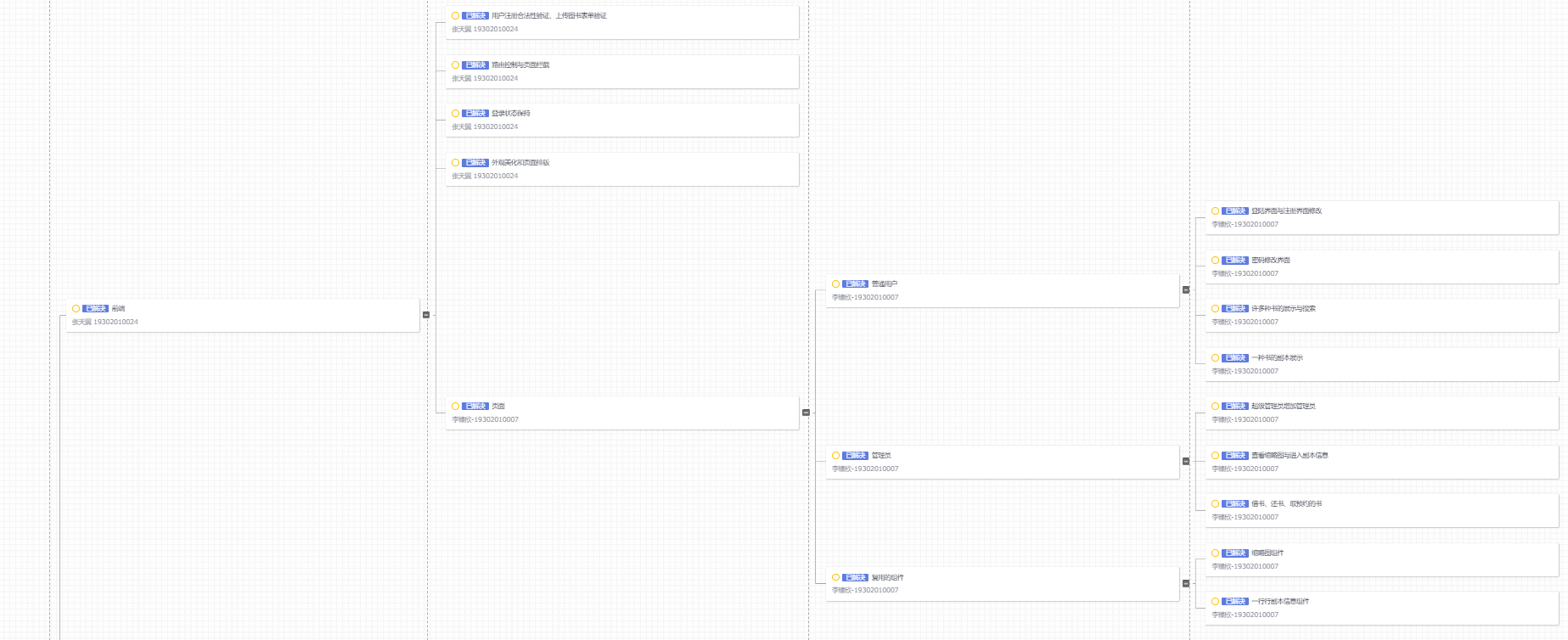
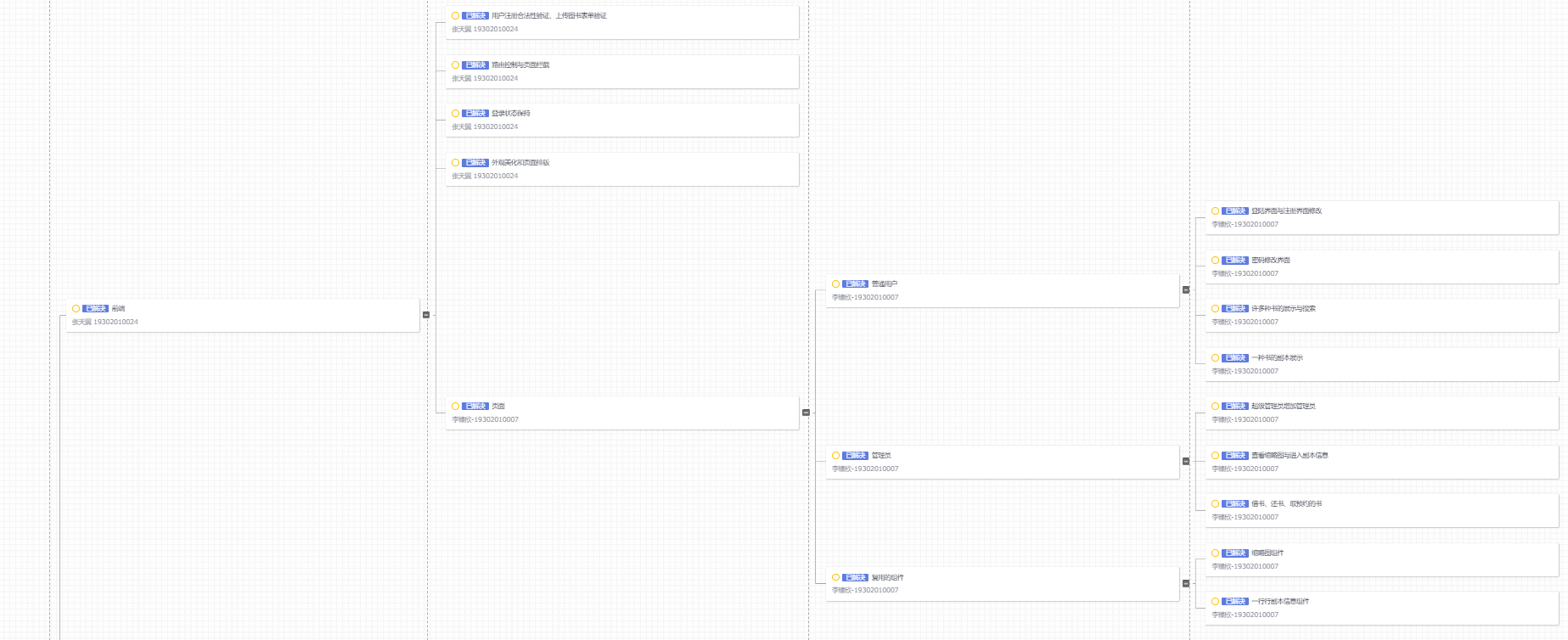
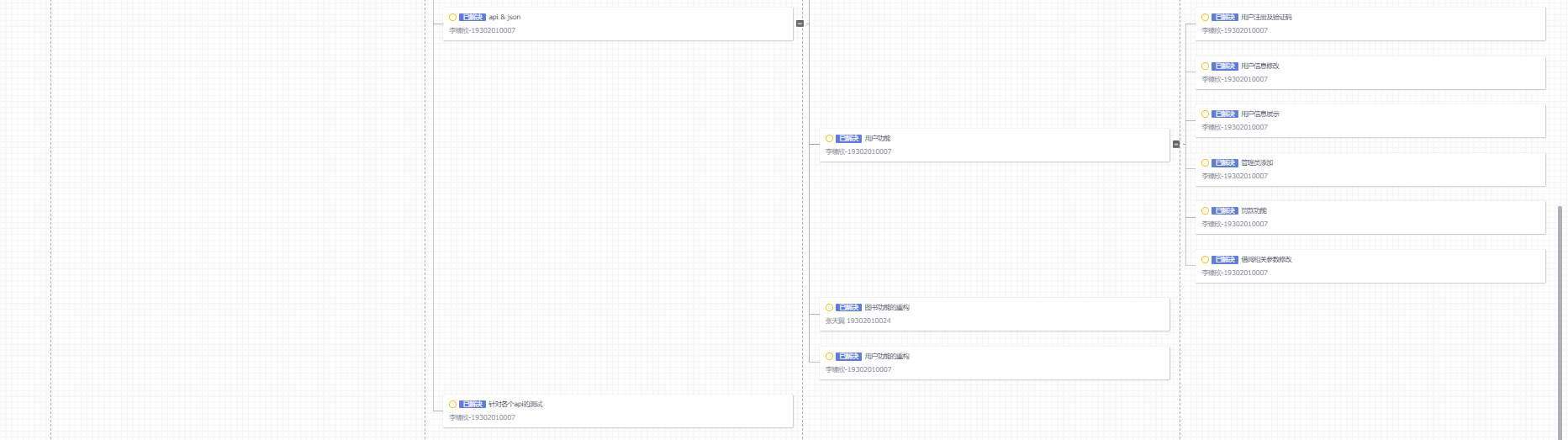
17组Lab4实验报告

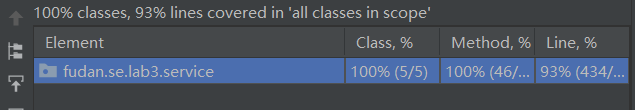
超级管理员账号及密码：admin Di17zufuwuqi

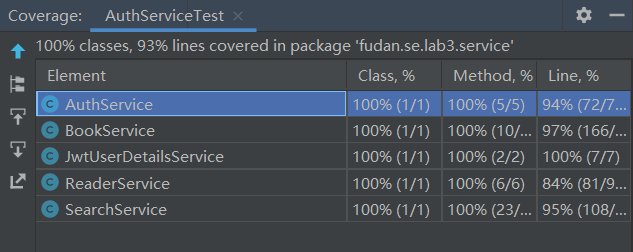
1. 项目需求规划：





二、单元测试及代码覆盖率检查结果截图：





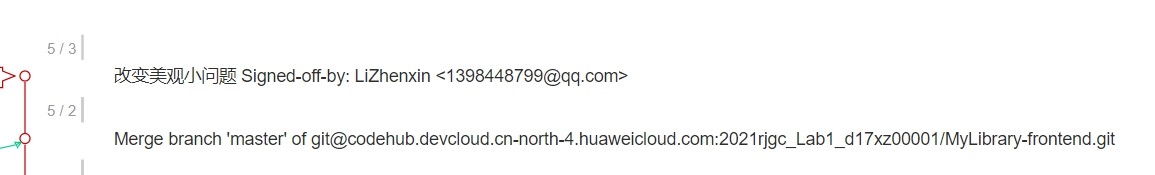
1. 代码检查结果截图：



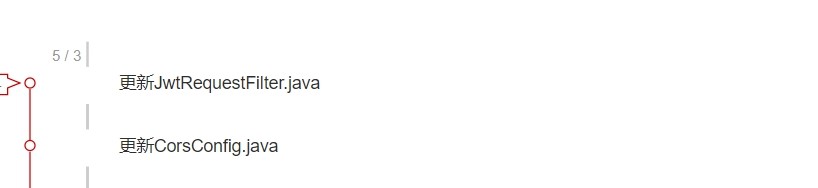


1. lab3截止截图：

前端：



后端：



1. lab4截止截图：



六、项目页面截图：

1. 管理员自己的记录



2. 管理员查看某一个用户



4. 管理员点击他的两个按钮，一个是完成事务，一个是未完成的

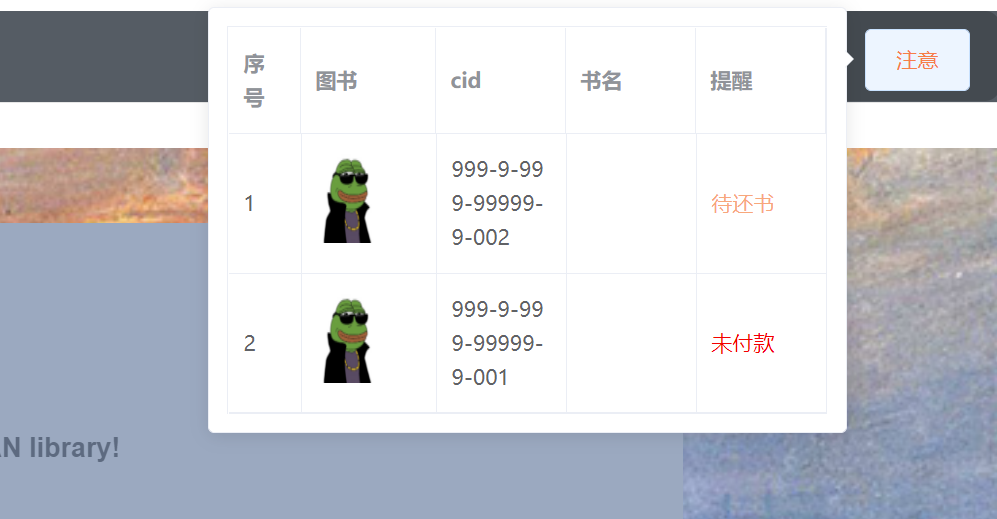




5. 超管新增两个界面

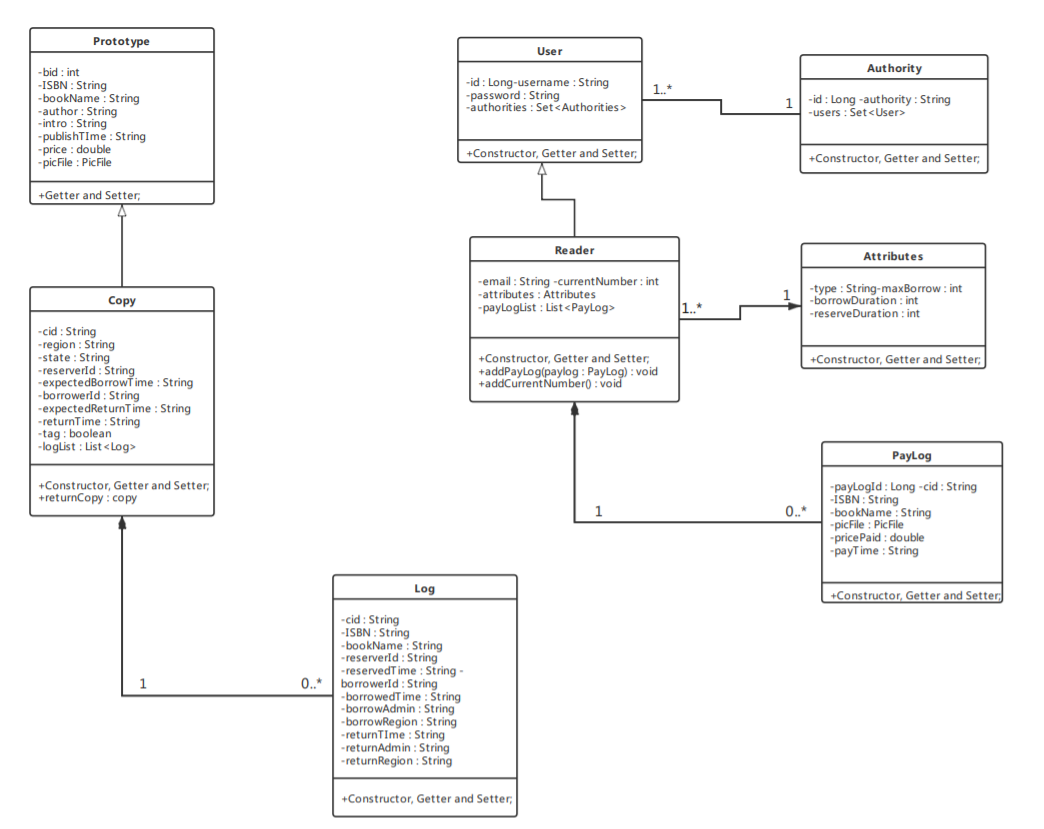


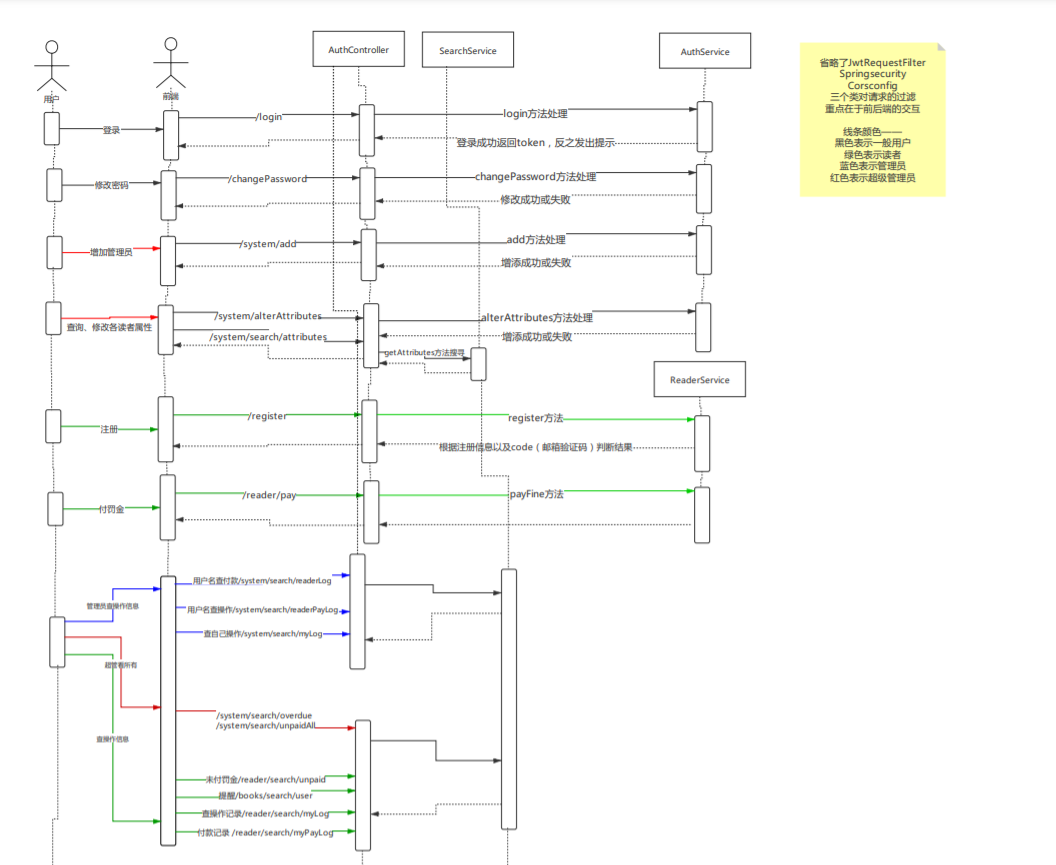
6. 用户查看提示信息和自己的未罚款、已罚款、操作记录

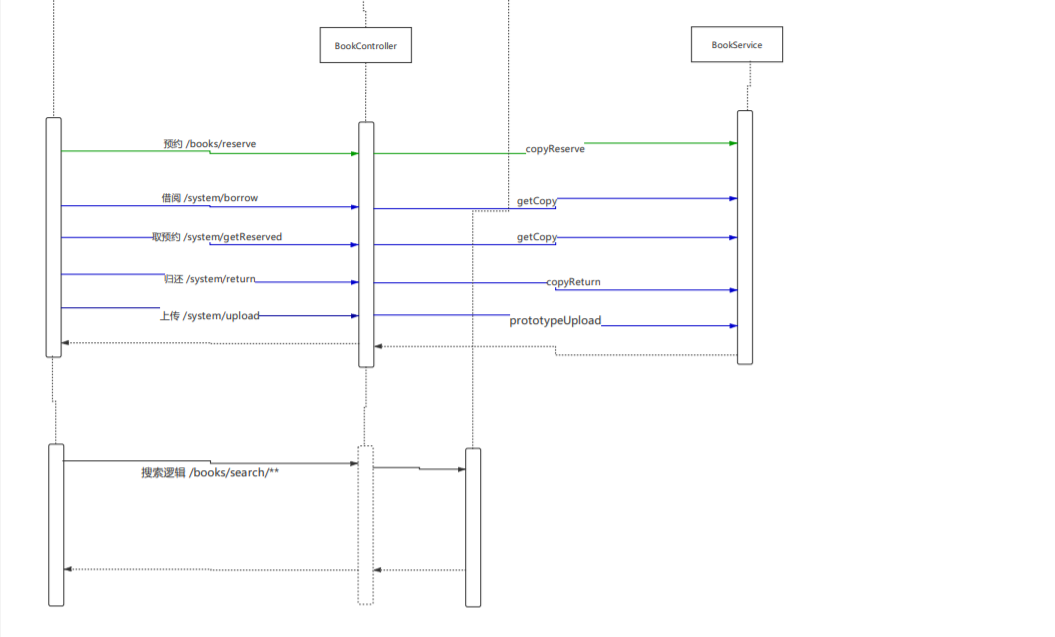


七、软件设计文档：

分成了类图和时序图，类图给了一些主要的domain，忽略了如图片类这些便于数据库操作的信息。主要可以分成书本与用户，在图片两侧。时序图主要便于前后端交互协作，明确一些事务以及对应的接口，因为接口已经有些过多了。







八、项目使用指南：

1. 管理员相关

1） 在登陆界面加入了提示，即超级管理员的账号和密码

admin

Di17zufuwuqi。

2） 缩小了导航栏，左侧大致可分为三个部分：书本操作（取预约、借阅、归还，与上个lab类似）、记录查询（自己的、用户的）、超管操作（增添管理员、调整读者属性、发送邮件提醒）。

3） 在用户的记录查询部分，先查询用户，根据用户名精确搜索或者根据读者类型筛选，点击缩略图下的两个按钮进入两个界面——一个是历来记录，分成书本操作与付款操作，另一个是正进行中的事务，即未预约、未归还、未付款。

2. 读者相关

1） 在用户界面，导航栏的左上角是给用户的提示消息，提示用户当前正在进行中的事务（未取预约、未归还书本、未付罚款）三个条目。

2） 在个人中心中加入了两个表格，总共有三个表格。从上往下依次为，付罚款、查付款记录、查借阅记录。

九、组员任务分配：

张天翼：图书类的重构，图书方法和搜索方法的部分重写，借阅期限设置和检查功能的实现

李臻欣：用户类的重构，用户方法和搜索方法的部分重写，发送邮件及罚款功能的实现，罚款页面的实现，前端管理员页面，部署

陈敏：前端用户相关界面，增加注册时邮箱验证环节，增加两个主要组件，提示用户信息组件与用户罚款、罚款记录组件

叶雨昂：后端测试，在原先测试类的基础上，针对重构后新增的SearchService与ReaderService的测试。

十、小组实验过程记录：

1.Lab4 项目需求的分析与讨论：

在对需求进行充分讨论之后，小组确立了此次 lab 所需要实现的各个功能，对需求中可能产生歧义与不同理解的方面进行统一并建立了思维导图便于组员查阅需求。由于lab4中需要对后端进行大量的重构，难免会有部分接口的改变，因此小组对接口进行了重新约定。

2.Devcloud仓库管理：

完成任务分配之后，小组成员根据自己的任务完成相关代码的编写，并在上传时标注新增、修改和删除的内容。由于在任务分工时对组员需要修改的文件做好了划分，避免了修改相同文件产生冲突的情况。本次lab小组采用直接在仓库中写代码的方式，一方面便于提交和合并，另一方面也避免了复制黏贴带来的文件未能成功删除的问题。

3.后端的重构：

本次lab4中后端的重构主要分为三部分，图书类、用户类、操作记录类以及搜索方法的重构。图书类将原Book类拆分成Prototype类与Copy类、用户类将原User类拆分为User类与Reader类，Copy与Reader分别继承Prototype和User，不仅将原型和副本、管理员和读者做出了明确区分，也减少了图书和用户方法的圈复杂度和认知复杂度。操作记录类从原来图书属性中独立，变为Log类，与Copy和Reader建立映射关系。搜索方法则将图书和用户的所有搜索方法全部整合如SearchService中，提高了代码的可读性以及测试对搜索逻辑的覆盖率。

4.代码的测试

由于后端的重构，lab4中的Service类增加到了五个。经过大量的测试和修改，最终让覆盖率达到90%以上，大部分测试类几乎覆盖了所有情况。

十一、小组成员实验总结：

张天翼：

图书类的重构：

在之前的lab中一直将所有图书对在一个类中，但在lab3中已经可以看出将原型和副本视为一个类的不妥之处。因此在lab4中进行了重构，分类Prototype和Copy两个类，原型存放书籍的基本信息，副本则包含了借阅状态和借阅记录相关信息。不仅降低了理解的难度，也让BookService中的方法变得简单和明确了许多。

搜索方法的部分重写：

原lab中Book类的搜索方法是直接写在Controller中的，这种写法是不利于测试的，也不利于功能的模块化。由于lab4中的搜索逻辑增多了许多，因此将图书类搜索方法统一进行了重构，写入SearchService中，并在测试类中对相关功能进行了测试。

借阅期限设置的实现：

由于lab4中需要增加过期判断的功能，在借阅信息日期这一栏中使用了Date类便于日期的比较和计算。在查询过期时只要根据Copy中的预约、借阅相关时间和当前时间比较，并发送邮件即可。

陈敏：

在lab4里我主要参与对前端用户逻辑的新增部分。

在注册页面新增对邮箱验证码的功能，向后端发送两次请求，第一次发送信息，提示填写验证码。第二次发送附加验证按。

在用户信息展示页面，新增展示用户的罚款记录同时增加提交罚款的组件。在用户信息展示页面从原来的展示借阅信息，新增两个与罚款相关的信息。同时在主页面新增用户的提示信息，当用户登录以后可以点击跳出弹框展示对用户预约、借阅和罚款的提示信息。

李臻欣：

在lab4里，我主要写了前端的管理员界面以及后端用户相关的逻辑，参与并协助了后端用户部分的重构。

之所以要重构是因为新需求决定了读者与管理员之间有一些属性必定不能共享，比如邮箱、最大借阅量等与读者相关的属性。于是决定建立了Reader类继承User类，前者与一个Attributes类进行关联。这个Attributes类存放了读者的身份，比如教师、本科生、研究生的类型标识，还有读者在数值上的属性，即最大借阅量和两个超期。以此做到将读者身份的属性和读者作为一个用户本身区分开来。同时加入一个PayLog类，用以记录读者的付款记录，完成了这块的逻辑。在我们的设计中，书本操作信息（借阅、归还、预约等）的Log和一个图书副本进行关联，而付款记录仅仅和一个读者个人相关联。付款记录和操作记录是分开存放的。

除此之外，规范、删去了一些Controller中的接口，并且统一用ResponseEntity进行封装，解决了接口混乱，意义不清的问题。

在前端中，规范并区分了admin文件夹中的文件，分成了三个主要的部分，即书籍相关操作、读者记录查询（在这一部分中，加入了管理员搜索读者的逻辑，便于查看某一个指定用户的相关操作）以及超级管理员的三个操作（增加管理员、更改指定用户类型的数值、查询问题书本、问题用户的总体信息并发统一邮件提醒）。随后根据调整后的Controller接口更新了所有前端的请求地址以及封装后的JSON属性。重构之后更改接口耗费了比较多的精力，所以开发前还是要先约定好接口。整合了前端的用户部分，经过反复测试得到一个较为完整的版本。

叶雨昂：

我在Lab4的工作内容与lab3无异，苦于icslab的折磨，这次我并未能完成本来划分给我的前端+测试工作，工作量不大，除了新功能需要在@Autowired下初始化新内容外，没有需要赘述的。