NEU程序设计技术

项目说明文档

作者：曹子帆 20181245

<https://github.com/czfzc/hw-shopsale-jsondb>

本项目代码可以在上面连接查看最新版本，即使已经交了作业，我可能也会在github仓库内继续更新，同本文档文件夹下的代码可能已经过时。您可以在源代码主目录下使用git pull origin master命令拉取以更新代码。

**前言**

本项目是用C++语言编写的一个功能不完全的小商城管理系统。不过与其说是商城，不如说是个小卖店，因为本项目仅仅是作为一个结课的示例程序出现，实用性比较低，毕竟一个真实的商城需要考虑的因素非常多，以我在开发和维护的东大一小时外送外包微信小程序（微信可以搜索“东大一小时”）的项目为例，其和微信支付对接的订单系统足足有1000多行，用户系统也有百余行之多，但是一个体量比较大的成熟稳定的服务端程序一定有很多无论是开发又或是维护方面的优点供小型程序借鉴，故根据我很长时间来开发东大一小时后端程序的经验，我在开发本项目的时候也用到了相应的思想和代码风格，同样我也认识到，完整地开发出一个程序对于编程经验和习惯是一个很好的增长和培养的过程。这里的编程经验，可以指某一种程序设计思想，可以指程序开发效率和准确率，也可以指对于某一种语言的底层特性的了解。简单来说，本项目着重想展示的是在程序主体框架和设计思想上的创新，并非程序的业务功能和业务细节，毕竟这个项目肯定不会被应用于实际场合。

**程序设计思想的介绍**

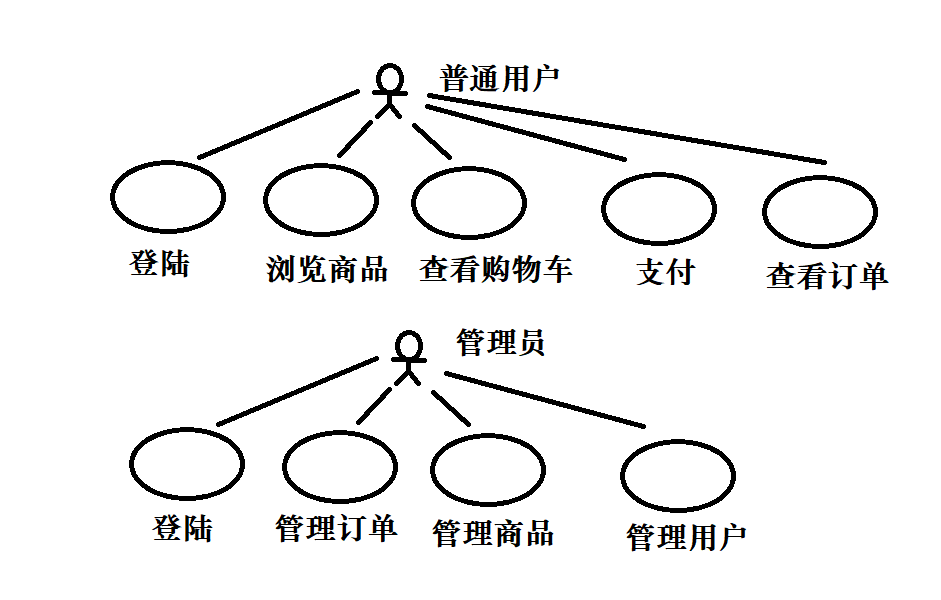
鉴于篇幅原因，我将长篇大论汇聚成短短的介绍。首先，本项目用到了网站后端开发过程中常用的MVC的设计模式，MVC即model（M，模型），view（V，试图）和controller（C，控制器），所谓model即数据库模型实例，我们在开发一些业务功能的时候往往需要面对一些业务实体的构建，比如说在网页里看到的一个商品信息卡片，上面可能包含着商品的具体信息，而几乎所有的信息在数据库中都是以字段的形式抽象记录，而一个信息字段的集合便成为数据库的一个记录，假如我们要实现一个信息卡片的功能，我们不得不考虑记录从数据库变成用户可以清晰阅览并且理解的模型。在这之间，经历了数据映射到模型，封装数据库中间层操作模型，控制器操作数据库中间层以获取数据在程序中呈现的实体，即数据库模型实例，然后再将此实例转化成可以被方便格式化并且传输的协议传送到客户端渲染成用户可见的形式，这里提到的实体就是模型，数据库中间层就是被控制器操作用来将数据转化成模型呈递给控制器理解的“工厂”，而控制器便是我们主要实现逻辑代码并且渲染成视图交给用户的地方，这样写有什么好处呢？我认为有以下几点，第一点是减少了开发阶段程序员对数据库语言编写的负担，从而使程序员可以专心处理业务逻辑，在业务逻辑中可以将数据库想象成模型，对模型的创建、修改可以被数据库中间层自动保存和映射到数据库中，简单来说就是数据库中间层可以自动处理程序和数据库的关系，包括但不限于自动生成数据库语句，自动连接数据库和处理数据库间的语法方言差异等。第二点就是方便业务逻辑层和数据层解耦，可以简单的修改数据库中间层引用的驱动，实现驱动抽象出来的接口（在java里面是叫做JDBC），就可以轻松的修改数据库类型而不造成任何影响，只要保证接口实现完全。在我的程序源代码里模型存放在model文件夹内，数据库中间层存放在manager文件夹内，控制器存放在controller文件夹内，不完全的view层你可以简单地理解成控制台窗口，实际上并未做实现，所以说这是个不完全实现MVC设计模式的小程序。

**数据库层介绍**

既然要实现商城管理系统就必然少不了数据层面的考虑，我们把这些信息转化成什么样的格式的数据流，然后以什么样的形式存储在计算机之中，我考虑如果直接以结构体对象的形式在内存中开辟空间并且存储实在是不可取，不然每次关闭程序数据就会丢失，而且对于大流量的业务内存实在是非常宝贵的资源。而如果以二进制的形式存储在文件之中虽然节省空间，但是不便于管理和解析，如果对数据在存储前加以某种格式的调整后再进行存储，这样下次读取的时候只需要简单的解析这种格式，便能对数据进行很好的管理，鉴于这个程序目的在说明某种程序思想而不是偏向于实用的目的，我就没有对数据库的一些其他算法和功能做考虑，比如数据的索引、二分查找、红黑树存储，以及关系型数据库必备的级联操作、主键功能、各种类型的数据类型的存储，作为一个示例程序，够用即可。

在实现上我将一个表作为一个文件存储，这个表内所有的记录做为json格式的数组，而每一个记录为一个json对象，数据库核心代码在util文件夹下的DataUtil.h，里面的一些工具函数用来实现了一个基于json格式存储的文件型弱关系型数据库（其实它甚至不能叫做数据库，它没有任何客户端与服务端，仅仅是作为一个头文件引入的依赖形式存在，它会将要存储的数据写入文件并且从文件里取出符合条件的数据，并且这些文件的位置与主程序的位置是高度依赖的，而且它也没有实现任何事务，原子性操作，以及数据关联级联功能甚至它都没有对文件进行加锁，以及对自身的函数进行加锁操作，原因有二，一是快期末了由于时间原因我无法再实现这些，二是我对linux编程比较熟悉，这些可以用linux系统调用的函数进行实现，比如文件锁flock和互斥锁mutex，但是这些会大大降低此应用的移植性，毕竟我还想编译一份windows版本作为作业提交）。

**用例图（需求分析）**



**运行**

您可以在linux里面直接运行项目源代码下的main文件，也可以用vscode直接打开项目文件夹，其项目文件为.vscode。

**总结**

IMG_256经过了为期一个星期的间断开发，我对于这种小程序的开发有了新的认识，本以为简单两百行就可以搞定的程序硬是被我写了2000行，实际上尝试运行后发现也没有多少内容，其实我大部分时间都是在思考如何使我开发的程序阅读性，二次开发性更强，更具有设计模式，而不是简简单单的堆砌代码，经过不停的删改，让我的程序地基（也就是数据库和数据库中间层）非常地稳定，令我后续的开发轻而易举、顺风顺水，几乎没有为程序的bug而操心，更使其能作为一个优秀的程序作业顺利提交，这就是磨刀不误砍柴功。如果老师您有任何问题或者是我的作业哪里有不足之处可以随时传唤我 ，QQ是1780312044。