《数据库系统原理》课程设计

系统实现报告

题目名称：“物外”校园二手交易平台

学号及姓名： 16061200 陈治齐

16061202 胡俊崧

2018年12月14日

# 系统功能需求分析

用户进行交易的大致模式如下：

1. 卖家将想要出售的商品进行登记，包括商品名，售价，数量，详细描述，商品图片。
2. 用户进入交易平台，搜索自己感兴趣的商品或者通过浏览商品列表找到商品，进行下单购买。
3. 用户下单之后生成订单，之后卖家确认订单并发货，卖家收货之后订单关闭。在未收到货前，用户可以取消订单。

经过分析,对于不同类型的用户，对应的具体功能描述如下：

1. 访客：
2. 浏览、查找商品条目，查看商品详细信息
3. 登陆或注册（填写个人信息姓名、联系方式、地址等）成为会员
4. 会员：
5. 作为买家：
6. 浏览、查找商品条目，查看商品详细信息
7. 收藏商品，为商品点赞
8. 评论商品
9. 查看、修改个人信息
10. 给个人账户充值
11. 查看卖家个人信息，与卖家交流
12. 下单购买，查看物流信息，确认收货或者取消订单
13. 作为卖家：
14. 发布商品，查看、修改、删除商品
15. 查看订单，确认订单以及处理发货
16. 管理员：
17. 浏览商品信息,查看用户信息
18. 接受用户请求，处理订单信息
19. 审核编辑商品信息

# 系统功能结构设计

系统的功能主要有用户管理、商品管理、订单管理和评论管理四大部分，如下图所示。



用户管理部分包括用户的注册和登录，用户名、更换等基本信息的修改，以及账户余额充值。

商品管理分为前台和后台两部分，前台主要有商品的展示和商品的收藏，后台部分包括商品发布、修改、删除以及管理商品的分类。

订单管理有订单创建、确认和取消。评论管理部分，可以发表评论，展示已经发表的评论。

用户可以在商品展示页面收藏或者购买商品，个人中心页面可以查看订单，进行订单的确认或者取消，也可以查看已收藏的商品，在收藏页面购买商品或者取消收藏商品。在商品页面，除了可以看到商品的基本信息外，还可以查看到不同用户对商品的评论，商品被收藏的次数，以及拥有该商品的卖家在平台上的在售商品数和交易数。在后台管理页面，用户可以查看、增加、修改或删除自己的商品，也可以为自己的商品添加新的分类。

# 数据库基本表的定义

1. 创建数据库：数据库名为“db\_course\_design”，使用UTF8编码格式。

**CREATE** **DATABASE** db\_course\_design **DEFAULT** CHARSET UTF8 **COLLATE**

UTF8\_GENERAL\_CI;

1. 用户信息表：用户的主要信息，包括用户编号、用户名，登录时使用的邮箱和密码。要求用户名和邮箱唯一。

**create** **table** **if** **not** **exists** `user\_main` (

`id` int(11) **unsigned** **NOT** NULL auto\_increment,

`name` varchar(20) **NOT** NULL **UNIQUE** **DEFAULT** '',

`email` varchar(50) **NOT** NULL **UNIQUE** **DEFAULT** '',

`password` char(32) **NOT** NULL **DEFAULT** '',

**PRIMARY** **KEY**(`id`)

) **default** charset=utf8;

1. 用户详细信息表：包括用户性别、账户余额、账户创建时间和上一次更新时间。表中id是此表的主码也是外码。

**create** **table** **if** **not** **exists** `user\_detail` (

`id` int(11) **unsigned** **NOT** NULL,

`sex` varchar(6) **DEFAULT** '',

`balance` DOUBLE **NOT** NULL **DEFAULT** 999.9,

`create\_time` int(11) **unsigned** **NOT** NULL **default** 0,

`update\_time` int(11) **unsigned** **NOT** NULL **default** 0,

**PRIMARY** **KEY** (`id`),

**FOREIGN** **KEY** (`id`) **REFERENCES** user\_main(`id`)

) **default** charset=utf8;

1. 商品分类表：包括分类的编号、分类名以及该分类的备注。要求分类名唯一，因此候选码可以是分类号或者分类名。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** category (

category\_no INT **UNSIGNED** **NOT** NULL AUTO\_INCREMENT,

name VARCHAR(20) **NOT** NULL **UNIQUE**,

category\_comment VARCHAR(100) **NOT** NULL,

**PRIMARY** **KEY** (category\_no)

) **default** charset=utf8;

1. 物品信息表：包括物品号、物品名、详细描述、价格、数量、收藏人数和分类编号。要求物品名唯一。物品号是主码，分类编号是外码。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** product (

product\_id INT **UNSIGNED** **NOT** NULL AUTO\_INCREMENT,

product\_name VARCHAR(50) **NOT** NULL **UNIQUE**,

detail TEXT **NOT** NULL,

price DOUBLE **NOT** NULL,

quantity INT **UNSIGNED** **NOT** NULL,

likes INT **UNSIGNED** **NOT** NULL **DEFAULT** 0,

category\_no INT **UNSIGNED** **NOT** NULL,

**PRIMARY** **KEY**(product\_id),

**FOREIGN** **KEY** (category\_no) **REFERENCES** category(category\_no)

) **default** charset=utf8;

1. 收藏表：用户对某件物品的收藏信息，包括物品号和用户号，候选码是全码，物品号、用户号均是外码。要求用户被删除或者产品被删除时，此表中有关信息也被删除。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** favorites (

product\_id INT **UNSIGNED** **NOT** NULL,

user\_id INT **UNSIGNED** **NOT** NULL,

**PRIMARY** **KEY** (product\_id, user\_id),

**FOREIGN** **KEY** (product\_id) **REFERENCES** product(product\_id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (user\_id) **REFERENCES** user\_main(id) **ON** **DELETE** **CASCADE**

) **default** charset=utf8;

1. 用户与产品联系表：此表记录每个用户发布的商品，包括用户号和产品号，候选码是全码，物品号、用户号均是外码。要求用户被删除或者产品被删除时，此表中有关信息也被删除。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** user\_product (

user\_id INT **UNSIGNED** **NOT** NULL,

product\_id INT **UNSIGNED** **NOT** NULL **UNIQUE**,

**PRIMARY** **KEY** (user\_id, product\_id),

**FOREIGN** **KEY** (user\_id) **REFERENCES** user\_main(id) **ON** **DELETE** **CASCADE**,

**FOREIGN** **KEY** (product\_id) **REFERENCES** product(product\_id) **ON** **DELETE** **CASCADE**

) **default** charset=utf8;

1. 物品评论表：有评论编号、用户号、产品号、评论内容、评论时间。评论编号是主码，用户号和产品号均是外码，产品被删除时，有关评论也会被删除。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** product\_comment (

comment\_id INT **UNSIGNED** **NOT** NULL AUTO\_INCREMENT,

user\_id INT **UNSIGNED** **NOT** NULL,

product\_id INT **UNSIGNED** **NOT** NULL,

content VARCHAR(500) **NOT** NULL,

comment\_time DATETIME **NOT** NULL,

**PRIMARY** **KEY** (comment\_id),

**FOREIGN** **KEY** (user\_id) **REFERENCES** user\_main(id),

**FOREIGN** **KEY** (product\_id) **REFERENCES** product(product\_id) **ON** **DELETE** **CASCADE**

) **default** charset=utf8;

1. 订单表：包括订单号、买家用户号、产品号、下单时间、数量、成交价格、订单状态。订单号是主码，买家用户号和产品号是外码。

**CREATE** **TABLE** **IF** **NOT** **EXISTS** orders (

order\_id INT **UNSIGNED** **NOT** NULL AUTO\_INCREMENT,

buyer\_id INT **UNSIGNED** **NOT** NULL,

order\_time DATETIME **NOT** NULL,

product\_id INT **UNSIGNED** **NOT** NULL,

quantity INT **UNSIGNED** **NOT** NULL,

price DOUBLE **NOT** NULL,

status VARCHAR(10) **NOT** NULL,

**PRIMARY** **KEY** (order\_id),

**FOREIGN** **KEY** (buyer\_id) **REFERENCES** user\_main(id),

**FOREIGN** **KEY** (product\_id) **REFERENCES** product(product\_id)

)**default** charset=utf8;

# 触发器的定义与实现

1. 订单状态改变触发器：当订单状态由“未完成”变为“已取消”时，调用取消订单过程，当订单状态由“未完成”变为“已完成”时，调用确认订单过程。订单状态变为“已完成”或者“已取消”后，不能再改变，这一约束由应用程序来保证。

**CREATE** **TRIGGER** t\_order\_status\_changed

AFTER **UPDATE** **ON** orders

**FOR** **EACH** ROW

BEGIN

**IF** new.status='已取消' **AND** old.status='未完成' **THEN**

**CALL** cancel\_order(new.buyer\_id, new.product\_id);

**ELSEIF** new.status='已完成' **AND** old.status='未完成' **THEN**

**CALL** confirm\_order(new.buyer\_id, new.product\_id);

END **IF**;

END;

1. 物品收藏触发器：当添加收藏时，更新对应产品信息表中的收藏数量likes。其值应加1。

**CREATE** **TRIGGER** t\_add\_favorite

AFTER **INSERT** **ON** favorites

**FOR** **EACH** ROW

BEGIN

**UPDATE** product SET likes=likes+1 **WHERE** product\_id=new.product\_id;

END;

1. 取消收藏触发器：当取消收藏时，更新对应产品信息表中的收藏数量likes，其值应减1。

**CREATE** **TRIGGER** t\_cancel\_favorite

AFTER **DELETE** **ON** favorites

**FOR** **EACH** ROW

BEGIN

**UPDATE** product SET likes=likes-1 **WHERE** product\_id=old.product\_id;

END;

# 存储过程的定义与实现

1. 创建订单过程：该过程有4个参数，输入的参数分别是买家用户号、物品号、购买数量，输出的参数是订单创建的结果。

插入订单表项前需要先检查买家账户余额是否足够购买所需物品、物品数量是否满足买家需求，如果任一个条件不成立，则不生成订单，返回出错信息“买光了”或者“余额不足”。如果符号创建订单的条件，则将订单表项插入订单表，更新物品余量和买家账户余额。如果买家收藏了该物品，则将该物品从收藏中删除。

**CREATE** **PROCEDURE** create\_order(**IN** bid INT **UNSIGNED**, **IN** pid INT **UNSIGNED**,

**IN** qty INT **UNSIGNED**, **OUT** msg VARCHAR(10))

BEGIN

**DECLARE** oprice DOUBLE;

**DECLARE** oquantity INT **UNSIGNED**;

**DECLARE** nquantity INT **UNSIGNED**;

**DECLARE** obalance DOUBLE;

**DECLARE** nbalance DOUBLE;

**DECLARE** userid INT **UNSIGNED**;

**SELECT** price, quantity **INTO** oprice, oquantity **FROM** product **WHERE** product\_id = pid;

**SELECT** balance **INTO** obalance **FROM** user\_detail **WHERE** id=bid;

SET nbalance = obalance – oprice \* qty;

SET nquantity = oquantity - qty;

**IF** nquantity < 0 **THEN**

SET msg = '卖光了';

**ELSEIF** nbalance < 0 **THEN**

set msg = '余额不足';

**ELSE**

**INSERT** **INTO** orders **VALUES** (null, bid, **CURRENT\_TIME**(), pid, qty, oprice, '未完成');

**UPDATE** product SET quantity=nquantity **WHERE** product\_id=pid;

**UPDATE** user\_detail SET balance=nbalance **WHERE** id=bid;

**SELECT** user\_id **INTO** userid **FROM** favorites **WHERE** product\_id=pid **AND** user\_id=bid;

**IF** !ISNULL(userid) **THEN**

**DELETE** **FROM** favorites **WHERE** product\_id=pid **AND** user\_id=bid;

END **IF**;

SET msg = 'success';

END **IF**;

END;

1. 取消订单过程：过程有两个输入参数，分别是买家用户号和物品号。可以通过设置订单的状态为“已取消”，由触发器调用此过程完成订单的取消。取消订单将创建订单时修改的产品余量和买家账户余额恢复到之前的状态。

**CREATE** **PROCEDURE** cancel\_order(**IN** bid INT **UNSIGNED**, **IN** pid INT **UNSIGNED**)

BEGIN

**DECLARE** oprice DOUBLE;

**DECLARE** qty INT **UNSIGNED**;

**SELECT** quantity, price **INTO** qty, oprice **FROM** orders **WHERE** buyer\_id=bid **AND** product\_id=pid;

**UPDATE** user\_detail SET balance=balance+oprice\*qty **WHERE** id=bid;

**UPDATE** product SET quantity=quantity+qty **WHERE** product\_id=pid;

END;

1. 确认订单过程：过程有两个输入参数，分别是买家用户号和物品号。可以通过设置订单的状态为“已完成”，由触发器调用此过程完成订单的确认。确认订单将买家支付款加入到卖家账户中。

**CREATE** **PROCEDURE** confirm\_order(**IN** bid INT **UNSIGNED**, **IN** pid INT **UNSIGNED**)

BEGIN

**DECLARE** oprice DOUBLE;

**DECLARE** seller\_id INT **UNSIGNED**;

**DECLARE** qty INT **UNSIGNED**;

**SELECT** quantity, price **INTO** qty, oprice **FROM** orders **WHERE** product\_id=pid **AND** buyer\_id=bid **LIMIT** 1;

**SELECT** user\_id **INTO** seller\_id **FROM** user\_product **WHERE** product\_id=pid;

**UPDATE** user\_detail SET balance=balance+oprice\*qty **WHERE** id=seller\_id;

END;

# 系统实现结果

# 总结