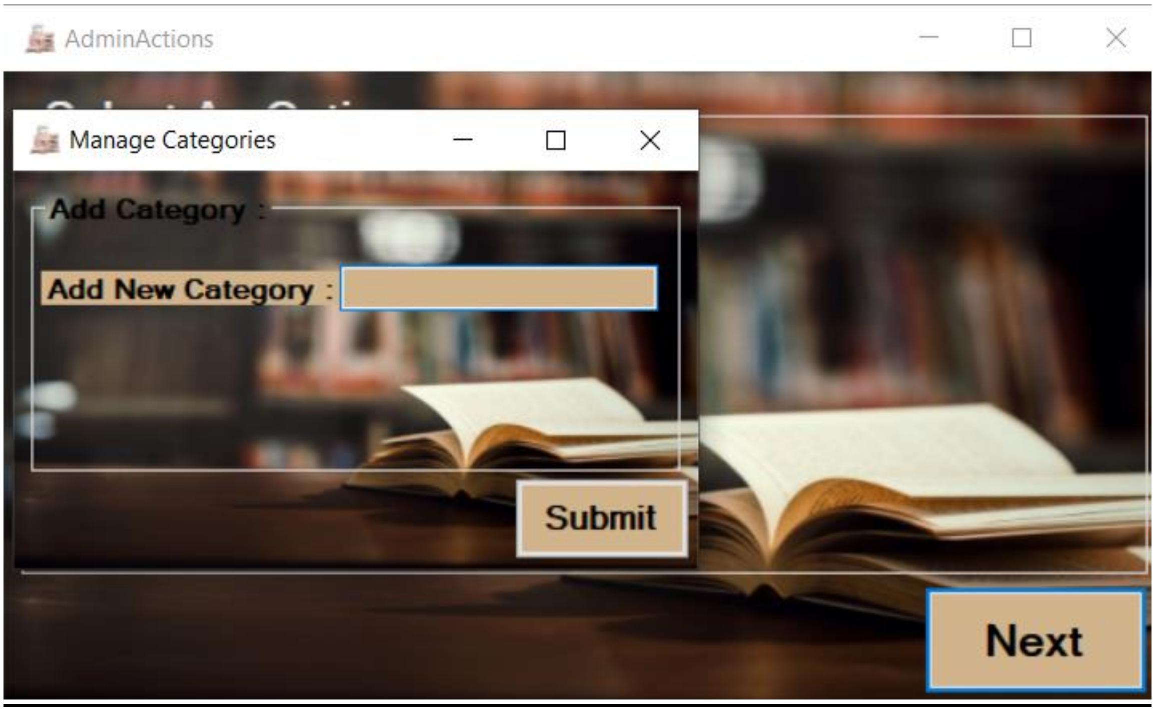
## Search a Book



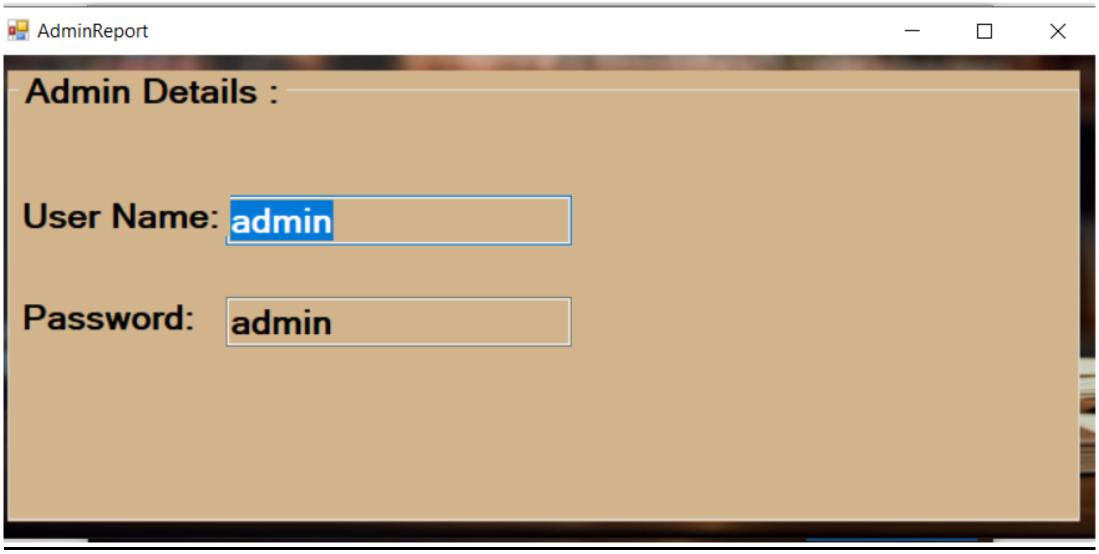
Category : Novels

| Id | ISBN | Title | Author | Category | Quantity |
|---|---|---|---|---|---|
| | 111 | Wish You Were ... | Jodi Picoult | Novels | 2 |
| | 123 | The Whistler | John Grisham | Novels | 2 |

## Manage Categories



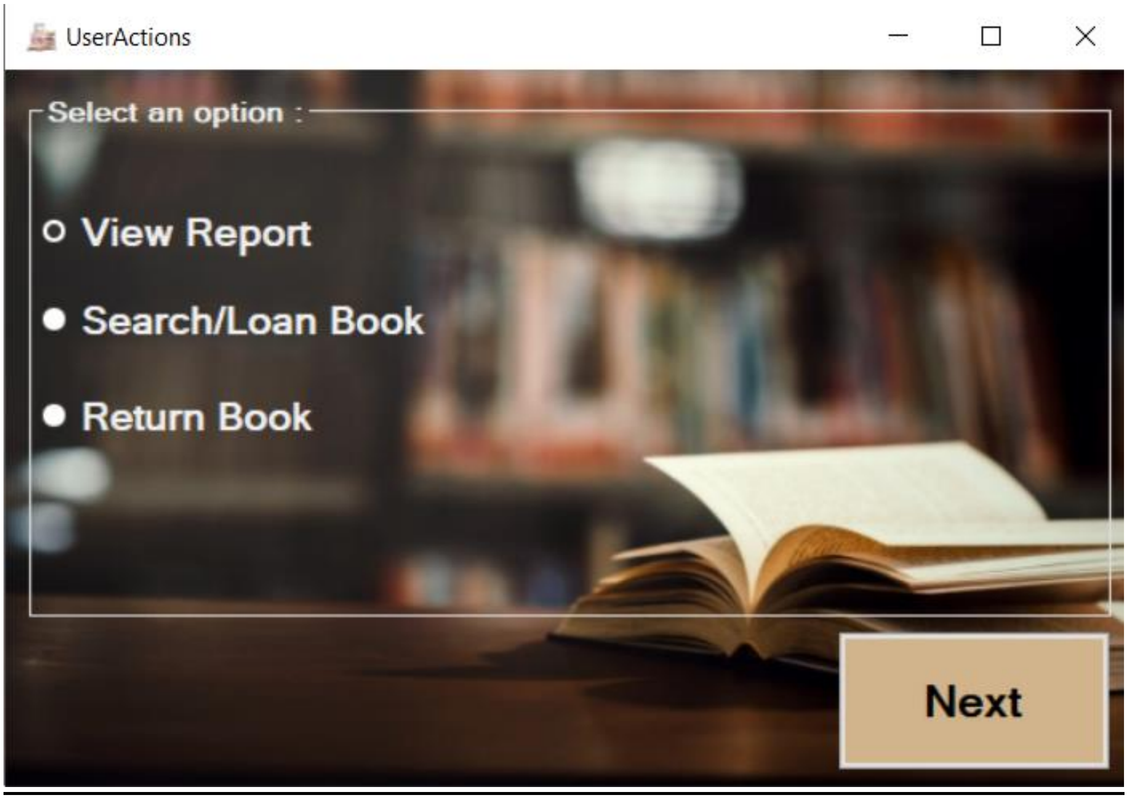
## Admin Report



User Name: admin

Password: admin

## 3.User Actions

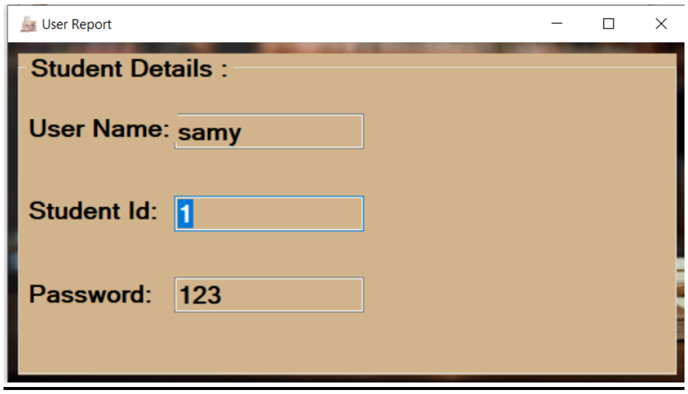


○ View Report

● Search/Loan Book

● Return Book

# View Report



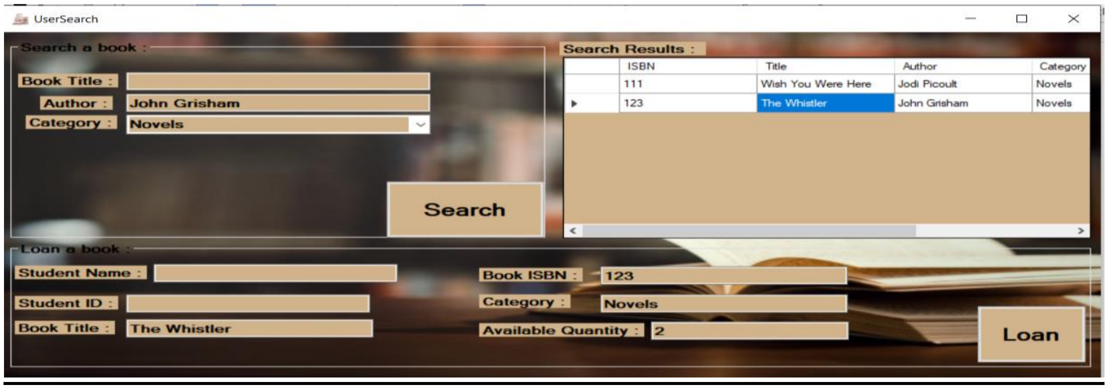
Student Details :

User Name: samy

Student Id: 1

Password: 123

# Search/Loan Book



Search a book :

Book Title :

Author : John Grisham

Category : Novels

Search Results :

| ISBN | Title | Author | Category |
|---|---|---|---|
| 111 | Wish You Were Here | Jodi Picoult | Novels |
| 123 | The Whistler | John Grisham | Novels |

Loan a book :

Student Name :

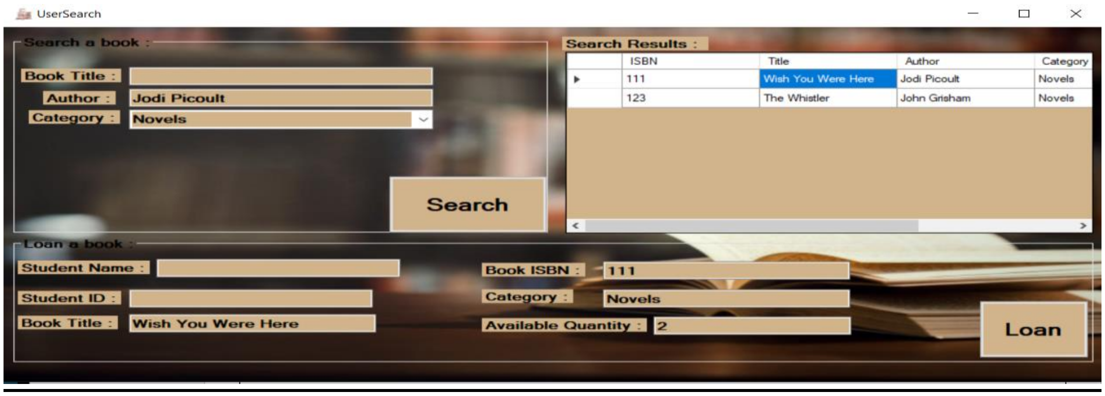
Student ID :

Book Title : The Whistler

Book ISBN : 123

Category : Novels

Available Quantity : 2



Search a book :

Book Title :

Author : Jodi Picoult

Category : Novels

Search Results :

| ISBN | Title | Author | Category |
|---|---|---|---|
| 111 | Wish You Were Here | Jodi Picoult | Novels |
| 123 | The Whistler | John Grisham | Novels |

Loan a book :

Student Name :

Student ID :

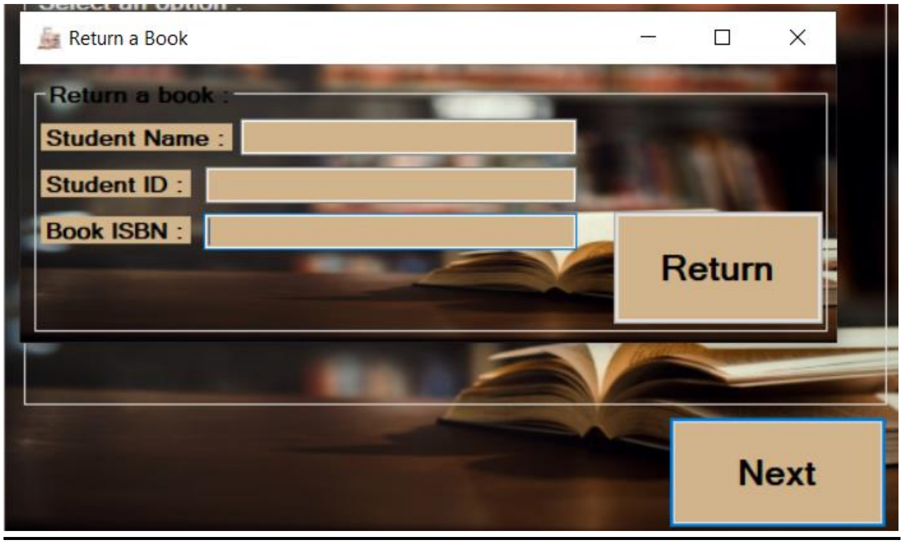
Book Title : Wish You Were Here

Book ISBN : 111

Category : Novels

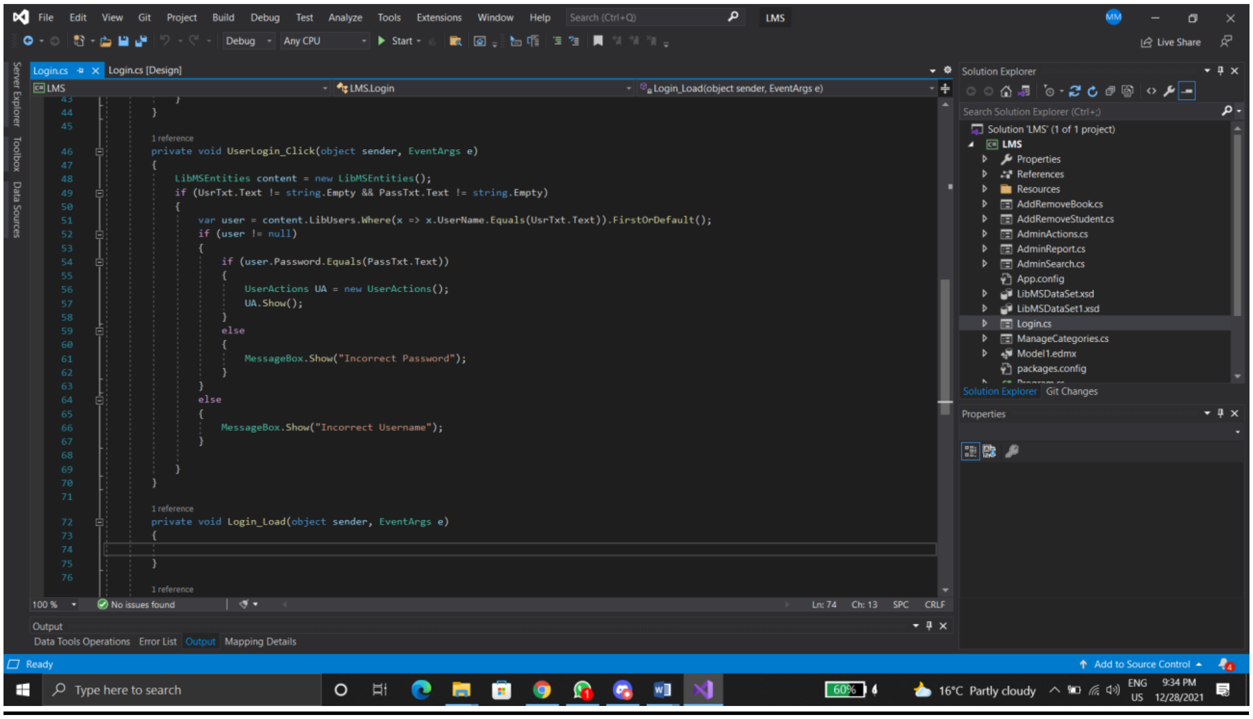
Available Quantity : 2

## Return a Book

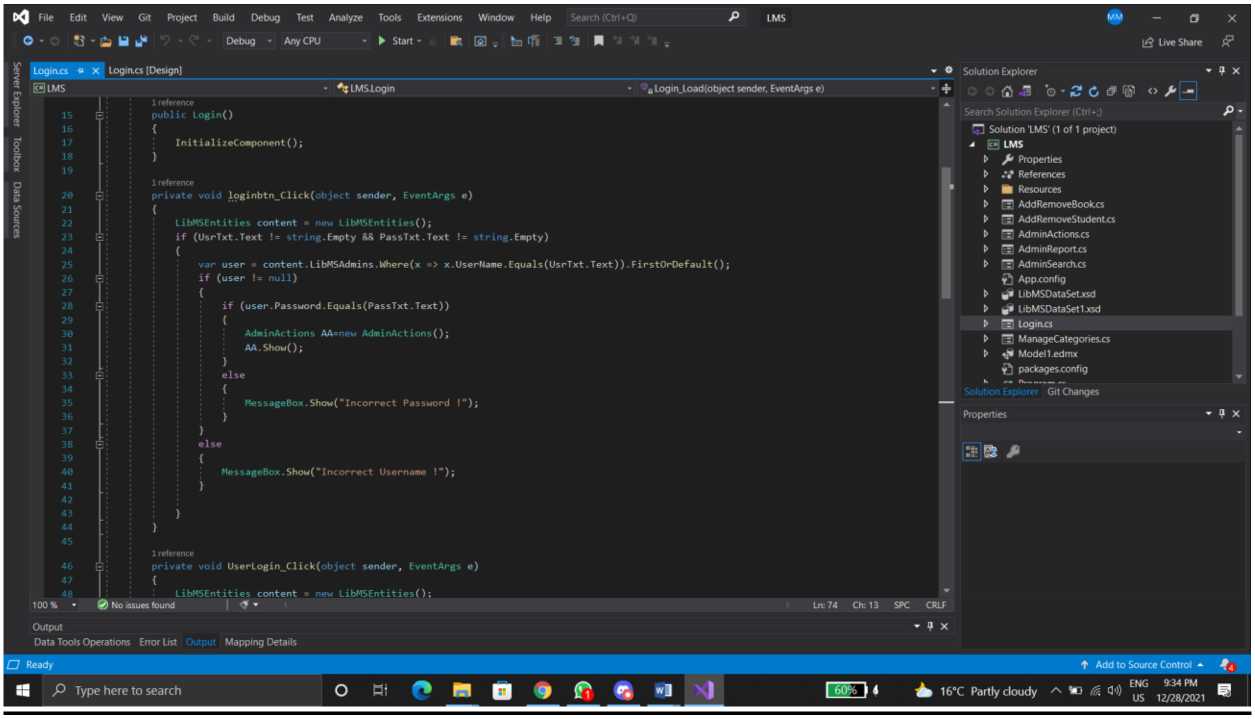


# 8.Implementation

## Login:

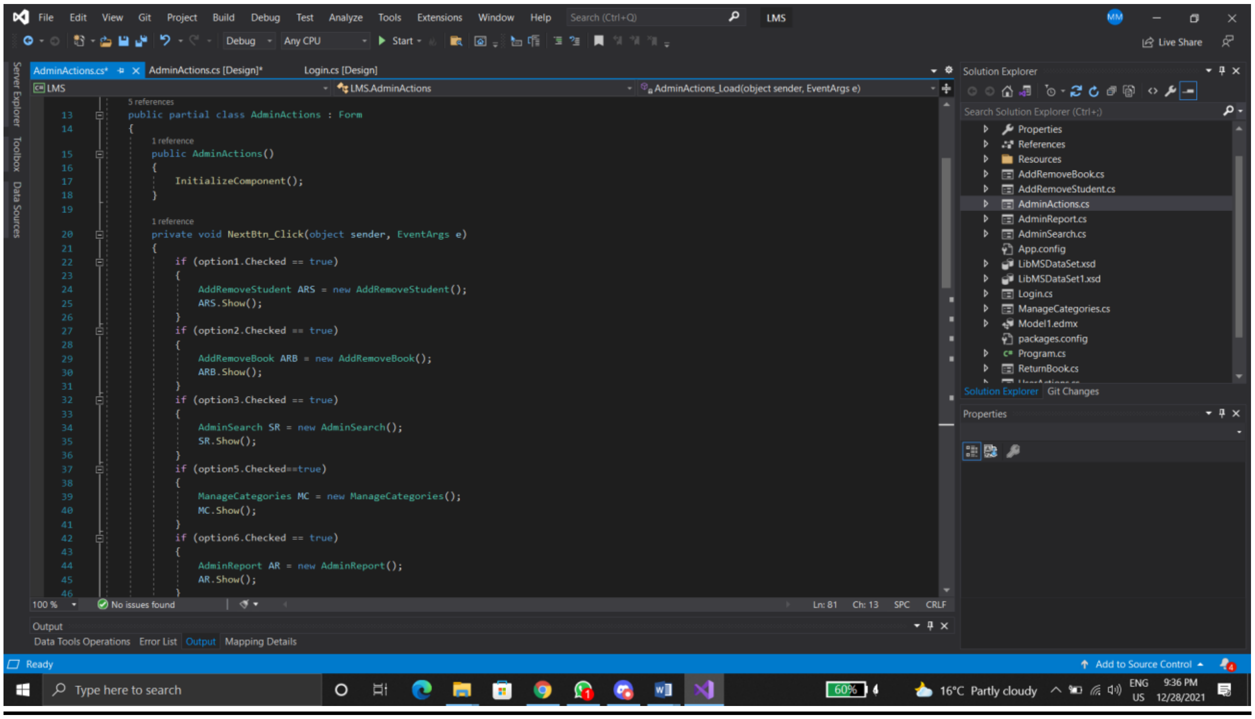


```
            }
        }

        1 reference
        private void UserLogin_Click(object sender, EventArgs e)
        {
            LibMSEntities content = new LibMSEntities();
            if (UsrTxt.Text != string.Empty && PassTxt.Text != string.Empty)
            {
                var user = content.LibUsers.Where(x => x.UserName.Equals(UsrTxt.Text)).FirstOrDefault();
                if (user != null)
                {
                    if (user.Password.Equals(PassTxt.Text))
                    {
                        UserActions UA = new UserActions();
                        UA.Show();
                    }
                    else
                    {
                        MessageBox.Show("Incorrect Password");
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect Username");
                }

            }
        }

        1 reference
        private void Login_Load(object sender, EventArgs e)
        {

        }

        1 reference
```



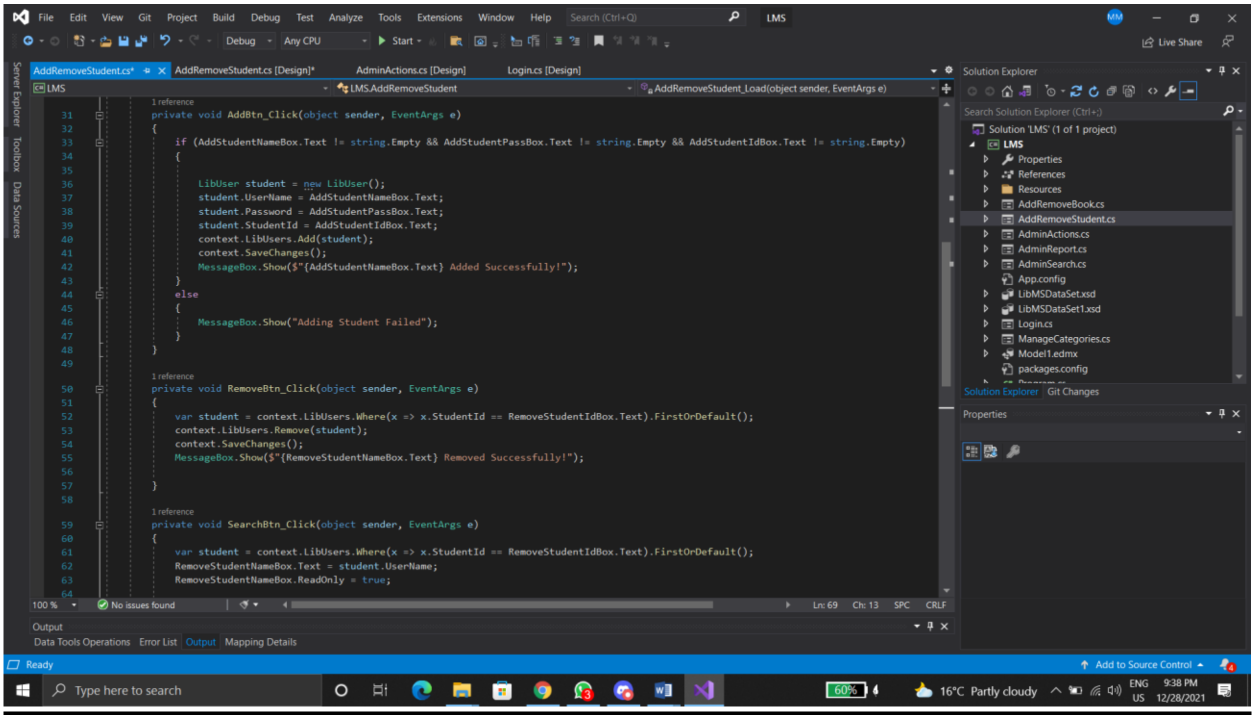
```
        1 reference
        public Login()
        {
            InitializeComponent();
        }

        1 reference
        private void loginbtn_Click(object sender, EventArgs e)
        {
            LibMSEntities content = new LibMSEntities();
            if (UsrTxt.Text != string.Empty && PassTxt.Text != string.Empty)
            {
                var user = content.LibMSAdmins.Where(x => x.UserName.Equals(UsrTxt.Text)).FirstOrDefault();
                if (user != null)
                {
                    if (user.Password.Equals(PassTxt.Text))
                    {
                        AdminActions AA=new AdminActions();
                        AA.Show();
                    }
                    else
                    {
                        MessageBox.Show("Incorrect Password !");
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect Username !");
                }

            }
        }

        1 reference
        private void UserLogin_Click(object sender, EventArgs e)
        {
            LibMSEntities content = new LibMSEntities();
```

## Admin actions



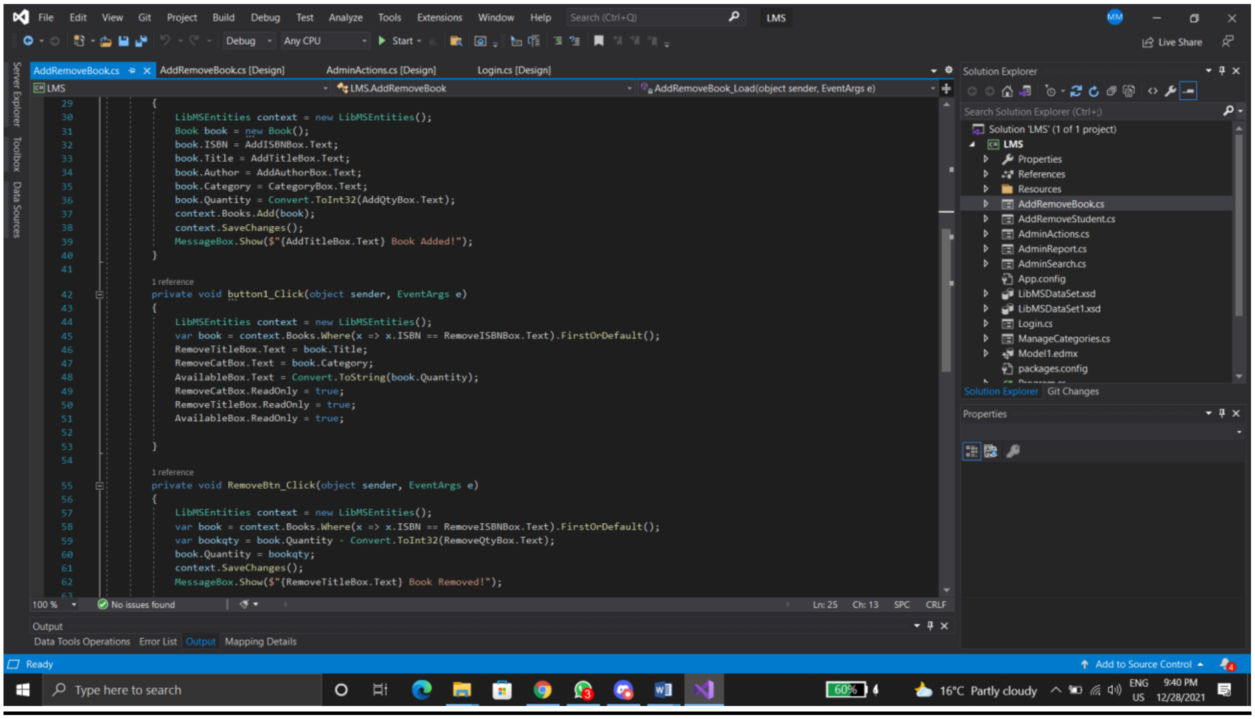
```
    5 references
    public partial class AdminActions : Form
    {
        1 reference
        public AdminActions()
        {
            InitializeComponent();
        }

        1 reference
        private void NextBtn_Click(object sender, EventArgs e)
        {
            if (option1.Checked == true)
            {
                AddRemoveStudent ARS = new AddRemoveStudent();
                ARS.Show();
            }
            if (option2.Checked == true)
            {
                AddRemoveBook ARB = new AddRemoveBook();
                ARB.Show();
            }
            if (option3.Checked == true)
            {
                AdminSearch SR = new AdminSearch();
                SR.Show();
            }
            if (option5.Checked==true)
            {
                ManageCategories MC = new ManageCategories();
                MC.Show();
            }
            if (option6.Checked == true)
            {
                AdminReport AR = new AdminReport();
                AR.Show();
            }
```

## Add/Remove Student



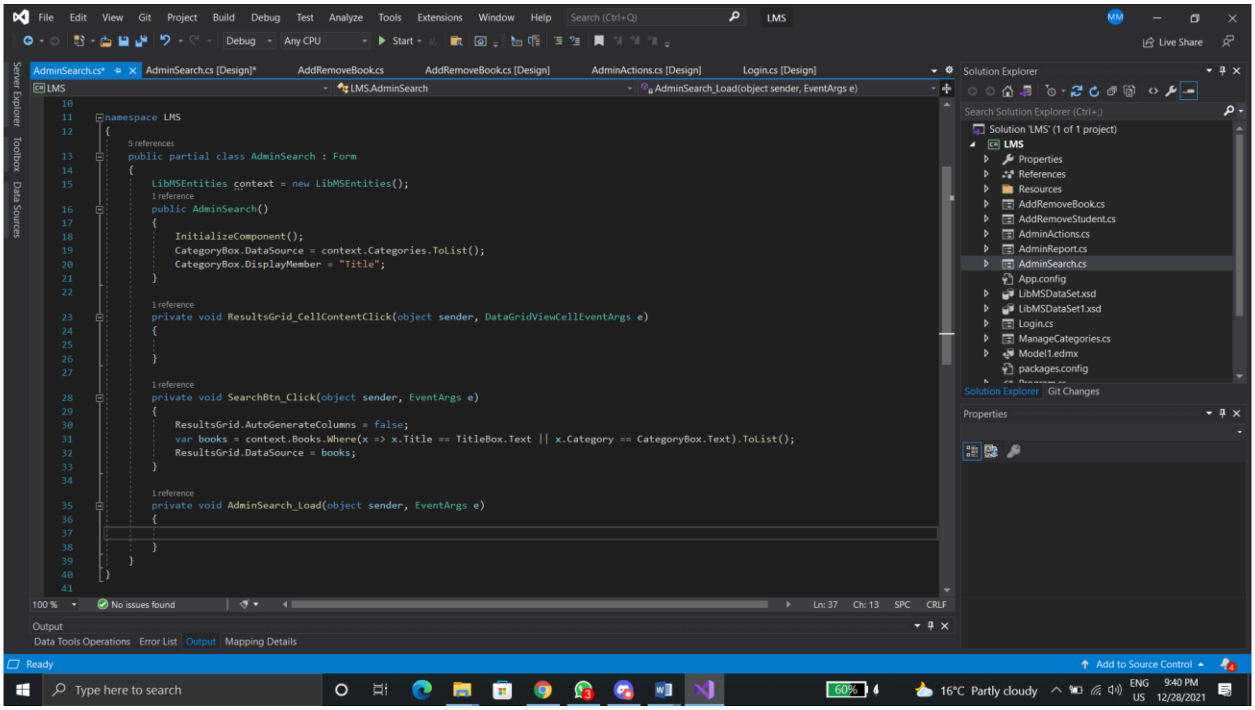
```
1 reference
private void AddBtn_Click(object sender, EventArgs e)
{
    if (AddStudentNameBox.Text != string.Empty && AddStudentPassBox.Text != string.Empty && AddStudentIdBox.Text != string.Empty)
    {

        LibUser student = new LibUser();
        student.UserName = AddStudentNameBox.Text;
        student.Password = AddStudentPassBox.Text;
        student.StudentId = AddStudentIdBox.Text;
        context.LibUsers.Add(student);
        context.SaveChanges();
        MessageBox.Show($"{AddStudentNameBox.Text} Added Successfully!");
    }
    else
    {
        MessageBox.Show("Adding Student Failed");
    }
}

1 reference
private void RemoveBtn_Click(object sender, EventArgs e)
{
    var student = context.LibUsers.Where(x => x.StudentId == RemoveStudentIdBox.Text).FirstOrDefault();
    context.LibUsers.Remove(student);
    context.SaveChanges();
    MessageBox.Show($"{RemoveStudentNameBox.Text} Removed Successfully!");

}

1 reference
private void SearchBtn_Click(object sender, EventArgs e)
{
    var student = context.LibUsers.Where(x => x.StudentId == RemoveStudentIdBox.Text).FirstOrDefault();
    RemoveStudentNameBox.Text = student.UserName;
    RemoveStudentNameBox.ReadOnly = true;
```

## Add/Remove Book



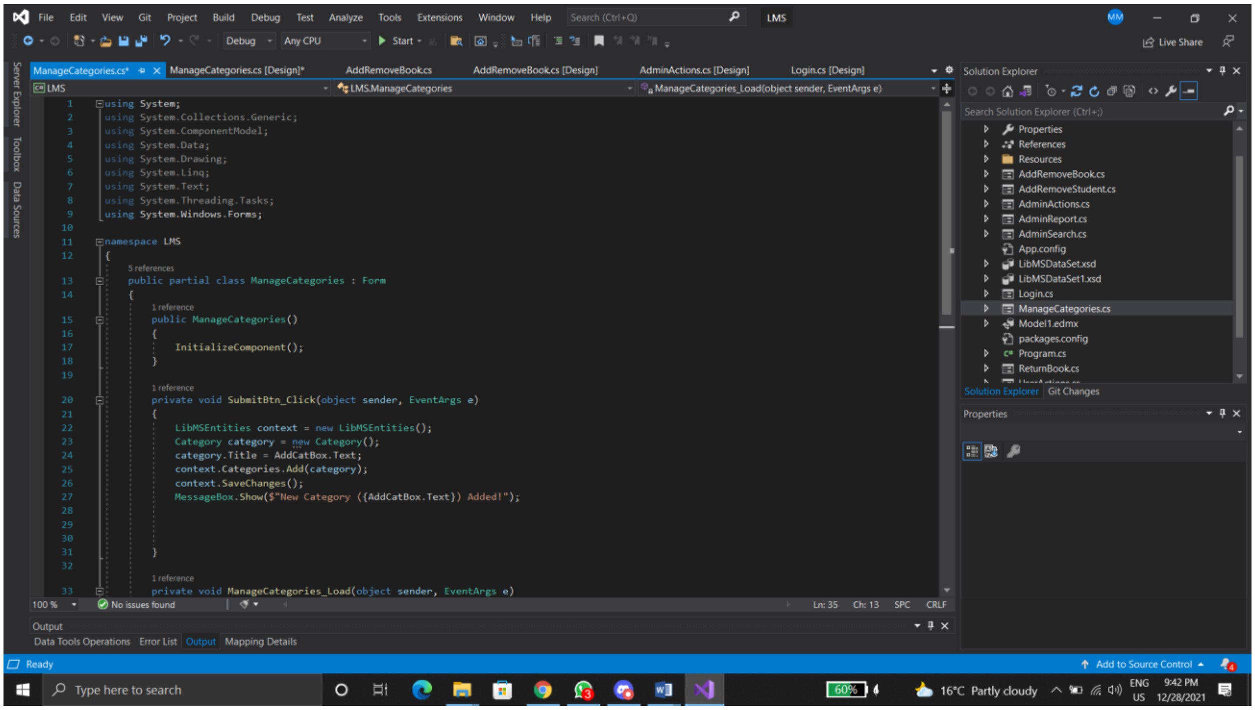
```
{
    LibMSEntities context = new LibMSEntities();
    Book book = new Book();
    book.ISBN = AddISBNBox.Text;
    book.Title = AddTitleBox.Text;
    book.Author = AddAuthorBox.Text;
    book.Category = CategoryBox.Text;
    book.Quantity = Convert.ToInt32(AddQtyBox.Text);
    context.Books.Add(book);
    context.SaveChanges();
    MessageBox.Show($"{AddTitleBox.Text} Book Added!");
}

1 reference
private void button1_Click(object sender, EventArgs e)
{
    LibMSEntities context = new LibMSEntities();
    var book = context.Books.Where(x => x.ISBN == RemoveISBNBox.Text).FirstOrDefault();
    RemoveTitleBox.Text = book.Title;
    RemoveCatBox.Text = book.Category;
    AvailableBox.Text = Convert.ToString(book.Quantity);
    RemoveCatBox.ReadOnly = true;
    RemoveTitleBox.ReadOnly = true;
    AvailableBox.ReadOnly = true;

}

1 reference
private void RemoveBtn_Click(object sender, EventArgs e)
{
    LibMSEntities context = new LibMSEntities();
    var book = context.Books.Where(x => x.ISBN == RemoveISBNBox.Text).FirstOrDefault();
    var bookqty = book.Quantity - Convert.ToInt32(RemoveQtyBox.Text);
    book.Quantity = bookqty;
    context.SaveChanges();
    MessageBox.Show($"{RemoveTitleBox.Text} Book Removed!");
```

# Admin Search



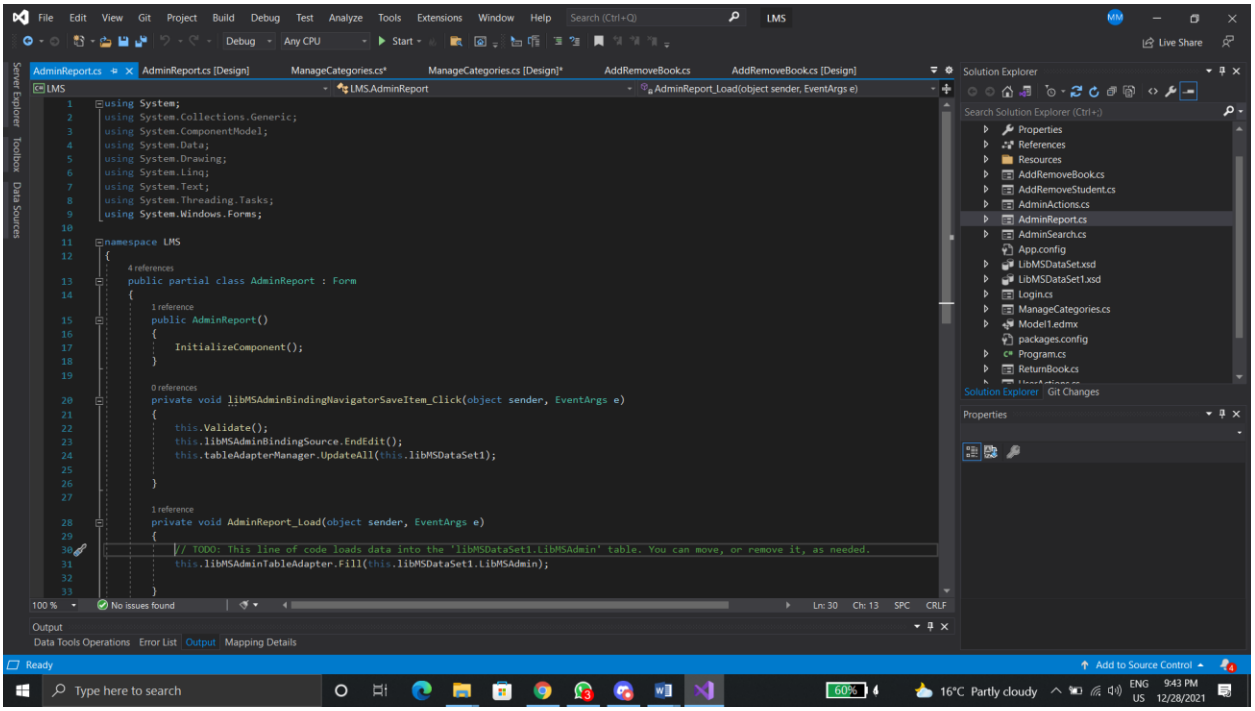
```
namespace LMS
{
    public partial class AdminSearch : Form
    {
        LibMSEntities context = new LibMSEntities();
        public AdminSearch()
        {
            InitializeComponent();
            CategoryBox.DataSource = context.Categories.ToList();
            CategoryBox.DisplayMember = "Title";
        }

        private void ResultsGrid_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {

        }

        private void SearchBtn_Click(object sender, EventArgs e)
        {
            ResultsGrid.AutoGenerateColumns = false;
            var books = context.Books.Where(x => x.Title == TitleBox.Text || x.Category == CategoryBox.Text).ToList();
            ResultsGrid.DataSource = books;
        }

        private void AdminSearch_Load(object sender, EventArgs e)
        {

        }
    }
}
```

# Manage Categories



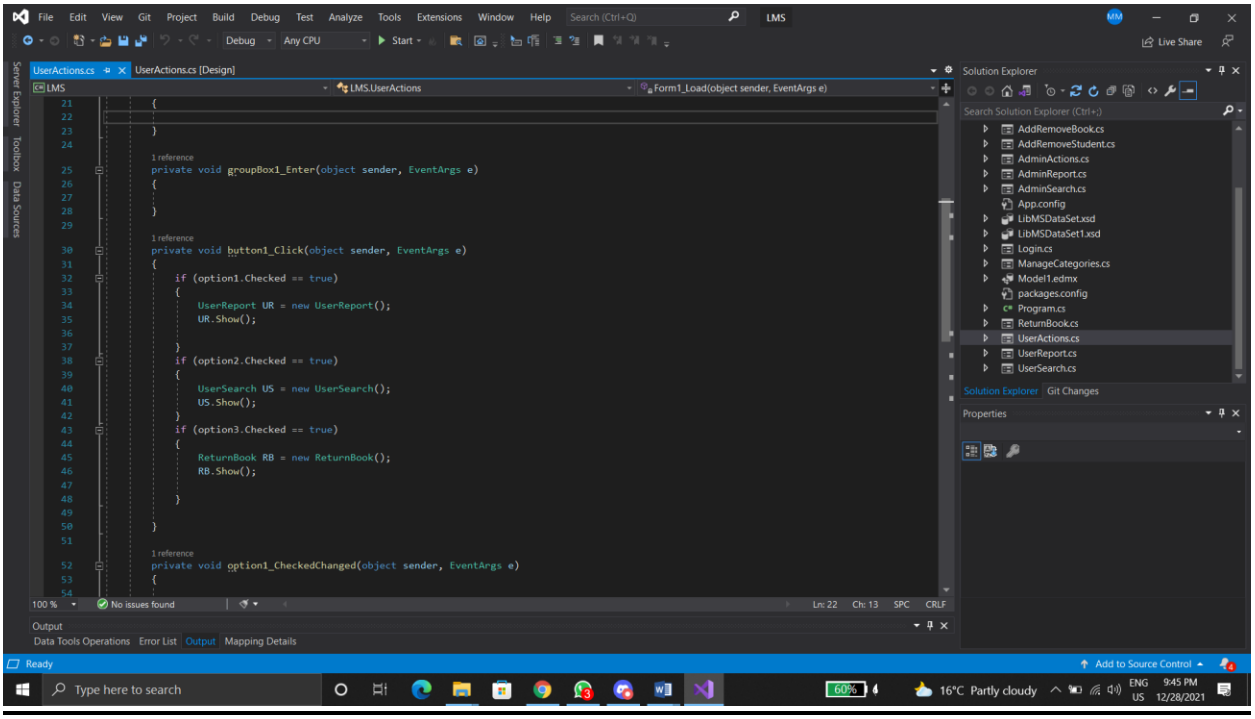
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LMS
{
    public partial class ManageCategories : Form
    {
        public ManageCategories()
        {
            InitializeComponent();
        }

        private void SubmitBtn_Click(object sender, EventArgs e)
        {
            LibMSEntities context = new LibMSEntities();
            Category category = new Category();
            category.Title = AddCatBox.Text;
            context.Categories.Add(category);
            context.SaveChanges();
            MessageBox.Show($"New Category ({AddCatBox.Text}) Added!");



        }

        private void ManageCategories_Load(object sender, EventArgs e)
```

# Admin Report



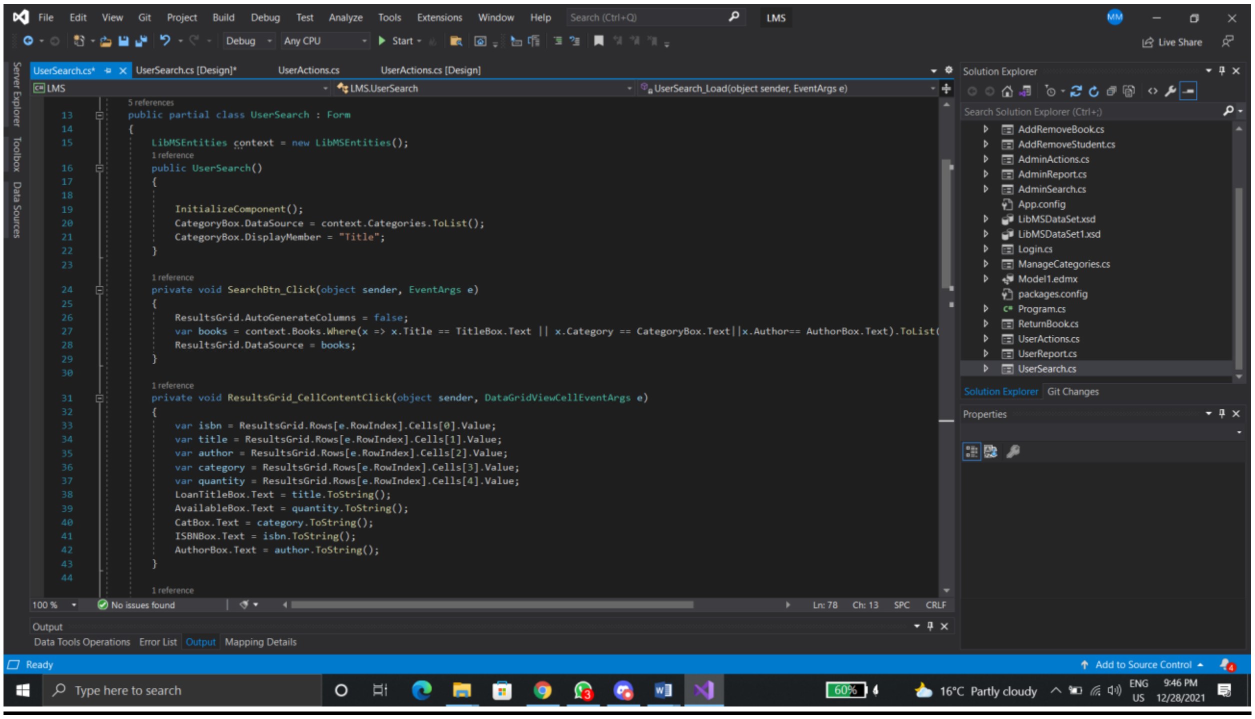
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LMS
{
    4 references
    public partial class AdminReport : Form
    {
        1 reference
        public AdminReport()
        {
            InitializeComponent();
        }

        0 references
        private void libMSAdminBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.libMSAdminBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.libMSDataSet1);

        }

        1 reference
        private void AdminReport_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'libMSDataSet1.LibMSAdmin' table. You can move, or remove it, as needed.
            this.libMSAdminTableAdapter.Fill(this.libMSDataSet1.LibMSAdmin);

        }
```

# User Actions

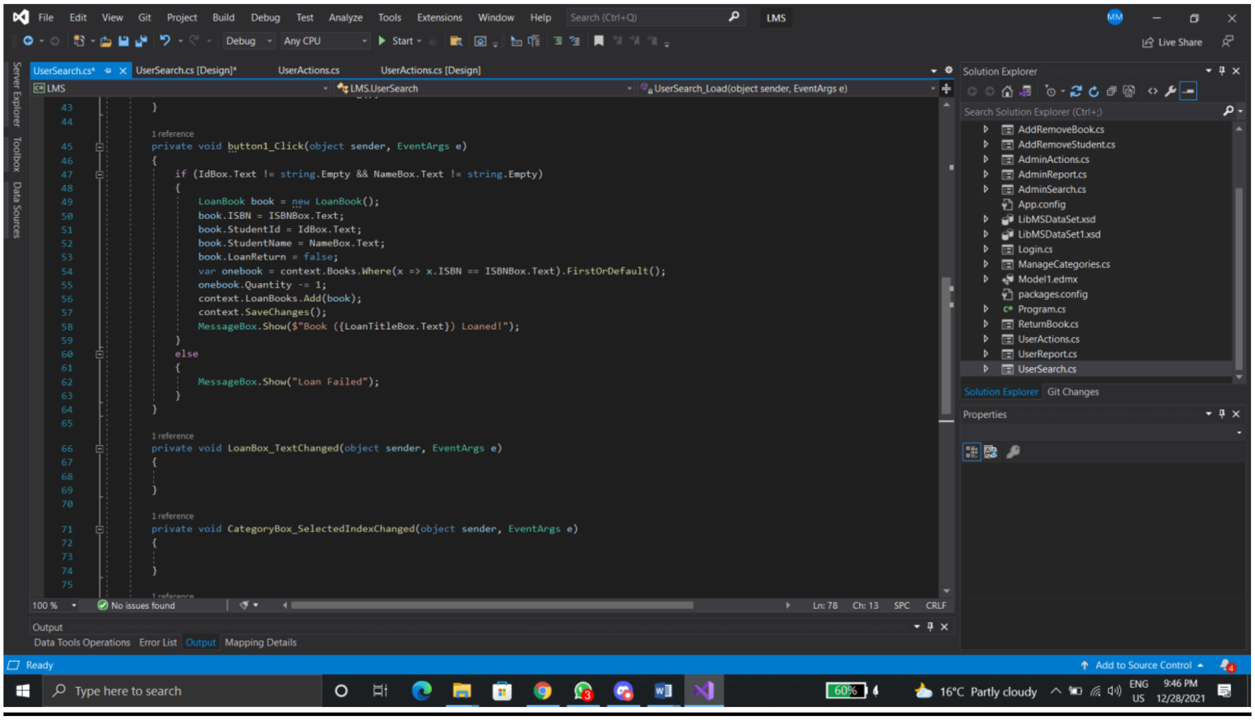


```
            {

            }

            1 reference
            private void groupBox1_Enter(object sender, EventArgs e)
            {

            }

            1 reference
            private void button1_Click(object sender, EventArgs e)
            {
                if (option1.Checked == true)
                {
                    UserReport UR = new UserReport();
                    UR.Show();

                }
                if (option2.Checked == true)
                {
                    UserSearch US = new UserSearch();
                    US.Show();
                }
                if (option3.Checked == true)
                {
                    ReturnBook RB = new ReturnBook();
                    RB.Show();

                }

            }

            1 reference
            private void option1_CheckedChanged(object sender, EventArgs e)
            {
```

# Search/Loan Book

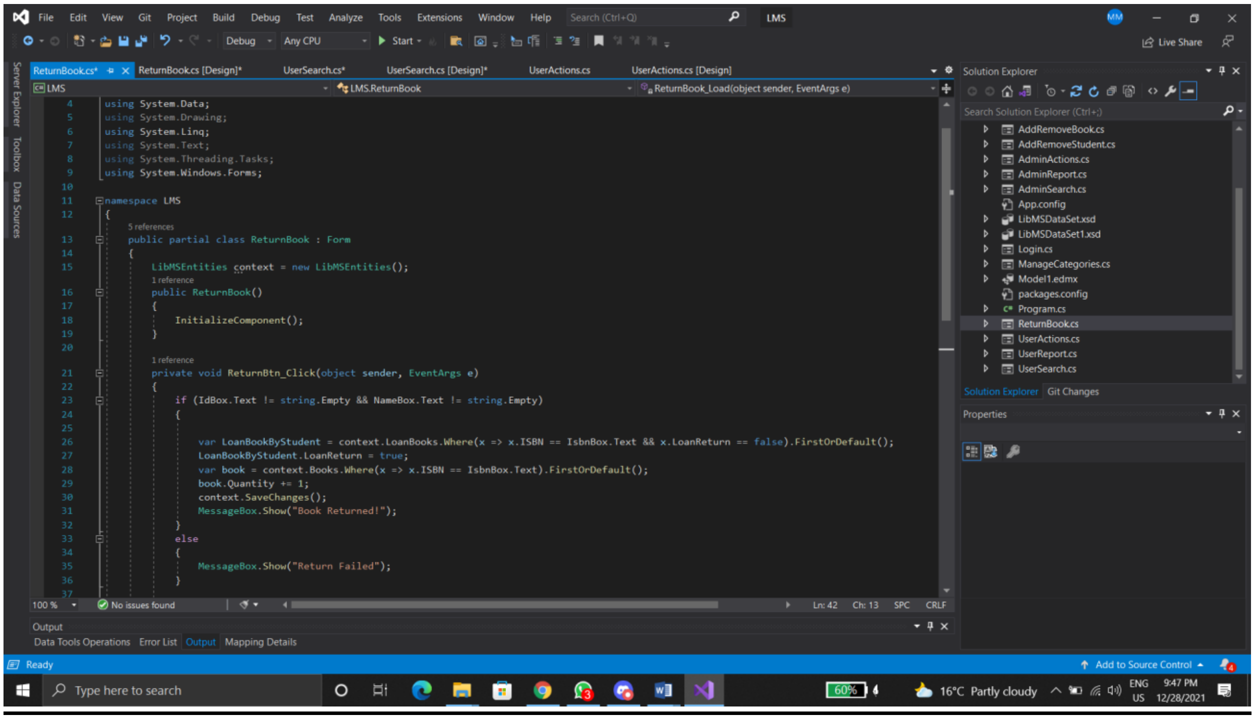


```csharp
public partial class UserSearch : Form
{
    LibMSEntities context = new LibMSEntities();
    public UserSearch()
    {

        InitializeComponent();
        CategoryBox.DataSource = context.Categories.ToList();
        CategoryBox.DisplayMember = "Title";
    }

    private void SearchBtn_Click(object sender, EventArgs e)
    {
        ResultsGrid.AutoGenerateColumns = false;
        var books = context.Books.Where(x => x.Title == TitleBox.Text || x.Category == CategoryBox.Text||x.Author== AuthorBox.Text).ToList(
        ResultsGrid.DataSource = books;
    }

    private void ResultsGrid_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        var isbn = ResultsGrid.Rows[e.RowIndex].Cells[0].Value;
        var title = ResultsGrid.Rows[e.RowIndex].Cells[1].Value;
        var author = ResultsGrid.Rows[e.RowIndex].Cells[2].Value;
        var category = ResultsGrid.Rows[e.RowIndex].Cells[3].Value;
        var quantity = ResultsGrid.Rows[e.RowIndex].Cells[4].Value;
        LoanTitleBox.Text = title.ToString();
        AvailableBox.Text = quantity.ToString();
        CatBox.Text = category.ToString();
        ISBNBox.Text = isbn.ToString();
        AuthorBox.Text = author.ToString();
    }

```



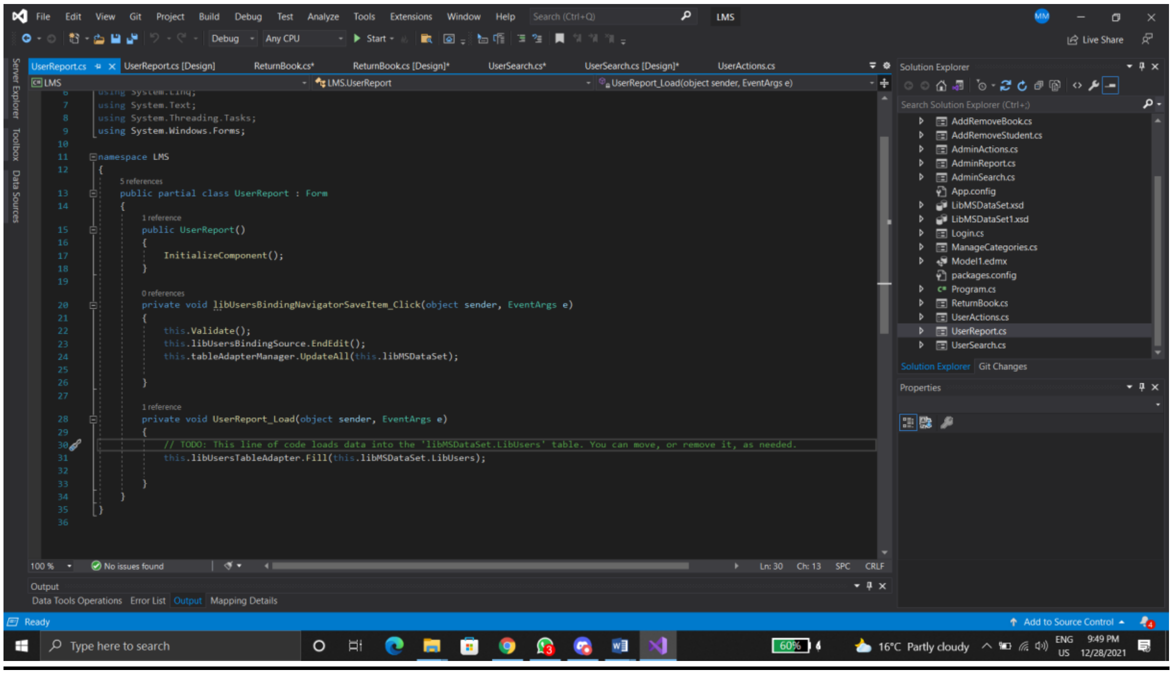
```csharp
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (IdBox.Text != string.Empty && NameBox.Text != string.Empty)
        {
            LoanBook book = new LoanBook();
            book.ISBN = ISBNBox.Text;
            book.StudentId = IdBox.Text;
            book.StudentName = NameBox.Text;
            book.LoanReturn = false;
            var onebook = context.Books.Where(x => x.ISBN == ISBNBox.Text).FirstOrDefault();
            onebook.Quantity -= 1;
            context.LoanBooks.Add(book);
            context.SaveChanges();
            MessageBox.Show($"Book ({LoanTitleBox.Text}) Loaned!");
        }
        else
        {
            MessageBox.Show("Loan Failed");
        }
    }

    private void LoanBox_TextChanged(object sender, EventArgs e)
    {

    }

    private void CategoryBox_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

```

# Return a Book



```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LMS
{
    public partial class ReturnBook : Form
    {
        LibMSEntities context = new LibMSEntities();
        public ReturnBook()
        {
            InitializeComponent();
        }

        private void ReturnBtn_Click(object sender, EventArgs e)
        {
            if (IdBox.Text != string.Empty && NameBox.Text != string.Empty)
            {

                var LoanBookByStudent = context.LoanBooks.Where(x => x.ISBN == IsbnBox.Text && x.LoanReturn == false).FirstOrDefault();
                LoanBookByStudent.LoanReturn = true;
                var book = context.Books.Where(x => x.ISBN == IsbnBox.Text).FirstOrDefault();
                book.Quantity += 1;
                context.SaveChanges();
                MessageBox.Show("Book Returned!");
            }
            else
            {
                MessageBox.Show("Return Failed");
            }
```

# User Report



```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LMS
{
    public partial class UserReport : Form
    {
        public UserReport()
        {
            InitializeComponent();
        }

        private void libUsersBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.libUsersBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.libMSDataSet);

        }

        private void UserReport_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'libMSDataSet.LibUsers' table. You can move, or remove it, as needed.
            this.libUsersTableAdapter.Fill(this.libMSDataSet.LibUsers);

        }
    }
}
```

# 9.Testing

| Function | Works well |
|---|---|
| Admin Login | ✓ |
| Add/Remove Student | ✓ |
| Add/Remove Books | ✓ |
| Manage Categories | ✓ |
| View Report | ✓ |
| User Login | ✓ |
| Search/Loan Book | ✓ |
| Return Book | ✓ |
| View Report | ✓ |