

# QT 大作业最终报告

## 一、程序功能介绍

本项目计划实现一个简易的线上二手交易平台，供北大校内用户使用，方便校内师生进行书籍、演出门票、日用品等二手物品的交易。设计目标是：相比闲鱼等成熟交易平台，能够更精细地覆盖校内交易需求；相比目前主流的通过树洞联系交易，做到更加方便快捷。

目前程序中实现的功能有：用户可以通过客户端在端注册账号并登录，登录后可以上传想要出售的商品的相关信息到服务器端，支持上传图片、文字描述、价格等。用户也可以在服务器搜索自己想要的商品，搜索的商品库内保存了所有使用该程序的用户发布的商品。在搜索到心仪的商品之后，可以给卖家留言，进一步商谈交易细节。

## 二、项目各模块

项目主要包含两个模块，服务器端模块 `tcpserver` 和客户端模块 `front_end`。

`tcpserver` 模块内包含处理传输任务的 `server` 类和若干操作用户数据库及商品数据库的函数。

其中，`server` 类继承自 `QDialog`，包含一个 `QTcpSocket` 类的对象，通过该对象基于 TCP 协议进行数据传输。同时，`server` 类包含有接收（`updateServerProgress()`）和发送（`startTransfer()`）两个函数。

接收函数由信号-槽机制控制，收到数据时会自动运行并将收到的数据存储在 `server` 类中的变量里，便于数据库相关函数的取用。接收函数还会根据客户端发送来的数据包更新 `whatToDo` 变量，根据该变量的值执行不同的操作。

发送函数则需要由其他函数进行调用，会将目前 `server` 类里存储的图片、文字信息打包成数据包，发送给客户端。

此外，`server` 类还有一个 `QDialog` 类的 `ui`，可以展示最新收到的图片和文字信息，主要目的是为了 `debug` 方便及测试图片显示函数，没有实际功能；仅 `ui` 上的“监听”键控制了服务器开始监听的动作。

数据库函数实现细节如下：

### 1. 文件夹树结构设计

我采用了文件夹树结构来存储用户数据和商品信息，其中每个用户拥有以其用户名命名的独立目录，进一步包含商品信息和密码文件。这种结构便于文件管理和权限检查，同时简化了文件操作流程。

### 2. Searchlist 维护

为了支持高效的关键词搜索功能，我创建并维护了一个名为 `searchlist` 的特殊文件。该文件记录了所有可用商品的名称和对应的卖家信息，每条记录占一行，以便于快速检索和匹配搜索查询。

### 3. 商品类构造

我设计了一个 `shangpin` 类来封装商品的所有相关信息，包括商品名称、描述、图片、价格。这个类便利了接口间的传输。

### 4. 数据库函数实现

实现了以下关键数据库函数：

`save_nameandpassword`：用于注册新用户并保存用户名和密码，返回一个整数标识是否成功，或者使用了重复的用户名。

query\_password : 用于验证用户登录凭据, 返回一个整数标识是否成功, 或者发生了密码错误, 或者用户名未注册。

save\_shangpin : 用于保存新商品信息到用户目录中, 并添加到 searchlist。

remove\_shangpin : 用于从系统中删除商品信息, 并修改 searchlist。

keywordsearch : 用于执行基于关键词的搜索, 返回匹配的商品列表。为此单独编写了一个基于动态规划的最大子串匹配函数, 用于查找与搜索词拥有最大相似度的商品, 返回一个 QList 保存商品名以及它的拥有者。

query\_shangpin: 用于从数据库中获取商品数据, 返回一个 shangpin 对象, 可以通过它得到商品名、拥有者、描述信息、图片信息、价格。

front\_end 模块内包含客户端的各个窗口的 ui 和控制函数, 以及处理数据传输任务的 client 类。

其中, client 类与服务器端的 server 类对接。client 类中控制数据传输的方式与 server 类相近, 有接收、发送两个函数, 接收函数由信号-槽机制调用, 发送函数在需要发送数据给服务器端时调用。

与 server 类不同点主要在: client 类在读取完服务器发来的数据包后, 会发出 readMe() 信号, 通过该信号控制各个窗口作出响应, 再根据由数据包更新的 whatToDo 变量执行对应的操作, 如登入账号、在 ui 上呈现搜索结果等。

此外, client 类还带有一个 name 变量, 指代用户的用户名, 用于在服务器数据库的检索。

各个窗口的控制代码没有特定的设计思路, 按功能需求编写。值得一提的是, 所有需要与服务器端进行交互的函数都包含了一个类型为 client 类指针的变量, 通过该变量控制 client 函数进行读写, 并由该变量发出的 readMe() 信号调用相应的槽函数。

### 三、小组成员分工状况

许昊楠, 主要负责数据传输部分, 完成了 server、client 类的编写以及前端窗口交互逻辑的编写。撰写了最终报告和录制演示视频。

姚俞周, 主要负责数据库部分。完成了数据库函数的编写, 并将 server 类与数据库函数对接。合作撰写了阶段性报告和最终报告。

杨舸, 主要负责 ui 设计部分。完成了对 ui 的前期设计和后期美化工作。撰写了阶段性报告, 合作撰写了最终报告。

### 四、项目总结与反思

#### (一) 项目总结

姚俞周: 在本项目中, 我深入理解了 Qt 框架在文件操作和数据管理方面的应用。通过实现数据库函数模块, 我确保了平台能够处理用户和商品数据的增删改查操作。文件夹树结构的设计提高了数据组织的清晰度和访问效率。Searchlist 的引入大大加快了搜索功能的速度, 而商品类的构造则提供了一个灵活且可扩展的方式来管理商品信息。

许昊楠: 在本项目中, 我掌握了 Qt 框架下进行文件传输的方法和细节, 也对 Qt 的槽函数机制有了比较深入的认识。在实现 server、client 两个类并将它们

对接的过程中，我能够比以往更加熟练地将需要的功能进行封装和接口化。在编写交互逻辑的过程中，我切实地体会到了接口设计带来的开发效率提升。

杨炯：在本项目中，我掌握了从设计稿到 ui 的基本转换，了解了 css/qss 文件与 Qt 的交互方法，对 ui 设计中需留有槽函数接口有初步意识。

## （二）反思

姚俞周：在参与开发这个二手交易平台项目的过程中，我获得了宝贵的经验和深刻的认识。以下是我对于项目开发过程中的一些反思：

### 1. 接口对齐的重要性：

我深刻意识到接口设计的一致性和预见性对于提高开发效率至关重要。接口对齐确保了不同模块之间的交互顺畅，减少了因接口不匹配而产生的额外工作量。在未来的项目中，我将更加注重接口的设计和文档化，确保所有团队成员对接口的理解和使用保持一致。

### 2. 数据结构与算法的选择：

在设计数据库函数时，我选择了文件夹树结构和维护 `searchlist` 来优化搜索效率。然而，我也发现随着数据量的增长，这种结构可能面临性能挑战。我需要进一步研究更高效的数据结构和搜索算法，如倒排索引，以应对未来可能的扩展需求。

### 3. 异常处理与代码健壮性：

项目中，我学习到了异常处理对于保证代码健壮性的重要性。在文件操作和数据交互过程中，合理的异常捕获和错误反馈机制可以显著提升系统的稳定性和用户体验。我将在未来的编码实践中加强这一方面。

### 4. 团队协作与沟通：

通过与队友的紧密合作，我认识到了有效沟通在项目成功中的关键作用。良好的沟通有助于及时解决问题，避免误解和重复劳动。我将探索更高效的团队协作工具和方法，以促进项目进度和团队士气。

### 5. 技术债务的管理：

在项目开发过程中，我意识到了技术债务对长期维护的影响。为了避免技术债务的累积，我将更加注重代码的可读性和可维护性，及时重构和优化代码。

### 6. 持续学习与自我提升：

技术领域不断进步，我认识到了持续学习的重要性。我将不断更新自己的技术知识，学习新的编程范式和工具，以适应不断变化的技术需求。

通过这次项目，我不仅提升了自己的技术能力，也加深了对软件开发流程的理解。我期待将这些经验应用到未来的工作中，为创造更优秀的软件产品贡献力量。

许昊楠：上文对编程方面的反思已经非常到位且深入了，我不做赘述。但在团队协作方面，作为组长，我觉得这次项目中我不甚称职。在项目开发的中后期，我没有组织大家做好工作的对接，在工作的分配上也没有在组内达成很好的共识，导致最后时间比较紧迫，我编写的窗口交互部分也因此较为简陋，影响了项目总体的完成质量。

杨炯：由于 Qt 对 css 文件交互的版本限制，在前期摸索效果上浪费了很多

时间，此外由于对代码部分结构上的认知极其有限，在设计接口时出现了错漏现象，为前后端的对接工作增添了很多不必要的麻烦。我认为在初学过程中，应该对项目的运行机制建立整体认知，并且对各组件的功能均要基本掌握，才能快速高效地进行页面设计和对接工作。