



**QG工作室项目报告**

学 院 计算机学院

专 业 计算机类

班 级 12班

组 别 后台组

姓 名 吴卓颖

学 号 3122004791

2023年 5月 10日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc134863854)

[2. 设计思路 1](#_Toc134863855)

[3. 详细功能设计 2](#_Toc134863856)

[4. 程序测试 4](#_Toc134863857)

[5. 项目亮点 8](#_Toc134863858)

[6. 训练营难点 13](#_Toc134863859)

[7. 心得体会 16](#_Toc134863860)

# 项目简介

项目背景:

某一企业有许多打印机，打印机们每秒都会产生一条**设备信息并**写入到一**txt**文件中。

项目基本需求:

设计一个系统，能够读取txt文件中的消息，并实时展示给前端，供管理人员时刻观察设备状态，并能提供一个实时预警效果。

# 设计思路

项目分析:

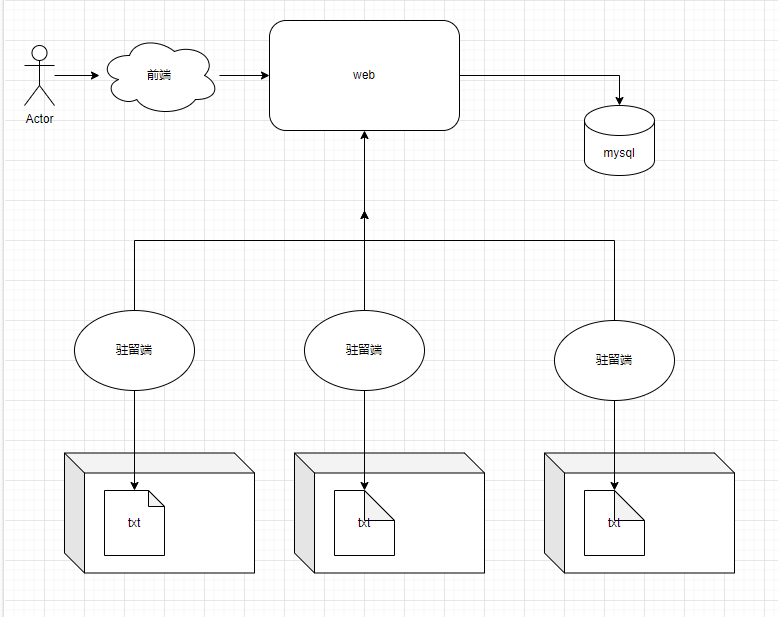
整个项目只是一个"查看系统"，需要程序读取的信息已经由由机器产生,程序需要做的**只是读取信息分析后显示到网页上**和一些其它用户模块功能。**产生数据**的规律为:**每秒**输出信息一条到**txt文件**。

简单来说就是要实现监听、读取新增信息并解析，再执行录入MySQL数据库和响应前端请求等等功能。此外还需要设计一个包含管理员、企业用户、普通用户三个角色和登录、注册、修改查看用户信息等基本功能的一个用户模块。

项目设计结构：

架构分为**web服务端（处理数据）和驻留端（接收数据）**两层。驻留端接收消息后，只需要把消息发送到web服务端。这样做减少了打印机客户端的负担。

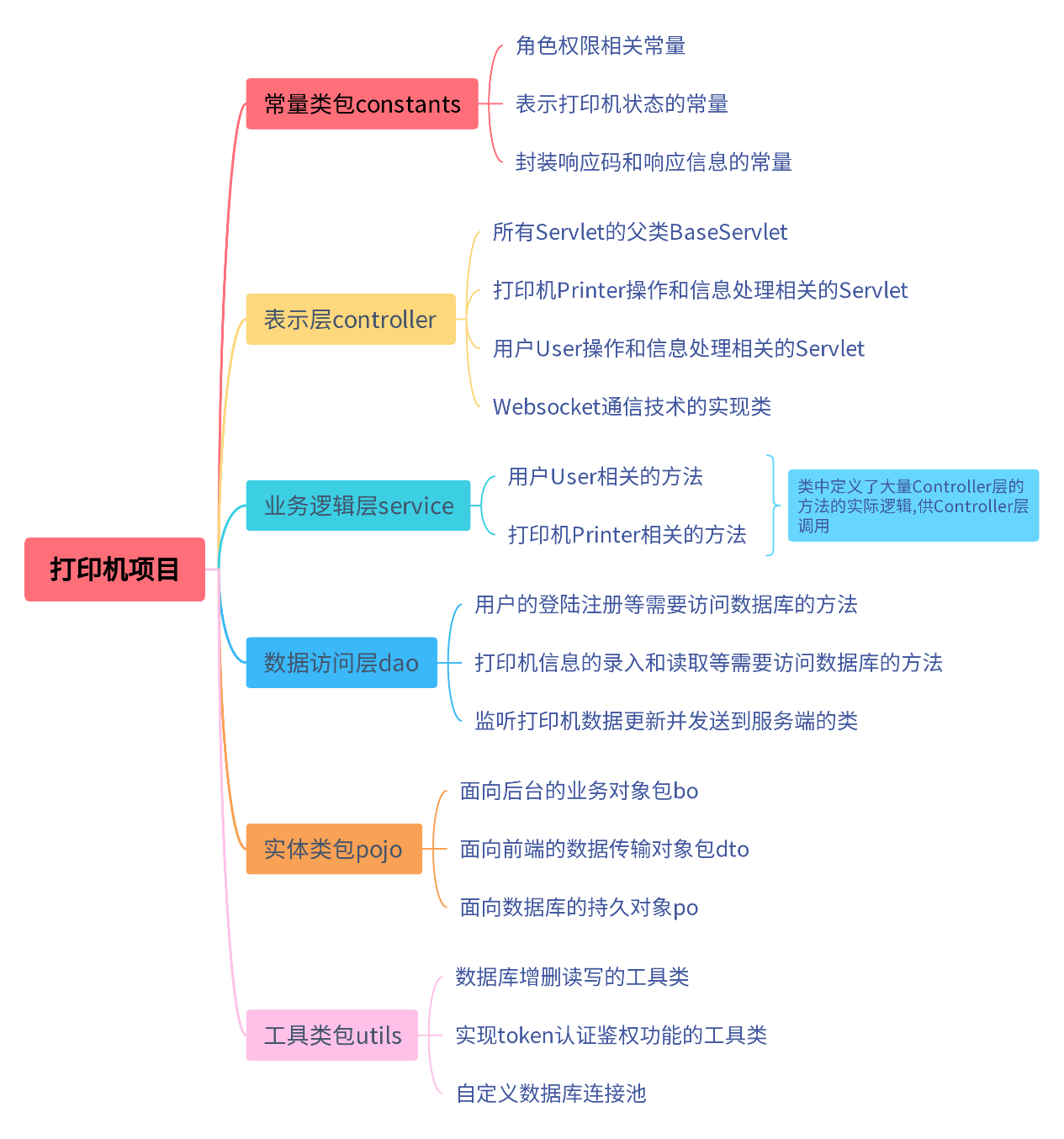
Web端接收驻留端发送的数据，处理筛选后按需求进行数据的持久化和响应前端请求。



# 详细功能设计

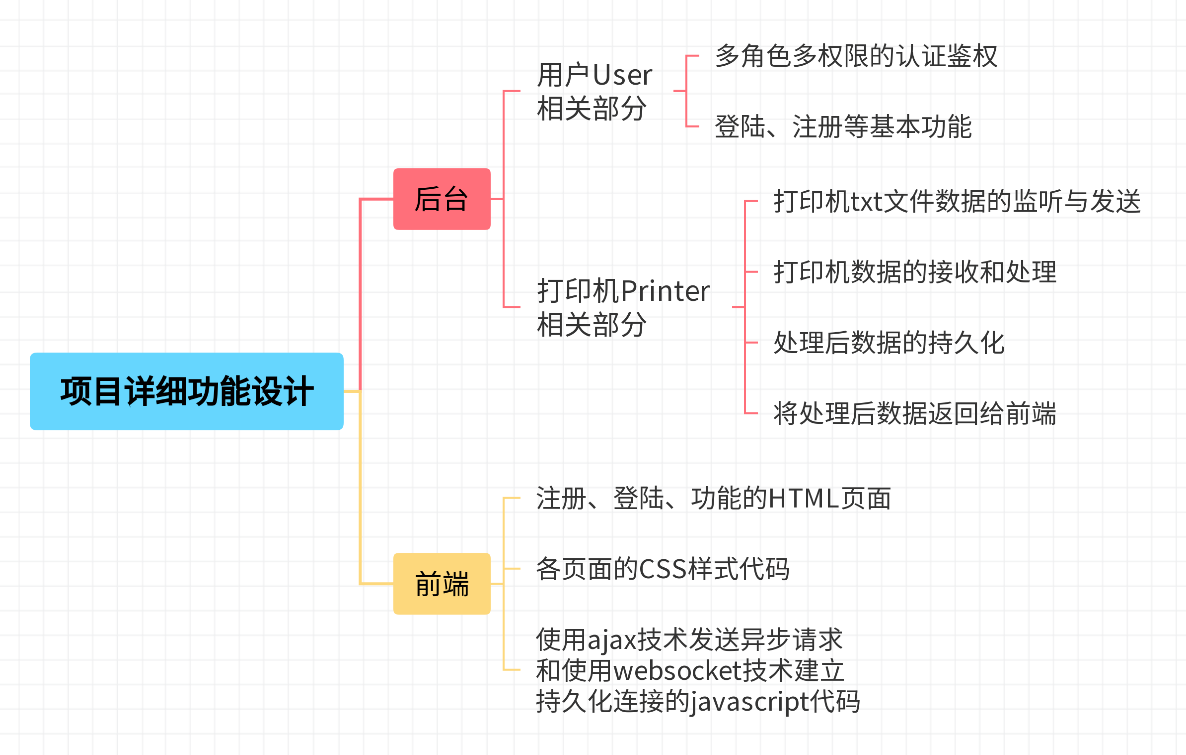
项目详细功能结构设计:

本项目遵循MVC设计规范，将代码分为了**表示层**(controller)、**业务逻辑层**(service)、**数据访问层**(dao)三层。此外还有常量类和工具类等**辅助模块，**实体类包pojo下针对数据对象的不同功能进行了进一步的细分。



项目详细功能设计:

具体功能设计部分，我将我的项目分为前端部分和后台部分两个大的部分。而后台部分又可再细分问用户User相关的功能和打印机Printer相关的功能。



# 程序测试

注册页面:



登陆页面:

登陆成功和失败都会有相应的弹窗提示，若登陆成功则会进一步跳转到用户大厅。



大厅页面：

进度大厅后可以点击上方的”个人中心”查看修改自己的个人信息，点击”查看可管理的打印机”则可以显示自己有权限查看的打印机名字，点击名字即可跳转。



个人中心页面:





普通用户需要绑定企业个人信息里的绑定码和企业账号名才能查看打印机

企业用户需要设置绑定码才能被绑定，且无企业账号名输入框(被隐藏)

管理员用户的此页面则无上述输入框(被隐藏)

点击上方的”首页”可返回大厅。

打印机详细信息页面:

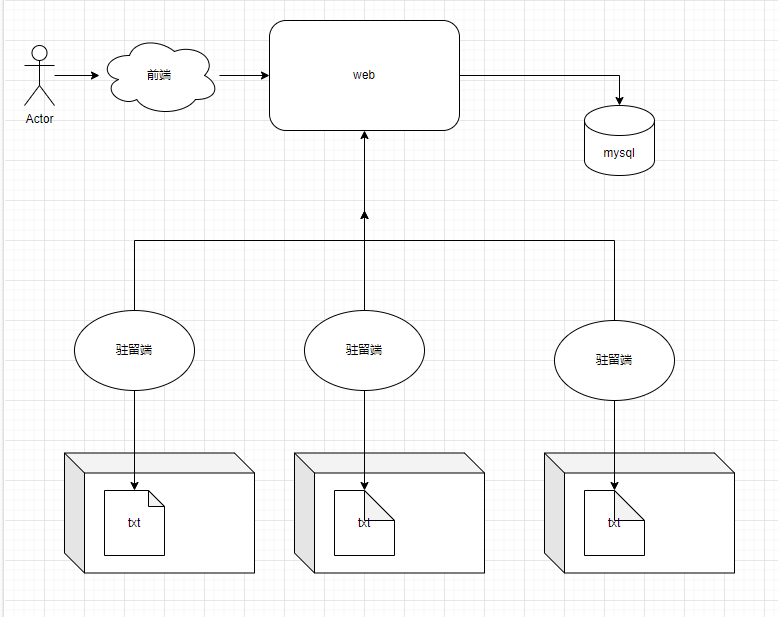


若后台监听线程未启动且打印机也未开始生产信息，则刚进入此页面时不会显示数据。当后台监听线程和打印机均启动时，分析后的打印机txt文件中的数据就会通过websocket连接广播到前端。

点击”启动后台监听线程”即可开始监听某一台打印机，当某台打印机已经被监听时，点击此按钮不会再在后台新建线程。

# 项目亮点

**1.**驻留端和服务端的分离

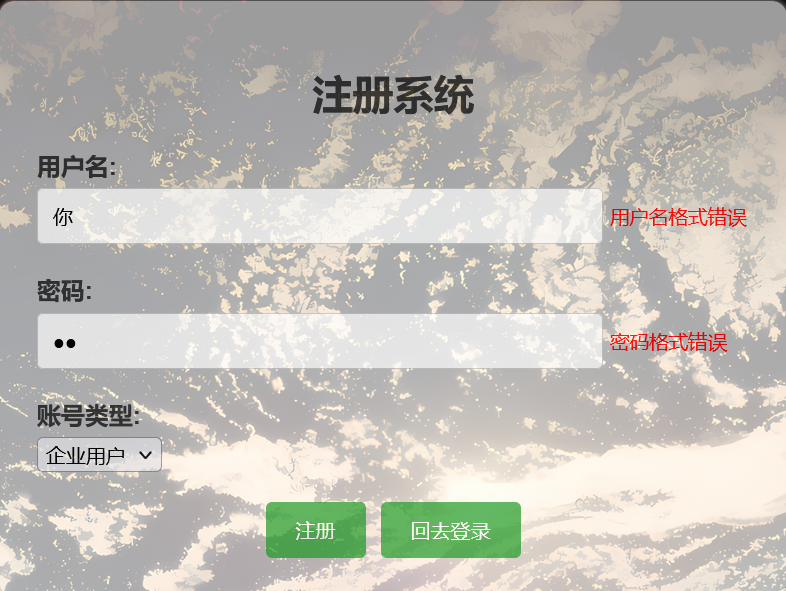


在我的项目中,我实现了数据**驻留端**和**服务端（处理端）**的分离。并且支持**同时处理**多个驻留端的情况, 即可以同时处理多个打印机同时向服务端发送信息。此实现较为贴近现实中的生活生产需求。

**驻留端**: 主要由监听类对象构成, 负责**读取**信息和**发送**至web端

**处理端**: 主要由web端中service层和controller层构成, 负责**接收**和**处理**驻留端和前端发送的信息

**2.**使用正则表达式进行输入校验



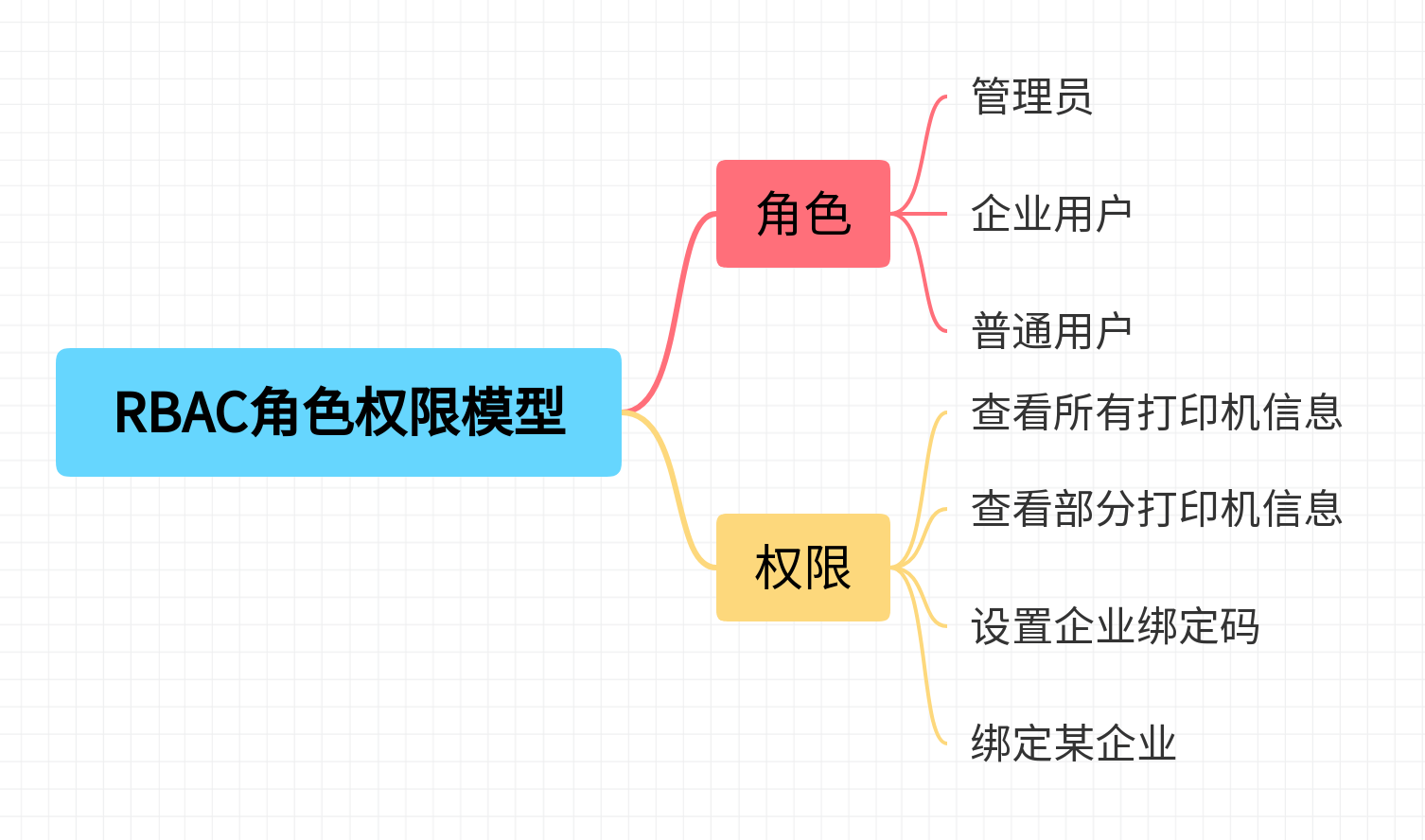
在我的前端的注册页面中, 我使用了**正则表达式**来实时检测用户注册的用户名和密码的格式是否符合要求。若不符合要求,则会有红色警示语,且注册按钮失效,无法注册。

而在登陆界面，若用户没有正常输入账号密码,也会有提示且无法正常登陆



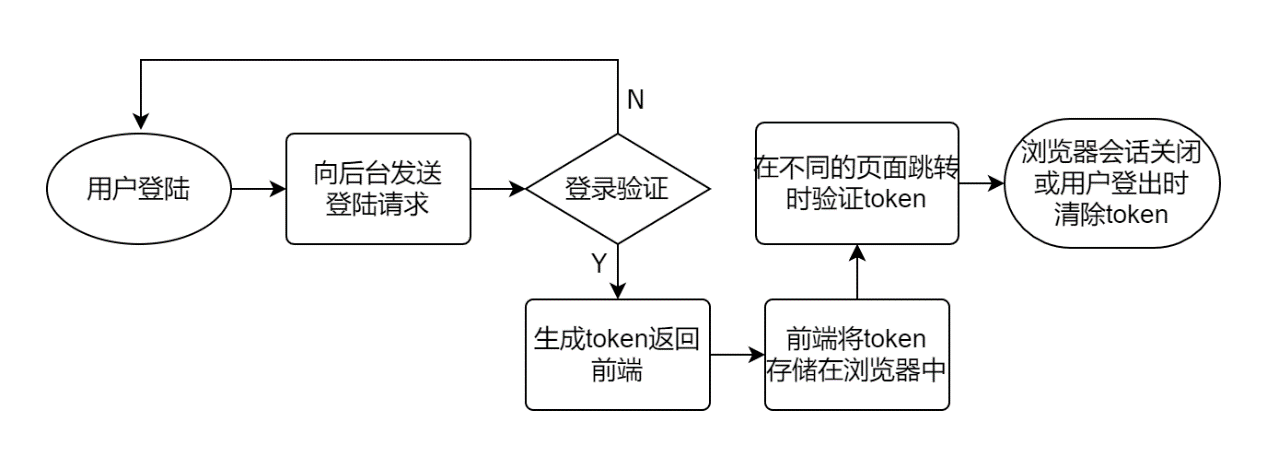
**3.**使用RBAC角色权限模型和token进行认证鉴权

RBAC角色权限模型:



每个用户都会至少有一个角色，每个角色拥有不同的权限。运用此模型能较为方便有效地实现项目中多角色多权限的需求。

Token认证鉴权:

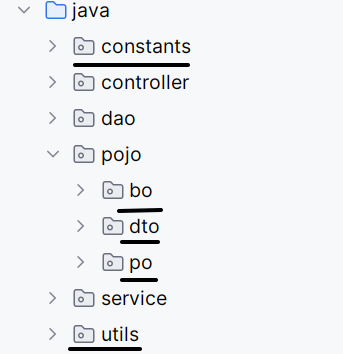


后台代码的Utils包中我设计了一个实现token的生成、解析、验证的工具类。保证相应角色只能查看其权限允许的内容, 通过修改url强行访问会被遣返至登陆页面。Token的使用流程如上。

**4.**项目分包结构的优化

我的后台代码在遵循MVC设计规范的基础上。还进一步设计了常量类包和工具类包等分包。在实体类包又进行了细分：**DTO**(数据传输对象,主要面向前端)、**PO**(持久对象,主要面向数据库)、**BO**(业务对象,主要面向前后端和后端内的交互)。

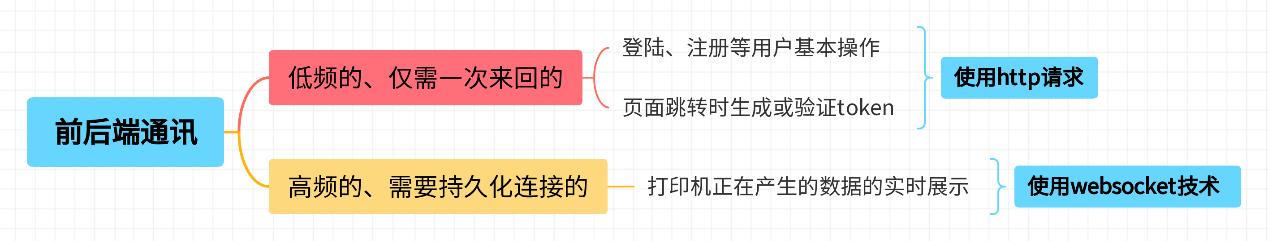
以下为项目分包目录截图:



**5.**使用Websocket技术优化前后端的通信

对于前端中需要与后台进行高频通信的部分，使用http请求的网络开销较大，且后台无法主动与前端进行通信。而我在项目中使用的websocket技术可以很好地避免这两个问题。

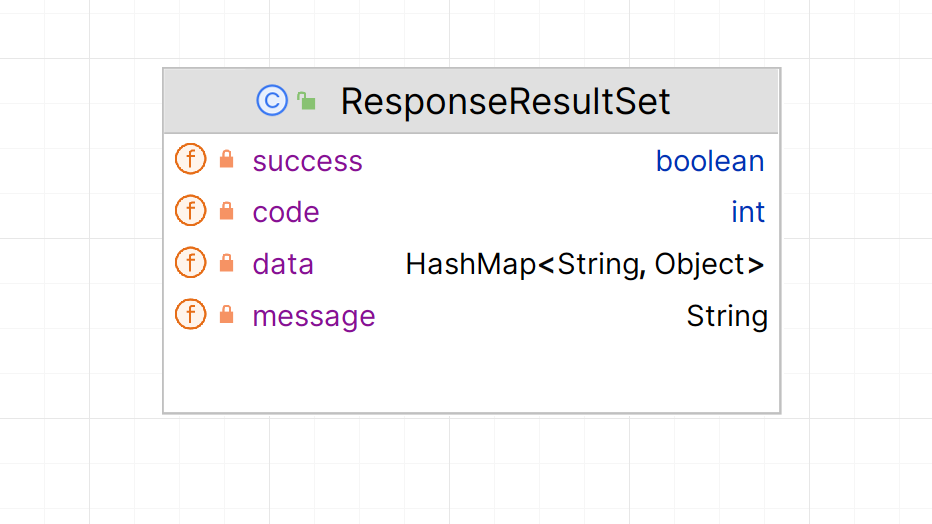
Websocket可以实现客户端和服务器之间的双向数据传输，在客户端创建一次websocket连接后就可以一直接受来自后台的消息。因为打印机产生信息的频率为每秒一条，信息密度较高，所以我使用了websocket技术来向前端发送处理过后的打印机信息。



**6.**统一的结果集处理

在本项目中，前后端信息的传递都是以JSON的格式来进行。所以我不仅在后台中封装了处理和转换JSON字符串的方法，还在pojo包下的dto包下设计了一个名为ResponseResultSet的统一的结果集处理的类，它负责包装需要发送给前端的信息和响应的状态码。响应前端时则直接发送包装有信息的此类对象。

ResponseResultSet类中定义的变量如下:

****

此外，此类中还实现了若干返回值为对象自身的方法，便于使用者使用链式编程操作此对象。

# 6.训练营难点

1、训练营知识学习和训练营作业进度

由于在训练营开始前，我自己学习的进度较慢，所以前四周的训练营一直都是在一边学习相关知识一边完成相关作业。导致每周的小组作业几乎都是卡着截止时间提交。但万幸大组作业中的数据结构我在上半学期学习acm相关知识的时候就已经接触过，完成大组作业只需思考如何用C语言去实现即可。

2、训练营内容和学校课程的平衡

因为训练营中大量的知识需要我利用课外的时间去学习积累，导致我一开始无法很好分配训练营和学校课程的时间精力上的平衡，甚至一度出现了跟不上高数课上老师所讲内容的情况。但好在随着体测的结束，我得以空出一部分时间重新分配，再加上周末两天泡在图书馆里的时间，在最后还是平衡好了训练营和学校课程。

3、实现二叉查找树的非递归版后序遍历

在第四周的大组作业中，我们需要实现一个二叉查找树的ADT并分别实现二叉查找树的三序遍历的递归和非递归版。递归版本非常简单，都只需要三行代码，但后序遍历的非递归版的实现却让我犯了难。最终在查阅大量资料和自己动手画图演示后才明白了用两个栈去实现的非递归版后序遍历。

4、自己看网课学习的Java知识技术不扎实

前期自己看网课学习Java相关知识时缺少代码实践和锻炼，导致一些基础知识记的不牢固。所以后期在做训练营作业和最终考核项目的时候需要频繁查阅资料。但好在训练营开始前的学习和整个训练营的过程中，我都有记笔记记录自己已学知识的习惯，这极大地便利了后期我作业和考核项目的制作。

5、自定义连接池的实现

最终考核的项目需求文档中要求了不能使用外部的数据库连接池包，这意味着我必须自定义一个数据库连接池来代替我之前一直在作业中使用的druid连接池包。一开始我觉得这个过程难如登天，但我又不可能在项目中每次都手动创建连接Connection再手动释放。可当我在查阅一些相关资料和动手实操后，才发现实际上，一个数据库连接池的简答实现也并不是非常困难。尽管我在编写这部分的代码时遇到了非常多的问题和bug，例如连接异常关闭和连接不能正常产生等，但这些问题和bug都在我的一步步调试下找到并解决了。

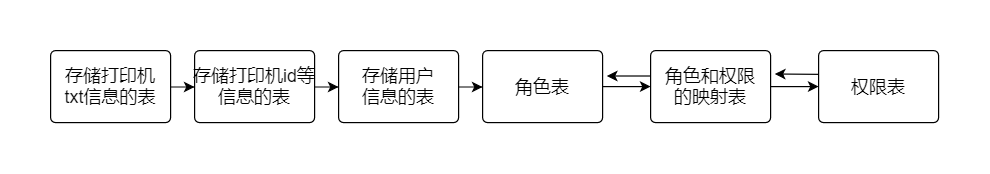
6、最终考核项目的需求分析和架构设计

所谓“千里之行始于足下”。我认为，如果要想整个项目在编写代码能够流畅自然，最开始的项目需求分析和框架设计必不可少。所以我在考核题目发布当天花了整整半天时间去选择、思考、分析考核项目和其需求的实现思路，写了一篇2000字的需求分析文档，并马上找到师兄咨询自己不懂的地方。

项目的架构设计我也花了不少时间，但我最后也在查阅相关资料后和师兄的帮助下敲定了项目的设计框架。

7、数据库表的结构设计

打印机产生的数据经后台处理后得到的信息大多都需要存入数据库中。如何设计数据库以便于数据持久化和数据的查询就成了绕不过去的一个难点。最终在师兄的意见指导下，我的数据库表结构修改了两个版本，最终敲定如下:



8、并发与多线程问题

由于项目中我需要新建多个线程，来模拟多台打印机驻留端同时向服务端发送txt数据的情况，所以在程序中不可避免地会遇到并发和多线程问题，而恰好这一部分在前期Java相关知识的学习中我掌握的并不是很牢固，这导致我在项目中实现相关功能之前一直都十分担心自己无法解决并发可能导致的问题。但最后在代码实现时，虽然也遇到了异常，但在经过调试debug后，最终锁定了问题的源头：我用来实现自定义数据库连接池的数据结构对象是线程不安全的。在修复此问题后，程序的异常便得到了解决。

9、前端的页面设计问题

开发一个JavaWeb项目时，不可避免的，需要使用html代码，css样式和JavaScript代码来设计前端的页面。尽管后台组对项目的前端页面的美观程度并无硬性要求，但由于html和css这两门标签语言和我之前学习的Java和C语言差距过大，自学看的网课又是面向后台开发人员的——前端内容浅尝辄止，同时html页面也无法像Java和C语言一样调试，所以我在完成训练营作业的前端页面时十分痛苦。但有了训练营作业的锻炼积累后，我在完成最终考核项目的前端页面时就比较轻松了。

# 7.心得体会

1、千里之行，始于足下

在进行实际的代码编写之前，我花了很长时间去分析三个选题的难度和所需要的技术栈，结合自己的兴趣和能力，在精挑细选后才最终确定自己的选题。而在确定好选题后，我又花了半天时间去进行详细的需求分析和需求实现的思考，并将这些内容整理成一篇文档。可以说，此后半个月我在编写代码时都是围绕自己的需求分析文档展开的。我认为，我之所以最终能够成功完成考核项目，正是因为我在正式开始代码编写前进行了细致的需求分析。这确保了我对项目目标和功能始终都能有清晰且一贯的理解，不会像无头苍蝇一样毫无目的地编写或修改自己的功能代码。如果我能将这一行为养成习惯的话，想必日后会大有裨益。

2、激流勇进，迎难而上

实际上，最终考核需求文档所要求的，一些听起来高大上的技术栈和知识，在考核开始前我大多只了解个大概或者完全没有听说过，例如驻留端和服务端的分离、jwt认证鉴权和websocket技术等。我当时也一度产生了畏难心理。但当我真的亲手去了解、去学习、去实践时，才恍然发现那些曾经我以为难如登天的问题，在我的亲手努力面前都不过是纸老虎。难题固然难，但畏难情绪大可不必有，因为世上无难事，只要肯攀登!

3、百折不挠，愈挫愈勇

在考核项目的开发过程中，我曾无数次遇到过各式各样的bug和异常。小到空指针异常、SQL语句错误异常、数据库连接失败异常，大到并发多线程问题和Servlet映射错误。它们之中的一小部分能被我很快地找到并解决，但是绝大部分，都需要经过漫长的调试后才能被我准确定位和想出原因和解决方案。它们之中不乏有让我一度以为是无解的bug和让我想直接放弃的异常。但在自我调整和师兄的指导鼓励后，我重新振作了起来，并最终解决了所有“无解”的bug，跨越了所有难关。

4、工欲善其事，必先利其器

随着项目的推进，我了解到的可以在项目中使用的新技术也就越多，例如httpclient技术——可以让后台具备发送http请求的能力、jjwt技术——创建和验证JSON Web Token。这些技术此前我都没有接触过，且考核项目的时间有限，学习新技术也需要时间。但在对这些可以在项目中使用的新技术进行调研，并评估其适用性和可行性后，我认为工欲善其身，必先利其器。学习新技术可能需要花费一些时间，但它们可以实打实地提高项目的质量和推进效率，此外也可以提高自身的技术水平。我认为这也是在推进考核项目时，我的一个重要收获。

5、勤于笔耕，温故知新

在整个训练营的过程中，每当我遇到一个问题，无论是技术难题还是知识点上的困惑，我都会立即记录下来，整理到笔记中，并在找到问题的答案时也将答案加入到笔记中。这样做的好处是可以随时回顾和复习，避免遗忘，同时还可以与他人分享，获得更多的建议和解决方案。记录项目进度笔记等文档还有助于我在项目的不同阶段进行追踪和监控，更好地管理项目进度。