一、代码实现讨论，包括：

（1）描述代码实现过程对于原来设计的迭代内容，即推翻和继承的部分；

推翻的部分：对实体类集合，删除了属性信息的反射方法，并优化了与JSON格式的序列化和反序列化，便于网络传输。

将DTO（数据传输对象）方法集成到客户端，而将Handler、DAO等方法集成到服务端，以共享配置信息。放弃了原来同步通信的想法，改用以Qt信号和槽机制为基础的异步通信，并重新实现了基础网络数据发送/接收对象。

继承的部分：继承了客户端-请求发送API-DTO-分发器/响应器-请求处理API-DAO-数据库的基本架构，坚持采用MVC框架，继续用JSON作为数据交换的媒介，坚持将不同请求的处理逻辑封装到不同类中。

（2）MVC模式的体验如何，坚持分层了吗？

MVC框架将一个大型软件工程分成几个相互独立、彼此协调的模块，制定了一套科学的方法论。它将上层业务与下层逻辑分离解耦，极大简化了程序开发的流程和提高了代码的可维护性。本人吸收借鉴现代软件开发中的经典架构和设计模式，建立了以下结构：客户端-请求发送API-DTO-分发器/响应器-请求处理API-DAO-数据库。

```
microtrade/
├── client/           # 客户端代码
│   ├── main.cpp      # 客户端入口
│   ├── mainwindow.*  # 主窗口
│   ├── welcomewidget.* # 欢迎界面(登录/注册)
│   ├── logindialog.*  # 登录对话框
│   ├── registerdialog.* # 注册对话框
│   ├── productdialog.*  # 商品详情对话框
│   ├── commander.*    # 全局状态管理
│   └── config.*       # 客户端配置
├── server/           # 服务器端代码
│   ├── main.cpp      # 服务器入口
│   ├── tcplocalserver.* # TCP服务器
│   ├── config.*       # 服务器配置
│   └── sql*.*         # 数据库操作类
├── entity/           # 数据实体类
│   ├── authorization.* # 授权信息
│   ├── product.*      # 商品信息
│   ├── promotion.*    # 促销信息
│   └── cart.*         # 购物车信息
└── tcpserver/        # TCP通信相关
    └── tcpserver.*   # TCP服务器基础类
```

（3）设计模式使用了吗？如何用的？

单例模式：配置管理类（Config），从INI配置文件初始化配置信息，全局单例访问；服务器类（TcpLocalServer），在主窗口初始化并开启监听；分发器类（TcpLocalDistributor），为全局服务器配置唯一的分发器，进行请求转发和包装处理结果。

工厂模式：创建处理器基类，和派生的具体请求处理器，由分发器/处理器工厂进行统一创建、调用和销毁。

策略模式：通过针对特定请求的发送API封装发送路由、参数列表；根据路由（route）信息调用相应的处理逻辑处理不同请求；处理器调用DAO策略完成请求处理。

观察者模式：Qt的信号和槽就是采用了观察者模式，槽函数通过信号触发。

（4）多线程用了吗？如何用的？

未使用多线程，而以信号和槽的伪异步代替，无法真正提高性能，但都可以实现异步操作。

（5）描述你认为比较经典的处理技术？

抽象化：将业务数据、数据传输、数据处理、业务逻辑等涉及的过程抽象成对象集合及其相关操作，例如模型（Model）、DAO等。

序列化/反序列化：将模型/实体转成可进行网络传输的数据，并可从中还原，从而完成数据交换。

二、本周遇见了那些技术问题，如何解决的？

问题	解决方法
请求发送和接收方法需要的参数多，造成代码冗余。	在应用项目中把基础请求相关工具二次封装，将固定的参数值（如超时阈值、服务器地址、端口号）与配置信息关联。
退出应用无法进行清理（如自动登出等）	在用户点击关闭时询问，如果是，系统自动启动清理，并设置超时保护，确保数据库数据准确。

三、你现在对自己的系统还有什么优化想法？

（1）使用多线程优化服务端处理请求和图片加载。

（2）不以模型而已其序列化结果作为容器元素，解决容器不能自动序列化导致的代码冗杂问题。

（3）将日期时间格式与SQLite统一。

四、本周拟解决什么问题，如何解决?

（1）优化用户体验，完善提示显示，能在购买前预览折扣和花费。

（2）开发管理员端，迁移部分数据库操作。

（3）引入搜索功能，使用第三方库，支持商品、购物车、订单、促销等列表文本搜索。

（4）加入商品分类筛选功能，添加相关请求处理逻辑。