**实验5 数据库程序设计**

**一、实验目的**

1. 设计并实现一个精简的图书管理系统，具有入库、查询、借书、还书、借书证管理等基本功能
2. 通过本次设计来加深对数据库的了解和使用，同时提高自身的系统编程能力。

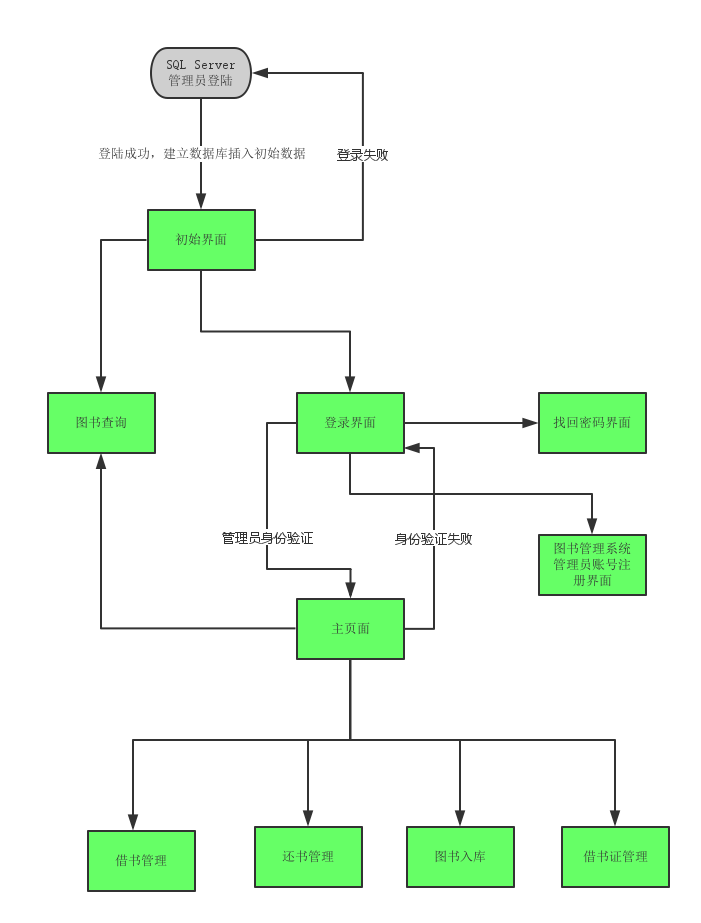
**二、实验平台**

1. 开发工具：Microsoft C#
2. 数据库平台：MySQL

**三、总体设计**

1. 系统架构描述

本系统主要包括管理员登录、管理员账号注册、管理员账号密码找回、图书查询、借书管理、还书管理、借书证管理八大功能模块。系统处理基本流程如下：



系统初始时需要用户登录本地的MySQL管理员账号以进行建库以及往后的数据操作。登录后进入第一页面，可以查询图书库或登录账号和创建图书管理系统账号。查询图书库不需要登录。登陆后可以进入图书管理系统进行图书查询、图书入库、还书管理、借书管理以及借书证管理。

各个功能模块说明如下：

|  |  |
| --- | --- |
| 模块名称 | 功能描述 |
| MySQL Server管理员登录 | 建库，初始化数据以及后续的对数据库的数据操作 |
| 图书系统管理员登陆 | 根据管理员ID和密码登录系统 |
| 图书系统管理员注册 | 1. 输入用户名、密码、真实姓名及联系方式 2. 若用户名重复则注册失败 |
| 找回管理员密码 | 1. 根据用户输入的真实姓名和联系方式来找回相应的用户名和密码。 2. 若找不到相应的信息则找回密码失败 |
| 图书查询 | 1. 按书的类别，书名，出版社，年份，作者，价格进行查询。 2. 可以点击标题来对相对的字段进行排序 |
| 借书 | 1. 输入借书证卡号，自动显示该借书证的所有借出记录条。 2. 记录条包含FID（标识列）、借书卡号、借出书籍的书号、借出日期、归还日期（若未还则显示“未还书”）、经手人（管理员ID） |
| 还书 | 1. 输入借书证卡号，自动显示该借书证所有借出记录条 2. 记录条包含FID（标识列）、借书卡号、借出书籍的书号、借出日期、归还日期（若未还则显示“未还书”）、经手人（管理员ID） 3. 输入书号，如果该书在已借书籍列表内，则还书成功，同时库存加一，否则输出出错信息呢 |
| 借书证管理 | 增加或删除借书证 |

2. 数据库表设计

图书信息表(Books)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 说明 |
| BookNo | Varchar(50) | Yes | 书号 |
| BookType | Varchar(50) |  | 图书类别 |
| BookName | Varchar(50) |  | 书名 |
| Publisher | Varchar(50) |  | 出版社 |
| Year | Int |  | 出版年份 |
| Author | Varchar(50) |  | 作者 |
| Price | Numeric(8, 2) |  | 图书单价 |
| Total | Int |  | 总藏书量 |
| Storage | Int |  | 库存数 |
| UpdateTime | Datetime |  | 添加时间 |

借书证表(LibraryCard)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 说明 |
| CardNo | Varchar(50) | Yes | 卡号 |
| Name | Varchar(50) |  | 姓名 |
| Department | Varchar(50) |  | 单位 |
| CardType | Varchar(50) |  | 类别 |
| UpdateTime | Datetime |  | 添加时间 |

用户表(Users)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 说明 |
| uid | Varchar(50) | Yes | 用户ID |
| upw | Varchar(50) |  | 密码 |
| Name | Varchar(50) |  | 姓名 |
| Contact | Varchar(50) |  | 联系方式 |

借书记录表(LibraryRecords)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 主键 | 说明 |
| FID | Int | Yes | 标识列 |
| CardNo | Varchar(50) |  | 书号 |
| BookNo | Varchar(50) |  | 书号 |
| LentDate | Datetime |  | 借书日期 |
| ReturnDate | Datetime |  | 还书日期 |
| Operator | Varchar(50) |  | 经手人(管理员ID) |

3、所用开发技术

本次开发所用技术主要从两方面来说明：MySQL，C#

本次程序编写上，使用的是C#开发工具。对于访问和连接数据库的方法，因为Microsoft C#有自带的用于访问MySQL数据库的API，使用起来极度方便且简洁，再且C#的winform开发工具上手极其简单，

因此C#也成了本次实验的首选编程工具。

MySQL是一个[**关系型数据库管理系统**](http://baike.baidu.com/view/1450387.htm)**，**由瑞典MySQL AB 公司开发，目前属于 [Oracle](http://baike.baidu.com/view/15020.htm) 旗下公司。MySQL 最流行的[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm)，在 WEB 应用方面MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](http://baike.baidu.com/view/1088.htm)的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](http://baike.baidu.com/view/394804.htm)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

由于其社区版的性能卓越，搭配 [PHP](http://baike.baidu.com/view/99.htm) 和 [Apache](http://baike.baidu.com/view/28283.htm) 可组成良好的开发环境。

C#是[微软公司](http://baike.baidu.com/view/39784.htm)发布的一种面向对象的、运行于[.NET Framework](http://baike.baidu.com/view/18370.htm)之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C#是微软公司研究员Anders Hejlsberg的最新成果。C#看起来与Java有着惊人的相似；它包括了诸如单一继承、接口、与Java几乎同样的语法和编译成中间代码再运行的过程。但是C#与Java有着明显的不同，它借鉴了Delphi的一个特点，与COM（组件对象模型）是直接集成的，而且它是微软公司 .NET windows网络框架的主角。

C#是一种安全的、稳定的、简单的、优雅的，由C和[C++](http://baike.baidu.com/view/824.htm)衍生出来的[面向对象](http://baike.baidu.com/view/125370.htm)的编程语言。它在继承C和C++强大功能的同时去掉了一些它们的复杂特性（例如没有[宏](http://baike.baidu.com/view/57.htm)以及不允许多重继承）。C#综合了[VB](http://baike.baidu.com/view/3063.htm)简单的可视化操作和C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向[组件](http://baike.baidu.com/view/379950.htm)编程的支持成为[.NET](http://baike.baidu.com/view/4294.htm)开发的首选语言。[1]

C#是[面向对象](http://baike.baidu.com/view/125370.htm)的编程语言。它使得程序员可以快速地编写各种基于MICROSOFT .NET平台的应用程序，MICROSOFT .NET提供了一系列的[工具](http://baike.baidu.com/view/74746.htm)和服务来最大程度地开发利用计算与通讯领域。

C#使得C++程序员可以高效的开发程序，且因可调用由 C/C++ 编写的本机原生函数，因此绝不损失C/C++原有的强大的功能。因为这种[继承](http://baike.baidu.com/view/125322.htm)关系，C#与C/C++具有极大的相似性，熟悉类似语言的开发者可以很快的转向C#。

对于C#访问MySQL数据库主要分为以下三步：

1. 建立connection handler (MySqlConnection)

e.g. 

1. 建立command handler (MySqlCommand)

e.g.



对command handler赋予指令，指令格式与MySQL操作指令相同



运行指令



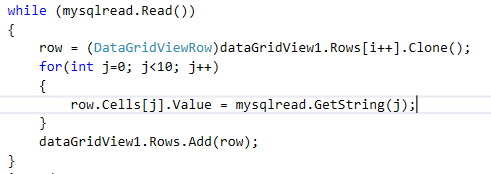
1. 建立data-reader handler (MySqlDataReader)



对command handler运行读取器并赋予data-reader handler



读取data-reader handler所获取到的数据



4、成员任务描述

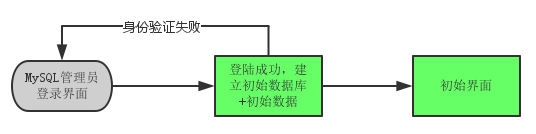
本次从数据库设计，到各个模块的功能编写，皆为一人独立完成。程序模块设计上主要参考了实验报告5模块的指导书说明，结合自己的理解来理清思路，在明确要完成的目标后，便收集和查阅了一些技术资料来学习，了解了C#中的MySQL API组件的使用及MySQL工具的使用，最后选定C# + MySQL方案进行开发完成本系统，同时花费了若干小时来安装配置开发环境。

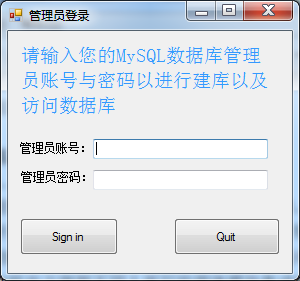
**四、详细设计**

1、MySQL Server管理员账号登录界面

本界面的存在主要目的是获取用户的本地MySQL数据库的账号和密码用以数据库建立、初始化数据库以及后续对于数据库的各类操作。正因为多了这一项功能所以可以赋予本程序在其他电脑上也能运行的权限，其他程序缺少了此功能可能会导致只能读取到开发者电脑上的数据库以至于缺少了程序的跨平台性。

主要流程如下：





程序代码：

string M\_str\_sql = "server=localhost;user=" + rootuid + ";password=" + rootpw + ";";

MySqlConnection mycon = new MySqlConnection(M\_str\_sql);

try

{

mycon.Open();

}

catch

{

MessageBox.Show("Login failed");

return;

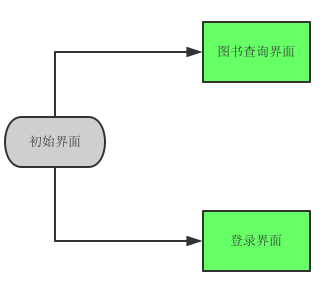
}

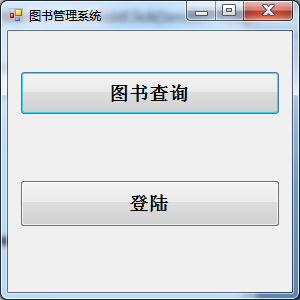
Rootuid和rootpw是记录着MySQL管理员的用户名和密码，若登录失败则显示“Login failed”

2、初始界面

本界面主要作用于引导用户进行图书查询或进入登录界面

主要流程如下：

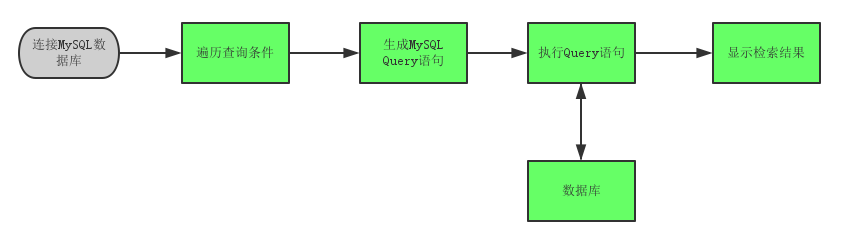




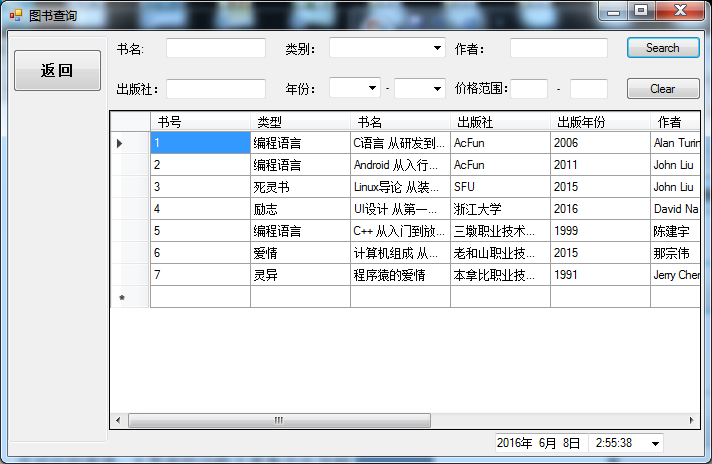
3、图书查询界面

图书查询不需要登录，在系统初始化后可以直接使用。其基本工作流程是：输入查询条件→组合查询条件→执行查询→返回结果。若查询结果为空则显示所有库存书籍。

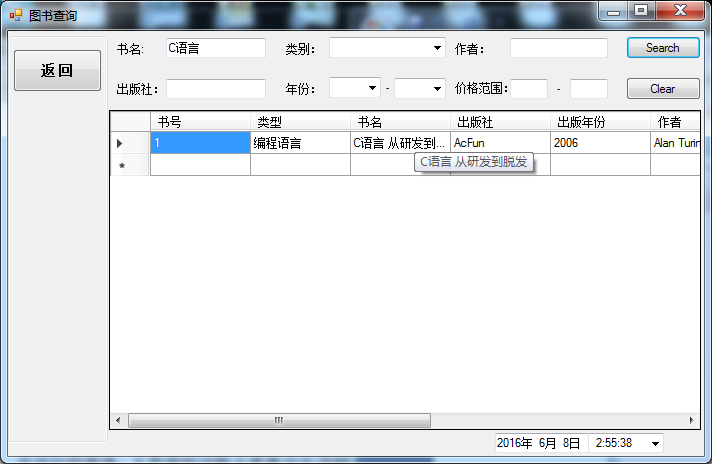
主要流程如下：



下图为当查询结果未全空时显示的所有库存书籍：



下图为通过书籍名字进行单一查询：



具体实现代码为下：

M\_str\_sql = "SELECT \* FROM lms.books";

if (bookname != "")

{

where += ("BookName LIKE " + "'%" + bookname + "%'");

isHead = false;

}

if(booktype != "")

{

if (!isHead) where += " and";

where += (" BookType = " + "'" + booktype + "'");

isHead = false;

}

if(publisher != "")

{

if (!isHead) where += " and";

where += (" Publisher = " + "'" + publisher + "'");

isHead = false;

}

if(yearinit != "")

{

if (!isHead) where += " and";

where += (" Year >= " + yearinit);

isHead = false;

}

if(yearfinal != "")

{

if (!isHead) where += " and";

where += (" Year <= " + yearfinal);

isHead = false;

}

if(author != "")

{

if (!isHead) where += " and";

where += (" Author LIKE " + "'%" + author + "%'");

isHead = false;

}

if(priceinit != "")

{

if (!isHead) where += " and";

where += (" Price >= " + priceinit);

isHead = false;

}

if(pricefinal != "")

{

if (!isHead) where += " and";

where += (" Price <= " + pricefinal);

isHead = false;

}

if(where != "")

{

M\_str\_sql += " WHERE " + where + ";";

}

MySqlCommand mysqlcom = new MySqlCommand(M\_str\_sql, mycon);

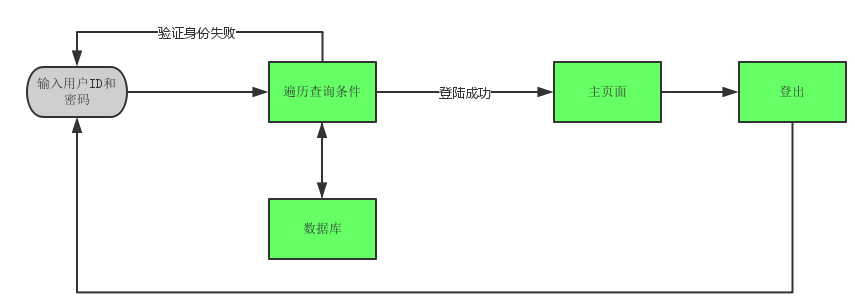
mysqlcom.ExecuteNonQuery();

这里用到的算法是逐个遍历搜索条件，若搜索条件不为空则将搜索条件加入Query语句，若为空则跳过。

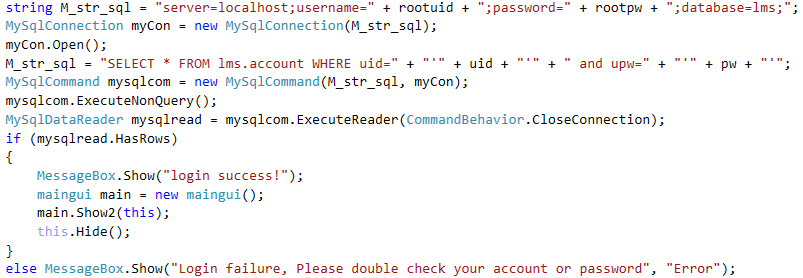
4、登录界面

Username捕获用户输入的UserID，Password捕获用户输入的Password以此来作为查询数据库条件，若查询完数据库后找到相应的记录则登陆成功，否则登录失败。

主要流程如下：







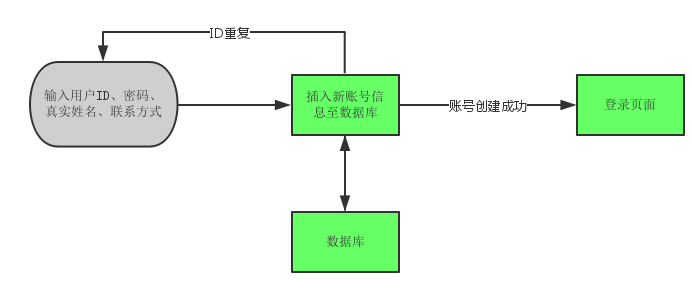
1.登陆MySQL Server并访问数据库。

2. 以uid和pw作为条件检索“Account”数据库，若发现了匹配的结果就引领用户进入主页面并显示登陆成功，否则显示登录失败并停留在登录页面。

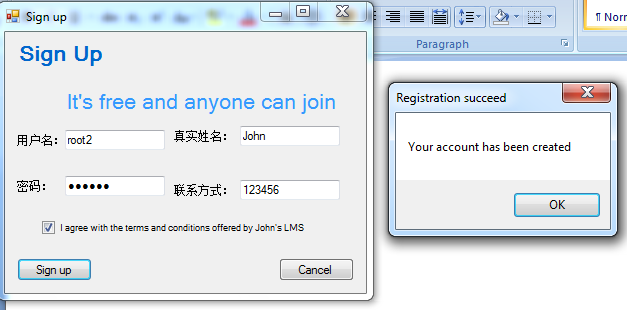
5、管理员账号注册页面

本模块实现了通过用户输入指定的用户信息（用户ID，密码，姓名，联系方式）并插入至数据库的Account库中，若用户ID非重复就显示账号建立成功，否则显示失败并返回注册页面。

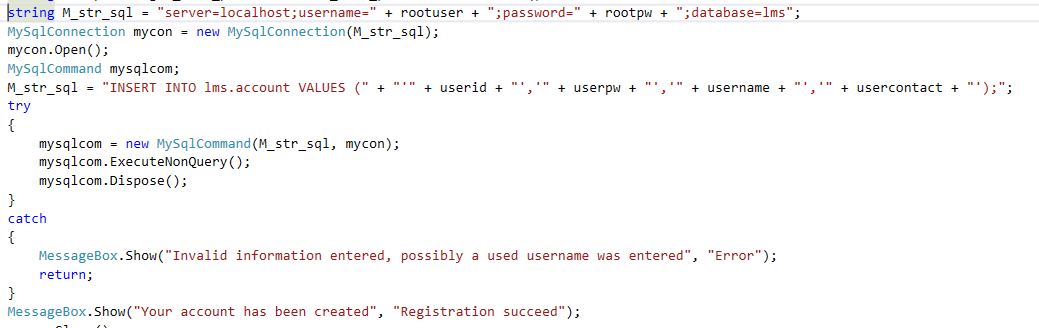
主要流程如下：







实现代码：

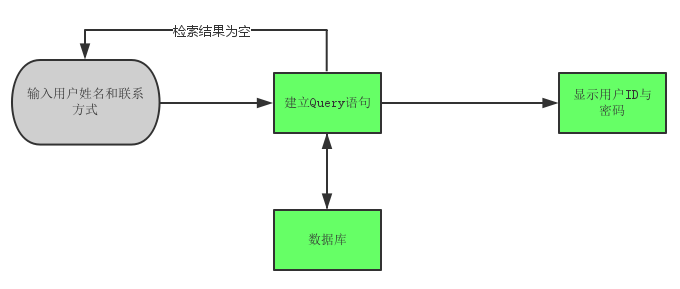


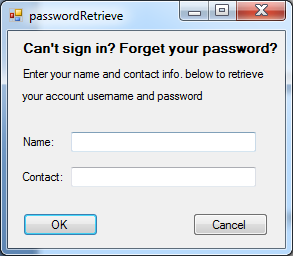
1. 连接数据库
2. 捕获用户输入的用户ID，密码，姓名和联系方式
3. 插入至数据库中
4. 若失败就抛出异常并返回注册界面，否则便显示注册成功并返回登录页面

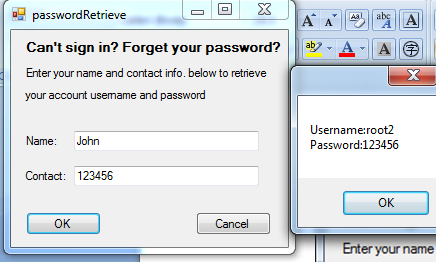
6、密码找回页面

本模块实现了通过用户输入的真实姓名和联系方式并以此为条件建立Query语句检索数据库，若检索结果不为空便返回用户ID和密码，否则显示没有找到账号。

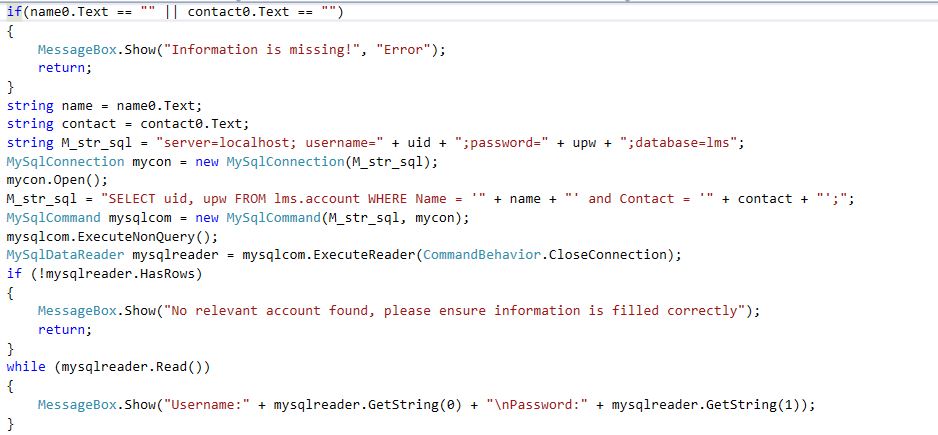
主要流程如下：







实现代码：

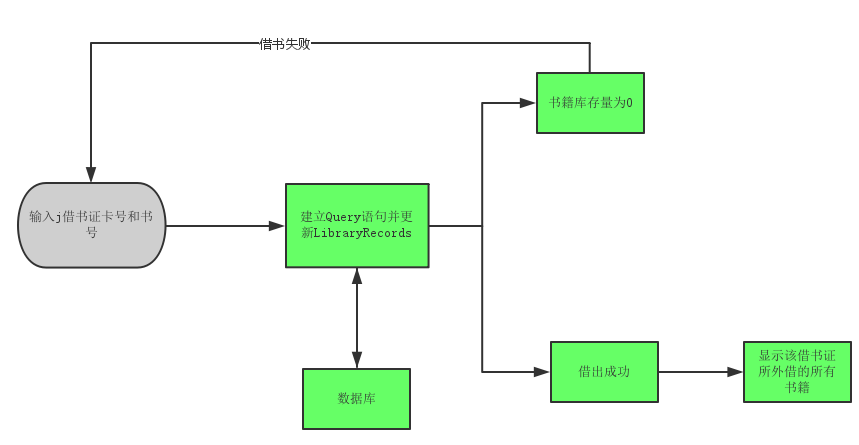


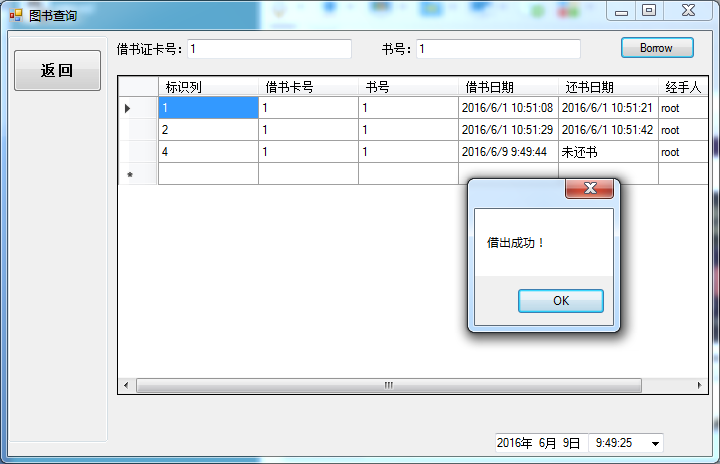
1. 若用户输入名字或联系方式为空便抛出异常并显示检索信息不足
2. 连接数据库
3. 基于捕获到的用户输入姓名与联系方式建立Query语句并执行
4. 若返回结果为空便显示“未找到相关账号信息”，否则显示检索到的用户ID和密码。

7、借书管理模块

本模块主要实现了根据用户输入的借书证卡号和书号进行借书的操作。若被借出书籍库存量大于等于1时便显示借出成功，否则显示借出失败。借出成功后会显示该借书证所外借的所有借书记录。

主要流程如下：





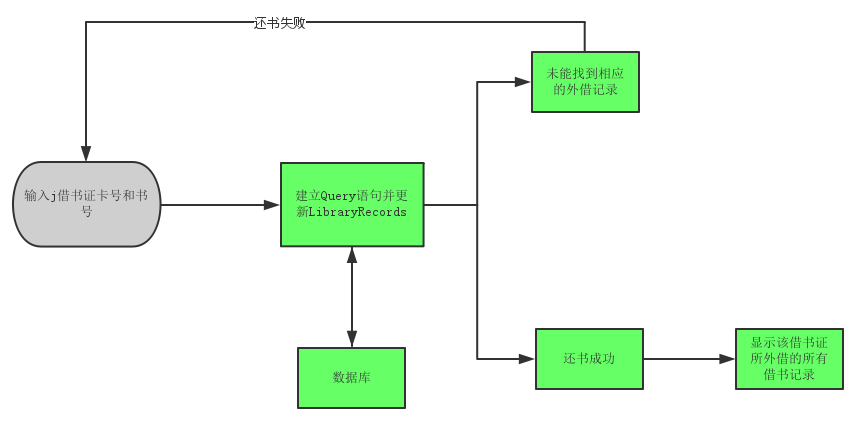
实现思路：

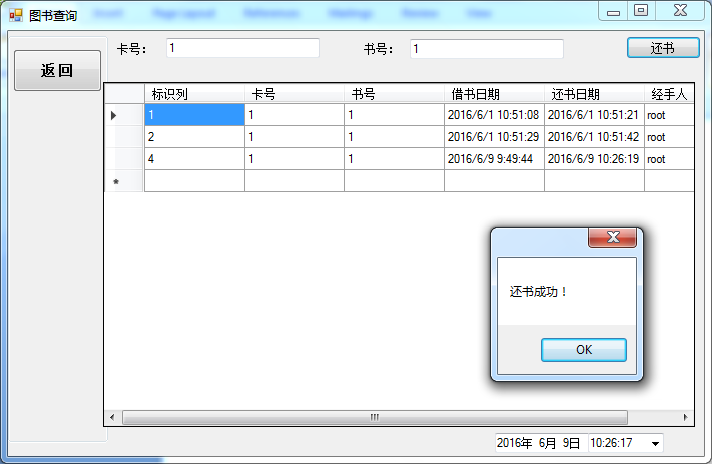
1. 若借书证卡号或书号为空则抛出异常“缺失信息”。
2. 连接数据库。
3. 根据书号搜寻书籍，若库存量低于1便显示库存量不足否则库存量减一。
4. 根据上列信息建立新条目并插入至LibraryRecords。
5. 根据借书证检索数据库并显示该借书证的所有外借书籍。

8、还书管理模块

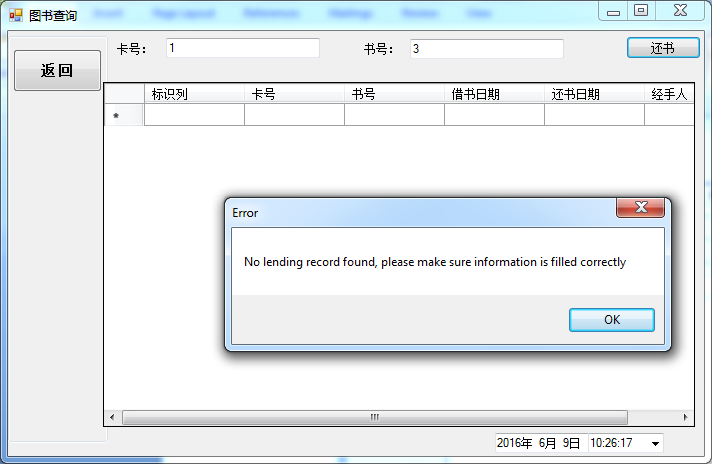
本模块实现的功能与借书管理模块相似，当用户输入借书证卡号以及书号后以此建立Query语句检索LibraryRecords中最老的一条未归还书籍的借书记录并为此更新还书日期并为相应书籍的库存量加一，随后显示该借书证所外借的所有借书记录。若未能找到相应的借书记录则显示还书失败。

基本流程如下：





当未找到相应的借书记录时：



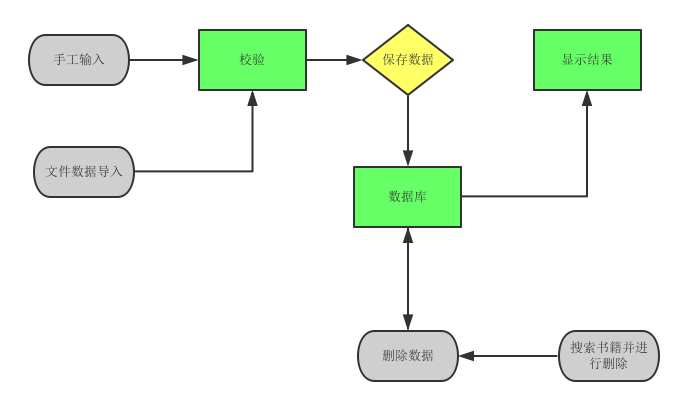
实现思路：

1. 若借书证卡号或书号为空则抛出异常“缺失信息”。
2. 连接数据库。
3. 根据借书证卡号和书号找到最老的一条未归还书籍的借书记录，并为此更新还书时间。
4. 根据书号找到数据库里的书籍条并为其库存量加一。
5. 显示该借书证的所有借书记录。

9、图书管理模块

本模块实现了根据用户输入的图书信息进行录入、批量导入书籍以及删除书籍的功能。

具体流程如下：



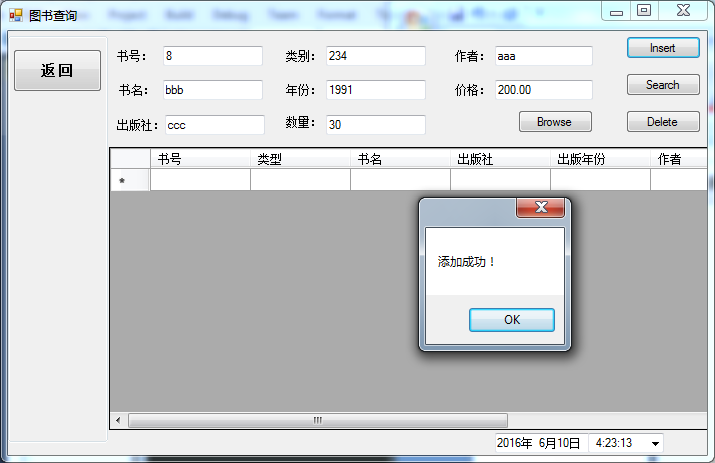
录入新书籍的操作如下：

1.输入图书信息，每一列都不能为空。

2.点击“Insert”录入到数据库中。

操作示例：







批量导入的操作如下：

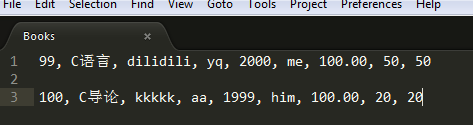
1.以 “书号，种类，书名，出版社，出版年份，作者，价钱，总藏书量，库存量”的格式写入到文档中。

2.点击“Browse”并选中记录了书籍信息的文档。

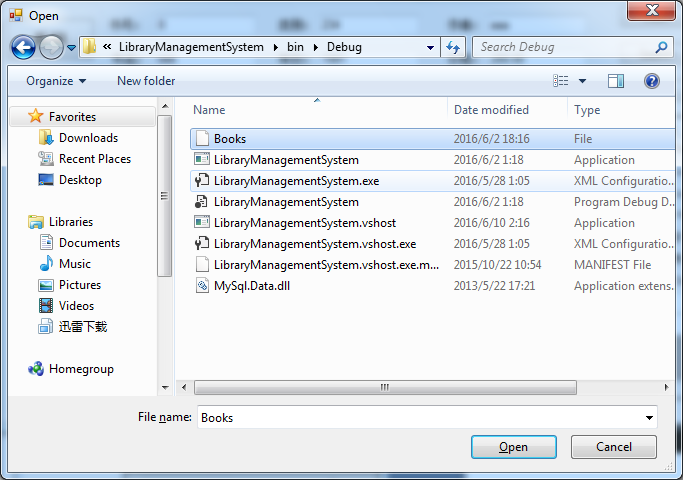
3.点击确定。

操作示例：

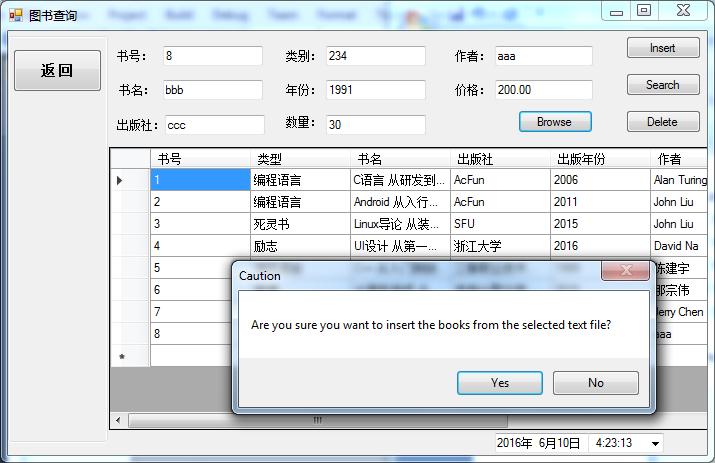
在文档中写入书籍信息



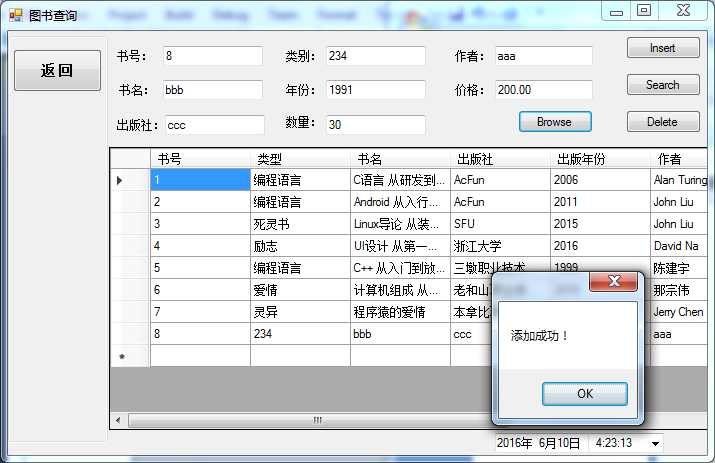
点击“Browse”并找到文档所在位置。

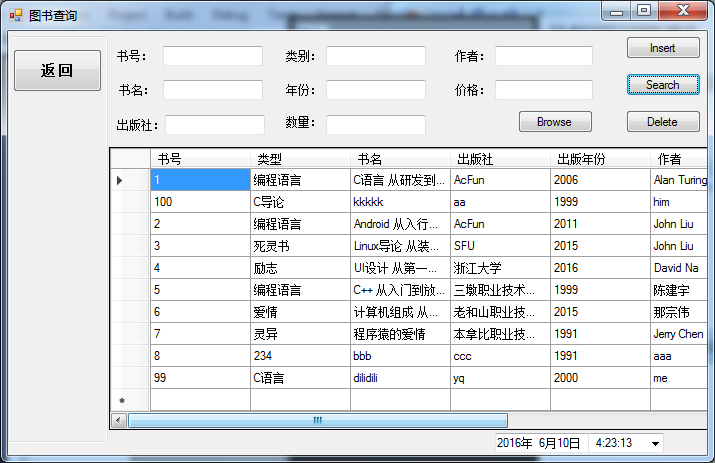


点击Yes



添加成功





删除书籍的操作流程如下：

1.输入书籍信息。

2.搜寻书籍。

3.选择书籍并点击“Delete”。

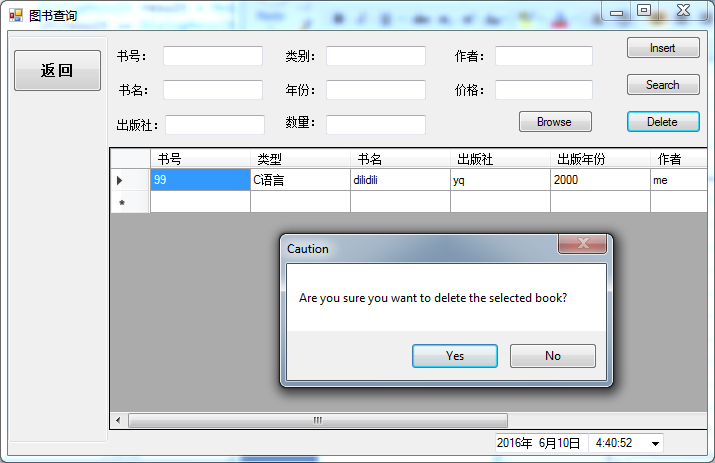
4.点击确认删除书籍。

操作示例：

输入搜索条件（若所有条件为空则判断为显示所有书籍）



点击搜索并选中要删除的书籍



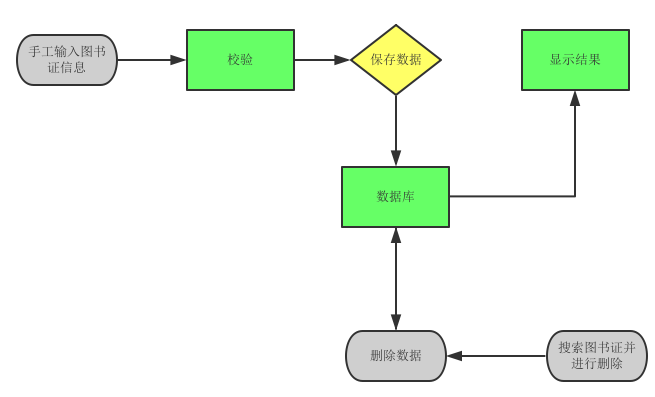
书籍删除成功。



10、借书证管理模块

本模块实现了通过用户输入的卡号，姓名，单位和类别的借书证信息对数据库进行更新和对数据库中的现存借书证进行删除的操作。

具体流程如下：



添加借书证操作流程：

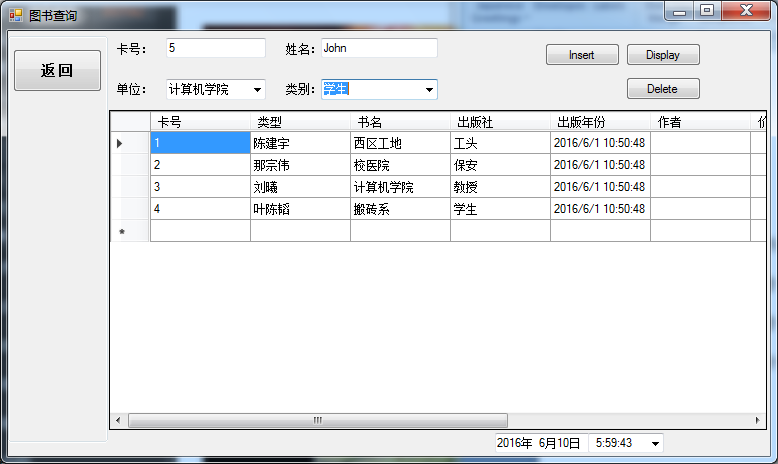
1.输入借书证信息

2.校对借书证信息，若卡号重复则退回重新输入

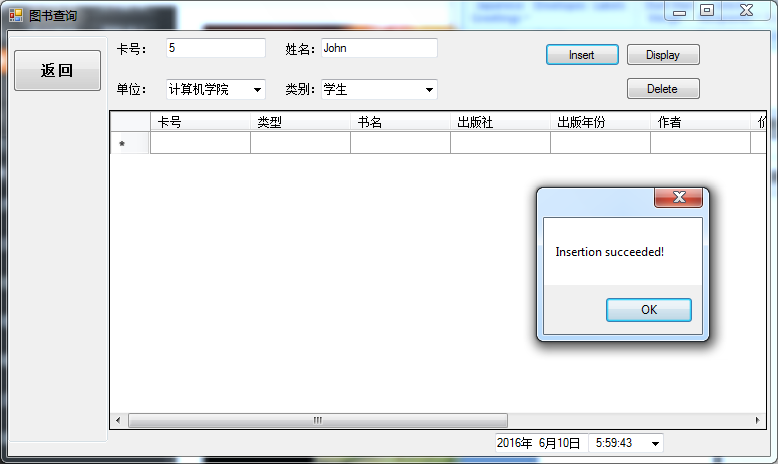
3.插入新借书证至数据库中

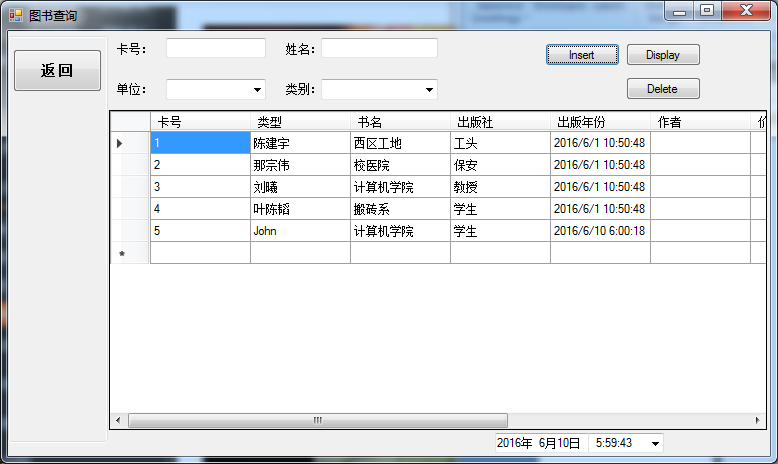
操作示例：

填写借书证信息



添加成功



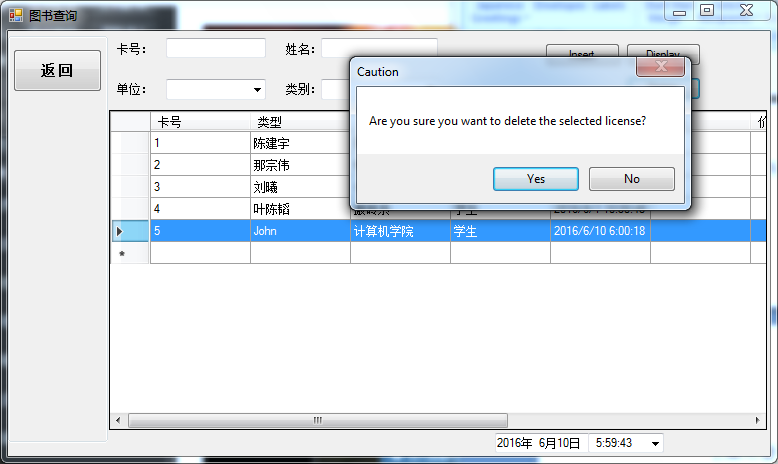


删除借书证流程：

1.填写借书证信息，选取目标借书证。

2.点击删除借书证

操作示例：



删除成功

