

内容销售系统

姓名：张军

学校： 华中科技大学

部门： 邮件事业部

职位： Java 工程师

目录

- 1. 技术环境.....3
- 2. 功能分析：3
- 3. 数据库设计：3
 - 3.1 E-R 图4
 - 3.2 数据库建表语句.....4
 - 3.3 数据库表结构.....5
- 4. 后台模型设计.....6
- 5. 功能实现：6
 - 5.1 登录功能：6
 - 5.2 展示功能.....6
 - 5.3 发布功能.....7
 - 5.4 查看功能.....7
 - 5.5 商品编辑功能.....8
 - 5.6 商品购买功能.....8
 - 5.7 账务功能.....8

1.技术环境

主要的使用的技术：

前端技术： freemarker + js+ css

后端技术： SpringMVC + myBatis + mysql

服务器： tomcat

2.功能分析：

根据具体的需求， 系统主要是包含了买家和卖家两种用户， 共同的需求包括： 系统登录， 商品浏览。 买家的需求包括： 购买， 查看购买记录， 结算， 以及账务。 卖家的需求包括了： 发布， 查看商品的购买情况， 以及编辑商品信息。

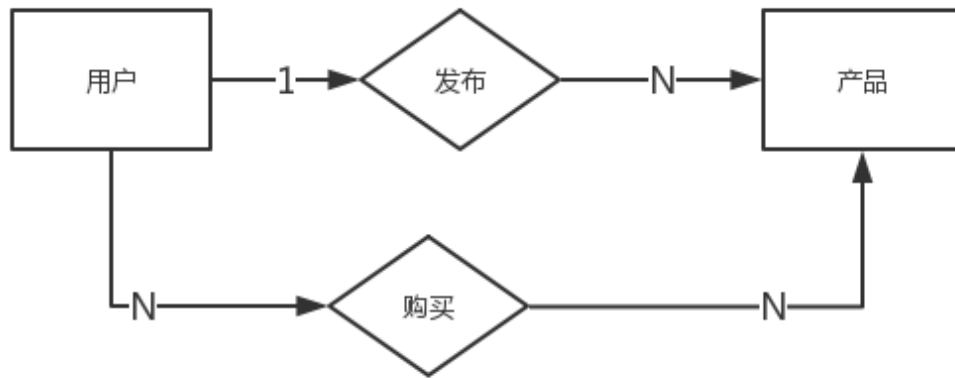
为了扩展方便： 本系统将不分卖家和买家， 用户同时具有两种身份。但是实现上面买家和卖家所需的全部功能， 系统将具有更强的可扩展性。

3.数据库设计：

内容销售系统中， 包含了用户， 产品 和订单。在 E-R 图中， 买家和产品之间的关系是 N-N 的关系， 卖家和产品之间的关系是： 1-N 的关系。所以数据库设计的过程中需要使用订单表保存买家和产品之间的关系， 在产品表中设立关键字 ownerId， 保存发布者的 Id 信息。

3.1 E-R 图

具体的 E-R 图如下所示：



3.2 数据库建表语句

数据库建表语句如下所示：

```
CREATE TABLE `userTable` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `userName` varchar(100) NOT NULL,  
  `password` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `orderTable` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `userId` int(11),  
  `productId` int(11),  
  `num` int(10) DEFAULT 1,  
  PRIMARY KEY (`id`),  
  foreign key(userId) references userTable(id),  
  foreign key(productId) references productTable(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `productTable` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `ownerId` int(11) NOT NULL,
```

```

`productName` varchar(80) DEFAULT NULL,
`price` DECIMAL(10,2),
`stock` int(10) DEFAULT 1,
`summary` varchar(140) DEFAULT NULL,
`detail` varchar(1000) DEFAULT NULL,
`photoAddress` varchar(500),
foreign key(ownerId) references userTable(id),
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

3.3 数据库表结构

建立之后，MySQL 中的表结构：

User 表：

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
userName	varchar(100)	NO		NULL	
password	varchar(100)	YES		NULL	

3 rows in set (0.00 sec)

Product 表：

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
ownerId	int(11)	NO	MUL	NULL	
productName	varchar(80)	YES		NULL	
price	decimal(10,2)	YES		NULL	
stock	int(10)	YES		1	
summary	varchar(140)	YES		NULL	
detail	varchar(1000)	YES		NULL	
photoAddress	varchar(500)	YES		NULL	

8 rows in set (0.00 sec)

OrderTable 表

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
userId	int(11)	YES	MUL	NULL	
productId	int(11)	YES	MUL	NULL	
num	int(10)	YES		1	
orderTime	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
price	decimal(10,2)	YES		NULL	

5 rows in set (0.00 sec)

4.后台模型设计

在该系统中,设计的业务模型重要包括了:User, Product, Order, Cart, 以及 CartProduct 类, 分别对应业务场景的用户, 商品, 订单, 购物车, 购物车商品信息。相应的实现代码见 com.netease.meta 包。

5.功能实现:

下面对系统中主要功能模块的设计实现方式进行简单的介绍。

5.1 登录功能:

登录功能中, 密码的加密使用 jquery.md5 实现对密码的加密, 数据库后端直接保存加密之后的信息。

登录之后, 系统使用 Session 保存登录的用户的个人信息, 主要是用户名, 和当前用户是否登录。

5.2 展示功能

展示功能用于向用户展示商品, 其分为两种情况, 每种情况分为三种类型。

- a) 当用户登录时，展示页面将分别展示：所有商品，用户已购买的商品，和用户发布的商品。 每种类型的商品展示在前端以 Tab 键的方式进行切换，当切换时，通过 ajax 的方式获取相应的信息。
- b) 当用户未登录时，主页面只展示所有商品。另外的两种类型 tab 键激活时，提示用户登录，以查看相应的信息

5.3 发布功能

发布功能主要为用户发布新的商品，商品信息以表单的方式提交，并保存在数据库中。商品的图片信息分为本地和网络图片两种。数据库中只保存图片的地址信息。当商品信息以本地图片的方式进行上传的时候，图片上传并保存到服务器的特定文件夹中。

5.4 查看功能

查看功能是共用户浏览商品的信息。

- c) 当用户未登录时，商品查看页面显示正常页面，同时提供添加购物车的功能。
- d) 当用户登录之后，根据商品和用户之间的关系进行显示。如果用户发布的该商品，就显示编辑按钮， 如果购买了，就显示已购买。

5.5 商品编辑功能

商品编辑功能供用户对商品的信息进行修改，编辑页面默认显示原有商品的信息，以方便用户进行信息修改。

5.6 商品购买功能

购物篮的功能是基于后端的 Session 来实现的，当用户添加商品到购物篮时，提醒是否添加，若用户添加，则以 ajax 的方式发送商品添加信息到后台。

后台并没有实际保存购物篮的数据库结构，个人认为购物篮时一种瞬时的记录信息，大部分用户会在一个 Session 期间将购物篮的商品购买。所以更合适的购物篮技术应该是 Session+ redis 缓存。使用缓存可以减少与数据库的交互，提升系统的性能。在实际的系统中，暂时是仅仅使用 Session 技术实现的简单购物篮功能

用户下单的时候，会以事务的方式同时生成订单和更新商品的库存数量。

5.7 账务功能

账务功能提供用户浏览历史的购买记录，后台主要是通过多表（order, product）进行联合查询，得到响应的购买信息。