**湖 南 科 技 大 学**

**毕 业 设 计（ 论 文 ）**

|  |  |
| --- | --- |
| **题目** | **基于JavaWeb的图书购物系统设计** |
| **作者** | **管严锋** |
| **学院** | **教育学院** |
| **专业** | **教育技术学** |
| **学号** | **1314020203** |
| **指导教师** | **林盾** |

二〇一七 年 六 月 二 日

**摘 要**

本系统采用了当前最为流行的编程语言Java开发完成，以SpringMVC+Spring+Mybatis主流框架为核心技术。前端使用JSP开发，数据存储使用Oracle数据库。以MVC（模型+视图+控制器）软件设计思想为指导，分层开发将页面、后台、数据库融为一体，设计出稳健、高效的购物系统。系统实现了用户、商品、购物车、订单等模块的基本功能，用户能在系统上安全快捷的购物。本文从先到后介绍了开发技术、可行性分析和需求分析、系统设计、代码实现、测试、维护等，较为完整的展示软件开发的全过程。

关键词：Java；主流框架；购物系统；软件开发

**ABSTRACT**

The system uses the most popular programming language Java development is completed, with SpringMVC + Spring + Mybatis mainstream framework as the core technology. Front use JSP development, data storage using Oracle database. With MVC (model + view + controller) software design ideas as a guide, layered development integrated the page, background, and database. System to achieve the user, goods, shopping carts, orders and other basic functions of the module, the user can safely and quickly in the system of shopping. This paper introduces the whole process of developing software, feasibility analysis and demand analysis, system design, code realization, testing and maintenance, and more complete display software development.

**Keywords:** Java; mainstream framework; system of shopping; software developme

**目录**

[**第一章 绪论 - 2 -**](#_Toc483592328)

[1.1系统开发背景 - 2 -](#_Toc483592329)

[1.2选题目的及意义 - 2 -](#_Toc483592330)

[1.3系统简介 - 2 -](#_Toc483592331)

[**第二章 开发环境及技术介绍 - 4 -**](#_Toc483592332)

[2.1开发环境 - 4 -](#_Toc483592333)

[2.2相关技术介绍 - 4 -](#_Toc483592334)

[**第三章 可行性分析和需求分析 - 5 -**](#_Toc483592335)

[3.1可行性分析 - 5 -](#_Toc483592336)

[3.2模块功能分析 - 5 -](#_Toc483592337)

[3.3系统功能流程图 - 5 -](#_Toc483592338)

[3.4 系统依赖分析 - 8 -](#_Toc483592339)

[**第四章 系统设计 - 9 -**](#_Toc483592340)

[4.1系统整体框架 - 9 -](#_Toc483592341)

[4.2功能详细设计 - 10 -](#_Toc483592342)

[4.3角色设计 - 11 -](#_Toc483592343)

[4.4页面字段设计 - 12 -](#_Toc483592344)

[4.5数据库设计 - 14 -](#_Toc483592345)

[**第五章 代码实现及效果图 - 17 -**](#_Toc483592346)

[5.1核心代码展示 - 17 -](#_Toc483592347)

[5.2效果图 - 22 -](#_Toc483592348)

[**第六章 系统测试 - 25 -**](#_Toc483592349)

[6.1单元测试 - 25 -](#_Toc483592350)

[6.2集成测试 - 25 -](#_Toc483592351)

[**第七章 反馈与维护 - 26 -**](#_Toc483592352)

[**第八章 总结 - 27 -**](#_Toc483592353)

[**参考文献 - 28 -**](#_Toc483592354)

[**致 谢 - 29 -**](#_Toc483592355)

**第一章 绪论**

**1.1系统开发背景**

近十几年来IT行业日新月异，推动着我们向着便捷、智能的时代迈进。就人们的购物方式就发生的翻天覆地的变化，由传统的线下直销变为网络直销、线下线上相结合等多种模式。

以IT行业的快速发展为动力，推动了国内电子商务行业的快速崛起，天猫、京东、当当等各种电商平台百花齐放。在此大环境下，于是开发一种以图书为核心商品的在线图书购物系统。

**1.2选题目的及意义**

互联网行业的快速发展，对纸质书本带来了前所为有的冲击，对纸质图书的需求量也是大大降低。提升大家对书本购买的体验对未来纸质书本的发展有着一定的意义。

将心理学、教育学、计算机等知识相结合运用到图书购物系统中更好的把握用户心理，能准确快捷、准确为用户服务。这有助于图书网站的推广和发展。

另一方面，纸质书本有着电子书无法相比的触觉感受和便捷性（如翻阅查找、随手笔记等）。纸质图书很难被电子书完全取代，纸质图书还有很大的市场，因此选做网上图书系统的开发。

**1.3系统简介**

该系统是以B/S模式开发的在线购物系统，以B2C(Business to Customer企业对消费者)新型的电子商务模式运营。用户可以在该平台上登录、注册、浏览商品、搜索商品、查看个人信息、修改个人信息、加入购物车、购物车内商品信息修改、购物车结算、支付、查看订单、常见问题、问题反馈等。

系统整体的功能结构图如下：

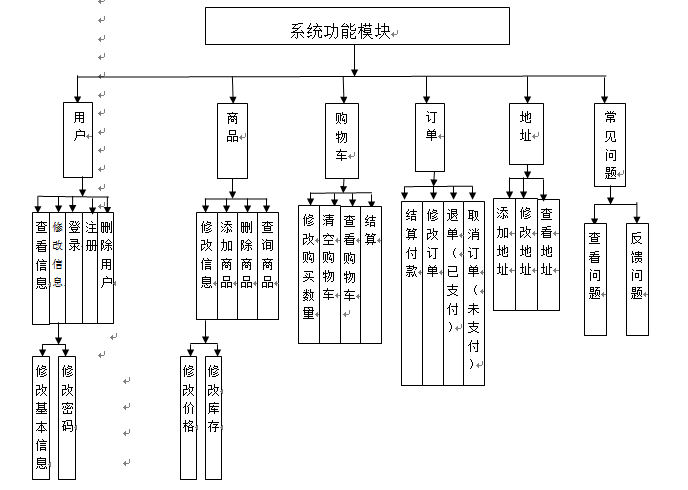


图1.1 功能结构图

**第二章 开发环境及技术介绍**

**2.1开发环境**

JDK1.6以上、Myeclipse、Tomcat6.0以上、Oracle数据库、主流流量器等。JDK是Java开发的核心条件，包含了Java代码的运行的必要条件和基本的类库。它是Java开发的最基本也是最关键的保障。Myeclipse是一个现今使用量最大的的专业集成开发环境，很好的支持各类插件。它能够完成代码的编写、编译、调试、测试等，是当前最为流程的Java开发工具。Tomcat是一款免费高性能的web服务器，是Java企业级开发不可少的工具。

**2.2相关技术介绍**

JavaWeb包含一系列的前后端技术，是以Java为核心技术解决web互联网领域开发问题。JavaWeb是Java企业级开发（Javaee）的部分，本系统就是基于JavaWeb开发完成。

本系统前台页面采用JSP+CSS+JavaScript。JSP是JavaWeb中特有的实现动态页面的技术。CSS全称是层叠样式表，能够快捷精准的调整页面的样式。JavaScript是一种运行在客服端的脚本语言，在web应用开发中使用的非常广泛。JavaScript脚本能为页面添加各种动态效果，提升用户浏览页面的体验。后台开发是以MVC模式的软件架构思想为基础，使用JavaWeb的主流开发框架完成。使用的主流框架是SpringMVC、Spring、Mybatis，它们分别使用在MVC架构的不同层中。

SpringMVC是现在web开发的重要技术，是SpringFrameWork更新产品。在本系统开发时主要pringMVC应用在controller层。Spring是Rod Johnson创建的一种轻量级的Java开源框架。高效、模块化等优点使他快速流行起来，它为JavaWeb开发提供一站式解决方案。在该系统中将他用在service层。Mybatis是应用在持久层的框架，将Java 的POJOs和数据库中的表形成映射关系。用XML配置相关信息管理起来不但方便还简单。

本系统信息存储使用的是由甲骨文公司开发的Oracle数据库。该数据库是当前世界上最流行的关系数据库之一。Oracle数据库有着优秀的安全性、适应性、效率高、大容量的数据库管理系统。

**第三章 可行性分析和需求分析**

**3.1可行性分析**

技术上，Java已是当前最流行是编程语言,发展成体系庞大又成熟健壮的面向对象类编程语言。GitHub上有着众多优秀的开源项目供我们学习参考，以后技术上的开发基本不会遇到太大难题。完成该系统的开发不需要较高的软硬件水平，一般的计算机就能保证整个系统的运行。在个人知识技能水平方面，本人对开发需要的Java基础、JavaWeb、数据库等有着良好的认知，能够保证系统顺利开发完成。

经济上，该系统是针对毕业设计，并不是真正的商业项目。系统开发时不需要投入较大的人员等方面的经费，只需要一台正常流畅运行的电脑和一些开发软件。

**3.2模块功能分析**

用户：用户登录进入系统后可以浏览商品、关键字查找商品、提交意见建议、加入购物车、结算、下单、支付、查看订单等。用户在操作这些功能时若是出现异常情况，统一跳转到异常页面并在页面显示相关的错误信息，给用户有用的提示帮助用户正确操作。

商品：在本系统中就是各类图书。图书模块的功能不是本身具有的，都是在一定情景下用户或管理员触发的。图书模块的功能有添加商品、删除商品以及修改商品信息。

购物车：用户进入购物车界面显示购物车信息，用户可以修改购物车内的商品信息（规格、购买数量等）、清空购物车、结算等。在结算时商品库存不足、账户余额不足时在错误页面提示。

订单：订单的所有功能游客是无法来操作的需要先登录。在订单页面用户可以查看订单、支付未完成的订单、对已完成订单申请退单等。删除订单、修改订单信息等功能需要进行权限的检验，需要由管理员来完成。

**3.3系统功能流程图**

用户模块登录、注册、修改个人信息等功能流程图

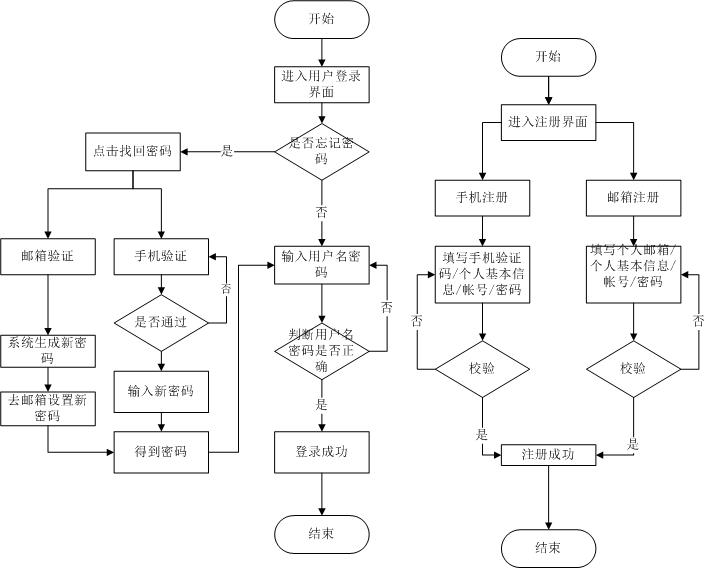


图3.1 用户登录注册流程图

 图3.2 查看、修改信息流程图

商品模块修改商品(库存、价格)、添加商品、删除商品、查看商品流程图

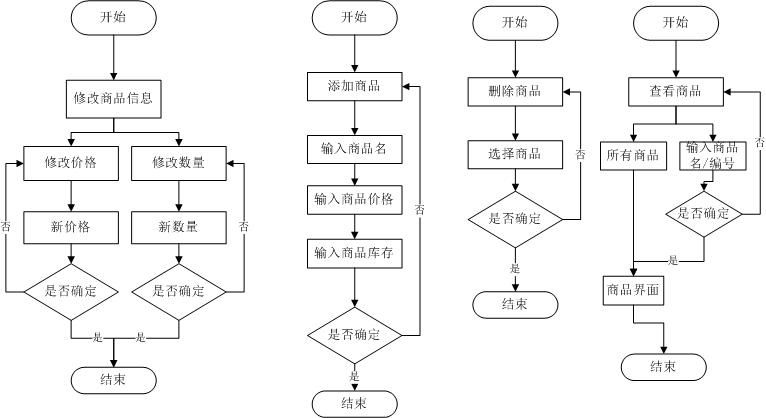


图3.3 商品流程图

购物车模块修改、清空、查看、结算流程图

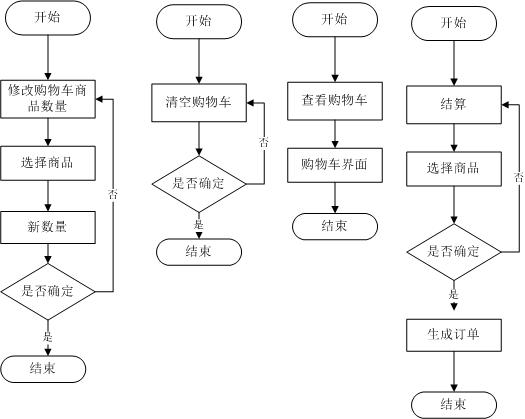


图3.4 购物车流程图

订单模块查询订单、修改订单、支付、退单流程图

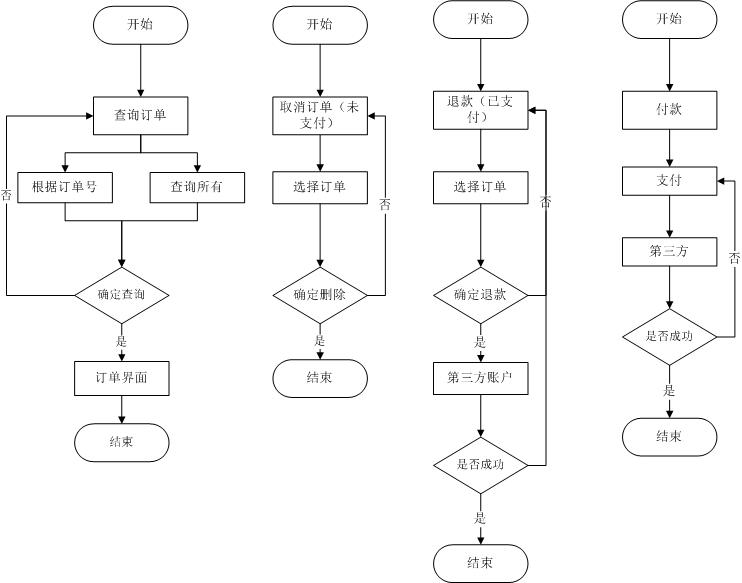


图3.5 订单流程图

**3.4 系统依赖分析**

由于该系统的开发是使用了SpringMVC、Spring、Mybatis三个主流的开源框架，并不是原生态的开发。完成系统的开发除了基本的开发环境（jdk、Tomcat、数据库等）以外，还需要导入这些框架相关的jar包。

**第四章 系统设计**

**4.1系统整体框架**

由于该系统的开发是使用了SpringMVC、Spring、Mybatis三个主流的开源框架，并不是原生态的开发。完成系统的开发除了基本的开发环境（jdk、Tomcat、数据库等）以外，还需要导入这些框架相关的jar包。系统采用MVC的设计思想，将代码分为Controller、Service、DAO三个层次完成，向高内聚低耦合的标准去靠拢。

先简单介绍一下MVC框架。MVC是Model View Controller的简称，不是具体实在的技术，而是一种长久经验积累抽象出来的软件设计的规范。模型（Model）就是业务模型，在Java中可以理解为实体类。这个系统中用户（User.java）、图书（Book.java）、购物车（ShoppingCart.java）、订单（Order.java）等就属于模型。视图（View）即界面显示，将数据显示给用户。控制器（Controller）是页面请求和后台逻辑处理的部分。它会根据前台的请求还调度对应的逻辑处理部分。MVC思想虽在结构上相对复杂，但是将不同部分的代码按照模型层、视图层、控制层分离出来，使得诸多优点提现出来。

耦合性低：不同代码分离之后，每一层需要关注其他层的具体实现，降低了每层之间的关联程度。其中任何一层的变更都不会影响到其他层，这也有助于团队分工合作。

重用性高：由于视图代码和控制层代码分层，不同页面的请求可以在同一个控制器中处理。例如PC端和移动端登录，虽然页面显示的不一样，但是服务端处理的方式和过程是一样的。MVC框架可以让代码最大化的重复使用。

维护性好：随着需求的变化增加、技术的提升，系统会需要不断地更新维护。三层分离使得系统更易于更新维护。比如系统要添加积分的功能，在视图层中加入积分的页面展示，在控制层加入对应请求处理，在模型层加入对积分的逻辑处理和数据。

在该系统中视图层的代码为JSP、CSS和JavaScript，负责展示登录注册、商品信息、用户信息、购物车、订单等。控制层中主要是用户、商品、购物车、订单等各个模块功能的控制调度的Java代码。根据前台页面不用请求路径，来调用对应功能的逻辑实现。模型层分为了Service和DAO两块。Service中负责每个功能的业务逻辑实现，DAO中则和数据库中的表实现对应，主要是对数据库表的insert、delete、update、select的操作。

**4.2功能详细设计**

系统的核心功能在设计时，分模块介绍各个功能数据如何在各层之间传递的，并画出个功能的时序图。

用户登录：在登录页面输入用户名和密码时，在页面JavaScript代码中完成前台的基本校验，该检验主要检验输入的用户名和密码格式。请求到达Controller层在由控制层去调用Service中的登录逻辑处理的方法。Service中的逻辑为判断用户是否存在和密码是否正确，错误情况时抛出异常并给出错误信息。DAO层中则负责向数据库用户表查询该用户的基本信息，Mybatis会自动将信息封装成对象并返回。

用户注册：在注册页面输入用户名、生日、密码、性别、邮箱等信息，并检验是否合法。Controller中将页面提交来的用户信息封装成用户对象，再调用注册的Service。注册的业务逻辑是根据传递来的用户对象判断用户是否已存在，不存在则注册。DAO中是向用户表中插入该用户的信息记录。

查看用户信息：在页面点击页面个人中心时，向服务器端发送请求。控制层中解析请求的路径再调用执行Service中对应的方法，在逻辑层中判断用户是否存在并调用DAO中查询数据库用户表的方法。

修改用户信息：用户根据自己需要修改密码、年龄等信息。在个人信息页面可以直接修改信息并提交。同样请求到达Controller根据路径解析去调用修改用户信息的业务逻辑代码。在Service中先判断用户是否存在，再去调用DAO层对数据库更新的方法。

添加商品：该功能是设置权限要求的。在系统的后台管理界面，管理员输入商品信息提交到服务器。Controller层中去调用Service中实现添加商品的逻辑方法，再由DAO层向数据库中商品表插入数据。

修改商品信息：主要是管理员来修改商品的价格、描述信息等。控制层根据路径调用逻辑层的方法。先查询商品是否存在再去调用DAO的方法将数据更新到数据库。

删除商品：商品因品质等原因需要下架淘汰时，需要将该商品从系统中删除。在页面上选择要删除的商品提交后将请求发送到服务端，在控制层去调用Service中的删除方法。方法中先检验商品是否存在，存在再去调用DAO层对数据库删除记录的方法。

查询商品：页面多处用到查询商品的功能，管理员和用户可以操作该功能，也可以根据关键字查询。在控制层解析路径并调用服务层的对应查询方法（根据商品名查、根据商品ID查、根据关键字查），在由数据层执行对商品表的查询操作。

添加购物车：新建购物车的功能不在界面提供，用户在注册时和用户一起创建。在用户创建的Service层中的方法中，同时有实现创建购物车的代码和创建用户逻辑一样再去调用DAO层方法，在数据库中插入一条该用户相关的购物车记录。

加入到购物车：在页面点击加入购物车时，请求中带着商品的ID。在控制层去调用逻辑层代码。逻辑层是将每个添加的商品创建一个购物车子项，购物车有个子项的集合用来存放子项。创建子项去调用DAO层子项的添加方法，向数据库购物车子项表中插入一条数据。子项添加成功后，再去调用DAO层购物车更新的方法，将购物车总金额更新到数据库。

购物车中删除已加入商品：在购物车显示界面中所有添加的商品都显示出来，并在每个子项后提供一个删除按钮。用户点击时将该子项的ID放入请求中一起到达服务端。控制层调用对应的逻辑层代码，先查到改用的购物车，在从购物车中删除该子项（调用子项的DAO层删除方法）并修改购物车的总金额，再调用DAO层的方法将数据更新到数据库。

清空购物车：购物车界面用户点击清空，控制层调用Service中清空的方法。该方法中将购物车的总金额重置为0，并调用DAO层将数据更新到数据库。在将购物车的子项都删除，调用DAO中方法删除数据库子项表对应的数据。

购物车结算：购物车界面提供结算按钮。用户点击时控制层调用对应逻辑实现代码。根据购物车的信息创建订单，订单状态为未支付，调用DAO层方法向数据库插入该订单的一条记录。每个购物车的子项创建对应的订单子项，并将数据插入到数据库中。在调用购物车清空的方法将购物车清空。

支付：点击支付时，后台先校验该订单是否是为未支付的。查询到用户将账户的余额减去订单的总金额，想管理员账户加入对应的金额，余额不足则支付失败抛出异常。再根据购买数量减去对应商品的库存，库存不足也会支付失败。该业务逻辑较为复杂，又是系统的核心功能且涉及到金额，需要在这个逻辑实现的方法添加事务控制来保证数据的统一。

**4.3角色设计**

游客：游客可以访问主页，浏览各种商品等详细信息(查看商品的图片、库存、价格)但是不可以加入购物车、不能生成订单、地址无法添加、不能加入收藏。如果要执行上述操作系统会自动跳到登陆界面登陆或注册后才可以执行

用户：用户拥有自己的唯一账号用户登录后可以在该系统上进行一系列的操作。用户可以完成游客的所有功能，对购物车、订单等操作会在数据库中保存相应的数据集，下次登录后可以继续操作而游客不会在该系统中留下任何记录。

管理员：主要是对系统信息的管理，通过后台管理系统登录。管理员可以商品信息的管理（修改价格、库存等），也可以对用户管理（删除用户等）。管理员要有对页面信息的更新、收集客户反馈的问题。

**4.4页面字段设计**

页面除了布局样式之外，页面的字段是交互式页面的关键。它们将后台数据库的数据显示到页面，同时将页面需要提交的数据发送到服务端。下面就系统的主要界面字段进行设计介绍：

表4.1 注册页面

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 页面字段 | 类型 | 备注 |
| 1 | 用户名 | Input：text |  |
| 2 | 用户密码 | Input: password |  |
| 3 | 确认密码 | Input：password |  |
| 4 | 验证码 | image |  |
| 6 | 确定 | Input：button |  |

表4.2 登录页面

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 页面字段 | 类型 | 备注 |
| 1 | 用户名 | Input：text |  |
| 2 | 用户密码 | Input：password |  |
| 3 | 自动登录 | Input：check |  |
| 4 | 登录 | Input：button |  |
| 5 | 用户注册 | a |  |

表4.3 用户信息

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 页面字段 | 类型 | 备注 |
| 1 | 用户名 | Input：text |  |
| 2 | 用户密码 | Input：password |  |
| 3 | 性别 | Text |  |
| 4 | 修改 | Input：button |  |
| 5 | 余额 | Input：text |  |

表4.4 首页

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 页面字段名 | 类型 | 备注 |
| 1 | 注册 | a |  |
| 2 | 登录 | a |  |
| 3 | 我的订单 | a |  |
| 4 | 购物车 | a |  |
| 5 | 我的收藏 | a |  |

表4.5 商品列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 备注 |
| 1 | 名字 | Text |  |
| 2 | 金额 | Text |  |
| 3 | 图片 | Img |  |
| 4 | 描述 | Text |  |
| 5 | 加入购物车 | Submit |  |
| 6 | 详情 | Submit |  |

表4.6购物车页面

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 页面字段 | 类型 | 备注 |
| 1 | 全选 | Input：check |  |
| 2 | 删除 | a |  |
| 3 | 修改 | Input：button |  |
| 4 | 清空购物车 | Input：button |  |
| 5 | 查询购物车 | Input：button |  |
| 6 | 商品名 | Text |  |
| 7 | 单价 | Text |  |
| 8 | 合计 | Text |  |
| 9 | 结算 | Input：Submit |  |

表4.7 订单页面

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 备注 |
| 1 | 收货地址 | Radio |  |
| 2 | 配送时间 | Radio |  |
| 3 | 发票 | Radio |  |
| 4 | 商品名 | Text |  |
| 5 | 单价 | Text |  |
| 6 | 数量 | Text |  |
| 7 | 是否有货 | Text |  |
| 8 | 合计 | Text |  |
| 9 | 确认支付 | Submit |  |

**4.5数据库设计**

数据库表E-R图设计



表4.1 E-R图

用户表（users）：存储用户和管理员信息。

表4.8 users表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 字段约束 | 备注 |
| 用户ID | id | Number(10) | Primary key |  |
| 密码 | password | Varchar2(20) |  |  |
| 姓名 | name | Varchar2(8) |  |  |
| 性别 | gender | Varchar2(4) |  |  |
| 年龄 | age | Number(10) |  |  |
| 出生日期 | birthday | Varchar2(10) |  |  |
| 账户 | Account | Number（8,2） |  |  |
| 邮箱 | Email | Number(15) |  |  |

商品表（book）设计，存储商品信息

表4.9 book表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 商品ID | Id | Number(15) | Primary key |  |
| 商品名 | Name | Varchar2(10) |  |  |
| 商品单价 | Price | Number(10,2) |  |  |
| 商品库存 | Store | Number(10) |  |  |
| 商品描述 | describe | Varchar2(200) |  |  |

购物车基本信息表（ShoppingCart）设计，存储购物车信息

表4.10shoppingCart表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 用户ID | userId | Number(10) | Foreign key |  |
| 购物车ID | Id | Number(10) | Primary key |  |
| 总金额 | Total | Number(10,2) |  |  |

购物车子项基本信息表（cartItem）：保存购物车子项基本信息

表4.11 cartItem表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 购物车ID | cartid | Number(10) | Foreign key |  |
| 子项ID | Id | Number(10) | Primary key |  |
| 商品ID | bookid | Number(15) |  |  |
| 商品名字 | Name | Varchar2(10) |  |  |
| 商品单价 | price | Number(10,2) |  |  |
| 购买数量 | number | Number(10) |  |  |

订单基本信息表（orders）设计，存储订单信息

表4.12 orders表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 用户ID | userid | Number(10) | Foreign key |  |
| 订单ID | Id | Number(10) | Primary key |  |
| 总金额 | total | Number(10,2) |  |  |
| 订单日期 | Date | Varchar2(20) |  |  |
| 状态 | status | Number(2) |  |  |

订单子项基本信息表（orderItem）设计，用于存储订单子项信息

表4.13 orderItem表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 订单ID | orderid | Number(10) | Foreign key |  |
| 订单子项ID | Id | Number(10) | Primary key |  |
| 商品ID | bookid | Number(15) |  |  |
| 商品名字 | Name | Varchar2(10) |  |  |
| 商品单价 | Price | Number(10,2) |  |  |
| 购买数量 | Num | Number(10) |  |  |

地址表信息表（address）设计，用于存储地址信息

表4.14 address表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 字段类型（精度） | 约束条件 | 备注 |
| 用户ID | Userid | Number(10) | Foreign key |  |
| 地址ID | Id | Number(10) | Primary key |  |
| 收件人 | Name | Varchar2(10) |  |  |
| 地址 | Site | Varchar2(100) |  |  |
| 联系方式 | phoner | Number(11) |  |  |

**第五章 代码实现及效果图**

**5.1核心代码展示**

从页面view到控制层controller到数据模型层model（Service+DAO），每层代码都是按照用户、购物车、订单这四个模块编写的。每层就展示一个模块的功能代码实现。工程目录结构如下：

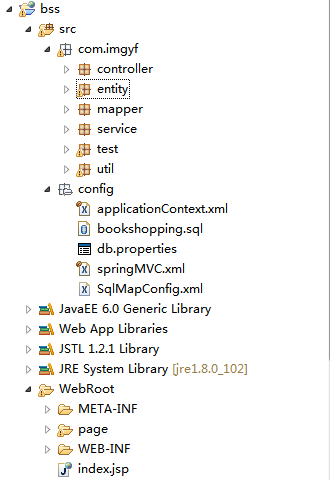


图5.1 工程结构图

DAO层中使用Mybatis框架通过mapper代理的方式和数据库中的表相映射。每一mapper接口类对应一个mapper.xml文件，包含address、book、cartItem、orderItem、orders、shoppingCart、users

以用户为例：UsersMapper.java

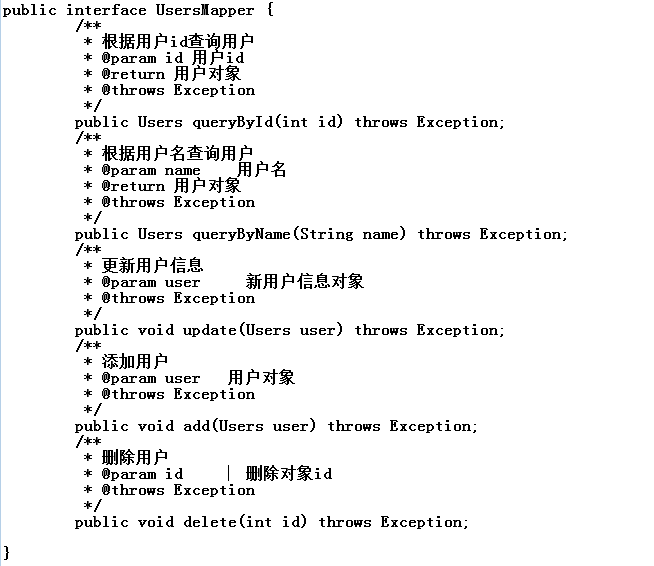


图5.2 UserMapper代码

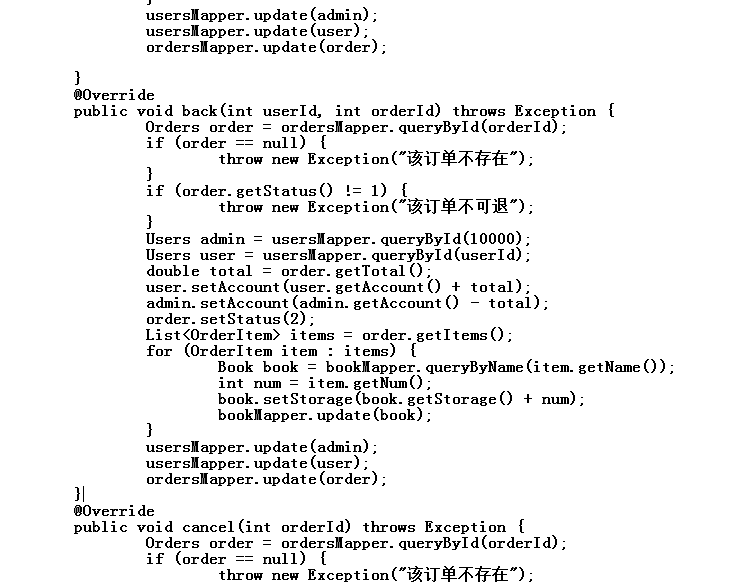
UsersMapper.xml



图5.3 UsersMapper配置文件代码

Service层是采用spring框架用到该框架的注解、面向切面等技术。该层主要功能是实现各个功能的逻辑代码，包含订单、用户、购物车等模块。该层以订单为例展示代码OrdersServiceImpl.java





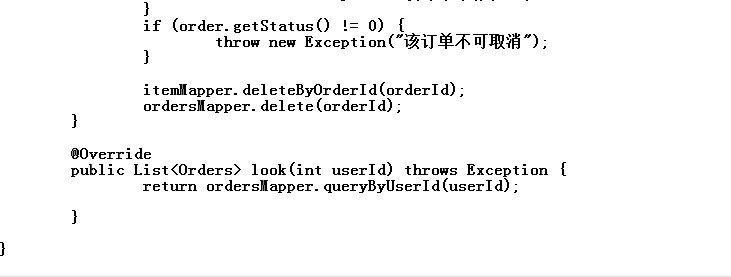


图5.4 order逻辑层代码

Controller层使用SpringMVC实现，该层主要根据前台请求的路径去调用对应功能的业务逻辑。该层按模块编写代码一个模块的所有功能在一个类中，包含用户、订单、购物车等。以购物车为例ShoppingCartController.java

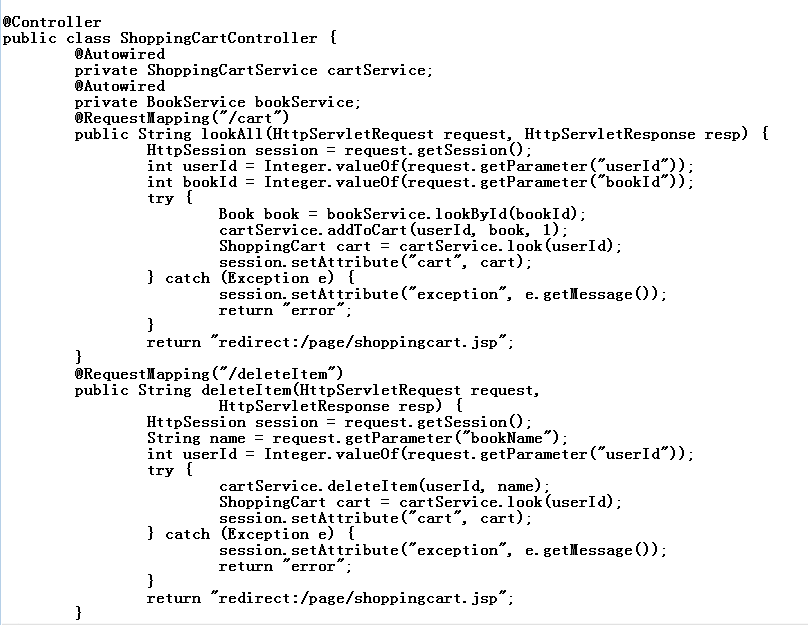




图5.5 ShoppingCart控制层代码

页面是以jsp技术实现的，系统实现一共编写了15个界面，包登录注册、首页、商品、购物车、订单、简介、异常界面等。以首页的核心代码为例展示index.jsp



图5.6 首页核心部分代码

**5.2效果图**



图5.7 注册界面



图5.8登录界面



图5.9 首页



图5.10商品界面



图5.11 购物车界面



图5.12 订单界面

**第六章 系统测试**

**6.1单元测试**

单元测试就是以不可再分割模块进行测试，对于Java语言写的系统而言就是以类为单位进行测试。为了不影响正常代码测试时一般新建一个包来编写测试代码。

先配置连接数据库，在数据库中新建系统需要的表。按层来编写代码，每完成一层就开始以类为单位测试。每个方法逐个测试，在测试类中调用并按预期编写合适的实参，数据库中查看是否和预期结果相符。每个方法正常测试没问题时还有进行异常测试，将数据给为非正常看能否到达预期的异常情况。

**6.2集成测试**

集成测试是多功能连起来测试，排除多层连接之后的异常情况。单元测试都通以后就可进行集成测试了。集成测试不是像单元测试那样随意，是要按照一定步骤进行。在集成测试时需要编写好测试用例，在具体实施测试时严格按照测试用例进行测试。

在该系统测试时先编写好测试文档，按模块为每个功能编写好测试用例。在界面按照测试用例步骤操作并查询数据库，若是出现异常情况要记录好错误信息并分析错误的原因。在修改代码是除了测试修改的功能，还要测试相关的功能以确保不会引入其他异常。

**第七章 反馈与维护**

系统上提供用户反馈的界面，用于用户提交意见建议。管理员根据这些意见来改进系统功能，进一步提升用户体验。听取用户反馈也是保证系统长久发展的重要保障。反馈不仅仅包含用户的反馈，还包括系统的普及度、影响力、专业人士测评等等方便。每一个成功的软件都是收集方方面面的信息不断的改进和提升。

维护是系统长久运行的保障，在将开发完成的系统交付使用之后，就开始进入了维护时期。系统的维护成本要占到软件开发总成本的一半以上，设计优良的系统可以大大减轻后期维护的工作量。

该系统的维护主要是改正性维护、适应性维护、完善性维护 三个方面的维护。随着用户的使用就不会暴露出一些程序潜藏的缺陷，根据用户反馈的问题来修正系统，保证系统的正常运行，这就是改正性维护。

适应性维护，计算机技术迅速进步，不管软件还是硬件都在快速发展，例如操作系统的升级、运行的软件升级。系统就需要适应就要做出一定的修改来保证系统的兼容、高效的运行。

完善性维护，系统使用一段时间后，随着用户的增多就需要新增一些功能或者对现有的功能进行改进。这个方面的维护也是维护的主要工作，许多优秀的软件都是不断添加改进功能，不断完善起来的。

**第八章 总结**

经历近两个月的开发和设计，完成6400多行的代码和8000多字的设计文档。也算是为大学画上比较圆满的句号。从四月初开始构思，如何实现、选用那些框架、前台页面如何布局等等。用了大概二十天完成了数据库和后台代码的编写，由于之前没有使用SpringMVC+Spring+Mybatis框架的经验，一个错误有时就要好几天才能解决，不过每解决一个错误都是成长。在五月的时候开始编写页面代码用了十天完成，再用十天完成从数据库到页面的连通，最后用一个星期完成开发文档的编写。虽然在开发过程中遇到不少问题麻烦，但是看到现在的作品也是倍感欣慰。

每一行代码都是组成这个系统的关键，每次老师的指导都是顺利开发的关键。

系统虽然开发完成，但是和真正的商业项目还有很大的差距，还需要不断地完善更新。今后踏入社会也要不断学习提升自己的知识技能水平。

**参考文献**

[1] 刘京华.Javaweb开发王者归来[M].北京:清华大学出版社,2010年1月

[2] 王珊、陈红.数据库系统原理教程[M].北京:清华大学出版社,2014年7月

[3] 张海潘 牟永敏.软件工程导论[M].北京:清华大学出版社,2013年1月

[4] 埃尔克.Java编程思想[M].北京:机械工业出版社,2007年1月

[5] 王艳清、陈红.基于SSM框架的智能web系统研发设计[J].计算机工程与设计,2012, 33 (12) :4751-4757

**致 谢**

毕业设计顺利的完成，首先要感谢我的指导老师林盾老师，从定题到完成林盾老师给了很多指导意见。在整个开发过程中林盾老师一直关心完成进度，多次面对面指导讨论问题。在此谨向林老师致以诚挚的谢意和崇高的敬意。

还要感谢湖南科技大学教育学院教育技术系的所有老师。感谢杨世伟、王贤文、林盾、张克敏、张伟平、马培峰、蒋玲玲、张进良、吴新宁等各位老师在我大学四年里的悉心教导，让我系统的学习专业知识成为合格的本科毕业生。

同样感谢陪伴我走过大学四年的同学们，在这人生最珍贵的时光了给我帮助和关心。同学的友谊是我大学中最大的收获，一起分享过喜怒哀乐。

最后向所有参考文献的作者表示谢意。