

暨南大学计算机科学系

《Windows 编程》课程设计

成绩: _____

设计题目: _____ 图书管理系统的设计与开发

姓 名: _____ 阮炜霖

学 号: _____ 2020101603

学 期: _____ 2022-2023 年下半年

学 分: _____ 2 学分

任课老师: _____ 王娜

提交日期: _____ 2023 年 6 月 8 号

图书管理系统的设计与实现

1. 概述

1.1 项目简介

本项目旨在设计一个图书管理系统，为暨南大学图书馆提供图书馆里客户端雏形。该图书管理系统采用.NET、C#、数据库、Windows Forms 等开发技术，实现读者快捷查询、借书还书以及管理员对书籍和用户的管理等功能。通过应用程序的开发，本项目旨在帮助图书管理员和读者更高效地利用学校图书资源、处理借阅信息、以及提供图书馆服务等。

1.2 项目背景与意义

该项目灵感来源于暨南大学的图书管理系统，其系统极大程度上提高了暨南大学图书馆的管理效率、提升读者的借阅体验，为全校师生提供了优质的图书资源服务。即便现有的图书管理系统方便快捷，庞大的书籍和用户数据也造成了检索效率低下、部分信息难以处理等问题；此外，现有系统缺少相应的客户端能够在远程为读者和图书管理人员提供服务。

因此，本项目基于暨南大学现有图书管理系统进行设计，并以轻量快捷的客户端形式呈现，希望能够为学校图书馆里系统的设计提供方法和经验。此外，通过为现有系统改进效率、弥补空缺，能够为图书馆管理人员提供更高效的管理方法并为读者提供更好的借阅体验。

1.3 系统建设的目标

该项目的目标是设计一个直观且用户友好的系统界面，该系统不仅能使图书管理员能够轻松地执行各种管理任务，如图书入库、借阅管理、图书归还、读者信息管理等，而且使读者更加方便修改读者信息和借阅书籍。通过对数据系统原理的掌握和应用，客户端能够连接 mysql 本地数据库，使用索引和视图等技术提高增删改查的速度，以便更好地满足读者和管理员的需求。

2. 系统的需求分析

2.1 目标客户描述

本项目的目标用户主要分为两大类，分别是学校中的图书馆管理人员和广大读者，下面是对客户需求的描述：

（1）图书馆管理员：图书馆管理员是该系统的主要目标客户群体之一。他们负责管理图书馆的日常运营，包括图书的录入、借阅管理、读者信息管理等。该系统将为图书馆管理员提供便捷的图书管理工具，帮助他们更高效地管理图书资源、处理借阅还书、查看事务。

（2）读者：读者是图书管理系统的另一个重要用户群体。他们可以通过系统进行图书检索和查询，了解图书的基本信息、可借阅状态等。读者还可以利用

系统进行借书操作，方便快捷地查看自己的借阅记录。系统提供的个人账户功能使读者能够查看个人借阅历史、设置提醒和预约图书等。

通过为图书馆管理人员和读者提供更便捷的服务，该系统旨在提高图书管理效率、优化读者体验，并促进学校图书馆运营与发展。

2.2 分析功能需求

针对两类客户，该项目的客户端分开进行了功能设计。读者的功能侧重于账号管理和借阅体验，而管理员的功能则侧重于对用户和书籍信息的快捷管理。另外还需要实现客户端的其他基本功能窗口保证体验完整。

(1) 账号管理：在客户端的启动界面，用户可以选择登录注册及注销。在用户在输入正确的用户名和密码后，可以登录到对应的借阅/管理系统。此外，还能查看用户信息、权限、修改密码等等。

(2) 书籍管理：包括录入、修改和删除图书信息，如书名、作者、分类、出版社、出版时间等等，并提供书籍借阅记录的查询功能。

(3) 读者管理：查看和管理读者信息，包括读者的个人资料、借书记录、以及权限，方便读者管理和查询并方便管理员进行管理。

(4) 借阅服务：提供便捷的图书检索和查询功能，读者可按照书名、作者、分类等条件进行搜索，查找所需图书并借阅；而管理员可在系统上按同样的方式查询书籍并标记借阅或归还。

3. 相关技术与工具

在开发该图书管理系统时，所需用到的技术主要有：

(1) .NET 框架：是一种用于构建和运行 Windows 应用程序的开发平台。它提供了广泛的类库和工具，包括 Windows Forms、WPF 和 UWP 等应用程序开发框架，以及用于开发和调试的工具集。

(2) C#语言：是一种面向对象的编程语言，是.NET 框架的主要编程语言之一。C#具有简洁、安全、高效的特点，可用于开发各种类型的 Windows 应用程序，并与.NET 框架无缝集成。

(3) Visual Studio：是 Microsoft 提供的集成开发环境 (IDE)，用于开发 Windows 应用程序。它提供了丰富的工具和功能，包括代码编辑器、调试器、界面设计器和构建工具，以支持开发人员的应用程序开发和调试工作。

(4) JetBrains Rider：是一种跨平台的集成开发环境 (IDE)，用于 .NET 和 .NET Core 应用程序的开发。它支持多种编程语言进行开发，并提供了丰富的功能和工具，使开发人员能够高效地创建和调试 Windows 应用程序。

在该项目的编写阶段中，我主要以 Rider 为主，Visual Studio 为辅进行开发。

(5) 数据库技术：Windows 应用程序通常需要与数据库进行交互。该项目主要使用了 MySQL 来对应用程序进行数据库系统的集成，并使得应用程序能够存储和检索数据。

(6) JetBrains DataGrip：使用 JetBrains 的 DataGrip 作为数据库开发工具，它是一款强大的跨平台数据库 IDE，支持多种关系型和非关系型数据库，提供了丰富的数据库管理和查询功能。

(7) Windows Forms：是一种用于创建传统 Windows 桌面应用程序的 UI 框架。

它提供了各种用户界面控件和事件模型，以及与系统 API 的集成，使开发人员可以创建界面简单、易于使用的 Windows 应用程序。

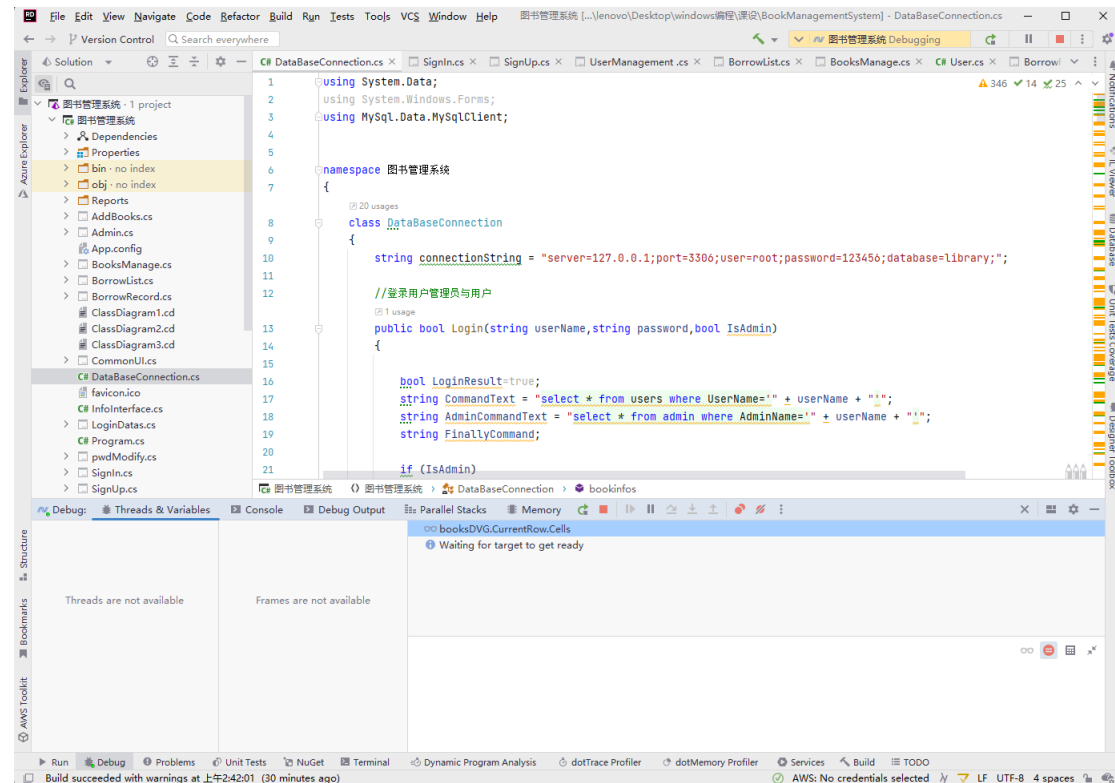


图 1 JetBrains Rider 编程环境

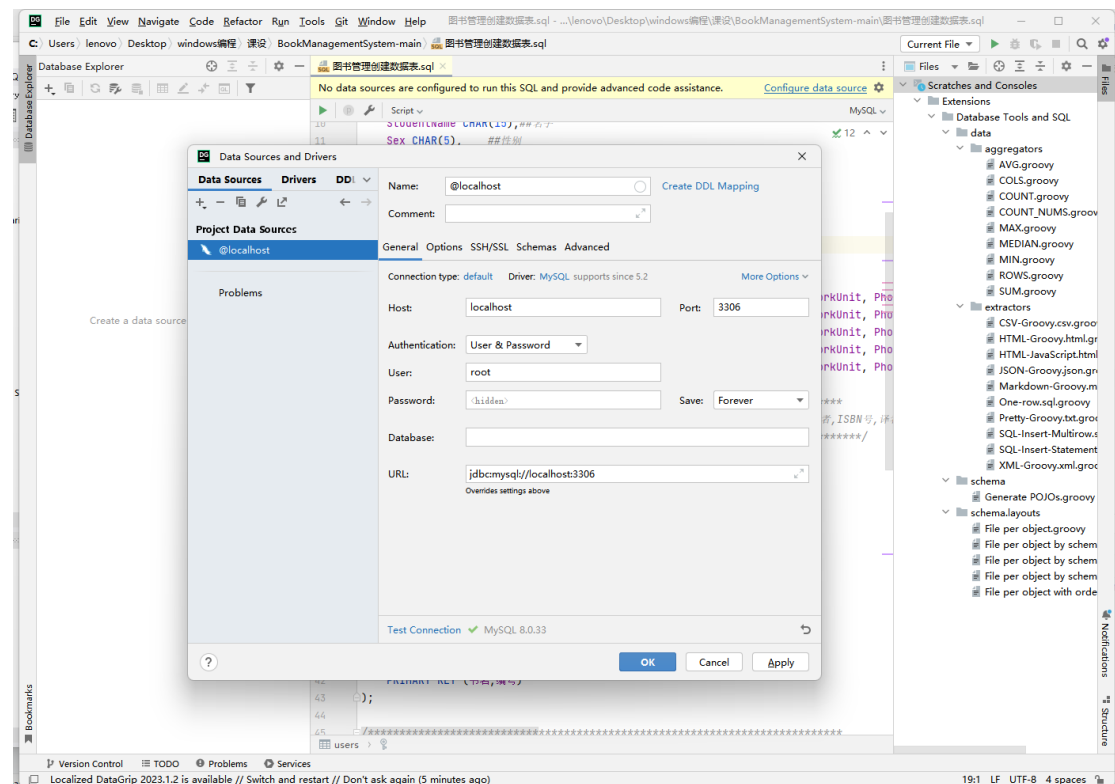


图 2 JetBrains Datagrip 连接 mysql 数据库

4. 系统的设计与实现

4.1 系统的数据模型与数据库设计

根据系统的逻辑结构，设计如下 5 个表，其中下划线为主键，斜体为外键：

- 普通用户（用户名，姓名，密码，学院，电话，注册时间）
- 书籍信息（书名，书籍编号，类别，作者，译者，索引号，别名，ISBN 号，语种，摘要，出版社，出版时间）
- 管理员（用户名，密码）
- 借阅信息（书籍编号，书名，用户名，借阅时间，应还时间，是否归还，归还时间）
- 登录信息（用户名，登陆时间）

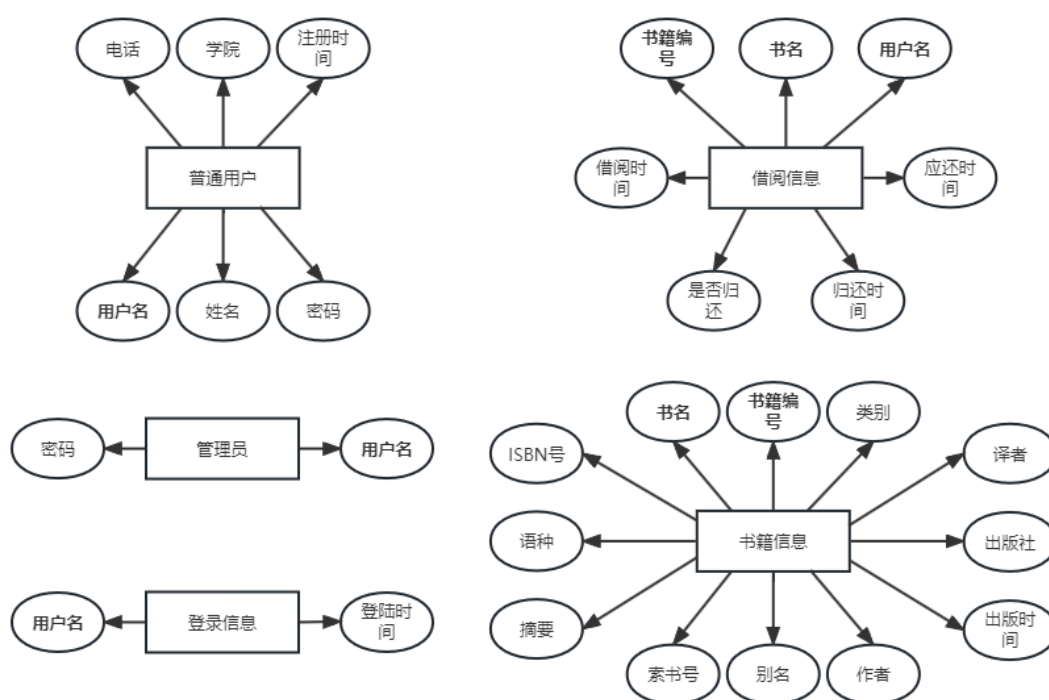


图 3 数据库实体

ER 图（实体关系图）是一种用于描述数据库中实体、属性和关系之间关联的图形表示方法。作为数据库设计的工具，ER 图可以帮助开发者和数据库管理员理清数据模型，定义实体、属性和关系，并确定它们之间的约束和规则。通过 ER 图，可以直观地展示数据库的结构和组成，为数据库的创建和维护提供指导和依据。根据数据库实体做出以下 ER 图：

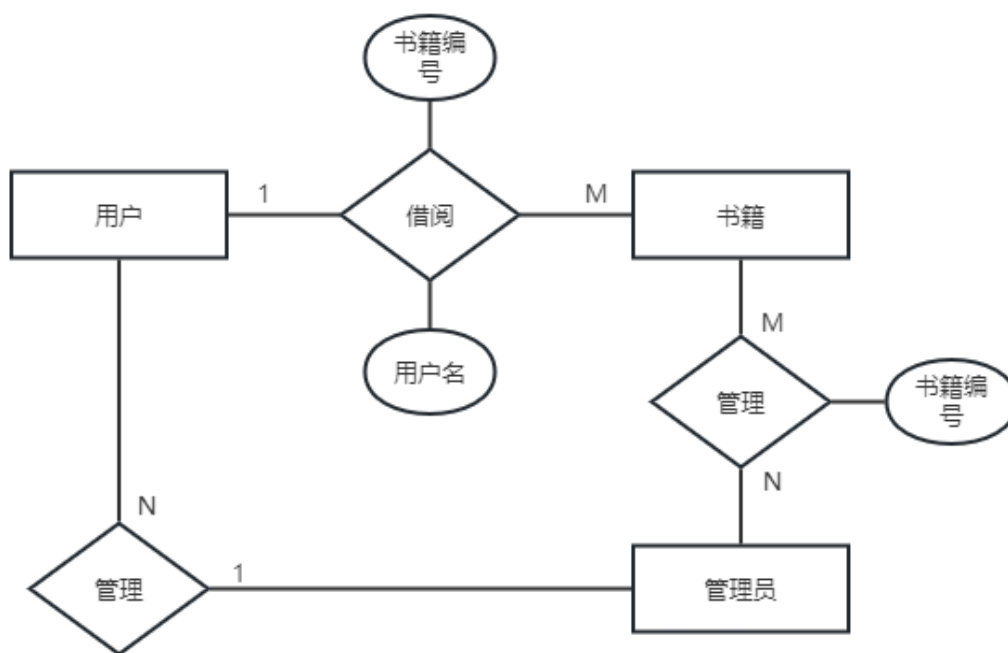


图 4 简单画出数据库 ER 图

按照功能需求，数据库关系设计管理员账号、用户账号、存书信息和借阅记录四个表。在该数据库中，管理员和用户的用户名、存书信息的书名与编号属于主码，而借阅信息中的图书编号、书名和借阅人属于外码，分别参考用户与存书的主码信息。Admin 用户可管理所有表，users 普通用户只能使用 loaninfo 借阅表。根据需求编写并运行 SQL 命令语句，并保存为路径下的 create.sql。

在 JetBrains Datagrip 中连接本地 mysql 数据库，运行上述 sql 代码，随后生成该项目系统数据库所需的所有表，如下图所示。

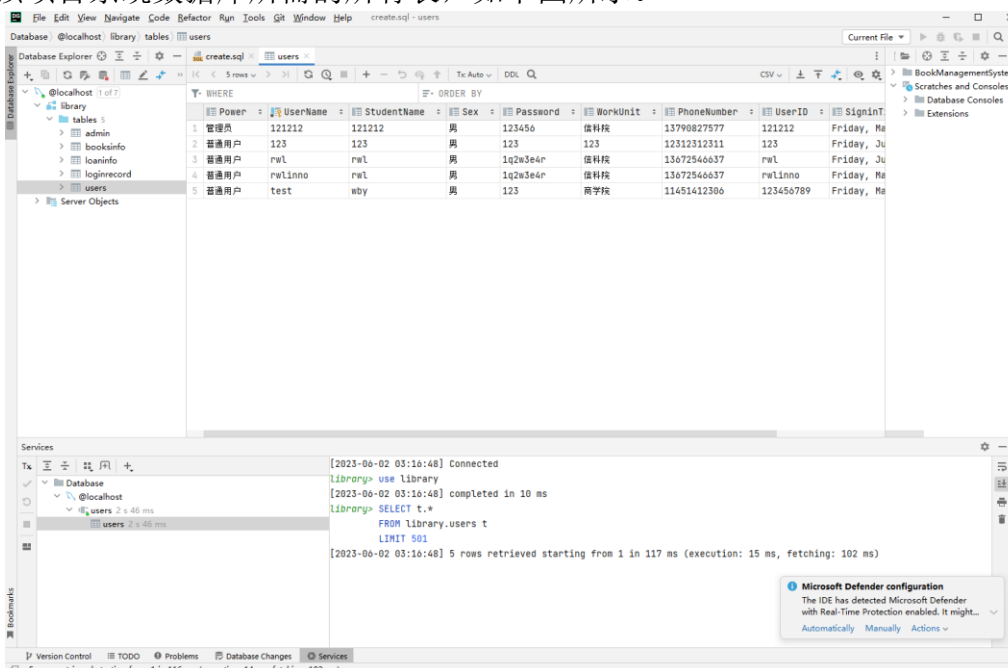


图 5 运行对应 sql 语句后的结果

随后利用 Jetbrains Datagrip 生成数据库的实体关系图，以直观的方式展示数据库中的表、字段以及它们之间的关系。这种可视化的方式使我能够更清晰地理解和浏览数据库的结构，便于进行数据模型的设计和维护。

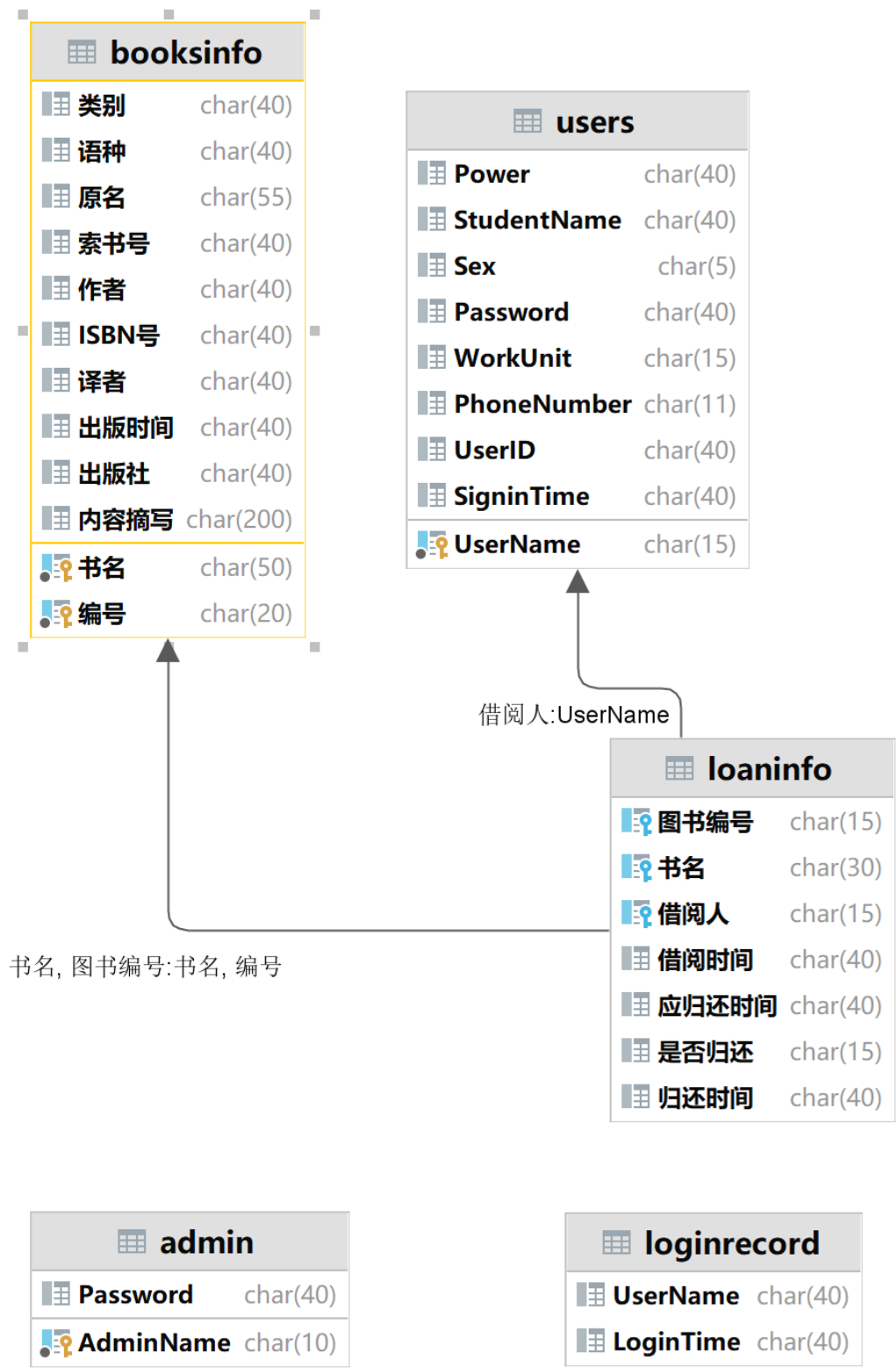


图 6 数据库实体 Diagram

4.2 系统的功能模型

在制作图书管理系统的过程中，预计需要实现的功能会比较多，并且在日后更新维护也需要逐步根据。因此，为了使系统做到高内聚、低耦合，我们采用自顶向下、模块化设计的思想将该项目逐层拆解成最小单元的模块。根据前面 2.2 提及的功能需求分析，可以画出下图所示功能模型。针对该功能模型，对应用程序的控件、方法等有序封装，并在下一节逐一实现和调试。

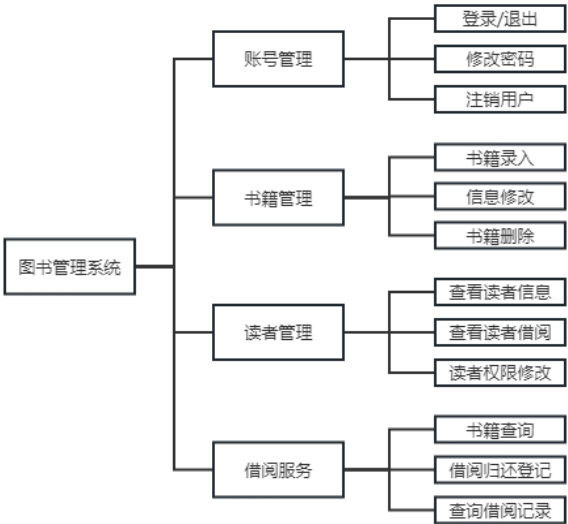


图 7 功能模型图

4.3 系统程序文件设计与编写

4.3.1 管理员首页 Admin.cs

主要实现管理员账号登陆后的窗口页面，包括了继承 Form 的 Admin 类，主要实现的方法有：UsersManageButton_Click()点击用户管理按钮将跳转到用户管理页面；BookManageButton_Click()点击图书管理按钮将跳转到图书管理页面；exitButton_Click()点击退出登陆按钮将关闭窗口并返回登陆页面。代码约在 40 行左右。

```
private void exitButton_Click(object sender, EventArgs e) //关闭窗口
{
    this.Close();
    form.Show();
}

private void UsersManageButton_Click(object sender, EventArgs e) //跳转到管理用户页面
{
    UserManagement userManagement = new UserManagement(form: this);
    this.Hide();
    userManagement.Show();
}

private void BookManageButton_Click(object sender, EventArgs e) //跳转到管理书籍页面
{
    BooksManage booksManage = new BooksManage(form: this);
    this.Hide();
    booksManage.Show();
}
```

图 8 管理员首页代码

4.3.2 书籍管理页面 BooksManage.cs

主要实现书籍管理窗口页面，包括继承了 Form 的 BooksManage 类，主要实现的功能有：exitButton_Click() 点击退出按钮退出该页面并返回主页面；bookAddButton_Click() 点击新增书目进入添加书籍页面；borrowRwcordButton_Click() 点击借书记录按钮进入借书记录页面；btnSearchBook_Click() 点击查找书名按钮在数据库中书籍表查找满足书名字段的记录；searchByKindButton_Click() 点击查找类别按钮在数据库中书籍表查找满足类别字段的记录；btnAlter_Click() 点击修改书目按钮将在数据库中选中对应书籍的信息更改成文本框中的值；booksDVG_CellClick() 按刷新按钮对该页面进行刷新；deleteBook_Click() 按删除书目按钮将在数据库中删除所选书籍。showBook() 使得控件实时显示数据库中满足条件的书目记录。代码约在 120 行左右。

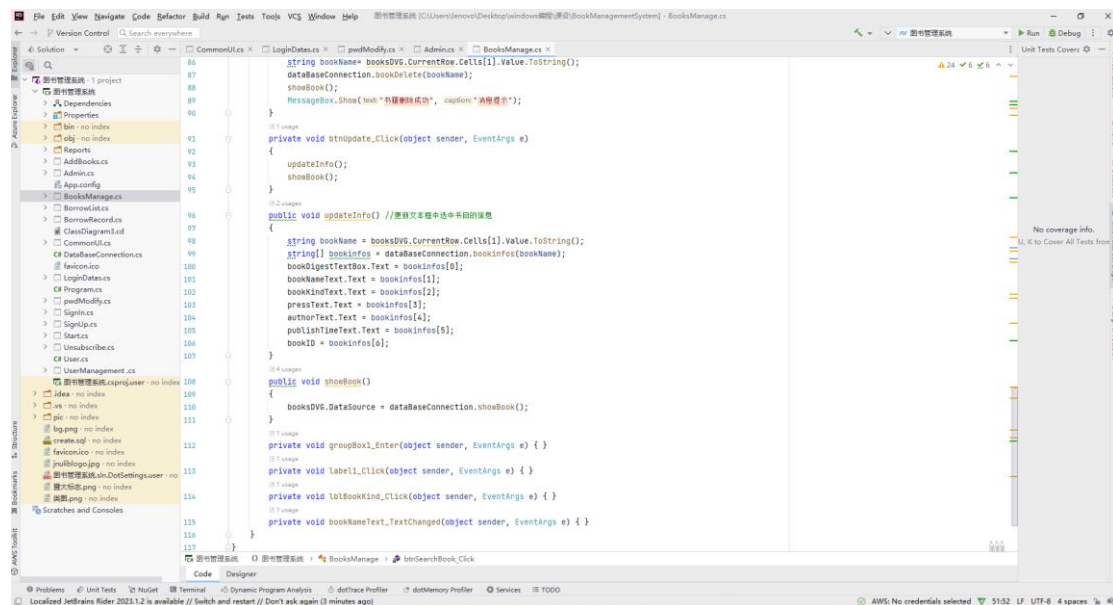


图 9 书籍管理页面代码

4.3.3 借阅清单页面 BorrowList.cs

主要实现借阅清单窗口页面，包括继承了 Form 的 BorrerList 类，主要实现的功能有：searchBookButton_Click() 点击查找书名按钮对该书名的借阅记录进行查询并更新显示列表；button1_Click() 点击查找用户按钮对用户的借阅列表进行查询并更新显示列表；showborrow() 更新显示数据库中借阅记录的控件；exitButton_Click() 点击退出按钮退出该页面。代码大约 60 行。

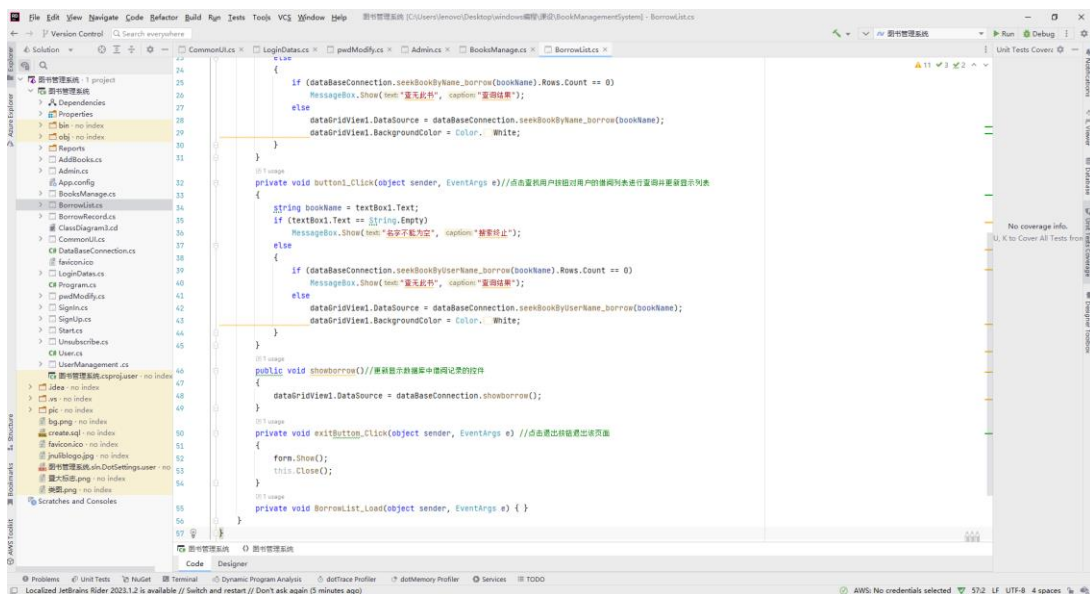


图 10 借阅清单页面代码

4.3.4 借书记录页面 BorrowRecord.cs

主要实现显示借书记录的窗口页面，包括继承了 Form 的 BorrowRecord 类，主要实现的功能有：btnBack_Click() 点击返回按钮返回主页面；btnUpdate_Click() 点击刷新按钮更新显示借书记录的控件；showRecord() 连接数据库使得列表显示数据库中满足用户字段的记录；returnBookButton_Click() 点击还书按钮归还选中书籍并更新数据库。代码大约 60 行。

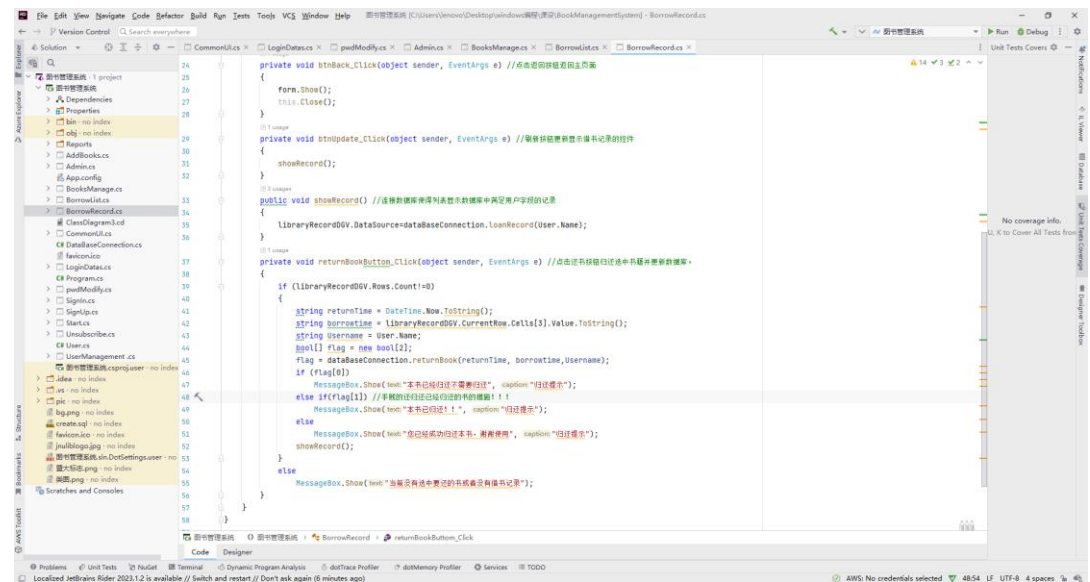


图 11 借书记录页面代码

4.3.5 普通用户首页 CommonUI.cs

主要实现普通用户的主页面，包括继承了 Form 的 CommonUI 类，主要实现的功能有：btnQuery_Click() 点击查看借书记录按钮跳转到借书记录页面；btnExit_Click() 点击退出按钮返回到登录页面；btnSearchBook_Click() 点击查找书名按钮在数据库中书籍表查找满足书名字段的记录；searchByKindButton_Click() 点击查找类别按钮在数据库中书籍表查找满足类别字段的记录；

booksDVG_CellClick()按刷新按钮对该页面进行刷新；showBook()使得控件实时显示数据库中满足条件的书目记录。代码约在 80 行左右。

```
36         else
37         {
38             if (dataBaseConnection.seekBookByName(bookName).Rows.Count == 0)
39                 MessageBox.Show(text: "查无此书", caption: "查询结果");
40             else
41                 booksDVG.DataSource = dataBaseConnection.seekBookByName(bookName);
42         }
43     }
44     [1 usage]
45     private void searchByKindButton_Click(object sender, EventArgs e) //点击查找类别按钮在数据库中书籍表查找满足类别字段的记录
46     {
47         string bookClass = bookClassComboBox.SelectedItem.ToString();
48         booksDVG.DataSource = dataBaseConnection.seekBookByClass(bookClass);
49     }
50     [1 usage]
51     private void booksDVG_CellClick(object sender, DataGridViewCellEventArgs e)
52     {
53         string bookName = booksDVG.CurrentRow.Cells[1].Value.ToString();
54         bookDigestTextBox.Text = dataBaseConnection.bookinfos(bookName)[0];
55     }
56     [1 usage]
57     private void borrowButton_Click(object sender, EventArgs e) //标记借书
58     {
59         string bookName = booksDVG.CurrentRow.Cells[1].Value.ToString();
60         string bookNumber = booksDVG.CurrentRow.Cells[0].Value.ToString();
61         string name = User.Name;
62         string borrowTime = DateTime.Now.ToString();
63         if (dataBaseConnection.borrowAdd(bookName, name, borrowTime, bookNumber))
64         {
65             MessageBox.Show(text: "您已成功借到此书, 请在一个月内归还", caption: "借书成功");
66             showBook();
67         }
68         else
69             MessageBox.Show(text: "您已借过该书且未归还", caption: "借书终止");
70     }
71     [3 usages]
72     public void showBook() //使得控件实时显示数据库中满足条件的书目记录
73     {
74         booksDVG.DataSource = dataBaseConnection.showBook();
75     }
76 }
```

图 12 普通用户页面

4.3.6 数据库链接功能模块 DataBaseConnect.cs

主要实现程序与 mysql 数据库的连接, 包括了一个基类 DataBaseConnection, 在所有需要连接数据库的类中均被调用。实现的功能有: Login()在登陆时连接数据库查找是否有该用户, 并返回登陆结果; LoginRecord()在登录成功时将登陆记录记录到数据库; loanRecord()在借书时将所需的书名、书号和结束用户记录到数据库的借书记录表中; showBook()显示数据库的书籍表; commonUser()在用户注册完成时将信息添加到数据库的用户表中; passwordChange()在用户修改密码后将对应用户名的密码项进行修改; seekBookByName()按照书名对数据库中的书目表进行查询; seekBookByClass 按照书类对数据库中的书目表进行查询; bookinfos()返回对应书目的所有字段; returnBook()修改数据库中的书目表的归还字段; seekUser()查找数据库中对应用户; deleteUser()删除数据库中的对应用户的记录; bookInfoChange()修改数据库中对应书籍的相关信息; bookDelete()删除数据库中对应书籍的记录; rollGRANT()回收数据库中相应用户的权限; GRANT()赋予数据库中相应用户的权限; addNewBook()将书目信息登记到数据库中; showborrow()显示数据库的借阅表; seekBookByName_borrow()显示数据库中满足名字字段的借阅书籍; seekBookByUserName_borrow()显示数据库中满足借书人字段的借阅书籍; borrowAdd()将借阅的书籍登记到数据库的借阅表中。代码大约有 600 行左右 (包括其他窗口按钮入口链接数据库功能的具体实现)。

```

550 + "and 是否归还=" + value + " ";
551
552 using (MySQLConnection userAddConnection = new MySqlConnection(connectionString))
553 {
554     userAddConnection.Open();
555     MySqlCommand AddCommand = new MySqlCommand(borrowBookCommand, userAddConnection);
556     MySqlCommand SeekCommand = new MySqlCommand(seekBookCommand, userAddConnection);
557     using (MySqlDataReader mySqlDataReader = SeekCommand.ExecuteReader())
558     {
559
560         if (mySqlDataReader.Read())
561         {
562             if (mySqlDataReader["是否归还"].ToString() == "否")
563             {
564                 IsborrowAdd = false;
565             }
566             else
567             {
568                 IsborrowAdd = true;
569             }
570         }
571         else IsborrowAdd = true;
572     }
573     if (IsborrowAdd)
574         AddCommand.ExecuteNonQuery();
575 }
576 return IsborrowAdd;
577 }
578 public DataTable seekLoginData(string userClass, string value)
579 {
580     string seekLoginDataByClassCommandText = "select * from loginrecord " +
581     "where " + userClass + " = '" + value + "'";
582     DataTable seekUsers;
583     using (MySQLConnection seekLoginDataConnection = new MySqlConnection(connectionString))
584     {
585         MySqlDataAdapter mySqlDataAdapter = new MySqlDataAdapter(seekLoginDataByClassCommandText, seekLoginDataConnection);
586         DataTable dataTable = new DataTable();
587         mySqlDataAdapter.Fill(dataTable);
588         seekUsers = dataTable;
589     }
590 }

```

图 13 数据库链接功能模块代码

4.3.7 启动页面 Start.cs

主要实现打开客户端时的主页面，包括继承 Form 的 Start 类。实现的功能有：linkLabel1_LinkClicked() 点击用户登录按钮后跳转到登陆页面；SignInlinkLabel_LinkClicked() 点击用户注册按钮后跳转到用户注册页面；UnsubscribelinkLabel_LinkClicked() 点击用户注销按钮后跳转到用户注销页面。代码约在 40 行左右。

```

7 10 public partial class Start : Form
8  {
9      1 usage
10     public Start()
11     {
12         InitializeComponent();
13     }
14
15     1 usage
16     private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) //点击用户登录按钮后跳转到登陆页面
17     {
18         loginlinkLabel.LinkVisited = true;
19         Form login = new SignIn(form: this);
20         login.Show();
21         Hide();
22     }
23
24     1 usage
25     private void SignInlinkLabel_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) //点击用户注册按钮后跳转到用户注册页面
26     {
27         SignInlinkLabel.LinkVisited = true;
28         Form signIn = new SignUp(form: this);
29         signIn.Show();
30         Hide();
31     }
32
33     1 usage
34     private void UnsubscribelinkLabel_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) //点击用户注销按钮后跳转到用户注销页面
35     {
36         /*
37         UnsubscribelinkLabel.LinkVisited = true;
38         Form unsubscribe = new Unsubscribe(this);
39         unsubscribe.Show();
40         Hide();
41         */
42     }
43
44     1 usage
45     private void Form1_Load(object sender, EventArgs e){}
46
47     1 usage
48     private void label2_Click_1(object sender, EventArgs e){}
49 }

```

图 14 开始页面代码

4.3.8 登陆页面 SignIn.cs

主要实现用户登录页面，包括继承 Form 的 SignIn 类。实现的功能有：CancelButton_Click() 点击取消按钮返回主页面；logInButton_Click/LoginInfo() 连接到数据库进行登陆确认；Clear() 清空输入的用户名和密码；passwordModificationButton_Click() 点击修改密码按钮跳转到修改密码页面；IsUserClassManage() 根据选项按钮返回用户类型(是否为管理员)。代码在 100 行左右。

```
44     name = nameTextBox.Text;
45     password = passwordTextBox.Text;
46     bool IsAdmin;
47     if (generalRadioButton.Checked)
48         IsAdmin = false;
49     else
50         IsAdmin = true;
51     CommonUI commonUI = new CommonUI(form: this, CommonName);
52     if (dataBaseConnection.Login(CommonName, Password, IsAdmin))
53     {
54         dataBaseConnection.LoginRecord(CommonName, DateTime.Now.ToString(format: "F"));
55         if (IsUserClassManage())
56         {
57             MessageBox.Show(text: "欢迎您登陆本系统", caption: "消息提示");
58             admin.Show();
59             this.Hide();
60             commonUI.Close();
61         }
62         else
63         {
64             MessageBox.Show(text: "欢迎您登陆本系统", caption: "消息提示");
65             commonUI.Show();
66             this.Hide();
67             admin.Close();
68         }
69     }
70     else
71     {
72         MessageBox.Show(text: "您的用户名或密码有误", caption: "登陆失败");
73         Clear();
74     }
```

图 15 登录界面代码实现

4.3.9 注册页面 SignUp.cs

主要实现用户注册页面，包括继承 Form 的 SignUp 类。实现的功能有：SignInInfo() 根据正则表达式确认输入的信息是否符合规范；Clear()/hintClear() 清空输入的内容；CancelButton_Click() 点击取消按钮返回上一给页面；signInButtom_Click() 点击注册按钮检验信息是否正确并登录到数据库中，并清空输入内容。代码大概在 140 行左右。

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
}
if (passwordTextBox.Text == String.Empty)
{
    hintNameLabel.Text = "用户密码不能为空";
}
if (!Regex.IsMatch(input: passwordTextBox.Text, pattern: passwordLimit ))
{
    hintNameLabel.Text = "用户密码只能由字母和数字组成";
    passwordTextBox.Clear();
    passwordTextBox.Focus();
    return;
}
if (passwordTextBox.Text != confirmPasswordTextBox.Text)
{
    hintNameLabel.Text = "前后密码不相同";
    passwordTextBox.Clear();
    confirmPasswordTextBox.Clear();
    passwordTextBox.Focus();
}
else
{
    name = nameTextBox.Text;
    if (manRadioButton.Checked )
        sex = manRadioButton.Text;
    else
        sex = womanRadioButton.Text;
    Studentname = nameTextBox2.Text;
    phoneNumber = phoneNumberTextBox.Text;
    password = passwordTextBox.Text;
    workUnit = workUnitTextBox.Text;
    userID = name;
    signInTime = DateTime.Today.ToString( format: "F");
    if (dataBaseConnection.userAdd(userName: CommonName, Studentname, Password, Sex,
        WorkUnit, PhoneNumber, UserId, SignInTime))
        MessageBox.Show( text: "感谢您的注册", caption: "消息提示");
    else
        MessageBox.Show( text: "用户已经存在", caption: "注册终止");
    Clear();
}

```

图 16 注册页面代码实现

4.3.10 添加新书页面 AddBooks.cs

主要实现新增书目页面，包括继承 Form 的 AddBooks 类。实现的功能有：
addBookButton_Click() 点击添加按钮检查输入信息是否正确，并连接数据库添加记录；getInfos() 获取文本框中的内容。该页面代码大概有 80 行。

```

43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
this.Close();
}
[0] 1 usage
private void addBookButton_Click(object sender, EventArgs e)
{
    getInfos();
    if (bookNameTextBox.Text == String.Empty || bookClassComboBox.Text == String.Empty
        || languageTextBox.Text == String.Empty || oldNameTextBox.Text == String.Empty ||
        bookNumberTextBox.Text == String.Empty || callNumberTextBox.Text == String.Empty ||
        authorTextBox.Text == String.Empty || ISBNTextBox.Text == String.Empty ||
        translatorTextBox.Text == String.Empty || publishDateTimePicker.Text == String.Empty ||
        pressTextBox.Text == String.Empty || contentTextBox.Text == String.Empty)
        MessageBox.Show( text: "所有资料不能留空", caption: "提示"); //检查所有信息是否都已经填写
    else
    {
        if (dataBaseConnection.addNewBook(BookName, BookClass, Language, OldName,
            BookNumber, CallNumber, Author, ISBN, Translator, Publish, Press, Content))
            MessageBox.Show( text: "你已经成功添加这本书", caption: "添加成功"); //调用dataBaseConnection对数据库进行插入
        else
            MessageBox.Show( text: "这本书已经存在, 无需添加", caption: "添加失败");
    }
}
[0] 1 usage
public void getInfos() //获取文本框控件中的内容
{
    bookName = bookNameTextBox.Text;
    bookClass = bookClassComboBox.Text;
    language = languageTextBox.Text;
    oldName = oldNameTextBox.Text;
    bookNumber = bookNumberTextBox.Text;
    callNumber = callNumberTextBox.Text;
    author = authorTextBox.Text;
    isbn = ISBNTextBox.Text;
    translator = translatorTextBox.Text;
    publish = publishDateTimePicker.Text;
    press = pressTextBox.Text;
    content = contentTextBox.Text;
}

```

图 17 添加书目功能代码实现

4.3.11 密码修改页面 pwdModify.cs

主要实现密码修改页面，包括继承 Form 的 pwdModify 类。实现的功能有：PwdModificationInfo() 点击确认修改按钮，当新密码符合规范时对用户密码进行修改。该页面的代码约在 70 行左右。

```
16 public pwdModify(Form form)
17 {
18     InitializeComponent();
19     this.form = form;
20 }
21 // 1 usage
22 public void PwdModificationInfo() // 点击确认修改按钮，当新密码符合规范时对用户密码进行修改。
23 {
24     if (nameTextBox.Text == String.Empty || passwordTextBox.Text == String.Empty)
25     {
26         hintLabel.Text = "用户名称或用户密码不能为空";
27         Clear();
28         return;
29     }
30     if (!Regex.IsMatch(input: nameTextBox.Text, pattern: nameLimit) ||
31         !Regex.IsMatch(input: passwordTextBox.Text, pattern: passwordLimit)) // 密码要符合字母和数字组成的正则表达式
32     {
33         hintLabel.Text = "用户名称和用户密码只能由字母和数字组成";
34         Clear();
35         return;
36     }
37     if (newPasswordTextBox.Text == String.Empty ||
38         !Regex.IsMatch(input: newPasswordTextBox.Text, pattern: passwordLimit))
39     {
40         hintLabel.Text = "新密码不能够为空，且密码又数字和字母组成";
41         newPasswordTextBox.Clear();
42         confirmNewPasswordTextBox.Clear();
43         return;
44     }
45     if (newPasswordTextBox.Text != confirmNewPasswordTextBox.Text) // 两个文本框内容应相同
46     {
47         hintLabel.Text = "新密码前后不同";
48         newPasswordTextBox.Clear();
49         confirmNewPasswordTextBox.Clear();
50     }
51 }
```

图 18 修改密码功能代码实现

4.3.12 程序启动类模块 Program.cs

该文件是整个程序的主入口，包含基类 Program。功能是对环境进行调用并且初始化主页面。

```
static void Main()
{
    if (Environment.OSVersion.Version.Major >= 6)
    {
        SetProcessDPIAware();
    }

    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(mainForm: new Start());
}
[System.Runtime.InteropServices.DllImport("user32.dll")]
// 1 usage
private static extern bool SetProcessDPIAware();
```

图 19 程序入口代码

4.3.13 登录记录页面 LoginDatas.cs

该文件主要实现管理员查询用户登陆记录页面,包括继承 Form 的 LoginDatas 类。实现的功能有: button1_Click()按下查询所有登陆记录后,在数据库中查询所有满足输入条件的登陆记录并返回给列表控件。代码大约在 50 行。

```
9      DataBaseConnection dataBaseConnection = new DataBaseConnection();
10     public 图书系统登录记录()
11     {
12         InitializeComponent();
13     }
14     private void signinDateTimePicker_ValueChanged(object sender, EventArgs e) { }
15     private void button1_Click(object sender, EventArgs e) //按下查询所有登陆记录后,在数据库中查询所有满足输入条件的登陆记录并返回给列表控件。
16     {
17         string userClass = "";
18         string value = "";
19         if (usersClassComboBox.SelectedItem.ToString() == "用户名")
20         {
21             userClass = "UserName";
22             value = userNameTextBox.Text;
23         }
24         if (usersClassComboBox.SelectedItem.ToString() == "姓名")
25         {
26             userClass = "StudentName";
27             value = nameTextBox2.Text;
28         }
29         if (usersClassComboBox.SelectedItem.ToString() == "性别")
30         {
31             userClass = "Sex";
32             value = sexTextBox.Text;
33         }
34         if (usersClassComboBox.SelectedItem.ToString() == "学院")
35         {
36             userClass = "WorkUnit";
37             value = workUnitTextBox.Text;
38         }
39         if (usersClassComboBox.SelectedItem.ToString() == "注册日期")
40         {
41             userClass = "SigninTime";
42             value = signinDateTimePicker.Value.ToString(format: "yyyy年M月dd日 0:00:00");
43         }
44         dataGridView1.DataSource = dataBaseConnection.seekUser(userClass, value);
```

图 20 登录记录功能实现

4.3.14 账号注销页面 Unsubscribe.cs

该文件主要实现账号注销页面,包括继承 Form 的 Unsubscribe 类。实现的功能主要有: infoConfirm()按下确认按钮后检查输入的用户名和密码是否有误,随后在数据库中删除对应的用户记录。代码约有 60 行。


```

15 10 public partial class Unsubscribe : Form
16 {
17     const string nameLinmit = @"^[a-zA-Z0-9]*$";
18     const string passwordLinmit = @"^[a-zA-Z0-9]*$";
19     private string name;
20     private string password;
21     Form form;
22     DataBaseConnection dataBaseConnection = new DataBaseConnection();
23
24     public Unsubscribe(Form form)
25     {
26         InitializeComponent();
27         this.form = form;
28     }
29     [1 usage]
30     public void infoConfirm() //按下确认按钮后检查输入的用户名和密码是否有误，随后在数据库中删除对应的用户记录。
31     {
32         if (nameTextBox.Text == String.Empty || passwordTextBox.Text == String.Empty) // 输入信息为空
33         {
34             Clear();
35         }
36         else if (!Regex.IsMatch(input: nameTextBox.Text, pattern: nameLinmit) ||
37             !Regex.IsMatch(input: passwordTextBox.Text, pattern: passwordLinmit)) // 用户或者密码不符合规范
38         {
39             Clear();
40         }
41         else
42         {
43             name = nameTextBox.Text;
44             password = passwordTextBox.Text;
45
46             if (dataBaseConnection.UserLogout(name, password))
47             {
48                 MessageBox.Show(text: "账户成功注销!", caption: "注销成功");
49             }
50             else
51             {
52                 MessageBox.Show(text: "账户信息有误.", caption: "操作失败");
53             }
54         }
55     }
56     [2 usages]

```

图 21 账号注销功能实现

4.3.15 用户管理页面 UserManagement.cs

该文件主要实现用户管理页面，包括继承 Form 的 UserManagement 类。实现的功能主要有：seekUserButton_Click()按下查询按钮返回满足输入的查询条件的用户，并显示在列表控件中；showUsers()将数据库中查询的用户记录返回到列表控件中显示；userDelteButton_Click()按下删除按钮将选中用户从数据库中删除；button1_Click()/utton2_Click()按下权限授予和权限回收按钮将改变列表中选中的用户的权限。代码约有 100 行。

```

62     {
63         usersDGV.DataSource= dataBaseConnection.seekUser(userClass, value);
64     }
65 }
66 // 1 usage
67 private void userDeleteButton_Click(object sender, EventArgs e) //按下删除按钮将选中用户从数据库中删除
68 {
69     string name = usersDGV.CurrentRow.Cells[1].Value.ToString();
70     dataBaseConnection.deleteUser(name);
71     MessageBox.Show(text: "用户删除成功", captions: "操作成功");
72     showUsers();
73 }
74 // 1 usage
75 private void usersDGV_CellContentClick(object sender, DataGridViewCellEventArgs e) { }
76 // 1 usage
77 private void button1_Click(object sender, EventArgs e) //按下权限授予和权限回收按钮将改变列表中选中的用户的权限。
78 {
79     string name = usersDGV.CurrentRow.Cells[1].Value.ToString();
80     string is_admin = usersDGV.CurrentRow.Cells[0].Value.ToString();
81     if (is_admin != "管理员") // 根据用户的类型来操作
82     {
83         if (!dataBaseConnection.GRANT(name)) //修改用户的权限
84         {
85             MessageBox.Show(text: "授予用户" + usersDGV.CurrentRow.Cells[2].Value.ToString() + "管理权限.", captions: "权限授予");
86         }
87     } else{
88         if(!dataBaseConnection.rollGRANT(name)){
89             MessageBox.Show(text: "收回用户" + usersDGV.CurrentRow.Cells[2].Value.ToString() + "管理权限.", captions: "权限收回");
90         }
91     }
92     showUsers();
93 }
94 private void button2_Click(object sender, EventArgs e) { }
95 // 1 usage
96 private void groupBox1_Enter(object sender, EventArgs e) { }
97 }
98 }

```

图 22 管理用户功能实现

4.3.16 用户类模块 User.cs

该文件定义一个基类 User，主要描述用户的基本属性。在 BorrowRecord 类和 CommonUI 中均被调用。该文件只有 13 行代码。

```

1 //该文件定义一个基类User，主要描述用户的基本属性。在BorrowRecord类和CommonUI中均被调用。
2 namespace 图书管理系统
3 {
4     // 4 usages
5     class User
6     {
7         private string name;
8         // 4 usages
9         public string Name { get { return name; } }
10        // 2 usages
11        public User(string name)
12        {
13            this.name = name;
14        }
15    }
16 }

```

图 23 用户类模块代码

4.3.17 暨大图书馆官网页面 browser.cs

该文件定义一个浏览器页面，包括一个初始化方法。实现的功能是在打开窗

口的同时通过浏览器控件浏览暨南大学图书馆官网。在主页面的标签按钮点击调用。代码为 40 行。

```
22 private void browser_Load(object sender, EventArgs e)
23
24     string url = "https://lib.jnu.edu.cn/"; //暨大官网http链接
25     var request = (HttpWebRequest)WebRequest.Create(url); //Create:创建WebRequest对象获取和操作HTTP请求
26     var response = (HttpWebResponse)request.GetResponse(); //GetResponse:获取和操作HTTP应答
27     //构造数据流对象实例
28     Stream stream = response.GetResponseStream(); //GetResponseStream:获取应答流
29     StreamReader sr = new StreamReader(stream); //从字节流中读取字符
30
31     //从流当前位置读取到末尾并显示在WebBrowser控件中
32     string content = sr.ReadToEnd();
33     webBrowser1.ScriptErrorsSuppressed = true; //防止脚本报错
34     webBrowser1.DocumentText = content;
35
```

图 24 实现浏览器功能代码

4.4 系统的类之间的关系

通过定义和理解类之间的关系，可以更好地组织和设计系统的结构，促进类的复用和模块化开发，提高系统的可扩展性和维护性。比方说在 Windows Forms 应用开发中，Form 类对几乎所有窗口设计都是必要的；另外 DataBaseConnection 类在所有需要对数据库进行增删改查的模块中都几乎用到。在该项目中各类之间的调用关系比较复杂，可以使用 Visual Studio 生成类图表示：



图 25 图书管理系统类图

5. 程序运行

经过对程序的编写和反复调试，最终运行结果以及程序操作流程如下。当开始运行程序时，跳转到主页面，用户需要选择登录、注册或注销。

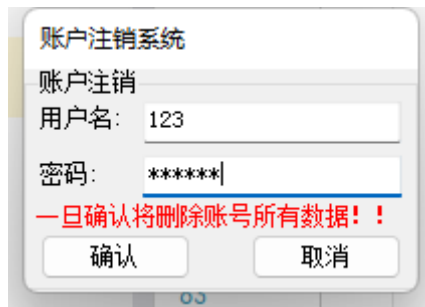


图 26 主页面效果图

当用户选择注册时，跳转到注册页面，新用户需要输入相应信息，点击注册后，检查信息符合规范的用户将注册成功并登记到数据库。当用户选择注销时，则会根据对应的用户名和密码，删除数据库中相应的用户记录。

The screenshot shows a registration form titled "欢迎注册" and "学生用户注册系统". The form contains several input fields: "用户名:" (Username), "姓名:" (Name), "性别:" (Gender) with radio buttons for "男" (Male) and "女" (Female), "学院:" (College), "联系手机:" (Contact Mobile), "用户密码:" (User Password), and "确认密码:" (Confirm Password). To the right of the form is the Jinan University logo, which is a circular emblem with the university's name in Chinese and English, and the year "1906". At the bottom right of the form are two buttons: "注册" (Register) and "取消" (Cancel).

图 27 注册页面效果图



The image shows a window titled "账户注销系统" (Account Deletion System). Inside, there's a sub-header "账户注销" (Account Deletion). Below it, there are two input fields: "用户名:" (Username) with the value "123" and "密码:" (Password) with the value "*****". A red warning message "一旦确认将删除账号所有数据!!" (Once confirmed, all data of the account will be deleted!!) is displayed in red text. At the bottom, there are two buttons: "确认" (Confirm) and "取消" (Cancel).

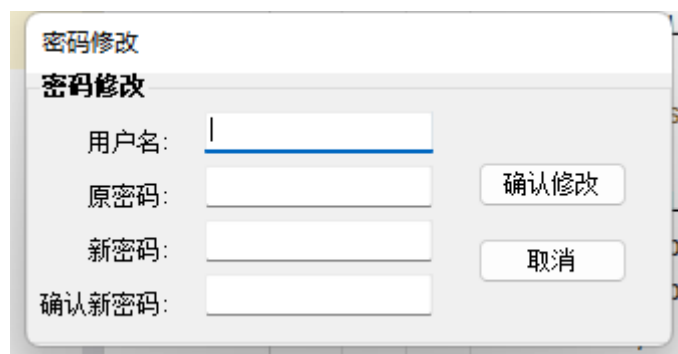
图 28 注销账号页面效果图

当读者选择登录按钮时，可以输入用户名和密码，并且选择以普通用户或者是管理员的模式登录，当输入的用户名或密码不正确时，会返回不正确的提示，否则跳到相应的用户界面。同时该页面支持跳转到修改密码。



The image shows a window titled "请登录" (Please Login). Inside, there's a sub-header "图书馆用户登陆界面" (Library User Login Interface). Below it, there are two input fields: "用户名:" (Username) with the value "root" and "用户密码:" (User Password) with the value "*****". Below the password field, there are two radio buttons: "普通用户" (Ordinary User) and "管理员" (Administrator), with the "管理员" button selected. At the bottom, there are three buttons: "登陆" (Login), "修改密码" (Change Password), and "取消" (Cancel).

图 29 登陆页面效果图



The image shows a window titled "密码修改" (Password Modification). Inside, there's a sub-header "密码修改" (Password Modification). Below it, there are four input fields: "用户名:" (Username) with the value "1", "原密码:" (Original Password), "新密码:" (New Password), and "确认新密码:" (Confirm New Password). To the right of the "原密码:" field is a button "确认修改" (Confirm Modification). To the right of the "新密码:" field is a button "取消" (Cancel).

图 30 修改密码页面效果图

如果是以普通用户的身份登录到该图书管理系统中，窗口主要呈现查书、借书和查看借书记录的功能。点击借书按钮，可以知道该用户有哪些书没有还，并提供快捷的一键借还书服务。在用户界面还可以按照书名或者书籍类别对书库进行查询。

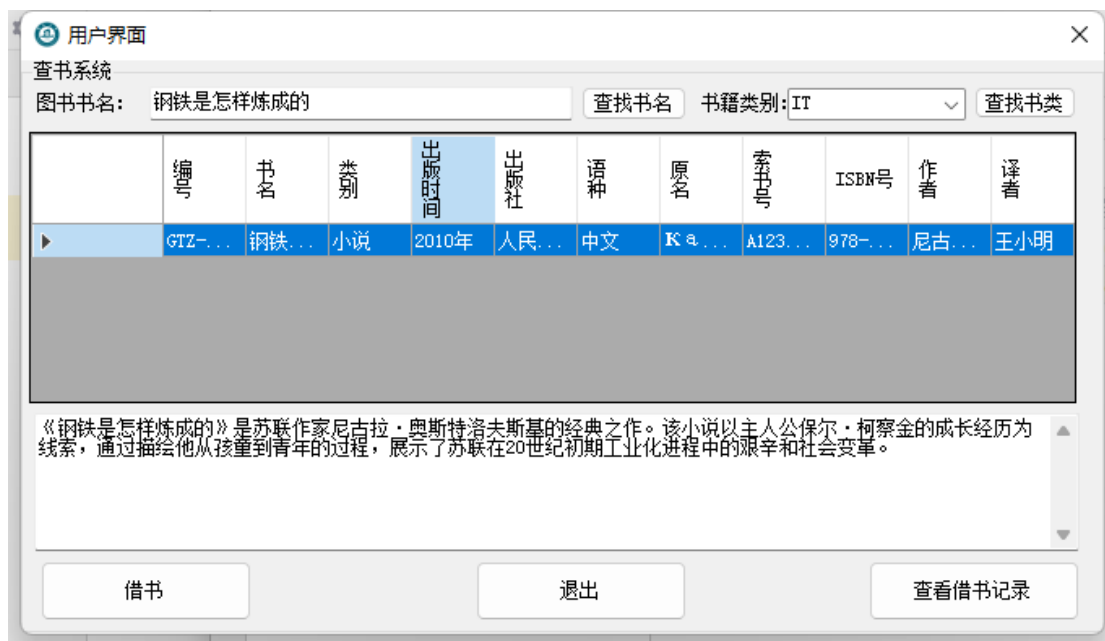


图 31 普通用户主界面效果图

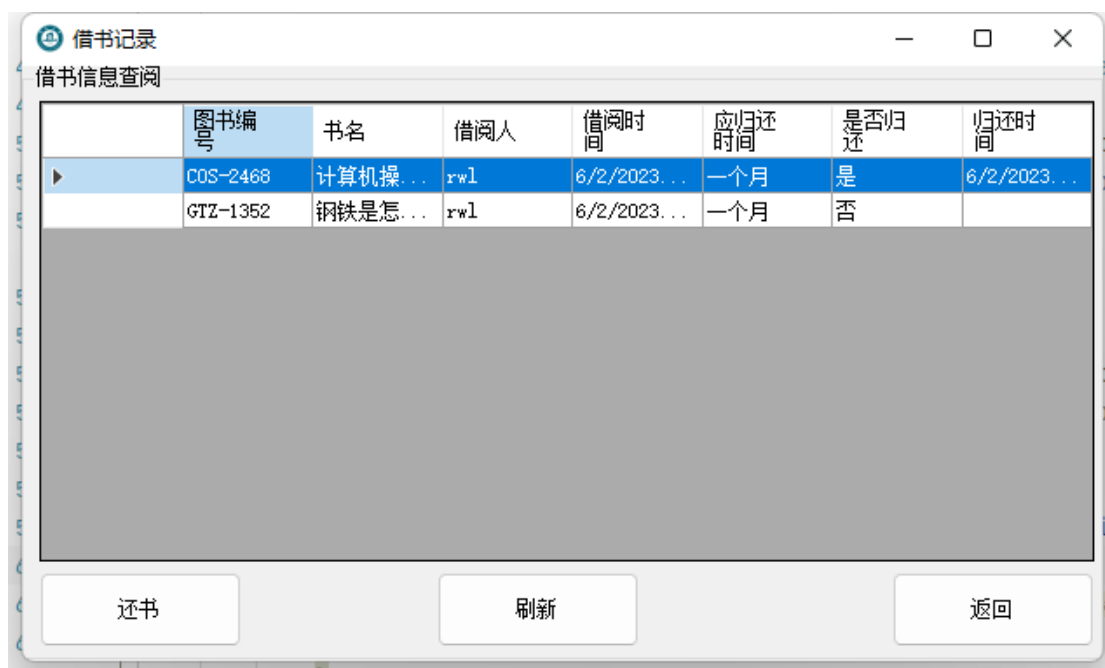


图 32 借书记录窗口效果图

如果是管理员的身份登录到图书管理系统中，那么在主页面可以跳转到用户管理和图书管理窗口。在用户管理系统的窗口中。可以看到所有已注册普通用户的信息，并且支持用户的条件查询、删除记录以及决定该用户的管理员权限。



图 33 图书管理菜单页面效果

用户管理

	权限	用户名	姓名	性别	学院	手机号码	注册时间
▶	管理员	121212	121212	男	信科院	1379082...	Friday, ...
	普通用户	123	123	男	123	1231231...	Friday, ...
	普通用户	rw1	rw1	男	信科院	1367254...	Friday, ...
	普通用户	rwlinno	rw1	男	信科院	1367254...	Friday, ...
	普通用户	test	wby	男	商学院	1145141...	Friday, ...
*							

用户查询

通过 用户名

用户名

姓名

性别

学院

注册日期 Friday , June 2,

查询

操作

删除

返回

权限授予

权限回收

图 34 用户管理系统页面效果图



图 35 图书管理系统页面效果图

在图书管理系统中，则可以看到数据库中的所有书籍，可以实现对所有书籍信息的增删改查。此外，可以根据书名或者借阅人查询对应的借书记录。对于选中相应书记记录的查询、修改和删除操作，均可在该页面进行，而添加书目则在另外一个窗口。



图 36 借书记录页面的效果图

添加书目

新增书目

书名: 软件测试

语种: 英文

原名: Software Testing

索书号: C543210987

作者: Dorothy Graham

图书类别: IT

译者: Erik van Veenendaal

编号: ST-7890

出版社: 机械工业出版社

ISBN号: 978-7-8765-4321-0

出版时间: Tuesday , November 9

添加

返回

内容摘要:

《软件测试》是一本权威的软件工程教材，由Dorothy Graham、Erik van Veenendaal、Isabel Evans和Rex Black共同编写。本书全面介绍了软件测试的基本概念、原理和最佳实践，是学习和实践软件测试的重要参考资料。

图 37 添加书目页面效果图

书号	书名	类别	语种	原名	编号	索书号	作者	ISBN号	译者
1	计算机操作系统	教材	中文	Computer Operation Systems	COS-2468	B987654321	安德鲁·S·坦尼斯鲍	978-7-1234-5678-9	佚名
2	软件测试	IT	英文	Software Testing	ST-7890	C543210987	Dorothy Graham	978-7-8765-4321-0	Erik van
3	钢铁是怎样炼成的	小说	中文	Как закалялась сталь	GTZ-1352	A123456789	尼古拉·奥斯特洛夫斯基	978-7-5078-1234-5	王小明

图 38 添加书目后数据库的变化

以上便是该图书馆里系统程序运行的大致效果，为了更好体现程序运行的具体流程和互动以及向读者介绍操作流程，本人特地制作了一个简短的操作流程视频放于目录下。尽管在仓促中完成的图书馆里系统仍有许多尚未实现的功能，该项目在应用开发方面以大体完成课程设计所需内容。

6. 总结

Windows 应用开发的意义在于为用户提供丰富、便捷和创新的应用程序，提升用户体验，推动技术进步。通过开发 Windows 应用，开发者可以为用户提供各种功能和特性丰富的应用程序，满足不同需求。优化的用户界面和交互设计提高用户满意度，简化操作流程，提升用户体验。同时，应用开发鼓励创新和技术进步，推动开发者不断探索新领域和实验性功能，提高应用质量和功能。此外，Windows 应用开发构建了一个庞大的生态系统，包括开发者、用户和第三方服务提供商，为各方带来经济利益和合作机会。总之，Windows 应用开发的意义在于创造更好的用户体验，推动技术发展，促进合作和创新。

完成图书管理系统开发并非一件容易的事，在实现期间我遇到了一些困难。比方说结合数据库技术，使用 MySQL 进行数据库连接时出现了很多错误；还有操控大量的控件是一项乏味的工作，容易让人凌乱；在面向对象这一方面，由于先前知识的欠缺，在制作过程中类的关系设计的并不是很合理；此外还有大大小小的代码错误……通过查阅资料、请教同学以及花费大量时间进行调试，最终也基本解决了先前遇到的问题。另外也在空余时间美化了 UI 和撰写项目报告。最终写到总结部分也是成就满满。

一开始我以为只需要对 Visual Studio 进行学习，但是我认识到要完成该项目是需要多项工具的参与的。因此我最终主要以 JetBrains Rider 和 JetBrains Datagrip 作为主要的编写环境，使用 Visual Studio 生成类图并且使用 Process On 等工具生成图表。按照项目涉及的技术总的来说，我对 .NET 技术以及 C# 语言的基本语法和面向对象编程思想有了更进一步的理解和掌握，还学会了如何使用其他技术开发 Windows 应用程序。

在此感谢王娜老师的辛勤付出，以及各位同学的倾囊相助。经过一个学期 Windows 编程的课程学习，我收获颇丰。在制作图书管理系统的过程中，累计了大量的理论基础和实战经验，锻炼了自己的编程思维和解决问题的能力。在此期间，我不仅学到了实际的技术知识和技能，还完成了完整的 windows 应用开发项目，这些经验对我未来的学习和工作都将其积极的推动作用。在未来我也将继续提高应用开发技术水平，并积极参与各种项目实践，做到学以致用。

7. 附录

为了提升用户体验，在程序完善过程中使用 Visual Studio2022 添加了 tab 顺序。这样便可以实现用户切换控件快捷键的需要并且弥补了 Windows Forms 控件顺序混乱的问题。

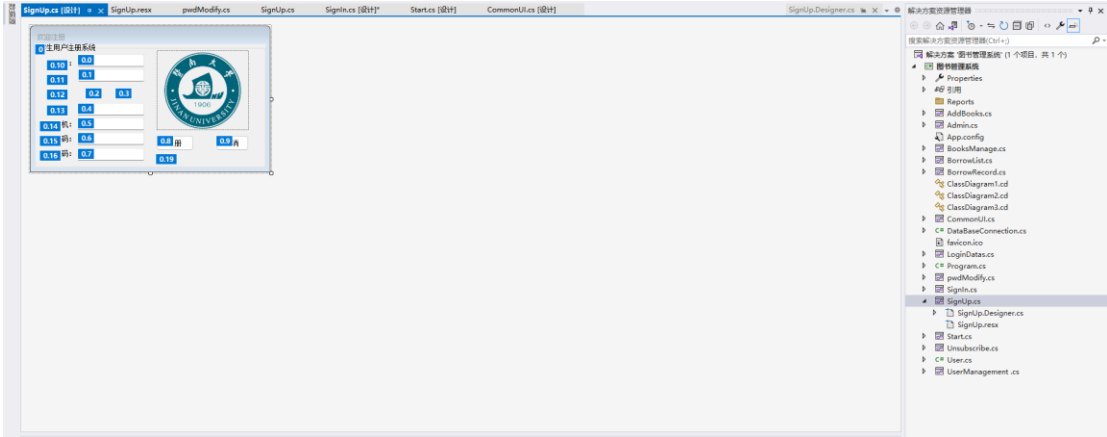


图 39 使用 Visual Studio 添加 Tab 顺序

此外，本项目中的程序中管理员是通过对用户授权产生的，因此最开始的管理员需要在本地数据库中写入。运行在项目目录下的 `create.sql` 查询指令不仅可以初始化数据库，还会产生第一个管理员用户名=root，密码=123456。

为了使得功能更多样化，该项目又追加了一个浏览器窗口可以访问暨大图书馆。链接标签放置在了主页面，通过点击事件创建浏览器页面，并实现无样式的暨大图书馆主页访问。



图 40 浏览器页面

参考文献

- [1] 周纳. ASP 动态网站编程与应用. 北京. 清华大学出版社. 2005. 8
- [2] 许春勤. 基于 ADO.NET 技术实现《图书管理系统》[J]. 南方农机, 2019, 50(01):113.
- [3] 周涛, 徐晓. 基于 SQL Server 与 C#的图书管理系统的设计与开发[J]. 智能计算机与应用, 2018, 8(04):199-202.
- [4] 李少玲. 图书馆项目管理信息化平台的设计及实现[J]. 信息系统工程, 2022(12):125-128.
- [5] 洪健. C#中的数据库编程技术研究[J]. 计算机光盘软件与应用, 2014, 17(17):265-266.
- [6] 王丹. 基于 ASP.NET 技术的图书管理系统的设计与实现[D]. 吉林大学, 2018.