关键字：
SpringBoot、权限管理、订单系统

1、绪论
共享经济的意义
互联网应该承担的责任
共享经济下的互联网
共享经济的农机共享平台

2、设计原则和开发内容
设计一款什么软件
实现什么功能
解决了什么痛点
原则是什么
程序描述图

3、软件架构设计和技术选型（选择性的写）
设计一款软件需要事先确定该软件所需的技术选型，例如应用什么样的软件技术， 利用什么样的编程语言，此外还要考虑软件的可移植性，适用性等。对于开发软件采 用的技术栈，要结合项目本身的需求，避免出现“重复造轮子”的情况，即已经有社 区贡献的开源库中提供了该功能的实现，但开发者却要花大量时间在这些已经被前人 实现过的功能上。本章将从技术选型，分析该项目的重难点等方面确定该系统的架构设计。

3.1 选择BS架构的优势
3.1.1 后端语言的选择Java
3.1.2 前端语言的选择JavaScript
3.1.3 关系型数据库的选择MySQL

3.2 后端框架SpringMVC
3.2.1 后端技术栈SpringMVC的介绍
3.2.2 数据库MyBatis，JDBC、MySQL Driver的关系
3.2.3 RestfulAPI的意义和实现
3.2.3 登陆拦截器在Spring的实现（权限验证）
3.2.4 一体化解决方案SpringBoot

3.3 前端框架React
3.3.1 React的介绍，设计哲学
3.3.2 Redux前端状态管理
3.3.3 ReactRouter前端路由
3.3.4 前端界面库Ant Design
3.3.5 前端图表库Echarts
3.3.6 前端一体化解决方案Umijs

4 各模块的设计
软件有哪些 功能，需要做权限区分，做数据库设计，做订单系统。
4.1 权限模块的设计
要写商家权限、农机租用者权限、超级管理员权限的意义和设计，为什么要做权限区分。
实现上，是用Spring内置的拦截器，拦截器根据Session来区分，Session靠Cookie来标记会话状态的，Cookie是目前最好的用户端标记身份的解决方案，拦截器如何维护和存储cookie池，讲讲cookie是什么，怎么实现的。参考安度因的视频。
每次请求带上cookie，后端判断cookie属于哪个权限，哪个session，做权限区分

4.2 mysql数据库表结构的设计
这个我就不用多说了。

4.3 农机信息查看与搜索的设计
农机信息是固定的，因为农机就是那么多，有5-10个农机，农机可以所有用户 自由查看，搜索。查看搜索就是查询操作，具体是怎么操作，是通过select，select like操作做关键字匹配。
前端通过POST发送数据，然后将接收到的数据渲染到前端组件上。结合图示说明这个搜索查询的过程

4.4 农机租用者的发布信息的设计
发布信息是农机租用者的权限，可以自由的增删查，对应insert select delete，具体如何设计。
前端如何设计，采用 表格的方式，请求一个什么接口，返回的数据，前端遍历数据，再渲染列表。结合图说明这个功能如何设计的。

4.5 商家发布信息设计

商家可以发布农机出租信息，相当于 发布商品。这里面可以填写哪些信息，发布之后要涉及库存，金额，生效时间等。对应的是数据库的什么操作，前端如何渲染，发布的信息的状态（有效，无效，无库存等）。

4.6 农机租用者可以下订单（租单）

租用者可以选择一个商家信息，选择数量，配送终点等，后台实时计算价格，然后支付，下订单。着重讲一下，为什么要区分配送目的地，金额如何计算的。金额在前端计算并显示。下订单的时候，后端要做很多事情：

检查目标商品状态、计算钱，计算配送费，算总金额，计算余额够不够。然后将订单写入数据库。将订单标记为：已下单状态

4.7 订单系统的设计（考虑跟上面一章合并）

订单系统的要素，需要做权限校验，算余额，订单还要有3个状态：已支付，商家已配送，配送完成。这需要在数据库中预留字段，做区分。

订单数据库如何设计，订单要一个uid，要结合时间戳，提交订单的用户，商品id等字段做md5计算，再加上随机噪声生成订单号。这样订单号不会重复。

为什么不能试用自增id当订单号？因为删除订单时可能会导致id不连续，容易泄露信息，不安全等。用随机uid做订单号可以保护用户隐私，可以做查询主键。

4.8 商家接单处理

商家接未处理订单，可以选择调配，或者取消订单。调配订单需要选调配者，类似于美团外卖的设计。设计逻辑要讲清楚，使用流程要配图。

4.9 配送者权限的设计

配送者登陆后，在我的，可以看到自己接到的配送任务，也可以将自己的状态改为：不自动接单状态。配送者可以在完成任务后，将任务改成完成状态。那么这个订单就完成了。配送者可以收到配送费，可以显示余额。可以显示配送历史。

5 重难点的设计

5.1 订单系统的详细设计，可以选择性的讲一下这里的难点。可以适当减少4.7章的内容，4.7章只讲业务流程，这里将代码和生成订单号的设计，订单号uid的生成，讲一讲为什么要将“用户名。时间戳，商品id”做哈希摘要，然后加入噪声。这样的uid唯一且无规律。适合并发场景，做正向生成，做查询缓存，做查询命中，结合布隆过滤器，还可以防止黑客攻击数据库。

（布隆过滤器基于Redis实现，在查询接口，查询不直接查数据库，先找布隆过滤器有没有值，然后再查缓存，最后没命中再找数据库。）订单uid属于高频操作，这部分是系统的弱点。

5.2 前后端分离，前端工程化的设计

前端采用react，antd，react-router、redux，这样可以提高前端一体性，并且后期做app的时候，后端可以复用。前后端用restful api通信。

前端、APP互换性设计，标准化设计。

前端用webpack打包，通过babel把ES6以上的语法转为ES5的js执行。（多了解一下前端工程化）

5.3 配送员登陆后，待配送事项提醒的设计

查询数据库，给订单一个未读状态flag，每次登陆就会查询，如果flag为未读就提醒。

6 软件功能测试

这章只能截图，详细解释功能能不能实现，用户约束是什么。稳定性测试，安全性测试。未登录不能访问任何接口，会被跳转回登陆页。

结论

夸一夸这个软件对XX有XX帮助

谢辞

写客套话。写这个软件有什么值得改进的地方。