复旦大学2020-2021春季学期软件工程Lab

第二十四小组

成员：张明、王海伟、许同樵、袁逸聪

目前阶段：Lab4

服务器IP：106.14.106.73

前端端口为80可直接访问，直接在浏览器输入即可进入首页

超级管理员账号"admin"密码"admin"(超管账号自动生成，无视格式检测。但人为创建的管理员仍然要检测)

[toc]

项目完成情况

- Lab4项目需求

功能列表	详细功能描述
读者注册学邮验证	在注册时，增加学邮的验证步骤，需要验证是真实的复旦邮箱。实现方式例如：后端可以给注册学邮发验证码，然后在前端输入验证码，将其发送到后端对输入的验证码和真实的验证码进行比较。 **注意**：可以采用某个同学的学邮作为发邮件方，给读者学邮发送邮件。在阿里云部署的时候，发邮件需要采用465端口。
读者身份区分	进一步区分读者身份，读者身份现分为教师、研究生、本科生三种。不同的读者身份有不同的 **最大借阅副本数量** 与 **借阅超期时长** 与 **预约超期时长** 设置，且可以由**超级管理员**在管理员界面动态设置每种读者身份的 **最大借阅副本数量** 与 **借阅超期时长** 与 **预约超期时长**。
预约、借阅超期设置	读者预约书籍必须在默认时间内取书，否则视为超期；读者借阅书籍必须在默认时间内还书，否则视为超期。 **超级管理员**可以对预约超期和借阅超期时长进行设置，时间精确到秒。
预约、借阅超期提醒	**超级管理员**点击预约、借阅超期及罚款批量提醒按钮后，系统自动对当前**预约已**

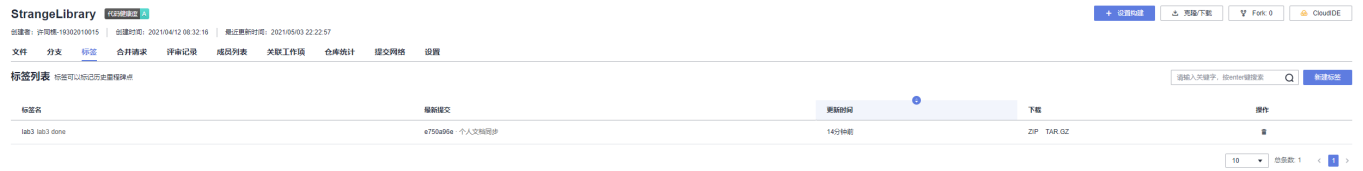
及罚款批量提醒	**超期**（此时系统要取消此预约）或 **借阅已超期** 或 **存在未缴纳罚款**的读者发送邮件提醒，每个读者发送一封邮件通知（多本书的预约超期、借阅超期提醒和罚款提醒合并在一起作为一封邮件通知）。
借阅超期罚款	读者在图书馆现场还书时，如果超过借阅期限那么按照一定的标准生成**罚款**。
图书报损及罚款	读者在图书馆现场还书时有些图书遗失或损坏，管理员确认后录入相关信息，系统生成相应的**罚款**并设置对应的图书副本状态为遗失或损坏（遗失或损坏两种情况要区分开）。遗失或损坏的图书副本不再可用，不能再借阅。
罚款查看和缴纳	图书新增 **价格** 属性，预约超期不罚款，借阅超期罚款原价 1/4 ，图书损坏罚款原价 1/2，图书遗失罚款原价。 读者在网页上自主查看目前所欠罚款，可以通过在线支付缴纳罚款。 **注意**：助教提供支付模拟接口: http://47.103.205.96:8080/api/payment 具体使用文档请见: https://github.com/h10gforks/pay-service/blob/main/README.md
系统记录	在读者预约、借阅、还书、被罚款等操作的时候，每个操作都需要记录一条信息作为系统历史记录。（精确到秒的时间、操作的管理员、读者、所在分馆等） 管理员可以看到每个副本下的历史记录，也可以根据读者信息查询对应的历史记录。 读者在个人信息页面，可以看到自己的历史记录。

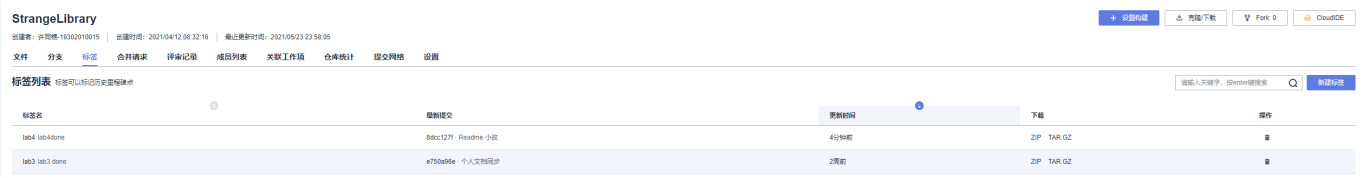
- 项目需求规划

  参见doc文件夹中lab4项目规划.pdf

- 代码检查结果

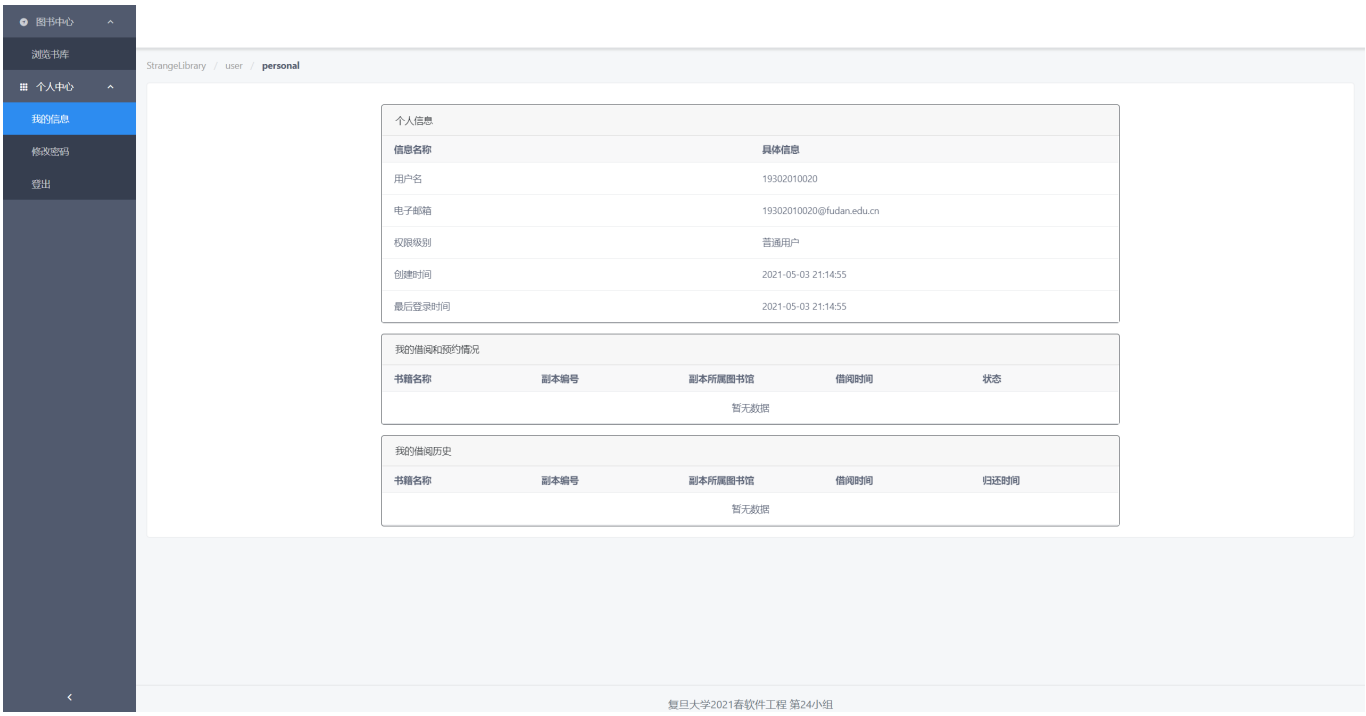
- 代码tag截图



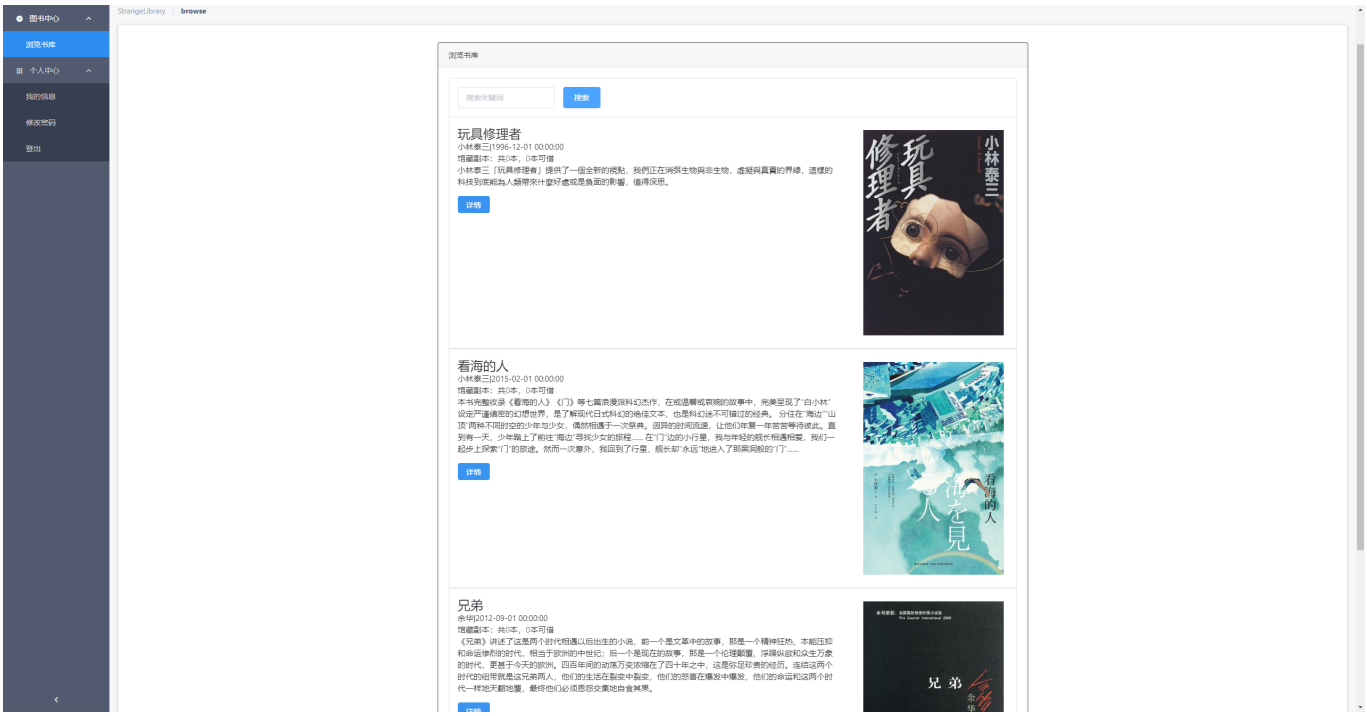


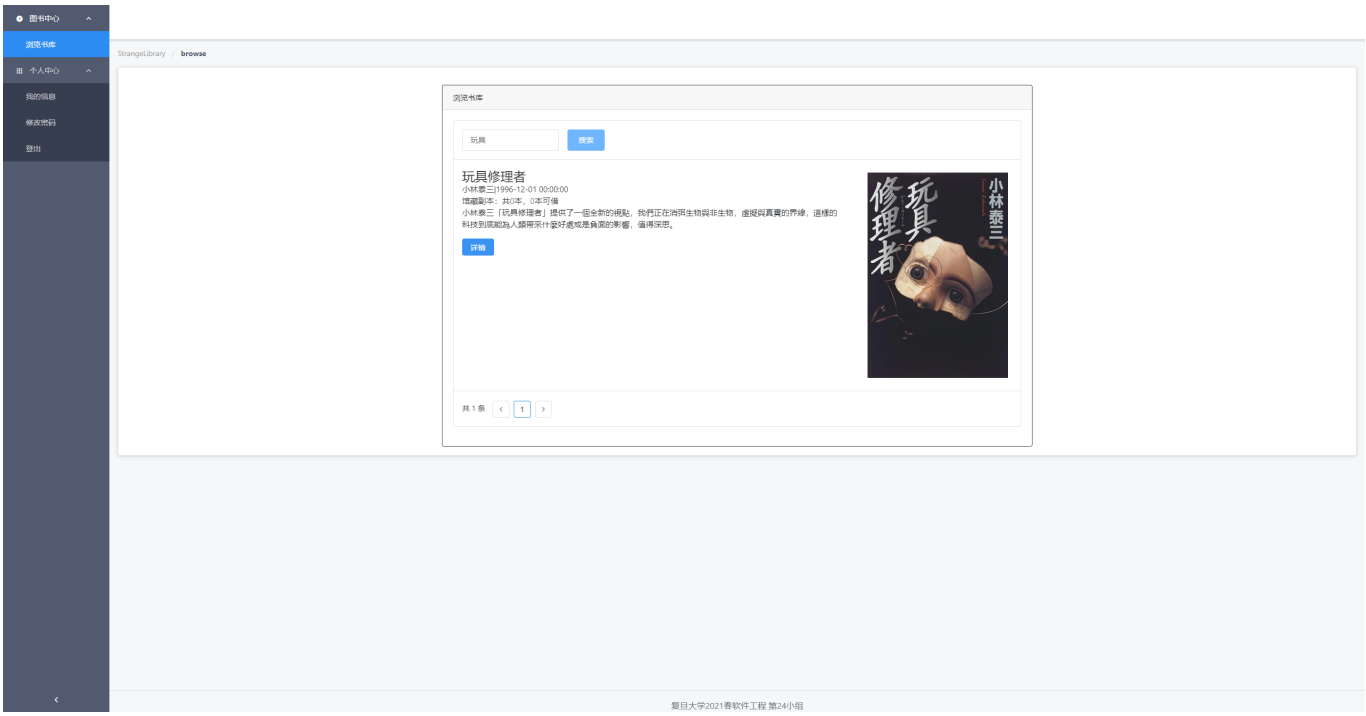
主要功能截图

- 个人中心页面

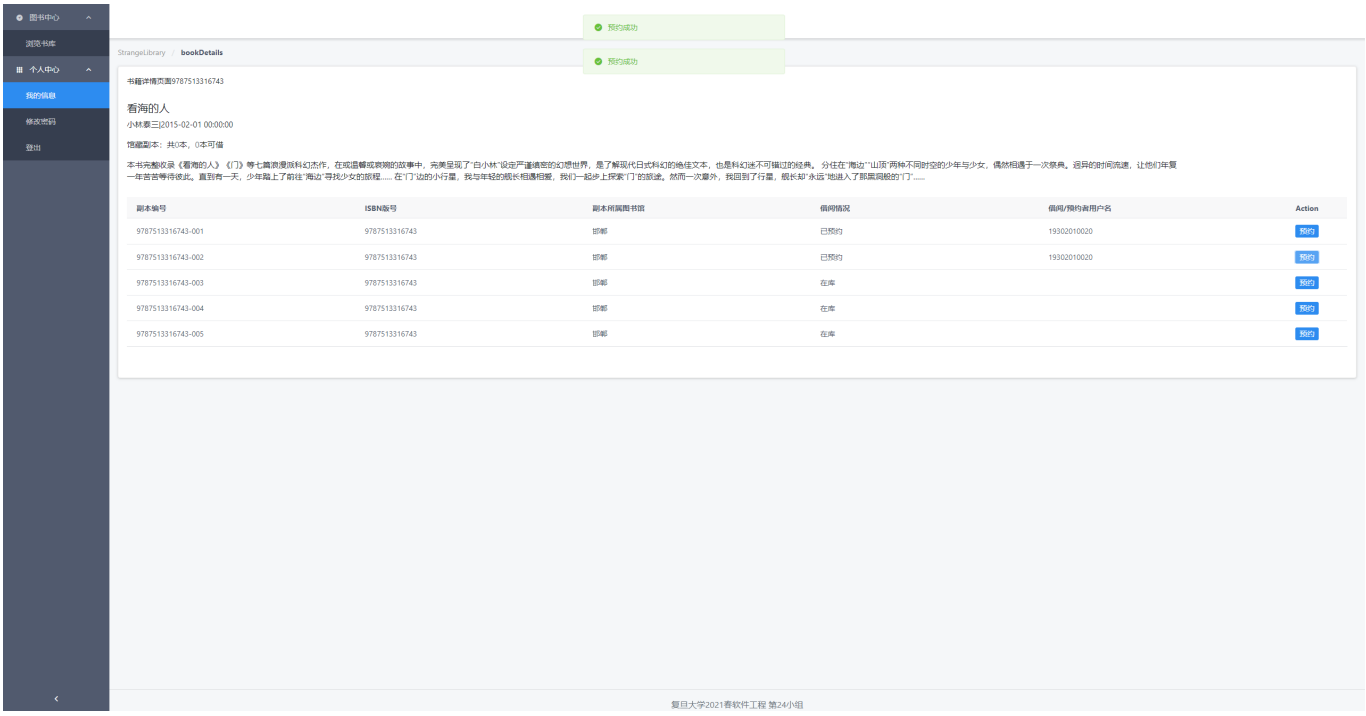

图书中心
浏览书库
个人中心
我的信息
修改密码
登出
StrangeLibrary / user / personal
个人信息
信息名称
具体信息
用户名
19302010020
电子邮箱
19302010020@fudan.edu.cn
权限级别
普通用户
创建时间
2021-05-03 21:14:55
最后登录时间
2021-05-03 21:14:55
我的借阅和预约情况
书籍名称
副本编号
副本所属图书馆
借阅时间
状态
暂无数据
我的借阅历史
书籍名称
副本编号
副本所属图书馆
借阅时间
归还时间
暂无数据
复旦大学2021春软件工程 第24小组

- 搜索书名

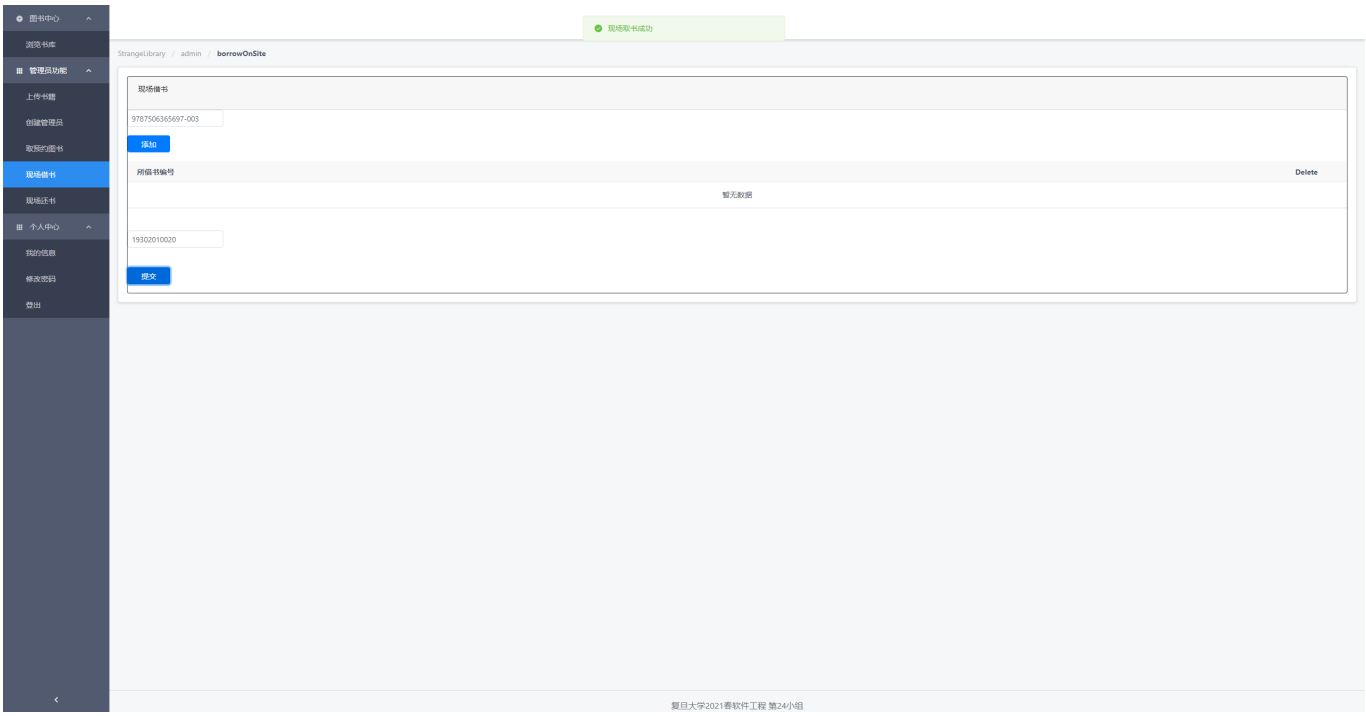

图书中心
浏览书库
个人中心
我的信息
修改密码
登出
StrangeLibrary / browse
浏览书库
搜索
玩具修理者
小林泰三|1996-12-01 00:00:00
馆藏副本：共0本，0本可借
详情
看海的人
小林泰三|2015-02-01 00:00:00
馆藏副本：共0本，0本可借
详情
兄弟
余华|2012-09-01 00:00:00
馆藏副本：共0本，0本可借
详情


图书中心
浏览书库
个人中心
我的信息
修改密码
登出
StrangeLibrary / browse
浏览书库
玩具
搜索
玩具修理者
小林泰三|1996-12-01 00:00:00
馆藏副本：共0本，0本可借
小林泰三「玩具修理者」提供了一個全新的視點，我們正在消弭生物與非生物，虛擬與真實的界線，這樣的科技到底能為人類帶來什麼好處或是負面的影響，值得深思。
詳情
玩具修理者
小林泰三
共 1 条
1
复旦大学2021春软件工程 第24小组

- 预约副本


图书中心
浏览书库
个人中心
我的信息
修改密码
登出
预约成功
预约成功
StrangeLibrary / bookDetails
书籍详情页面9787513316743
看海的人
小林泰三|2015-02-01 00:00:00
馆藏副本：共0本，0本可借
本书完整收录《看海的人》《门》等七篇浪漫派科幻杰作，在或遗憾或救赎的故事中，完美呈现了"白小林"设定严谨缜密的幻想世界，是了解现代日式科幻的绝佳文本，也是科幻迷不可错过的经典。分住在"海边"与"山顶"两种不同时空的少年与少女，偶然相遇于一次祭典。迥异的时间流速，让他们年复一年苦苦等待彼此。直到有一天，少年踏上了前往"海边"寻找少女的旅程……在"门"边的小行星，我与年轻的舰长相遇相爱，我们一起步上探索"门"的旅途。然而一次意外，我回到了行星，舰长却"永远"地进入了那黑洞般的"门"……
副本编号
ISBN版号
副本所属图书馆
借阅情况
借阅/预约者用户名
Action
9787513316743-001
9787513316743
邯郸
已预约
19302010020
预约
9787513316743-002
9787513316743
邯郸
已预约
19302010020
预约
9787513316743-003
9787513316743
邯郸
在库
预约
9787513316743-004
9787513316743
邯郸
在库
预约
9787513316743-005
9787513316743
邯郸
在库
预约
复旦大学2021春软件工程 第24小组

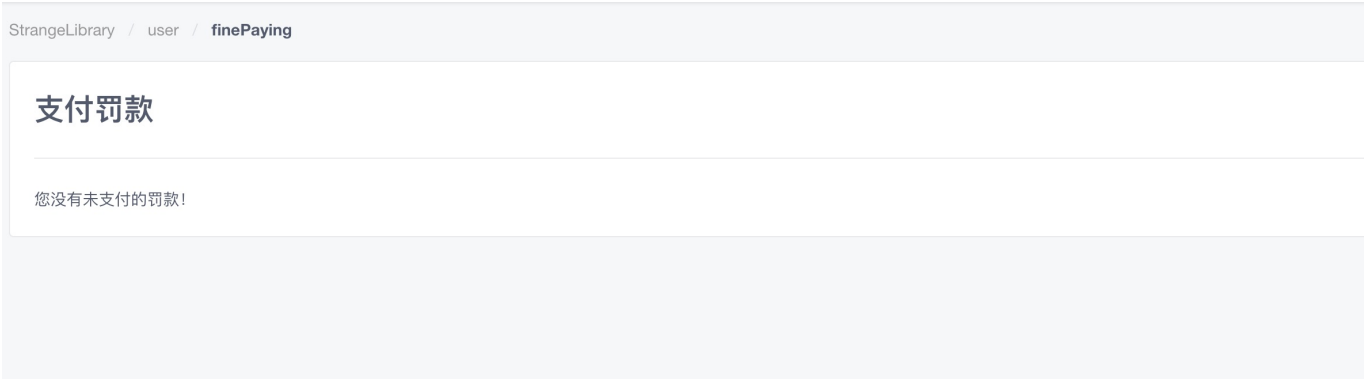
- 借书还书


图书中心
浏览书库
管理员功能
上传书籍
创建管理员
取预约图书
现场借书
现场还书
个人中心
我的信息
修改密码
登出
现场取书成功
StrangeLibrary / admin / borrowOnSite
现场借书
9787506365697-003
添加
所借书编号
Delete
暂无数据
19302010020
提交
复旦大学2021春软件工程 第24小组

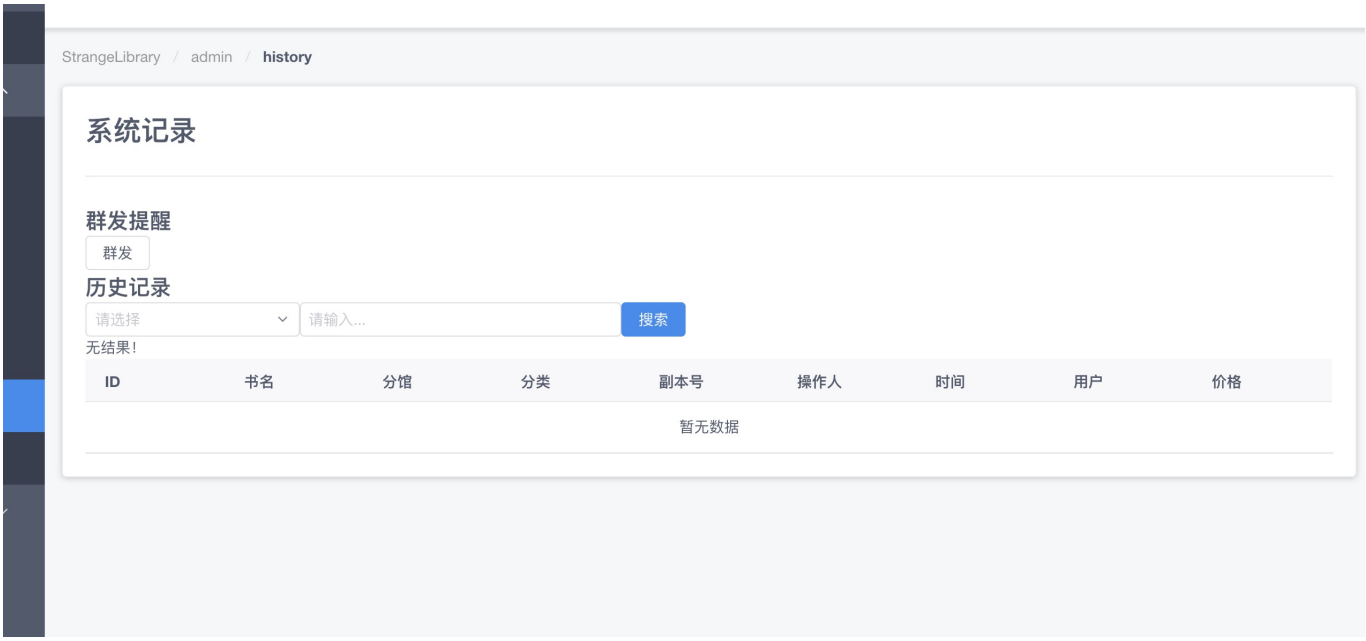
- 用户规则


图书中心
浏览书库
管理员功能
上传书籍
创建管理员
取预约图书
现场借书
现场还书
系统记录
用户规则
个人中心
用户规则
教师规则
* 最大借书数目 20
* 借书有效期/天 60
* 预约有效期/天 20
本科生规则
* 最大借书数目 5
* 借书有效期/天 30
* 预约有效期/天 10
研究生规则
* 最大借书数目 10
* 借书有效期/天 30
* 预约有效期/天 10

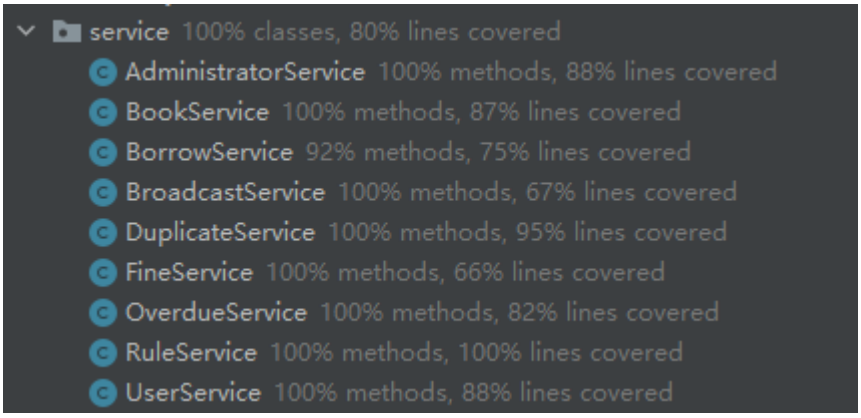
- 罚款界面



- 系统记录



单元测试覆盖率



小组成员个人实验报告

张明 183000200008

负责任务

（1）微服务服务拆分

（2）UML图的绘制

（3）部分后端的工作

服务拆分

我们原本做好服务拆分的框架设计之后就打算开始先去拆分原本的服务。我负责拆分了原来代码中bookservice中的部分。在后面的搭建中，我们首先是参考了`eureka-example`，但是在尝试过程中虽然server可以成功启动，但是provider总是无法启动，或者启动之后无法访问到里面的服务，然后又参考了`spring-cloud-demo`,同样也是有一个服务无法启动，虽然修改代码之后可以正常启动了，但是服务之间的数据通讯依旧是出现问题，往往是拿不到需要的数据的。相关docker框架的设计也就无法继续。 所以到最后决定先暂时放弃微服务的框架，完成要求实现的功能。

UML图的绘制与后端的工作

负责绘制本次实验的UML图。在我们初次尝试微服务的拆分时，我根据不同的service进行了uml的绘制，但是在经过重大战略决策（指暂时放弃微服务的搭建，用原来的方法实现lab的功能）后，我将uml图进行了修改，比较清楚地表达了我们项目目前service-domain-conrtoller-repoisitory之间的层级关系。这也为我们之后进行微服务的搭建做了准备，因为我们能通过目前的uml图较为清晰地看到耦合程度比较高的服务。在后端中，我主要实现了通过注册功能判断学生类别并赋上identity属性的逻辑，直接通过注册时的后端的操作，就可以给每个普通用户身份的认证，方便后面馆规的设计。

袁逸聪 19302010020

工作内容

1. 新增功能架构设计
2. 所有新增的单元测试
3. 微服务框架尝试
4. 后端除罚金、预约超时检查外所有新增业务的实现
5. 构架改动下旧有业务及单元测试的所有重构工作

新增功能架构设计

Lab4中数据表架构发生较大变化：

原本借阅被当做一个流程对待：生于预约或直接借书，止于归还，最后所有借阅条目都将停止在“归还”状态。这与系统记录是相悖的，因为预约和借出的记录终将被覆盖。

因而我们拓展了副本表，更充分的信息能够表达出副本当下的状态，与一本真实书目的语义相符合。而原本的预约表则被改造成了系统记录表，保留了其原有的信息丰富全面之特点，但不再参与业务逻辑，仅做历史记录之用。

新增功能也需要一些设计来实现，譬如新增的罚金表和逾期预约表，实际上相当于两个消息栈。当罚金产生或检测到预约过期的时候就往表中添加消息，直到改消息得到了需要的处理再将其清除。

单元测试

之前的单元测试往往只追求方法覆盖率，但对各种情况考虑得较少

本次新增的单元测试多考虑了些不同情况的测试，以求提高代码覆盖率(即让测试用例能够触发尽可能多的代码，确保写下的代码在运行是不会出问题)

但web中的单元测试仍是意见比较麻烦的事，包括在数据库中模拟数据的步骤较为繁琐、使用流程中靠后的方法很难独立快速测试等困难

微服务框架尝试

本次Lab原本进行了拆分出9个服务的设计，但实际执行中遇到很大困难：spring家族的配置错综复杂，当服务多起来就很难分辨哪些配置对哪些服务是必须的。同时我们所找到的微服务demo在运行时也会出现比较多的问题，包括无法挂起、或有时能挂起而有时就不行……

不过从尝试的经验中，我们初步掌握了docker-compose的用法

本次Lab将前端、后端、数据库拆分开部署在了3个docker容器中，以docker-compose启动。但后端内部更细的拆分还需要帮助

许同樵 19302010015

负责任务

所有前端开发工作

前端页面的重构

为了保证前端的简洁性，将bootstrap残留的部分class删除。 页面的结构布置时，将导航栏单独存在一层，而不是存到App的层中。这种情况下就可以在管理员选择分馆时不使用导航栏，以免管理员直接通过导航栏跳转而引发错误。

冷却时间实现

发送验证码邮件需要冷却时间，通过setInterval和clearInterval实现。注意内部操作函数需用箭头符号而非回调函数实现，否则无法改变外部的变量。

```
let interval = setInterval(()=>{
this.time= --i;
if(i===0) {
 this.canSendCaptcha = true;
 clearInterval(interval);
}
},1000);
```

接口

一般get接口可在后端使用restful API格式，通过在最后加参数来实现。因此前端直接动态改变API即可。使用这种方式有时可以大大简化api，例如只有一个参数需要传的时候。

关于美化和用户友好问题

（1）在还书列表中，在输入框后选择状态，直接添加入表格。又例如还书失败时将失败的条目留在表格中，将成功还好的书删除。

（2）部分内容无需显示的就隐去。在个人中心中，如果没有逾期预约的书，就不会在个人中心显示这一条。

（3）导航栏过长可能被挡住，小改

（4）表格中的数据需进行处理

王海伟 19302010011

负责任务

（1）微服务框架搭建

（2）后端罚金功能以及预约超时检查

（3）代码质量优化

微服务框架

我们原本做好服务拆分的框架设计之后就打算开始先去拆分原本的服务。

首先是参考了`eureka-example`，但是在尝试过程中虽然server可以成功启动，但是provider总是无法启动，或者启动之后无法访问到里面的服务，然后又参考了`spring-cloud-demo`,同样也是有一个服务无法启动，虽然修改代码之后可以正常启动了，但是服务之间的数据通讯依旧是出现问题，往往是拿不到需要的数据的。相关docker框架的设计也就无法继续。

所以到最后决定先暂时放弃微服务的框架，完成要求实现的功能。

后端罚金功能以及预约超时检查

两者的思路大致是相同的，通过新建相关的库表来记录相关的数据，然后在需要的时候把记录放到存放所有记录的表当中。

罚金的计算和产生是发生在用户还书的操作上，需要前端向后端多发送一个书本状态的数据，来判断是完好，遗失，或者损坏，然后根据不同的情况来计算和产生罚金，并进入记录罚金的表单，然后更新副本状态为在库，损坏，或者遗失，随后把需要的记录存放在记录的数据库中。当用户付清罚金之后记录会被删除，同时也会更具是否有现存的罚金记录来限制用户的预约和借书。

预约超时检查发生在书本浏览页面的请求接口当中，回去获取所有的预约记录，并通过记录的有效时间来计算当前预约是否过期，如果过期则在记录过期的表中增加一条记录，在这里取消用户的预约，把副本状态调整为在库，并生成记录放在数据库中。当管理员向用户发送邮件提醒的时候或者用户主动确认，这条预约过期的记录会被删除，但是保留记录。

后续的准备

准备实现一个微服务框架然后拆分，要在下一个lab中利用spring-cloud实现微服务的框架，以及搭建好docker的自动化部署和相关的自动化编译脚本。同时处理好服务之间的解耦，便于服务拆分。