**叮叮图书商城**

1. **功能描述**
   1. 普通用户购物功能
      1. 可以根据不同的类别查找响应的图书
      2. 单击某一本图书查看详细信息
      3. 在详情页面能够加入购物车
      4. 在浏览界面能够加入购物车。
      5. 能够查看购物车
      6. 能对购物车进行增删查改
      7. 能模拟支付（暂定）
      8. 能进行订单管理（暂定）
      9. 能对收货地址进行增删操作（暂定）
   2. 后台管理员能够实现的功能。

（1）超级管理员

能够实现对角色管理、用户管理、图书分类、图书管理、图书上架、发货管理

（2）图书管理员

能够实现屠苏分类、图书管理、图书上架的管理。

（3）发货人员

拥有发货的权限。

1. **实现的技术要点。**
2. 整体设计思想使用三层架构。即MVC层、服务层、数据访问层。
3. 购物车使用的数据结构：Map

使用Map而不是用List的原因是对购物车的增删查改操作将会简单很多。

具体类型是：Map<String,Car>，其中，Car是一种数据类型， Book对象将会作为Car的一个成员变量。Key值是Book类的id。

1. 实现登陆过滤：分别使用两个过滤器实现对普通用户订单管理页面的过滤和管理员管理页面的过滤。
2. 权限认证：使用认证过滤器实现对已经登陆的管理员对安全资源的访问。
3. 不使用数据库实现跨浏览器访问购物车与数据同步
4. 使用监听器实现对所有session的创建和销毁的监听。
5. 使用两个单例模式的对象分别管理两个map单例对象

一个是用户id与sessionid的映射

另外一个是sessionid和HttpSession对象的映射。

1. 实现跨浏览器的核心思想：同一个购物车，不同的引用。

同一个购物车：不管什么情况，一定要让所有用户访问的都是同一个购物车，即在内存中是同一个地址的对象。

不同的引用：一个购物车可以被多个session会话同时引用。

1. 监听器将新的session保存到map之后，立即在该会话中创建一个空的购物车。
2. 用户单击退出立刻使得浏览器cookie失效，服务器重新分配一个sessionid。
3. 分页管理。
4. 分页的关键是需要定义每页显示多少条数据，需要给用户显示多少个按钮。
5. 需要根据（1）中定义的几个信息来计算一共产生多少页
6. 当页码位置能够处于中间的时候就让该页码在中间显示，能够达到最好的用户体验。
7. 事务管理
8. 使用注解+动态代理在服务层实现事务管理。
9. 注解：使用系统注解限制自定义注解TransactionFlag必须加在方法上，而且在运行的时候不删除。
10. 动态代理：能够代理一切拥有接口的类。所以如果想要开启事务，该service必须拥有至少一个接口。注解加在接口上。
11. 动态代理返回的实例是接口在内存中生成的对象，所以返回值一定是接口，而不是接口的相应子类，否则报异常。
12. 该功能必须借助ThreadLocal模式来实现
13. 文件上传。
14. 文件上传使用的第三方jar包：apache公司的fileupload。
15. 实现目录打散，使用两级目录
16. 实现文件重命名：解决中文路径不识别问题，解决文件名相同问题。
17. 如何更加灵活的响应浏览器的请求信息。
18. 定义BaseServlet来作为所有Servlet的父类，该Servlet不在web.xml文件中注册。
19. 所有子类都没有service方法或者doGet、doPost方法，这些方法在BaseServlet中重写，这些Servlet都必须在web.xml文件中注册。
20. BaseServlet通过反射获取子类Servlet中的方法，该方法名为请求的cmd，并执行。
21. 重定向和转发的统一管理
22. 使用配置文件views.properties统一管理需要转发或者重定向的uri，如果转发，直接在Servlet方法中返回该uri字符串，如果是重定向则返回302：uri。
23. 配置文件的名字和位置应当更加灵活，所以在web.xml中对该配置进行配置。为此，需要借助ServletContextListener监听器，这样在ServletContext初始化的时候就读取web.xml文件，生成的Properties对象由PageUtils统一管理，包括根据key值获取value。
24. BaseServlet通过调用pageUtils中的相应方法获取需要转发或者重定向的url。
25. **数据库设计。**
26. 后台权限管理。
27. 通过使用分配角色的思想对权限进行划分，使得权限分配更加灵活。

目的：确保不同级别的管理员登陆之后能够看到不同的管理菜单。

1. 实现该目的必须借助中间表。
2. 中间表不需要形成JavaBean。
3. 如何高效的获取请求的信息并保存成JavaBean
4. 表单类型不是使用multipar/form-data：使用BeanUtils第三方jar包的populate方法直接封装。
5. 表单类型是multipart/form-data：先将数据封装到map，再使用BeanUtils的populate方法进行封装。
6. 数据库工具：使用c3p0数据库连接池，效率更高，管理更加灵活。
7. 怎样更加高效的获取查询之后的结果集List对象

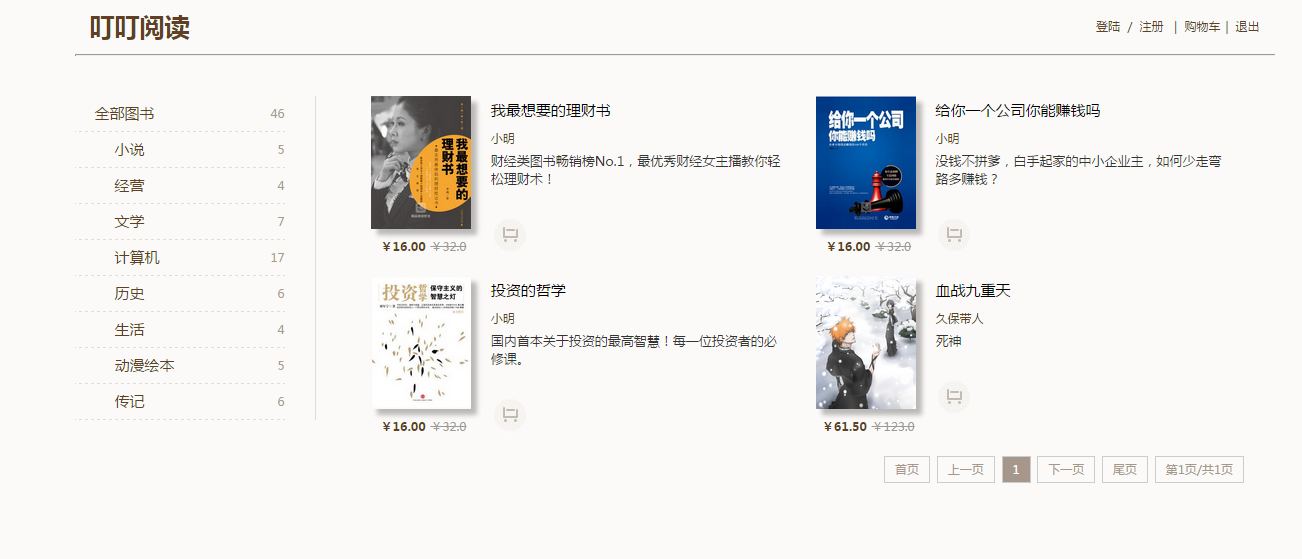
使用第三方jar包：dbutils.jar

1. **项目演示**
2. 首页：

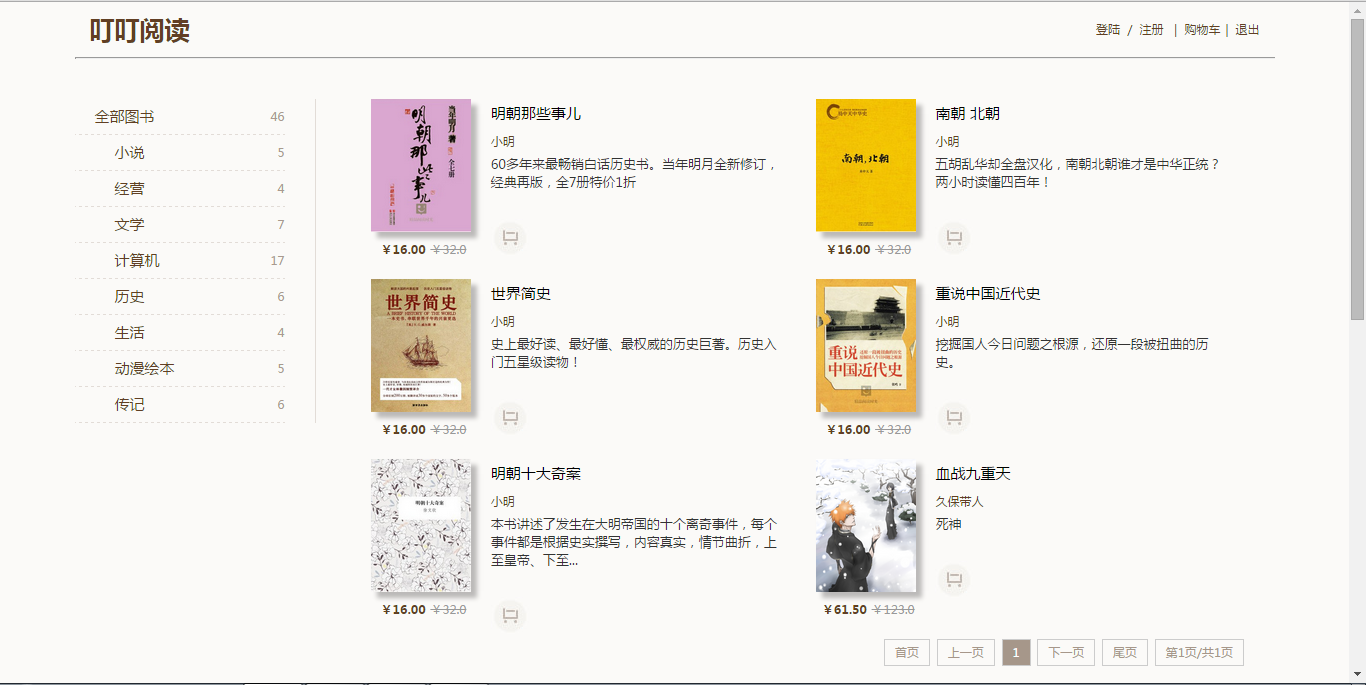


1. 查看不同分类的图书。

查看经营类的：



查看历史类：



1. 加入购物车
2. 浏览界面加入购物车



1. 详情界面加入购物车



加入购物车之后：



1. 查看购物车。



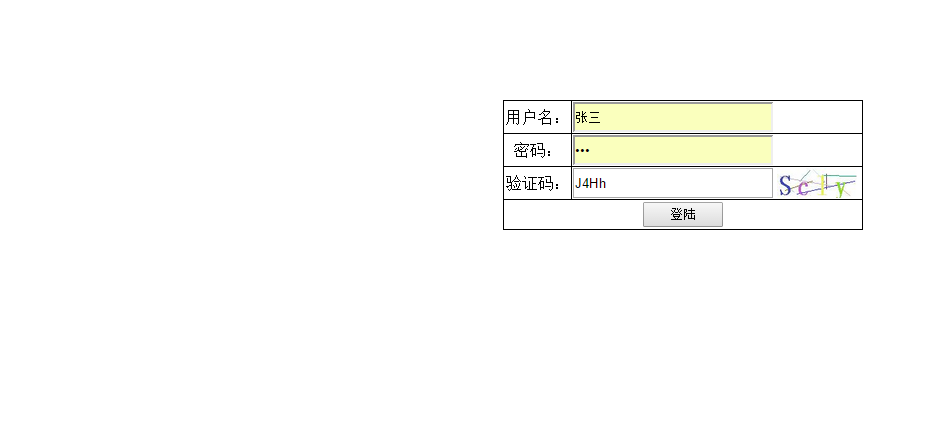
1. 结算
2. 未登录。



1. 未注册



1. **后台管理**
2. 账户登录



1. 登陆成功



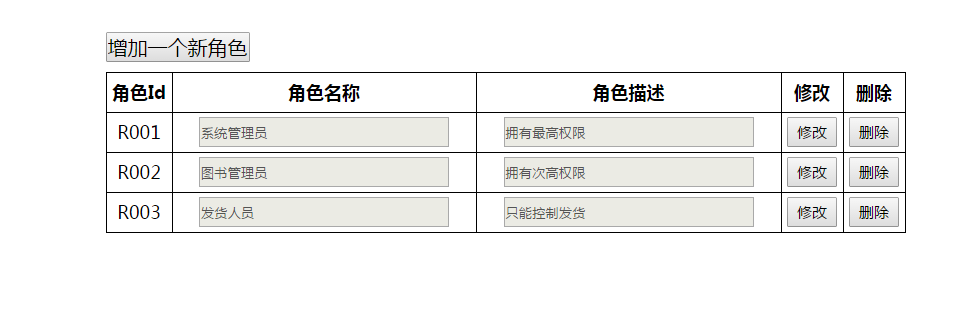
1. 管理角色



添加



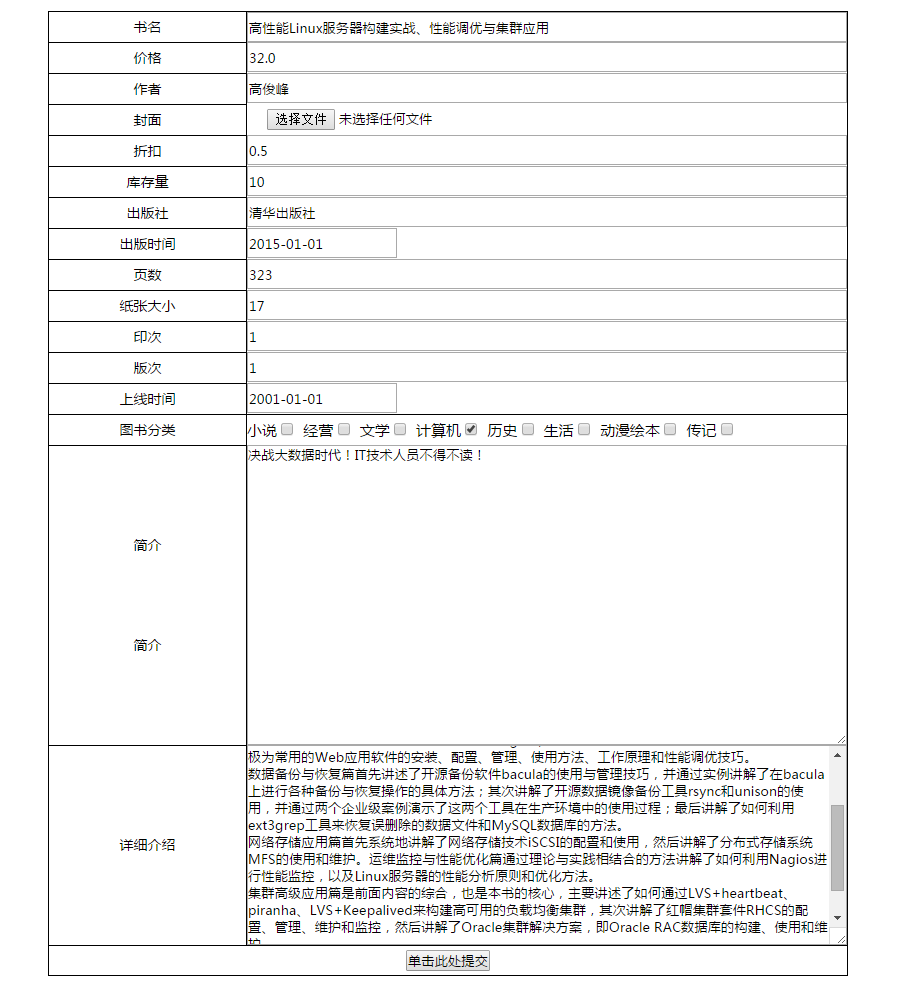
删除角色



1. 图书管理



1. 修改

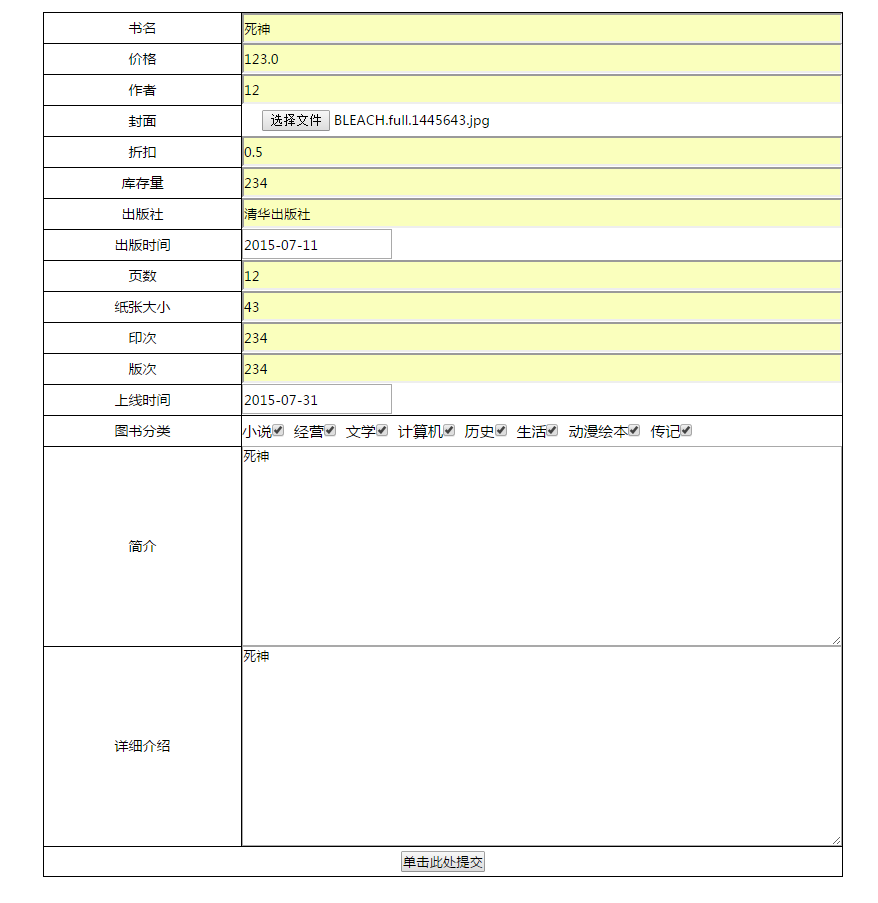


图书封面如果没有手动选择则使用之前的老封面，如果上传了新封面则使用新上传的封面。

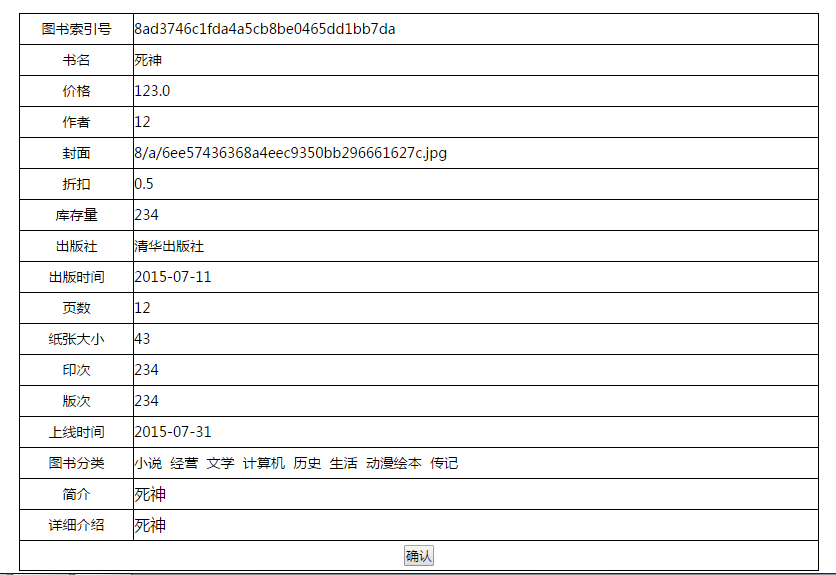
修改成功：

1. 添加





提交之后：



前台查看：



具体信息查看



后台查看：



1. 删除

删除之后，前台、后台都没有了。