



## 《数据库系统原理》课程设计

### 系统实现报告

题目名称： “乐购” 团购平台

学号	姓名	班级
13071079	卢嵩	140612
14061034	王奕夫	140612

2017 年 1 月

## 目录

目录 .....	1
一、 实现环境 .....	3
1. 前端 .....	3
2. 后端 .....	3
二、 系统功能结构图 .....	3
1. 系统功能需求分析 .....	3
1) 注册与登陆 .....	3
2) 用户信息管理 .....	3
3) 商家后台管理功能 .....	4
4) 商品检索与购买功能 .....	4
5) 评价反馈功能 .....	4
2. 系统功能结构设计图 .....	4
三、 基本表定义 .....	4
1. 管理员信息表 (admin) .....	5
2. 用户信息表 (users) .....	5
3. 商户信息表 (shop) .....	5
4. 商品信息表 (item) .....	6
5. 类别信息表 (class) .....	6
6. 团购订单表 (orders) .....	6
7. 评论信息表 (comment) .....	7
8. 订单详情表 (iteminorder) .....	7
四、 安全性设计 .....	7
1. 前端 .....	7
2. 后端 .....	7
五、 存储过程、触发器和函数 .....	8
1. 存储过程 .....	8
1) 商品数量设置存储过程 (setrest) .....	8
2) 商品评价统计存储过程 (itemcomment) .....	8
3) 订单管理存储过程 (orderManage) .....	9
2. 触发器 .....	10
1) 商品评分更新触发器 (item_star_update) .....	10

2) 商家评分更新触发器 (shop_star_update) .....	10
3) 用户年龄检查触发器 (check_user_age) .....	11
4) 商品详情检查触发器 (check_item_num) .....	11
5) 订单价格更新触发器 (order_price_update) .....	11
6) 用户支出统计触发器 (user_outcome_adapter) .....	12
六、 实现技术.....	12
1. 前端技术.....	12
2. 后端技术.....	12
七、 运行实例.....	12
1. 主页页面.....	13
2. 登陆界面.....	13
3. 商品页面.....	14
4. 用户信息预览及设置.....	16
5. 商家管理页面.....	17
6. 搜索页面.....	19
7. 管理员登陆及管理接口.....	20
八、 源程序简要说明.....	22
1. 前端程序.....	22
2. 后端程序.....	23
九、 收获和体会.....	24

## 一、 实现环境

### 1. 前端

编程环境：Microsoft Windows 10 64bit

### 2. 后端

编程环境：Microsoft Windows 10 64bit

服务器： Apache Server v 2.4.23 本地部署

数据库：MySQL + PHPMyAdmin v 4.5.1

PHP 版本：PHP 5.6.24

## 二、 系统功能结构图

### 1. 系统功能需求分析

“乐购”团购平台的设计目标是：方便顾客在网上完成团购工作，享受线上折扣；为商家提供在线的商品展示、推广与销售渠道；顾客可以给购买过的商品进行评价。为此，本系统应该具有以下几个功能：

#### 1) 注册与登陆

用户在使用本网站之前需要首先进行注册，否则将不能够使用完整的购物流程。在进行注册时，用户需要输入自己注册的用户名、密码以及邮箱等信息，不能够使用同一个邮箱进行重复注册。在注册完成后，用户便可以通过登录界面进行登陆，并可以使用网站的全部购物功能。

商家在本平台进行商品销售之前需要首先进行商家注册。在进行注册时，商家需要输入自己注册的用户名、密码以及邮箱等信息，不同的商家不能够使用相同的邮箱重复注册。在注册完成之后，商家可以通过登录界面进行登陆。登陆之后的商家可以随时查看自己的信息并作出修改，还可以看到自己正在出售的商品以及销量信息。

本系统设置后台管理员，管理员可以对自己负责类别板块的所有商品进行编辑修改，还可以查看所有商家的具体信息。

从上面三点可以看出，本系统具有用户、商家和管理员三种角色。三种角色分别可以使用不同的方式进行登陆，并且分属不同的角色组，具有不同的操作权限。因而本系统具有权限管理的功能，能在一定程度上保证系统的安全性。

#### 2) 用户信息管理

用户在完成登陆之后可以查看并完善自己的个人信息，此外，个人信息将会统计该用户在本网站的总支出为多少。此外，用户在进行登陆之后还可以进行历史订单的查询，查看自己曾经购买过

的商品。

### 3) 商家后台管理功能

商家在完成登陆之后，可以选择添加新的商品。在添加商品的时候，需要提供商品的名称、价格、原价、类别、库存等信息，还可以提供商品的描述和图片。如果商品的库存不足将无法继续出售，此时商家可以选择删除该商品或者修改商品信息中的库存属性以恢复销售。此外，登陆后的商家还可以进行商户信息的修改。

### 4) 商品检索与购买功能

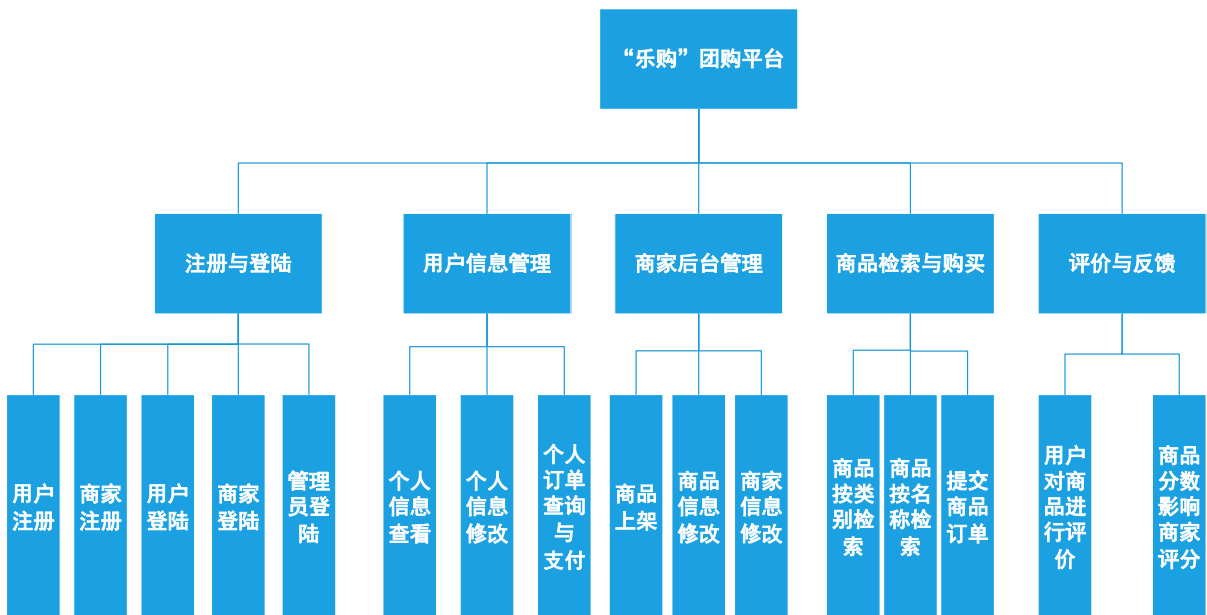
用户在进行登陆之后可以进行商品的购买。购买商品需要首先进入商品界面，然后选择需要购买的数量，最后点击下单。完成下单之后，系统将会生成一个订单，订单中包含购买商品的数量以及商品的信息。用户可以在个人信息界面看到自己曾经下过的订单，并得到自己过去的总支出。

此外，本网站还支持商品检索功能。用户可以选择自己想要选择商品的信息或者输入商品的关键字，以实现商品的查询功能。

### 5) 评价反馈功能

用户在完成商品购买之后可以进行商品评价。用户的评分将会直接展示在商品页面和商家的页面上。商品的评分由该商品的所有评价的平均值决定，商家的评分由该商家的所有商品决定。

## 2. 系统功能结构设计图



## 三、 基本表定义

根据之前的设计文档以及规范化原则，本系统采用共采用了 8 个基本表，它们分别为：管理员信息表 (admin)，用户信息表 (users)，商户信息表 (shop)，商品信息表 (item)，类别信息表 (class)，团购订单表 (orders)，订单详情表 (iteminorder) 和评论信息表 (comment)。各基本表定义如下，其

中的所有关系均已规范化到 3NF 以上。

### 1. 管理员信息表（admin）

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
admin_id	int	11	是	是	主键	管理员 ID
admin_pw	varchar	30	是	否	否	管理员密码
admin_class	int	11	是	否	外键	管理员类别

表 3.1 管理员信息表

### 2. 用户信息表（users）

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
user_id	int	11	是	是	主键	用户 ID
user_name	varchar	50	是	否	否	用户名
user_email	varchar	30	否	否	UNIQUE	用户邮箱
user_password	varchar	30	是	否	否	用户密码
user_addr	varchar	200	否	否	否	用户地址
user_reg_date	date	~	是	否	否	注册日期
user_phone	bigint	11	否	否	否	用户电话
user_image	varchar	200	否	否	否	用户头像
user_age	int	11	否	否	否	用户年龄
user_outcome	int	11	是	否	否	用户支出

表 3.2 用户信息表

### 3. 商户信息表（shop）

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
shop_id	int	11	是	是	主键	商户 ID
shop_name	varchar	30	是	否	否	商户名称
shop_password	varchar	30	是	否	否	商户密码
shop_phone	bigint	11	否	否	否	商户电话

shop_stars	float	~	是	否	否	商户评级
shop_addr	varchar	50	否	否	否	商户地址
shop_email	varchar	30	是	否	UNIQUE	商户邮箱

表 3.3 商户信息表

#### 4. 商品信息表 (item)

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
item_id	int	11	是	是	主键	商品 ID
item_name	varchar	30	是	否	否	商品名称
item_price	float	~	是	否	否	商品价格
item_origin	float	~	是	否	否	商品原价
item_brief	text	~	否	否	否	商品描述
item_shop	int	11	是	否	外键	商品商家
item_class	int	11	是	否	外键	商品类别
item_image	varchar	100	否	否	否	商品图片
item_stars	floats	~	是	否	否	商品评分
item_rest	int	11	是	否	否	商品库存

表 3.4 商品信息表

#### 5. 类别信息表 (class)

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
class_id	int	11	是	是	主键	类别 ID
class_name	varchar	20	是	否	否	类别名称

表 3.5 类别信息表

#### 6. 团购订单表 (orders)

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
order_id	int	11	是	是	主键	订单 ID
user_id	int	11	是	否	外键	用户 ID
order_time	date	~	是	否	否	成交时间

order_price	float	~	是	否	否	订单价格
-------------	-------	---	---	---	---	------

表 3.6 团购信息表

## 7. 评论信息表（comment）

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
comment_id	int	11	是	是	主键	评论 ID
comment_user	int	11	是	否	外键	用户 ID
comment_item	int	11	是	否	外键	评论商品
comment_time	date	~	是	否	否	评论时间
comment_star	int	11	是	否	否	评论等级
comment_content	text	~	是	否	否	评论内容

表 3.7 评论信息表

## 8. 订单详情表（iteminorder）

字段名称	数据类型	字段属性				
		字段大小	必填字段	主键	索引	说明
item_id	int	11	是	是	主键，外键	商品 ID
order_id	int	11	是	是	主键，外键	订单 ID
number	int	11	是	否	否	商品数量

表 3.8 订单详情表

# 四、 安全性设计

本系统能够满足较为基本的安全性设置，此章将从前端和后端两个方面说明本系统的安全性设计。

### 1. 前端

前后端通信主要使用 POST 方法而不是 GET 方法，能有效防止其他人对网站信息的恶意截取与利用。用户使用用户名和密码登录的过程中，会首先检索用户名和密码填写的完整性，保证了在登录界面不会被渗透。

### 2. 后端

后端的安全性设计主要采用了不同用户登陆的形式。商家、管理员、用户分别以三种不同的用



户身份登陆到数据库中，不同角色之间的操作权限各不相同，这可以在很大程度上保证数据库的结构不被破坏。

此外，在 php 连接后台的函数中，对于特定数据类型的数据加以验证，防止了 SQL 注入所造成的危险。例如，验证某个变量是否为整型，之后再进行相应操作

```
$type = intval($type);
```

或者是将需要检索的变量名使用引号括起来：

```
$result = mysql_query("select * from users where user_email = '$name' and  
user_password = '$pw'", $conn);
```

## 五、 存储过程、触发器和函数

### 1. 存储过程

本平台后端共采用了三个存储过程，下面我将分别对其进行介绍。

#### 1) 商品数量设置存储过程 (setrest)

过程输入：商品 id id，需要设置的商品数量 itemnum

过程输出：操作成功返回 0，否则返回 1

作用：将对应 id 的商品库存数量设置为 itemnum。由于商品的库存不可能为负值，因此首先判断 itemnum 是否大于等于 0，如果是，那么进行赋值操作，输出 0；否则操作不合法，返回 1。

实现代码：

```
CREATE PROCEDURE setrest(IN id int, IN itemnum int, OUT numres tinyint())
BEGIN
    DECLARE num int;
    DECLARE t_error INTEGER DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET t_error=1;
    SET numres = 1;
    START TRANSACTION;
    IF itemnum >= 0 THEN
        UPDATE item SET item_rest = itemnum WHERE item_id = id;
        SET numres = 0;
    ELSE
        SET numres = 1;
    END IF;
    IF t_error = 1 THEN
        ROLLBACK;
    ELSE
        COMMIT;
    END IF;
END
```

图 5.1 商品数量设置存储过程实现代码

#### 2) 商品评价统计存储过程 (itemcomment)

过程输入：商品 id itemid

过程输出：1 星评价的占比 n1, 2 星评价的占比 n2, 3 星评价的占比 n3, 4 星评价的占比 n4, 5

星评价的占比 n5，该商品的评价总数 total

作用：统计 itemid 对应商品的所有评价的数量，并得到各等级评价所占的比例，以方便前端建立报表图形。

实现代码：

```
CREATE PROCEDURE itemcomment(IN itemid int, OUT n1 float, OUT n2 float, OUT n3 float, OUT n4 float, OUT n5 float,
                             OUT total int)
BEGIN
    DECLARE m1 int;
    DECLARE m2 int;
    DECLARE m3 int;
    DECLARE m4 int;
    DECLARE m5 int;
    DECLARE t_error INTEGER DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET t_error=1;
    START TRANSACTION;
    SET total =(select COUNT(*) from comment WHERE comment_item = itemid GROUP BY comment_item);
    SET m1 = (select COUNT(*) from comment WHERE comment_item = itemid AND comment_star = 1 GROUP BY comment_item);
    SET m2 = (select COUNT(*) from comment WHERE comment_item = itemid AND comment_star = 2 GROUP BY comment_item);
    SET m3 = (select COUNT(*) from comment WHERE comment_item = itemid AND comment_star = 3 GROUP BY comment_item);
    SET m4 = (select COUNT(*) from comment WHERE comment_item = itemid AND comment_star = 4 GROUP BY comment_item);
    SET m5 = (select COUNT(*) from comment WHERE comment_item = itemid AND comment_star = 5 GROUP BY comment_item);
    SET n1 = m1 / total;
    SET n2 = m2 / total;
    SET n3 = m3 / total;
    SET n4 = m4 / total;
    SET n5 = m5 / total;
    IF t_error = 1 THEN
        ROLLBACK;
    ELSE
        COMMIT;
    END IF;
END
```

图 5.2 商品评价统计存储过程实现代码

### 3) 订单管理存储过程（orderManage）

过程输入：用户名 username, 商品 id iid, 购买数量 num

过程输出：运行结果 res 0 为执行成功，否则为 1

作用：在用户进行购物操作的时候，根据不同的场景进行不同的订单操作。如果该用户不存在未结算的订单，那么系统将会为此用户新创建一个未结算的订单，之后将用户购买的操作插入到该订单的订单详情之中；如果该用户名下存在未结算的订单，那么系统将会直接在此订单的详情中插入用户现在购买的商品的信息。

实现代码：

```

CREATE PROCEDURE orderManage(IN username varchar(50), IN iid int, IN num int, OUT res tinyint(1))
BEGIN
    DECLARE rest int;
    DECLARE oid int;
    DECLARE t_error INTEGER DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET t_error=1;
    SET res = 1;
    START TRANSACTION;
    SET rest = (SELECT item_rest FROM item WHERE item_id = iid);
    IF num <= rest THEN
        update item set item_rest = (rest-num) where item_id = iid;
        set usid = (select user_id from users where user_name = username);
        set flag = (select count(*) from users,orders where users.user_id = usid and orders.user_id = users.user_id
                    and order_state = '1' GROUP BY users.user_name);
        IF (flag is NULL) THEN
            set res = 2;
            insert into orders (user_id,order_time,order_price,order_state) values (usid,CURRENT_DATE(),0,1);
        END IF;
        set oid = (select order_id from orders where user_id = usid AND order_state = 1);
        set res = 3;
        insert into iteminorder values (iid,oid,num);
        SET res = 0;
    ELSE
        SET res = 1;
    END IF;
    IF t_error = 1
    THEN
        ROLLBACK;
    ELSE
        COMMIT;
    END IF;
END

```

图 5.3 订单管理存储过程实现代码

## 2. 触发器

### 1) 商品评分更新触发器 (item\_star\_update)

作用对象：商品信息表 (item)

触发条件：comment 表插入操作后

功能：在 comment 表中插入了一条新的评论之后，也就是用户对某商品进行评价之后，根据此评价重新计算该商品的评分 (item\_stars)

实现代码：

```

CREATE TRIGGER item_star_update
AFTER INSERT ON comment
FOR EACH ROW BEGIN
    UPDATE item SET item_stars =
        (SELECT AVG(comment_star) FROM comment
         WHERE comment_item = new.comment_item GROUP BY comment_item)
        WHERE item_id = new.comment_item;
END

```

图 5.4 商品评分更新触发器实现代码

### 2) 商家评分更新触发器 (shop\_star\_update)

作用对象：商家信息表 (shop)

触发条件：商品新表做出修改后

功能：根据商品评分的修改更新商家的评分

实现代码：

```
CREATE TRIGGER shop_star_update
AFTER UPDATE ON item
FOR EACH ROW BEGIN
    UPDATE shop SET shop_stars =
        (SELECT AVG(comment_star) FROM item,comment
         WHERE item_id = comment_item AND item_shop = new.item_shop GROUP BY item_shop)
        WHERE shop_id = new.item_shop;
END
```

图 5.5 商家评分触发器实现代码

### 3) 用户年龄检查触发器 (check\_user\_age)

作用对象：用户信息表 (users)

触发条件：在用户表进行更新之前

功能：首先判断输入的年龄是否合法，如果合法，更新用户年龄，否则终止操作

实现代码：

```
CREATE TRIGGER check_user_age BEFORE UPDATE ON users
FOR EACH ROW BEGIN
    IF new.user_age < 0 THEN
        SIGNAL SQLSTATE 'TX000' SET MESSAGE_TEXT = 'age cannot be less than zero';
    END IF;
END
```

图 5.6 用户年龄检查触发器实现代码

### 4) 商品详情检查触发器 (check\_item\_num)

作用对象：订单详情表 (iteminorder)

触发条件：在订单详情表进行更新之前

功能：判断用户购买的商品数量是否合法，如果合法，更新订单详情，否则终止操作。

实现代码：

```
CREATE TRIGGER check_item_num BEFORE UPDATE ON iteminorder
FOR EACH ROW BEGIN
    IF new.number < 0 THEN
        SIGNAL SQLSTATE 'TX000' SET MESSAGE_TEXT = 'item number cannot be less than zero';
    END IF;
END
```

图 5.7 商品详情检查触发器实现代码

### 5) 订单价格更新触发器 (order\_price\_update)

作用对象：订单详情表 (iteminorder)

触发条件：在订单详情表插入新条目之后

功能：根据更新的商品的 id 和数量更新当前商品的价格。

实现代码：

```

CREATE TRIGGER order_price_update AFTER INSERT ON iteminorder
FOR EACH ROW BEGIN
    DECLARE oriprice FLOAT;
    DECLARE itprice FLOAT;
    SET oriprice = (SELECT order_price from orders where order_id = new.order_id);
    SET itprice = (SELECT item_price from item where item_id = new.item_id);
    UPDATE orders SET order_price = (oriprice + itprice * new.number)
    where order_id = new.order_id;
END

```

图 5.8 订单价格更新触发器实现代码

#### 6) 用户支出统计触发器 (user\_outcome\_adapter)

作用对象：订单信息表 (orders)

触发条件：订单信息表进行更新之后

作用：根据更新订单的用户名，更新该用户的总支出情况。

实现代码：

```

CREATE TRIGGER user_outcome_adapter AFTER UPDATE ON orders
FOR EACH ROW BEGIN
    UPDATE users SET user_outcome = (select SUM(order_price) from orders where user_id = new.user_id)
    where user_id = new.user_id;
END

```

图 5.9 用户支出统计触发器实现代码

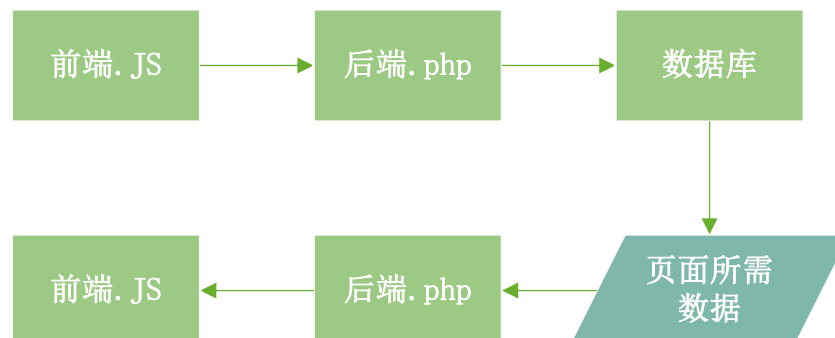
## 六、 实现技术

### 1. 前端技术

整个前端使用了很少 bootstrap 的 css 和 js 样式，基本的样式都是通过 html + css + javascript 手工实现，也运用了少许的 jquery 内容。前后端通信通过 ajax 进行，传递信息格式是 json。

### 2. 后端技术

后端未使用现有的成熟框架，采用 PHP+MySQL 手工搭建。每当前端发来请求的时候，后端会首先解析前端所发来的 JSON 数据，之后进行相应的数据库查询，最终将查询的结果重新打包成 json 格式返还给前端。大致的实现流程如下：



## 七、 运行实例

## 1. 主页页面

主页页面采用 HTML + CSS + JS 实现动态加载，各个分栏中的内容都来源于数据库，可以跟随数据库的变化而变化。未进行登陆以及登陆后的页面会有所不同。

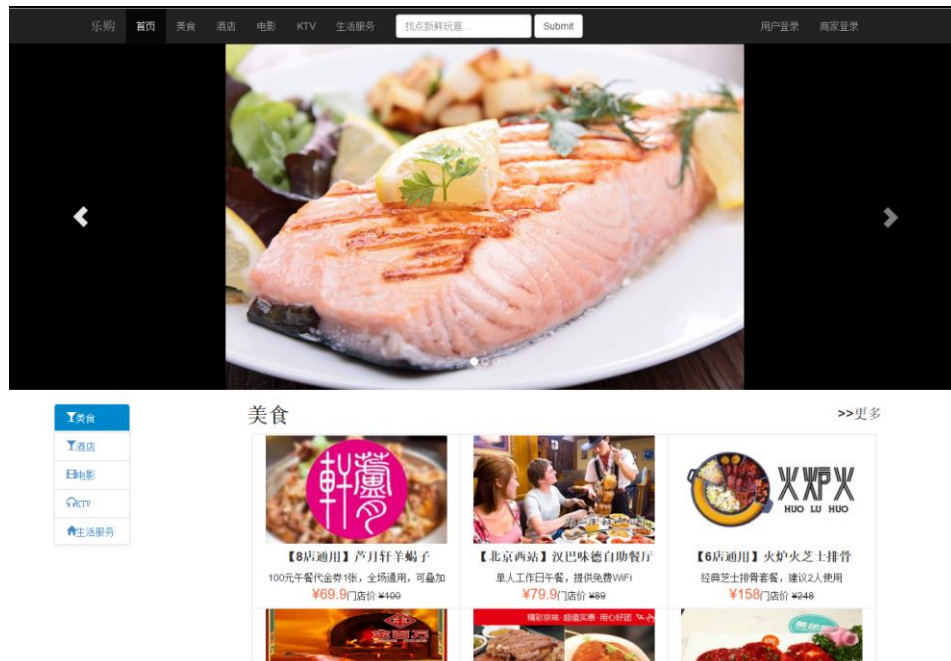


图 7.1 主页页面（未登录）

## 2. 登陆界面

在主页点击登陆按钮之后将会弹出对应的登陆弹窗，在弹窗内输入注册邮箱和密码，然后点击立刻登陆。此时前端将会连接后端的服务器，进行身份校验，在确定输入的密码和登陆的用户一致后方能完成登陆操作。

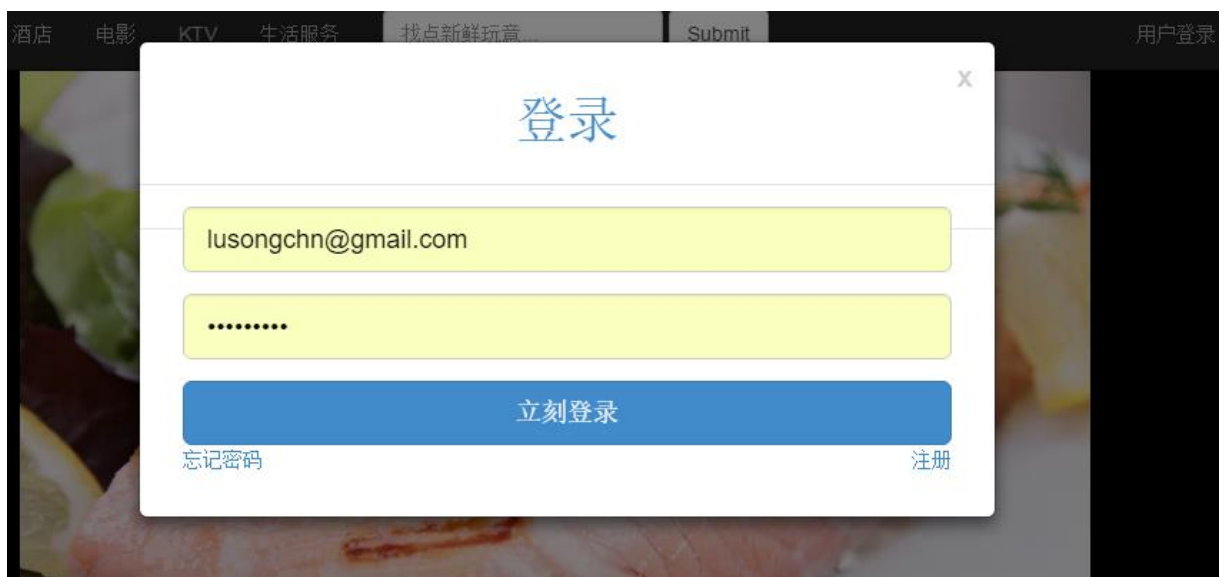


图 7.2 登陆页面

登陆后的主页右上角将会显示用户的用户名或者商家的商家 ID



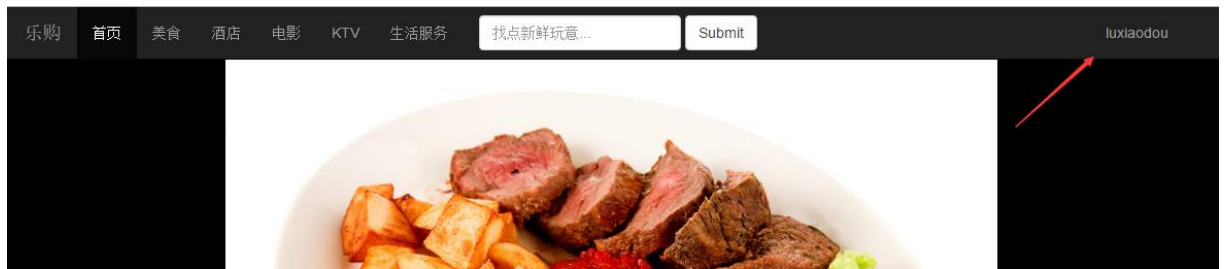


图 7.3 主页（登陆后）

### 3. 商品页面

商品页面可以通过点击主页的商品图片或者通过点击搜索后的结果到达，展示了商品的各个属性和评价。用户可以在此购买商品，也可以对商品进行评价。

如下图所示，页面最上方显示商品的名称和描述，左侧有商品的配图，右侧有商品的优惠价、原价以及库存等信息。下方显示此商品的消费评价等信息，用户也可以通过点击提交评价按钮进行评价的提交。提交后的评价将会显示在页面的最下方。

#### 【8店通用】芦月轩羊蝎子

100元午餐代金券1张，全场通用，可叠加



团购价: **¥69.9** 门店价: ~~¥100~~

库存: 86

3.25分

评价: 8

数量:

- 1 +

✓立即抢购

#### 消费评价

用户评价

提交评价

**3.6分**



2017-01-04

味道不错!

luxiaodou  
3分

图 7.4 商品页面

## 消费评价

用户评价

提交评价

评论测试

评分： 4分 ▼

提交评价

图 7.5 消费评价



#### 4. 用户信息预览及设置

## 个人主页

编辑信息

年龄:

20

邮箱:

lusongchn@gmail.com

电话:

13691515258

地址:

北京市海淀区北京航空航天大学

头像:

http://localhost/img/user.jpg

总支出:

729

图 7.6 个人主页预览

## 编辑个人信息

年龄	<input type="text"/>
邮箱	<input type="text"/>
电话	<input type="text"/>
地址	<input type="text"/>
头像	<input type="text"/>

确认提交

图 7.7 个人信息的设置

## 5. 商家管理页面

商家登陆后即可进入到商品管理页面，商家可以在这里修改自身的信息，也可以在此对商品做出添加，删除，修改操作。

## 商家信息界面

编辑信息

添加商品

名称:

汉巴味德自助餐厅

邮箱:

hanba@gmail.com

电话:

1234567

地址:

北京

收入:

0



【北京西站】汉巴味德自助餐厅



asdf

图 7.8 商家信息界面以及商品管理选项

## 编辑商品信息

商品名称	<input type="text"/>
商品描述	<input type="text"/>
商品原价	<input type="text"/>
商品价格	<input type="text"/>
商品图片	<input type="text"/>
商品库存	<input type="text"/>
商品类型	美食 ▼

确认提交

图 7.9 商品信息编辑页面可以在此编辑商品

## 6. 搜索页面

搜索功能分为按类别搜索以及按照关键字搜索两类，点击页面上方的页面即可跳转到商品的类别搜索界面

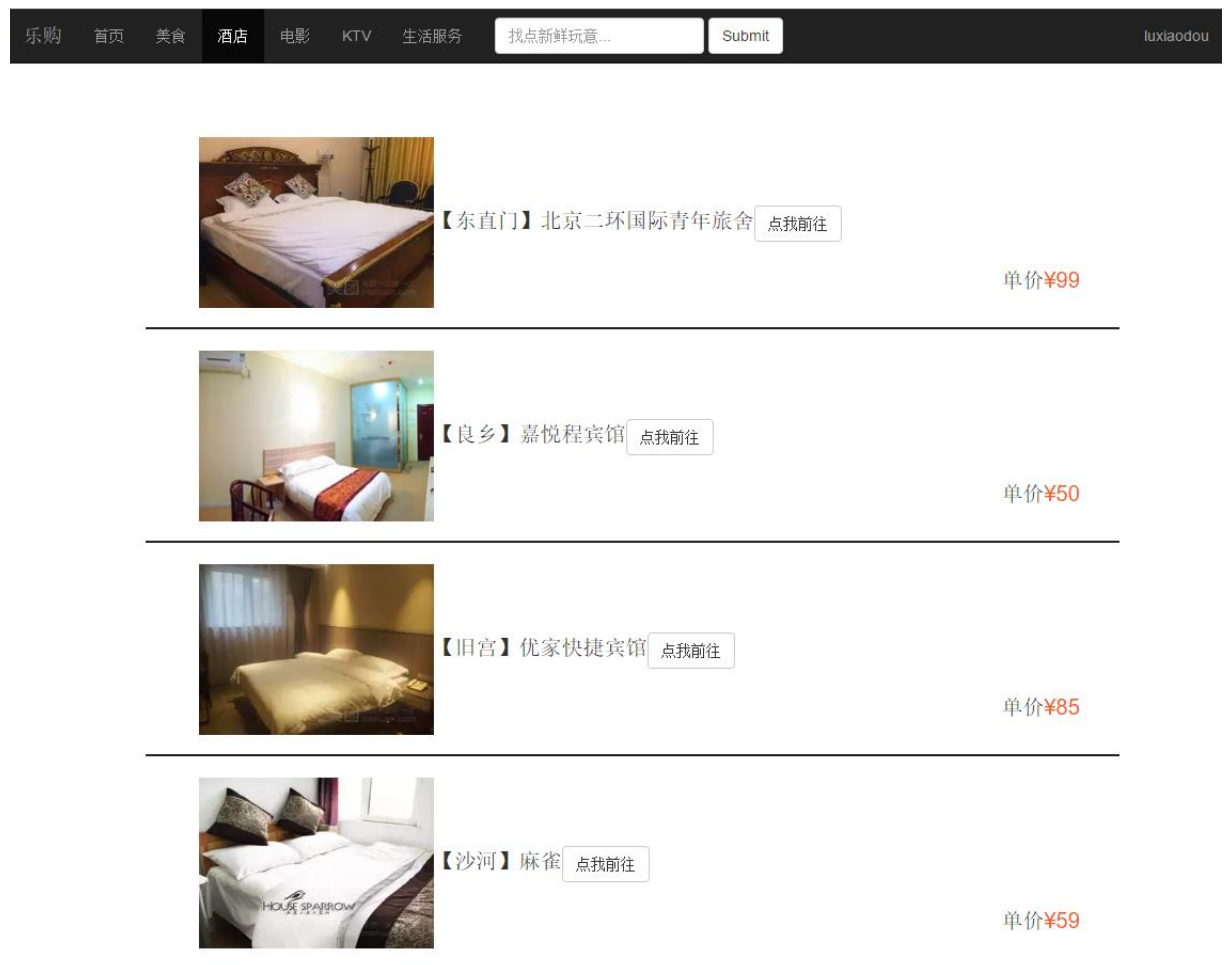


图 7.10 商品类别检索页面

## 7. 购物订单

购物订单上半部分显示未支付的商品，按下显示金额的按钮可以进行支付，一次支付的所有内容算作一个订单。下半部分显示已支付的订单，每个订单用一圈黑色包裹，里面展示了订单具体内容。

## 我的购买清单

未支付的订单：



【8店通用】芦月轩羊蝎子

单价 ¥69.9 X3



【北京西站】汉巴味德自助餐厅

单价 ¥79.9 X1

¥289.6

已支付的订单：

图 7.13

### 8. 管理员登陆及管理接口

## 乐购平台管理员接口

管理员用户账号：

管理员用户密码：

图 7.11 管理员登陆接口

## 商品信息界面



图 7.12 管理员商品管理页面

## 八、源程序简要说明

### 1. 前端程序

前端内容基本以 html 的格式放在根目录下，而每个 html 文件另配备一个 javascript 文件来实现动态加载以及通信。因为 css 优先级的问题，所以大部分 css 样式都嵌在 html 内部来重写 bootstrap 代码。

**index.html:** 主页面，顶端摆放大型横幅，来宣传推广商品。下面左侧有个移动导航条，分类导航到各个栏目的商品，栏目共分五类：美食、酒店、电影、KTV、生活服务。每类各展示几个商品，点击更多会跳转到类别目录下。

**mer.html:** 商品信息界面，上半部分展示商品具体信息，用户提交订单。下半部分展示商品评价并且用户可以自己填写评价。

merchant.html: 列出商户具体信息, 提供按钮进行编辑。

merchantEdit.html: 商户编辑信息界面, 可编辑名称、邮箱、电话、地址。

merEdit.html: 商品信息编辑界面, 可编辑名称、描述、原价、价格、图片、库存、类型。

user.html: 用户个人信息界面, 显示用户年龄、邮箱、电话、地址、头像、总支出。

userEdit.html: 用户个人信息编辑界面, 可编辑用户年龄、邮箱、电话、地址、头像。

register.html: 登录界面。

result.html: 顶部搜索结果界面, 同时也是分类显示界面, 界面每一排显示一个符合结果的商品。

paylist.html: 订单界面, 上面是未支付的订单, 按下方按钮可以进行支付。下面是已支付订单, 每一个订单会展示在一起。每个订单可以有多个不同的商品。

## 2. 后端程序

后端连接数据库所用的函数全部保存在 htdocs/php 文件夹下面, 各文件的具体功能如下:

addComment.php: 创建商品评价的函数

addItem.php: 供商户添加商品所用的函数

adminlogin.php: 管理员登陆功能的函数

createorder.php: 创建用户清单的函数

deleteitem.php: 删除物品的函数

itemcommnet.php: 显示某商品评价的函数

loadindex.php: 加载主页内容的函数

loadinfo.php: 读取个人信息的函数

loadmer.php: mer 页面的动态填写函数

loadshop.php: 读取商家信息的函数

login.php: 用户/商家登陆用函数

mysql.php: 连接数据库所用的函数

pay.php: 处理订单结算的函数

register.php: 实现登陆功能的函数

search.php: 实现搜索功能的函数

shopitem.php: 实现商户页面商品填充的函数

test.php: 单纯的测试用函数

updateitem.php: 更新商品信息的函数

updateshop.php: 更新商户信息的函数

updateuser.php: 更新用户信息的函数

userorder.php: 读取用户所有订单的函数



## 九、收获和体会

通过这次数据库课设的实践，我们对于课上所讲授的数据库知识有了更进一步的了解。不仅亲手从模型开始进行了完整的数据库设计流程，还亲自实现了事务处理、触发器、索引、存储过程等机制，对于课上学习的内容进行了进一步的巩固和加强，同时也深刻地体会到了“实践”对于计算机科学这个学科的重要性。纸上得来终觉浅，绝知此事要躬行。由于刘瑞老师课上讲授的 SQL 语句多为 T-SQL，而且之前的题目大多都是用笔写 SQL，这导致我们的数据库编程经验时期是非常短缺的。对于手写 SQL 语句中的小错误，我们常常会归结于自己的马虎，但是等真正敲起代码来的时候，一个小小的错误都会导致 SQL 语句无法执行。不能熟练使用的结果就是提笔就报错，然后进入漫长的 debug 阶段，在经过比较漫长的熟悉过程之后，开发的效率才慢慢提高了起来。在数据库编程面前，细心，才是保证开发效率的唯一捷径。

从我的个人角度来讲，我认为这次的课设并不简单。二人从零开始学习网站的搭建，从零开始进行数据库的设计，中间实现的过程也充满了坎坷。在设计数据库的时候，我们的网站功能思路还没有定型，这导致了我们的订单基本表在中途进行重新设计，徒增了不少工作量；由于前端同学的经验比较缺乏，加上时间紧迫，在实际运行的过程中可能会出现各种各样的问题，这些问题的调试也给我们带来了不小的压力；后端实现的过程中，由于我之前一直使用的是 SQLServer，发现二者的语法并不完全通用之后，需要自行实现部分功能，这也给后端的开发带来了阻力。虽然实现的过程并非是一帆风顺，但是我们还是团结一致，相互鼓励，最终坚持着完成了整个项目，达到了自己比较满意的效果。

除了数据库和网页编写知识上面的收获之外，我觉得这次数据库课设的实现使我们更加懂得如何与他人进行合作。因为课设的代码量比较大，我们组采用的是前后端有两人分开编写，这对于之前合作编程经验较少的我们来说是比较新鲜的事。我们采用 github 进行了项目的管理，并且协商好了前后端通信的接口。在进行编程的过程中尽可能频繁地进行交流，这使得我们在进行前后端对接的时候并没有遇到太大的障碍。经过不断的磨合，我们的开发效率有了比较显著的提升，而且二人也不断激发彼此的思路，这使得我们的设计变得更加完善。

总的来数，这次数据库课设使我们亲手实践了课上所学到的数据库相关知识，使我们学会了比较基础的网站的构建，同时还提升了我们的结对编程能力。在完成整个项目之后，我们对于自己能力的自信也有了提升，这样我们切身感受到了自己的成长。期间虽然熬了不少夜，但多年之后再回顾当时的努力与付出，相信也会是我们人生中比较难忘的一段回忆吧。