

目 录

第一章 绪论	3
1.1 编写目的	3
1.2 项目背景	3
1.3 术语定义	3
第二章 系统需求分析	4
2.1 功能需求分析	4
2.1.1 用户管理模块	4
2.1.2 图书管理模块	4
2.1.3 借还书模块	4
2.2 系统性能需求分析	5
2.2.1 精度	5
2.2.2 时间特性	5
2.2.3 灵活性	5
2.2.4 数据管理能力要求	5
2.2.5 故障处理要求	5
2.3 系统相关功能描述	6
2.3.1 图书管理	6
2.3.2 用户管理	6
2.3.3 借书功能	6
2.3.4 还书功能	6
2.3.5 对图书进行查询	6
2.3.6 密码修改	7
第三章 系统总体设计	8
3.1 系统架构	8
3.2 功能架构图	8
3.2 E-R 图设计	9
第四章 系统详细设计与实现效果	11
4.1 数据库接口	11
4.2 基本功能	12
4.2.1 首页	12
4.2.2 登陆	12
4.2.3 密码修改	13
4.3 用户管理功能	15
4.3.1 添加用户	15
4.3.2 查询用户	16
4.3.3 修改用户	17
4.3.4 删除用户	18
4.4 图书管理功能	19

4.4.1 添加图书	19
4.4.2 图书查询	20
4.4.3 修改图书	21
4.4.4 删除图书	22
4.5 借还书功能	23
4.5.1 借阅图书	23
4.5.2 归还图书	24
4.6 借阅信息查询	25
结束语	27
附录	28

第一章 绪论

1.1 编写目的

明确图书管理系统的详细需求，提供使用单位确认系统的功能和性能，作为软件设计人员的设计依据和使用单位的验收标准，对图书管理系统软件功能的实现和评判进行描述，为开发人员、维护人员、客户人员间提供共同的协作而创立基础。

1.2 项目背景

图书馆作为一种信息资源的集散地，图书和用户借阅资料繁多，包含很多的信息数据的管理，现今，有很多的图书馆都是初步开始使用，甚至尚未使用计算机进行信息管理。根据调查得知，他们以前对信息管理的主要方式是基于文本、表格等纸介质的手工处理，对于图书借阅情况的统计和核实等往往采用对借书卡的人工检查进行，对借阅者的借阅权限、以及借阅天数等用人工计算、手抄进行。数据信息处理工作量大，容易出错；由于数据繁多，容易丢失，且不易查找。总的来说，缺乏系统，规范的信息管理手段。有必要建立一个图书管理系统，使图书管理工作规范化，系统化，程序化，避免图书管理的随意性，提高信息处理的速度和准确性，能够及时、准确、有效的查询和修改图书情况。

1.3 术语定义

- 系统：图书管理软件
- 图书信息：图书的编号、名称、作者、出版社、出版日期、库存、ISBN、馆藏位置、借阅状态等信息，便于读者查询借阅。
- 借阅规则：对借阅者有一些借阅册数要求，对用户超时还书有一定的罚款措施。
- 借书记录：包括借阅者的姓名、ID 号以及所借书的书名和借书日期等信息。

第二章 系统需求分析

2.1 功能需求分析

2.1.1 用户管理模块

用户管理模块的功能及功能描述如图表 2-1 所示。

表 2-1 用户管理模块的功能及功能描述

模块		功能描述
用户管理模块	添加用户信息	包括（用户名、密码、院系、用户类型、当前借阅数、账号状态）。
	查询用户	建立查询表单，设置查询条件以及默认查询结果每页显示条数。
	修改用户	显示该用户的所有信息，所有信息都可被修改除了用户的唯一标识 ID 不可被修改。
	删除用户信息	设置系列删除条件，满足条件时输入用户名或者用户编号可进行删除用户信息。

2.1.2 图书管理模块

图书管理模块的功能及功能描述如图表 2-2 所示。

表 2-2 图书管理模块的功能及功能描述

模块		功能描述
图书管理模块	图书录入	录入信息包括（书名、作者、出版社、出版日期、库存、ISBN、馆藏地、借阅状态）。
	图书修改和删除	系统管理员修改图书的基本信息并确保系统删除操作只允许对外借的图书进行操作。
	图书查询	查询的条件包括（全部、书名、作者、出版社）以及设置搜索框搜索条件。

2.1.3 借还书模块

借还书模块的功能及功能描述如图表 2-3 所示。

表 2-3 借还书模块的功能及功能描述

模块		功能描述
借还书模块	借阅图书	设置借书条件，用户需满足借书条件后方可登录系统进行借书，以及借阅过程的该书库存和用户借阅量。

	借阅记录查询	输入借书卡号或者图书 ID 可获得其相关信息。
	归还图书	由管理员对具体还书用户进行归还操作，还书时要显示用户所借图书和未归还图书用户的相关信息。
	修改图书记录	用户归还图书后，将该书库存加一，并显示该书的库存容量。

2.2 系统性能需求分析

2.2.1 精度

图书管理系统对数据的精度要求是根据信息存储的形式、借书还书的结果等量化而制定的。

查询时应保证查全率，所有相应域包含查询关键字的记录都应能查到；

查询时应保证查准率，查到的记录应与给定的单项或组合查询条件不完全匹配的模糊查询；

录入数据合法性的检验应当精确；

2.2.2 时间特性

由于此开发项目针对图书馆，使用频度较高，使用性要求比较高。为防止对信息资料和管理程序的恶意破坏，要求有较为可靠的安全性能。总之，要求稳定、安全、便捷，易于管理和操作。

2.2.3 灵活性

作为独立运行的系统和其他管理系统集成的系统。

图书管理系统的设计是做为独立运行的系统而进行的。本系统具有独立的服务器系统和数据库系统，具有完善数据输入输出功能和数据维护及查询的报表生成与打印系统。

可能提供鼠标选择和键盘输入双重输入功能。方便用户操作和管理。

2.2.4 数据管理能力要求

数据管理分为增加（INSERT）、修改（UPDATE）、和删除（DELETE）。

2.2.5 故障处理要求

正常使用时不应出错，若运行时遇到不可恢复的系统错误，也必须保证数据库完好无损。

要在项目报名时的每隔一段时间进行数据备份，以免在资料意外丢失时，无法进行恢复。

对系统故障的处理要求区分故障的严重程度，尽可能的对错误进行恢复。随时监控，在文档、报表处理，打印机，操作系统等软硬件出现故障时、具备备份数据的功能，并及时反映到主机中。

2.3 系统相关功能描述

2.3.1 图书管理

图书基本情况的录用：对于购进的新书，系统必须具备图书信息资料的录入功能。当图书资料发生变化，如图书丢失或有错误信息输入时，则应能够及时对数据进行修改和补充。图书基本情况包括录入、修改、删除等基本操作。

图书的基本信息包括图书的编号、名称、作者、出版社、出版日期、库存、ISBN、馆藏位置、借阅状态的描述。

录入：包括除图书编号的之外的所有信息。

修改：包括除图书编号的之外的所有信息。

删除：只允许对未外借的图书进行操作。

2.3.2 用户管理

管理员进行借书者用户进行管理：对系统用户进行添加、修改、删除等基本操作

用户信息包括用户编号、用户名、密码、读者类型、院系、当前借阅数、读者账号状态。

添加：包括除用户编号之外的所有信息。

修改：包括除用户编号之外的所有信息。

删除：只允许对未有外借书籍的用户进行操作。

2.3.3 借书功能

用户进行借书，借书时修改用户表的当前借阅数以同时向借阅表添加借阅记录（包括用户 id、图书 id、借阅时间）。

2.3.4 还书功能

用户还书，由管理员进行操作。还书时修改用户表的当前借阅数以同时修改借阅表相关借阅记录添加还书日期。

用户超时还书，由管理员进行操作，还书时根据超时还书的时间按一定比例收取一定费用。

2.3.5 对图书进行查询

图书查询功能可无需登录也可使用。可通过书名、作者、出版社等方式从查询，可进行模糊查询。

2.3.6 密码修改

每个用户都可修改其密码。

第三章 系统总体设计

3.1 系统架构

本程序是在 eclipse 环境中开发的基于开发 java 开发语言的软件。本系统包括图书管理、用户管理、借还书管理。

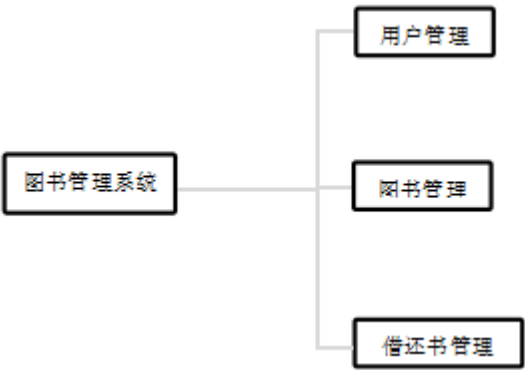


图 3-1 系统组织结构图

3.2 功能架构图

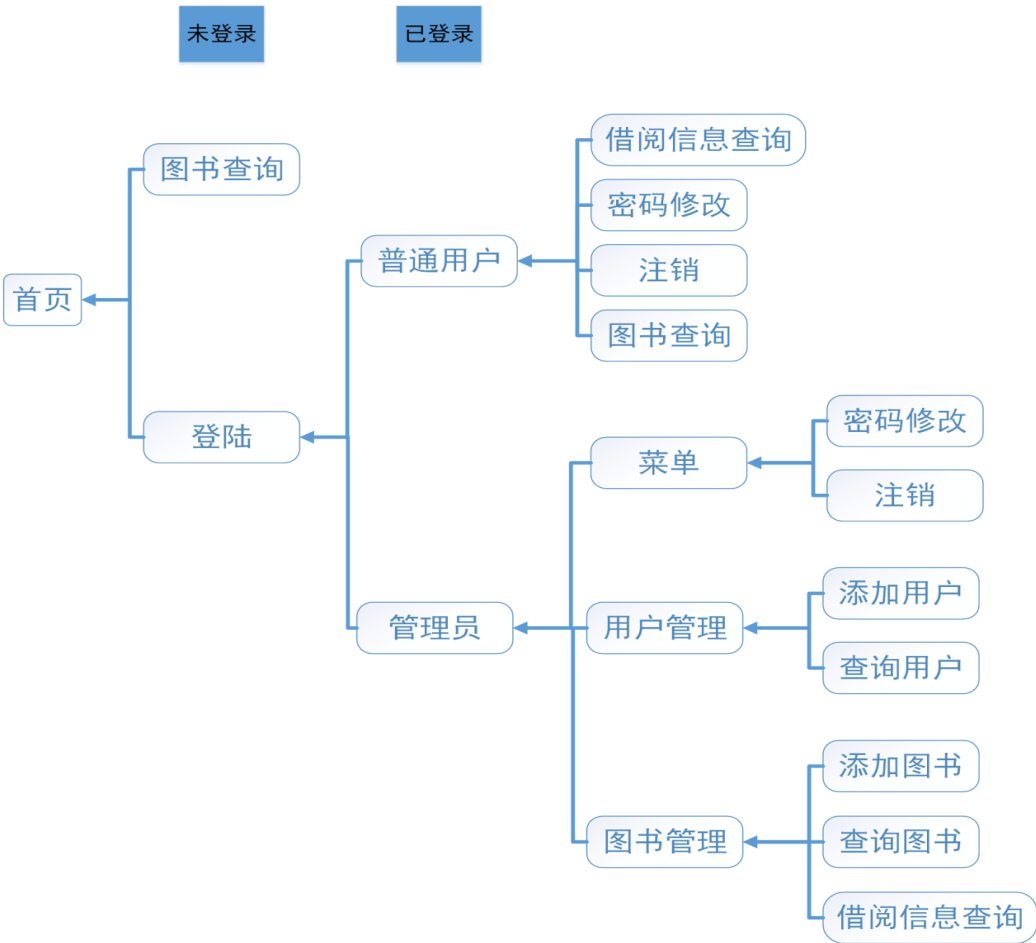


图 3-2 系统功能架构图

3.2 E-R 图设计

图书类系统主要实体如下：

(1) 图书类实体包含主要属性有：书名、作者、出版社、出版日期、库存、ISBN、馆藏地、借阅状态。图书管理员可以录入图书、借阅图书和归还图书，图书类实体与借阅类实体和用户类实体存在一对多关系。管理员实体 E-R 图如 3-3 所示：

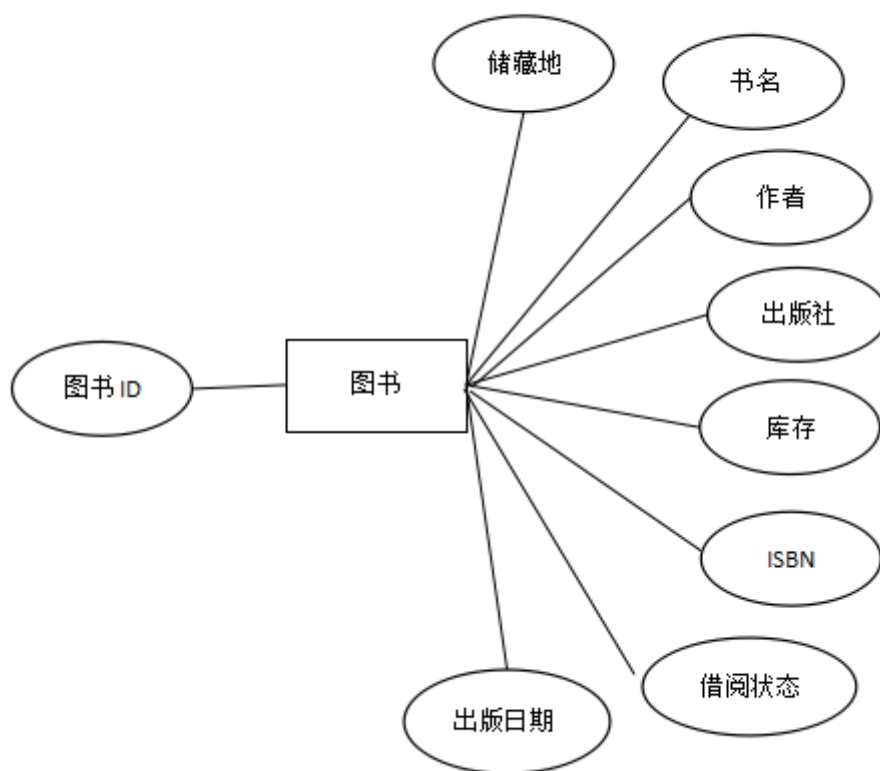


图 3-3 图书类实体 E-R 图

(2) 用户实体包含主要属性有：用户名、密码、院系、用户类型、当前借阅数、账号状态。用户可以查询、借阅图书和修改除用户 ID 之外的用户信息。用户实体 E-R 图如 3-4 所示：

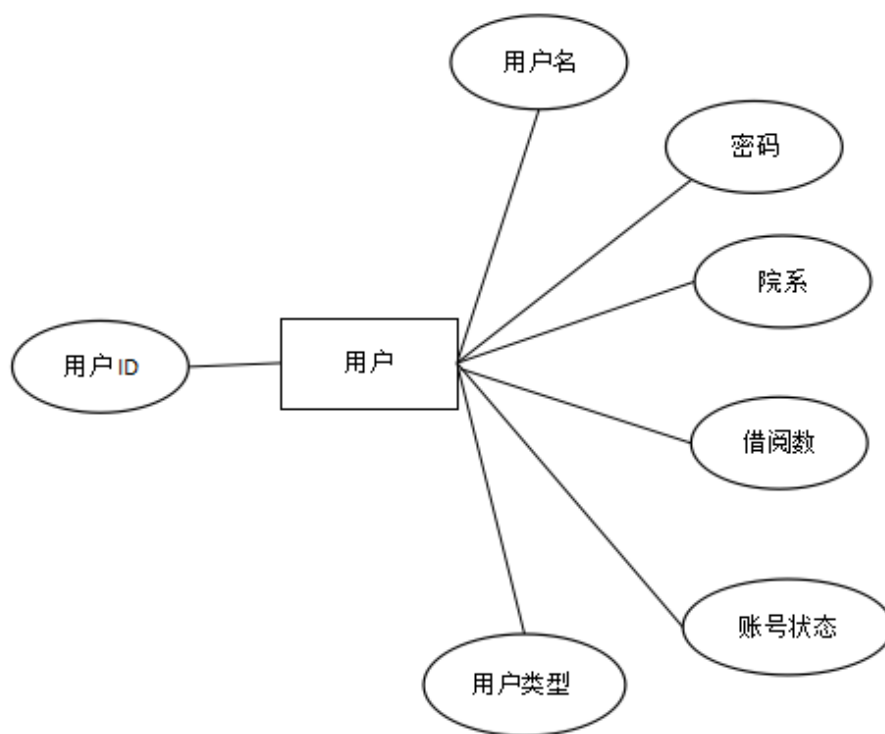


图 3-4 用户实体 E-R 图

(3) 借阅记录实体包含主要属性有：借阅者 ID、图书 ID、借出日期、归还日期。借阅记录实体 E-R 图如 3-5 所示：

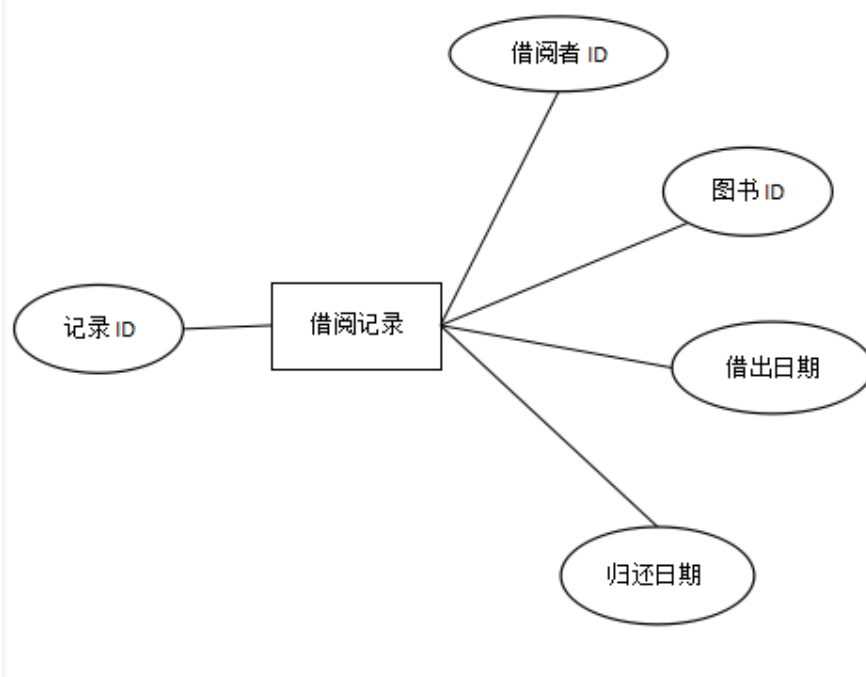


图 3-5 借阅记录实体 E-R 图

第四章 系统详细设计与实现效果

4.1 数据库接口

用户表：（UserDao）

- public User **login**(String username): //登陆专用，通过用户名返回对应的用户信息
- public User **findById**(int id): //通过用户 id 返回对应的用户信息
- public ArrayList<User> **check**(User user,int page): //通过 user 对象（user 内存储搜索条件），返回匹配的用户信息列表，每次查找最多返回 20 条，使用 page 进行定位数据
- public boolean **password**(int id,String password): //通过用户 id，以及传过来的新密码，修改对应的用户密码
- public boolean **add**(User user): //通过传过来的 user 对象添加用户
- public boolean **modify**(User user): //通过传过来的 user 对象修改用户
- public boolean **addNumById**(int uid): //通过传过来的用户 id 修改借阅数量，即增加借阅数量
- public boolean **cutNumById**(int uid): //通过传过来的用户 id 修改借阅数量，即减少借阅数量
- public boolean **delete**(int id): //通过传过来的用户 id 修改用户状态，即删除用户（此处并未在表中彻底删除，而是打上删除标记，以便后期查找）

图书表：（BookDao）

- public Library **findById**(int id): //通过图书 id 返回对应的图书信息
- public ArrayList<Library> **check**(Library library,int page): //通过 library 对象（library 内存储搜索条件），返回匹配的图书信息列表，每次查找最多返回 20 条，使用 page 进行定位数据
- public boolean **add**(Library library): //通过传过来的 library 对象添加图书
- public boolean **modify**(Library library): //通过传过来的 library 对象修改图书
- public boolean **borrow**(int id,String status): //通过图书 id 修改图书状态，即借阅图书
- public boolean **back**(int id,String status): //通过图书 id 修改图书状态，即归还图书
- public boolean **delete**(int id): //通过传过来的用户 id 修改图书状态，即删除图书（此处并未在表中彻底删除，而是打上删除标记，以便后期查找）

借阅记录表：（RecordDao）

- public ArrayList<BookRecord> **check**(BookRecord bookRecord): //通过 bookRecord 对象（bookRecord 内存储搜索条件），返回匹配的借书记录信息列表
- public boolean **borrow**(int uid,int bid): //通过图书 id 和用户 id 进行新增一条记录，即借书
- public boolean **back**(int id): //通过借阅记录 id 修改对应的还书时间，即还书
- public ArrayList<BookRecord> **findNoBackBook**(int uid,int bid,int page): //查询已借该书但还未归还图书的记录信息列表
- public ArrayList<UserRecord> **userRecord**(UserRecord userRecord,int page): //用户借阅信息查询，专用查询用户借阅信息，此处使用了关联查询，userRecord 也是该方法的专有类对象

4.2 基本功能

4.2.1 首页

包含图片和标题



图 4-1 用户登录系统首页图

4.2.2 登陆

包含图片以及登陆表单（用户名、密码输入框和确认、清除按钮）

登陆时若用户名或密码输入框为空则提示“请填写完整登陆信息！”。

若查询该用户名不存在，则提示“用户不存在！”，然后在判断密码是否正确。
若查到该用户状态为 0 或 2，则提示“该用户已被删除！”或“该用户已被禁用！”。
以上都正常时，最后根据查到的用户类型，弹出对应的用户窗口。（1：管理员、0：普通用户）

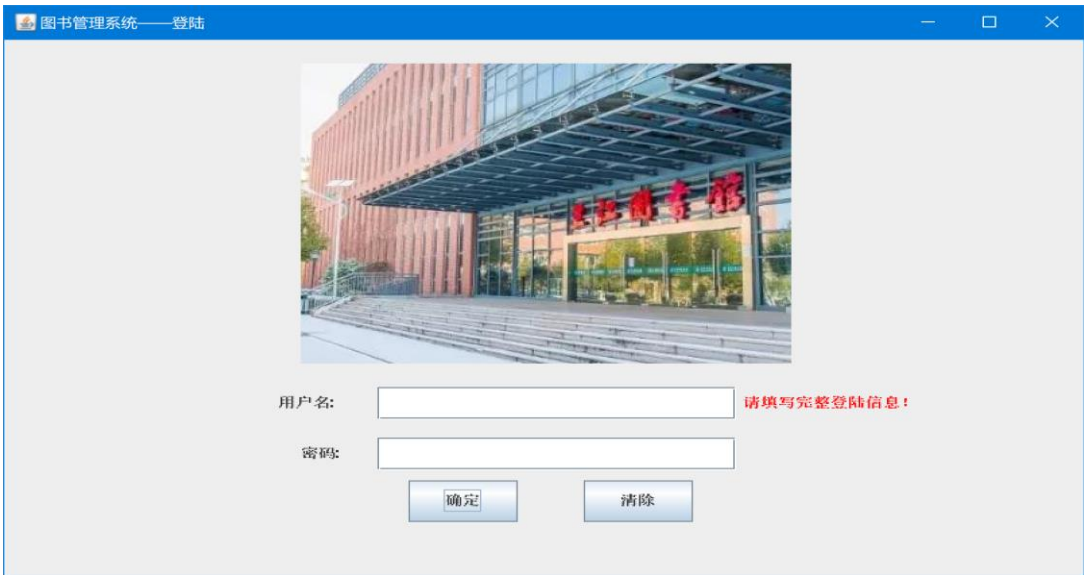


图 4-2 用户登录界面图

如下为用户登陆的主要代码：

```

if (name.equals("")||password.equals("")) {
    label.setText("请填写完整登陆信息！");
}else {
    System.out.println(name);
    UserDao userDao = new UserDao();
    User user = userDao.login(name);
    password = new MD5().MD5(password);
    String username = user.getUsername();
    int id = user.getId();
    String type = user.getType();
    System.out.println(user);
    System.out.println(password);
    if (user.getId()==0) {
        label.setText("用户不存在！");
    }else if(user.getUsername().equals(name)&&user.getPassword().equals(password)){
        System.out.println(type);
        if (user.getStatus().equals("1")) {
            if (type.equals("1")) {
                System.out.println("管理员登陆成功！");
                AdminHome adminHome = new AdminHome(new User(id,username,type));
                adminHome.show();
            }else if (type.equals("0")) {
                System.out.println("普通用户登陆成功！");
                UserHome userHome = new UserHome(new User(id,username,type));
                userHome.show();
            }
        }else if (user.getStatus().equals("0")) { //0为删除、1为可用、2为禁用
            label.setText("该用户已被删除！");
        } else {
            label.setText("该用户已被禁用！");
        }
        this.setVisible(false);
    }else {
        label.setText("密码错误！");
    }
}
t1.setText("");
t2.setText("");

```

4.2.3 密码修改

包含用户名、用户原密码、新密码、新密码确认的输入框和确认、取消按钮。

当点击确认按钮后进行如下判断：

- 1、输入框是否都填写完整；
- 2、新密码与新密码确认密码是否相同；
- 3、原密码是否正确。

若以上都无误，则提示密码修改成功，同时跳转到首页。否则提示相关错误。



图 4-3 用户密码修改界面图

如下为用户密码修改的主要代码：

```
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String old_pwd = textField.getText();
        String new_pwd = textField_1.getText();
        String confirm = textField_2.getText();
        if (old_pwd.equals("") || new_pwd.equals("") || confirm.equals("")) {
            label.setText("请完整填写相关信息！");
        } else if (!new_pwd.equals(confirm)) {
            label.setText("新密码与确认密码不相同！");
        } else {
            UserDao userDao = new UserDao();
            User u = userDao.login(user.getUsername());
            old_pwd = new MD5().MD5(old_pwd);
            new_pwd = new MD5().MD5(new_pwd);
            //数据库操作
            if (!old_pwd.equals(u.getPassword())) {
                label.setText("原密码错误！");
            } else {
                boolean rs = userDao.password(user.getId(), new_pwd);
                if (rs) {
                    JOptionPane.showMessageDialog(null, "密码修改成功！", "提示框", JOptionPane.INFORMATION_MESSAGE);

                    contentPane.removeAll();
                    Init m = new Init();
                    JPanel panel = m.view();
                    panel.setBounds(0, 0, 817, 516);
                    panel.setLayout(null);
                    contentPane.add(panel);
                    contentPane.repaint();

                } else {
                    label.setText("密码修改失败，请重试！");
                }
            }
        }
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
    }
});
```

4.3 用户管理功能

4.3.1 添加用户

弹出用户添加窗口。

添加用户要填的信息包括（用户名、密码、院系、用户类型（0 为管理员、1 为普通用户）、当前借阅数、账号状态（0 为删除、1 为可用、2 为禁用））

点击保存按钮，首先判断信息是否都填写完整，不完整则提示。保存成功则提示“用户录入成功！”。

点击取消按钮可直接关闭该窗口。



图 4-4 管理员添加用户图

如下为管理员添加用户的主要代码：

```
final JButton btnNewButton = new JButton("保存");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String username_val = username.getText();
        String password_val = password.getText();
        String type_val = type.getText();
        String num_val = num.getText();
        String department_val = department.getText();
        String state_val = state.getText();
        if (username_val.equals("") || password_val.equals("") || type_val.equals("") ||
            num_val.equals("") || num_val.equals("") || department_val.equals("") || state_val.equals("")) {
            label_5.setText("请完整填写相关信息!");
        } else {
            //数据库操作
            UserDao userDao = new UserDao();
            User user = new User();
            user.setUsername(username_val);
            user.setPassword(new MD5().MD5(password_val));
            user.setType(type_val);
            user.setNum(num_val);
            user.setDepartment(department_val);
            user.setStatus(state_val);
            boolean rs = userDao.add(user);
            if (rs) {
                JOptionPane.showMessageDialog(null, "用户录入成功!", "提示框", JOptionPane.INFORMATION_MESSAGE);
                dispose();
            } else {
                label_5.setText("信息录入失败, 请重试!");
            }
        }
    }
});
btnNewButton.setFont(new Font("宋体", Font.PLAIN, 24));
btnNewButton.setBounds(219, 360, 110, 35);
contentPane.add(btnNewButton);
```

4.3.2 查询用户

弹出用户查询窗口。

最上方为查询表单（包含用户 ID、用户名、院系、用户类型的多种查询组合方式），当所有查询条件都为空时，则为查询所有用户。

点击查询按钮后，会在下方显示用户信息（序号、用户 ID、用户名、用户类型、当前借阅数、院系、账号状态）以及对应的修改和删除按钮。

同样默认查询结果每页最多只显示 20 条记录，超过 20 条可使用底部按钮进行翻页。

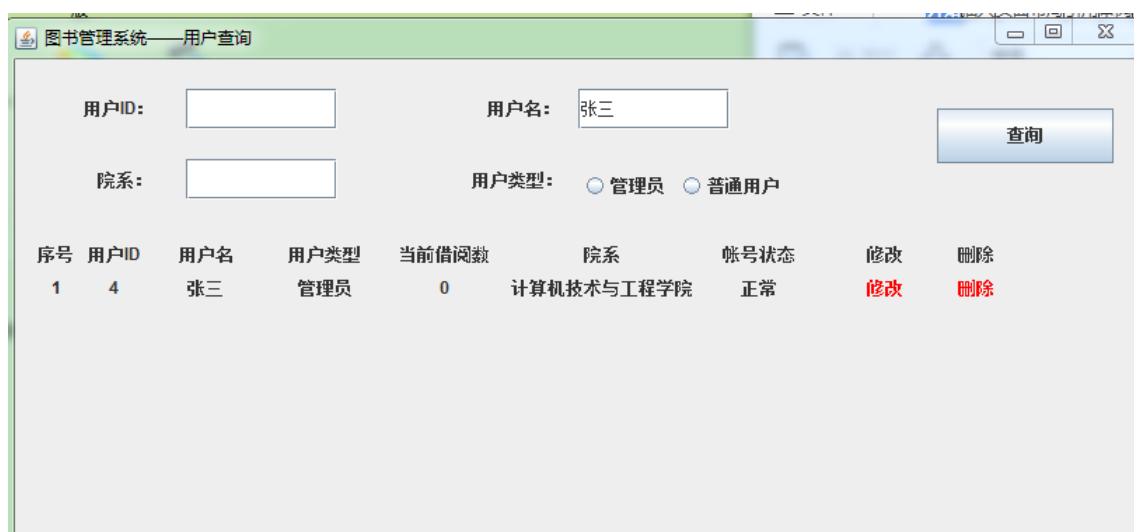


图 4-5 管理员查询用户图

如下为管理员查询用户的主要代码：

```
UserDao userDao = new UserDao();

ArrayList<User> list = userDao.check(searchUser, page);

JLabel my_label = new JLabel();
GridBagConstraints my_gbc_label = new GridBagConstraints();
int j = 1;
for (User u : list) {
    for (int i = 0; i <= 8; i++) {

        System.out.println(list.get(i));
        if (i==0) { //序号
            my_label = new JLabel(String.valueOf(j));
        } else if (i==1) { //UserId
            my_label = new JLabel(String.valueOf(u.getId()));
        } else if (i==2) { //用户名
            my_label = new JLabel(u.getUsername());
        } else if (i==3) { //用户类型
            String type = u.getType().equals("1")?"管理员":u.getType().equals("0")?"普通用户":u.getType();
            my_label = new JLabel(type);
        } else if (i==4) { //当前借阅数
            my_label = new JLabel(u.getNum());
        } else if (i==5) { //院系
            my_label = new JLabel(u.getDepartment());
        } else if (i==6) { //帐号状态 0为删除、1为可用、2为禁用
            String status = u.getStatus().equals("1")?"正常":u.getStatus().equals("0")?"已删除":"禁用";
            my_label = new JLabel(status);
        } else if (i==7) { //修改
            my_label = new JLabel("修改");
        }
```


4.3.3 修改用户

调出用户修改窗口，若要修改的用户在数据库中找到，则提示“用户信息读取失败！”。

用户修改窗口：

显示该用户在数据库中的所有基本信息（用户名、密码、院系、用户类型（0 为管理员、1 为普通用户）、当前借阅数、账号状态（0 为删除、1 为可用、2 为禁用）），所有信息都可修改，除数据库中的用户唯一标识 id 外。

单击保存时，会检查所有输入框都有值，若有空，则提示“请完整填写相关信息！”。保存成功后，提示成功信息，并自动关闭窗口。点击取消按钮也可关闭窗口。



图 4-6 管理员修改用户图

如下为管理员修改用户的主要代码：

```
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String username_val = username.getText();
        String password_val = password.getText();
        String type_val = type.getText();
        String num_val = num.getText();
        String department_val = department.getText();
        String state_val = state.getText();
        if (username_val.equals("") || type_val.equals("") ||
            num_val.equals("") || num_val.equals("") || department_val.equals("") || state_val.equals("")) {
            label_5.setText("请完整填写相关信息！");
        } else {
            //数据库操作
            User u = new User();
            u.setId(uid);
            u.setUsername(username_val);
            if (password_val.equals("")) {
                u.setPassword(user.getPassword());
            } else {
                u.setPassword(new MD5().MD5(password_val));
            }
            u.setType(type_val);
            u.setNum(num_val);
            u.setDepartment(department_val);
            u.setStatus(state_val);
            boolean rs = userDao.modify(u);
            if (rs) {
                JOptionPane.showMessageDialog(null, "信息修改成功！", "提示框", JOptionPane.INFORMATION_MESSAGE);
                dispose();
            } else {
                label_5.setText("信息修改失败，请重试！");
            }
        }
    }
});
```

4.3.4 删除用户

删除条件：

- 1、检查该用户不存在外借书籍
- 2、删除前进行删除确认

若该用户存在外借图书且还未归还，则不允许删除，并提示“该用户存在外借书籍，暂不允许删除！”。若允许删除则弹出删除确认框。



图 4-7 管理员删除用户图

如下为管理员删除用户的主要代码：

```
my_label = new JLabel("删除"); //可借、不可借
my_label.setForeground(new Color(255, 0, 0));
my_label.setToolTipText(String.valueOf(u.getId()));

my_label.addMouseListener(new MouseAdapter() {

    @Override
    public void mouseClicked(MouseEvent arg0) {
        /*
         * 删除用户流程
         * 删除条件：
         * 1、检查该用户不存在外借书籍
         * 2、删除前进行删除确认
         */
        if (!u.getNum().equals("0")) {
            JOptionPane.showMessageDialog(null, "该用户存在外借书籍，暂不允许删除！", "提示框", JOptionPane.INFORMATION_MESSAGE);
        } else {
            int i = JOptionPane.showConfirmDialog(null, "确定删除该用户吗？", "提示框", JOptionPane.YES_NO_OPTION);
            if (i == 0) {
                System.out.println("删除操作");
                boolean rs = userDao.delete(u.getId());
                if (rs) {
                    JOptionPane.showMessageDialog(null, "用户删除成功！", "提示框", JOptionPane.INFORMATION_MESSAGE);
                } else {
                    JOptionPane.showMessageDialog(null, "用户删除失败，请重试！", "提示框", JOptionPane.INFORMATION_MESSAGE);
                }
                check(); //重载界面
                contentPane.updateUI();
            }
            System.out.println(i); // -1: 为叉掉窗口 1: 为否 0: 为是
        }
    }
});
```

4.4 图书管理功能

4.4.1 添加图书

弹出图书录入窗口。

添加图书要填的信息包括（书名、作者、出版社、出版日期、库存、ISBN、馆藏地、借阅状态（默认为可借））

点击保存按钮，首先判断信息是否都填写完整，不完整则提示。保存成功则提示“图书录入成功！”。

点击取消按钮可直接关闭该窗口。

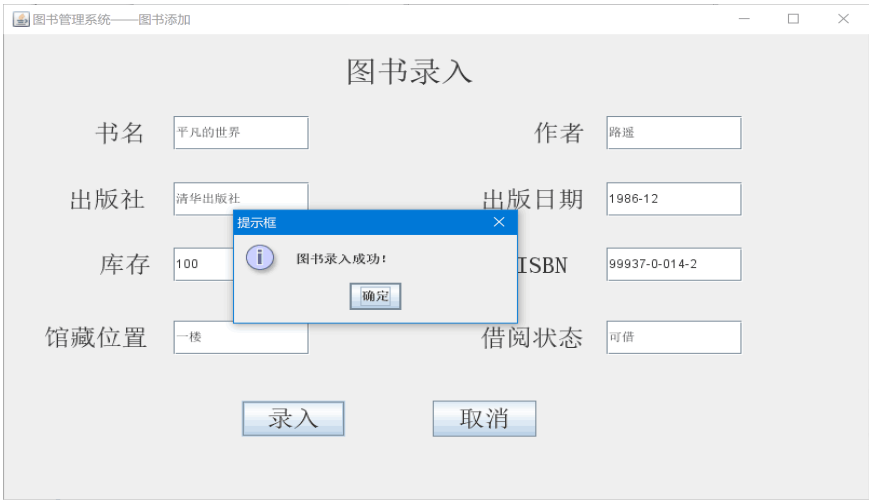


图 4-8 图书录入图

如下为图书录入的主要代码：

```
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name_val = name.getText();
        String author_val = author.getText();
        String press_val = press.getText();
        String publishDate_val = publishDate.getText();
        String reserve_val = reserve.getText();
        String isbn_val = isbn.getText();
        String location_val = location.getText();
        String status_val = status.getText();
        if (name_val.equals("") || author_val.equals("") || press_val.equals("") ||
            publishDate_val.equals("") || reserve_val.equals("") || isbn_val.equals("") ||
            location_val.equals("") || status_val.equals("")) {
            label_7.setText("请完整填写相关信息!");
        } else {
            //数据库操作
            BookDao bookDao = new BookDao();
            Library library = new Library();
            library.setName(name_val);
            library.setAuthor(author_val);
            library.setPress(press_val);
            library.setPublishDate(publishDate_val);
            library.setReserve(reserve_val);
            library.setIsbn(isbn_val);
            library.setLocation(location_val);
            library.setStatus(status_val);
            boolean rs = bookDao.add(library);
            if (rs) {
                JOptionPane.showMessageDialog(null, "图书录入成功!", "提示框", JOptionPane.INFORMATION_MESSAGE);
                dispose();
            } else {
                label_7.setText("信息录入失败, 请重试!");
            }
        }
    }
});
```

4. 4. 2 图书查询

包含图片和标题以及搜索条件选择和输入框、检索按钮。

搜索条件包含：全部、书名、作者、出版社。

默认为全部，此时输入框不能使用。

当选择其他条件时，搜索框可用，且必须有内容才允许检索，否则提示“请输入搜索内容！”。

搜索条件：书名、作者、出版社，均为模糊搜索。

点击检索按钮，弹出搜索结果窗口。



图 4-9 图书查询图

图书搜索结果窗口：（对于未登录、普通用户、管理员的不同用户类型，显示有所不同。）

图书信息（序号、书名、作者、出版社、出版日期、库存、ISBN、馆藏地、借阅状态）、每页最多只显示 20 条记录，超过 20 条可使用底部按钮进行翻页。

- 未登录状态：仅包含相关图书信息。
- 普通用户：除基本信息外还包含“借阅”按钮。
- 管理员：除基本信息外还包含“借阅”、“归还”、“修改”和“删除”按钮。

入下为图书查询的主要代码：

```
button.addActionListener(new ActionListener() { //添加检索监听事件
    public void actionPerformed(ActionEvent arg0) {
        String label = (String) comboBox.getSelectedItemAt(0); //获取下拉框的值
        System.out.println(label);
        String value = textField.getText(); //获取文本框的值
        if (value.equals("") && label.equals("全部")) { //当判断为输入检索内容时，显示提示内容，设置提示框
            label_1.setText("请输入搜索内容!");
        } else { //当有内容输入时，弹出搜索结果页面
            BookCheck bookCheck = new BookCheck(user, label, value); //实例化BookCheck同时传入相关参数
            bookCheck.show(); //弹出实例化的窗口
        }
    }
});

//数据库操作，获取书目信息
BookDao bookDao = new BookDao();
UserDao userDao = new UserDao();
RecordDao recordDao = new RecordDao();

Library library = new Library();
if (label.equals("书名")) {
    library.setName(search);
} else if (label.equals("作者")) {
    library.setAuthor(search);
} else if (label.equals("出版社")) {
    library.setPress(search);
}
ArrayList<Library> list = bookDao.check(library, page);

JLabel my_label = new JLabel();
GridBagConstraints my_gbc_label = new GridBagConstraints();
int j = 1;
for (Library book : list) {
    for (int i = 0; i < 13; i++) {
        System.out.println(list.get(i));
        if (i==0) { //序号
            my_label = new JLabel(String.valueOf(j));
        } else if (i==1) { //书名
            my_label = new JLabel(book.getName());
        } else if (i==2) { //作者
            my_label = new JLabel(book.getAuthor());
        } else if (i==3) { //出版社
            my_label = new JLabel(book.getPress());
        } else if (i==4) { //出版日期
            my_label = new JLabel(book.getPublishDate());
        } else if (i==5) { //库存
            my_label = new JLabel(book.getReserve());
        } else if (i==6) { //ISBN
            my_label = new JLabel(book.getIsbn());
        } else if (i==7) { //馆藏地
            my_label = new JLabel(book.getLocation());
        } else if (i==8) { //图书状态
```

4.4.3 修改图书

调出图书修改窗口，若要修改的图书在数据库中找到，则提示“图书信息读取失败！”。

图书修改窗口

显示该图书在数据库中的所有基本信息（书名、作者、出版社、出版日期、库存、ISBN、馆藏位置、借阅状态），所有信息都可修改，除数据库中的图书唯一标识 id 外。

单击保存时，会检查所有输入框都有值，若有空，则提示“请完整填写相关信息！”。保存成功后，提示成功信息，并自动关闭窗口。点击取消按钮也可关闭窗口。



图 4-12 管理员修改图书图

如下为管理员修改图书的主要代码：

```
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name_val = name.getText();
        String author_val = author.getText();
        String press_val = press.getText();
        String publishDate_val = publishDate.getText();
        String reserve_val = reserve.getText();
        String isbn_val = isbn.getText();
        String location_val = location.getText();
        String status_val = status.getText();
        if (name_val.equals("") || author_val.equals("") || press_val.equals("") ||
            publishDate_val.equals("") || reserve_val.equals("") || isbn_val.equals("") ||
            location_val.equals("") || status_val.equals("")) {
            label_7.setText("请完整填写相关信息！");
        } else {
            //数据库操作
            Library library = new Library();
            library.setId(bid);
            library.setName(name_val);
            library.setAuthor(author_val);
            library.setPress(press_val);
            library.setPublishDate(publishDate_val);
            library.setReserve(reserve_val);
            library.setIsbn(isbn_val);
            library.setLocation(location_val);
            library.setStatus(status_val);
            boolean rs = bookDao.modify(library);
            if (rs) {
                JOptionPane.showMessageDialog(null, "图书修改成功!", "提示框", JOptionPane.INFORMATION_MESSAGE);
                dispose();
            } else {
                label_7.setText("信息修改失败, 请重试!");
            }
        }
    }
});
```

4. 4. 4 删除图书

删除条件：

- 1. 检查该书在借阅记录中不存在未归还的图书
- 2. 删除前进行删除确认

若该书存在外借且还未归还，则不允许删除，并提示“该书存在外借书籍，暂不允许删除！”。若允许删除则弹出删除确认框。



图 4-13 管理员删除图书图

如下为管理员删除图书的主要代码：

```
my_label.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        /*
         * 删除图书流程
         * 删除条件：
         * 1、检查该书在借阅记录中不存在未归还的图书
         * 2、删除前进行删除确认
         */
        ArrayList<BookRecord> bList = recordDao.findNoBackBook(0,book.getId(),1);
        if (bList.size()!=0) {
            JOptionPane.showMessageDialog(null,"该书存在外借书籍，暂不允许删除！", "提示框",JOptionPane.INFORMATION_MESSAGE);
        }else {
            int i=JOptionPane.showConfirmDialog(null, "确定删除该图书吗？", "提示框", JOptionPane.YES_NO_OPTION);
            if (i==0) {
                System.out.println("删除操作");
                boolean rs = bookDao.delete(book.getId());
                if (rs) {
                    JOptionPane.showMessageDialog(null,"图书删除成功!", "提示框",JOptionPane.INFORMATION_MESSAGE);
                }else {
                    JOptionPane.showMessageDialog(null,"图书删除失败，请重试!", "提示框",JOptionPane.INFORMATION_MESSAGE);
                }
                check();//重载界面
                contentPane.updateUI();
            }
            System.out.println(i); //-1: 为叉掉窗口 1: 为否 0: 为是
        }
    }
});
```

4.5 借还书功能

4.5.1 借阅图书

借阅条件:

 查询用户状态, 若已被禁用, 则提示禁止借阅

 查询该书库存是否大于等于 1, 且状态为可借

借阅过程:

 将该书库存减一, 该用户借阅数量加一

 同时判断库存是否还有剩余, 若未有剩余, 则修改图书状态为无库存

 完成借阅后生成借阅记录

 所有操作无误, 则提示借阅完成, 否则提示相关错误。



图 4-10 用户借阅图书图 (相应的图书库存减一)

如下为用户借阅图书的主要代码:

```
User u = userDao.findById(user.getId());
Library b = bookDao.findById(book.getId());
if (u.getStatus().equals("0")) {
    JOptionPane.showMessageDialog(null, "该账户已被禁用, 禁止借阅!", "提示框", JOptionPane.INFORMATION_MESSAGE);
} else if (Integer.parseInt(b.getReserve()) <= 0) {
    JOptionPane.showMessageDialog(null, "该书库存不足!", "提示框", JOptionPane.INFORMATION_MESSAGE);
} else if (!b.getStatus().equals("可借")) {
    JOptionPane.showMessageDialog(null, "该书不可借!", "提示框", JOptionPane.INFORMATION_MESSAGE);
} else {
    boolean rs = false;
    if (b.getReserve().equals("1")) {
        rs = bookDao.borrow(book.getId(), "库存不足");
    } else {
        rs = bookDao.borrow(book.getId(), "可借");
    }
    if (rs) {
        boolean rs1 = recordDao.borrow(user.getId(), book.getId()); // 生成借阅记录
        boolean rs2 = userDao.addNumById(user.getId()); // 增加借阅数量
        if (rs1 && rs2) {
            JOptionPane.showMessageDialog(null, "借阅成功!", "提示框", JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(null, "生成相关借阅记录失败!", "提示框", JOptionPane.INFORMATION_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(null, "借阅失败!", "提示框", JOptionPane.INFORMATION_MESSAGE);
    }
    check(); // 重载界面
    contentPane.repaint();
    contentPane.updateUI();
}
```


4.5.2 归还图书

调出还书窗口，选择具体还书对象。

调出窗口时若检测该书无外借书籍，则提示“该书无外借记录！”。若有外借书籍，则显示借阅者信息。然后管理员对具体用户进行归还操作。

归还后，将该书库存加一，图书状态为可借，该用户借阅数量减一，同时记录归还时间。



图 4-11 用户归还图书图（相应的图书库存加一）

还书窗口：

顶上包含当前的图书信息（书名、出版社、作者、ISBN、库存）

下方为所有借阅该书且未归还的用户信息（序号、用户 ID、用户名、用户类型、当前借阅数、院系、帐号状态、借阅时间）

同样每页最多只显示 20 条记录，超过 20 条可使用底部按钮进行翻页。

如下为用户还书的主要代码：

```

}else if (i==8) { //归还图书
    my_label = new JLabel("归还");
    my_label.setForeground(new Color(255, 0, 0));
    my_label.setToolTipText(String.valueOf(bookRecord.getId()));
    my_label.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
            //归还图书
            boolean rs = bookDao.back(book.getId(), "可借");
            book.setReserve(String.valueOf(Integer.parseInt(book.getReserve())+1));
            boolean rs1 = recordDao.back(bookRecord.getId());
            boolean rs2 = userDao.cutNumById(user.getId());
            if (rs) {
                if (rs1&&rs2) {
                    JOptionPane.showMessageDialog(null,"还书操作成功!", "提示框",JOptionPane.INFORMATION_MESSAGE);
                }else {
                    JOptionPane.showMessageDialog(null,"生成相关还书记录失败!", "提示框",JOptionPane.INFORMATION_MESSAGE);
                }
            }else {
                JOptionPane.showMessageDialog(null,"还书操作失败, 请重试!", "提示框",JOptionPane.INFORMATION_MESSAGE);
            }
            check();//重载界面
            contentPane.updateUI();
        }
    });
}
my_gbc_label = new GridBagConstraints();
my_gbc_label.insets = new Insets(0, 0, 5, 5);
my_gbc_label.gridx = i;
my_gbc_label.gridy = j;
contentPane.add(my_label, my_gbc_label);
}
```


4.6 借阅信息查询

（分普通用户和管理员）

普通用户：

直接弹出借阅信息查询窗口，显示当前用户的借阅信息记录（包含序号、用户名、用户类型、院系、图书名、作者、出版日期、借阅日期、还书日期）

同样默认查询结果每页最多只显示 20 条记录，超过 20 条可使用底部按钮进行翻页。

序号	用户名	用户类型	院系	图书名	作者	出版日期	借阅日期	还书日期
1	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-22 11:18:24	2020-6-22 11:23:42
2	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:18	2020-6-20 22:56:44
3	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:13	
4	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:10	2020-6-20 22:57:26
5	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:09	2020-6-20 22:56:28
6	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:06	2020-6-20 22:57:24
7	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:05	2020-6-20 22:56:26
8	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:02	2020-6-20 22:57:22
9	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:00	2020-6-20 22:56:23
10	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:27:23	2020-6-20 22:57:20

图 4-14 普通用户借阅信息查询图

管理员：

首先显示借阅信息查询界面，可在此界面指定搜索条件（全部、用户名、图书名、用户类型、院系、作者）

默认为全部，此时输入框不能使用。

当选择其他条件时，搜索框可用，且必须有内容才允许检索，否则提示“请输入搜索内容！”。

搜索条件：书名、作者，为模糊搜索。

点击检索按钮，弹出借阅信息查询结果窗口，显示条目与普通用户相同。

序号	用户名	用户类型	院系	图书名	作者	出版日期	借阅日期	还书日期
1	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-22 11:18:24	2020-6-22 11:23:42
2	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:57:40	2020-6-20 22:57:48
3	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:18	2020-6-20 22:56:44
4	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:13	
5	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:10	2020-6-20 22:57:26
6	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:09	2020-6-20 22:56:28
7	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:06	2020-6-20 22:57:24
8	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:05	2020-6-20 22:56:26
9	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:29:02	2020-6-20 22:57:22
10	user	普通用户	计算机科学与工程学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 22:29:00	2020-6-20 22:56:23
11	user	普通用户	计算机科学与工程学院	百年孤独	[哥伦比亚] 加西亚·马尔克斯	2011-6	2020-6-20 22:27:23	2020-6-20 22:57:20
12	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 20:41:58	
13	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 20:24:50	2020-6-20 20:42:06
14	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:12:14	2020-6-20 19:12:22
15	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:11:49	2020-6-20 19:11:52
16	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:10:51	2020-6-20 19:10:55
17	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:09:21	2020-6-20 19:09:25
18	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:07:51	2020-6-20 19:08:01
19	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:06:28	2020-6-20 19:07:55
20	admin	管理员	计算机学院	时间简史	[英] 史蒂芬·霍金	2010-04-01	2020-6-20 19:04:02	2020-6-20 19:06:33

图 4-15 管理员借阅信息查询图

如下为借阅信息查询主要代码：

```
//数据库操作，获取书目信息
RecordDao recordDao = new RecordDao();
UserRecord userRecord = new UserRecord();
if (!user.getType().equals("1")) {
    userRecord.setUsername(user.getUsername());
} else {
    if (label=="用户名") {
        userRecord.setUsername(search);
    }else if (label=="用户类型") {
        String type = search.equals("管理员")?"1":search.equals("普通用户")?"0":search;
        userRecord.setType(type);
    }else if (label=="院系") {
        userRecord.setDepartment(search);
    }else if (label=="图书名") {
        userRecord.setName(search);
    }else if (label=="作者") {
        userRecord.setAuthor(search);
    }
}
ArrayList<UserRecord> list = recordDao.userRecord(userRecord, page);
JLabel my_label = new JLabel();
GridBagConstraints my_gbc_label = new GridBagConstraints();
int j = 1;
for (UserRecord uRecord : list) {
    for (int i = 0; i < 13; i++) {
        if (i==0) { //序号
            my_label = new JLabel(String.valueOf(j));
        }else if (i==1) { //用户名
            my_label = new JLabel(uRecord.getUsername());
        }else if (i==2) { //用户类型
            String type = uRecord.getType().equals("1")?"管理员":uRecord.getType().equals("0")?"普通用户":"其他用户";
            my_label = new JLabel(type);
        }else if (i==3) { //院系
            my_label = new JLabel(uRecord.getDepartment());
        }else if (i==4) { //图书名
            my_label = new JLabel(uRecord.getName());
        }else if (i==5) { //作者
            my_label = new JLabel(uRecord.getAuthor());
        }else if (i==6) { //出版日期
```

结束语

本次实训内容选题是图书管理系统设计与实现，它是在 eclipse 环境中开发的基于开发 java 开发语言的软件，其主要功能包括：用户登陆注册、用户个人信息管理、图书添加、删除修改等功能管理、以及借还书功能。

本次实训内容课题在设计和完成基本的主要功能时，还有很多不足。其中包括：

（1）系统的界面设计比较基于传统，显得略有些老套；

（2）本系统中的数据处理没有做的很完善，对于数据量较大的处理方法没有实现；

（3）系统中考虑的一些图书管理、用户管理不是很全面。

根据整体的图书管理系统设计与实现，大致的实现了其相关的用户登陆、图书管理和借还书功能，但在此仍有一些不足的地方和没考虑到的问题，其中包括一些还书超时问题没有涉及到，还需要不断改善其功能。对未实现的功能进行各种尝试，争取考虑到关于图书管理系统会出现的所有问题。

附录

各文件作用：（简述）

cn.zero.dao：（用于操作数据库）

BookDao：用于操作图书表

RecordDao：用于操作借阅记录表

UserDao：用于操作用户表

cn.zero.entity：（实体类对象，分别对应相关表）

Library：图书类

User：用户类

BookRecor：借阅记录类

UserRcord：借阅信息查询结果类

cn.zero.util：（工具）

JDBCUtils：数据库连接工具及其配置

MD5：用户密码 MD5 加密工具

cn.zero.panel：（具体展示面板程序）

Index：系统启动窗口

Init：初始化界面

Login：登陆界面

UpdatePassword：修改密码界面

UserHome：普通用户登陆成功窗口

AdminHome：管理员登陆成功窗口

UserAdd：添加用户窗口

UserCheck：查询用户窗口

UserModify：用户修改窗口

BookAdd：录入图书窗口

SearchBook：搜索图书界面

BookCheck：图书搜索结果窗口

BookModify：图书修改窗口

ReturnBook：归还图书操作窗口

SearchRecord：查询借阅记录界面

RecordCheck：借阅信息查询结果窗口

SearchUser：查询用户界面（未使用）