

# 图书管理系统

## 一、 功能描述

| 角色    | 功能编号 | 功能      |
|-------|------|---------|
| 学生    | 1    | 用户登录    |
|       | 2    | 借阅图书    |
|       | 3    | 归还图书    |
|       | 4    | 续借图书    |
|       | 5    | 预约图书    |
|       | 6    | 查看借阅信息  |
|       | 7    | 图书评论    |
|       | 8    | 查看借阅排行  |
|       | 9    | 查看新书排行  |
|       | 10   | 修改用户密码  |
|       | 11   | 浏览图书    |
|       | 12   | 查找图书    |
| 系统管理员 | 1    | 用户登录    |
|       | 2    | 添加管理员   |
|       | 3    | 修改管理员属性 |
|       | 4    | 删除管理员   |
|       | 5    | 修改用户密码  |
| 学生管理员 | 1    | 用户登录    |

|       |   |          |
|-------|---|----------|
|       | 2 | 添加学生信息   |
|       | 3 | 维护指定学生信息 |
|       | 4 | 删除学生     |
|       | 5 | 导入学生     |
|       | 6 | 修改用户密码   |
|       |   |          |
| 图书管理员 | 1 | 用户登录     |
|       | 2 | 维护指定图书信息 |
|       | 3 | 导入图书     |
|       | 4 | 修改用户密码   |

表格 1 系统功能设计表

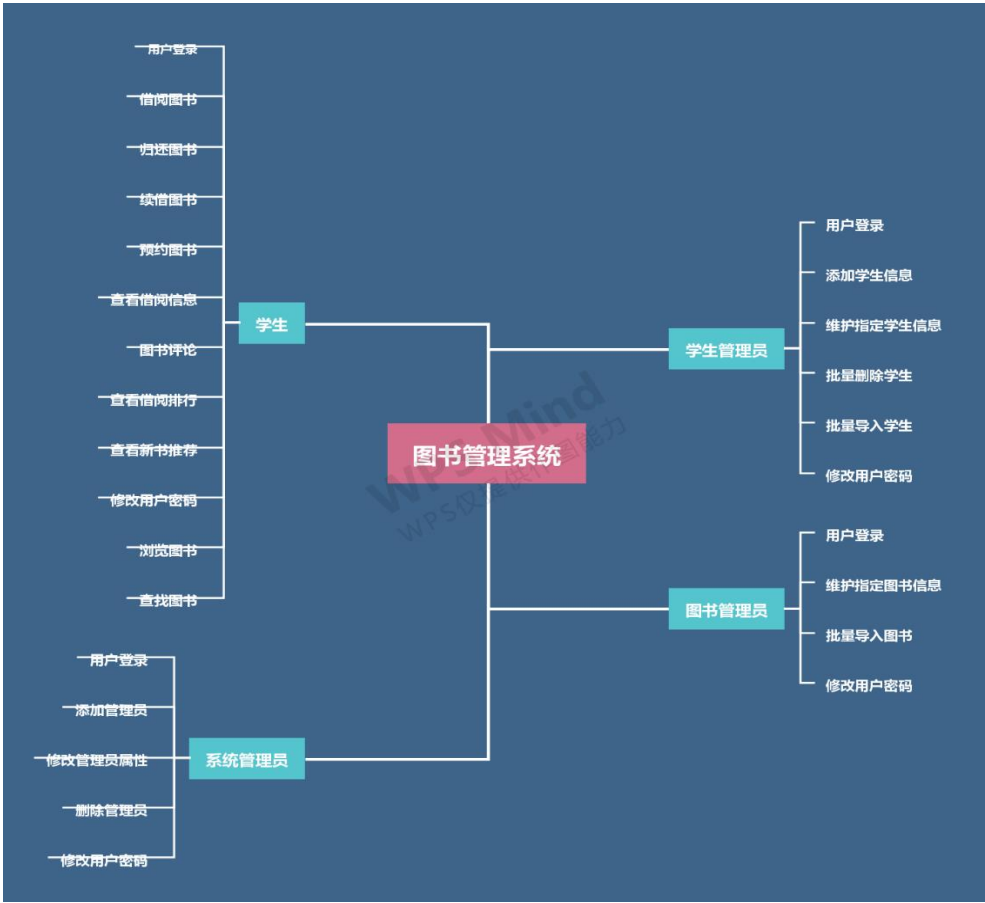


图 1 系统功能设计

## 二、 类的设计

| 类             | 属性                                                                                                                                   | 接口        | 备注    |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------|-----------|-------|
| BmStudent     | no(学号)<br>password(密码)<br>name(姓名)<br>class(班级)<br>major(专业)<br>contact(联系方式)                                                        | to_json() | 学生类   |
| BmManager     | username(用户名)<br>name(姓名)<br>password(密码)<br>role(角色)                                                                                | to_json() | 管理员类  |
| BmBook        | no(编号)<br>title(书名)<br>author(作者)<br>publisher(出版社)<br>p_date(出版日期)<br>inventory(总库存)<br>lend_num(已借出)<br>borrow_num(借阅次数)           | to_json() | 图书类   |
| BmBorrow      | id<br>stu_no(学号)<br>book_no(图书编号)<br>borrow_date(借阅日期)<br>deadline(应还日期)<br>return_date(归还日期)<br>is_return(归还状态)<br>is_renewal(续借状态) | to_json() | 借阅信息类 |
| BmComment     | id<br>stu_no(学号)<br>book_no(图书编号)<br>content(评论内容)<br>score(评分)<br>date(评论日期)                                                        | to_json() | 图书评论类 |
| BmReservation | id<br>stu_no(学号)<br>book_no(图书编号)<br>date(预约日期)                                                                                      | to_json() | 图书预约类 |

表格 2 系统类设计表

### 三、用例图及用例描述

#### (一) 用例图

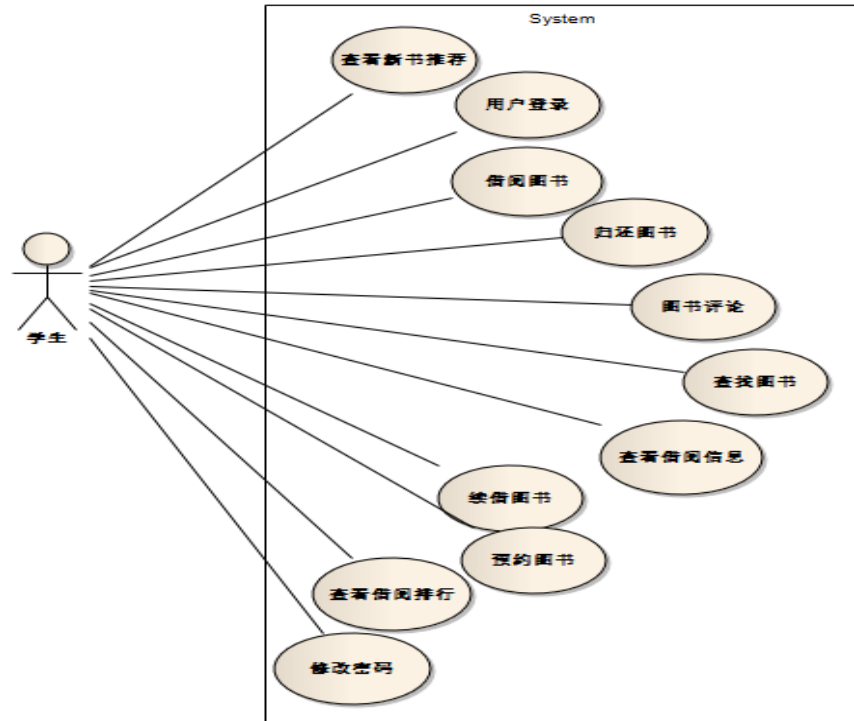


图 2 学生用例图

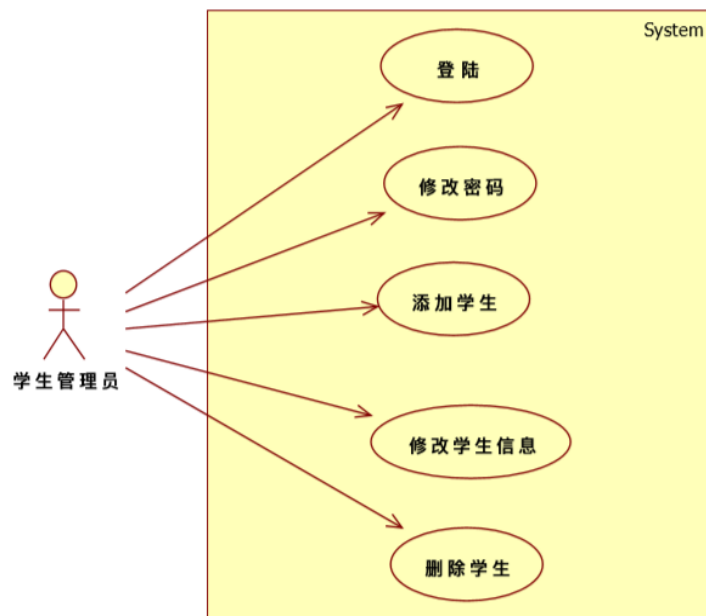


图 3 学生管理员用例图

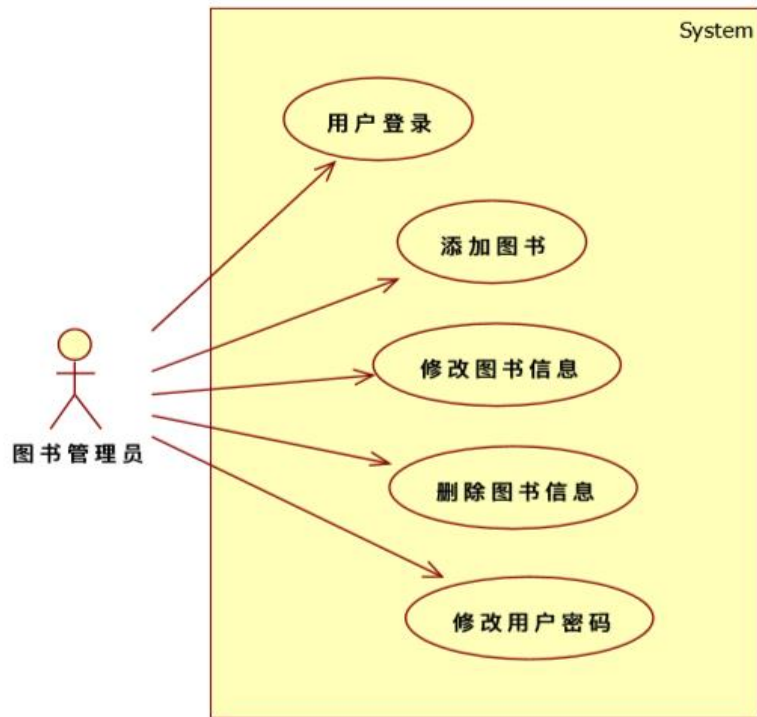


图 4 图书管理员用例图

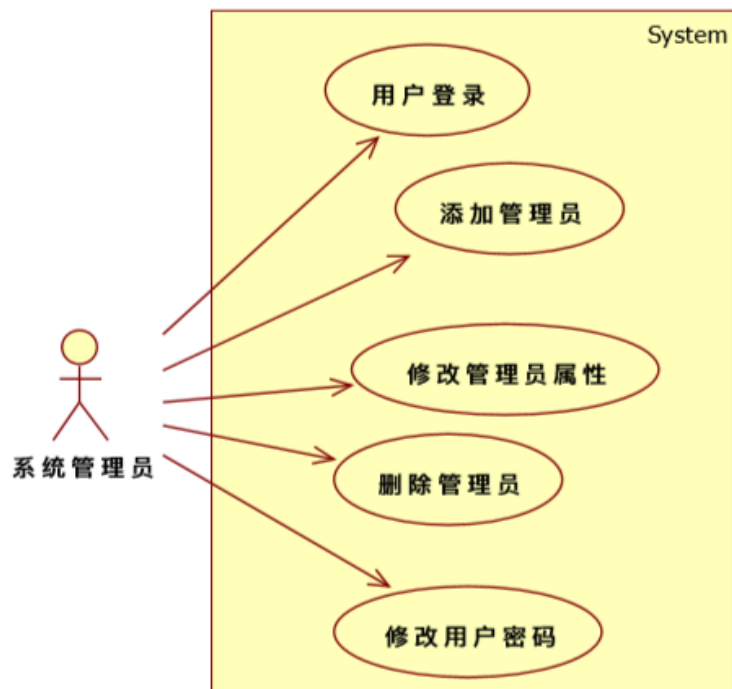


图 5 系统管理员用例图

## （二）用例描述

### 1、 用户登录用例

用例名：登录系统

参与执行者：学生、图书管理员、学生管理员、系统管理员

入口条件：打开图书管理系统

事件流：

• 通常情况：

- 1) 输入账号和密码
- 2) 系统验证账号信息
- 3) 成功登录系统
- 4) 系统发送预约和还书提醒
- 5) 进入学生界面（可以直接看到借阅信息）

• 分支流

- 1.3) 进入图书管理界面
- 2.3) 进入学生管理界面
- 3.3) 进入系统管理员界面

• 例外情况：

- 1) 输入的账号或密码错误、重新登陆

出口条件：成功登陆系统

### 2、 图书维护用例

用例名：维护指定图书

参与执行者：图书管理员

入口条件：打开图书浏览界面

事件流：

• 通常情况：

- 1) 增加图书
- 2) 删除图书
- 3) 修改图书信息
- 4) 查找某种图书

• 例外情况：

- 1) 查找的图书不存在，增加后再查找
- 2) 删除的图书已借出，等归还后再删除
- 3) 修改图书信息，id 不支持修改

出口条件：成功执行某一项分支

## 四、系统类图（静态模型）

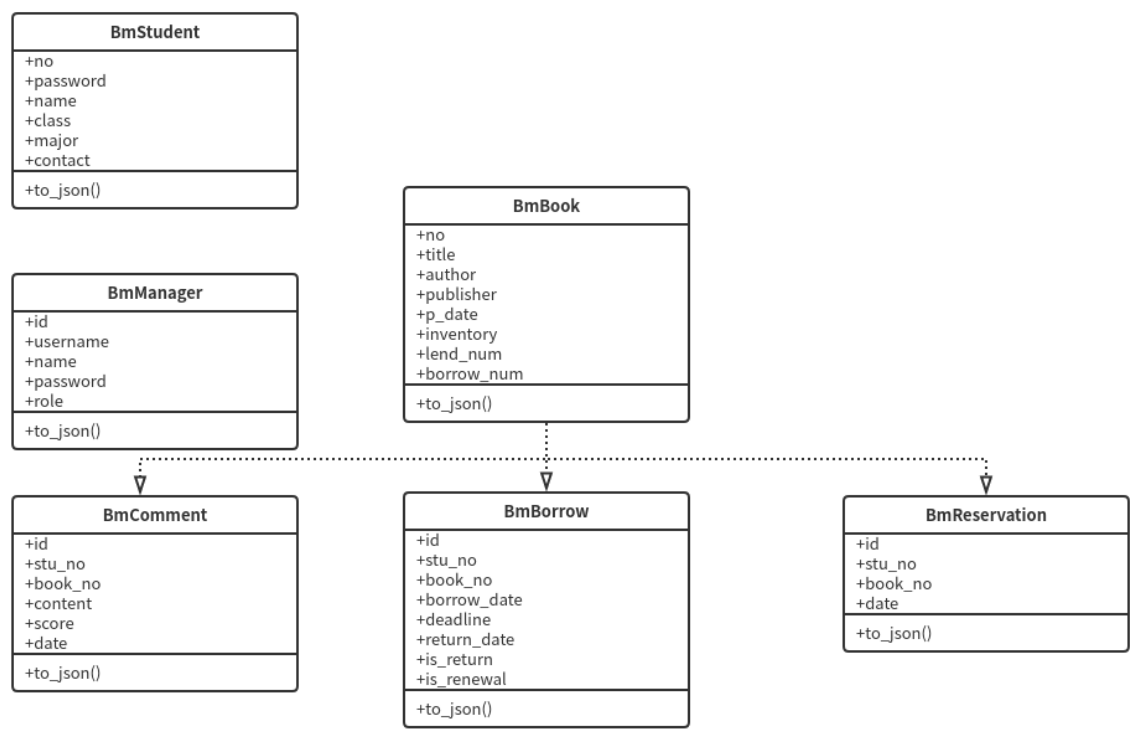


图 6 系统类图

## 五、时序图（动态模型）

1、学生登录，登录时需要对学生信息进行检查

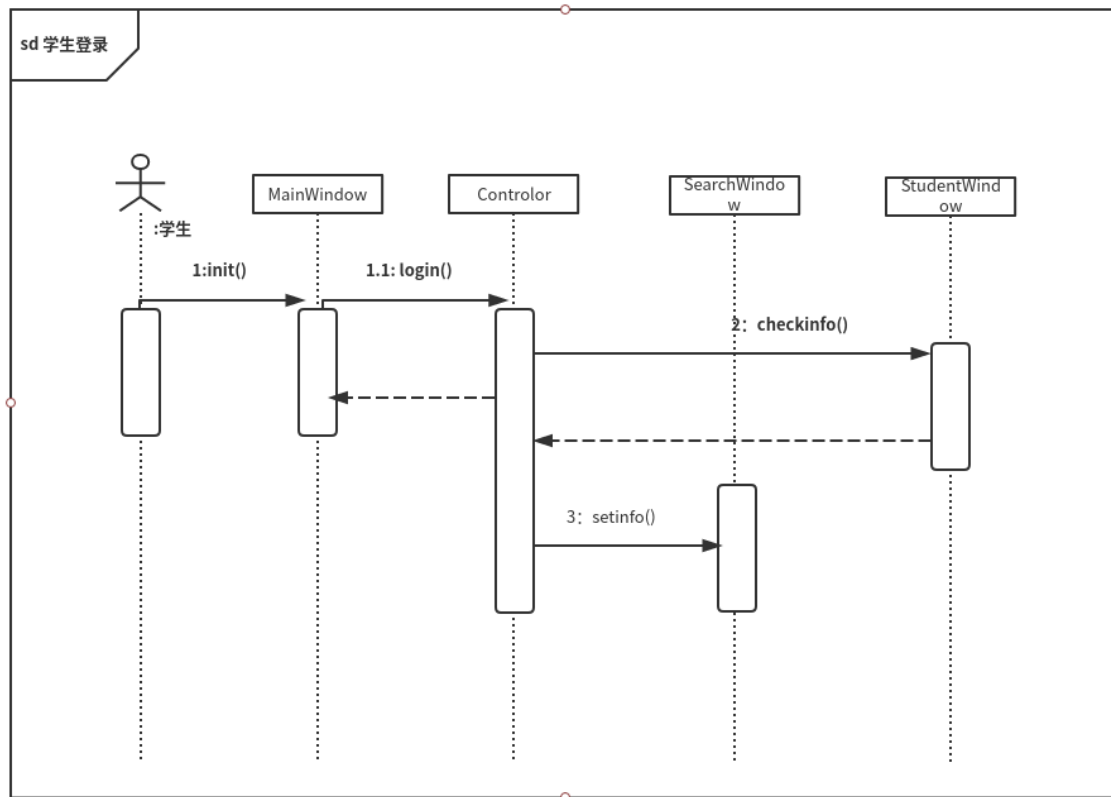


图 7 学生登录时序图

- 2、图书借阅，学生登录后进入搜索界面进行搜索，搜索结果界面需要对当前图书可借阅信息进行检查，若可借阅则进行借阅操作，否则可进行预约操作。

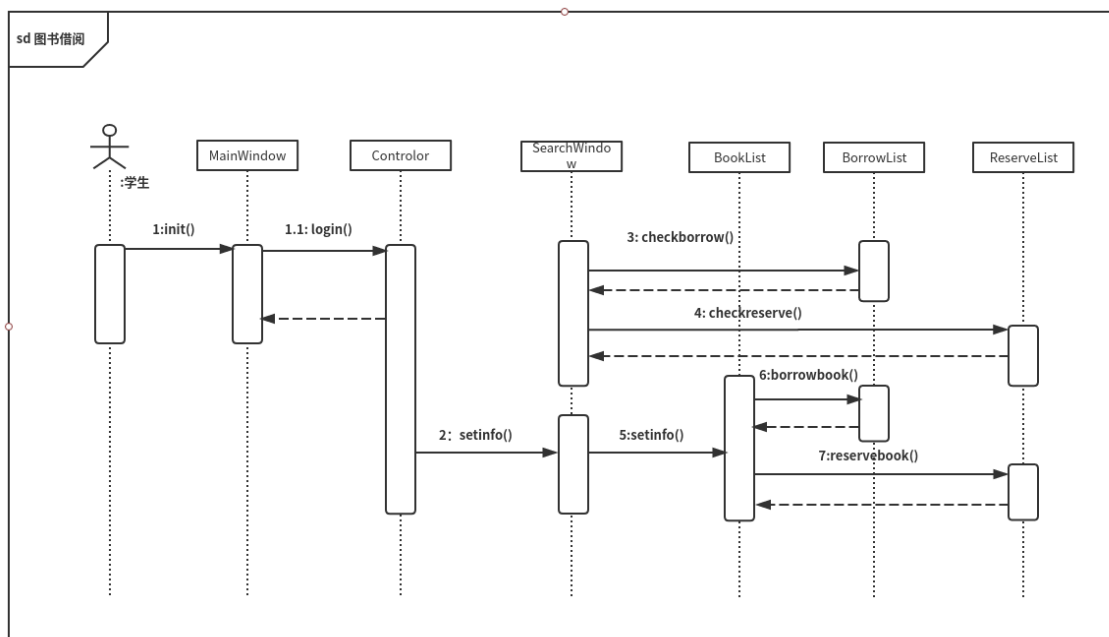


图 8 图书借阅时序图



## 六、系统功能描述

### （一）系统功能树状图

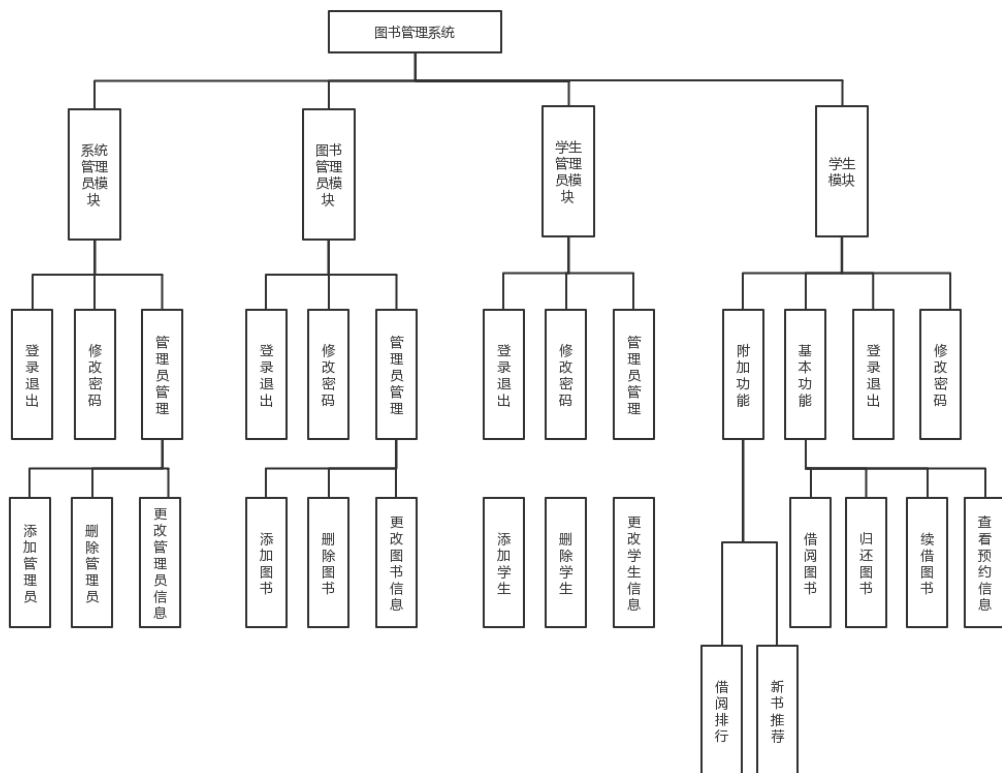


图9 系统功能设计图

### （二）重点算法及实现描述

#### 1. 学生登录模块

登陆成功的条件：1. 存在该学生。2. 学号和密码相匹配。

同时在登录的过程中会检查是否有预约和将要更换的书。如图所示：

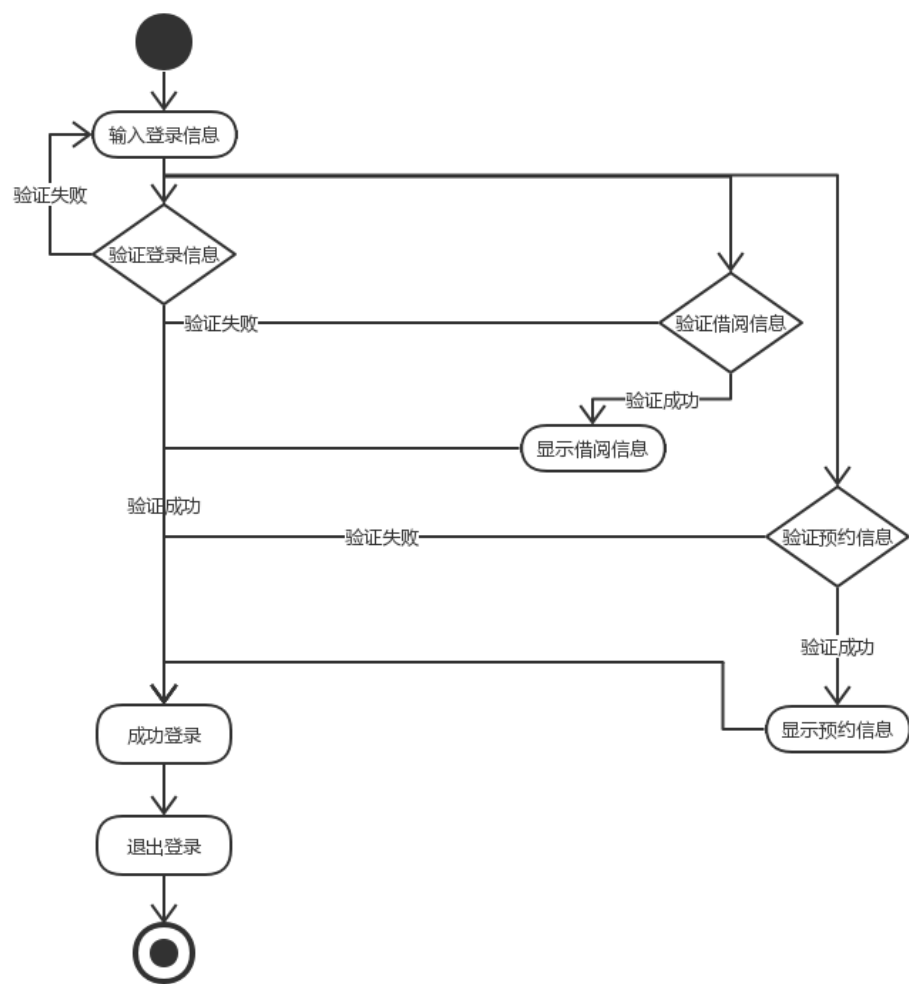


图 10 学生登录流程图

## 七、系统实现描述

### （一）图书查询

```
def search_booklist(request):
    """ 渲染图书查询界面 """
    if request.method == 'GET':
        info = request.GET.get('info')
        return render(request, 'book/booklist.html', {'cur_user':
request.session.get('userinfo')['no'], 'search_info': info})
    if request.method == 'POST':
        option = int(request.POST.get('option', None))
        info = unquote(request.POST.get('info', None))
        page = int(request.POST.get('page', None))
        limit = int(request.POST.get('limit', None))

        if option == 0: # 书名查询
            book_objects = BmBook.objects.filter(title__contains=info)[
                (limit * (page - 1)): (limit * page - 1)]
        elif option == 1: # 作者查询
            book_objects = BmBook.objects.filter(author__contains=info)[
                (limit * (page - 1)): (limit * page - 1)]
        elif option == 2: # 出版社查询
            book_objects = BmBook.objects.filter(publisher__contains=info)[
                (limit * (page - 1)): (limit * page - 1)]
        data = []
        for book in book_objects:
            data.append(book.to_json())
        return HttpResponse(json.dumps(Formatter.return_table_data(0, "", len(data), data)),
content_type="application/json")
```

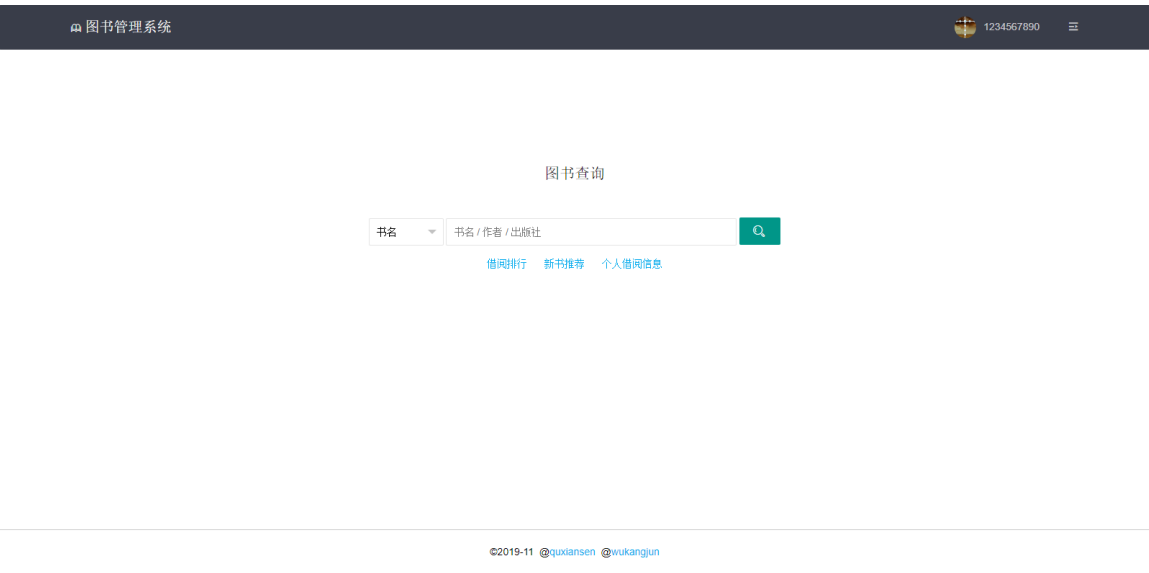


图 11 图书查询界面

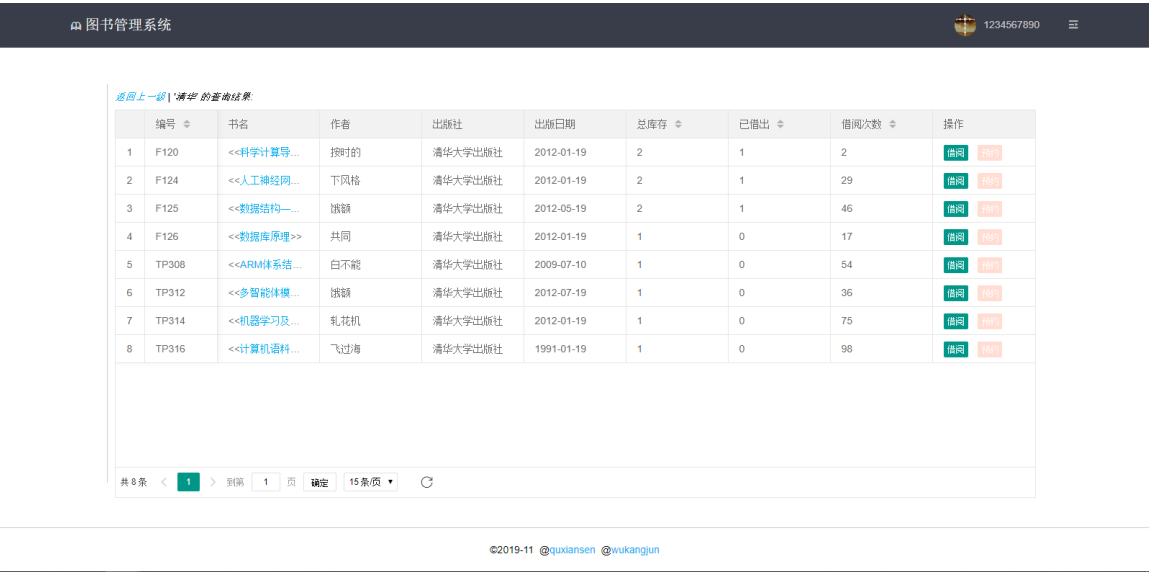


图 12 图书查询结果界面

## （二）书籍详情查看

```
def search_book(request, no):  
    """ 渲染单本图书查询界面 """  
  
    if request.method == 'GET':  
  
        try:
```

```

book_object = BmBook.objects.get(no=no) # 查询图书信息
book = book_object.to_json()
book['borrowable'] = book['inventory'] - book['lend_num']
except BmBook.DoesNotExist:
    book = {}
finally:
    comment_objects = BmComment.objects.filter(book_no=no)

    comment_list = []
    for i, comment in enumerate(comment_objects):
        comment_list.append(comment.to_json())
        student_object = BmStudent.objects.get(no=comment.stu_no)
        comment_list[i]['stu_name'] = student_object.name

    return render(request, 'book/book.html', {
        'cur_user': request.session.get('userinfo')['no'], 'book': book, 'comment_list':
comment_list})

```

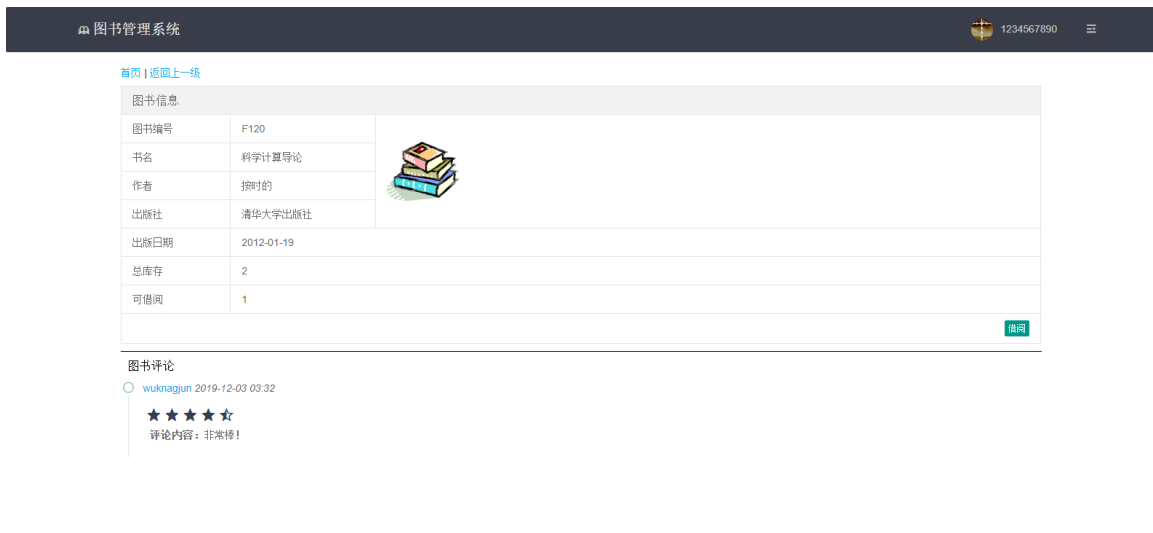


图 13 图书详情页面

### （三）新书推荐

```
def new_book(request):  
    """ 渲染新书推荐界面 """  
    if request.method == 'GET':  
        return render(request, 'book/new_book.html', {'cur_user':  
request.session.get('userinfo')['no']})  
    elif request.method == 'POST':  
        book_objects = BmBook.objects.all().order_by("-p_date")[0: 14]  
  
        books = []  
        for book in book_objects:  
            books.append(book.to_json())  
  
        return HttpResponse(json.dumps(Formatter.return_table_data(0, "", len(books),  
books)))
```

| 图书管理系统       |       |               |     |           |            |     |      |    |       |
|--------------|-------|---------------|-----|-----------|------------|-----|------|----|-------|
| 1234567890   |       |               |     |           |            |     |      |    |       |
| 返回上一级   新书推荐 |       |               |     |           |            |     |      |    |       |
| 编号           | 书名    | 作者            | 出版社 | 出版日期      | 总库存        | 已借出 | 借阅次数 | 操作 |       |
| 1            | TP309 | <<ARM微控制...   | 仍然如 | 航空航天大学... | 2013-10-03 | 2   | 1    | 63 | 借阅 下架 |
| 2            | TP313 | <<关于知识的...    | 如同眼 | 巴蜀书社      | 2013-01-19 | 1   | 0    | 78 | 借阅 下架 |
| 3            | G545  | <<语义网简明...    | 仆仆和 | 高等教育出版社   | 2012-09-19 | 2   | 2    | 97 | 借阅 下架 |
| 4            | G547  | <<智能Agent及... | 彻底  | 邮电大学出版社   | 2012-08-19 | 2   | 0    | 45 | 借阅 下架 |
| 5            | TP312 | <<多智能体模...    | 战顿  | 清华大学出版社   | 2012-07-19 | 1   | 0    | 36 | 借阅 下架 |
| 6            | F125  | <<数据结构一...    | 战顿  | 清华大学出版社   | 2012-05-19 | 2   | 1    | 46 | 借阅 下架 |
| 7            | F121  | <<嵌入式实时...    | 方法他 | 航空航天大学... | 2012-03-19 | 1   | 1    | 92 | 借阅 下架 |
| 8            | G543  | <<数据挖掘次...    | 天天  | 机械工业出版社   | 2012-02-19 | 3   | 2    | 78 | 借阅 下架 |
| 9            | F120  | <<科学计算导...    | 按时的 | 清华大学出版社   | 2012-01-19 | 2   | 1    | 2  | 借阅 下架 |
| 10           | F122  | <<嵌入式系统...    | 好看  | 航空航天大学... | 2012-01-19 | 2   | 2    | 61 | 借阅 下架 |
| 11           | F124  | <<人工智能网...    | 下风格 | 清华大学出版社   | 2012-01-19 | 2   | 1    | 29 | 借阅 下架 |
| 12           | F126  | <<数据库原理>>     | 共同  | 清华大学出版社   | 2012-01-19 | 1   | 0    | 17 | 借阅 下架 |
| 13           | G544  | <<无线网络结...    | 明明内 | 科学出版社     | 2012-01-19 | 2   | 0    | 76 | 借阅 下架 |
| 14           | G546  | <<语义信息模...    | 自涪州 | 电子工业出版社   | 2012-01-19 | 2   | 1    | 20 | 借阅 下架 |

图 14 新书推荐页面

### （四）图书借阅

```
def borrow(request):
```

```
""" 处理图书借阅响应 """

if request.method == 'POST':
    book_no = request.POST.get('book_no', None)

    borrowed_object = BmBorrow.objects.filter(Q(stu_no=request.session.get(
        'userinfo')['no']) & Q(book_no=book_no) & Q(is_return=0))

    if BmBorrow.objects.filter(Q(stu_no=request.session.get('userinfo')['no']) &
        Q(is_return=0)).count() == 10:
        code = 1 # 超过 10 本可借阅数量
    elif len(borrowed_object) != 0:
        code = 1 # 不可重复借阅
    else:
        # 修改图书信息
        book = BmBook.objects.get(no=book_no)
        book.lend_num += 1
        book.borrow_num += 1
        book.save()

        # 添加借阅信息
        BmBorrow.objects.create(
            stu_no=request.session.get('userinfo')['no'],
            book_no=book_no,
            borrow_date=datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            deadline=(datetime.now()+timedelta(days=30)
                ).strftime("%Y-%m-%d %H:%M:%S"),
            return_date=None,
            is_return=0,
```

```

        is_renewal=0
    )

    # 查询预约信息

    try:

        BmReservation.objects.get(

            Q(stu_no=request.session.get('userinfo')['no']) &
            Q(book_no=book_no)).delete()

        except BmReservation.DoesNotExist:

            pass

        code = 0

    return HttpResponse(json.dumps(Formatter.return_data(code)))

```

[首页](#) | [返回上一级](#)  
 姓名: wuknagjun  
 学号: 1234567890  
 班级: C01  
 专业: ruanjiangongcheng  
 联系方式: 13123454567

|      |     |     |     |
|------|-----|-----|-----|
| 借阅记录 | 借阅中 | 已归还 | 预约中 |
|------|-----|-----|-----|

|   | 编号    | 书名          | 借阅日期             | 应还日期             | 归还日期 | 归还状态 | 续借状态 | 操作                                    |
|---|-------|-------------|------------------|------------------|------|------|------|---------------------------------------|
| 1 | TP316 | <<计算机数据库... | 2019-12-09 12:37 | 2020-01-08 12:37 |      |      |      | <a href="#">归还</a> <a href="#">续借</a> |

图 15 图书借阅界面

## （五）图书预约

```

def reserve(request):

    """ 处理图书预约响应 """

    if request.method == 'POST':

        book_no = request.POST.get('book_no', None)

```



```

try:
    if BmReservation.objects.get(Q(stu_no=request.session.get('userinfo')['no']) &
Q(book_no=book_no)):
        code = 1
except BmReservation.DoesNotExist:
    code = 0
    # 添加预约信息
    BmReservation.objects.create(
        stu_no=request.session.get('userinfo')['no'],
        book_no=book_no,
        date=datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    )

return HttpResponse(json.dumps(Formatter.return_data(code)))

```

```

def cancel_reserve(request):
    """ 取消预约响应 """
    if request.method == 'POST':
        id = request.POST.get('id', None)

        try:
            BmReservation.objects.get(id=id).delete()
        except BmReservation.DoesNotExist:
            pass

    return HttpResponse(json.dumps(Formatter.return_data()))

```



图 16 图书预约界面



图 17 取消预约

## （六）图书续借

def renewal(request):

""" 处理图书续借响应 """

if request.method == 'POST':

id = request.POST.get('id', None)

try:

borrow\_object = BmBorrow.objects.get(id=id)

borrow\_object.deadline = (

```

        borrow_object.deadline +
timedelta(days=30)).strftime("%Y-%m-%d %H:%M:%S")

        borrow_object.is_renewal = 1

    borrow_object.save()

    code = 0

except BmBorrow.DoesNotExist:

    code = 1

return HttpResponse(json.dumps(Formatter.return_data(code)))

```



图 18 图书续借页面

## (七) 图书归还

```

def return_book(request):
    """ 处理图书归还响应 """

    if request.method == 'POST':

        id = request.POST.get('id', None)

        book_no = request.POST.get('book_no', None)

    try:

        borrow = BmBorrow.objects.get(id=id)

```

```

borrow.return_date = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

borrow.is_return = 1

borrow.is_renewal = 0

borrow.save()

book = BmBook.objects.get(no=book_no)

book.lend_num -= 1

book.save()

code = 0

except:

    code = 1

return HttpResponse(json.dumps(Formatter.return_data(code)))

```



图 19 图书归还页面

## (八) 图书评价

```

def comment_add(request):

    """ 处理添加评论响应 """

    if request.method == 'POST':

```

```
BmComment.objects.create(
    stu_no=request.session.get('userinfo')['no'],
    book_no=request.POST.get('book_no'),
    content=request.POST.get('content'),
    score=request.POST.get('score'),
    date=datetime.now().strftime("%Y-%m-%d %H:%M:%S")
)

return HttpResponse(json.dumps(Formatter.return_data()))
```

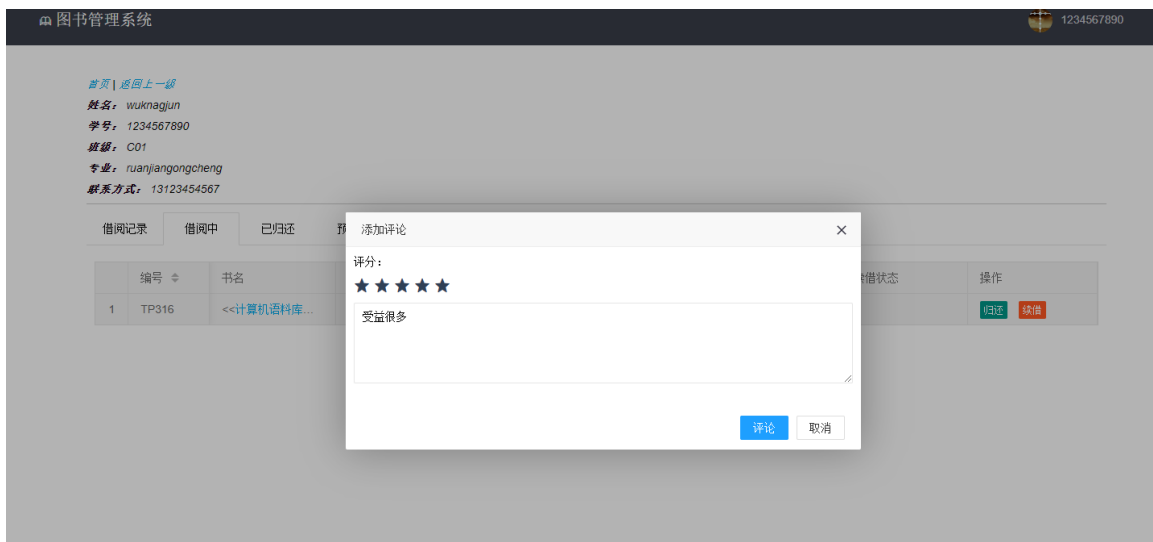


图 20 图书评价页面

## 八、 总结

从范围上来说，该项目最终在预定时间内基本上实现了所要求的功能，并没有做更多的扩充。

但从学习的角度来说，本次的课程项目对个人的能力也有一定的提升作用。首先，我们选择了一种并不大众的 `python` 的 `Web` 端框架来实现，尽管开始的时候遇到了各种小的问题，但最终也得到解决，这对我们的个人技术和合作开发方面有很大的帮助；其次，本次课程项目的时间尽管很短，但也为我们提供了一个很好的时间机会。总体来说，在项目开发中个人收获颇多。