**工作报告（第二周）**

1. **小组/项目信息**

成员信息

|  |  |  |
| --- | --- | --- |
| 姓名 | 班级 | 学号 |
| 刘启熙（组长） | 软件1班 | 201830730307 |
| 陈嘉俊 | 软件1班 | 201830720063 |
| 龚喜 | 软件2班 | 201830660345 |

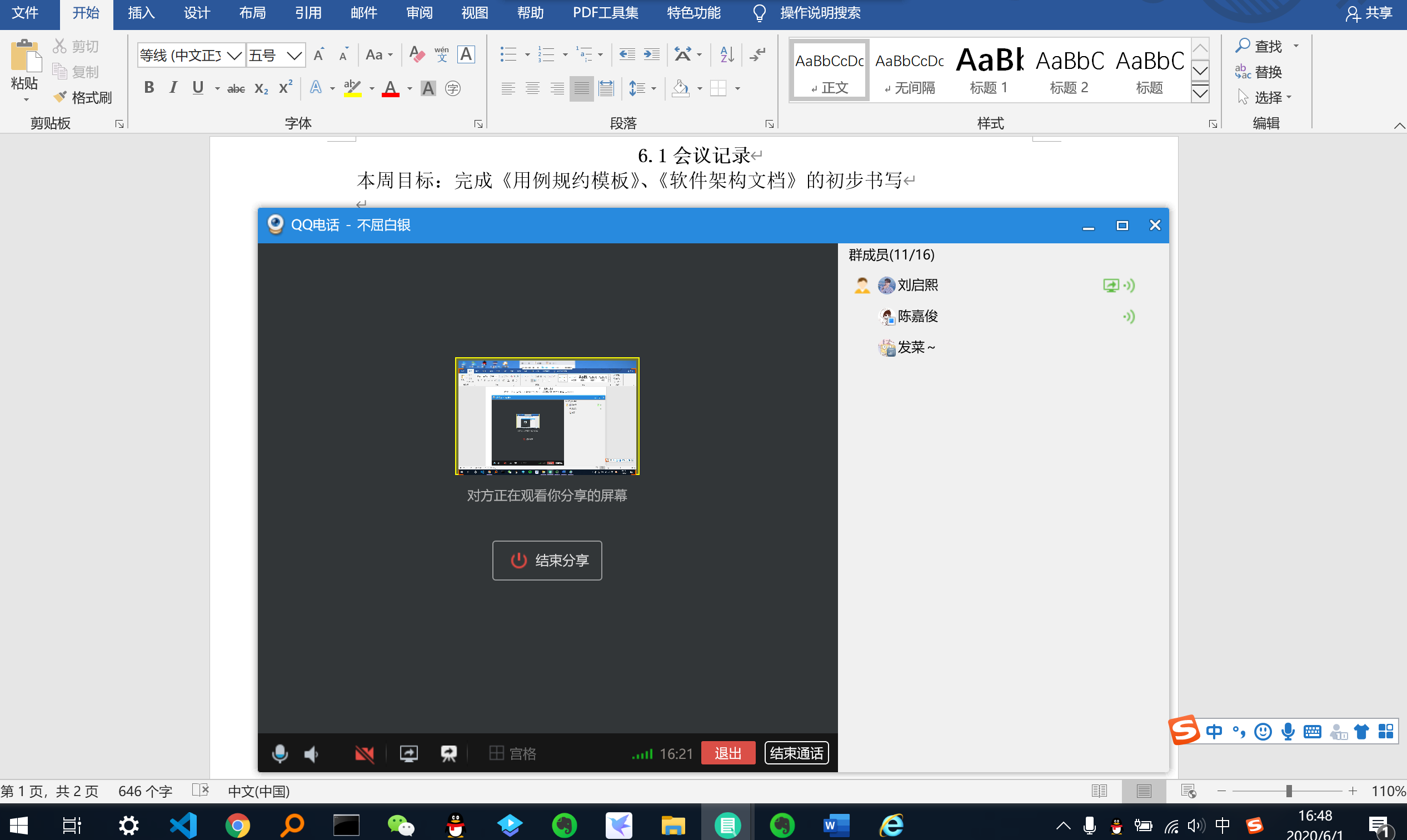
项目名称：白银电商网站

（仓库地址：https://github.com/baebae996/Database-project.git）

1. 本周工作进度
2. 根据周一发布的任务，在原来的基础上增加了商家管理的新需求，包括：修改商品规格、上架商品、下架商品等；
3. 已完成《用例规约模板》和《软件架构文档》的初始版本；
4. 已完成商城首页和商品详情页面的静态布局；
5. 已完成注册登录页面的测试；
6. 初步完成了数据表结构的设计。
7. 数据库系统的设计

为了满足我们制定的一系列需求，并达到高效管理商家和商品的目的，我们花了很多时间讨论数据表的构成以及前后端交换数据的大致方案。以下是我们初步确定的数据表的具体结构。

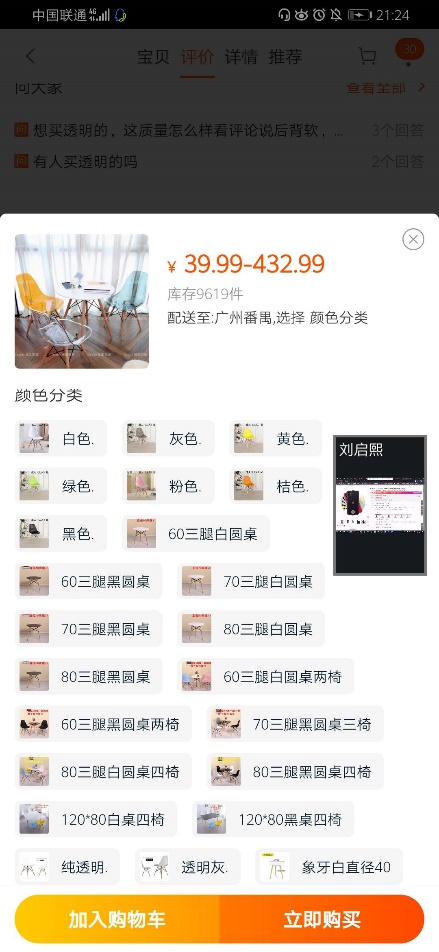
* **用户信息**（用户ID，密码）
* **商家信息**（商家ID，密码）
* **SPU表**（SPUID，商品名称）
* **商品表**（SPUID，商品ID，商家ID，库存，销量）
* **商品规格表**（商品ID，商家ID，属性列表）
* *属性列表举例：{“存储”：128G；“颜色”：黑色}*
* **销量信息表**（SPUID，商家ID，销量）
* **订单信息**（订单ID，订单金额，用户ID，商品ID，商品名称，属性列表，商品数量，创建时间，订单状态）



由于时间关系，我们选取了一个有代表性的实例：京东和淘宝商品详情页中，商品规格设计的对比。



如上图所示，在商品详情页中，京东将商品的每个属性都单独排成一行，用户在将商品加入购物车之前，需要对每一个属性都作出选择，以iphone11为例，用户需要选择颜色、存储大小、套装、保障等等。这种排列方式与“商品类别表（SPUID，商品名称，属性）{“存储”：[128G,256G]，“颜色”：[黑色，白色，黄色]}”的设计理念一致，一个商品详情页对应这个表中的一条记录，优点在于：用户可以清晰地看到属性的所有选项，并且数据表的冗余程度不会特别高；缺点在于：不是任何类型的数据库都支持{key:[value,value],key:[value,value]}这种存储格式，并且，当某种属性的组合无库存或不存在时，用户就需要重新选择属性，一定程度上降低了用户体验，例如：用户可能在选择了“紫色”之后，才发现这家商店没有提供“紫色+256G”这样的组合。



而在移动端的淘宝中，系统将有库存的属性都进行了排列组合，例如“60黑圆桌”、“70白圆桌”等等，用户找到自己心仪的属性组合，就可以将其加入购物车。这种排列方式与“**商品规格表**（商品ID，商家ID，属性列表）属性列表采用JSON格式存储{存储：128G，颜色：黑色}”的设计理念一致，每个最小库存单元都对应这个表中的一条记录，优点在于：用户只需做一次选择即可将商品加入购物车，不会出现像京东那样属性组合不存在的情况，且数据表的实现相对简单；缺点在于：当属性的选项个数较多时，可能会产生较多的属性组合，一定程度上增加了数据表的空间开销。