
Southwest University

Course Design

Couse Name	<u>数据结构课程设计</u>
Semester	<u>2021 - 2022 - 2</u>
Grade	<u>2020</u>
Student Name	<u>廖润泽</u>
Student No.	<u>222020321102007</u>
Tutor	<u>欧灵</u>

School of Computer and Information Science

目录

1 概述和亮点	3
2 功能需求分析	4
3 数据库设计	5
4 系统总体设计	9
4.0 开发/测试环境.....	9
4.1 用户登录模块.....	11
4.2 图书信息管理模块.....	12
4.3 管理员信息修改模块.....	13
4.4 图书借阅还书模块.....	14
4.5 图书查询模块.....	15
4.6 图书借阅模块.....	17
4.7 查询借阅历史模块.....	19
4.8 书籍登记模块.....	21
4.9 书籍入库模块.....	23
5 Test Case	26
6 总结与展望	29
6.0 缺点分析.....	29
6.1 课程设计总结.....	29
6.2 展望.....	30
7 附录	31
8 参考文献	32
9 致谢	33

图书馆管理系统

1 概述和亮点

随着计算机的普及和信息技术的发展，人们的生活发生了日新月异的变化，各类计算机软件逐渐渗透到了社会的每个角落，大大地改善了人们的生活质量，提高了人们的工作效率。在高校中，图书借阅是学生获取知识的一个很重要的途径，如何既能方便学生借书，又能减轻图书馆管理人员的工作负担，高效地完成图书借阅管理工作，是一件非常重要的事情。

西南大学的计信院将会拥有一个小型图书馆，为全校师生提供学习、阅读的空间。近几年来，随着生源的不断扩大，图书馆的规模也随之扩大，图书数量也相应地打量增加，有关图书借阅的各种信息成倍增加。面对如此巨大的信息量，图书馆管理人员很难支撑。因此，我们打算开发和建立一套合理实用的图书借阅管理系统软件，以对校内的图书借阅信息进行统一、集中的管理。

本程序最大的几个亮点在于

1. 数据管理：使用了这学期我们专业的核心课程数据库，将数据放进了数据库后端，这样数据不是放在内存或者一个简单的文件里面，设计了相应的标准 ER 图，并通过与老师和同学对系统的不断讨论，研究和探索如何设计一个合理、最合适的代码类型来建立数据库，并将数据库中的数据连接到程序中，最后使用 SQL 语句进行查询。
2. 前端和后端的互联：使用 LayUI 进行前端框架设计，LayUI 可以快速帮我们搭建一个系统化的 web 前端，我们只在后端系统设计和代码上花了不少心思，但是对于我们这种只会敲代码做的不注重美学设计的它可以帮到我们很多，给大家安利，一个非常好用的前端框架。
3. 模块和模块之间设计的独立：尽量减少每个模块之间的依赖关系使得各自独立完成自己要实现的功能，这一点对于工程性能角度来说十分重要，因为相对独立的几个模块利于程序后续的扩展与开发维护。
4. 结合数据结构课程设计和数据库课程设计和计算机网络的一点知识，学习 flask。Flask 是一个用 Python 编写的轻量级 Web 应用程序框架。学习成本不高。结合之前的前端模板框架，我们可以用少量的 Code 就可以写出非常漂亮的程序和界面，而不需要我们自己造轮子。越学越觉得结合使用别人做的轮子很重要，不管是日常工作还是科研。

2 功能需求分析

本系统主要实现对图书馆图书借阅信息的管理，主要管理读者信息、图书信息、借阅与归还信息、系统用户的信息。

（1）读者信息管理：能够对读者的基本信息进行管理，读者信息的修改，如学生转到别的专业，此时要修改学生的基本信息；删除读者的信息，比如某个学生中途退学了，可以将其信息删除。查询读者的信息，看这个学生借阅了多少书，是否欠费，是否未还

（2）图书信息管理：能够对图书的基本信息进行管理，包括新增图书，学校每年会购进新书，此时需要将新书的信息录入系统中；图书信息的修改，如学生借书后将图书丢失，此时需要修改图书的总数量，使总数减 1；删除图书，学校在购进新书的同时，每年会对过期的图书进行清理，不再提供借阅，此时就需要将这些图书的信息从系统中删除。查询图书的信息，比如要查看有哪些是 Java 相关的书籍或者指定 ISBN 号的图书等。

（3）图书借阅信息管理：能够对图书的借阅信息进行记录，包括读者信息、图书信息、借阅时间等信息。

（4）图书归还信息管理：能够对图书的借阅信息进行记录，包括读者信息、图书信息、归还时间、是否超期、罚金等信息。

（5）系统用户信息管理：能够对系统用户的信息进行管理，包括增加新的系统操作用户，对当前系统用户的密码进行修改，以及删除某一用户。

根据上述功能需求，本系统具体的功能模块图如下图所示：

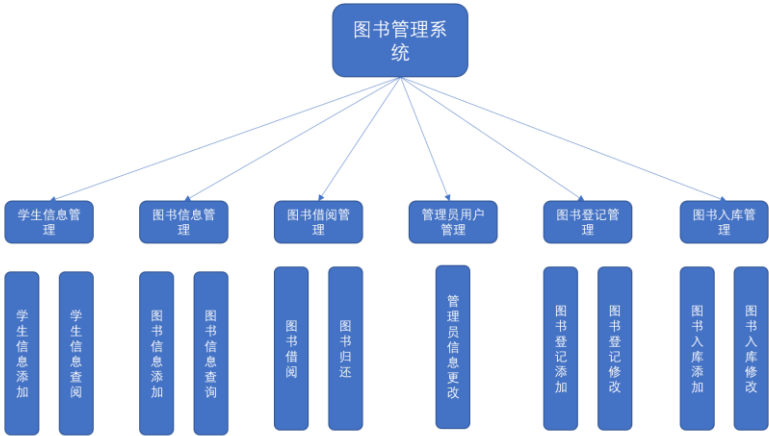


图 2.1 功能模块图

3 数据库设计

首先要分析的是每个实体需要什么属性。比如在 student 这个实体中，我们设计了 card_id（认为我们需要借用借书证，类似于西南大学的图书馆，比如参考现有的系统设计也可以极大的帮助我们设计自己的 system）、student_id、student_name、sex、telephone、enroll_date、valid_date（学生等基本信息）。

之后，我们像学生一样一步一步地设计实体，并参考了很多其他的系统来验证我们自己设计的系统的合理性。

我们解释设计实体的关系和属性如下：

- Entity: admin

It should contain four attributes:

- Admin_id: The primary key is automatically generated by the database auto-increment function.
- Admin_name: This attribute contains the name of the administrators.
- Password: With the password the administrator can have access to the database
- Right: The administrator's permissions to the library

- Entity: book

It should contain five attributes:

- ISBN: The International Standard Book Number (ISBN) is a numeric commercial book identifier that is intended to be unique.
- Book_name: This attribute contains the name of the books, so the readers can retrieve it.
- Author: The author of the book, the basic information.
- Press: The publishing house of the book, the basic information.
- Class_name: Type of book, the basic information.

- Entity: student

It should contain several attributes:

- Card_id: Each library card has a card number, and the library uses the card number to retrieve the information of the corresponding readers.
- Student_name: This attribute contains the name of the student, so the readers can retrieve it.
- Student_id: The author of the book, the basic information.
- Sex: The sex of the reader.
- Telephone: The telephone number of the book.
- Enroll_date: The date that the library card in effect.
- Valid_date: The date that the card is out of dated.
- Loss: Determine whether the reader lose the book.
- Dept: The money that library owes the reader.

- Entity: inventory

It should contain several attributes:

- Barcode: The primary key is automatically generated by the database auto-increment function. A unique barcode corresponds to a unique book.
- ISBN: The International Standard Book Number (ISBN) is a numeric commercial book identifier that is intended to be unique.
- Storage_date: The date that the inventory income the new book.
- Location: The location of the book in the library.
- Withdraw: Whether the book is withdrawn.
- Status: The date that the library card is in effect
- Admin: The administrator who put them in the library.

● Relation: readbook

It should contain several attributes:

- Id: The identifier of the every time borrowing a book.
- Start_date: The date when the book was borrowed for the first time.
- End_date: The date the reader returned the book
- Due-date: Expiry date of the borrowed book.
-

● Relation: register

It should contain several attributes:

- ISBN: The International Standard Book Number (ISBN) is a numeric commercial book identifier
- Re_date: The date when the book was borrowed for the first time.
- Admin_id: The administrator who registers the whole book.

● Relation: store_in

It should contain several attributes:

- Barcode: The primary key is automatically generated by the database auto-increment function. A unique barcode corresponds to a unique book.
- In_date: The date when the book was borrowed for the first time.
- Location: The location of the book in the library.
- Admin_id: The administrator who registers the whole book.

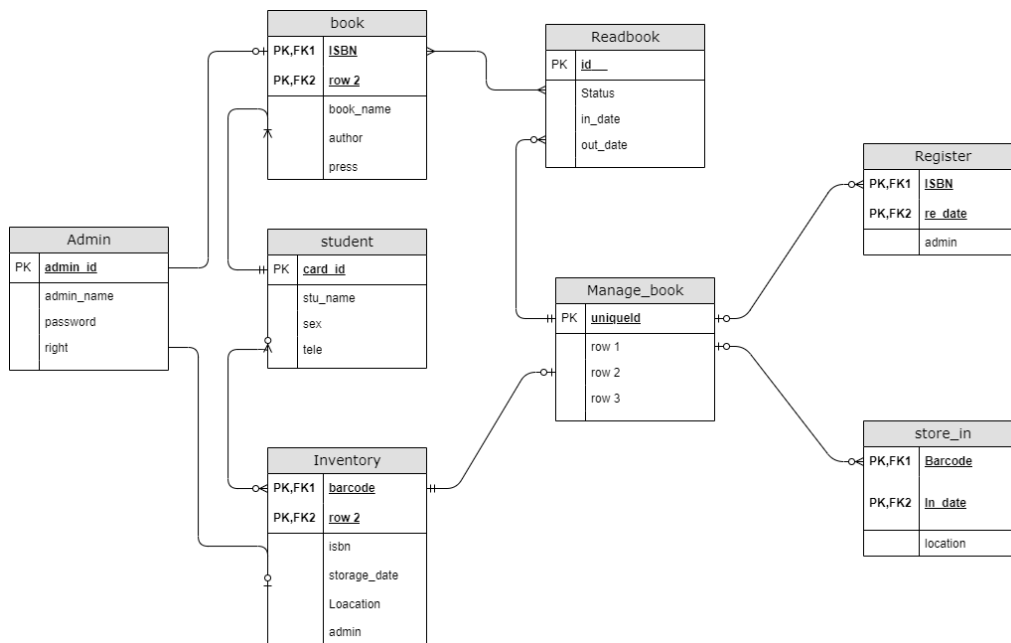


图 3.1 Entity Graph

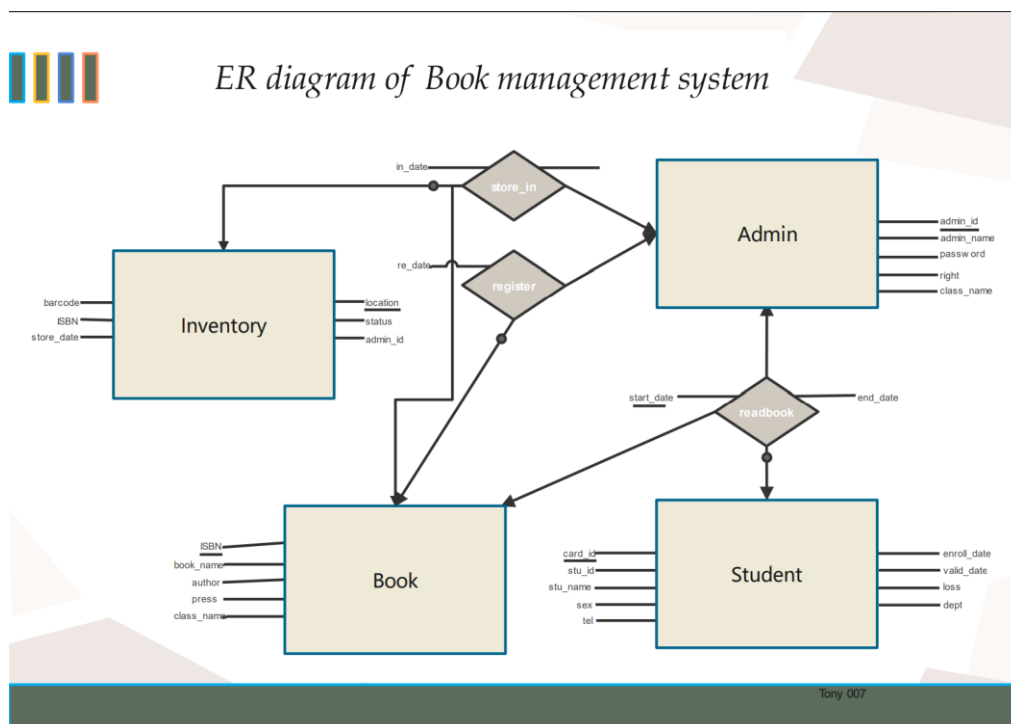


图 3.2 ER diagram of the book management system

和之前实验的 book 管理相比，我们的重点不在记录 bookinfo 上，但是对于 bookinfo 我们还是有一定的设计的。这一次，我们在课程设计上采取了不同的做法，重新按照现实生活中的图书馆设计，参考了西南大学图书馆和众多公共图书馆的设计。图书管理员购买图书后，首先对图书进行登记和分类。注册分类后，我们就可以将图书存放在图书馆对应的图书馆中了。图书馆或书架。因此，我们在 ER 中又设计了两个关系，store in 和 register。在我们现实生活的系

统中，管理员可能不是一个人，而是学生选择的还书机器设备（图书馆里常见的自助）。还书机）。

4 系统总体设计

4.0 开发/测试环境

软件环境：Windows 10 专业版 64 位

硬件环境：ROG 幻 14

1.处理器：AMD Ryzen 7 4900H

2.运行内存：16GB DDR4 3200Mhz

3.显卡：NVIDIA GeForce RTX 2060

开发工具：Pycharm, LayUI, Flask, Sqlite.

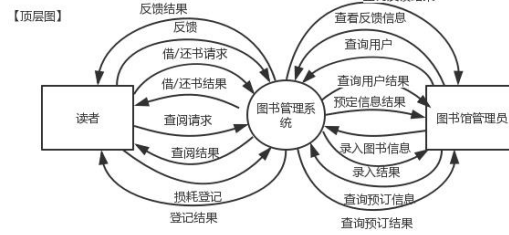
软件环境：Mac

硬件环境：Macbook pro

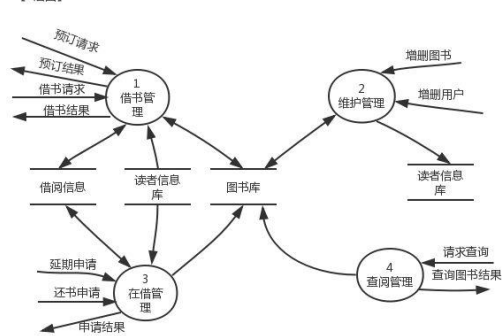
1.处理器：M1

2.运行内存：8GB DDR4 3200Mhz

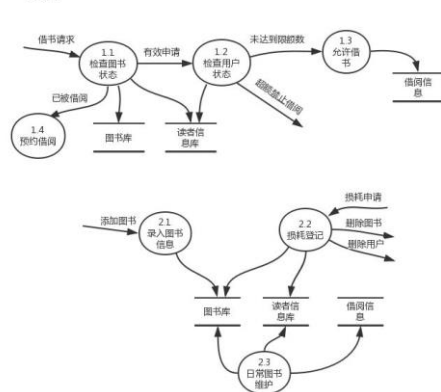
开发工具：Pycharm, LayUI, Flask, Sqlite.



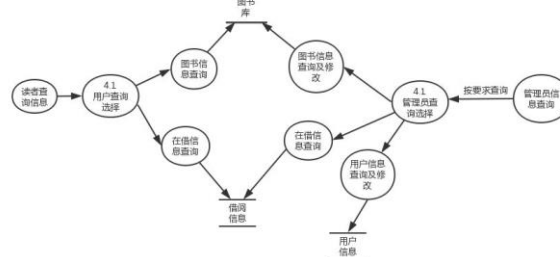
顶层图



1 层图



2 层图

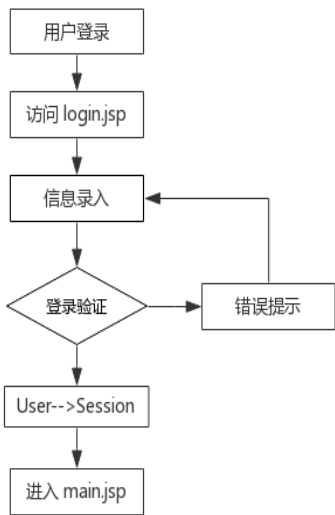


底层图

4.1 用户登录模块

当用普通用户输入用户名和密码后，进行验证，如果用户名和密码其中有一个不正确，则不能登录，并有相应的提示，如果用户没有账号，可以在登录页面进行注册，普通用户可以进行个人资料的查询、修改和修改密码，普通用户可以查询图书信息并借阅和退还，在首页面会展示图书借阅信息和借阅历史信息。

当用管理员输入用户名和密码后，进行验证，如果用户名和密码其中有一个不正确，则不能登录，并有相应的提示，管理员可以进行个人资料的查询、修改和修改密码，管理员可以对图书信息，读者信息，借阅信息进行查看和管理，对图书进行分类



登录流程图

登录

201801

...

登录

忘记密码请与系统管理员联系

登陆界面

测试：账号密码输入错误 提示，账号或密码输入错误，请重新输入

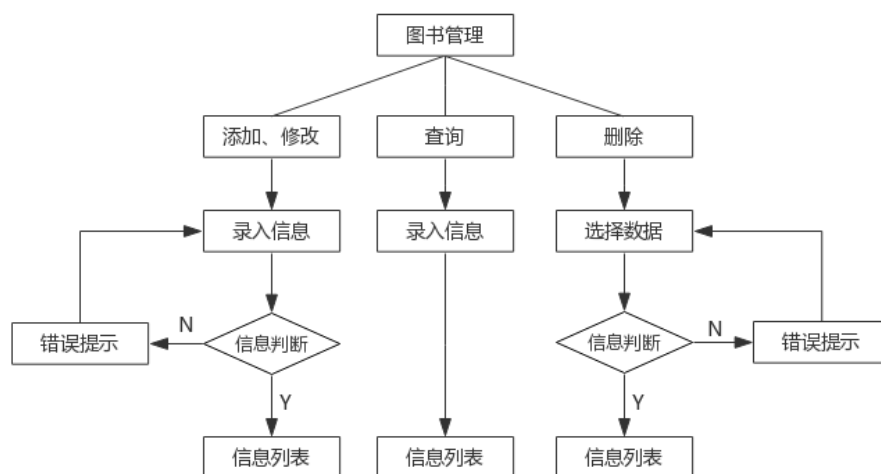


其登陆界面模块代码如下：

```
@app.route('/', methods=['GET', 'POST'])
def login():
    form = Login()
    if form.validate_on_submit():
        user = Admin.query.filter_by(admin_id=form.account.data,
password=form.password.data).first()
        if user is None:
            flash('账号或密码错误! ')
            return redirect(url_for('login'))
        else:
            login_user(user)
            session['admin_id'] = user.admin_id
            session['name'] = user.admin_name
            return redirect(url_for('index'))
    return render_template('login.html', form=form)
```

4.2 图书信息管理模块

管理员对图书的数量，图书号，图书类型，图书名称等进行修改、删除管理，如果普通用户在线借书，图书总数会相应减少。



4.3 管理员信息修改模块

管理员对管理员的信息可以进行进行修改、删除，并且可以主动添加新用户，该模块显示用户借阅天数，天数会自动变化，管理员可以在线监控读者借阅天数，如果到期未还者，账号则不能登录。

— 修改个人信息 —

账号：

201801

用户名：

廖润泽

权限：

入库 借书 还书 注销

账号、权限无法修改，若要修改请联系系统管理员

提交

修改管理员信息界面

其修改信息界面模块代码如下：

```
def change_info():
    form = EditInfoForm()
    if form.validate_on_submit():
```

```

        current_user.admin_name = form.name.data
        db.session.add(current_user)
        flash(u'已成功修改个人信息!')
        return redirect(url_for('user_info', id=current_user.admin_id))
    form.name.data = current_user.admin_name
    id = current_user.admin_id
    right = current_user.right
    return render_template('change-info.html', form=form, id=id,
right=right)

```

4.4 图书借阅还书模块

功能描述：管理员可以在线管理借阅信息，可以帮读者还书。

学生还书

学生借阅卡号码：	<input type="text" value="16000001"/>	<input type="button" value="搜索"/>				
图书编号	ISBN	书名	起始日期	应还日期	操作	
102341	9787302423287	机器学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
411362	9787543862326	中国知青终结	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
020324	9787115552860	深度学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
309331	9787115488763	Python深度学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
102348	9787115373991	Flask Web开发: 基于Python的...	2019-03-28	2019-05-07	<input type="button" value="归还"/>	

点击归还 Flask Web 开发，我们可以看到，已经没有这一项了

学生还书

学生借阅卡号码：	<input type="text" value="16000001"/>	<input type="button" value="搜索"/>				
图书编号	ISBN	书名	起始日期	应还日期	操作	
102341	9787302423287	机器学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
411362	9787543862326	中国知青终结	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
020324	9787115552860	深度学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	
309331	9787115488763	Python深度学习	2019-03-28	2019-05-07	<input type="button" value="归还"/>	

图书借阅还书模块相应代码如下：

```

@app.route('/out', methods=['GET', 'POST'])
@login_required
def out():

```

```

    today_date = datetime.date.today()
    today_str = today_date.strftime("%Y-%m-%d")
    #today_stamp = time.mktime(time.strptime(today_str + ' 00:00:00',
'%Y-%m-%d %H:%M:%S'))
    today_stamp = 1553739404.0
    barcode = request.args.get('barcode')
    card = request.args.get('card')
    book_name = request.args.get('book_name')
    readbook = ReadBook()
    readbook.barcode = barcode
    readbook.card_id = card
    readbook.start_date = int(today_stamp)*1000
    readbook.due_date = (int(today_stamp)+40*86400)*1000
    readbook.borrow_admin = current_user.admin_id
    db.session.add(readbook)
    db.session.commit()
    book = Inventory.query.filter_by(barcode=barcode).first()
    book.status = False
    db.session.add(book)
    db.session.commit()
    bks =
db.session.query(Book).join(Inventory).filter(Book.book_name.contains(b
ook_name), Inventory.status == 1).\
    with_entities(Inventory.barcode, Book.isbn, Book.book_name,
Book.author, Book.press).all()
    data = []
    for bk in bks:
        item = {'barcode': bk.barcode, 'isbn': bk.isbn, 'book_name':
bk.book_name,
                'author': bk.author, 'press': bk.press}
        data.append(item)
    return jsonify(data)

```

4.5 图书查询模块

功能描述：读者可以显示当前在库里的所查询的图书信息，通过 ISBN、书名、作者、类别进行搜索，搜索后返回书籍的所有信息，本查询支持模糊搜索

查询图书信息

ISBN

查询内容: 978

搜索

ISBN	书名	出版社	作者	类别	馆藏数量	可借数量
978704015109X	ERP原理与应用实训	高等教育出版社	汪清明	管理社会科学	3	3
9787040273243	管理信息系统	高等教育出版社	黄梯云	管理社会科学	2	1
9787115335500	深入浅出node.js	人民邮电出版社	朴灵	自动化技术、计算机技术	3	3
9787121204869	移动设计	电子工业出版社	傅小贞	自动化技术、计算机技术	2	2
9787302292609	用友ERP生产管理系统实...	清华大学出版社	张莉莉	管理社会科学	3	3
978710800982x	万历十五年	生活·读书·新知三联书店	[美]黄仁宇	中国史	3	3
9787115488763	Python深度学习	人民邮电出版社	弗朗索瓦.肖莱	自动化技术、计算机技术	3	2

< 1 2 3 >

到第 1 页

确定

共 17 条

10 条/页

查阅图书信息界面

其查阅图书信息模块代码如下:

```
@app.route('/search_book', methods=['GET', 'POST'])
@login_required
def search_book():
    form = SearchBookForm()
    return render_template('search-book.html', name=session.get('name'),
form=form)

@app.route('/books', methods=['POST'])
def find_book():

    def find_name():
        return
Book.query.filter(Book.book_name.like('%'+request.form.get('content')+'
%')).all()

    def find_author():
        return
Book.query.filter(Book.author.contains(request.form.get('content'))).al
l()

    def find_class():
        return
Book.query.filter(Book.class_name.contains(request.form.get('content'))
).all()

    def find_isbn():
```

```

        return
Book.query.filter(Book.isbn.contains(request.form.get('content'))).all()

methods = {
    'book_name': find_name,
    'author': find_author,
    'class_name': find_class,
    'isbn': find_isbn
}
books = methods[request.form.get('method')]()
data = []
for book in books:
    count = Inventory.query.filter_by(isbn=book.isbn).count()
    available = Inventory.query.filter_by(isbn=book.isbn,
status=True).count()
    item = {'isbn': book.isbn, 'book_name': book.book_name, 'press':
book.press, 'author': book.author,
            'class_name': book.class_name, 'count': count,
'available': available}
    data.append(item)
return jsonify(data)

```

4.6 图书借阅模块

功能描述：通过搜索书名和输入学生卡号，进行图书借阅，生成当前借阅的时间，默认借书逾期时间不超过 1 个月

学生借书

学生借书卡号码: 书名:

图书编号	ISBN	书名	作者	出版社	操作
102348	9787115373991	Flask Web开发: 基于Python的...	(美) 格林布戈...	人民邮电出版社	<input type="button" value="借出"/>

< 1 > 列表 1 页 确定 共 1 条 10 条/页

图书借阅模块界面视图

测试：系统提示是否进行借书

学生借阅卡号码: 16000001 书名: Flask 搜索

图书编号	ISBN	书名	作者	出版社	操作
102348	9787115373991	Flask Web开发: 基于Python的...	(美) 格林布戈...	人民邮电出版社	借出

信息

确认借出吗?

确定 取消

< 1 > 到第 1 页 确定 共 1 条 10 条/页

点击确认

即可以在学生借阅查询界面查到刚刚借出的书籍:

查询学生信息

请输入学生借阅卡号码: 16000001 搜索

姓名: 王智远 性别: 女 有效期至: 2022-05-

图书编号	书名	作者	起
102348	Flask Web开发: 基于Pyt...	(美) 格林布戈...	20

< 1 2 3 > 到第 3 页 确定 共 15 条 10 条/页

图书借阅相关代码如下:

```
@app.route('/out', methods=['GET', 'POST'])
@login_required
def out():
    today_date = datetime.date.today()
    today_str = today_date.strftime("%Y-%m-%d")
    #today_stamp = time.mktime(time.strptime(today_str + ' 00:00:00',
    '%Y-%m-%d %H:%M:%S'))
    today_stamp = 1553739404.0
    barcode = request.args.get('barcode')
    card = request.args.get('card')
    book_name = request.args.get('book_name')
    readbook = ReadBook()
    readbook.barcode = barcode
    readbook.card_id = card
    readbook.start_date = int(today_stamp)*1000
    readbook.due_date = (int(today_stamp)+40*86400)*1000
```

```

        readbook.borrow_admin = current_user.admin_id
        db.session.add(readbook)
        db.session.commit()
        book = Inventory.query.filter_by(barcode=barcode).first()
        book.status = False
        db.session.add(book)
        db.session.commit()
        bks =
db.session.query(Book).join(Inventory).filter(Book.book_name.contains(b
ook_name), Inventory.status == 1).\
        with_entities(Inventory.barcode, Book.isbn, Book.book_name,
Book.author, Book.press).all()
        data = []
        for bk in bks:
            item = {'barcode': bk.barcode, 'isbn': bk.isbn, 'book_name':
bk.book_name,
                    'author': bk.author, 'press': bk.press}
            data.append(item)
        return jsonify(data)

```

4.7 查询借阅历史模块

功能描述：显示当前登录用户的图书借阅记录。并显示是否归还图书。

请输入学生借阅卡号码:

姓名: 王智远 性别: 女 有效期至: 2022-05-30 是否欠费: 否

图书编号	书名	作者	起始日期	应还日期	还书日期
102341	机器学习	周志华	2022-04-20	2019-01-19	2022-05-30
310321	万历十五年	[美]黄仁宇	2022-04-20	2019-02-07	2022-05-30
203773	ERP原理与实践	苟娟琼,常丹	2022-04-20	2019-02-07	2022-05-30
309333	Python深度学习	弗朗索瓦.肖莱	2022-04-20	2019-02-11	2022-05-30
102341	机器学习	周志华	2022-04-20	2019-02-21	2022-05-30
102347	Flask Web开发: 基于Pyt...	(美) 格林布戈...	2022-04-20	2019-02-21	2022-05-30
102342	机器学习	周志华	2022-04-20	2019-05-07	2022-05-30

< 1 2 > 到第 1 页 确定 共 14 条 10 条/页 v

借阅历史模块界面

学生借阅历史模块相关代码如下：

```

@app.route('/search_student', methods=['GET', 'POST'])
@login_required

```

```
def search_student():
    form = SearchStudentForm()
    return render_template('search-student.html',
name=session.get('name'), form=form)

def timeStamp(timeNum):
    if timeNum is None:
        return timeNum
    else:
        timeStamp = float(float(timeNum)/1000)
        timeArray = time.localtime(timeStamp)
        print(time.strftime("%Y-%m-%d", timeArray))
        return time.strftime("%Y-%m-%d", timeArray)

@app.route('/student', methods=['POST'])
def find_student():
    stu =
Student.query.filter_by(card_id=request.form.get('card')).first()
    if stu is None:
        return jsonify([])
    else:
        valid_date = timeStamp(stu.valid_date)
        return jsonify([{'name': stu.student_name, 'gender': stu.sex,
'valid_date': valid_date, 'debt': stu.debt}])

@app.route('/record', methods=['POST'])
def find_record():
    records =
db.session.query(ReadBook).join(Inventory).join(Book).filter(ReadBook.c
ard_id == request.form.get('card'))\
        .with_entities(ReadBook.barcode, Inventory.isbn, Book.book_name,
Book.author, ReadBook.start_date,
                        ReadBook.end_date, ReadBook.due_date).all()

    data = []
    for record in records:
        start_date = timeStamp(record.start_date)
        due_date = timeStamp(record.due_date)
        end_date = timeStamp(record.end_date)
        if end_date is None:
            end_date = '未归还'
```

```
        item = {'barcode': record.barcode, 'book_name':  
record.book_name, 'author': record.author,  
                'start_date': start_date, 'due_date': due_date,  
                'end_date': end_date}  
        data.append(item)  
    return jsonify(data)
```

4.8 书籍登记模块

功能描述：通过输入新书的 ISBN、书名、出版社、作者、类别对图书馆新购置的图书进行登记，登记完了才能入库。

新书登记

ISBN:

ISBN9787020104260

书名:

罗生门

出版社:

人民文学出版社

作者:

[日]芥川龙之介

类别:

文学

提交

书籍登记模块界面图

测试：

- 提示 ISBN 长度错误，需要我们重新输入

ISBN长度错误

新书登记

ISBN:

书名:

出版社:

作者:

类别:

提交

- 图书注册成功，我们已经可以在图书信息中检索到，但还无法进行操作，因为图书还没有入库。

图书信息添加成功!

新书登记

ISBN:

书名:

出版社:

作者:

类别:

提交

查询图书信息

书名

查询内容: 罗生门

搜索

ISBN	书名	出版社	作者	类别	馆藏数量	可借数量
9787020104260	罗生门	人民文学出版社	[日]芥川龙之介	文学	0	0

图书登记注册相应代码:

```
@app.route('/new_store', methods=['GET', 'POST'])
@login_required
def new_store():
    form = NewStoreForm()
    if form.validate_on_submit():
        if len(request.form.get('isbn')) != 13:
            flash(u'ISBN 长度错误')
        else:
            exist =
Book.query.filter_by(isbn=request.form.get('isbn')).first()
            if exist is not None:
                flash(u'该图书信息已经存在，请核对后再录入；或者填写入库表。')
            else:
                book = Book()
                book.isbn = request.form.get('isbn')
                book.book_name = request.form.get('book_name')
                book.press = request.form.get('press')
                book.author = request.form.get('author')
                book.class_name = request.form.get('class_name')
                db.session.add(book)
                db.session.commit()
                flash(u'图书信息添加成功!')
```

```
        return redirect(url_for('new_store'))
        return render_template('new-store.html',
name=session.get('name'), form=form)
```

4.9 书籍入库模块

功能描述：通过已经存在相应的书籍和其信息，我们将他设定唯一的书籍号并将它存储到书库中。

测试：对书籍进行补充（错误 ISBN 号）

添加失败，请注意本书信息是否已录入，若未登记，请在'新书入库'窗口录入信息。

库存补充

图书编号：

ISBN：

位置：

提交

对书籍进行补充

库存补充

图书编号：

985007

ISBN：

9787020104260

位置：

3楼,1书架,1层,4排

提交

点击提交，入库成功

入库成功!

库存补充

图书编号:

ISBN:

位置:

提交

入库成功后，我们可以借阅相应的图书

学生借书

学生借阅卡号码:

16000001

书名:

罗生门

搜索

图书编号	ISBN	书名	作者	出版社	操作
985007	9787020104260	罗生门	[日]芥川龙之介	人民文学出版社	借出

图书入库相应代码如下:

```
@app.route('/storage', methods=['GET', 'POST'])
@login_required
def storage():
    form = StoreForm()
    if form.validate_on_submit():
        book =
Book.query.filter_by(isbn=request.form.get('isbn')).first()
        exist =
Inventory.query.filter_by(barcode=request.form.get('barcode')).first()
        if book is None:
            flash(u'添加失败，请注意本书信息是否已录入，若未登记，请在‘新书入
            库’窗口录入信息。')
        else:
            if len(request.form.get('barcode')) != 6:
                flash(u'图书编码长度错误')
            else:
                if exist is not None:
```

```
        flash(u'该编号已经存在! ')
    else:
        item = Inventory()
        item.barcode = request.form.get('barcode')
        item.isbn = request.form.get('isbn')
        item.admin = current_user.admin_id
        item.location = request.form.get('location')
        item.status = True
        item.withdraw = False
        today_date = datetime.date.today()
        today_str = today_date.strftime("%Y-%m-%d")
        #today_stamp = time.mktime(time.strptime(today_str
+ ' 00:00:00', '%Y-%m-%d %H:%M:%S'))
        today_stamp = 1553739404.0
        item.storage_date = int(today_stamp)*1000
        db.session.add(item)
        db.session.commit()
        flash(u'入库成功! ')
    return redirect(url_for('storage'))
    return render_template('storage.html', name=session.get('name'),
form=form)
```

5 Test Case

Graph 5.1 系统登陆界面功能测试

Project name:	Book management system	Tester:	Hansomeboy liao	Test date:	2022-5-1
Test software version:	1.0	Test description:	是否能正常登陆系统管理员账号		
Test phase:	<input checked="" type="radio"/> Unit testing <input type="radio"/> Integration testing <input type="radio"/> System test <input type="radio"/> User site test				
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	Tony		
Database	None	Test Case ID:	input-007		
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入 201801	管理员账号	进入到系统管理员界面	同预期	P
2	输入 123	管理员账号密码	程序结束, 感谢使用	同预期	D

Graph 5.2 图书查阅界面功能测试

Project name:	Book management system	Tester:	Hansomeboy liao	Test date:	2022-05-23
Test software version:	1.0	Test description:	图书信息查询		
Test phase:	<input checked="" type="radio"/> Unit testing <input type="radio"/> Integration testing <input type="radio"/> System test <input type="radio"/> User site test				
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	Tony		
Database	None	Test Case ID:	input-007		
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入图书信息	能够正确查阅到图书信息	1 成功查询	同预期	P

Graph 5.3 学生信息界面功能测试

Project name:	Book management system	Tester:	Hansomeboy liao	Test date:	2022-5-20
Test software version:	1.0	Test description:	学生信息界面		
Test phase:	<input checked="" type="radio"/> Unit testing <input type="radio"/> Integration testing <input type="radio"/> System test <input type="radio"/> User site test				
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	Tony		

subsystem					
Database	None	Test Case ID: runze liao			
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入学生卡号	能够通过查阅学生的信息	成功查阅	同预期	P
2	16000001	能够通过查阅借书信息	成功查询	同预期	P

Graph 5.4 管理员信息修改模块

Project name:	Book management system	Tester:	Hansomeboy liao	Test date:	2025-05-21	
Test software version:	1.0	Test description:	管理员信息修改			
Test phase:	<input checked="" type="radio"/> Unit testing	<input type="radio"/> Integration testing	<input type="radio"/> System test	<input type="radio"/> User site test		
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)					
Module/subsystem	menu();	Coder	Liaorunze			
Database	None	Test Case ID: input-007				
Testing procedure						
Serial number	Input description and operation	Expected results		Test data and results	Actual results	Result
1	输入廖润泽	能够正确修改管理员名字		成功修改	同预期	P
2	输入 520123	能够正确修改管理员账号		成功修改	同预期	P

Graph 5.5 书籍注册功能测试

Project name:	Book management system	Tester:	Hansomeboy liao	Test date:	2022-5-23
Test software version:	1.0	Test description:	对图书进行登记注册		
Test phase:	<input checked="" type="radio"/> Unit testing	<input type="radio"/> Integration testing	<input type="radio"/> System test	<input type="radio"/> User site test	
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	LJJ		
Database	None	Test Case ID:	input-007		
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入 ISBN、书名	能够正确登记注册	成功注册	同预期	P
2	输入出版社	能够正确登记注册	成功注册	同预期	P

Graph 5.6 书籍入库功能测试

Project name:	Book management system	Tester:	Handsome boy liao	Test date:	2022-5-22
Test software version:	1.0	Test description:	完成对书籍入库的修改		
Test phase:	<input checked="" type="radio"/> Unit testing	<input type="radio"/> Integration testing	<input type="radio"/> System test	<input type="radio"/> User site test	
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	LRz		
Database	None	Test Case ID:	input-007		
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入书号	能够正确完成对书籍入库的修改	成功入库	同预期	P
a	输入位置	能够正确完成对书籍入库的修改	成功入库	同预期	P

Graph 5.6 学生借书功能处理测试

Project name:	Book management system	Tester:	Handsome boy liao	Test date:	2022-5-23
Test software version:	1.0	Test description:	学生借书功能		
Test phase:	<input checked="" type="radio"/> Unit testing	<input type="radio"/> Integration testing	<input type="radio"/> System test	<input type="radio"/> User site test	
Definition of test results:	P: Pass A: Level 1 bug (fatal) B: Level 2 bug (functionality) C: Level 3 bug (alarm) D: Level 4 bug (constructive)				
Module/subsystem	menu();	Coder	LRz		
Database	None	Test Case ID:	input-007		
Testing procedure					
Serial number	Input description and operation	Expected results	Test data and results	Actual results	Result
1	输入卡号	成功借书	成功	同预期	P
2	输入书名	成功借书	借书成功	同预期	P

6 总结与展望

6.0 缺点分析

- 没有很好的做出需求分析，虽然和朋友同学老师都在讨论这个系统该实现什么功能，但没有很好的考虑到了实现这些功能需要什么样的数据支撑，比如之前想做像电子科技大学图书馆的学院签到图书馆统计模块，但就是不知道我们需要怎样的数据才能做出这样的效果。

- 还有图书借阅信息的统计，之前想实现我统计哪些书很受欢迎，虽然 python 中有很多好用的数据分析包（numpy, pandas），但这些很难和我用的工具 flask 对应上，很难实现上面的接口。

- 还有指导老师提到的我数据库的设计可能存在问题，在 Inventory 这个数据库实体存在一定的问题，可能没有很好的设计到它的 primary key 和实际情况的联系。

这些缺点是我在做这个系统中很容易出现的，希望在之后的程序设计和系统开发中能注意到相应的情况，只有敢做才有进步。

6.1 课程设计总结

在我们设计和实现这个系统的时候，由于时间和我们自己的编码能力，我们只实现了整个图书馆管理系统的一小部分。数据库想结合上本书的实验，扩展实现对数据库的增、删、改、查等功能（通过 SQL 语句）

- 我们通过我们假设的需求分析和设计需求，设计了相应的标准 ER 图，并通过与老师和同学对系统的不断讨论，研究和探索如何设计一个合理、最合适的代码类型来建立数据库，并将数据库中的数据连接到程序中，最后使用 SQL 语句进行查询。

- 使用 LayUI 做前端框架设计，LayUI 可以快速帮我们搭建一个系统化的 web 前端，我们只在后端系统设计和代码上花了不少心思，但是对于我们这种只会敲代码做的不注重美学设计的它可以帮到我们很多，给大家安利，一个非常好用的前端框架。

- 学会了 python 连接数据库和操作数据，对我们以后的学年设计甚至毕业设计都有铺垫作用。python 中有很多有用的包。对于我们使用 mysql 作为数据库的设计，我们可以使用 pymysql 包，它有许多新颖的数据库操作形式。

- 结合数据结构课程设计和数据库课程设计，学习 flask。Flask 是一个用 Python 编写的轻量级 Web 应用程序框架。学习成本不高。结合之前的前端模板框架，我们可以用少量的 Code 就可以写出非常漂亮的程序和界面，而不需要我们自己造轮子。越学越觉得结合使用别人做的轮子很重要，不管是日常工作还是科研。

- 最后，通过这期间课程设计的综合实践，我的编码能力也提高了很多。从前端框架模板的使用到后端连接数据库到使用 flask 写 web 端，感觉这些计算机领域需要大量的自学。根本没有足够的时间在课堂上教学。网上有很多强大的资料，比如 flask 官方文档的教程。这些方便的资料真的帮了我很多，我的编码能力也一点一点的提高。思考代码的能力是通过长期锻炼提高的。

6.2 展望

我们最终想要达到的效果就是达到这样一个效果，实现一个真正的前后端互联的网络电子图书馆，可以实现信息检索、服务指南、服务资料搜索、图书信息录入、管理。根据不同书目类型的大型图书馆、研修室预约、外人参考预约等功能，估计要实现这些功能需要大量的时间和精力。这是对我们完成如此大规模系统设计能力的考验。但是，可能我毕业设计不想做一个系统，可能想做点和我的科研方向相关的题目。

7 附录

测试数据:

#login in

id: 201801

password: 123

#register

ISBN: 9787115552860

Book_name: 深度学习

pub: 人民邮电出版社

author: [美]伊恩·古德费洛

class: 计算机科学

#store_in

barcode: 020324

ISBN: 9787115552860

location: 5楼,20书架,13层,14排

#student info query:

16000001

#Student borrows book

16000001 机器学习

#Student returns book

16000001

8 参考文献

[1]阮宏一, & 鲁静. (2011). 数据结构课程设计. 电子工业出版社.

[2]MiguelGrinberg, 格林布戈, & 安道. (2015). Flask Web 开发:基于 Python 的 Web 应用开发实战. 人民邮电出版社.

9 致谢

经过了一学期的忙碌和工作，我们的课程设计已经到了尾声，作为一个本科生的课程设计，由于经验和设计次数的匮乏，我们的程序设计难免有许多考虑不周全的地方，如果没有导师的督促和指导，以及一起工作学习的同学的支持，想要完成这个设计是难以想象的。

在这里首先要感谢我的课程设计导师欧灵老师，老师平时工作繁忙，又要完成本科生教学任务又要指导研究生的科研项目，但在我做这个课程设计的每个阶段，从查询资料到设计草案，中期检查，后期详细设计都深入浅出的给了我悉心的指导。我的设计存在着许多的问题，但是欧灵老师仍然细心地给我纠正数据库设计的错误、ER 图的错误。除了敬佩欧灵老师的专业水平之外，他的治学严谨和科学研究的精神也是我永远学习的榜样，并将积极的影响我今后的学习和工作。

其次要感谢和我一起做课程设计的同学们，我和你们一起讨论学业、技术上的问题，一起讨论系统设计是否合理。志远、颖然、宇鹏……你们的笑容一直让我如沐春风，享受并怀念一起吃火锅、楼下杏园食堂聚餐的时光。



和欧灵老师深入浅出讨论