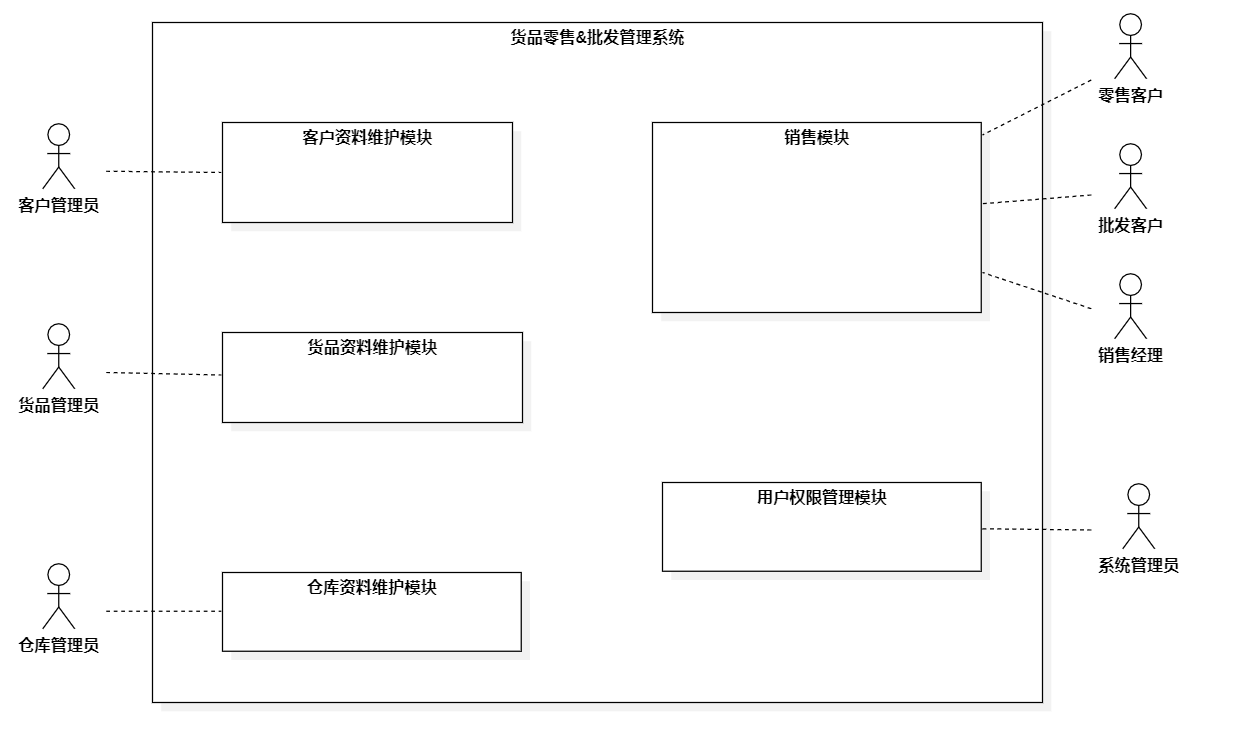
本项目取自前置课程“软件过程与工具”的小组大作业

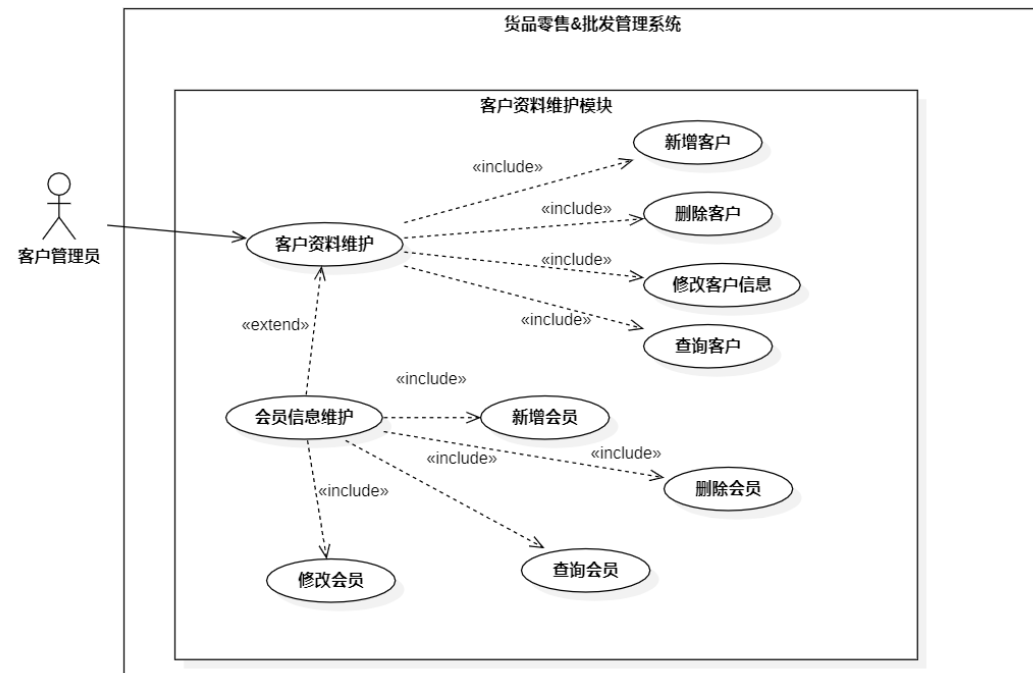
1. 用例视图

该系统可供多种不同身份的用户来登录使用，不同用户登录后所能够执行的模块不尽相同，主要如下

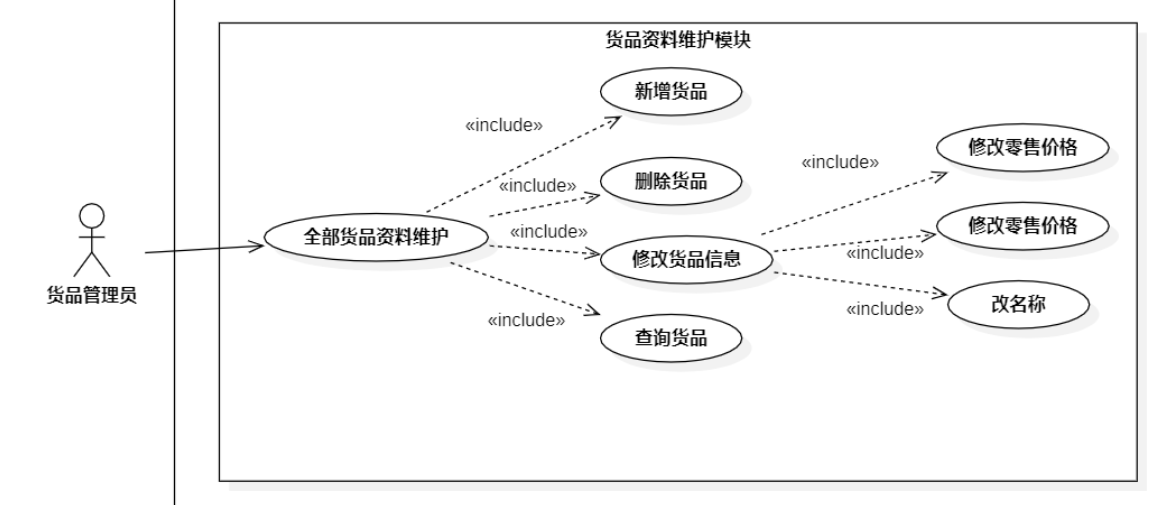


下面针对不同用户分别展示他们更为细致的用例视图

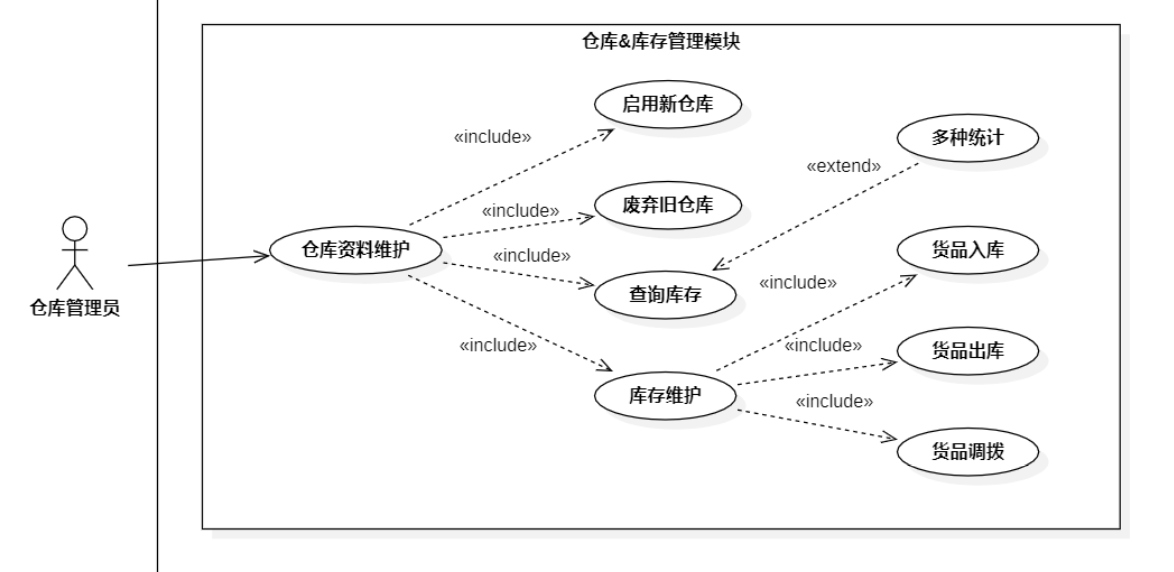
* 客户管理员



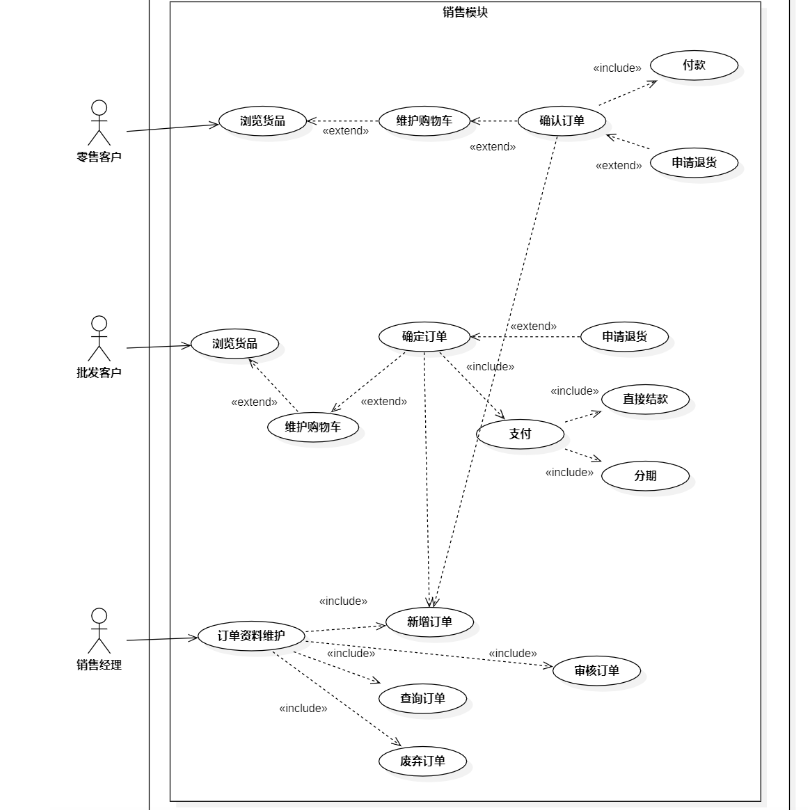
* 货品管理员



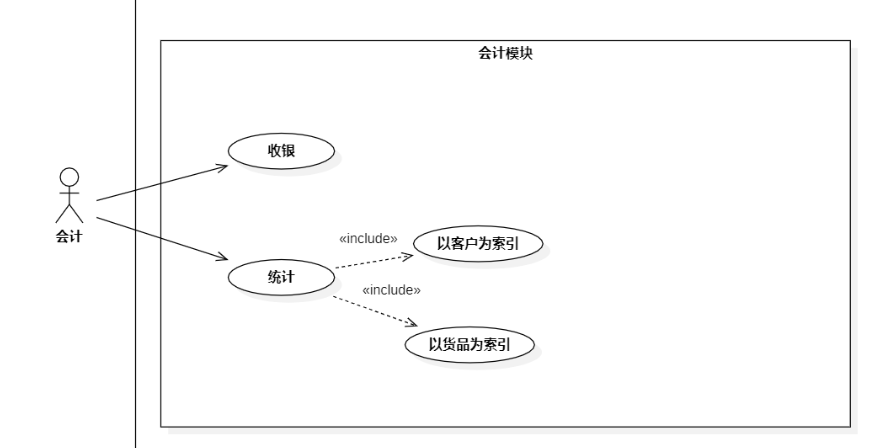
* 仓库管理员



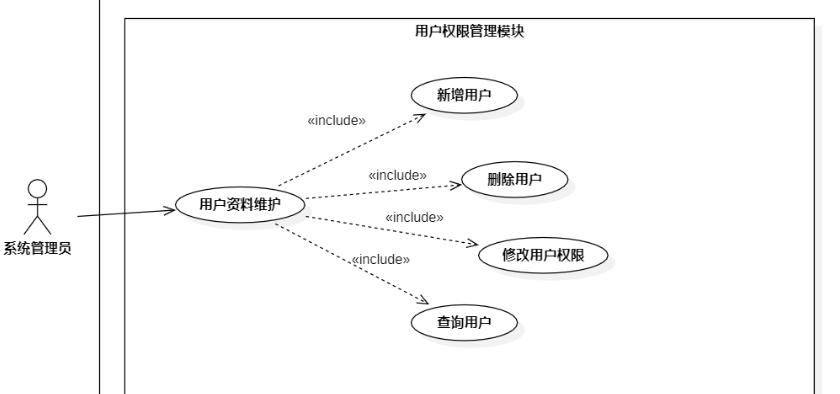
* 零售客户&批发客户&销售经理



* 会计



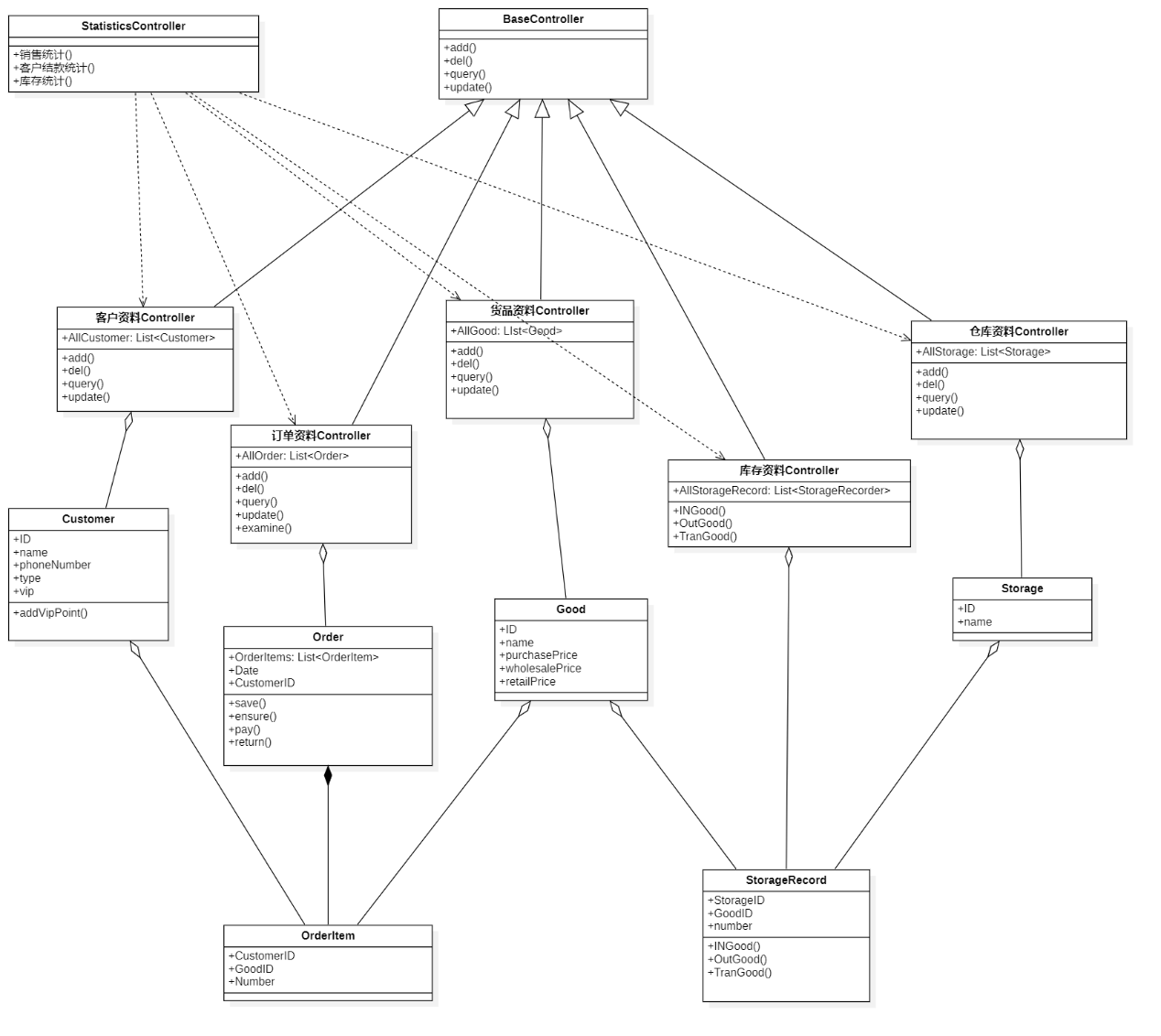
* 系统管理员



展示一下实际项目中的页面



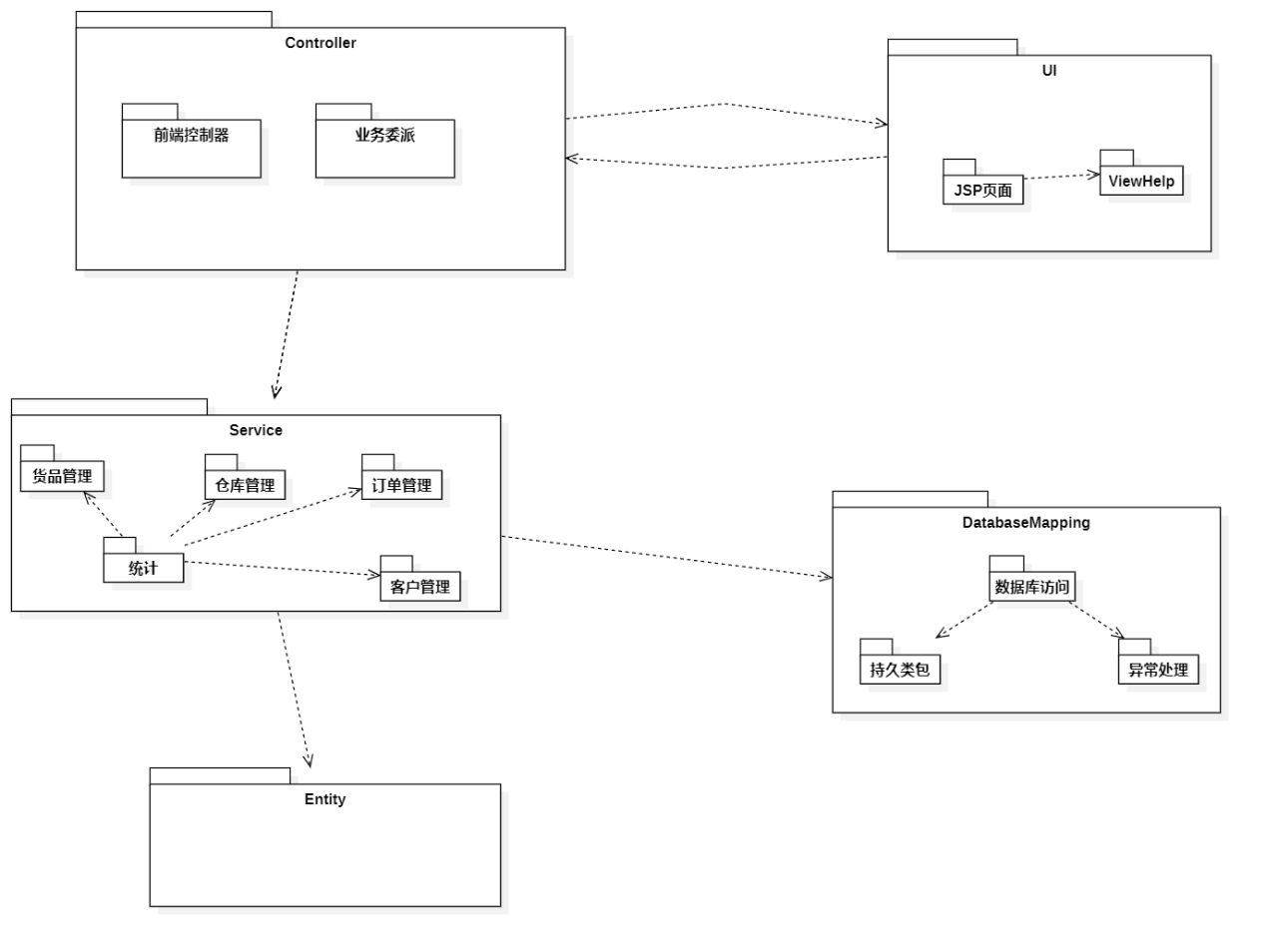
1. 逻辑视图



此项目基于Spring Boot框架进行开发，所以它包含“表示层”、“控制层”和“实体类层”，为了便于展示，此处仅给出了“控制层”和“实体类层”的Class Diagram

在上图中，“控制层”的类均以“Controller”为后缀进行命名，例如“客户资料Controller”、“订单资料Controller”等；其余类均是“实体类层”中的Class

1. 开发视图

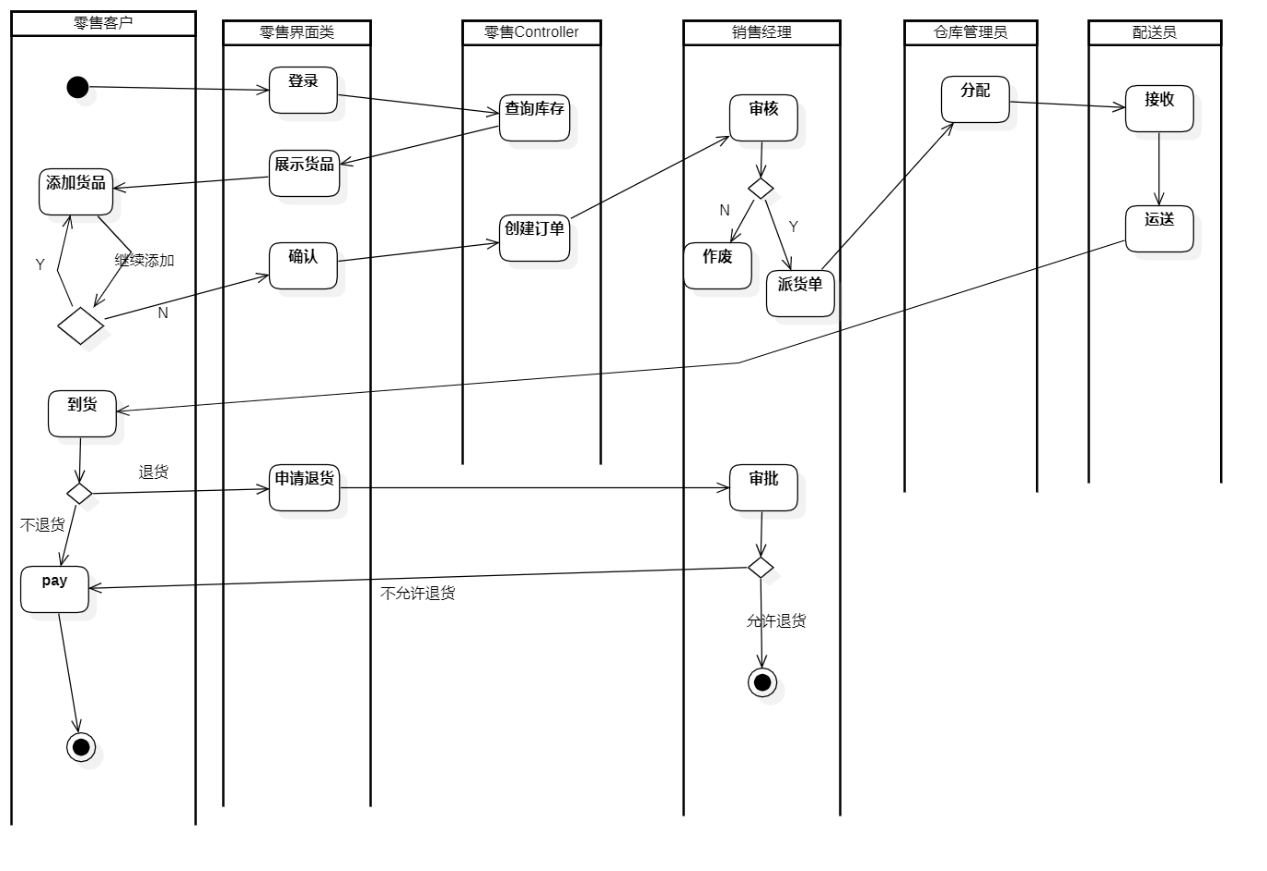


在具体的开发过程中，依据Spring Boot框架来实现，将不同功能的类放置在同一个包内，所以上述的包图中会包含：

* “控制类”包：Controller
* “实体类”包：Entity
* “边界表示类”包：UI

同时，为了和数据库服务进行想连接，我们还需要将各种访问数据库的类放置在一个包内：DataBaseMapping；在“控制类”和“实体类”中间增加一层“服务类”，“控制类”实际上是委托“服务类”来实现各个功能，这样可以使得整个系统功能分区更加明确

1. 进程视图

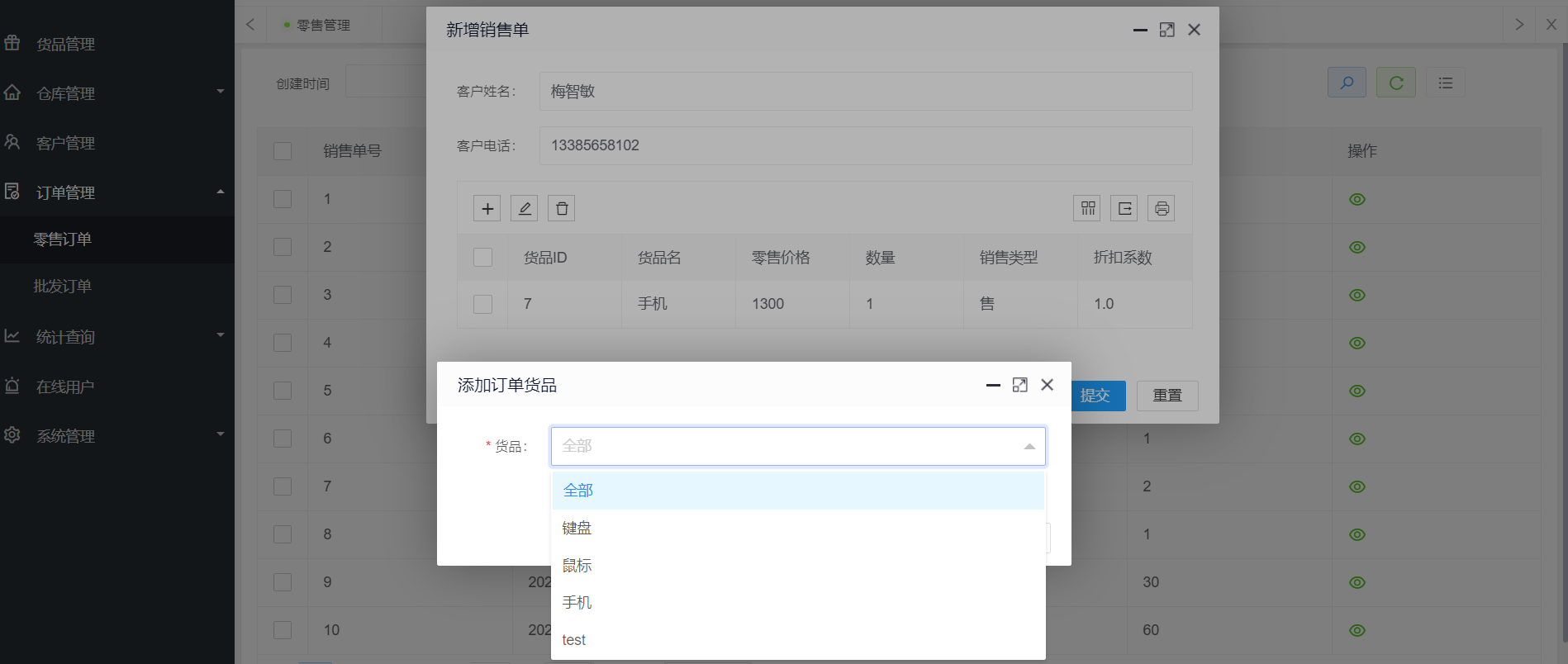


此项目功能繁多，故展示其中最重要的一个业务流程：用户购货流程，并画出活动图。

从上图可见，“用户购货”这个业务流程中，

* 客户本身仅需要和“零售界面类”进行交互即可，并不需要管系统内部的运转逻辑。
* 其他相关人物（如：销售经理）的任务则是在用户确认订单之后，系统通过“零售控制类”自动新建订单，当销售经理登录进系统之后，便可以进行审批。不需要直接和客户进行对接，从而提高了整体效率

展示一下实际项目中的客户购货界面



1. 物理视图

