

数据库系统系统实验报告

21311471 梁珏鸣

1.实现功能说明

用户注册

- i. 用户输入用户名、性别、住址、手机号码(有一定格式控制)等完善个人信息；
- ii. 用户需要二次确认密码；
- iii. 用户可以选择三种身份(学生、教职工、管理员)进行注册。

用户注销(非管理员)

- i. 用户输入用户名与密码；
- ii. 点击"确定注销"按钮后进行注销。

修改密码

- i. 用户输入用户名、原密码、新密码、二次确认新密码；
- ii. 验证通过后则修改成功。

用户登录

- i. 学生只能以学生身份登录；
- ii. 教职工只能以教职工身份登录；
- iii. 管理员可以以任意身份登录。

普通用户查询/借阅图书

- i. 用户可根据不同搜索方式搜索图书；
- ii. 搜索失败进行提示；
- iii. 点击图书显示对应简介；
- iv. 选中图书后可点击借阅，不可重复借阅同一本书；
- v. 使用事务实现借阅，借阅后会留下借阅记录，保存图书状态，更新馆藏信息。

普通用户归还/挂失图书

- i. 用户若在期限内没有损坏/丢失书籍，可直接归还；
- ii. 若损坏书籍，应按原价赔偿；
- iii. 丢失书籍应按双倍赔偿；
- iv. 凡未按期归还/损坏/丢失书籍的不能直接归还，应联系管理员归还；
- v. 使用事务实现归还，归还后借阅记录不会删除。

读者信息管理

- i. 打开界面的同时直接加载信息，管理员可搜索用户；
- ii. 管理员可删除普通用户；
- iii. 管理员可对除自己和超级管理员以外的用户进行授权/权限回收；
- iv. 管理员可设置学生/教职工身份。

图书信息管理

- i. 打开界面的同时直接加载信息，管理员可搜索图书；
- ii. 管理员可修改/删除/添加图书信息；
- iii. 管理员可以修改图书馆藏信息。

出版社/类别信息管理

- i. 打开界面的同时直接加载信息，管理员可增加/删除/修改出版社信息；
- ii. 被引用的出版社不得被删除/修改。

借阅信息/罚款办理

- i. 打开界面的同时直接加载信息，管理员可查询/删除借阅信息；
- ii. 管理员确认提交罚款后可办结罚款事务，使用事务实现；
- iii. 管理员可以帮助用户借书/还书，还书后罚款信息仍会记录在数据库中。

登录记录管理

- i. 打开界面的同时直接加载信息，管理员可查询登录记录。

2.界面展示

(1)首页(系统主界面)

登陆界面

图书管理系统(Copyright @Sober 2023 All Rights Reserved)

UserName

Password

☒ Student

☐ Faculty

☐ Admin

Login

注册界面

图书管理系统(Copyright @Sober 2023 All Rights Reserved)

UserName

Department

PhoneNumber

Address

Password

Confirm Password

Sex

☒ male

☐ female

Character

☒ student

☐ faculty

☐ admin

Age

0

Register

修改密码界面

图书管理系统(Copyright @Sober 2023 All Rights Reserved)

UserName

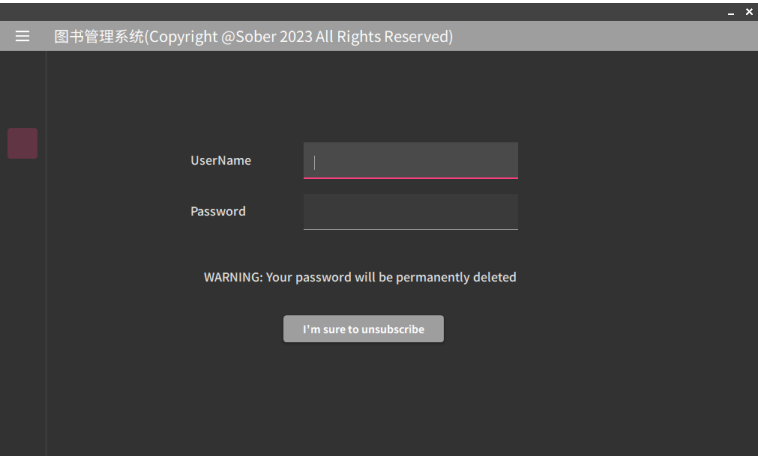
OriginPwd

NewPwd

ConfirmPwd

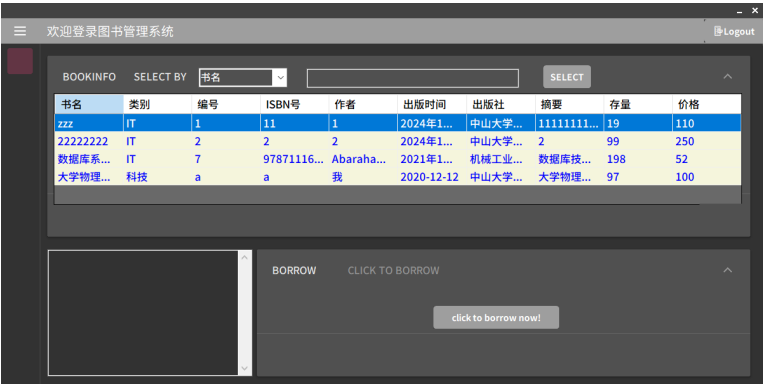
I'm sure to change

注销界面

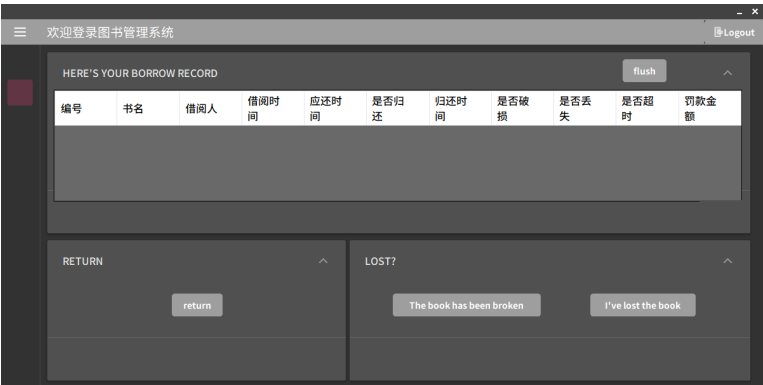


(2)普通用户界面

图书查询/借阅界面

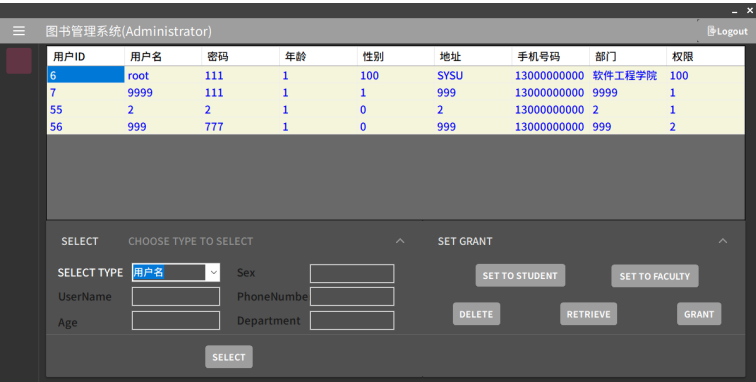


图书归还/挂失界面



(3)管理员界面

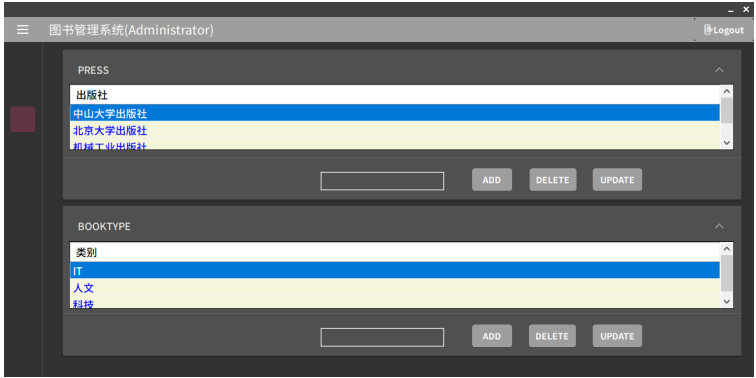
用户信息管理界面



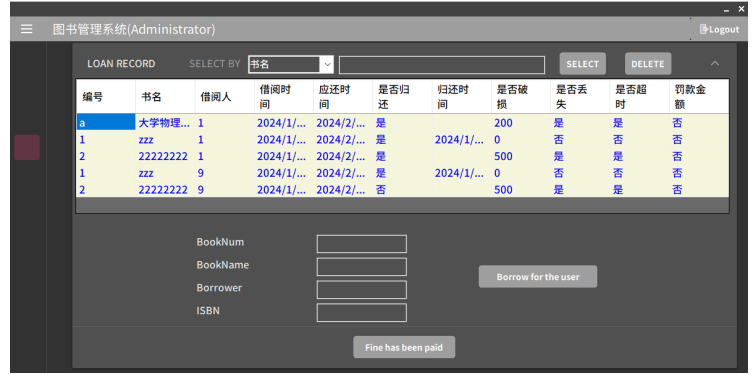
图书信息管理界面



出版社/图书类别管理界面



借阅记录管理界面



登录记录管理界面

UserName	LoginTime
1	2024年1月25日 18:57:48
2	2024年1月25日 18:59:35
999	2024年1月25日 18:59:44
999	2024年1月25日 19:03:11
999	2024年1月25日 19:03:18
1	2024年1月25日 19:04:43
2	2024年1月25日 19:06:11
999	2024年1月25日 19:06:28
999	2024年1月25日 19:09:49
999	2024年1月25日 22:35:33
999	2024年1月25日 22:46:28
999	2024年1月25日 22:58:11
999	2024年1月25日 23:02:17
999	2024年1月25日 23:02:47
1	2024年1月25日 23:56:34

3.功能演示

(1)用户注册

- i. 学生注册，用户名没有被注册过则提示注册成功：

图书管理系统(Copyright @Sober 2023 All Rights Reserved)

UserName: 6, Department: 6, PhoneNumber: 13099999999, Address: 6, Password: [masked], Sex: ☒ male ☐ female, Age: 0

Register

消息提示: 注册成功, 确定

- ii. 教职工注册，用户名没有被注册过则提示注册成功：

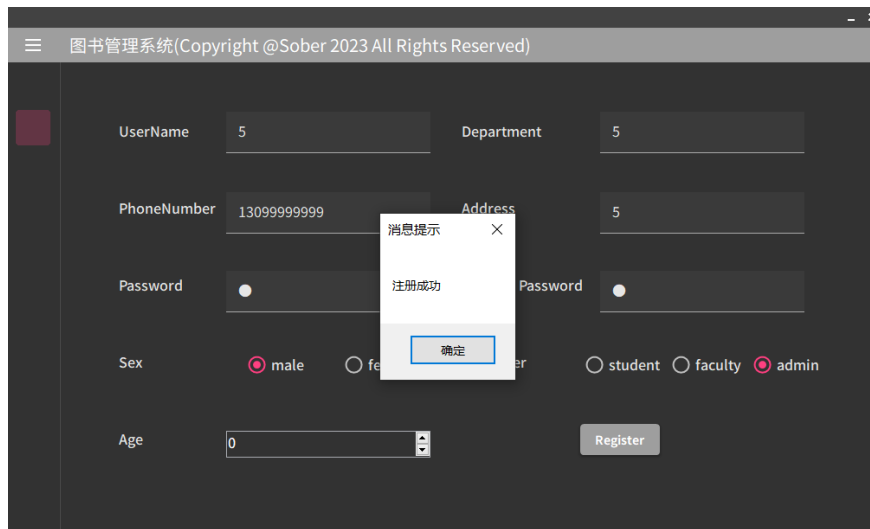
图书管理系统(Copyright @Sober 2023 All Rights Reserved)

UserName: 7, Department: 7, PhoneNumber: 13099999999, Address: 7, Password: [masked], Sex: ☒ male ☐ female, Age: 0

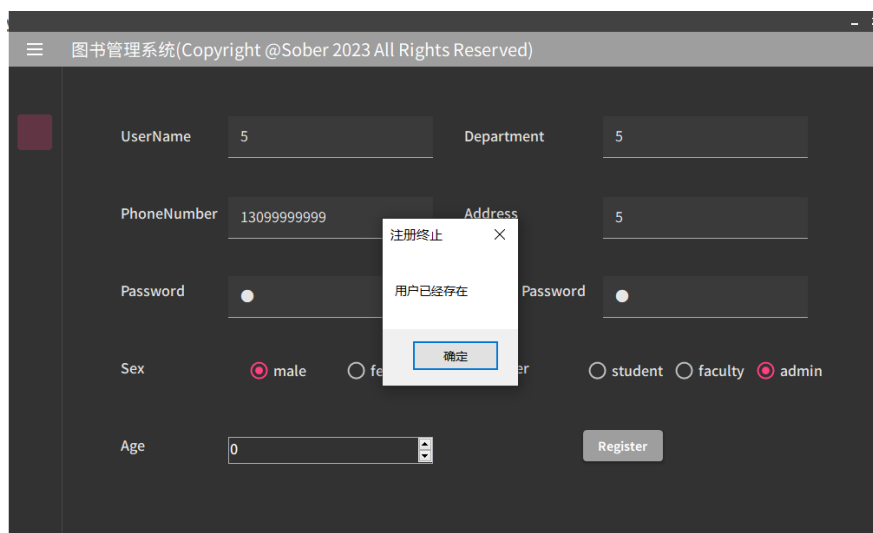
Register

消息提示: 注册成功, 确定

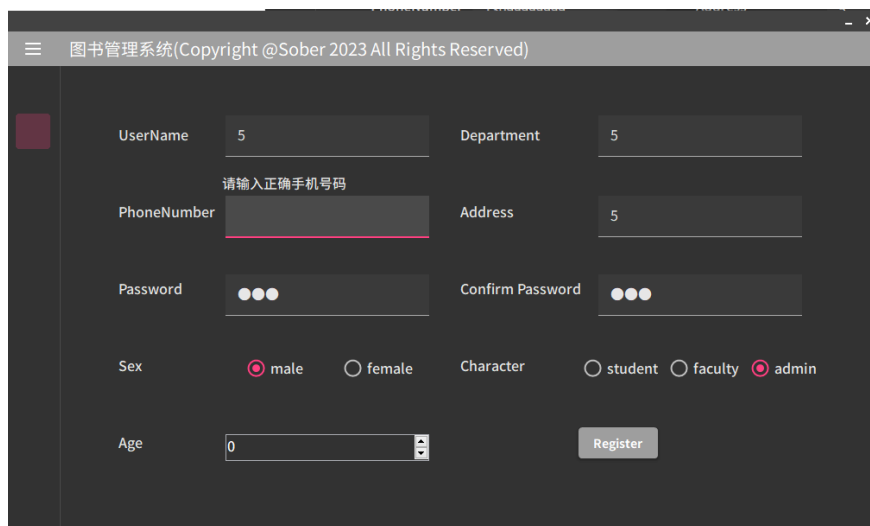
- iii. 管理员注册，用户名没有被注册过则提示注册成功：



iv. 禁止重复注册，这种情况会显示用户已存在：

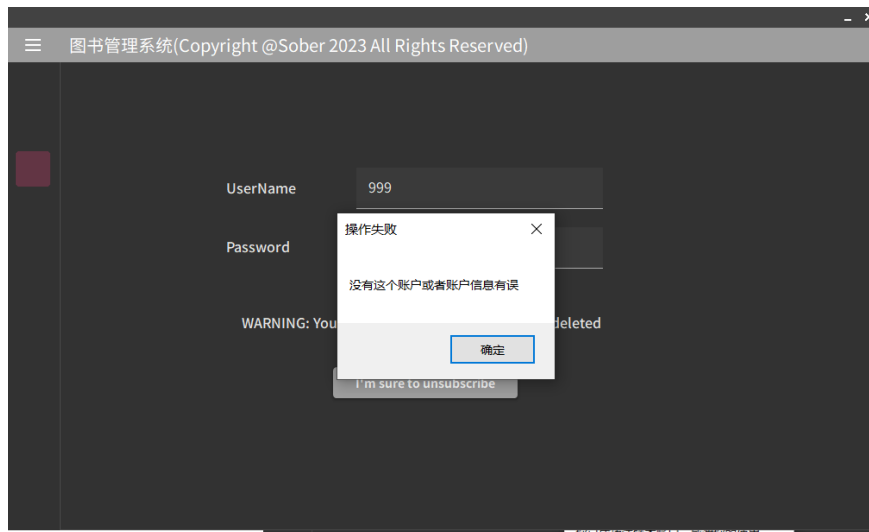


v. 手机号码、密码等需按照一定格式，否则对应输入框会被清空且提示：

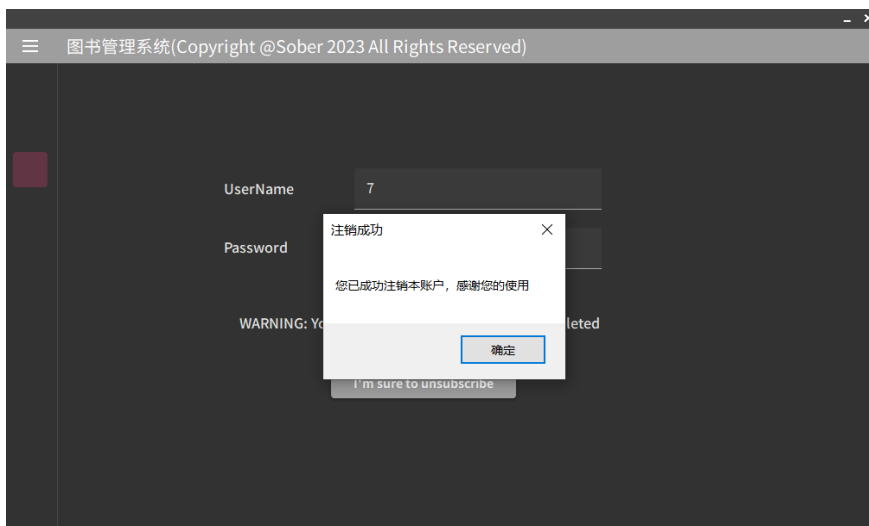


(2)用户注销(非管理员)

i. 管理员不得注销账号，否则提示账户信息有误：

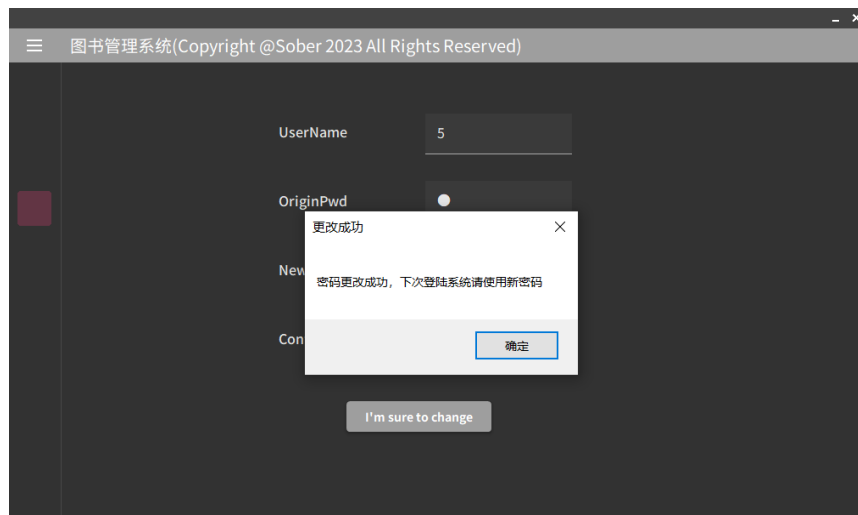


ii. 普通用户可正常注销，输入正确信息则提示注销成功：

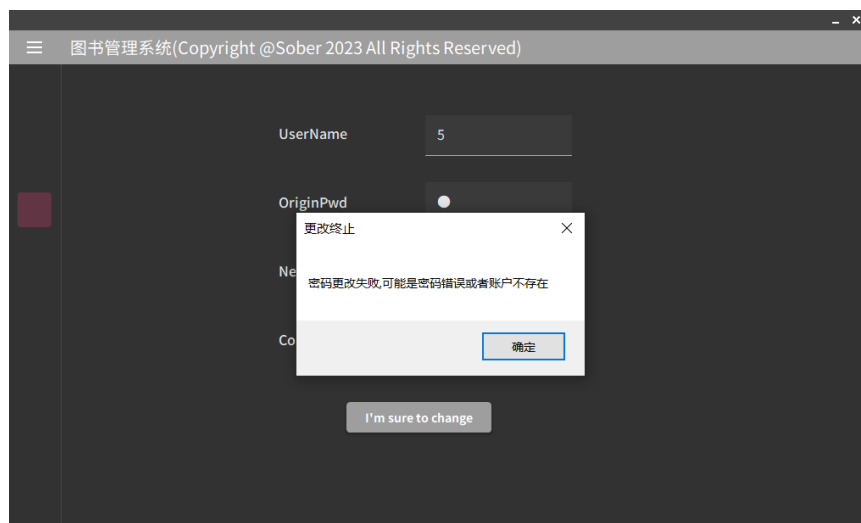


(3)修改密码

i. 普通用户/管理员修改密码都是一样的程序，输入正确的信息就会提示更改成功：

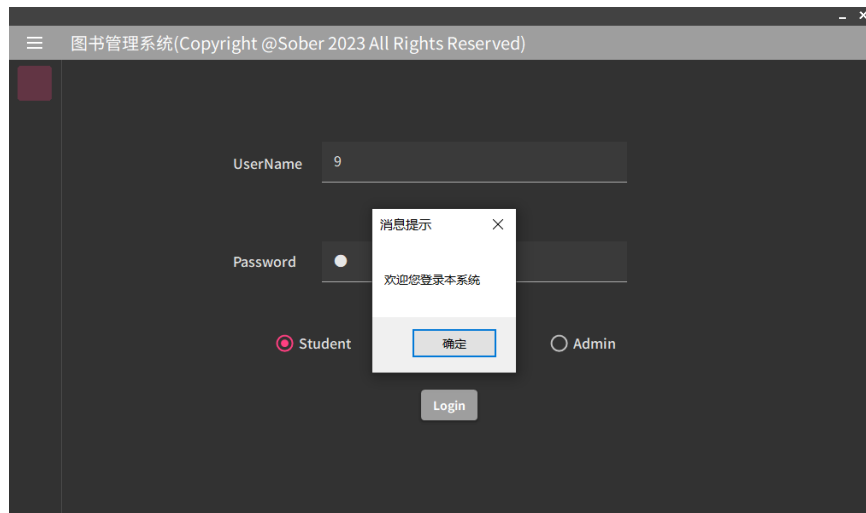


ii. 验证错误不能修改密码，并进行错误提示：

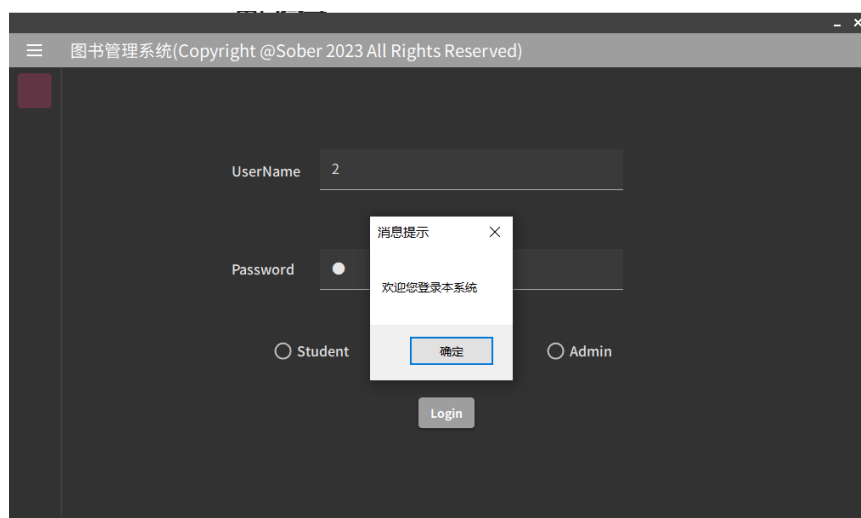


(4)用户登录

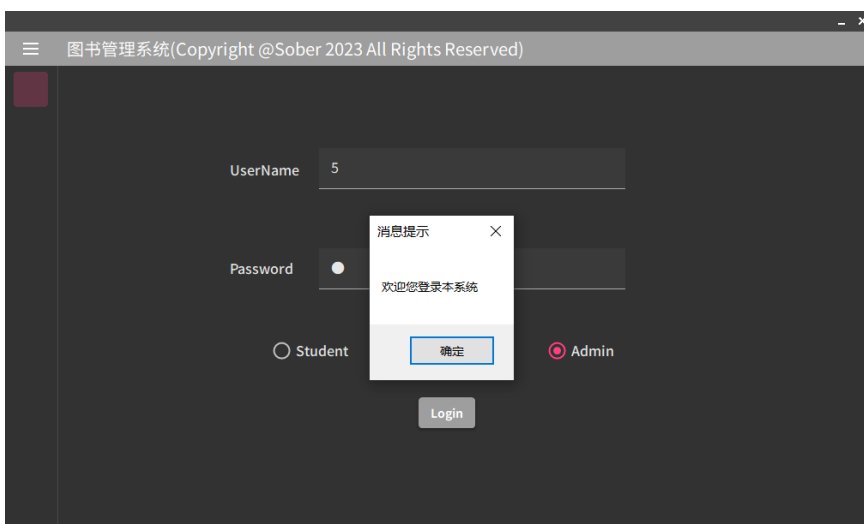
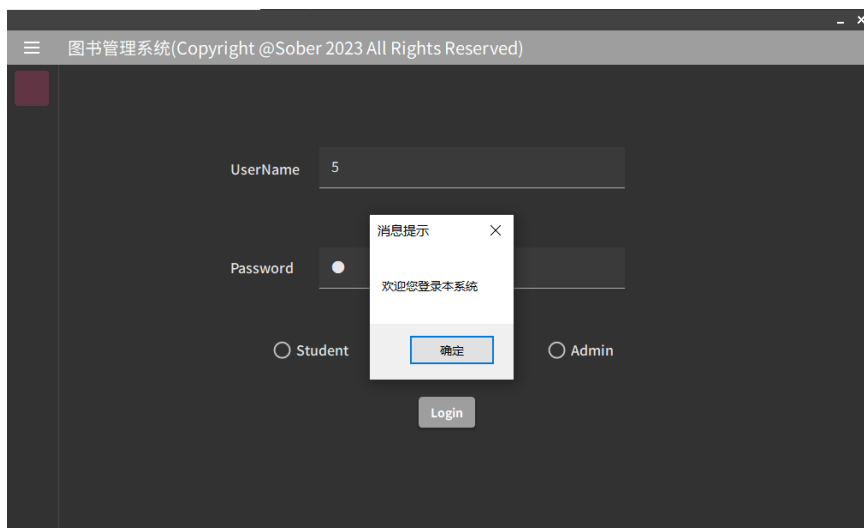
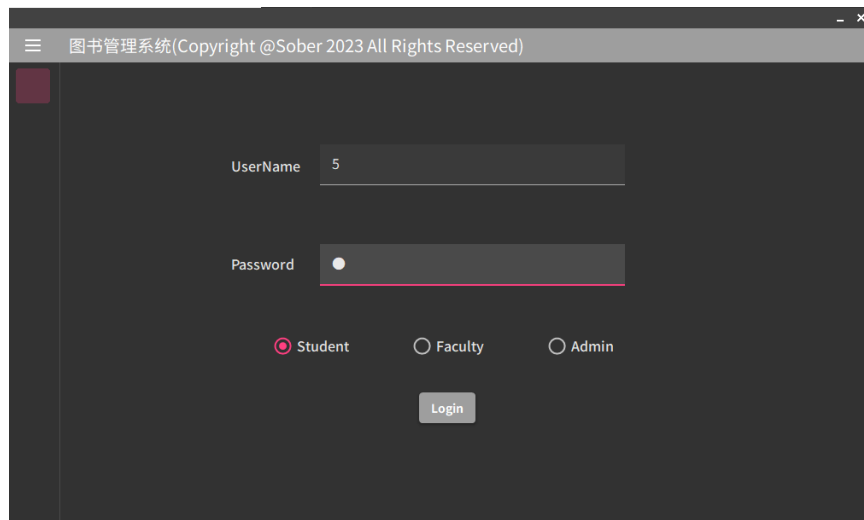
- i. 学生只能在学生通道登录，其他两个通道都会报错：



- ii. 教职工只能在教职工通道登录，其他两个通道都会报错：

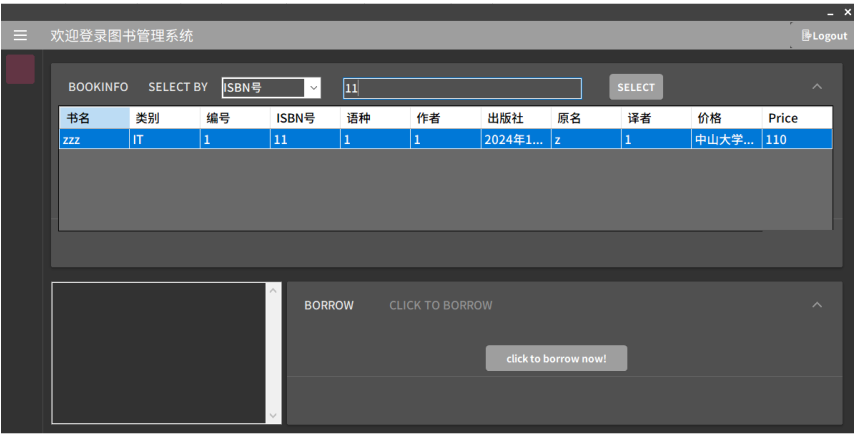


- iii. 管理员可在三种通道登录，因为管理员本身可以管理各种信息，所以放开权限：

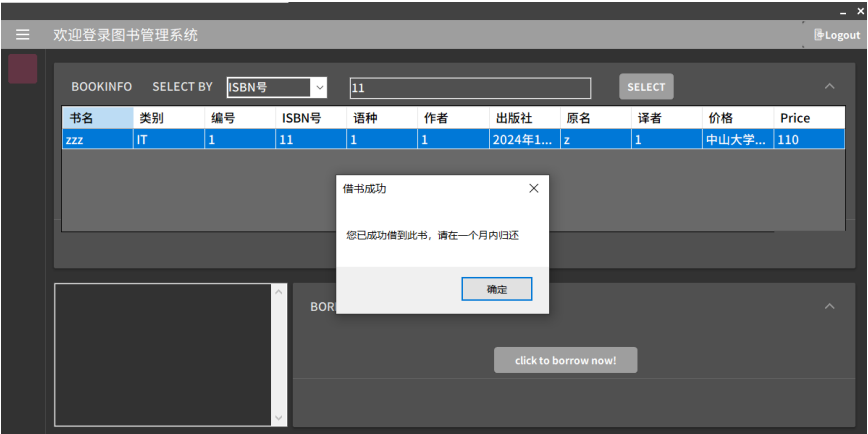


(5)普通用户查询/借阅图书

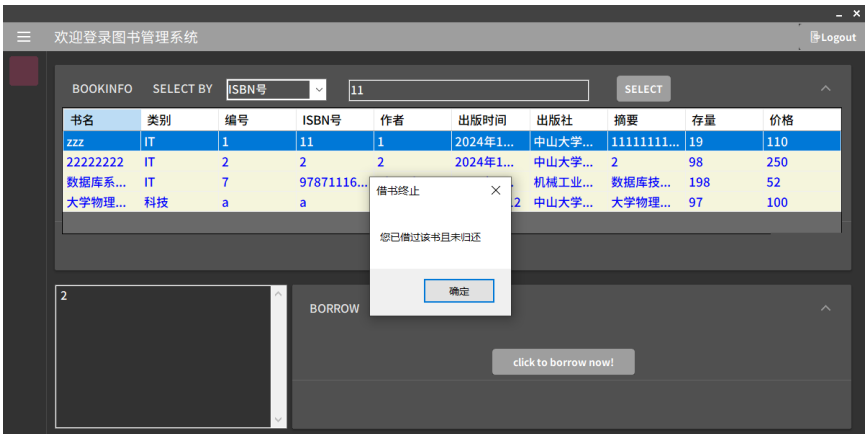
- i. 查询图书，直接显示图书信息：



ii. 借阅图书，借阅成功则提示归还时间，并且更新馆藏信息使图书存量减少：

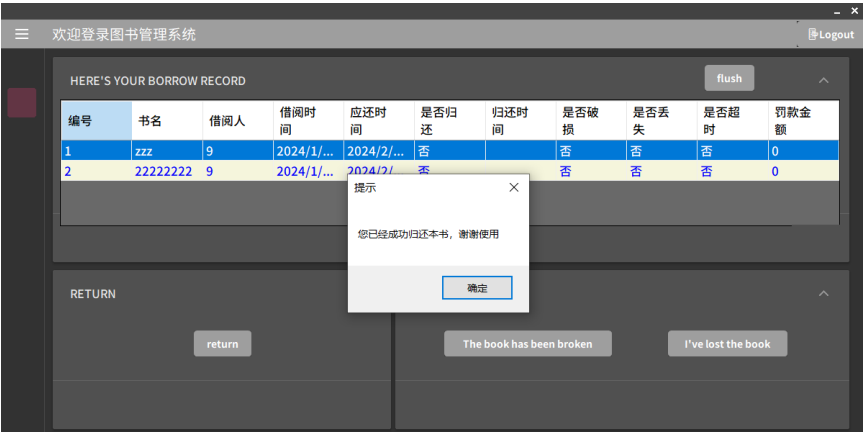


iii. 不可重复借阅图书，重复借阅会进行提示：

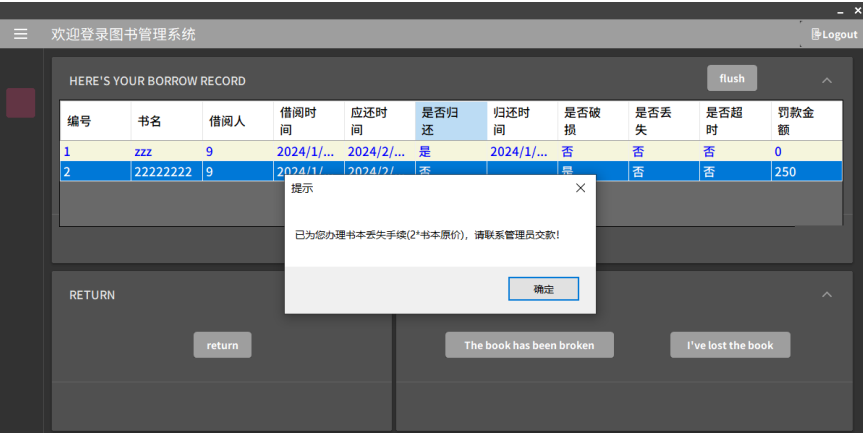
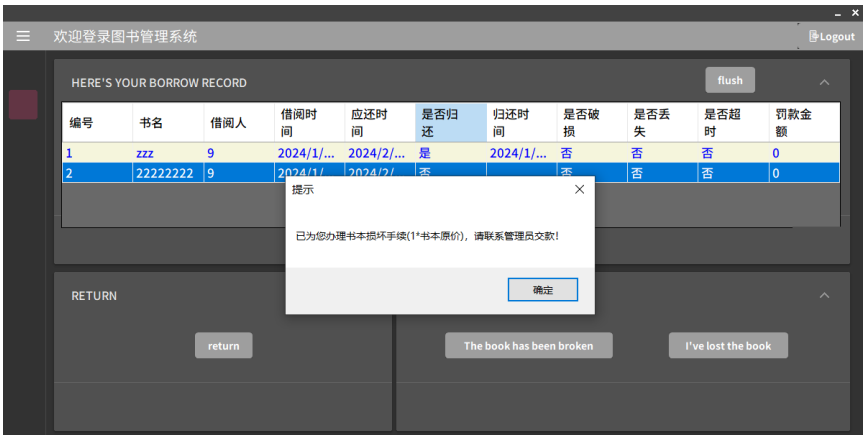


(6)普通用户归还/挂失图书

i. 归还图书，后端检查书本无破损、丢失、超时则直接归还：

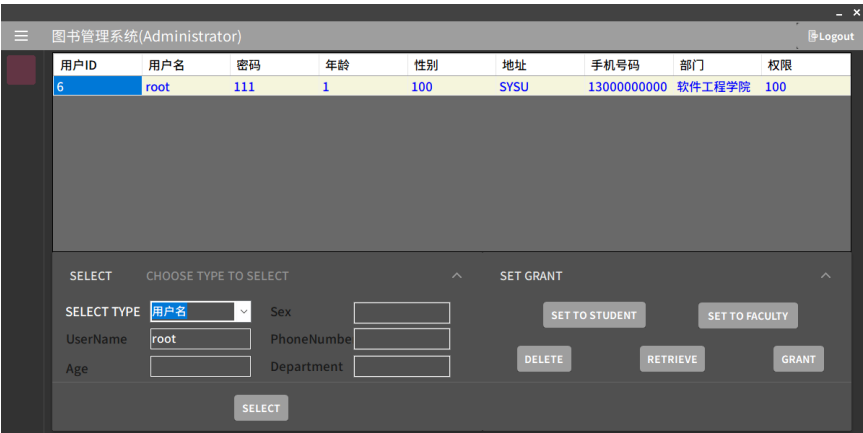


ii. 挂失图书，在书本破损/丢失时直接罚款，此时不能由用户直接归还，否则提醒用户交款：



(7)用户信息管理

i. 搜索用户，对搜索类型对应字段进行模糊查询：



ii. 设置用户为学生/教师；授权用户为管理员；删除用户并进行提示：

图书管理系统(Administrator)

Logout

用户ID	用户名	密码	年龄	性别	地址	手机号码	部门	权限
6	root	111	1	100	SYSU	13000000000	软件工程学院	100
7	9999	111	1	1	999	13000000000	9999	1
56	999	777	1	0	999	13000000000	999	2
57	9	9	1	0	9	13099999999	9	2
60	5	6	1	0	5	13099999999	5	2

SELECT

CHOOSE TYPE TO SELECT

SET GRANT

SELECT TYPE

用户名

Sex

UserName

PhoneNumbe

Age

Department

SET TO STUDENT

SET TO FACULTY

DELETE

RETRIEVE

GRANT

SELECT

操作成功

用户99999的身份已设置为学生

确定

用户ID	用户名	密码	年龄	性别	地址	手机号码	部门	权限
6	root	111	1	100	SYSU	13000000000	软件工程学院	100
7	9999	111	1	1	999	13000000000	9999	1
55	2	2	1	0	2	13000000000	2	1
56	999	777	1	0	999	13000000000	999	2
57	9	9	1	0	9	13099999999	9	0
60	5	6	1	0	5	13099999999	5	2

SELECT

CHOOSE TYPE TO SELECT

SET GRANT

SELECT TYPE

用户名

Sex

UserName

PhoneNumbe

Age

Department

SET TO STUDENT

SET TO FACULTY

DELETE

RETRIEVE

GRANT

SELECT

操作成功

用户99999的身份已设置为教师

确定

SELECT

CHOOSE TYPE TO SELECT

SET GRANT

SELECT TYPE

用户名

Sex

UserName

PhoneNumbe

Age

Department

SET TO STUDENT

SET TO FACULTY

DELETE

RETRIEVE

GRANT

SELECT

操作成功

用户删除成功

确定

SELECT

CHOOSE TYPE TO SELECT

SET GRANT

SELECT TYPE

用户名

Sex

UserName

PhoneNumbe

Age

Department

SET TO STUDENT

SET TO FACULTY

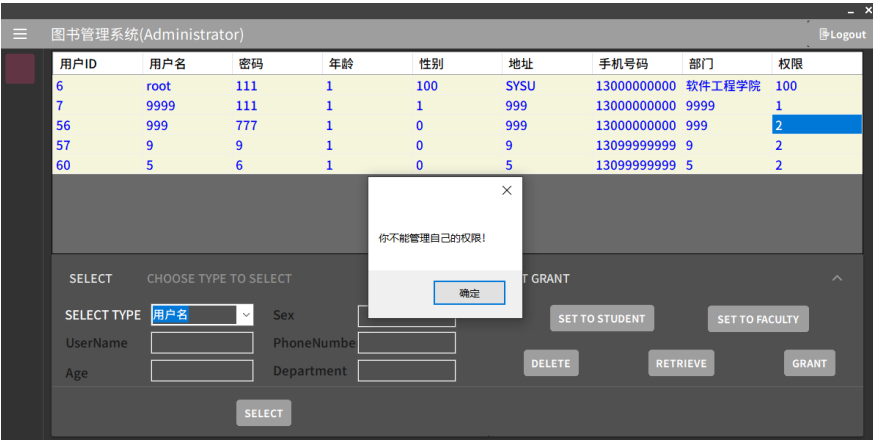
DELETE

RETRIEVE

GRANT

SELECT

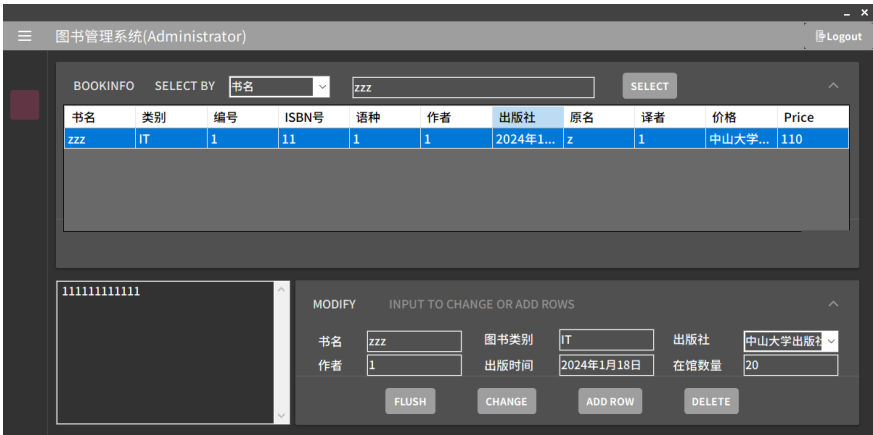
iii. 不得对超级管理员或自身进行操作，会产生警告信息：



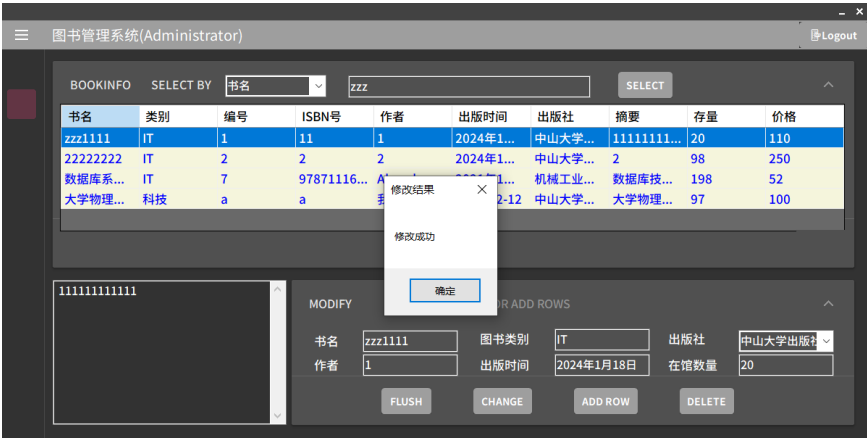
(8)图书信息管理

注：flush用于刷新界面

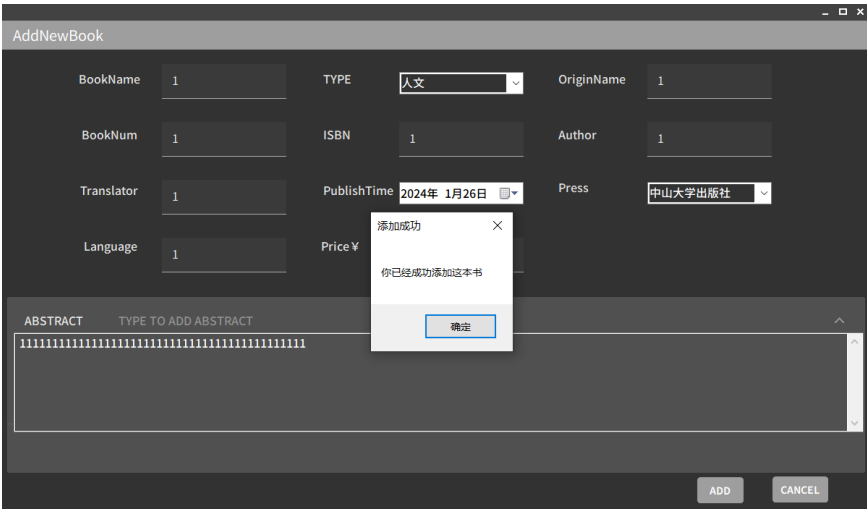
i. 查询图书：



ii. 修改图书信息并提示：



iii. 增加图书信息并提示：

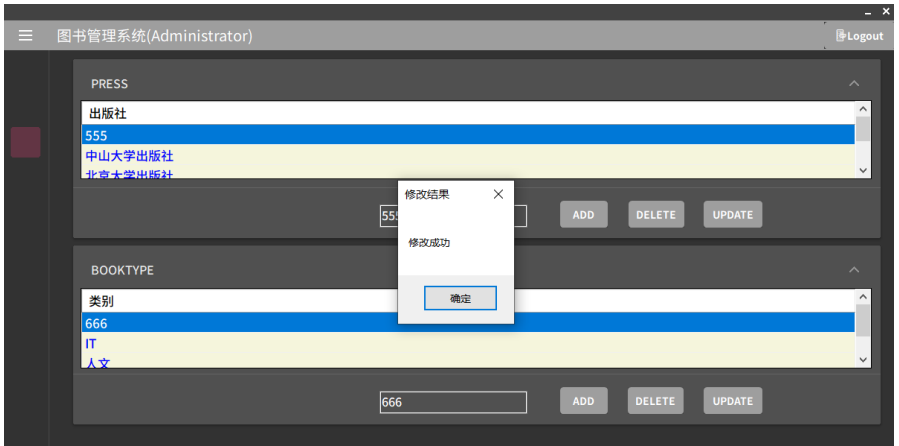
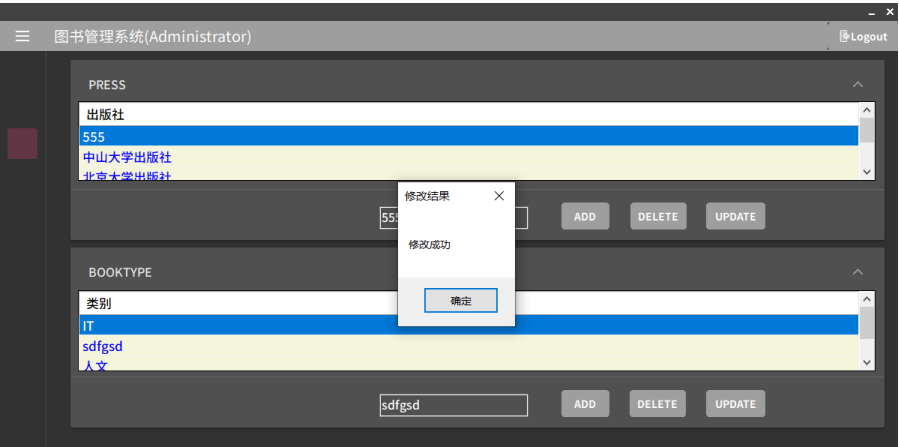
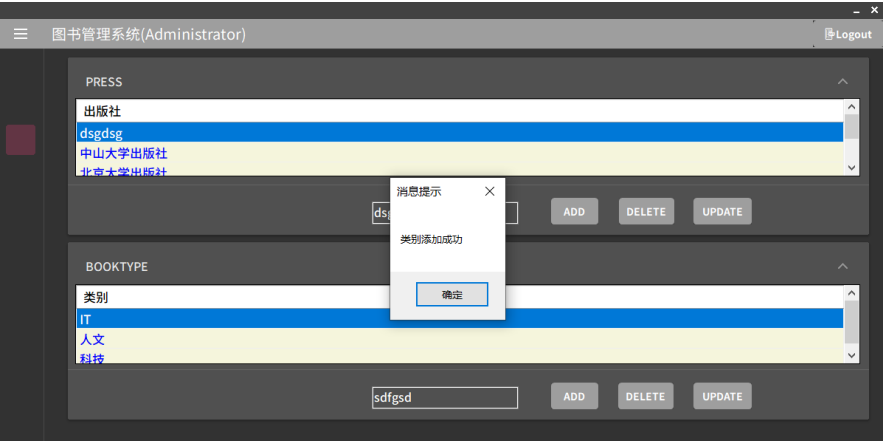
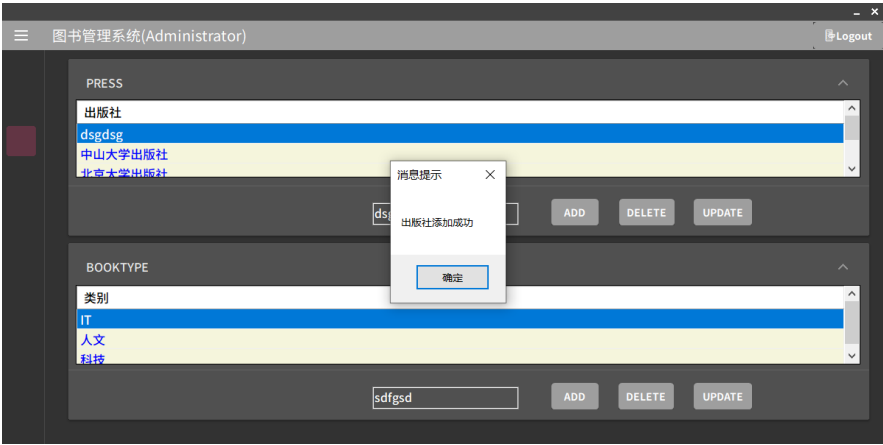


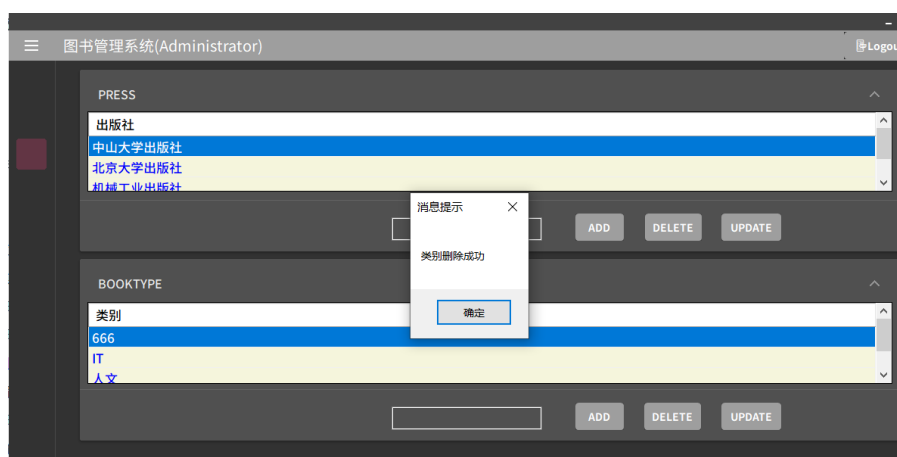
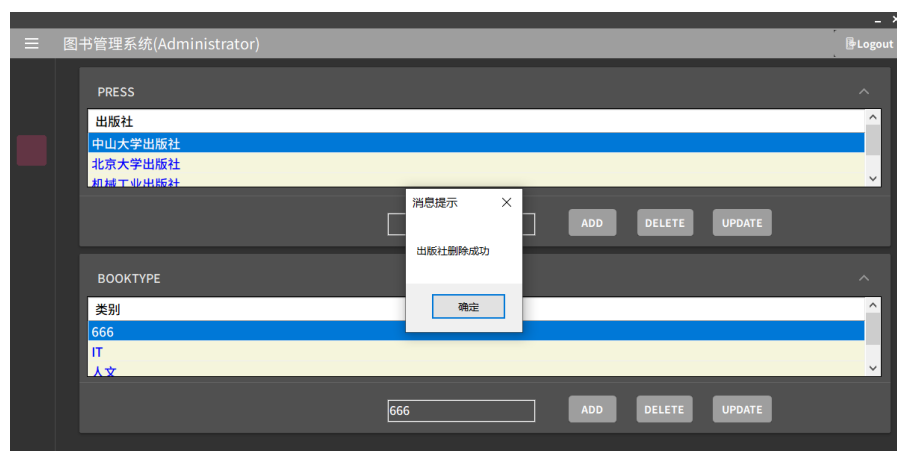
iv. 删除图书信息并提示：



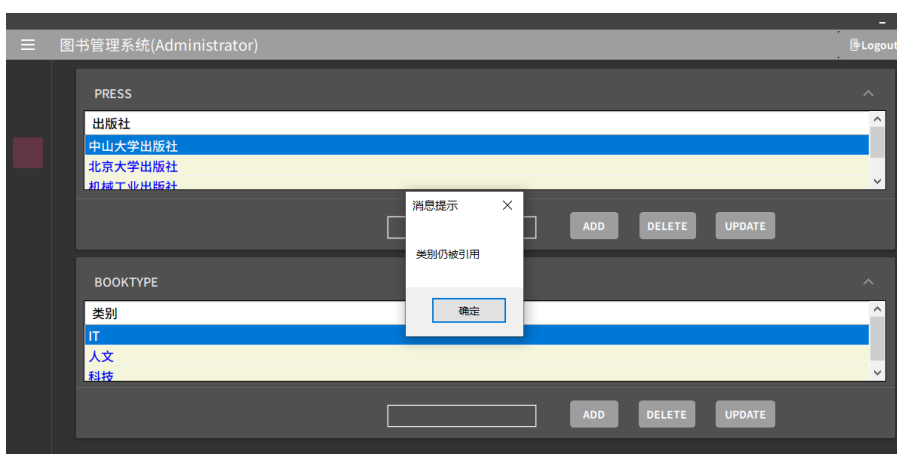
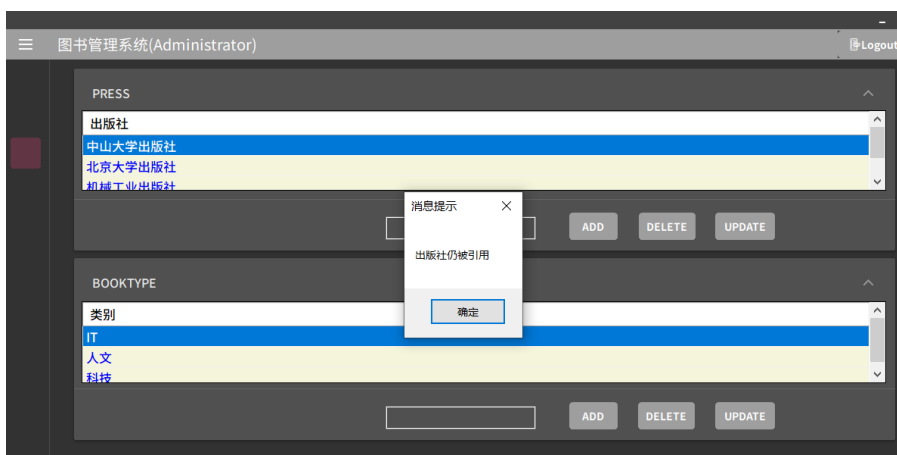
(9)出版社/类别信息管理

i. 增加/修改/删除出版社、类别信息并进行提示：



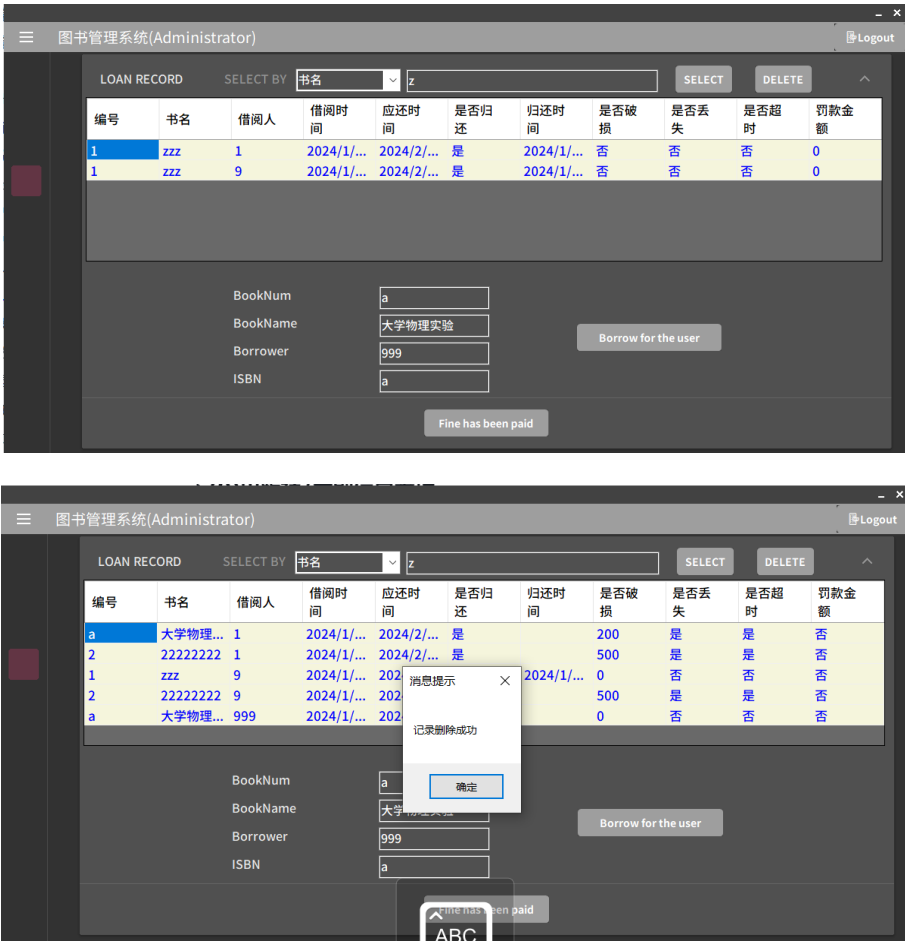


ii. 若出版社、类别信息仍被引用则不能修改，并进行提示：

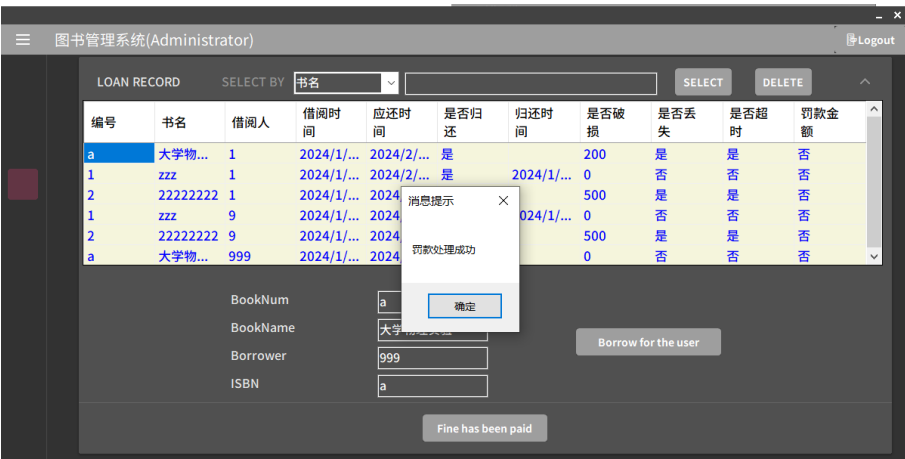


(10) 借阅信息/罚款办理

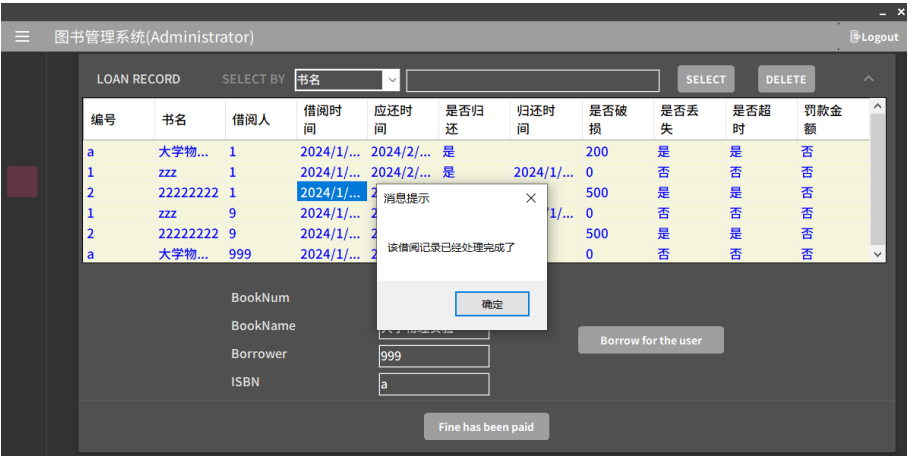
i. 管理员可按照书名/借阅者进行借阅记录查找/删除，删除成功会进行提示：



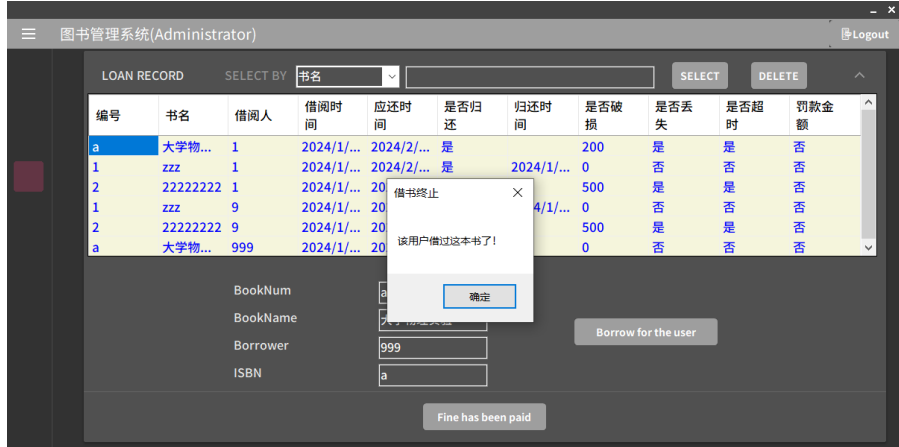
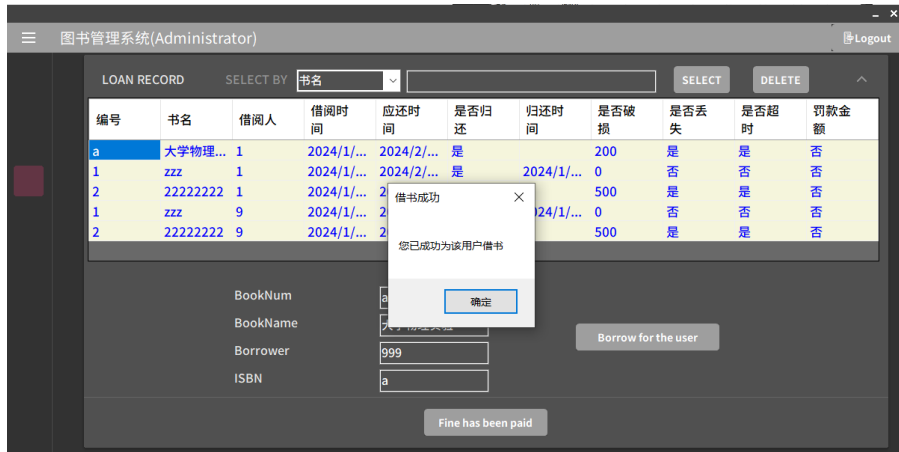
ii. 管理员确定罚款完毕后可办理归还手续，办理成功后提示罚款办理完毕：



iii. 重复归还会被系统提示：

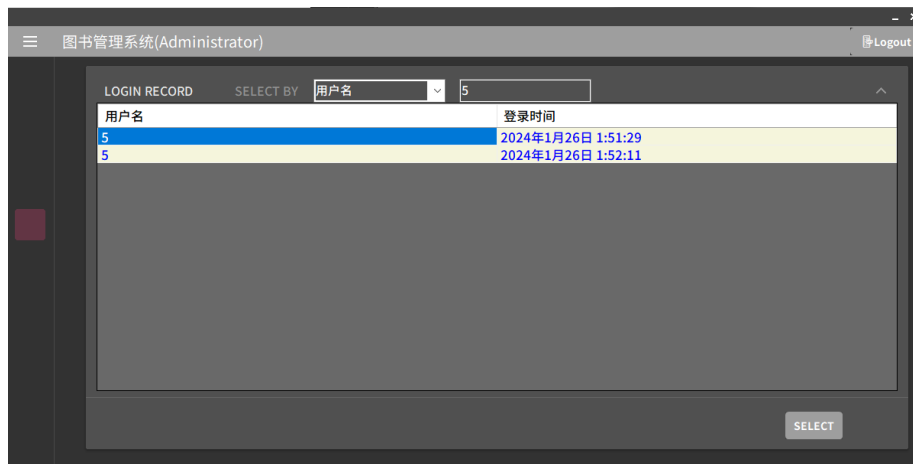


iv. 管理员帮助用户借书，如果已经借过也不能再借，两种情况都输出相应提示信息：



(11)登录记录管理

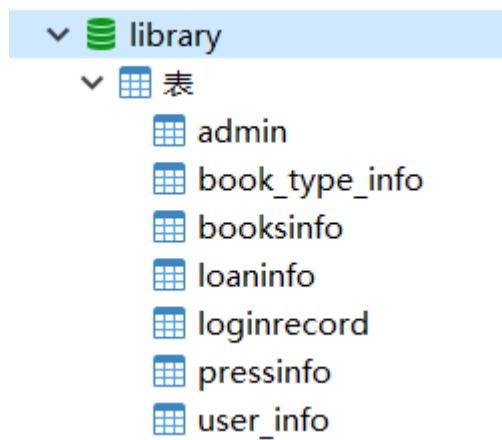
i. 管理员可搜索登录记录：



4.代码实现(数据库交互部分)

(1)数据库设计简要说明

本项目利用library数据库中的以下表运行：



其中，user_info为用户信息表，admin为管理员信息表，其中admin.UserName引用user_info.UserName;

book_type_info记录书本类别信息；pressinfo记录出版社信息；

booksinfo记录书本信息以及馆藏数量，且booksinfo.Press引用pressinfo.Press；
booksinfo.BookType引用book_type_info.BookType；

loaninfo记录借阅/归还状态等信息，loaninfo.UserName引用user_info.userName;loaninfo.BookName引用booksinfo.BookName;

loginrecord记录用户登录信息，loaninfo.UserName引用user_info.UserName。

(2)建立数据库连接

定义DataBaseConnection类对所有与数据库的交互方法进行封装，其中connectionString提供了连接方法：

```
string connectionString =
```

```
"server=localhost;port=3306;user=root;password=123;database=library;"
```

(3)登录相关操作

登录需要查询用户信息表以及向登录记录表进行记录，前端利用以下方法返回的信息判断操作是否成功：

```
/******登录相关
******/
public bool Login(string userName, string password, int IsAdmin)
{
    bool LoginResult = true;
    string CommandText = string.Format("select * from user_info
where UserName='{0}'", userName);
    string AdminCommandText = string.Format("select * from admin
where AdminName='{0}'", userName);
    string FinallyCommand;

    if (IsAdmin == 2)
        FinallyCommand = AdminCommandText;
    else
        FinallyCommand = CommandText;

    using (MySqlConnection libraryConnection = new
MySqlConnection(connectionString))
    {
        libraryConnection.Open();
        MySqlCommand mySqlCommand = new
MySqlCommand(FinallyCommand, libraryConnection);
        MySqlDataReader nameReader = mySqlCommand.ExecuteReader();
        if (nameReader.Read())
        {
            string passwordRead =
nameReader.GetString(nameReader.GetOrdinal("Password"));
            if (IsAdmin < 2)
            {
                if (password == passwordRead && ("100" ==
nameReader.GetString(nameReader.GetOrdinal("Cha")) || "2" ==
nameReader.GetString(nameReader.GetOrdinal("Cha")) || "" + IsAdmin ==
nameReader.GetString(nameReader.GetOrdinal("Cha"))))
                    LoginResult = true;
                else
                    LoginResult = false;
            }
        }
    }
}
```

```

        else
        {
            if (password == passwordRead)
                LoginResult = true;
            else
                LoginResult = false;
        }
    }
    else
        LoginResult = false;
}
return LoginResult;
}
public void LoginRecord(string name, string LoginTime)
{
    string LoginRecordCommandText = string.Format("insert into
loginrecord(UserName,LoginTime) values('{0}','{1}')" , name, LoginTime);

    using (MySqlConnection loginRecords = new
MySqlConnection(connectionString))
    {
        loginRecords.Open();
        MySqlCommand mySqlCommand = new
MySqlCommand(LoginRecordCommandText, loginRecords);
        mySqlCommand.ExecuteNonQuery();
    }
}
/*****登录相关*****/

```

(4)注册相关操作

注册需要从前端传递一部分信息，使用userAdd方法向user_info表中添加信息并返回操作是否成功：

```

/*****注册相关*****/
public bool userAdd(string UserName, string Password, int sex, int
Age, string Address, string Telephone, string workUnit, int character)
{
    bool IsUserAdd;
    string userAddCommand = string.Format("insert into
user_info(UserName, Password, Age, Sex, Address, Telephone, Department,
Cha) values('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}');" ,
UserName, Password, sex, Age, Address, Telephone, workUnit, character);
}

```

```

        string seekUserCommand = string.Format("select * from user_info
where UserName='{0}';", UserName);

        string adminAddCommand = string.Format("insert into
admin(AdminName, Password) values('{0}', '{1}');", UserName, Password);
        string seekAdminCommand = string.Format("select * from admin
where AdminName='{0}';", UserName);

        using (MySqlConnection userAddConnection = new
MySqlConnection(connectionString))
        {
            userAddConnection.Open();
            MySqlCommand AddCommand = new MySqlCommand(userAddCommand,
userAddConnection);
            MySqlCommand SeekCommand = new
MySqlCommand(seekUserCommand, userAddConnection);

            using (MySqlDataReader mySqlDataReader =
SeekCommand.ExecuteReader())
            {
                if (mySqlDataReader.Read()) IsUserAdd = false;
                else IsUserAdd = true;
                userAddConnection.Close();

                userAddConnection.Open();
                if (IsUserAdd) AddCommand.ExecuteNonQuery();
            }
            userAddConnection.Close();

            userAddConnection.Open();
            MySqlCommand AddAdminCommand = new
MySqlCommand(adminAddCommand, userAddConnection);
            MySqlCommand SeekAdminCommand = new
MySqlCommand(seekAdminCommand, userAddConnection);
            using (MySqlDataReader mySqlDataReader =
SeekAdminCommand.ExecuteReader())
            {
                if (character == 2)
                {
                    if (mySqlDataReader.Read()) IsUserAdd = false;
                    else IsUserAdd = true;
                    userAddConnection.Close();

                    userAddConnection.Open();
                    if (IsUserAdd) AddAdminCommand.ExecuteNonQuery();
                }
            }
        }
    }
}

```

```

        return IsUserAdd;
    }
    /*****注册相关*****/
}

```

(5)改密相关操作

改密需要提供用户名、密码、新密码、二次确认的新密码等信息，利用passwordChange方法进行
操作并返回操作情况：

```

/*****改密相关*****/
public bool passwordChange(string name, string password, string
newPassword)
{
    bool IsPasswordChange;

    string passwordChangeCommand = string.Format("update user_info
set Password='{0}' where UserName='{1}'", newPassword, name);
    string IsPasswordCommand = string.Format("select * from
user_info where UserName='{0}'", name);

    using (MySqlConnection passwordChangeConnection = new
MySqlConnection(connectionString))
    {
        passwordChangeConnection.Open();
        MySqlCommand isPassword = new
MySqlCommand(IsPasswordCommand, passwordChangeConnection);
        MySqlCommand passwordChange = new
MySqlCommand(passwordChangeCommand, passwordChangeConnection);
        using (MySqlDataReader mySqlDataReader =
isPassword.ExecuteReader())
        {
            if (mySqlDataReader.Read())
            {
                string passwordRead =
mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Password"));
                if (password == passwordRead)
                {
                    IsPasswordChange = true;
                }
                else
                {
                    IsPasswordChange = false;
                }
            }
            if
(mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Cha")) == "2" ||

```



```

mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Password")) == "100")
    {
        string changeAdminPassword =
string.Format("update admin set Password='{0}' where AdminName='{1}'",
newPassword, name);
        using (MySqlConnection adminChangeConnection =
new MySqlConnection(connectionString))
        {
            adminChangeConnection.Open();
            MySqlCommand adminChange = new
MySqlCommand(changeAdminPassword, adminChangeConnection);
            adminChange.ExecuteNonQuery();
            adminChangeConnection.Close();
        }
    }
    else
        IsPasswordChange = false;
}
if (IsPasswordChange)
    passwordChange.ExecuteNonQuery();
}
return IsPasswordChange;
}
/*****改密相关
*****/

```

(6)注销相关操作

注销相关操作有用户主动注销以及管理员直接销号操作，管理员销号(deleteUser)只需利用用户名，用户主动注销需要确认密码(userLogout)：

```

/*****注销相关
*****/
public void deleteUser(string name)
{
    string deleteUserCommand = string.Format("delete from user_info
where UserName='{0}'", name);
    using (MySqlConnection deleteUserConnection = new
MySqlConnection(connectionString))
    {
        deleteUserConnection.Open();
        MySqlCommand mySqlCommand = new
MySqlCommand(deleteUserCommand, deleteUserConnection);
        mySqlCommand.ExecuteNonQuery();
    }
}

```

```

    }
}
public bool userLogout(string name, string password)
{
    bool IsDelete;

    string userLogoutCommand = string.Format("delete from user_info
where UserName='{0}']", name);
    string seekUserCommand = string.Format("select * from user_info
where UserName='{0}']", name);
    using (MySQLConnection userLogoutConnection = new
MySQLConnection(connectionString))
    {
        userLogoutConnection.Open();
        MySqlCommand logoutCommand = new
MySQLCommand(userLogoutCommand, userLogoutConnection);
        MySqlCommand seekCommad = new MySqlCommand(seekUserCommand,
userLogoutConnection);
        using (MySQLDataReader mySqlDataReader =
seekCommad.ExecuteReader())
        {
            if (mySqlDataReader.Read())
            {
                string passwordRead =
mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Password"));
                if (password == passwordRead &&
mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Cha")) != "2" &&
mySqlDataReader.GetString(mySqlDataReader.GetOrdinal("Cha")) != "100")
                {
                    IsDelete = true;
                }
            }
            else
            {
                IsDelete = false;
            }
        }
        else
        {
            IsDelete = false;
        }
    }
    if (IsDelete)
        logoutCommand.ExecuteNonQuery();
}
return IsDelete;
}

/*****注销相关*****/

```

(7)书本相关操作

书本的增删改查都要利用该分组中的相关方法，这里不一一解释：

```

/*****书本相关*****/
public DataTable showBook()
{
    DataTable books;
    string showBookCommandText = string.Format("select
BookName,BookType,BookNum,ISBN,Author,PressTime,Press,Abstract,ExistNum,Pric
e from booksinfo");

    using (SqlConnection showBookConnection = new
SqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(showBookCommandText, showBookConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        books = dataTable;
    }
    books.Columns[0].ColumnName = "书名";
    books.Columns[1].ColumnName = "类别";
    books.Columns[2].ColumnName = "编号";
    books.Columns[3].ColumnName = "ISBN号";
    books.Columns[4].ColumnName = "作者";
    books.Columns[5].ColumnName = "出版时间";
    books.Columns[6].ColumnName = "出版社";
    books.Columns[7].ColumnName = "摘要";
    books.Columns[8].ColumnName = "存量";
    books.Columns[9].ColumnName = "价格";
    return books;
}
public DataTable seekBookByName(string bookName, string selectType)
{
    switch (selectType)
    {
        case "书名":
            selectType = "BookName";
            break;
        case "ISBN号":
            selectType = "ISBN";
            break;
        case "类别":
            selectType = "BookType";
            break;
        case "编号":

```

```

        selectType = "BookNum";
        break;
    case "作者":
        selectType = "Author";
        break;
    case "译者":
        selectType = "Translator";
        break;
    case "出版社":
        selectType = "Press";
        break;
    case "存量":
        selectType = "ExistNum";
        break;
    }

    string seekBookByNameCommand = string.Format("select
BookName,BookType,Language,OriginName,BookNum,Author,ISBN,Translator,PressTi
me,Press,Price from booksinfo where {0} like '%{1}%'", selectType,
bookName);

    DataTable bookData;

    using (MySqlConnection seekBookByNameConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekBookByNameCommand, seekBookByNameConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        bookData = dataTable;
    }

    bookData.Columns[0].ColumnName = "书名";
    bookData.Columns[1].ColumnName = "类别";
    bookData.Columns[2].ColumnName = "语种";
    bookData.Columns[3].ColumnName = "原名";
    bookData.Columns[4].ColumnName = "编号";
    bookData.Columns[5].ColumnName = "作者";
    bookData.Columns[6].ColumnName = "ISBN号";
    bookData.Columns[7].ColumnName = "译者";
    bookData.Columns[8].ColumnName = "出版时间";
    bookData.Columns[8].ColumnName = "出版社";
    bookData.Columns[9].ColumnName = "价格";
    return bookData;
}

public string[] bookinfos(string bookName)
{

```

```

        string bookDigestCommand = string.Format("select * from
booksinfo where BookName='{0}'", bookName);
        string[] bookinfos = new string[9];

        using (MySqlConnection bookDigestConnection = new
MySqlConnection(connectionString))
        {
            bookDigestConnection.Open();
            MySqlCommand mySqlCommand = new
MySqlCommand(bookDigestCommand, bookDigestConnection);
            MySqlDataReader mySqlDataReader =
mySqlCommand.ExecuteReader();
            while (mySqlDataReader.Read())
            {
                bookinfos[0] = mySqlDataReader["BookName"].ToString();
                bookinfos[1] = mySqlDataReader["ISBN"].ToString();
                bookinfos[2] = mySqlDataReader["BookType"].ToString();
                bookinfos[3] = mySqlDataReader["Press"].ToString();
                bookinfos[4] = mySqlDataReader["Author"].ToString();
                bookinfos[5] = mySqlDataReader["PressTime"].ToString();
                bookinfos[6] = mySqlDataReader["BookNum"].ToString();
                bookinfos[7] = mySqlDataReader["ExistNum"].ToString();
                bookinfos[8] = mySqlDataReader["Abstract"].ToString();
            }
        }
        return bookinfos;
    }

    public bool addNewBook(string bookName, string bookClass, string
language, string oldName,
        string bookNumber, string price, string author, string iSBN,
string translator,
        string publish, string press, string content)
    {
        bool IsBookAdd;
        string addNewBookCommandText = string.Format("insert into
booksinfo(BookName,BookType,Language,OriginName,BookNum,Price,Author,ISBN,Tr
anslator,PressTime,Press,Abstract) value('{0}','{1}','{2}','{3}','{4}',
{5}','{6}','{7}','{8}','{9}','{10}','{11}')" , bookName, bookClass, language,
oldName, bookNumber, price, author, iSBN, translator, publish, press,
content);

        string IsBookEixstedCommandText = string.Format("select * from
booksinfo where BookName='{0}'", bookName);
        using (MySqlConnection addNewBookConnection = new
MySqlConnection(connectionString))
        {
            addNewBookConnection.Open();

```

```

        MySqlCommand AddCommand = new
MySqlCommand(addNewBookCommandText, addNewBookConnection);
        MySqlCommand IsExistedCommand = new
MySqlCommand(IsBookEixstedCommandText, addNewBookConnection);
        using (MySqlDataReader IsExistedReader =
IsExistedCommand.ExecuteReader())
        {
            if (IsExistedReader.Read())
                IsBookAdd = false;
            else
                IsBookAdd = true;
        }
        if (IsBookAdd)
            AddCommand.ExecuteNonQuery();
    }
    return IsBookAdd;
}

public void bookInfoChange(string bookNumber, string bookName,
string bookClass, string press, string author, string bookCount, string
publishTime, string digestText)
{
    string bookInfoChangeCommandText = string.Format("update
booksinfo set BookName='{0}',BookType='{1}',Press='{2}'," +
"Author='{3}',PressTime='{4}',ExistNum={5}, Abstract='{6}' where
BookNum='{7}'", bookName, bookClass, press, author, publishTime, bookCount,
digestText, bookNumber);

    using (MySqlConnection bookInfoChangeConnection = new
MySqlConnection(connectionString))
    {
        bookInfoChangeConnection.Open();
        MySqlCommand mySqlCommand = new
MySqlCommand(bookInfoChangeCommandText, bookInfoChangeConnection);
        mySqlCommand.ExecuteNonQuery();
    }
}

public void bookDelete(string ISBN)
{
    string bookDeleteCommand = string.Format("delete from booksinfo
where ISBN='{0}'", ISBN);
    string foreign_key_checks_false = "SET foreign_key_checks = 0;";
    string foreign_key_checks = "SET foreign_key_checks = 1;";
    using (MySqlConnection bookDeleteConnection = new
MySqlConnection(connectionString))
    {
        bookDeleteConnection.Open();

```

```

        MySqlCommand mySqlCommand = new
MySqlCommand(bookDeleteCommand, bookDeleteConnection);
        MySqlCommand foreign_key_checks_falseCommand = new
MySqlCommand(foreign_key_checks_false, bookDeleteConnection);
        MySqlCommand foreign_key_checksCommand = new
MySqlCommand(foreign_key_checks, bookDeleteConnection);
        foreign_key_checks_falseCommand.ExecuteNonQuery();
        mySqlCommand.ExecuteNonQuery();
        foreign_key_checksCommand.ExecuteNonQuery();
        bookDeleteConnection.Close();
    }
}
public DataTable getBookType()
{
    DataTable bookType;

    string seekType = string.Format("select * from book_type_info");

    using (MySqlConnection seekTypeConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekType, seekTypeConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        bookType = dataTable;
    }
    return bookType;
}
public DataTable showBookType()
{
    DataTable bookType;
    string showBookTypeCommandText = string.Format("select BookType
from book_type_info");

    using (MySqlConnection showBookTypeConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(showBookTypeCommandText, showBookTypeConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        bookType = dataTable;
    }
    bookType.Columns[0].ColumnName = "类别";
    return bookType;
}

```

```

public bool bookTypeDelete(string bookTypeName)
{
    bool isDeleted;
    string selectbookTypeString = string.Format("select * from
booksinfo where BookType='{0}'", bookTypeName);
    using (MySqlConnection selectbookTypeConnection = new
MySqlConnection(connectionString))
    {
        selectbookTypeConnection.Open();
        MySqlCommand selectbookTypeCommand = new
MySqlCommand(selectbookTypeString, selectbookTypeConnection);
        MySqlDataReader bookTypeReader =
selectbookTypeCommand.ExecuteReader();
        if (!bookTypeReader.Read())
        {
            string bookTypeDeleteCommand = string.Format("delete
from book_type_info where BookType='{0}'", bookTypeName);
            using (MySqlConnection bookTypeDeleteConnection = new
MySqlConnection(connectionString))
            {
                bookTypeDeleteConnection.Open();
                MySqlCommand mySqlCommand = new
MySqlCommand(bookTypeDeleteCommand, bookTypeDeleteConnection);
                mySqlCommand.ExecuteNonQuery();
                bookTypeDeleteConnection.Close();
                isDeleted = true;
            }
        }
        else
        {
            isDeleted = false;
        }
        selectbookTypeConnection.Close();
    }
    return isDeleted;
}

public bool bookTypeAdd(string bookTypeName)
{
    bool isAdded;

    string addbookTypeString = string.Format("insert into
book_type_info(BookType) values('{0}') ", bookTypeName);

    using (MySqlConnection addbookTypeConnection = new
MySqlConnection(connectionString))
    {
        addbookTypeConnection.Open();

```



```

        MySqlCommand mySqlCommand = new
MySqlCommand(addbookTypeString, addbookTypeConnection);
        isAdded = mySqlCommand.ExecuteNonQuery() > 0 ? true : false;
        addbookTypeConnection.Close();
    }
    return isAdded;
}

public bool bookTypeInfoChange(string oldBookType, string
newBookType)
{
    bool isChanged;

    string selectbookTypeString = string.Format("select * from
booksinfo where BookType='{0}'", oldBookType);
    using (MySqlConnection selectbookTypeConnection = new
MySqlConnection(connectionString))
    {
        selectbookTypeConnection.Open();
        MySqlCommand selectbookTypeCommand = new
MySqlCommand(selectbookTypeString, selectbookTypeConnection);
        MySqlDataReader bookTypeReader =
selectbookTypeCommand.ExecuteReader();
        if (!bookTypeReader.Read())
        {
            string changeBookTypeString = string.Format("update
book_type_info set BookType='{1}' where BookType='{0}'", oldBookType,
newBookType);
            using (MySqlConnection changeBookTypeConnection = new
MySqlConnection(connectionString))
            {
                changeBookTypeConnection.Open();
                MySqlCommand mySqlCommand = new
MySqlCommand(changeBookTypeString, changeBookTypeConnection);
                isChanged = mySqlCommand.ExecuteNonQuery() > 0 ?
true : false;
                changeBookTypeConnection.Close();
            }
        }
        else
        {
            isChanged = false;
        }
        selectbookTypeConnection.Close();
    }
    return isChanged;
}

```

(8)归还/损坏/丢失/罚款相关操作(使用事务实现)

使用事务实现书本归还/损坏/丢失/罚款操作，这四个操作的相关方法都返回bool数组，用于表示书本当前属于哪些复杂状态，前端利用该数组详细判断目前应该进行何种形式的提示：

```

/*****归还丢失损坏相关
*****/
public bool[] returnBook(string bookName, string bookNumber, string
returnTime, string borrowTime, string userName)
{
    bool isBorrow = false;
    bool IsReturn = false;
    bool IsBroken = false;
    bool IsLost = false;
    bool IsOverTime = false;
    bool[] flag = new bool[5];
    string returnCommandText = string.Format("update loaninfo set
IsReturn='是',ReturnTime='{0}' where BorrowTime='{1}'" + " and
Borrower='{2}'", returnTime, borrowTime, userName);
    string IsReturnCommandText = string.Format("select * from
loaninfo where BorrowTime='{0}'", borrowTime);
    string updateBookCommandText = string.Format("update booksinfo
set ExistNum=ExistNum+1 where BookName='{0}' and BookNum='{1}'", bookName,
bookNumber);

    using (MySQLConnection returnBookConnection = new
MySQLConnection(connectionString))
    {
        returnBookConnection.Open();
        MySqlCommand returnCommand =
returnBookConnection.CreateCommand();
        MySqlCommand IsReturnCommand =
returnBookConnection.CreateCommand();
        MySqlCommand UpdateCommand =
returnBookConnection.CreateCommand();
        MySQLTransaction transaction;

        //启动事务
        transaction = returnBookConnection.BeginTransaction();

        //设定SqlCommand的事务和连接对象
        returnCommand.Connection = returnBookConnection;

```

```

returnCommand.Transaction = transaction;

IsReturnCommand.Connection = returnBookConnection;
IsReturnCommand.Transaction = transaction;

UpdateCommand.Connection = returnBookConnection;
UpdateCommand.Transaction = transaction;

// 开始执行操作
try
{
    returnCommand.CommandText = returnCommandText;
    IsReturnCommand.CommandText = IsReturnCommandText;
    UpdateCommand.CommandText = updateBookCommandText;

    using (MySqlDataReader IsReturnRead =
IsReturnCommand.ExecuteReader())
    {
        if (IsReturnRead.Read())
        {
            if (IsReturnRead["IsReturn"].ToString() == "是")
IsReturn = true;

            else IsReturn = false;
            if (IsReturnRead["IsBroken"].ToString() == "是")
IsBroken = true;

            else IsBroken = false;
            if (IsReturnRead["IsLost"].ToString() == "是")
IsLost = true;

            else IsLost = false;
            if (IsReturnRead["IsOverTime"].ToString() ==
"是") IsOverTime = true;

            else IsOverTime = false;
            DateTime dt =
DateTime.Parse(IsReturnRead["ShouldReturnTime"].ToString());
            if (dt < DateTime.Now) IsOverTime = true;
            else IsOverTime = false;
            isBorrow = true;
        }
        else isBorrow = false;
    }
    if (!IsReturn && isBorrow && (!IsBroken && !IsLost &&
!IsOverTime))
    {
        returnCommand.ExecuteNonQuery();
        UpdateCommand.ExecuteNonQuery();
    }
}

```

```

        // 完成提交
        transaction.Commit();
    }
    catch (Exception ex)
    {
        //数据回滚
        transaction.Rollback();
    }
}

flag[0] = IsReturn;
flag[1] = isBorrow;
flag[2] = IsBroken;
flag[3] = IsLost;
flag[4] = IsOverTime;
return flag;
}

public bool[] brokenBook(string bookName, string bookNumber, string
returnTime, string borrowTime, string userName)
{
    bool isBorrow = false;
    bool IsReturn = false;
    bool IsBroken = false;
    bool IsLost = false;
    bool IsOverTime = false;
    bool[] flag = new bool[5];
    string price = "";

    string selectBookCommandText = string.Format("select Price from
booksinfo where BookName='{0}' and BookNum='{1}'", bookName, bookNumber);
    using (MySQLConnection selectBookConnection = new
MySQLConnection(connectionString))
    {
        selectBookConnection.Open();
        MySqlCommand SelectCommand =
selectBookConnection.CreateCommand();
        SelectCommand.Connection = selectBookConnection;
        SelectCommand.CommandText = selectBookCommandText;
        using (MySQLDataReader selectRead =
SelectCommand.ExecuteReader())
        {
            if (selectRead.Read())
                price = selectRead["Price"].ToString();
        }
        selectBookConnection.Close();
    }
}

```

```

        string brokenCommandText = string.Format("update loaninfo set
IsBroken='是',Punishment={0} where BorrowTime='{1}' and Borrower='{2}'",
price, borrowTime, userName);
        string IsBrokenCommandText = string.Format("select * from
loaninfo where BorrowTime='{0}'", borrowTime);

        using (MySQLConnection brokenBookConnection = new
MySQLConnection(connectionString))
        {
            brokenBookConnection.Open();
            MySqlCommand brokenCommand =
brokenBookConnection.CreateCommand();
            MySqlCommand IsBrokenCommand =
brokenBookConnection.CreateCommand();
            MySQLTransaction transaction;

            //启动事务
            transaction = brokenBookConnection.BeginTransaction();

            //设定SqlCommand的事务和连接对象
            brokenCommand.Connection = brokenBookConnection;
            brokenCommand.Transaction = transaction;

            IsBrokenCommand.Connection = brokenBookConnection;
            IsBrokenCommand.Transaction = transaction;

            // 开始执行操作
            try
            {
                brokenCommand.CommandText = brokenCommandText;
                IsBrokenCommand.CommandText = IsBrokenCommandText;

                using (MySQLDataReader IsBrokenRead =
IsBrokenCommand.ExecuteReader())
                {
                    if (IsBrokenRead.Read())
                    {
                        if (IsBrokenRead["IsReturn"].ToString() == "是")
IsReturn = true;

                        else IsReturn = false;
                        if (IsBrokenRead["IsBroken"].ToString() == "是")
IsBroken = true;

                        else IsBroken = false;
                        if (IsBrokenRead["IsLost"].ToString() == "是")
IsLost = true;

                        else IsLost = false;
                    }
                }
            }
            catch { }
            finally { transaction.Commit(); }
        }
    }
}

```

```

        DateTime dt =
DateTime.Parse(IsBrokenRead["ShouldReturnTime"].ToString());
        if (dt < DateTime.Now) IsOverTime = true;
        else IsOverTime = false;
        isBorrow = true;
    }
    else isBorrow = false;
}
if (!IsReturn && isBorrow && (!IsBroken && !IsLost))
    brokenCommand.ExecuteNonQuery();

    // 完成提交
    transaction.Commit();
}
catch (Exception ex)
{
    //数据回滚
    transaction.Rollback();
}
}
flag[0] = IsReturn;
flag[1] = isBorrow;
flag[2] = IsBroken;
flag[3] = IsLost;
flag[4] = IsOverTime;
return flag;
}
public bool[] lostBook(string bookName, string bookNumber, string
borrowTime, string userName)
{
    bool isBorrow = false;
    bool IsReturn = false;
    bool IsBroken = false;
    bool IsLost = false;
    bool IsOverTime = false;
    bool[] flag = new bool[5];
    string price = "";

    string selectBookCommandText = string.Format("select Price from
booksinfo where BookName='{0}' and BookNum='{1}'", bookName, bookNumber);
    using (MySqlConnection selectBookConnection = new
MySqlConnection(connectionString))
    {
        selectBookConnection.Open();
        MySqlCommand SelectCommand =
selectBookConnection.CreateCommand();
        SelectCommand.Connection = selectBookConnection;

```

```

        SelectCommand.CommandText = selectBookCommandText;
        using (MySqlDataReader selectRead =
SelectCommand.ExecuteReader())
        {
            if (selectRead.Read())
                price = selectRead["Price"].ToString();
        }
        selectBookConnection.Close();
    }

    string lostCommandText = string.Format("update loaninfo set
IsLost='是',Punishment=2*{0} where BorrowTime='{1}' and Borrower='{2}'",
price, borrowTime, userName);
    string IsLostCommandText = string.Format("select * from
loaninfo where BorrowTime='{0}'", borrowTime);

    using (MySqlConnection lostBookConnection = new
MySqlConnection(connectionString))
    {
        lostBookConnection.Open();
        MySqlCommand lostCommand =
lostBookConnection.CreateCommand();
        MySqlCommand IsLostCommand =
lostBookConnection.CreateCommand();
        MySqlTransaction transaction;

        //启动事务
        transaction = lostBookConnection.BeginTransaction();

        //设定SqlCommand的事务和连接对象
        lostCommand.Connection = lostBookConnection;
        lostCommand.Transaction = transaction;

        IsLostCommand.Connection = lostBookConnection;
        IsLostCommand.Transaction = transaction;

        // 开始执行操作
        try
        {
            lostCommand.CommandText = lostCommandText;
            IsLostCommand.CommandText = IsLostCommandText;

            using (MySqlDataReader IsLostRead =
IsLostCommand.ExecuteReader())
            {
                if (IsLostRead.Read())
                {

```

```

        if (IsLostRead["IsReturn"].ToString() == "是")
            IsReturn = true;
        else IsReturn = false;
        if (IsLostRead["IsBroken"].ToString() == "是")
            IsBroken = true;
        else IsBroken = false;
        if (IsLostRead["IsLost"].ToString() == "是")
            IsLost = true;
        else IsLost = false;
        DateTime dt =
        DateTime.Parse(IsLostRead["ShouldReturnTime"].ToString());
        if (dt < DateTime.Now) IsOverTime = true;
        else IsOverTime = false;
        isBorrow = true;
    }
    else isBorrow = false;
}
if (!IsReturn && isBorrow && !IsLost)
{
    lostCommand.ExecuteNonQuery();
}

// 完成提交
transaction.Commit();
}
catch (Exception ex)
{
    //数据回滚
    transaction.Rollback();
}
}
flag[0] = IsReturn;
flag[1] = isBorrow;
flag[2] = IsBroken;
flag[3] = IsLost;
flag[4] = IsOverTime;
return flag;
}
/*****归还丢失损坏相关*****/

```

(9)借阅相关操作(使用事务实现)

借阅书本的操作borrowAdd使用了事务的方式实现；其他的方法是管理员对借阅记录进行管理时使用的操作：


```

/*****借阅相关*****/
public DataTable loanRecord(string name)
{
    DataTable loanRecord;

    string CommandText = string.Format("select
BookNum,BookName,Borrower,BorrowTime,ShouldReturnTime,IsReturn,ReturnTime,Is
Broken,IsLost,IsOverTime,Punishment from loaninfo where Borrower='{0}'",
name);

    using (MySqlConnection loanRecordConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(CommandText, connectionString);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        loanRecord = dataTable;
    }
    loanRecord.Columns[0].ColumnName = "编号";
    loanRecord.Columns[1].ColumnName = "书名";
    loanRecord.Columns[2].ColumnName = "借阅人";
    loanRecord.Columns[3].ColumnName = "借阅时间";
    loanRecord.Columns[4].ColumnName = "应还时间";
    loanRecord.Columns[5].ColumnName = "是否归还";
    loanRecord.Columns[6].ColumnName = "归还时间";
    loanRecord.Columns[7].ColumnName = "是否破损";
    loanRecord.Columns[8].ColumnName = "是否丢失";
    loanRecord.Columns[9].ColumnName = "是否超时";
    loanRecord.Columns[10].ColumnName = "罚款金额";
    return loanRecord;
}

public bool borrowAdd(string bookName, string ISBN, string name,
string borrowTime, string bookNumber)
{
    bool IsborrowAdd = false;
    string shouldReturnTime = DateTime.Now.AddMonths(1).ToString();
    string borrowBookCommand = string.Format("insert into
loaninfo(BookNum,BookName,Borrower,BorrowTime,ShouldReturnTime,IsReturn,IsBr
oken,IsLost,IsOverTime)
values('{0}','{1}','{2}','{3}','{4}','否','否','否','否')", bookNumber,
bookName, name, borrowTime, shouldReturnTime);
    string seekBookCommand = string.Format("select * from loaninfo
where BookName='{0}'" + " and Borrower='{1}' and IsReturn='否'", bookName,
name);
}

```

```

        string updateBookCommand = string.Format("update booksinfo set
ExistNum=ExistNum-1 where ISBN='{0}'", ISBN);

        using (MySqlConnection userAddConnection = new
MySqlConnection(connectionString))
        {
            userAddConnection.Open();
            MySqlCommand AddCommand = userAddConnection.CreateCommand();
            MySqlCommand SeekCommand =
userAddConnection.CreateCommand();
            MySqlCommand UpdateCommand =
userAddConnection.CreateCommand();
            MySqlTransaction transaction;

            //启动事务
            transaction = userAddConnection.BeginTransaction();

            //设定SqlCommand的事务和连接对象
            AddCommand.Connection = userAddConnection;
            AddCommand.Transaction = transaction;

            SeekCommand.Connection = userAddConnection;
            SeekCommand.Transaction = transaction;

            UpdateCommand.Connection = userAddConnection;
            UpdateCommand.Transaction = transaction;

            // 开始执行操作
            try
            {
                AddCommand.CommandText = borrowBookCommand;
                SeekCommand.CommandText = seekBookCommand;
                UpdateCommand.CommandText = updateBookCommand;

                using (MySqlDataReader mySqlDataReader =
SeekCommand.ExecuteReader())
                {
                    if (mySqlDataReader.Read())
                    {
                        if (mySqlDataReader["IsReturn"].ToString() ==
"否")
                        {
                            IsborrowAdd = false;
                        }
                        else
                        {
                            IsborrowAdd = true;

```

```

        }
    }
    else IsborrowAdd = true;
}
if (IsborrowAdd)
{
    AddCommand.ExecuteNonQuery();
    UpdateCommand.ExecuteNonQuery();
}

// 完成提交
transaction.Commit();
}
catch (Exception ex)
{
    //数据回滚
    transaction.Rollback();
}
}
return IsborrowAdd;
}
public DataTable showAllLoanRecord()
{
    DataTable loanRecord;
    string CommandText = string.Format("select * from loaninfo");
    using (MySqlConnection loanRecordConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(CommandText, connectionString);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        loanRecord = dataTable;
    }
    loanRecord.Columns[0].ColumnName = "编号";
    loanRecord.Columns[1].ColumnName = "书名";
    loanRecord.Columns[2].ColumnName = "借阅人";
    loanRecord.Columns[3].ColumnName = "借阅时间";
    loanRecord.Columns[4].ColumnName = "应还时间";
    loanRecord.Columns[5].ColumnName = "是否归还";
    loanRecord.Columns[6].ColumnName = "归还时间";
    loanRecord.Columns[7].ColumnName = "是否破损";
    loanRecord.Columns[8].ColumnName = "是否丢失";
    loanRecord.Columns[9].ColumnName = "是否超时";
    loanRecord.Columns[10].ColumnName = "罚款金额";
    return loanRecord;
}

```

```

        public DataTable seekLoanRecord(string selectType, string name)
        {
            DataTable loanRecord;
            string CommandText = string.Format("select
BookNum,BookName,Borrower,BorrowTime,ShouldReturnTime,IsReturn,ReturnTime,Is
Broken,IsLost,IsOverTime,Punishment from loaninfo where {0} like '%{1}%'",
selectType, name);
            using (MySqlConnection loanRecordConnection = new
MySqlConnection(connectionString))
            {
                MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(CommandText, connectionString);
                DataTable dataTable = new DataTable();
                mySqlDataAdapter.Fill(dataTable);
                loanRecord = dataTable;
            }
            loanRecord.Columns[0].ColumnName = "编号";
            loanRecord.Columns[1].ColumnName = "书名";
            loanRecord.Columns[2].ColumnName = "借阅人";
            loanRecord.Columns[3].ColumnName = "借阅时间";
            loanRecord.Columns[4].ColumnName = "应还时间";
            loanRecord.Columns[5].ColumnName = "是否归还";
            loanRecord.Columns[6].ColumnName = "归还时间";
            loanRecord.Columns[7].ColumnName = "是否破损";
            loanRecord.Columns[8].ColumnName = "是否丢失";
            loanRecord.Columns[9].ColumnName = "是否超时";
            loanRecord.Columns[10].ColumnName = "罚款金额";
            return loanRecord;
        }

        public void loanDelete(string borrowTime)
        {
            string CommandText = string.Format("delete from loaninfo where
BorrowTime='{0}'", borrowTime);
            using (MySqlConnection deleteConnection = new
MySqlConnection(connectionString))
            {
                deleteConnection.Open();
                MySqlCommand mySqlCommand = new MySqlCommand(CommandText,
deleteConnection);
                mySqlCommand.ExecuteNonQuery();
                deleteConnection.Close();
            }
        }

        public void loanFineProgress(string borrowTime)
        {
            string CommandText = string.Format("update loaninfo set
isReturn='是' where BorrowTime='{0}'", borrowTime);

```

```

        using (MySqlConnection deleteConnection = new
        MySqlConnection(connectionString))
        {
            deleteConnection.Open();
            MySqlConnection transaction;
            //启动事务
            transaction = deleteConnection.BeginTransaction();
            MySqlCommand loanCommand = deleteConnection.CreateCommand();
            //设定SqlCommand的事务和连接对象
            loanCommand.Connection = deleteConnection;
            loanCommand.Transaction = transaction;
            // 开始执行操作
            try
            {
                loanCommand.CommandText = CommandText;
                loanCommand.ExecuteNonQuery();
                // 完成提交
                transaction.Commit();
            }
            catch (Exception ex)
            {
                //数据回滚
                transaction.Rollback();
            }
            deleteConnection.Close();
        }
    }

    /*****借阅相关*****/

```

(10)权限管理相关操作

权限管理主要有：赋权、收回权限、设为学生、设为教职工，只需提供用户名即可实现，方法返回bool类型变量，前端利用该变量判断操作是否成功：

```

    /*****权限相关*****/
    //收回权限
    public bool rollGRANT(string name)
    {
        string getAdmin = string.Format("select * from admin where
        AdminName='{0}'", name);
        using (MySqlConnection rollGRANTConnection = new
        MySqlConnection(connectionString))
        {

```

```

        bool getAdminData = false;
        rollGRANTConnection.Open();
        MySqlCommand getAdminCommand = new MySqlCommand(getAdmin,
rollGRANTConnection);
        string rollGRANT = string.Format("delete from admin where
AdminName='{0}'", name);
        MySqlCommand rollGRANTCommand = new MySqlCommand(rollGRANT,
rollGRANTConnection);
        string setUserPower = string.Format("update user_info set
Cha=0 where UserName='{0}'", name);
        MySqlCommand setUserPowerCommand = new
MySqlCommand(setUserPower, rollGRANTConnection);
        using (MySqlDataReader getAdminDataReader =
getAdminCommand.ExecuteReader())
        {
            if (getAdminDataReader.Read()) getAdminData = true;
            else getAdminData = false;
            getAdminDataReader.Close();
        }
        if (!getAdminData) return getAdminData;
        else
        {
            rollGRANTCommand.ExecuteNonQuery();
            setUserPowerCommand.ExecuteNonQuery();
            return getAdminData;
        }
    }
}

public bool GRANT(string name)
{
    string getAdmin = string.Format("select * from admin where
AdminName='{0}'", name);
    string getUserPassword = string.Format("select * from user_info
where UserName='{0}'", name);
    string grantUserCommand = "";
    using (MySqlConnection GRANTConnection = new
MySqlConnection(connectionString))
    {
        bool isAdminData = false;
        GRANTConnection.Open();
        //读取是否在管理员下
        MySqlCommand getAdminCommand = new MySqlCommand(getAdmin,
GRANTConnection);
        using (MySqlDataReader getAdminDataReader =
getAdminCommand.ExecuteReader())
        {
            if (getAdminDataReader.Read()) isAdminData = true;

```

```

        else isAdminData = false;
        getAdminDataReader.Close();
    }
    //读取密码
    MySqlCommand getUserPasswordCommand = new
MySqlCommand(getUserPassword, GRANTConnection);
    using (MySqlDataReader getUserPasswordDataReader =
getUserPasswordCommand.ExecuteReader())
    {
        if (getUserPasswordDataReader.Read())
        {
            grantUserCommand = string.Format("insert into
admin(AdminName,Password) values('{0}','{1}']", name,
getUserPasswordDataReader["Password"].ToString());
            getUserPasswordDataReader.Close();
        }
    }
    MySqlCommand mySqlgetUserCommand = new
MySqlCommand(grantUserCommand, GRANTConnection);
    string setPowerList = string.Format("update user_info set
Cha=2 where UserName='{0}'", name);
    MySqlCommand setPowerListCommand = new
MySqlCommand(setPowerList, GRANTConnection);
    if (isAdminData) return isAdminData;
    else
    {
        mySqlgetUserCommand.ExecuteNonQuery();
        setPowerListCommand.ExecuteNonQuery();
        return isAdminData;
    }
}

public bool setStudent(string UserName)
{
    bool setStudentSuccess = false;
    using (MySqlConnection setStudentConnection = new
MySqlConnection(connectionString))
    {
        setStudentConnection.Open();
        string setChaStudent = string.Format("update user_info set
Cha=0 where UserName='{0}'", UserName);
        MySqlCommand setChaStudentCommand = new
MySqlCommand(setChaStudent, setStudentConnection);
        if (setChaStudentCommand.ExecuteNonQuery() > 0)
setStudentSuccess = true;
        else setStudentSuccess = false;
        setStudentConnection.Close();
    }
}

```

```

    }
    return setStudentSuccess;
}
public bool setTeacher(string UserName)
{
    bool setTeacherSuccess = false;
    using (MySqlConnection setTeacherConnection = new
MySqlConnection(connectionString))
    {
        setTeacherConnection.Open();
        string setChaTeacher = string.Format("update user_info set
Cha=1 where UserName='{0}'", UserName);
        MySqlCommand setChaTeacherCommand = new
MySqlCommand(setChaTeacher, setTeacherConnection);
        if (setChaTeacherCommand.ExecuteNonQuery() > 0)
setTeacherSuccess = true;
        else setTeacherSuccess = false;
        setTeacherConnection.Close();
    }
    return setTeacherSuccess;
}
/*****权限相关
*****/

```

(11)出版社信息管理相关操作

主要涉及出版社查询、增加、删除、修改方面的方法：

```

/*****出版社相关
*****/
public DataTable getPress()
{
    DataTable press;
    string seekPresses = string.Format("select * from pressinfo");
    using (MySqlConnection seekPressConnection = new
MySqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekPresses, seekPressConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        press = dataTable;
    }
    return press;
}

```



```

        public DataTable showPress()
        {
            DataTable press;
            string showPressCommandText = string.Format("select Press from
pressinfo");
            using (MySqlConnection showPressConnection = new
MySqlConnection(connectionString))
            {
                MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(showPressCommandText, showPressConnection);
                DataTable dataTable = new DataTable();
                mySqlDataAdapter.Fill(dataTable);
                press = dataTable;
            }
            press.Columns[0].ColumnName = "出版社";
            return press;
        }
        public bool pressDelete(string pressName)
        {
            bool isDeleted;
            string selectPressString = string.Format("select * from
booksinfo where Press='{0}'", pressName);
            using (MySqlConnection selectPressConnection = new
MySqlConnection(connectionString))
            {
                selectPressConnection.Open();
                MySqlCommand selectPressCommand = new
MySqlCommand(selectPressString, selectPressConnection);
                MySqlDataReader pressReader =
selectPressCommand.ExecuteReader();
                if (!pressReader.Read())
                {
                    string pressDeleteCommand = string.Format("delete from
pressinfo where Press='{0}'", pressName);
                    using (MySqlConnection pressDeleteConnection = new
MySqlConnection(connectionString))
                    {
                        pressDeleteConnection.Open();
                        MySqlCommand mySqlCommand = new
MySqlCommand(pressDeleteCommand, pressDeleteConnection);
                        mySqlCommand.ExecuteNonQuery();
                        pressDeleteConnection.Close();
                        isDeleted = true;
                    }
                }
            }
            else
            {

```

```

        isDeleted = false;
    }
    selectPressConnection.Close();
}
return isDeleted;
}
public bool pressAdd(string pressName)
{
    bool isAdded;
    string addPressString = string.Format("insert into
pressinfo(Press) values('{0}')" , pressName);
    using (MySqlConnection addPressConnection = new
MySqlConnection(connectionString))
    {
        addPressConnection.Open();
        MySqlCommand mySqlCommand = new
MySqlCommand(addPressString, addPressConnection);
        isAdded = mySqlCommand.ExecuteNonQuery() > 0 ? true : false;
        addPressConnection.Close();
    }
    return isAdded;
}
public bool pressInfoChange(string oldPress, string newPress)
{
    bool isChanged;
    string selectPressString = string.Format("select * from
booksinfo where Press='{0}'", oldPress);
    using (MySqlConnection selectPressConnection = new
MySqlConnection(connectionString))
    {
        selectPressConnection.Open();
        MySqlCommand selectPressCommand = new
MySqlCommand(selectPressString, selectPressConnection);
        MySqlDataReader pressReader =
selectPressCommand.ExecuteReader();
        if (!pressReader.Read())
        {
            string changePressString = string.Format("update
pressinfo set Press='{1}' where Press='{0}'", oldPress, newPress);
            using (MySqlConnection changePressConnection = new
MySqlConnection(connectionString))
            {
                changePressConnection.Open();
                MySqlCommand mySqlCommand = new
MySqlCommand(changePressString, changePressConnection);
                isChanged = mySqlCommand.ExecuteNonQuery() > 0 ?
true : false;

```

```

        changePressConnection.Close();
    }
}
else
{
    isChanged = false;
}
selectPressConnection.Close();
}
return isChanged;
}
/*****出版社相关
*****/

```

(12) 用户信息相关操作

用于管理员查询用户信息时使用的方法，返回DataTable类型变量，使前端DataGridView显示查询内容：

```

/*****用户相关
*****/
public DataTable commonUser()
{
    DataTable commonUser;
    string CommonUserCommandText = "select * from user_info";
    using (MySQLConnection commonUserConnection = new
MySQLConnection(connectionString))
    {
        MySQLDataAdapter mySqlDataAdapter = new
MySQLDataAdapter(CommonUserCommandText, commonUserConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        commonUser = dataTable;
    }
    commonUser.Columns[0].ColumnName = "用户ID";
    commonUser.Columns[1].ColumnName = "用户名";
    commonUser.Columns[2].ColumnName = "密码";
    commonUser.Columns[3].ColumnName = "年龄";
    commonUser.Columns[4].ColumnName = "性别";
    commonUser.Columns[5].ColumnName = "地址";
    commonUser.Columns[6].ColumnName = "手机号码";
    commonUser.Columns[7].ColumnName = "部门";
    commonUser.Columns[8].ColumnName = "权限";
    return commonUser;
}

```

```

public DataTable seekUser(string userClass, string value)
{
    switch (userClass)
    {
        case "用户名":
            userClass = "UserName";
            break;
        case "年龄":
            userClass = "Age";
            break;
        case "性别":
            userClass = "Sex";
            break;
        case "手机号码":
            userClass = "Telephone";
            break;
        case "学院":
            userClass = "Department";
            break;
    }
    string seekUserByClassCommandText = string.Format("select *
from user_info where {0} like '%{1}%', userClass, value);
    string seekUserAllCommandText = string.Format("select * from
user_info");
    DataTable seekUsers;
    if (value != "")
    {
        using (MySqlConnection seekUserConnection = new
MySqlConnection(connectionString))
        {
            MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekUserByClassCommandText, seekUserConnection);
            DataTable dataTable = new DataTable();
            mySqlDataAdapter.Fill(dataTable);
            seekUsers = dataTable;
        }
    }
    else
    {
        using (MySqlConnection seekUserConnection = new
MySqlConnection(connectionString))
        {
            MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekUserAllCommandText, seekUserConnection);
            DataTable dataTable = new DataTable();
            mySqlDataAdapter.Fill(dataTable);
            seekUsers = dataTable;
        }
    }
}

```

```

    }
}
seekUsers.Columns[0].ColumnName = "用户ID";
seekUsers.Columns[1].ColumnName = "用户名";
seekUsers.Columns[2].ColumnName = "密码";
seekUsers.Columns[3].ColumnName = "年龄";
seekUsers.Columns[4].ColumnName = "性别";
seekUsers.Columns[5].ColumnName = "地址";
seekUsers.Columns[6].ColumnName = "手机号码";
seekUsers.Columns[7].ColumnName = "部门";
seekUsers.Columns[8].ColumnName = "权限";
return seekUsers;
}
/*****用户相
关*****/

```

(13)登录记录相关操作

登录记录的查询以及搜索：

```

/*****登录记录相关
*****/
public DataTable seekLoginData(string userClass, string value)
{
    string seekLoginDataByClassCommandText = string.Format("select
loginrecord.UserName,loginrecord.LoginTime from user_info,loginrecord where
user_info.{0} like '{1}%' and user_info.UserName=loginrecord.UserName",
userClass, value);
    DataTable seekUsers;
    using (SqlConnection seekLoginDataConnection = new
SqlConnection(connectionString))
    {
        MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekLoginDataByClassCommandText, seekLoginDataConnection);
        DataTable dataTable = new DataTable();
        mySqlDataAdapter.Fill(dataTable);
        seekUsers = dataTable;
    }
    seekUsers.Columns[0].ColumnName = "用户名";
    seekUsers.Columns[1].ColumnName = "登录时间";
    return seekUsers;
}
public DataTable showAllLoginRecord()
{

```

```

        string seekLoginDataByClassCommandText = string.Format("select
* from loginrecord");
        DataTable seekUsers;
        using (SqlConnection seekLoginDataConnection = new
SqlConnection(connectionString))
        {
            MySqlDataAdapter mySqlDataAdapter = new
MySqlDataAdapter(seekLoginDataByClassCommandText, seekLoginDataConnection);
            DataTable dataTable = new DataTable();
            mySqlDataAdapter.Fill(dataTable);
            seekUsers = dataTable;
        }
        return seekUsers;
    }
}
/*****登录记录相
关*****/

```

5.实验总结

本次数据库系统实验的大作业是设计和实现一个图书管理系统。该系统旨在管理图书馆的图书信息、借阅记录以及读者信息。在本次实验中，我成功完成了以下任务：

- 数据库设计：**我首先进行了数据库设计，确定了系统所需的实体和关系。我创建了几个主要的表，如图书表、读者表和借阅记录表，并定义了它们之间的关系；
- 数据库建模：**基于数据库设计，使用SQL语言创建了相应的表，并定义了各个表之间的外键关系，确保了数据的一致性和完整性；
- 数据库查询：**实现常见的查询操作，如根据图书名称或作者搜索图书、查询某个读者的借阅记录等。我通过使用SQL语句编写了这些查询，并在实验中进行了测试；
- 触发器和存储过程：**为了增强系统的功能，我们使用触发器和存储过程实现了一些自动化的操作。例如，当借阅记录中的归还日期已到时，触发器会自动将借阅状态设置为“超时”；
- 用户界面：**为了方便用户使用系统，我设计了一个简单的用户界面。用户可以通过界面进行图书搜索、借阅和归还图书等操作。

通过这个大作业，我学到了许多有关数据库设计和实现的知识，了解了数据库的重要性，学会了使用SQL语言进行数据库操作，并且掌握了触发器和存储过程的使用；

在实验过程中，我们遇到了一些挑战，如数据库设计的复杂性和SQL语句的编写。但是通过查阅各种博主分享的开发经验，我成功地克服了这些问题，并最终完成了一个功能完善的图书管理系统；

总的来说，本次数据库系统实验的大作业是一次非常有意义和有益的实践。通过设计和实现图书管理系统，我巩固了数据库知识，并提高了数据库操作和项目开发的能力。这个实验不仅让我深入了解了数据库系统的原理和应用，还培养了我即时学习和解决问题的能力。

