

# OOP 大作业实验报告

马珩峻 171860627

## 一、引言

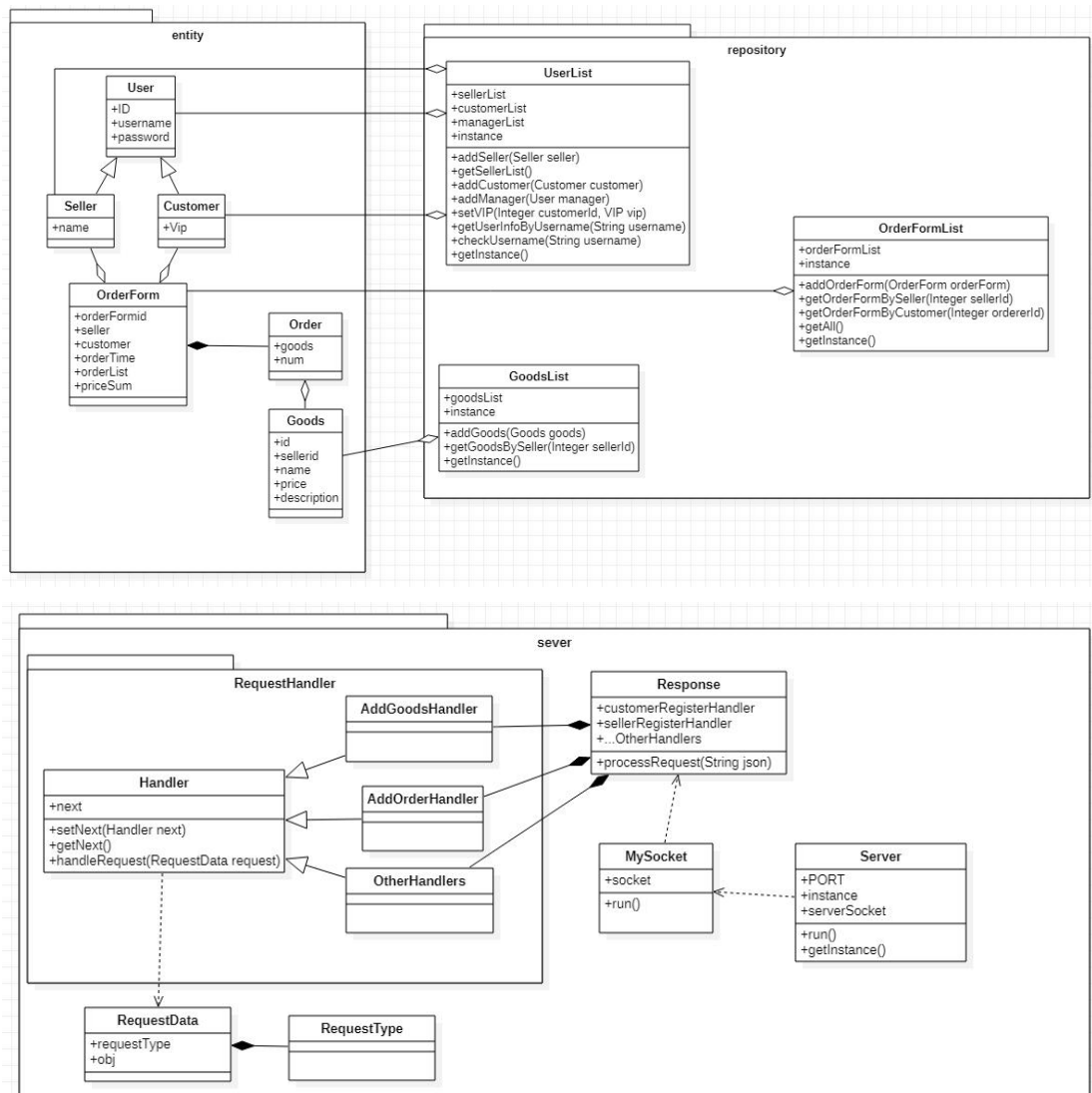
本项目使用 java 语言进行编写，客户端与服务端之间使用 socket 进行通信，客户端使用 javafx 实现了 GUI 设计。

## 二、分析设计

简易外卖软件首先需要分别设计客户端和服务端，然后总共有 3 种类型的客户：商家，客户，管理员，然后涉及到要操作的对象有商品，订单，购物车等，另外客户端和服务端通信时需要一个 Socket 对象进行通信，需要对请求数据包进行处理，需要将用户发送的数据存入服务器中。

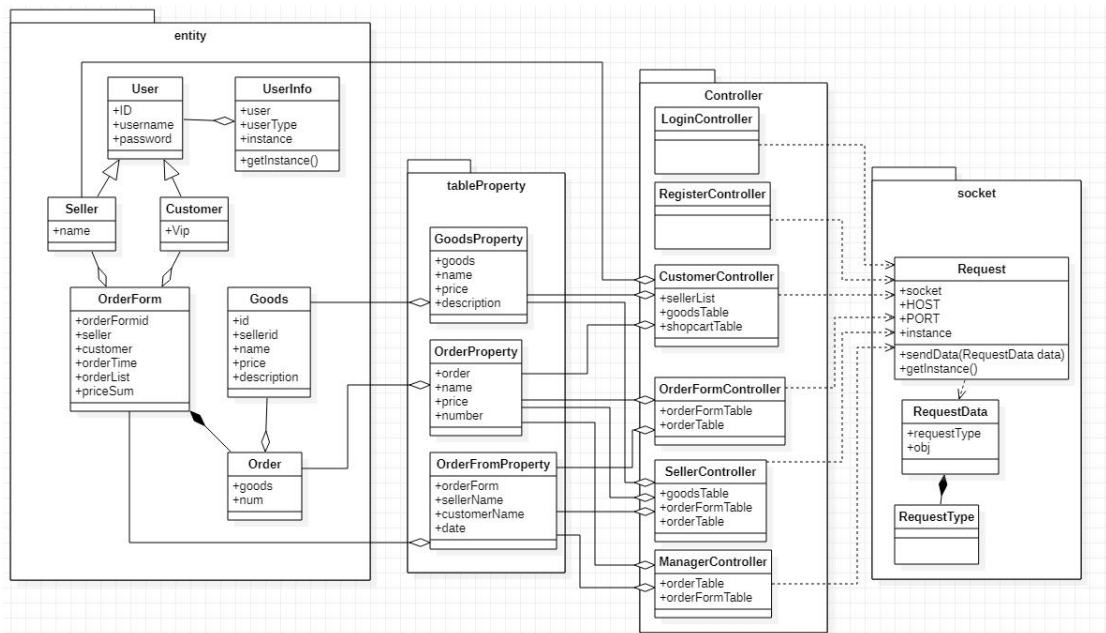
对于服务端，服务器 Server 只会有一个实例，所以采用单例模式，同样的，各种与服务器数据交互的类也只会有一个实例，也采用单例模式，另外 java 原生的 socket 是没有多线程的，需要转换为能够多线程运行的 MySocket 类，这里使用了适配器模式，处理客户端请求时，创建不同请求的处理类，使用责任链模式对客户端的发来的请求进行处理。

分析后可以得到如下 UML 建模（所有类方法省略了对属性的 set 和 get 方法）：



对于客户端，首先一个客户端只会有一个用户使用，所以使用 UserInfo 类管理单前用户，并且使用单例模式。对于 javaFx 主要使用 Controller 来对 GUI 上的元素进行操作。对于 javaFx 的表格，所需要的数据需要特别定义的属性才能显示，将原本 entity 中的实体对象显示到 table 中时需要使用适配器模式将类转换成能在 table 中显示的对象。然后对于 javaFx 的表格和列表的点击响应事件，需要添加监听器对象来对事件进行监听，这里使用了观察者模式。另外负责发送请求的 Request 类也只应该有一个实例，所以使用单例模式。

分析后可以得到如下 UML 建模：（Controller 中的 GUI 控件属性太多且不影响不具体写明，只说明有聚合关系的，控件方法同理）



### 三、实现方案

如分析中所说，按上一步的分析给出了具体的实现，对于服务器上的数据存储使用了 HashMap 的数据结构存储相关数据，数据库与服务端通信时主要将数据先转为 Json 字符串通过 socket 字符流传输，收到后再将 json 字符串反序列化获得相应的对象。设计模式又上一步分析中有用到：单例模式（并且大量使用），责任链模式，适配器模式，观察者模式。

### 四、功能实现

实现了所有基础功能和扩展功能

展示：用户注册登录

LaoBaOrder

老八外卖

注册

用户名

密码

确认密码

客户

不成为VIP

确认

取消

LaoBaOrder

老八外卖

注册

用户名

密码

确认密码

商家

商家名称

确认

取消

LaoBaOrder

老八外卖

注册

用户名

密码

确认密码

管理员

确认

取消

LaoBaOrder

老八外卖

test1

•

登录

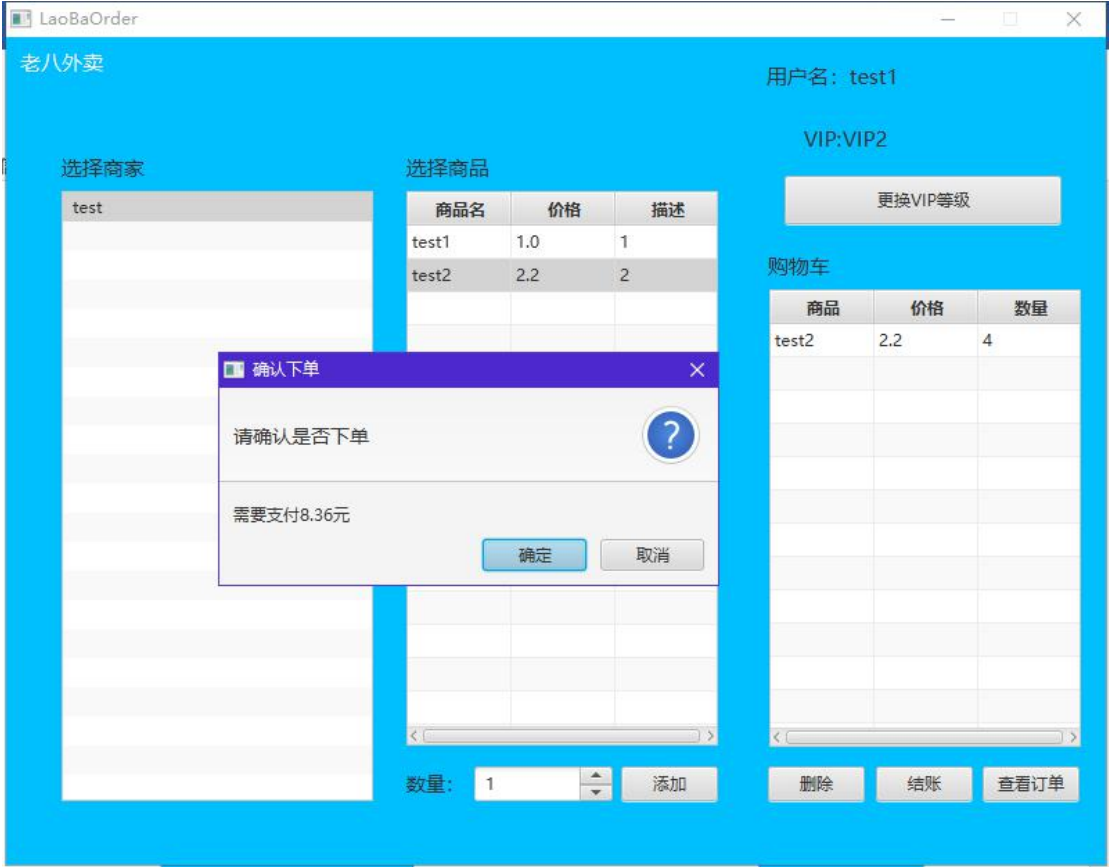
注册

登入

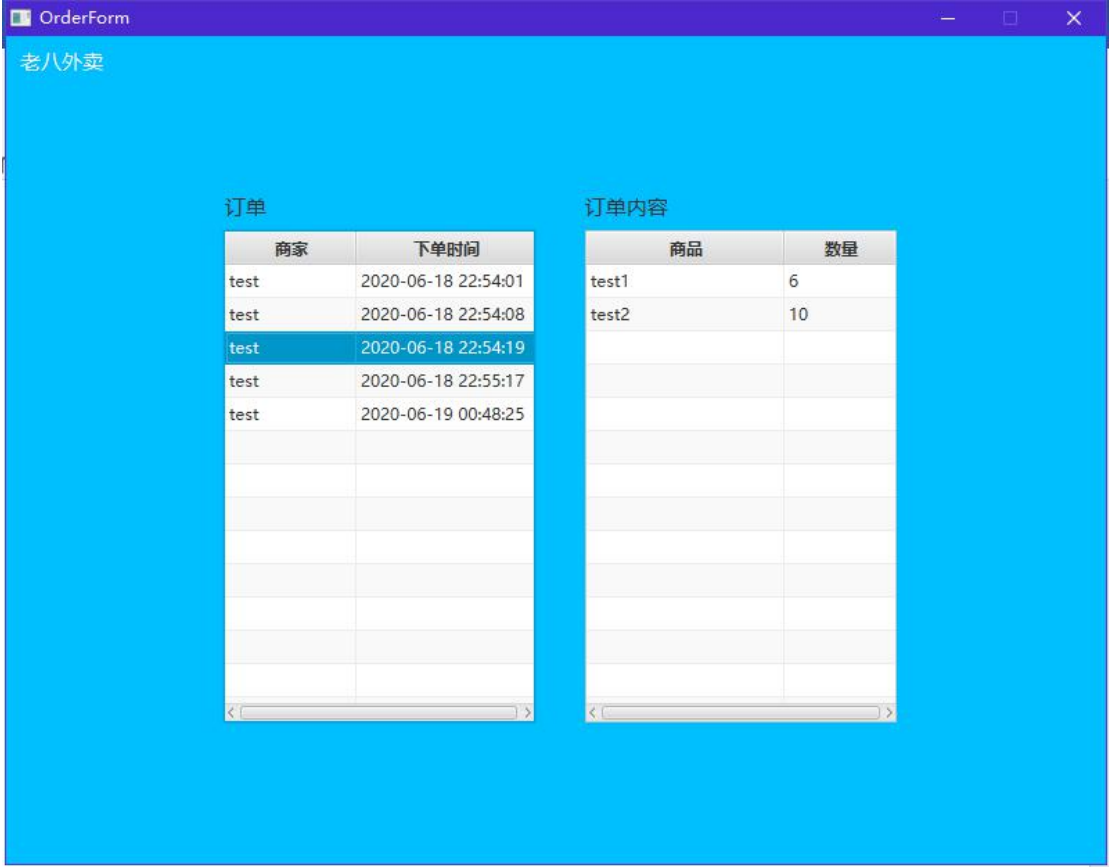
登陆成功

确定

客户界面：



客户查看订单：



商家页面：

The screenshot displays the LaoBaOrder application interface. The title bar at the top reads "LaoBaOrder". The main content area is divided into several sections:

- 老八外卖 (LaoBa Order):** Located at the top left, it includes a "用户名: test2" (Username: test2) label and a "店铺名: test" (Store Name: test) label.
- 已有商品 (Existing Items):** A table with columns "商品名" (Item Name), "价格" (Price), and "描述" (Description). It contains two rows: "test1" with price 1.0 and description 1, and "test2" with price 2.2 and description 2. Below the table is a scroll bar.
- 订单 (Orders):** A table with columns "客户" (Customer) and "下单时间" (Order Time). It contains four rows: "test1" with time 2020-06-18 22:54:01, "test1" with time 2020-06-18 22:54:08, "test1" with time 2020-06-18 22:54:19, and "test1" with time 2020-06-19 00:48:25. Below the table is a scroll bar.
- 订单内容 (Order Details):** A table with columns "商品" (Item) and "数量" (Quantity). It contains two rows: "test1" with quantity 5, and "test2" with quantity 3. Below the table is a scroll bar.

At the bottom of the interface, there are three input fields and two buttons:

- 商品名:** A text input field.
- 价格:** A text input field.
- 描述:** A text input field.
- 添加新商品 (Add New Item):** A button.
- 刷新订单信息 (Refresh Order Information):** A button.

管理员界面:

设计原则：遵守了里氏替换原则，单一职责原则，合成复用原则

## 五、小结

经过本次大作业，对设计模式的使用有了一定的理解，能够将一些设计模式用于项目实施中，另外对 java 的 socket 通信和 javafx 的 gui 编写也学到了不少。