# 图书商城

## 功能分析

1. 用户注册
2. 用户登录
3. 添加商品（CURD）
4. 商品查看—列表查询
5. 商品详情页面
6. 将商品添加购物车
7. 查看购物车
8. 修改购物车
9. 生成订单
10. 订单查看（取消）
11. 在线支付
12. 下载销售榜单

游客（未登录）：注册、登录、商品查看

商城注册用户：商品查看、添加商品到购物车、购物车管理、生成订单、订单管理、在线支付

管理员：添加商品、商品管理、查看订单、榜单查看（导出）

## 2、系统设计

### 1、技术选型（系统架构）

JSTL + JSP + Servlet + JavaBean + BeanUtils + FileUpload + JavaMail + DBUtils(JDBC) + C3P0 + MySQL + MyEclipse10 + Tomcat7.0 + JDK6 + Windows

MVC 模式

JavaEE三层结构

DAO 模式

### 数据库设计

系统存在以下实体：用户、商品、订单、购物车

购物车不放入数据库（Session或者Cookie实现）

用户表

create table users(

id int primary key autu increament,

username varchar(40),

password varchar(100),

nickname varchar(40),

email varchar(100),

role varchar(100),

state int,

activecode varchar(100),

updatetime timestamp );

商品表

create table products(

id varchar(100) primary key,

name varchar(40),

price double,

category varchar(40),

pnum int,

imgurl varchar(100),

description varchar(255));

订单表

create table orders(

id varchar(100) primary key,

money double,

receiverinfo varchar(255),

paystate int,

ordertime timestamp,

user\_id int,

foreign key(user\_id) references user(id)

);

用户与订单之间存在一对多关系：在多方添加一方主键作为外键

订单与商品之间存在多对多关系：创建第三张关系表，引入两张表主键作为外键（联合主键）

订单项

create table orderitem(

order\_id varchar(100),

product\_id varchar(100),

buynum int,

primary key(order\_id,product\_id),

foreign key(order\_id) references orders(id),

foreign key(product\_id) references products(id)

);

设置数据库环境

数据库：create database estoresystem

### 3、工程环境搭建和网络部署

#### 1）导入jar包

导入mysql驱动 mysql driver / mysql-connector-java-5.0.8-bin.jar

导入 c3p0 c3p0/c3p0-0.9.1.2.jar 将c3p0-config.xml 复制src 下 将DataSourceUtils复制cn.itcast.estore.utils -----配置c3p0-config.xml 数据库连接参数

导入 dbutils apache commons\dbutils\commons-dbutils-1.4.jar

导入 beanutils commons-beanutils-1.8.3.jar commons-logging-1.1.1.jar

导入fileupload commons-fileupload-1.2.1.jar commons-io-1.4.jar

导入javamail mail.jar

导入jstl jstl.jar standard.jar

#### 2）建立package结构 按照JavaEE三层结构

cn.itcast.estore.web.servlet

cn.itcast.estore.web.filter

cn.itcast.estore.web.listener

cn.itcast.estore.service

cn.itcast.estore.dao

cn.itcast.estore.domain

cn.itcast.estore.utils

#### 3）Domain类编写

#### 4）工程发布

将bookEstore 项目配置虚拟主机，以顶级域名方式发布

1. 将工程根目录bookEstore目录配置虚拟主机目录 ---- 配置conf/server.xml

<Host name="www.bookEstore.com" appBase="myeclipse下的工程路径"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

</Host>

1. 将工程目录下WebRoot目录，配置缺省web应用

<Host name="www.estore.com" appBase=" myeclipse下的工程路径"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

<Context path="" docBase="WebRoot" />

</Host>

1. 将www.estore.com 虚拟主机配置缺省虚拟主机 ----- 能够用ip直接访问主机

<Engine name="Catalina" defaultHost="www.estore.com">

1. 修改本机域名解析文件 hosts ---- c:\windows\system32\drivers\etc

添加 127.0.0.1 www.bookEstore.com

## 3、功能实现

### 1 用户注册

1、 功能一： 用户注册 （技术点：验证码技术 、激活邮件技术）

1) 一次性验证码

原理：在生成验证码Servlet程序中，将生成验证码保存Session中，用户提交验证码 与 保存在Session验证码进行比较，如果相同，请求合法

2) 注册表单 JS校验

考虑JS校验是否可以抽取成框架

3) 处理form乱码问题 ---- 通用get post 乱码过滤器

4) 密码 MD5 加密

5) 激活邮件发送

6) 配置通用错误处理页面web.xml

<error-page>

<error-code>500</error-code>

<location>/500.jsp</location>

</error-page>

<error-page>

<error-code>404</error-code>

<location>/404.jsp</location>

</error-page>

500.jsp 显示错误的信息

404.jsp 自动刷新跳转回主页面

### 2 用户登录

原理：用户输入正确用户名和密码，登录成功，用户信息将会被保存Session对象中。--------记住用户名和密码、自动登录

1) 登陆表单中，添加记住用户名 和 自动登陆功能

2) 登陆过程中判断 账户是否激活

3) 在login.jsp 显示记住用户名

在username 的input项中添加 value="${cookie.username.value}"

在勾选自动登陆 checkbox 添加

4) 登陆后注销功能

注销Session

### 3 用户自动登录过滤器

自动登陆功能 对系统所有页面有效 （例如访问index.jsp list\_product.jsp info\_product.jsp 这些页面在访问时都将执行自动登陆 ）

\* 对于登陆相关页面不会执行自动登陆逻辑 （login.jsp 、LoginServlet 、InvalidateServlet ）

1) 判断该请求页面是否需要自动登陆

2) 是否已经登陆

3) 是否含有自动登陆cookie

4) 自动登陆

5) 在LoginServlet 添加对于没有勾选记住用户名和自动登陆 处理代码 ！！！！！！！

6) 在退出功能 InvalidateServlet 清除自动登陆信息

### 4 商品添加

文件上传注意三个注意事项：

1) input输入框 必须有name属性

2) 表单form 必须post提交方式

3) 设置form的enctype 为 multipart/form-data

表单提交时，检验分为两种：客户端校验、服务器端校验 -------只有服务器端校验才能确保数据准确

商品图片上传到服务器端后，保存在哪个目录 ？？？ 必须直接在WebRoot下及其除WEB-INF、META-INF 子目录外

一般情况下，一张表对应Domain类 --- DAO类 ---- Service类

### 5 商品列表查看

列表中显示原图，因为原图比较大，页面加载非常缓慢，页面布局不会很美观 -----------

缩略图

Java中通过图形界面技术，生成小图

在product类中添加getmgurl\_s方法，用来获取缩略图路径

### 6 商品详细信息查看

通过在列表中点击商品图片或者商品名称进入详情查看页面

### 7 添加商品到购物车

购物车对象，不保存在数据库中，使用session来保存用户购物车数据

保存购物车对象Map<Product,Integer> key 商品对象value商品购买数量

流程：添加商品到购物车流程，点击添加到购物车，将商品id传递servlet，从session中取出购物车对象，判断购物车对象，判断商品是否已经在购物车中，如果不在添加商品到购物车数量1，在购物车取出原有数量+1

1) 如果Map的key是一个自定义对象，重写 hashcode和equals

2) 如果商品不在购物车中，需要根据商品 id 查询商品所有信息，添加购物车

3) 添加商品到购物车

Session中信息在服务器正常关闭时，会被序列化到硬盘中 ---- Product实现 Serializable接口 完成序列化

### 8 显示与修改购物车数据

购物车信息保存在session中，不需要去查询数据库，将session中信息显示出来

1) 在购物车中显示总价

2) 购物车修改 ：

清空购物车 ： request.getSession().removeAttribute("cart");

删除购物车中单项商品 ： cart.remove(product) ; 注意：删除一项后，判断购物车是否为空，如果为空 移除购物车对象

\* 删除确认功能

写法一：直接在href中 触发js函数，询问用户是否确认，如果确认，location.href 发起删除请求 ----- <a href="javasript:confirmDel();" >...</a>

写法二：<a href="/delCart?id=xx" onclick="confirmDel(); "></a> 在链接中添加 onclick事件 ,询问用户是否确认，如果用户取消，通过JS阻止href事件提交

阻止href默认事件 ： e.preventDefault() ---- 必须支持事件 FF支持、IE不支持

IE阻止href默认事件

function confirmDel(e){

// 询问用户是否确认

var isConfirm = window.confirm("商品不要了吗？多好的商品啊！");

if(!isConfirm){

// 用户选择取消，阻止 a 标签 默认事件 href发生

if(e&&e.preventDefault){

// e对象存在，preventDefault方法存在 ---- 火狐浏览器

e.preventDefault();

}else{

// 不支持e对象，或者没有preventDefault方法 ---- IE

window.event.returnValue = false;

}

}

}

修改购物车中商品购买数量 ---- JavaScript 控制购物数量修改

### 9 订单生成

重点：

1) 向orders表插入订单信息后，同时需要向orderitem表插入 订单中每项数据

2) 多表插入 （数据完整性问题） --- 事务管理

3) 订单生成后 ，商品数量更新减少

### 10 订单列表查询与取消

管理查询订单 --- 所有人订单信息

普通用户 ---- 只能查询自己的订单

在订单生成后，查看订单列表 、也可以通过index.jsp 进入订单查看页面

重点：查询订单时，同时查询订单项目信息

1) 查询订单基本信息时，查询下单用户昵称和用户名

在Order类中 添加 用户的 username 和 nickname字段

select orders.\*,users.username,users.nickname from orders,users where orders.user\_id = users.id;

2) 查询订单项信息时，查询商品的名称和单价

在OrderItem类中 添加商品的name和price 字段

select orderitem.\*,products.name,products.price from orderitem,products where orderitem.produtct\_id = products.id and orderitem.order\_id = ?

订单取消 ： 删除订单表信息时，同时删除订单项信息 （订单项信息依赖 订单信息 ，必须先删除订单项）

注意事项

1) 可以取消未支付的订单，如果订单已经支付，将无法取消

2) 管理员不能取消任何用户未支付订单，普通用户只能取消 自己的未支付的订单

取消订单进行事务管理 ：删除订单项、删除订单、恢复商品数量

### 11 订单在线支付

1) 网站进行支付，根据易宝支付接口请求规范，生成易宝需要数据，将数据提交到易宝指定网址 https://www.yeepay.com/app-merchant-proxy/node

\* 在提交支付请求给易宝，需要将请求中数据，使用易宝提供密钥和算法进行加密，获得数字签名 hmac码 ，将hmac码发送给易宝

2) 易宝会连接银行，进行支付

3) 支付后，浏览器会以重定向方式访问 网站回调程序，易宝在收到银行转账后，会以Socket方式通知网站

\* 在浏览器重定向通知后，提供给用户支付成功显示页面 （不要去修改订单状态）

\* 在网站收到易宝点对点通知后，回复易宝success，修改订单状态

数字签名原理：将数据 使用密钥和算法加密后 获得数字签名hmac，将数据和数字签名hmac一起发送给易宝，易宝采用同样密钥和算法对数据进行加密获得数字签名hmac，比较请求中数字签名hmac与加密后获得数字签名hmac是否一致，如果一致 签名hmac有效 （数据没有被篡改 ）

流程入口： 未支付订单可以进行支付，管理员不允许支付

在线支付

1) pay.jsp 银行选择页面

2) OnlinePayServlet 准备易宝支付参数

\* 提供密钥和算法 商家编号 真实的 将PaymentUtil 复制 utils包 ， 将 merchantInfo.properties 复制 src

\* responseURL 支付成功后 回调地址 ，该地址会收到 浏览器重定向和服务器点对点两次通知 ，确保该地址 可以被易宝访问 (estore项目必须要配置缺省虚拟主机)

将OnlinePayServlet准备易宝参数，传递JSP 确认

3) confirm.jsp 通过form的隐藏域 提交参数给易宝网址 https://www.yeepay.com/app-merchant-proxy/node

4) CallbackServlet 回调程序

收到浏览器重定向和服务器点对点两个通知

浏览器重定向，显示支付成功，不要修改订单状态

服务器点对点，修改订单状态，回复success

### 12 销售榜单导出功能

获得商品销售情况，需要查询orderitem表 ------- 统计已支付订单项内容

1) 榜单中存在哪些信息？（已支付订单中商品）

商品信息 products表

销售数量 orderitem表

订单支付情况 orders表

select \* from products,orderitem,orders where products.id = orderitem.product\_id and orderitem.order\_id = orders.id ;

进行商品分组查询 group by

select products.\* , sum(orderitem.buynum) totalSaleNum from products,orderitem,orders where products.id = orderitem.product\_id and orderitem.order\_id = orders.id and orders.paystate = 1 group by products.id order by totalSaleNum desc;

2) 榜单文件是什么格式？

导出Excel 使用 POI类库

csv 格式文件 , 逗号分隔文件

1) 信息当中有,在两端加 双引号

2) 信息当中有" 在之前加双引号 转义

文件下载

设置Content-Type、Content-Disposition 头信息

文件流输出 （输出文件内容）

Excel 默认读取字符集gbk