

运动装备管理及订购系统

摘要

随着时代的发展，21世纪的今天，全球信息化，互联网成为影响人们生活的重要而又不可替代的因素。近年来，电子商务在人们生活中掀起了一股声势浩大的浪潮，逐渐改变着人们的生活方式。人们的经济能力也在时代发展的潮流中逐渐提高，这就意味着更高水平的消费，更繁忙的工作规划，不可避免的会出现上班族因缺乏体育锻炼而导致身体素质下降、没有办法支撑高强度的工作。于是，人们更加清晰地认识到运动于自己而言的重要性。

本系统主要目的在于方便用户选择各种品牌的运动装备，它又拥有一套尽量完整的运动装备供应体系，满足不同层次，不同年龄段，不同运动类型人群的各种需求，致力于打造分类精细化，检索便捷化，品牌多样化，界面舒适化的一体化运动装备订购平台。我从需求分析、可行性分析、概念、逻辑、物理设计等方面对该系统进行了详细分析和设计。

关键词：运动装备 电子商务 需求分析 可行性分析 概念设计 逻辑设计 物理设计

Sports Equipment Management and Ordering System

ABSTRACT

With the development of the times, today in the 21st century, the global informatization, the Internet has become an important and irreplaceable factor affecting people's lives. In recent years, e-commerce has set off a huge wave in people's lives, gradually changing people's lifestyle. People's economic capabilities are also gradually improving in the trend of the times. This means higher levels of consumption and more busy work planning. It is inevitable that office workers will lose their physical fitness due to lack of physical exercise and cannot support it. High-intensity work. As a result, people have become more aware of the importance of exercise for themselves.

The main purpose of this system is to facilitate users to choose various brands of sports equipment. It also has a complete sports equipment supply system as much as possible to meet the various needs of different levels, different age groups and different types of sports. Integrated sports equipment ordering platform with convenient retrieval, diversified brands, and comfortable interface. I conducted a detailed analysis and design of the system from the aspects of requirements analysis, feasibility analysis, concept, logic, and physical design.

Key words: sports equipment, e-commerce, demand analysis, feasibility analysis, conceptual design, logical design, physical design

目 录

一、概述.....	1
1.1 定义.....	1
1.2 背景.....	1
1.3 核心流程.....	2
二、需求分析.....	3
2.1 用户需求.....	3
2.2 系统用例.....	4
2.3 数据流图.....	4
2.4 性能需求.....	4
三、可行性分析.....	6
3.1 经济可行性.....	6
3.2 技术可行性.....	6
3.3 运营可行性.....	6
四、概念设计.....	7
4.1 实体.....	7
4.2 实体之间的联系.....	11
4.3 实体属性局部 E-R 图.....	12
4.3.1 会员属性局部 E-R 图 (vip)	12
4.3.2 游客属性局部 E-R 图 (visitor)	13
4.3.3 运动装备属性局部 E-R 图 (equipment)	13
4.3.4 运动装备种类属性局部 E-R 图 (sports_type)	13
4.3.5 运动装备品牌属性局部 E-R 图 (brand)	13
4.3.6 订单属性局部 E-R 图 (order)	14
4.3.7 订单明细属性局部 E-R 图 (order_detail)	14
4.3.8 管理员属性局部 E-R 图 (admin)	14
4.3.9 管理员类别属性局部 E-R 图 (admin_type)	15
4.4 实体联系局部 E-R 图.....	15
4.4.1 运动装备—运动类型局部 E-R 图.....	15
4.4.2 运动装备—品牌局部 E-R 图	15
4.4.3 订单明细—运动装备局部 E-R 图.....	16
4.4.4 订单明细—订单局部 E-R 图	16
4.4.5 订单—会员局部 E-R 图	16
4.4.6 管理员—管理员类别局部 E-R 图.....	17
4.5 全局 E-R 图.....	17
五、逻辑设计.....	19
5.1 关系模型设计	19
5.1.1 模型分析.....	19
5.1.2 模型设计.....	19
5.1.2.1 会员关系模型	19
5.1.2.2 游客关系模型	19
5.1.2.3 运动装备关系模型	19

5.1.2.4	运动装备品牌关系模型	20
5.1.2.5	运动装备种类关系模型	20
5.1.2.6	订单关系模型	20
5.1.2.7	订单明细关系模型	20
5.1.2.8	管理员关系模型	20
5.1.2.9	管理员类别关系模型	21
5.2	表结构设计	21
5.2.1	会员表结构设计	21
5.2.2	游客表结构设计	21
5.2.3	运动装备表结构设计	21
5.2.4	运动装备品牌表结构设计	22
5.2.5	运动装备种类表结构设计	22
5.2.6	订单表结构设计	22
5.2.7	订单明细表结构设计	22
5.2.8	管理员表结构设计	22
5.2.9	管理员类别表结构设计	23
六、	物理设计	24
6.1	概述	24
6.2	事务数据访问特性	24
6.2.1	数据库查询事务	24
6.2.1.1	用户查询运动装备信息	24
6.2.1.2	用户查询运动装备品牌信息	24
6.2.1.3	用户查询运动装备用途信息	25
6.2.1.4	用户查询订单信息	25
6.2.1.5	管理员查询用户及游客	25
6.2.1.6	管理员查询订单	25
6.2.1.7	查询管理员信息	26
6.2.2	数据库更新事务	26
6.2.2.1	用户注册、注销账号、修改注册资料	26
6.2.2.2	管理员管理订单	26
6.2.2.3	创建、删除管理员	26
6.2.2.4	管理员对运动装备进行管理	27
6.3	数据库索引设计	27
6.3.1	运动装备索引表	27
6.3.2	管理员索引表	28
6.3.3	订单索引表	29
6.4	完整性设计	29
6.4.1	域完整性	30
6.4.2	实体完整性	30
6.4.3	参照完整性	30
6.5	安全性设计	31
6.5.1	数据库不安全因素分析及应对措施	31
6.5.2	安全模型	31

6.5.3 存取控制.....	31
七、系统实现.....	32
7.1 整体实现方案.....	32
7.1.1 整体框架.....	32
7.2 数据库实现.....	33
7.2.1 部署平台.....	33
7.2.2 创建数据表.....	34
7.3 前端实现.....	36
7.3.1 前端功能	36
7.3.2 实现效果	36
7.3.3 功能模块	37
7.3.4 关键数据结构	50
7.3.5 关键技术	53
八、心得体会.....	61
九、参考文献.....	62

同濟大學

数据库课程设计（论文）

1 概述

1.1 定义

在网上购买运动各种运动相关的装备，具体包括在后台进行装备管理，在前端用户进行商品的订购。本系统主要目的在于方便用户选择各种品牌的运动装备，它又拥有一套尽量完整的运动装备供应体系，满足不同层次，不同年龄段，不同运动类型人群的各种需求。同时我们会为运动装备提供不同角度的展示图片，用户操作方便，展现形式自然，并且会为产品定制详细的参数，让用户对产品质量及性能有最全面的把控。管理员可以发布最新商品及相应动态，通过后台控制管理系统的展现形式。致力于打造分类精细化，检索便捷化，品牌多样化，界面舒适化的一体化运动装备订购平台。

背景

随着时代的发展，21世纪的今天，全球信息化，互联网成为影响人们生活的重要而又不可替代的因素。近年来，电子商务在人们生活中掀起了一股声势浩大的浪潮，逐渐改变着人们的生活方式。同时，人们的经济能力也在时代发展的潮流中逐渐提高，这就意味着更高水平的消费，更繁忙的工作规划，不可避免的会出现上班族因缺乏体育锻炼而导致身体素质下降、没有办法支撑高强度的工作。于是，人们更加清晰地认识到运动于自己而言的重要性。

体育锻炼对于人来说，不仅是一种生活方式，更是唤醒自己精神力的兴奋剂。当下的体育运动，不再仅仅是跑步，跳高等力量型运动，更多的比如乒乓球、羽毛球、篮球、足球等技巧型运动也开始被人们所青睐。特别是羽毛球等技巧型运动，各个年龄层次的人群都会参与进来，甚至可以说已经进入了全民运动的局面。即使是尚未在国内流行的网球运动，也不再是贵族性运动。

当然，极致的运动体验必然得有与之相映衬的优质运动装备来供给和满足。只有与运动类型相匹配的运动装备，才能真正发挥出一项运动最本质的意义和最高限度的水平，同时也能给运动人群带来最贴心的保护。身体是革命的本钱，只有保护好身体，拥有强健的体格，才能规划自己的现实生活，走向理想的未来。那么购买运动装备，拥有优质的运动装备便是当下时代人的最佳选择。

21世纪以来，网络购物引领了人们购物的主流。在手机网民方面，数据显示，截止2018年6月，中国手机网民规模达到7.88亿，2018年上半年新增手机网民数量3509万人，与2017年相比增长4.7%。值得一提的是，手机网民占网民数量的比重在持续攀升，2018年占比高达98.3%。尤其是农村地区的网民数量近年来也在不断提升。2018年双11网络促销日，各大电商品台交易量大增，11日24点，天猫双11成交金额定格在2135亿元，首次突破2000亿大关；京东

商城在 11 月 1 日到 11 日累计下单金额 1598 亿元；苏宁易购数据显示，线上线下销售同比增长 132%。各大电商平台都再一次刷新了成交记录。

2016 年，京东商城运动健身部数据显示，中部省份体育消费增速最快，消费者在购买体育用品时，越来越倾向细分的专业运动，并更加关注商品的品质。体育消费已成全民经济，报告预计，随着体育消费潜力更大程度释放，未来 10 年，中国体育产业的年增长率将超过 20%，远超 GDP 增幅。

1.2 核心流程

1.2.1 前端

- ① 用户进入“运动装备订购及管理系统”首页，注册会员并登录系统。
- ② 用户通过浏览导航栏，查看系统对运动装备的品牌分类或运动类型分类，进入自己想要的运动装备分区。
- ③ 用户在搜索栏输入自己想要的目标运动装备。
- ④ 系统在后台数据库中根据用户的关键字进行检索。
- ⑤ 系统响应用户，返回包含所有用户关键字的结果。
- ⑥ 会员用户可以添加购物车，确认购买之后可以提交订单，付款。
- ⑦ 查看物流信息、物流状态等。

1.2.2 后台

- ① 系统总管理员享有对所有用户数据、其他管理员的控制权（禁用启用权）。
- ② 管理员注册、登录后台。
- ③ 对各品类的运动装备进行检索及增、删、查、改。
- ④ 控制前台显示商品动态、降价秒杀等活动。
- ⑤ 对系统进行运行维护，设置浏览日志，方便用户查询购买自己浏览过的商品。

2 需求分析

2.1 用户需求

(1) 前台用户

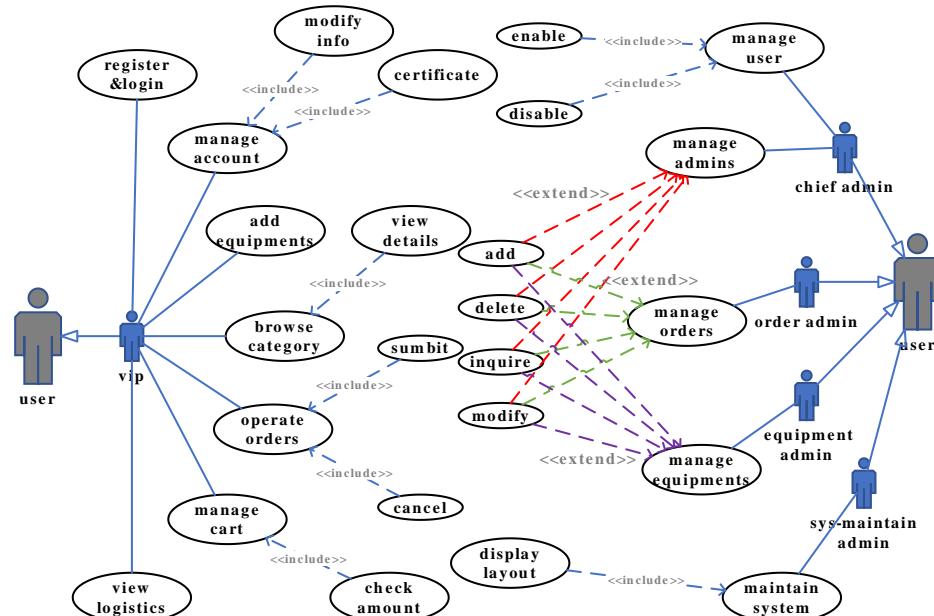
需求	描述
在线注册 (register)	用户首先注册成为系统会员。注册时填写用户名，密码，联系方式。注册之后可以继续完善个人信息（修改密码，实名认证，填写详细地址等等）
分类浏览 (browse category)	不同品牌的运动装备，不同类型的运动装备，会有导航栏进行导航，方便用户寻找相应品牌、相应类型的运动装备。
添加运动装备到购物车 (add equipment)	会员在检索和浏览商品时可以将商品添加到购物车，随时关注该商品的动态。
管理购物车 (manage cart)	会员可以控制购买运动装备的数量，可以确定是否购买该装备，删除运动装备，提交订单，清空购物车。
管理订单 (operate orders)	会员提交订单时，需要将自己的收货地址、收货人姓名等信息进行完善，若还有其他需要，还可以添加备注信息。同时也可取消自己的订单。
付款和退款 (pay and refund)	会员在确定好订单信息之后支付货款，申请退款后可进行退款操作
查看物流信息 (view logistics)	对每已支付订单，会员可以查看物流信息，查询物流状态等。

(2) 后台管理员用户

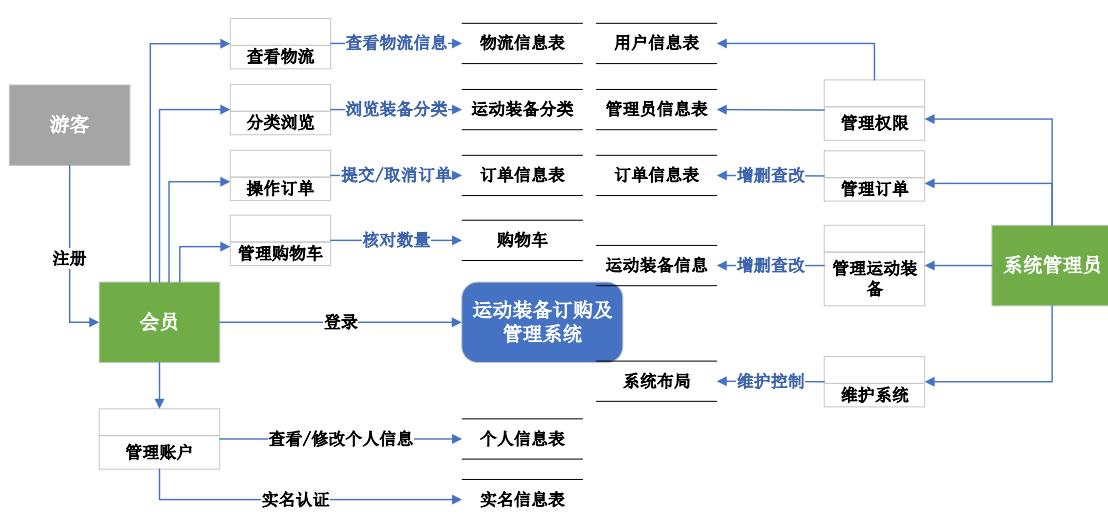
需求	描述
用户和权限管理 (manage authority)	系统总管理员可以对注册的会员用户进行账户注销、查看部分信息操作。更改或设置管理员权限，增加/删除管理员。
运动装备管理 (manage equipments)	运动装备管理员可以对其参数进行登记，比如价格、厂家、名称等录入数据库系统。
订单管理	订单管理员应该对每日的订单数量进行统计，以及

(manage orders)	查阅订单的详细信息,每个订单的处理情况,删除,修改订单。
系统维护 (maintain system)	系统维护管理员可以控制页面运动装备显示布局。

2.2 系统用例



2.3 数据流图



2.4 性能需求

(1) 界面美观, 交互友好: UI 界面采用 Jquery+CSS+HTML 方式实现, 运动装

备布局页面安排合理，整体给人一种清新典雅，操作舒适的感觉。

- (2) 响应速度快：因为系统设计是基于 web 页面的，故网页的响应速度应该较快。再者查询数据库的 SQL 代码的性能也会影响到查询数据的速度，故也会做好这方面的优化。
- (3) 系统稳定性，查询数据准确性，安全可靠性得到保证。
- (4) 可维护性与可扩展性：整体的框架应该具有良好的扩展性，以便于后续补充相应模块的功能；系统有较好的可维护性，在出现数据安全问题时应能及时补救。

装

订

线

3 可行性分析

3.1 经济可行性

因为本系统是基于 web 端实现的，故可直接运行于现在主流的浏览器上，比如 Chrome 浏览器，Firefox 浏览器，Microsoft Edge 浏览器等，在此基础上使用该系统是不需要连接网络的，故而不涉及到服务器方面的问题。

若是考率到部署运行，则需要搭建自己的服务器，因而涉及到服务器购买，可供选择的云端服务器有很多种，比如当前国内流行的有腾讯云、华为云等云服务器，而且它们针对校园用户有相应的优惠，可以考虑在它们提供的服务器上进行部署运行。

3.2 技术可行性

本系统具备相对完备的前端及后端功能，用相应的编程工具进行代码编写，技术要求在可控范围之内。

前端拟采用 Adobe Dreamweaver+JQuery, javascript, CSS, html 等网页编程语言设计导航栏，运动装备展示布局等，而且网页编程语言的兼容性使得前端的设计变得十分方便。在将它们移植到 jsp 页面中，结合 EL 表达式实现前端和后台的融合。

后端拟采用 MyEclipse2017+MySQL8.0.17+JavaWeb 框架进行数据库部署设计，当前比较经典的就是 Spring4+Struts2+Hibernate5 的 JavaWeb 框架，hibernate5 中集成和封装了许多数据库查询语言，理解和使用极为方便。当然还有许多值得借鉴和使用的框架，在后续开发中将会选择使用。

我在前期的学习中接触过 JavaWeb 开发，所以技术方面没有任何问题。

3.3 运营可行性

本系统主要是对运动装备的经营，故主要客户群是运动爱好者，可以通过当前发达的信息传播媒介——网络分享和转发，可以在众多的体育运动爱好者微信群，QQ 群内进行宣传，引起体育爱好者的注意，然后通过对产品质量的严格把控及价格的相对平衡来积累用户，提高用户粘度，获取高点击量，并且根据用户的反馈随时进行系统的调整与修改，为客户提供一个满意的运动装备订购平台。

考虑到现今使用手机网上购物的人群增多，可以将该系统设计成一个 App 或者微信小程序，这样在宣传方面的效果会更有效，而且也方便了用户购物。在后期的系统设计中，会根据个人能力和时间，在 web 端和手机端中选择一种实现。

4 概念设计

4.1 实体

经分析可得，本次“运动装备管理及订购系统”项目涉及到的实体有：

会员 (vip)

游客 (visitor)

运动装备 (equipment)

运动装备种类 (sports_type)

运动装备品牌 (brand)

订单 (order)

订单明细 (order_detail)

管理员 (admin)

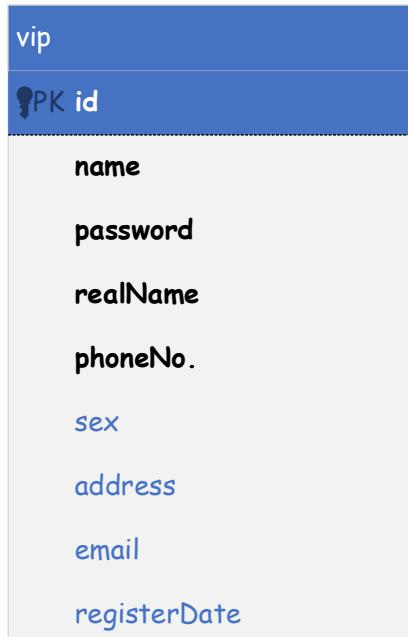
管理员类别 (admin_type)

4.1.1 会员实体

4.1.1.1 实体属性分析

注册网站的会员，其中必须包含的字段是会员名，会员密码，真实姓名，绑定的电话号码。在此基础之上会员可在个人信息页面修改相关信息，比如设置性别，添加常用收货地址，添加邮箱，此外系统将会自动添加注册日期属性。

4.1.1.2 实体属性图（PK 为主键，黑体加粗表示 not null）

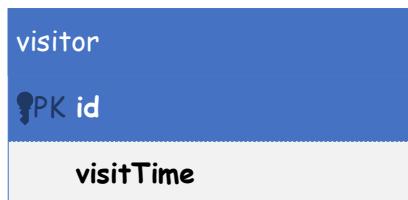


4.1.2 游客实体

4.1.2.1 实体属性分析

游客只有浏览网站的一个动作，故此处在数据库中，只需为其添加浏览时间和游客 id 即可。

4.1.2.2 实体属性图 (PK 为主键, 黑体加粗为 not null)



4.1.3 运动装备实体

4.1.3.1 实体属性分析

运动装备作为项目的主体，其主要属性有 id 编号作为主键，所属运动种类，及品牌，关键字及它的具体文字描述，还有它的定价，折扣，规格，积分，库存，编码，展示图片及用途（比如是穿戴装备还是运动器材），鉴于用途的不同还可能要对相应的用途进行属性划分。

4.1.3.2 实体属性图

装

订

线



4.1.4 运动装备品牌实体

4.1.4.1 实体属性分析

运动装备品牌是顾客购买装时比较看重的，其属性有 id 编号及品牌名称。

4.1.4.2 实体属性图

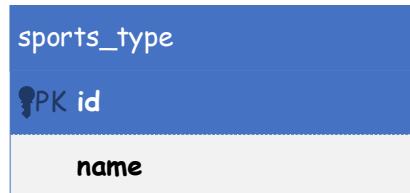


4.1.5 运动装备种类实体

4.1.5.1 实体属性分析

运动装备种类主要包括该装备所属的运动类型的 id 和名称。

4.1.5.2 实体属性图

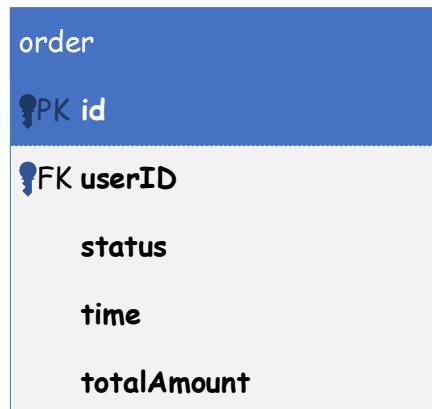


4.1.6 订单实体

4.1.6.1 实体属性分析

订单是在客户端显示的，其主要属性应该包括订单 id 编号，用户 id 编号，订单的状态（处理中，已受理），下单时间和订单总金额。

4.1.6.2 实体属性图

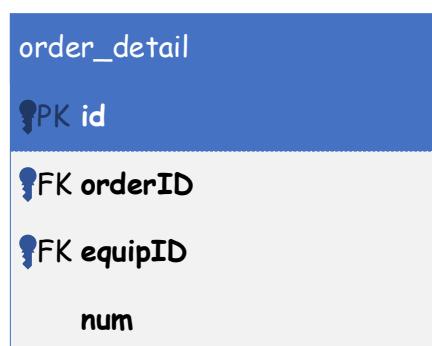


4.1.7 订单明细实体

4.1.7.1 实体属性分析

订单明细主要包含的属性有它所属的订单号，购买的运动装备的信息，及购买的数量。

4.1.7.2 实体属性图

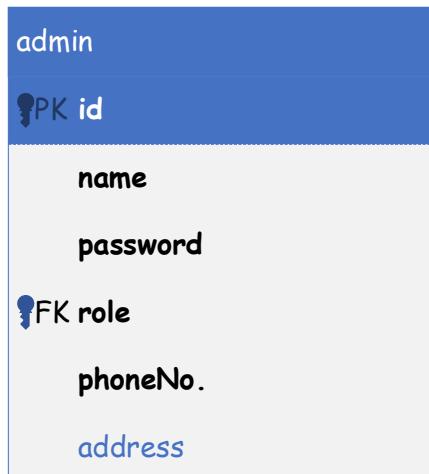


4.1.8 管理员实体

4.1.8.1 实体属性分析

管理员实体主要包含 id 编号，姓名，后台登录系统的密码，扮演的角色以及住址和电话号码。

4.1.8.2 实体属性图



4.1.9 管理员类别实体

4.1.9.1 实体属性分析

后台管理员主要有四种，分属不同的角色并管理不同的模块，故属性有类别编号和类别名，及相应管理模块的 id，并且为多值属性，外键。

4.1.9.2 实体属性图



4.2 实体之间的联系

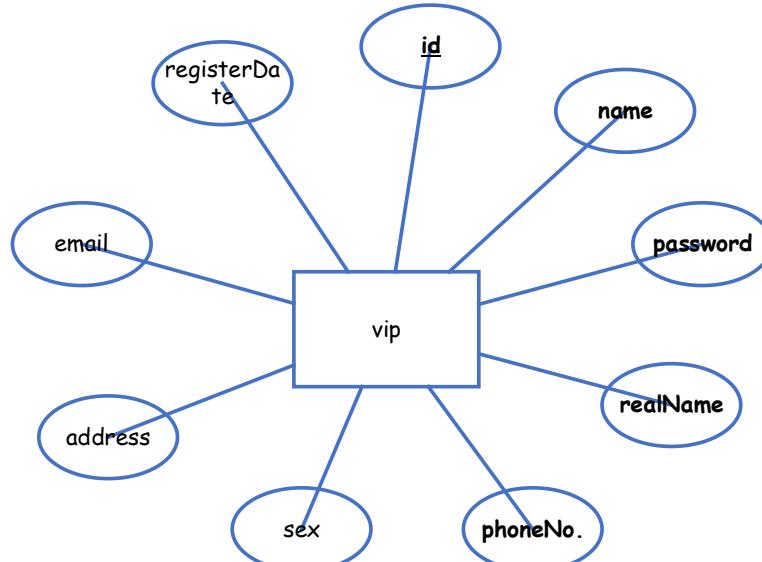
- (1) 运动装备实体与运动类型实体之间存在着互相关联关系，在运动装备实体中有 sportsType 属性，作为外键其关联的属性就是 sports_type 实体的主键

id。

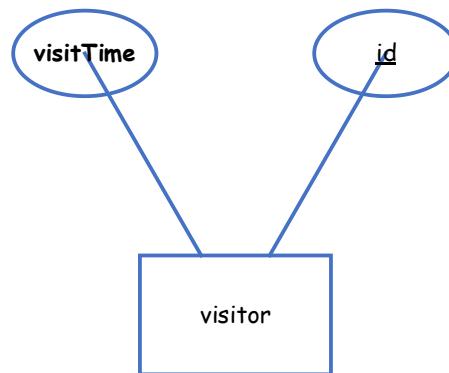
- (2) 运动装备实体与品牌实体之间存在相互关联关系，在运动装备实体存在 brand 属性，为了使数据表更加直观和避免冗余，将每一个品牌的 id 作为主键，并且作为运动装备实体的外键，name 作为每个 id 对应的属性。
- (3) 订单明细实体与订单实体存在相互关联的关系，在订单明细实体中有 orderID 属性，代表其所属的订单号，订单实体中有一个 id 号作为主键来唯一标识每个订单。
- (4) 订单明细实体与运动装备实体存在相互关联的关系，订单明细实体中有外键属性 equipID 作为对订单明细中具体运动装备的标识，在运动装备实体中又有一个装备 id 号作为主键，从而实现每个运动装备的一一对应。
- (5) 订单实体与会员实体存在相互关联的关系，每个订单都唯一所属于一个用户，故有一个用户 id 来标识其所属的会员实体，此 id 作为订单实体的外键，同时也作为会员实体的主键。
- (6) 管理员实体与管理员类别实体之间存在相互关联的关系，在管理员实体中有 role 属性作为外键引用自管理员类别实体的主键 id，通过 id 来识别管理员的角色，每个角色承担相应的任务。
- (7) 管理员类别与会员实体，订单实体，管理员实体存在相互关联的关系，管理员类别表示管理员承担的相应的角色，以及其相应的管理任务，包括管理会员信息，管理订单信息，主管理员管理其他管理员

4.3 实体属性局部 E-R 图

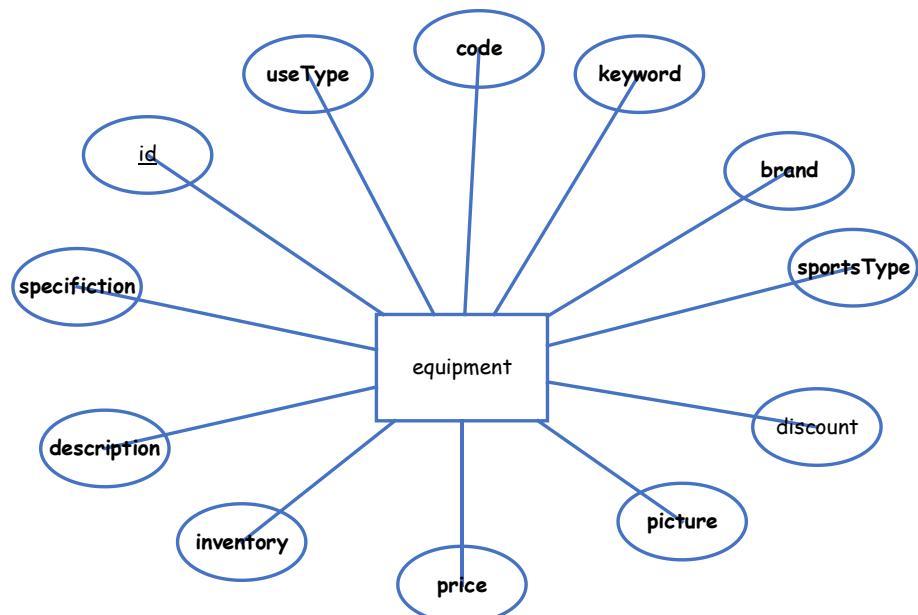
4.3.1 会员实体



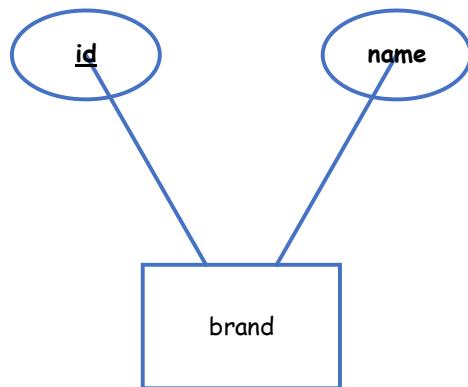
4.3.2 游客实体



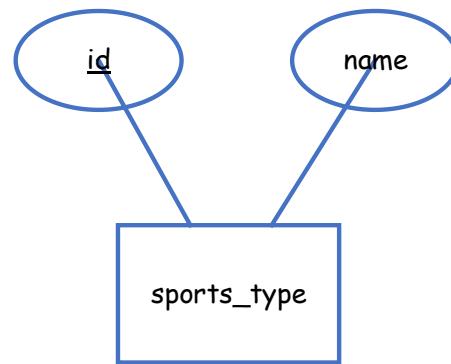
4.3.3 运动装备实体



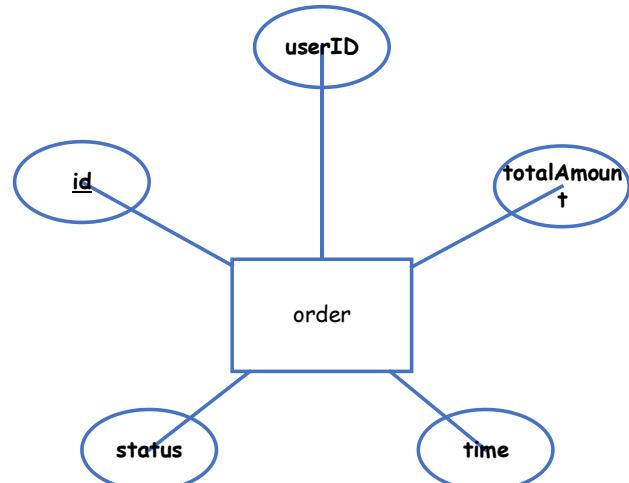
4.3.4 品牌实体



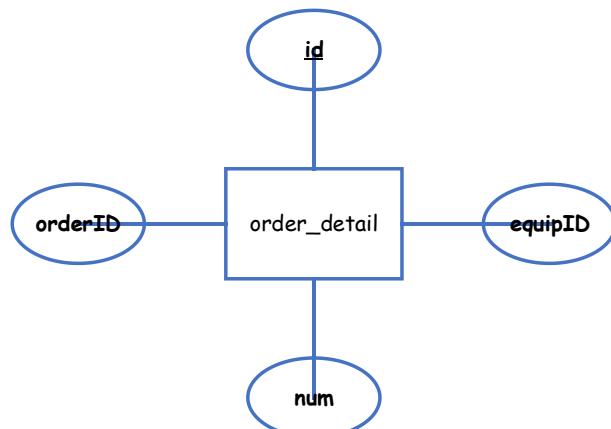
4.3.5 运动装备种类实体



4.3.6 订单实体



4.3.7 订单明细实体

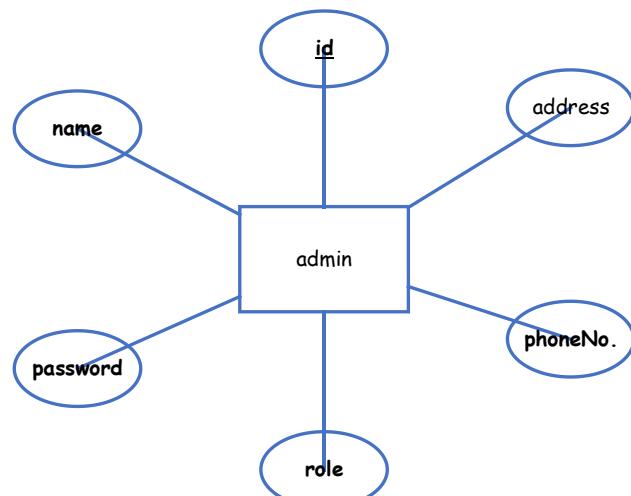


4.3.8 管理员实体

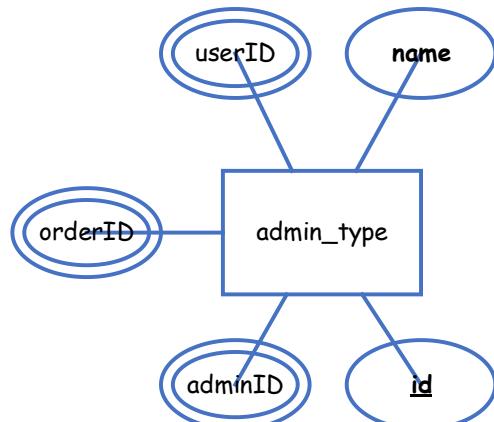
装

订

线

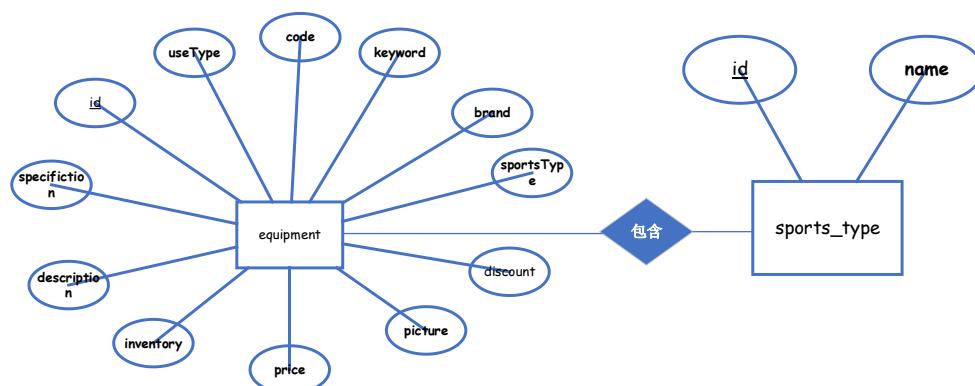


4.3.9 管理员类别实体

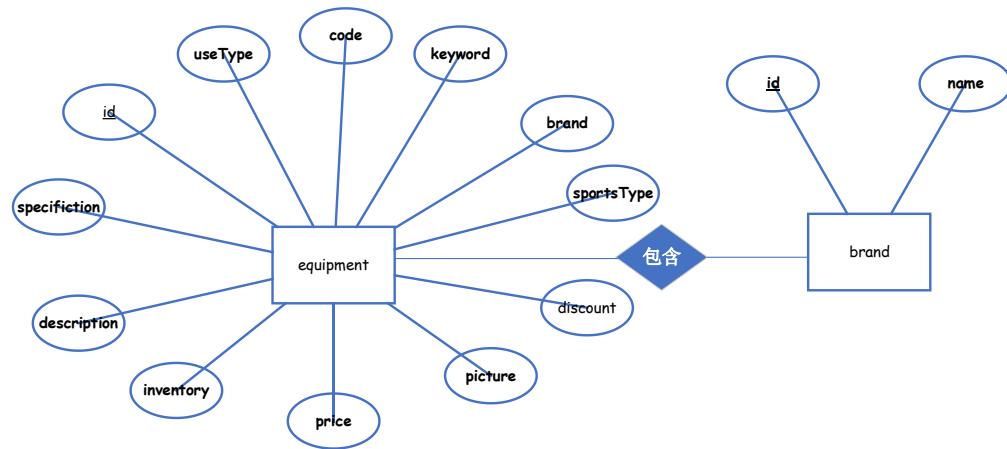


4.4 实体联系局部 E-R 图

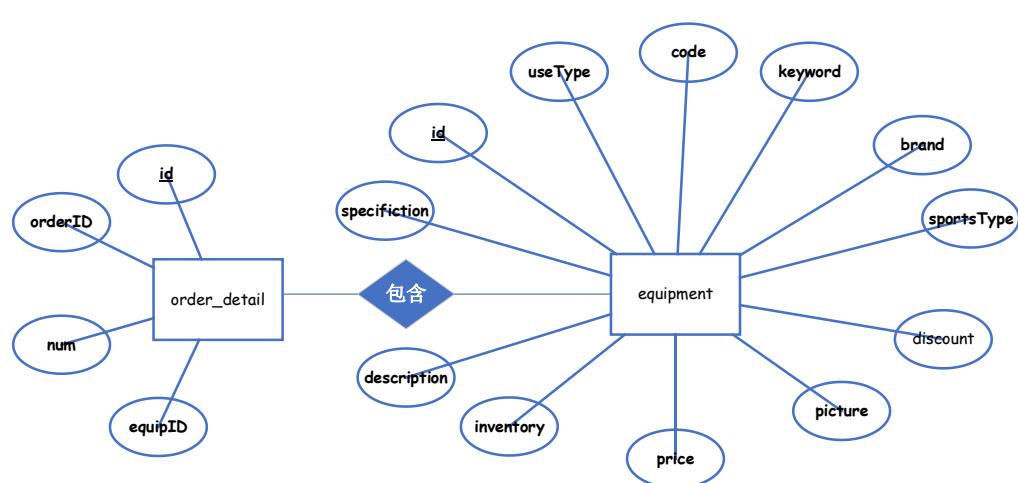
4.4.1 运动装备实体与运动类型实体



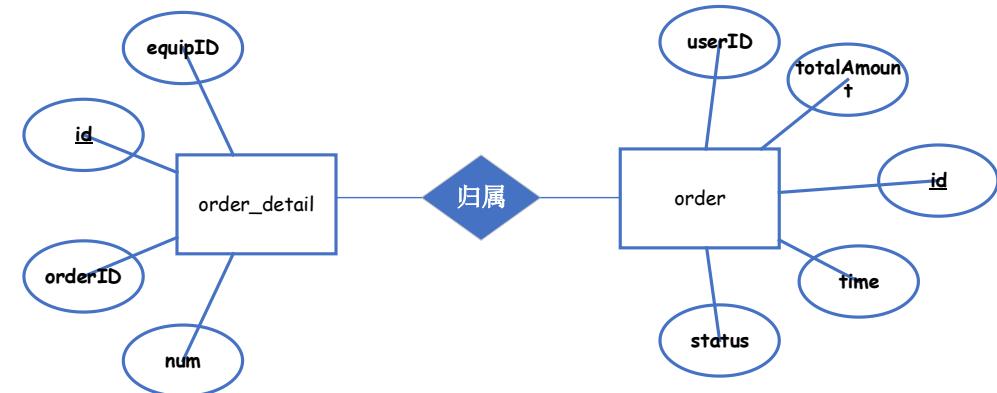
4.4.2 运动装备实体与品牌实体



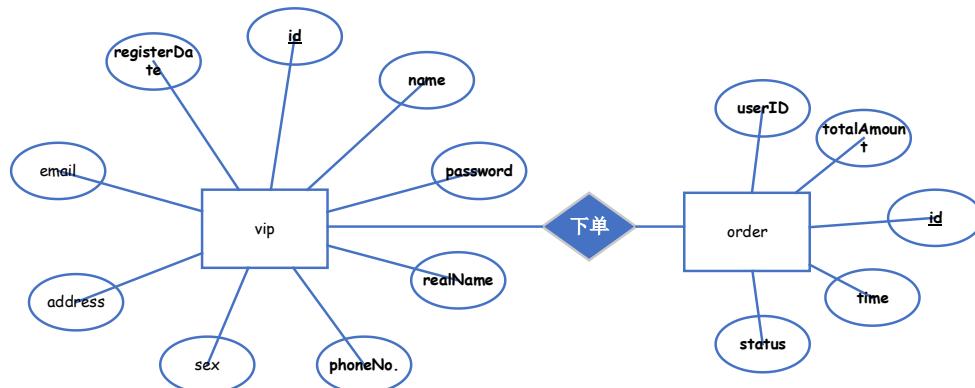
4.4.3 订单明细实体与运动装备实体



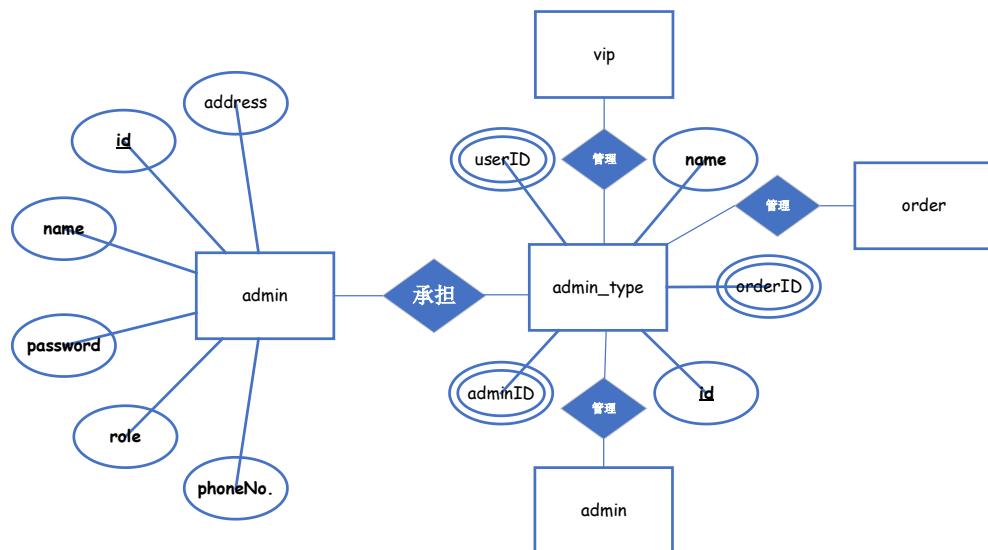
4.4.4 订单明细实体与订单实体



4.4.5 订单实体与会员实体

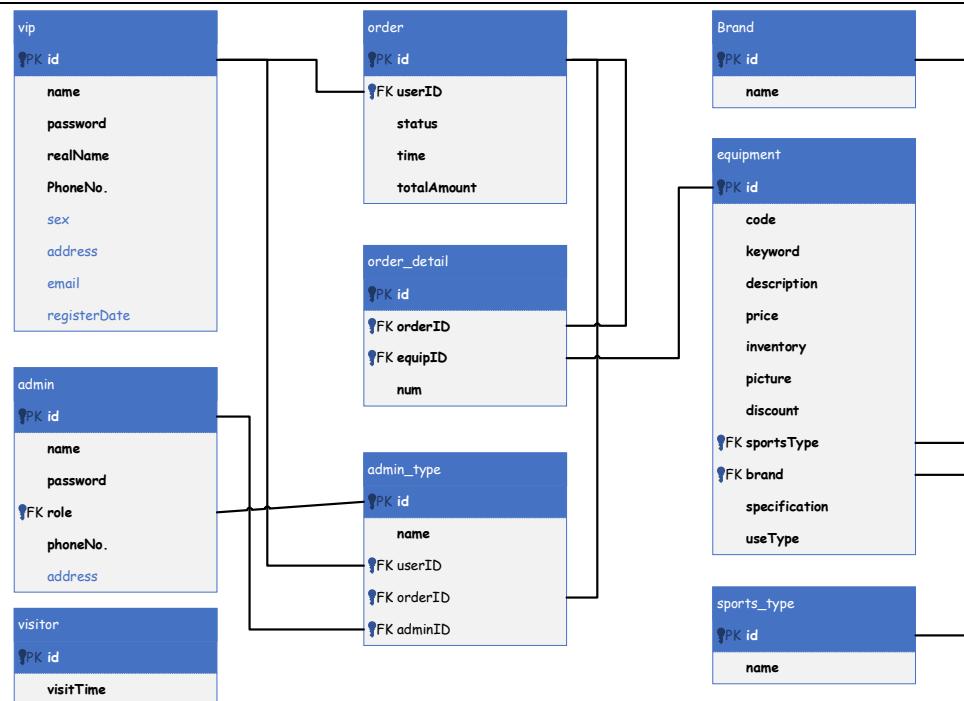


4.4.6 管理员实体与管理员类别实体



4.5 全局 E-R 图

装
订
线



装

订

线

5 逻辑设计

5.1 关系模型设计

5.1.1 模型分析

在概念设计中我总结出了本次“运动装备管理及订购系统”的实体及属性，并且对它们之间的联系有了清晰地认识和划分。在全局 E-R 图中，将整个系统的数据库实体串联起来，其逻辑结构便一目了然地展现出来了。

关系模型便是在上次概念设计的基础上，将每个实体的属性用元组模式表示出来，并标明主键，外键等码属性。

5.1.2 模型设计

5.1.2.1 会员关系模型

- (1) 会员关系模型的主键: id;
- (2) 非空属性: 会员名, 登录密码, 真实姓名, 电话号码等属性;
- (3) 关系模式如下

$vip = (\underline{id}, name, password, realName, phoneNo., sex, address, email, trgiaterDate);$

5.1.2.2 游客关系模型

- (1) 游客关系模型的主键:id;
- (2) 非空属性: 访问时间;
- (3) 关系模式如下

$visitor = (\underline{id}, visitTime);$

5.1.2.3 运动装备关系模型

- (1) 运动装备关系模型的主键:id 编号;
- (2) 非空属性: 装备识别码, 关键字, 描述, 定价, 库存, 展示图片, 运动类型, 品牌等属性;
- (3) 关系模式如下

$equipment = (\underline{id}, code, keyword, description, price, inventory, picture, discount, sportsType, brand, specifiation, useType);$

5.1.2.4 运动装备品牌关系模型

- (1) 运动装备品牌关系模型的主键:id；
- (2) 非空属性： name 属性；
- (3) 关系模式如下

$brand = (\underline{id}, name);$

5.1.2.5 运动装备种类关系模型

- (1) 运动装备种类关系模型的主键:id 编号；
- (2) 非空属性： name 属性；
- (3) 关系模式如下

$sports_type = (\underline{id}, name);$

5.1.2.6 订单关系模型

- (1) 订单关系模型的主键:id 编号；
- (2) 外键:用户 id 编号， 用于标明该订单的归属；
- (3) 非空属性： 订单状态、下单时间、总金额等属性。
- (4) 关系模式如下

$order = (\underline{id}, userID, status, time, totalAmount);$

5.1.2.7 订单明细关系模型

- (1) 订单明细关系模型的主键:订单明细 id 编号；
- (2) 外键:所属订单 id,该订单含有的运动装备 id 号；
- (3) 非空属性约束： 数量不能为空且大于零。
- (4) 关系模式如下

$order_detail = (\underline{id}, orderID, equipID, num);$

5.1.2.8 管理员关系模型

- (1) 管理员关系模型的主键:id 编号；

- (2) 外键:管理员扮演的角色 id 编号;
 (3) 非空属性: 姓名、登录后台的密码, 电话号码不能为空;
 (4) 关系模式如下

$admin = (\underline{id}, name, password, role, phoneNo., address);$

5.1.2.9 管理员类别关系模型

- (1) 管理员类别关系模型的主键:id 编号;
 (2) 外键:该角色管理的实体编号,分别为用户 id, 订单 id, 管理员 id;
 (3) 非空属性: 角色名属性;
 (4) 关系模式如下

$admin_type = (\underline{id}, name, userID, orderID, adminID);$

5.2 表结构设计

表结构设计部分是根据关系模型对其进行二维表格化。同时也要对主键, 外键, 非空进行约束。

5.2.1 会员表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
<code>id</code>	<code>int</code>	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
<code>name</code>	<code>varchar</code>	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<code>password</code>	<code>varchar</code>	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<code>realName</code>	<code>varchar</code>	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<code>phoneNo.</code>	<code>varchar</code>	15	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<code>sex</code>	<code>varchar</code>	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
<code>address</code>	<code>varchar</code>	50	0	<input type="checkbox"/>	<input type="checkbox"/>	
<code>email</code>	<code>varchar</code>	20	0	<input type="checkbox"/>	<input type="checkbox"/>	
<code>registerDate</code>	<code>datetime</code>	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.2 游客表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
<code>id</code>	<code>int</code>	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
<code>name</code>	<code>varchar</code>	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.3 运动装备表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
code	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
keyword	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
description	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
price	float	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
inventory	int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
picture	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
discount	float	4	0	<input type="checkbox"/>	<input type="checkbox"/>	
sportsType	int	3	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
brand	int	3	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
specification	varchar	30	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
useType	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.4 运动装备品牌表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
name	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.5 运动装备种类表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
name	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.6 订单表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
userID	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
status	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
time	datetime	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
totalAmount	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.7 订单明细表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
orderID	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
equiplD	int	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
num	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.8 管理员表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
name	varchar	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
password	varchar	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
role	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
phoneNo.	varchar	15	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
address	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.9 管理员类别表结构设计

名	类型	长度	小数点	不是 null	虚拟	键
id	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
name	varchar	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
userID	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
orderID	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
adminID	int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

装

订

线

6 物理设计

6.1 概述

数据库的物理设计是数据库项目设计的后半段。数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于给定的计算机系统。为给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，称为数据库的物理设计。

物理设计的目的主要是有效地实施逻辑模式，确定所采取的存储策略。此阶段是以逻辑设计的结构作为输入，并结合具体的 DBMS 特点与存储设备的特性进行设计，选定数据库在物理设备上存储结构与存取方法。

数据库的物理设计分为两步：

- 确定数据库的物理结构，在关系数据库中主要指存储结构和存取方法。
- 对物理结构进行评价，评价的重点是时间和空间效率。

如果评价结果满足设计要求，则可进入物理实施阶段，否则还需重新设计或修改物理结构，甚至需要修改逻辑设计阶段的数据模型。

6.2 事务数据访问特性

6.2.1 数据库的查询事务

6.2.1.1 用户查询运动装备信息

【事务内容】用户在运动装备列表浏览装备信息的时候，可以通过点击装备图片信息或者装备的描述信息即可查看商品的详细参数信息，诸如装备识别码，装备关键字，装备库存，更多维度展示图片，所属的运动类型，使用途径，价格折扣等。若要查看商品的具体品牌，则将外键 brand 和 brand 表的 id 相连接；若要查看运动装备所属具体运动类型，则可将 sportsType 和 sports_type 表的 id 相连接。

【操作频率】20000 次/天

【性能要求】2s 内完成

6.2.1.2 用户查询运动装备品牌信息

【事务内容】用户可以在检索栏搜索自己想要购买的运动装备品牌，通过点击该运动装备品牌的关键字或者 logo 即可进入该品牌类的运动装备专区，挑选自己心仪的运动装备，用户可以通过使用类型 useType，运动类型 sportsType 等字段来查询具体品类的运动装备信息。

装
订
线

【操作频率】2000 次/天

【性能要求】1s 内完成

6.2.1.3 用户查询运动装备用途信息

【事务内容】用户可以在检索栏搜做各种使用用途的运动装备，在各种使用用途列表内点击相应的使用用途（运动器材、穿戴装备等），即可看到系列装备，如果想看不同的品牌该用途类别的装备，则将 useType 表中的 sportsTypeID 与 brand 表中的 id 连接，即可找到相应的目标品牌的运动装备，至于要查看运动装备的详细信息，则操作方式同上。

【操作频率】2000 次/天

【性能要求】1s 内完成

6.2.1.4 用户查询订单信息

【事务内容】用户在购买运动装备之后，在“我的订单”一栏可以查看自己的当前已付款订单及详细信息，包括订单编号 id，订单当前状态 status（处理/未处理），下单时间 time，订单总金额 totalAmount，购买数量 num 以及订单中所含有的运动装备 id 编号，若是想查看运动装备的详细信息，则将 equipID 和 equipment 表中的 id 连接，即可得到当前购买装备的详细规格参数信息。

【操作频率】1000 次/天

【性能要求】0.5s 内完成

6.2.1.5 管理员查询用户及游客

【事务内容】用户管理员可在后台对当前系统平台的用户进行查询和管理，包括对其详细信息的掌握，及用户的启用和禁用功能，对其留言评论进行相应的处理。当查看平台用户信息的时候，需要将管理员类型表 admin_type 的 userID 与会员表的 id 相连接。显然 admin_type 表中的 userID 是一个多值属性且允许为空的。用户管理员查询游客时，需要根据系统为每位游客自动编的 id 进行查询。

【操作频率】100 次/周

【性能要求】0.1s 内完成

6.2.1.6 管理员查询订单

【事务内容】订单管理员可实施对订单的管理，它可以查看所有用户的订单，通过将 adminType 表中的 orderID 和 order 表中的 id 相连接来实现。至于订单的具体信息，其查看方式同前面。

【操作频率】1000 次/周

【性能要求】1s 内完成

6.2.1.7 查询管理员信息

【事务内容】通过查询 admin 表来查询管理员的 id 编号，姓名 name，登录系统的密码 password，及角色 role，电话号码，地址等基本信息。

【操作频率】10 次/周

【性能要求】0.1s 内完成

6.2.2 数据库的更新事务

6.2.2.1 用户注册、注销账号、修改注册资料

【事务内容】需要访问 vip 表。用户注册时，需要向 vip 表中添加新的记录，其中相对比较隐私的记录可设置为 NULL；用户注销账号时需要通过索引找到要删除的 vip 用户的记录，同时需要将该用户在整个系统平台中的相关表中的信息都要删除，采用级联删除的方式即可完成；用户修改资料时，需要找到该用户在 vip 表中的相应记录，并对相应字段的值做出修改，并更新 vip 表。

若系统中有游客进行访问，则在 visitor 表中为其创建一条记录，包括系统自动为其生成的 id 及访问时间。若想删除游客记录，则按照某个规则，比如访问时间久远程度或者频繁程度执行删除操作，定期清理数据库。

【操作频率】100 次/天

【性能要求】3s 内完成

6.2.2.2 管理员管理订单

【事务内容】当用户在提交订单之后，管理员便需要在 order 表中添加一条相应的订单记录，包括订单所属的用户 id 等字段值，同时还要对订单明细表 order_detail 进行更新，插入一条订单明细记录，表示其所属订单 id,商品 id 及数量等内容。当用户订单完成时，还可对用户订单进行删除操作，同时要级联删除订单明细表中的相应记录。

【操作频率】100 次/周

【性能要求】1s 内完成

6.2.2.3 创建、删除管理员

【事务内容】需要操作 admin 表，创建管理员时，在 admin 表中添加一条记

录，并为其创建 id 号，创建角色 role，填写其他详细信息等；删除管理员时，按照索引在 admin 表中找到该管理员的记录，并将其删除，对于该管理员在后台中的所有事务进行解除。

【操作频率】10 次/月

【性能要求】0.5s 内完成

6.2.2.4 管理员对运动装备进行管理

【事务内容】需要操作 equipment 表。添加运动装备时，需要在 equipment 表中添加相应的运动装备记录，每个字段填写装备对应的信息。删除运动装备时，需要根据索引找到运动装备在 equipment 表中的具体位置，对该装备记录执行删除操作，同时要对订单明细表 order_detail 中该装备的记录信息进行级联删除。更新运动装备某个属性信息时，找到对应记录并对相应字段值做出更新。

【操作频率】50 次/天

【性能要求】0.5s

6.3 数据库索引设计

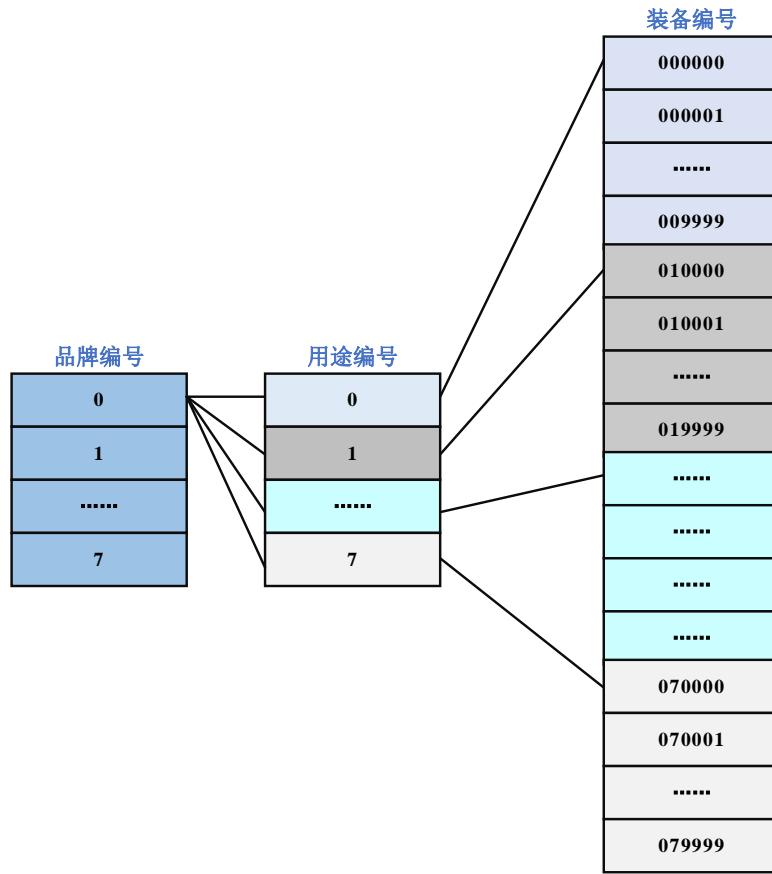
6.3.1 运动装备索引表

运动装备数据是本系统中的最主要的数据，其数量可能会达到成千上万条，设计一个高效的检索算法对用户体验至关重要。故在此需要对运动装备建立索引，运动装备的主码是运动装备编号（equipment 表中的 id），在该表中还有两个主要的属性——useType 和 sportsType 作为外键存在。考虑到用户检索的时候主要有两种不同的方式——按照运动装备品牌检索和按照运动装备用途检索。此处设计两种不同的检索方式。

【方式 1】

通过多级索引，第一级索引为装备品牌编号，假设品牌共有 8 种，则分配给该级索引 1 位索引号；第二级索引为装备用途编号，假设用途共有 8 种，则分配给该级索引 1 位索引号；第三级索引为装备编号，由于装备的数目可能非常庞大，假设每个品牌每种用途的装备数目有 10000 个，则分配给该级索引 6 位索引号。

装 订 线



【方式二】

通过多级索引，索引方法同【方式 1】，但是此时前两级索引进行交换，生成的索引和【方式 1】相同，但是每个商品对应的编号却和上图不同了，这种方式的第一级索引以装备用途基准，故索引得到的装备总是先以用途为参考。

6.3.2 管理员索引表

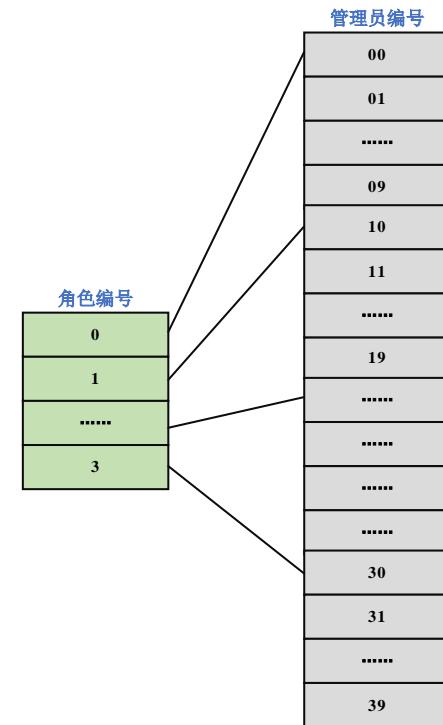
管理员是该系统中必不可少的部分，庞大的用户基数及运动装备基数必然导致庞大的订单量，故各种类型的管理员数量都不能太少，以便于后台的管理，从而减少服务器的压力，让系统运行更加流畅，满足客户端极致的体验。

在管理员索引设计方面，我采用多级索引。管理员有 4 种角色，故第一级索引是管理员角色编号，索引位是 1 位；每种角色的管理员数量都不能太少（假设数量为 10），故第二级索引是某角色管理员编号，索引位是 1 位。

装

订

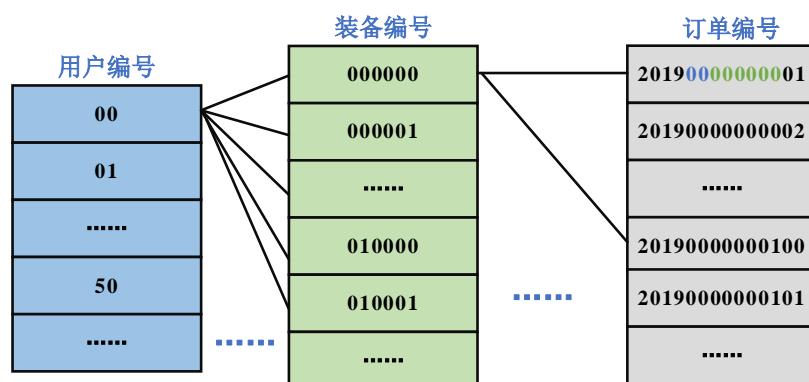
线



6.3.3 订单索引表

庞大的用户基数必定会有庞大的订单量，故查询某个订单也需要加快查询速度。每个订单都是和唯一的用户和相应的装备绑定在一起的，但一个用户可能有多个相同订单，所以订单的编号必须在订单所属用户及包含商品基础上进行索引。

采用多级索引，第一级索引为用户编号，假设为索引位是 2 位；第二级索引位是装备编号，在前面的设计中可知索引位是 6 位；第三级索引位订单编号，加入下单时间作为订单编号。



6.4 完整性设计

数据库完整性是指数据库中的数据在逻辑上的一致性、正确性、有效性和相容

性。数据库完整性由各种各样的完整性约束来保证，因此数据库完整性设计就是数据库完整性约束的设计。数据库完整性可以由 DBMS 或者应用程序来实现，基于 DBMS 的完整性约束作为模式的一部分存入数据库中。

数据库完整性包括域完整性，实体完整性，参照完整性等。此处主要讨论提及的 3 种完整性。

6.4.1 域完整性

域完整性是对数据表中的字段属性的约束，通常是指数据的有效性，包括字段的值域、字段的类型及字段的有效规则等约束，它是由确定关系结构时所定义的属性决定的。限制字段数据类型、格式、值域范围、是否允许空值等。域完整性确保输入的值有效。

域完整性限制了某些属性中出现的值，把属性限制在有限集合中，可以使用 CHECK 约束、UNIQUE 约束、DEFAULT 默认值、IDENTITY 自增、NOT NULL、NULL 保证列值的完整性。

6.4.2 实体完整性

实体完整性是对关系中的记录唯一性，也就是主键的约束。即关系中主码属性值不能为空且不能有两个值相同。定义表中的所有行能唯一的标识，一般用主键，唯一索引 unique 关键字。

6.4.3 参照完整性

参照完整性是对关系数据库中建立关联关系的数据表之间的数据参照引用的约束，即对外键的约束。参照完整性是指关系中的外键必须是另一个关系中的主键，或是空值。参考完整性用于维护表间数据的完整性、有效性和一致性。一般通过建立外键联系另一表的主键来实现。

6.5 安全性设计

数据库安全性是指保护数据库以防止不合法使用所造成的数据泄露、更改或者损坏。系统安全保护措施是否有效是数据库主要技术指标之一。

6.5.1 数据库不安全因素分析及应对措施

- 1) 非授权用户对数据库的恶意存取和破坏

措施：用户身份鉴别，存取控制，视图

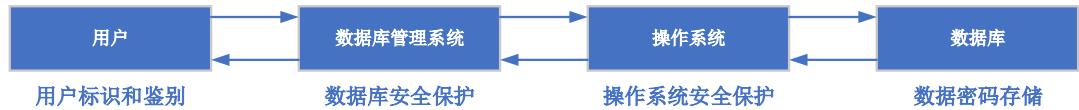
- 2) 数据库中重要或敏感的数据被泄露

措施：强制存取控制，数据加密存储，加密传输

3) 安全环境比较脆弱

措施：加强计算机系统的安全性，建立完整的可信标准

6.5.2 安全模型



6.5.3 存取控制

数据库安全最重要的一点就是确保将权限授予有资格访问数据库的用户。该功能就是通过存取控制实现的。SQL 语句中使用 GRANT 和 REVOKE 语句来实现向用户授予或收回权限。

装

1) GRANT

GRANT 授予用户权限，其具体执行语句如下：

```
GRANT...  
ON...  
TO...  
[ with grant option ];
```

订

其含义为将对指定操作对象的指定操作权限授予指定用户。可以由数据库管理员、数据库对象创建者，也可以是已经用于该权限的用户。接收授权的用户可以使一个或多个具体的用户，也可以是全体用户。其中 with grant option 就是获得某种权限的用户可以将权限或其子集授予其他用户，但不允许循环授权，也就是被授权者不能将权限授予其祖先或者授权者。若没有该语句，则授权不能向下传播。

线

2) REVOKE

REVOKE 收回用户权限，其具体执行语句如下：

```
REVOKE...  
ON...  
FROM...  
[ cascade | restrict ];
```

其含义为将指定操作对象的指定操作权限从指定用户收回。使用该语句的为数据库管理员或者其他授权者。此处默认为 CASCADE，即没有自动执行级联操作，只是收回了该用户的权限，该用户授予的其他用户的权限保留，若使用 RESTRICT 就是收回该用户

7 整体实现方案

7.1 整体框架

本次课程设计我的主体是体育运动装备管理系统，实现了用户日常所需的基本功能。我采用设计模式为经典的 WEB 项目 MVC 模式，即模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

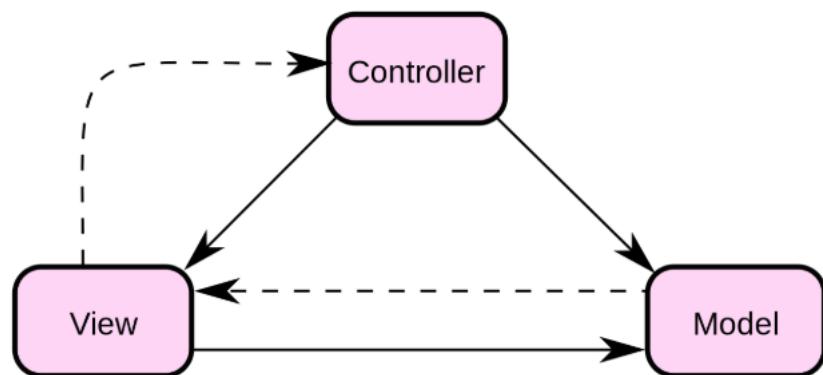
Model (模型) 表示应用程序核心（比如数据库记录列表）。

View (视图) 显示数据（数据库记录）。

Controller (控制器) 处理输入（写入数据库记录）。

Model (模型) 是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。View (视图) 是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。Controller (控制器) 是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

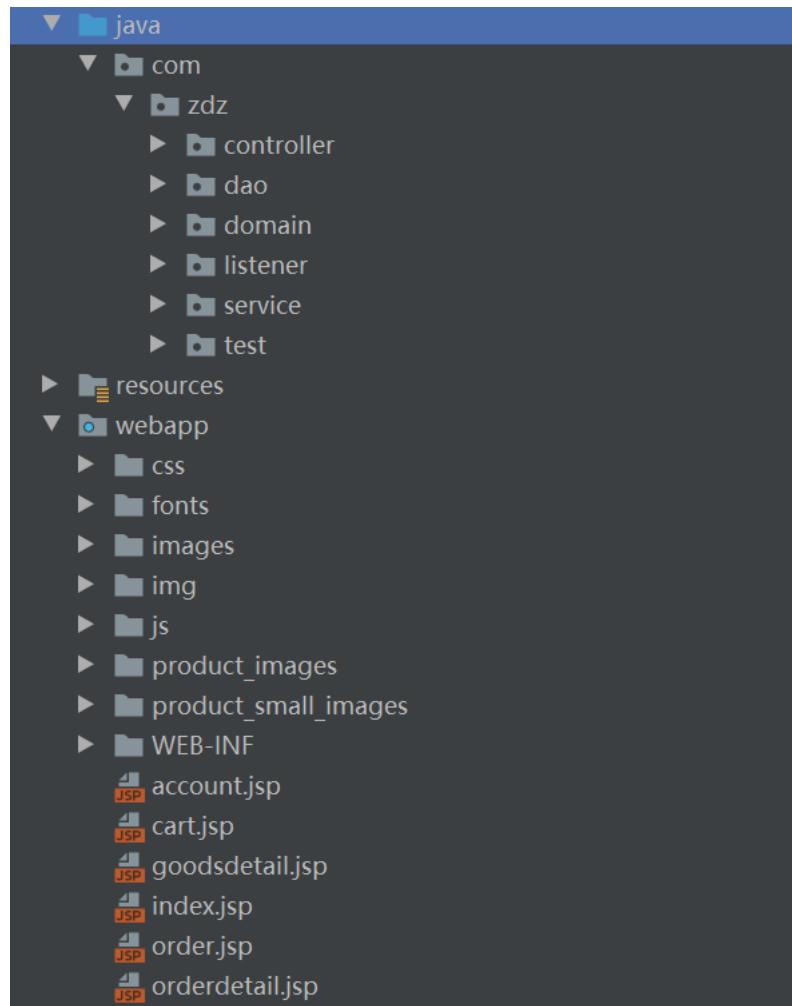
其主要调用及依赖关系如下图所示：



根据其设计的特性和采用的方法，我本次选择的设计框架为基于 Java 的 SSM 框架，该框架（Spring+SpringMVC+MyBatis）框架集由 Spring、MyBatis 两个开源框架整合而成（SpringMVC 是 Spring 中的部分内容）。常作为数据源较简单的 web 项目的框架。它的特点正好符合了我本次课程设计的主题和初衷、目的。所以，基于该框架，行开始自学到边学边用，再到逐渐掌握该框架一些底层的内容，最后完成独立开发该系统，是一个循序渐进，不断探索和尝试的过程。

该框架是将整个系统彻底前后端分离，有助于复杂应用程序的管理，可以在关注一个方面。可以在不依赖业务逻辑的情况下专注试图设计，同时也让应用的测试变得更加容易。简化分组开发，可以并行设计，不同的人各司其职。

在本次“体育运动装备订购管理系统中”，利用 MVC 设计模式，我的代码结构大致如下：



装
订
线

7.2 数据库实现

7.2.1 部署平台

本次课设基于 JavaWeb 开发，经过一番资料搜索和询问，决定采用最常用的 Tomcat9+IntelliJIDEA 进行开发，tomcat 是充当 web 端容器和服务器，用来部署项目和运行项目，是整个开发中的核心组件。本次开发平台为 Windows 系统，运行在主流浏览器端（Chrome 浏览器、FireFox 浏览器等）。由于现在的主流开发方式及部署平台都于此相似，且具有较高的稳定性和可查错性，本次开发中我采取的方式便是如此。

7.2.2 创建数据表

在本次开发过程中，在以前的课程报告中已经进行了逻辑设计、结构设计等一系列前导设计，包括数据表结构的设计、类与类之间的关联、本系统参与者等等。在设计本系统之前，我又重新温习一遍，理清表与表之间的关系，因此，我进行了系统的设计和思考、分析，对前期做的不够好的地方又重新进行了改进，于是数据库表的建立也发生了一些变化。

1) 管理员表

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
password	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
role	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
phoneNo	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
address	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		

管理员表主要包含的字段就是姓名、登录密码和角色，包括该管理员是订单管理员还是用户管理员等等。

2) 品牌种类表

id	name
1	lining
2	tebu
3	qiaodan
4	anta
5	san
6	peak

本系统开发设计中，由于是体育运动装备管理系统，因此我寻找了几个国内比较知名的运动品牌作为数据来源，主要就是李宁、特步、安踏、乔丹等，由于我平时关注比较多，因此对其销售的商品种类有一个大概的了解。

3) 功能表

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	PK1	编号
name	varchar	20	0	<input type="checkbox"/>	<input type="checkbox"/>		功能菜单
parentid	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>		父id
url	varchar	50	0	<input type="checkbox"/>	<input type="checkbox"/>		链接
isleaf	bit	1	0	<input type="checkbox"/>	<input type="checkbox"/>		订单是否离开
nodeorder	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>		订单顺序

这个表是后台管理时用到的功能表，这张表主要描述功能种类，父子 id 已及该功能对应的相关链接。其次就是和订单相关的信息，包括订单信息、订单是否生效

等等。

4) 商品表

id	code	gid	bid	name	description	price	picture	inventory	status	shop	location	img2
1 158798	1	1	1	李宁运动器材	李宁 LI-NING C	0000000399	ln-eq-1.jpg	0165	1	李宁官方旗舰店	上海	ln-eq-1-sm-2.jpg
2 101951	1	1	1	李宁运动器材	李宁篮球韦德7I	0000000484	ln-eq-2.jpg	0224	1	李宁官方旗舰店	上海	ln-eq-2-sm-2.jpg
3 183619	1	1	1	李宁运动器材	李宁AE15李宁I	0000000356	ln-eq-3.jpg	0197	1	李宁官方旗舰店	上海	ln-eq-3-sm-2.jpg
4 173273	1	1	1	李宁运动器材	李宁 LI-NING 3	0000000315	ln-eq-4.jpg	0278	1	李宁官方旗舰店	上海	ln-eq-4-sm-2.jpg
5 102478	1	1	1	李宁运动器材	LI-NING李宁丘	0000000255	ln-eq-5.jpg	0256	1	李宁官方旗舰店	上海	ln-eq-5-sm-2.jpg
6 149447	1	1	1	李宁运动器材	李宁LBDM780	0000000271	ln-eq-6.jpg	0194	1	李宁官方旗舰店	上海	ln-eq-6-sm-2.jpg
7 187685	2	1	1	李宁运动鞋	李宁 LI-NING C	0000000246	ln-sh-1.jpg	0177	1	李宁官方旗舰店	上海	ln-sh-1-sm-2.jpg
8 162810	2	1	1	李宁运动鞋	李宁篮球韦德7I	0000000403	ln-sh-2.jpg	0127	1	李宁官方旗舰店	上海	ln-sh-2-sm-2.jpg
9 109543	2	1	1	李宁运动鞋	李宁AE15李宁I	0000000483	ln-sh-3.jpg	0260	1	李宁官方旗舰店	上海	ln-sh-3-sm-2.jpg
10 115025	2	1	1	李宁运动鞋	李宁 LI-NING 3	0000000351	ln-sh-4.jpg	0270	1	李宁官方旗舰店	上海	ln-sh-4-sm-2.jpg
11 145691	3	1	1	李宁运动服	李宁官网男子短	0000000360	ln-wr-1.jpg	0223	1	李宁官方旗舰店	上海	ln-wr-1-sm-2.jpg
12 184262	3	1	1	李宁运动服	【李宁AYKKMO!	0000000443	ln-wr-2.jpg	0193	1	李宁官方旗舰店	上海	ln-wr-2-sm-2.jpg
13 129275	3	1	1	李宁运动服	李宁 (LI-NING	0000000458	ln-wr-3.jpg	0246	1	李宁官方旗舰店	上海	ln-wr-3-sm-2.jpg
14 197655	3	1	1	李宁运动服	【华晨宇心选】	0000000468	ln-wr-4.jpg	0182	1	李宁官方旗舰店	上海	ln-wr-4-sm-2.jpg
15 171012	2	2	特步运动鞋	特步男鞋运动鞋	0000000381	tb-sh-1.jpg	0131	1	特步官方旗舰店	上海	tb-sh-1-sm-2.jpg	
16 106856	2	2	特步运动鞋	特步男鞋运动鞋	0000000360	tb-sh-2.jpg	0241	1	特步官方旗舰店	上海	tb-sh-2-sm-2.jpg	
17 116497	2	2	特步运动鞋	【特步880119】	0000000439	tb-sh-3.jpg	0260	1	特步官方旗舰店	上海	tb-sh-3-sm-2.jpg	
18 112023	2	2	特步运动鞋	特步男鞋篮球鞋	0000000358	tb-sh-4.jpg	0100	1	特步官方旗舰店	上海	tb-sh-4-sm-2.jpg	
19 192557	2	2	特步运动服	特步运动短裤男	0000000290	tb-wr-1.jpg	0221	1	特步官方旗舰店	上海	tb-wr-1-sm-2.jpg	
20 179887	2	2	特步运动服	【套装】特步运	0000000409	tb-wr-2.jpg	0131	1	特步官方旗舰店	上海	tb-wr-2-sm-2.jpg	
21 179904	2	2	特步运动服	【上衣+裤子2件	0000000500	tb-wr-3.jpg	0112	1	特步官方旗舰店	上海	tb-wr-3-sm-2.jpg	
22 110861	2	2	特步运动服	【特步881329(0000000359	tb-wr-4.jpg	0166	1	特步官方旗舰店	上海	tb-wr-4-sm-2.jpg	

这张表基本和之前定义的表差不多，就是进行了数据填充，也是本系统在服务器启动之后第一次访问数据库时调用查询的表。里面的数据是和品牌种类、商品种类、库存、商品描述、关键字等等相关的信息，根据该信息，我对数据库可以进行一系列前后端结合的操作。

5) 商品种类表

id	name
1	sportsEquip
2	sportsShoes
3	sportsWear

根据需求分析，我对系统需要的商品划分为 3 种，分别是运动器材类、运动鞋类、运动服装类，并分别为其标号。

6) 订单表

id	uid	time	price	status_
16		17 2020-06-27 04:41:49	2115.00	未处理
17		17 2020-06-27 04:46:18	839.00	未处理
18		17 2020-06-27 04:47:37	484.00	未处理
19		17 2020-06-27 04:47:40	484.00	未处理

该表是订单表，这张表是通过用户在购物车界面直接进行结算产生的数据表，以用户 id 作为外键来进行关联用户，主要字段有商品下单时间，每一笔订单总额以及该订单的处理状态。

7) 订单明细表

id	oid	gid	quantity
20		16	4 6
21		16	46 1
22		17	3 1
23		17	9 1
24		18	2 1

该表是在上一张表的基础上进行创建的，在上一张表数据存储之后，则进行该表的创建并进行数据存储，主要就是要该商品明细项里面的 oid 要和对应的订单表 id 相符合，gid 是对应商品明细项的 id，用以索引相应的商品。Quantity 则是该项商品的个数。

8) 用户表

id	username	password	email	phoneNo	realName	age	sex	address	registerDate
17	zdz	zdz199804	zhaodez1771707859	赵得泽		17	男	黑龙江省牡丹江市	2020-06-24
18	zzz	zdz1998	zhaodez1771707859	赵得泽		19	女	江苏省徐州市	2020-06-24

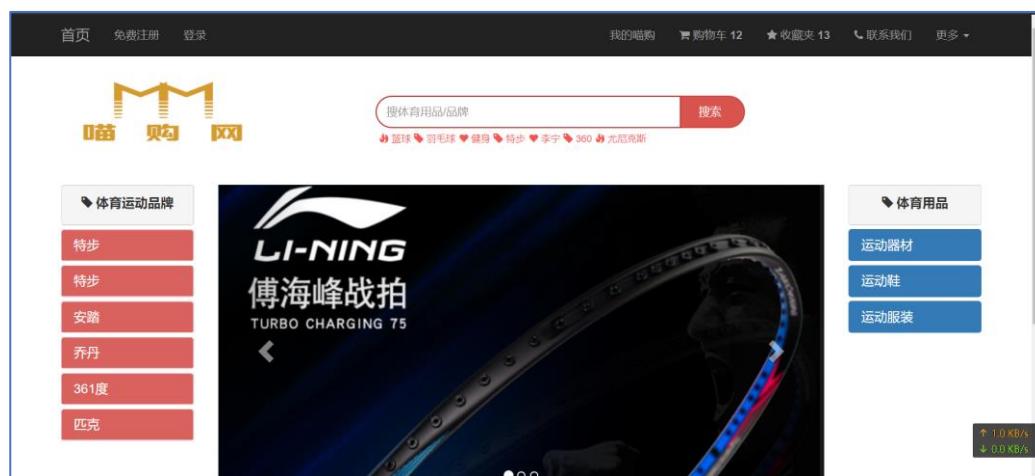
该表是用户注册信息表，用来存储用户的注册信息，用户在注册时会实时访问该数据库，查看是否有用户名已经与输入的用户名相同，若相同则进行错误提示，知道不相同，通过所有验证，才可注册成功。在每次登陆的时候也会访问这张表，用来和前端交互，判断是否登录成功。

7.3 前端实现

7.3.1 前端功能

在本次“体育运动装备订购管理系统”中，我实现的前端功能主要有：用户注册功能、用户登录功能、用户浏览数据库商品功能、用户在不同页面进行访问商品详情功能、用户在不同页面进行“添加购物车操作”，用户在购物车页面下单（结算）操作，生成订单功能，查看订单功能，查看订单明细功能，订单操作功能，还有几本的商品搜索功能。

7.3.2 实现效果



这是系统的首页，服务器启动之后便首先跳转到此页面，在此页面，用户可以进行一些相关数据操作，从中我们可以看出，在主页的主体部分，左右侧是对商品进行分类浏览的“手风琴”折叠菜单，用户可以点击每个菜单项对立面的商品分类进行仔细查看和分类浏览。在这个页面之下的部分，则是数据库中所有商品缩略图概览。具体如下。



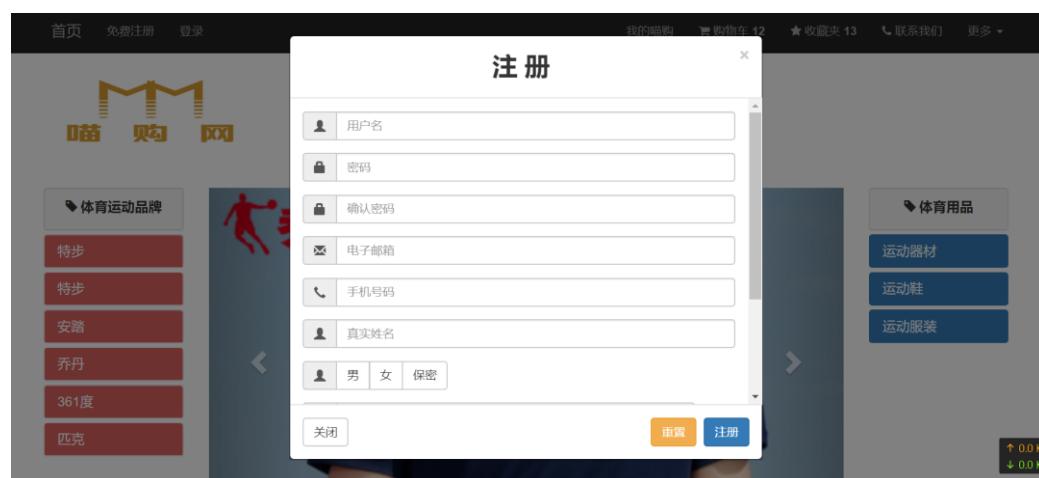
此处，用户便可以进行相关的商品操作具体就是立即购买，生成订单，加入购物车进行商品累计记录结算操作等等。

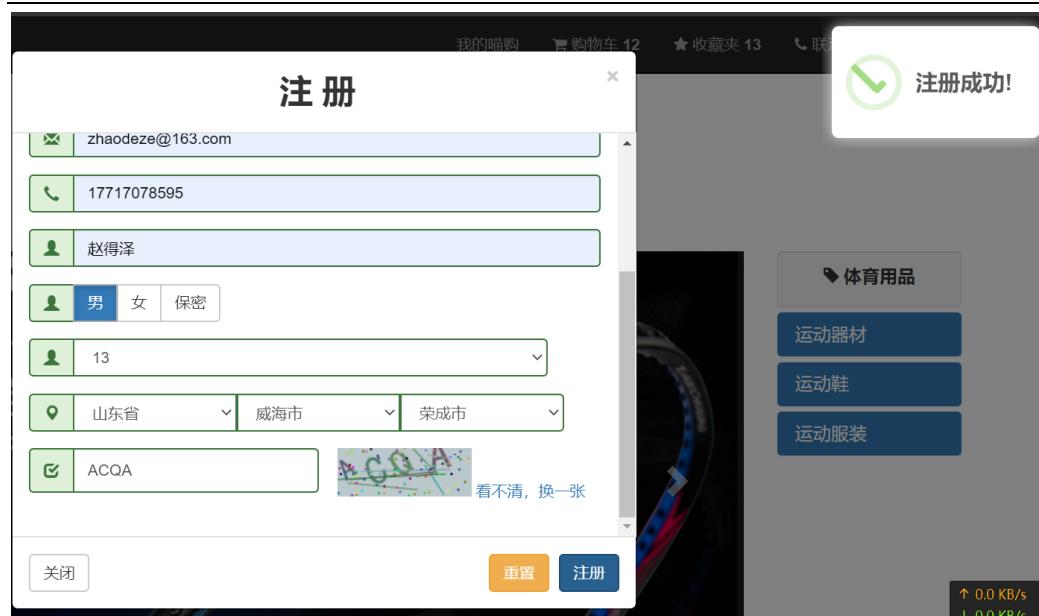
以上便是本系统的概貌，具体的功能将在下面的论述中一一解释、介绍。

7.3.3 功能模块

1) 用户注册功能







ID	用户名	密码	邮箱	手机号	真实姓名	年龄	性别	地址	注册日期
17	zdz	zdz199804	zhaodeze@163.com	17717078595	赵得泽	17	男	黑龙江省牡丹江市	2020-06-24
18	zzz	zdz1998	zhaodeze@163.com	17717078595	赵得泽	19	女	江苏省徐州市	2020-06-24
19	zdz22	123456	zhaodeze@163.com	17717078595	赵得泽	20	男	山东省聊城市	2020-06-27
20	zdz1212	zzzzzz	zhaodeze@163.com	17717078595	赵得泽	13	男	山东省威海市	2020-06-27

上述过程便是注册的全过程演示，在该过程中，从图片展示我们可以看出，用户可以通过点击主界面的注册按钮进行注册操作，注册过程中，对于“用户名”字段，会进行实时查重，看是否与数据库里面的用户名相同，要是相同则需要重新输入，直到填项变成绿色，则表示这一项的填写是符合“差错验证”的，我会在每一项的底下都给出差错提示，提高用户的可辨识度和交互性。在全部填写结束并验证通过之后，点击注册按钮，后台就会将数据插入数据库（如上图），并给出消息反馈。

2) 用户登录功能







上图便是登录功能的全部流程，在该流程里面，用户通过点击主页导航栏的“登录”按钮，并可弹出模态对话框进行登录的操作。同样，和注册一样，我同样会为用户的输入做容错性检查和验证，来判断用户的输入是否合法，在全部通过时候，选项框变绿色，点击登录按钮即可进行后台验证，是否和数据库里面的数据相匹配，若匹配，就提示登录成功的对话框，否则，提示登录失败的对话框。

需要注意的是，在验证用户输入不通过时，登录按钮是“非激活”状态，用户是不能点击的，这样一来大大提高了系统的安全性，防止用户随意点击带来的系统隐患。



登录成功之后，原先导航栏的“登录”按钮变成了红色的“欢迎提示”。

3) 用户不同页面添加购物车功能

① 主页



韦德推荐

春夏换季 满99减5 元 5号7号4.62 起

¥ 399.0 89人付款

李宁 LI-NING CBA联赛官方比赛篮球...

新品 包邮

加入购物车 **立即购买**



¥ 484.0 89人付款

李宁篮球韦德7号签名黑粉色吸湿PU耐...

新品 包邮

加入购物车 **立即购买**

首页 我的购物车 12 ★ 收藏夹 13 联系我们 更多 ▾

我的购物车
搜索购物车商品

<input type="checkbox"/> 全选	宝贝描述	单价	数量	小计	操作
<input type="checkbox"/>	 李宁 LI-NING CBA联赛官方比赛篮球室内外儿童成人7号PU篮球 443-1	¥ 399.0	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>	¥ 399.0	<input type="button" value="删除"/>
<input type="checkbox"/>	 李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>	¥ 484.0	<input type="button" value="删除"/>

已选商品 0 件 合计(不含运费): **¥ 0.00**

如图所示，在用户点击“加入购物车”按钮之后，系统之自动跳转到“cart.jsp”，也就是购物车页面，并将刚才的商品添加到购物车，显示出来。

② 商品详情页

This screenshot shows a product detail page for a Li-Ning jump rope. At the top, there's a large image of a woman holding the jump rope. Below it is a smaller image showing the jump rope and its packaging. To the right of the images, product information is displayed: "李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童 成人健身运动健身器材套装计数跳绳 780", "选购商品 李宁运动器材", "商品编号 149447", "销售价格 ￥271.0", "库存数量 194", and "选购数量" with a value of 1. A red arrow points to the "加入购物车" (Add to Cart) button. On the far right, there's a sidebar titled "浏览排行榜" (Browsing Ranking) showing three items from Adidas.

This screenshot shows the user's shopping cart. It lists three items: a basketball, another basketball, and the same jump rope from the previous screen. Each item has a checkbox next to its thumbnail, a quantity selector (set to 2 for the first two, 1 for the jump rope), and a "删除" (Delete) button. The total count is 0 items and the total price is 0.00. A red box highlights the jump rope row.

宝贝描述	单价	数量	小计	操作
李宁 LI-NING CBA联赛官方比赛篮球室内外儿童成人7号PU蓝球 443-1	¥ 399.0	- 2 +	¥ 798.0	<button>删除</button>
李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	- 2 +	¥ 968.0	<button>删除</button>
李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	- 1 +	¥ 271.0	<button>删除</button>

在该页面点击“加入购物车”按钮，同样可以实现购物车添加功能。

4) 删除购物车商品及小计功能

This screenshot shows the user's shopping cart with multiple items selected for deletion. The checkboxes for all three items (the basketballs and the jump rope) are checked. The total count is 12 items and the total price is 4743.00. A red box highlights the "删除" (Delete) button for the jump rope row.

宝贝描述	单价	数量	小计	操作
李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	- 7 +	¥ 3388.00	<button>删除</button>
李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	- 5 +	¥ 1355.00	<button>删除</button>

我的购物车

		宝贝描述	单价	数量	小计	操作
<input checked="" type="checkbox"/>	李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	- 7 +	¥ 3388.00	删除	
<input checked="" type="checkbox"/>	李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	- 5 +	¥ 1355.00	删除	

清空购物车 已选商品 12 件 合计(不含运费): ￥4743.00 **结算**

我的购物车

		宝贝描述	单价	数量	小计	操作
<input type="checkbox"/>	李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	- 1 +	¥ 271.0	删除	

清空购物车 已选商品 0 件 合计(不含运费): ￥0.00 **结算**

从上述操作中，我们可以看出，再点击全选或者某个商品前面的复选框时，会在下面的统计栏进行汇总金额，对选中商品的金额进行小计，显示给用户。再点击每个商品最右侧的删除按钮时，会将该商品条目从购物车移除，达到删除目的。

5) 购物车结算及添加订单功能

我的购物车

		宝贝描述	单价	数量	小计	操作
<input checked="" type="checkbox"/>	李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	- 2 +	¥ 968.0	删除	
<input checked="" type="checkbox"/>	李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	- 3 +	¥ 813.0	删除	
<input checked="" type="checkbox"/>	李宁官网男子短袖运动套装训练系列轻便透气T恤短裤吸湿排汗跑步健身套装ADTP001 标准黑/标准黑 L	¥ 360.0	- 1 +	¥ 360.0	删除	

清空购物车 已选商品 6 件 合计(不含运费): ￥2141.00 **结算**

This screenshot shows the 'My Shopping Cart' section of a web application. At the top, there are navigation links: '首页', '我的购物车 12', '收藏夹 13', '联系我们', and '更多'. Below this is a red header bar with the text '我的购物车' and a search bar. A table header row includes columns for '全选', '宝贝描述', '单价', '数量', '小计', and '操作'. In the center, a message in a red-bordered box says '您的购物车是空的哦...去[主页](#)逛逛吧。'. At the bottom, there are buttons for '清空购物车', '已选商品 0 件', '合计(不含运费): ￥0.00', and a yellow '结算' button.

This screenshot shows the 'My Orders' section. The top navigation is identical to the shopping cart page. The main content area has a red header '我的订单' and a table with columns: '全选', '下单时间', '实付款', '交易状态', and '交易操作'. One row in the table is highlighted with a red arrow pointing to the '下单时间' column, which shows '2020-06-27 09:17:13'. Another red arrow points to the '实付款' column, which shows '¥ 2141.0'. To the right of the table is a '交易状态' dropdown menu with options: '未处理' (selected), '待审核', '待发货', '待收货', and '已确认收货'. At the bottom of the page, there is a '删除所有订单' button and a '合计(不含运费): ￥0.00' summary. The Windows taskbar at the bottom shows the date as '2020/6/27' and the time as '9:17'.

从上述操作中我们可以看出，当在购物车中点击“结算”按钮，即可将此时购物车中的商品生成订单，当跳转到“我的订单”页面时就会显示该订单的生成时间，付款金额，交易状态等。在数据库中查对应的表项，即可检验订单是否生成成功。

A screenshot of a database table with columns: id, uid, time, price, and status_. The table shows one row with id 25, uid 17, time '2020-06-27 09:17:13', price '2141.00', and status_ '未处理'. A red arrow points to the 'status_' column.

id	uid	time	price	status_
25	17	2020-06-27 09:17:13	2141.00	未处理

6) 查看订单明细功能

This screenshot shows the 'My Orders' section of a web application. It displays a single order from June 27, 2020, at 09:17:13. The total amount is 2141.0. The order status is '未处理' (Unprocessed), with options to '查看物流' (Check Logistics) or '待确认收货' (Pending Confirmation of Receipt). A red button at the bottom left says '删除所有订单' (Delete All Orders), and a grey bar at the bottom right shows a total of '合计(不含运费): ¥ 0.00'.

This screenshot shows the 'Order Details' page for the same order. It lists three items:

编号	宝贝描述	单价	数量	小计(元)
101951	李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	2	¥ 968.0
149447	李宁LBDM780 李宁 LI-NING 计数跳绳 中考专用学生 儿童成人健身运动健身器材套装计数跳绳 780	¥ 271.0	3	¥ 813.0
145691	李宁官网男子短袖运动套装训练系列轻便透气T恤短裤吸湿排汗跑步健身套装ADTP001 标准黑/标准黑 L	¥ 360.0	1	¥ 360.0

This screenshot shows the MySQL database interface with the 'order_detail' table selected. The table has four columns: id, oid, gid, and quantity. The data shows three rows with values (34, 25, 2, 2), (35, 25, 6, 3), and (36, 25, 11, 1). A red arrow points from the 'order_detail' table in the database sidebar to the 'quantity' column of the third row in the table.

	id	oid	gid	quantity
▶	34	25	2	2
	35	25	6	3
	36	25	11	1

从订单明细表中可以看出，当点击“订单详情”按钮的时候，会跳转到“orderdetail.jsp”页面，也就是订单详情页面，显示该笔订单里面的所有商品明细。最后，经与数据库中“order_detail”表中的数据对照，发现结果符合预期。

7) 删除订单操作



		id	uid	time	price	status_
		(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

		id	oid	gid	quantity
		(N/A)	(N/A)	(N/A)	(N/A)

当点击订单最右侧的“交易操作”中的“删除订单”按钮，便会将该订单从“我的订单”中删除，从上图中我们可以看出，当删除某个订单的时候，它对应的明细也会被删除，这是因为在订单明细中有订单的外键做索引，所以会触发级联删除，

将该订单的所有订单明细级联删除。经查表发现，操作符合预期。

8) 不同页面查看商品详情功能

① 主页



② 购物车页面

宝贝描述	单价	数量	小计	操作
李宁篮球韦德7号签名黑粉色吸湿PU耐磨水泥地成人小学生正品蓝球	¥ 484.0	- +	¥ 484.0	删除

③ 订单详情页面

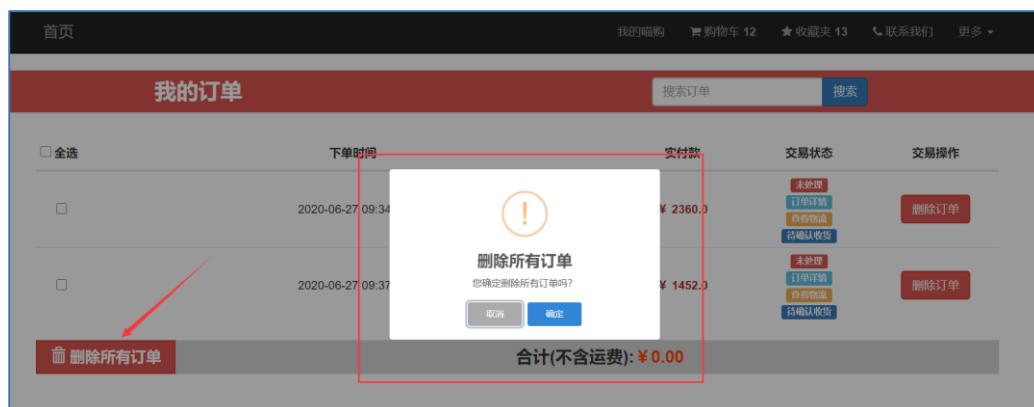
编号	宝贝描述	单价	数量	小计(元)
183619	李宁AE15李宁 LI-NING 羽毛球 鹅毛球 比赛 耐打飞行稳定12只装 AE15	¥ 356.0	4	¥ 1424.0
197655	【华晨宇心选】李宁官网运动裤2020新品女子收口运动裤休闲裤运动时尚系列女装AYKP204 标准 黑. 跑粉色	¥ 468.0	2	¥ 936.0

查看订单的详情可以从以上 3 个页面点击图片链接或者文字描述链接进行图片详情的查看。

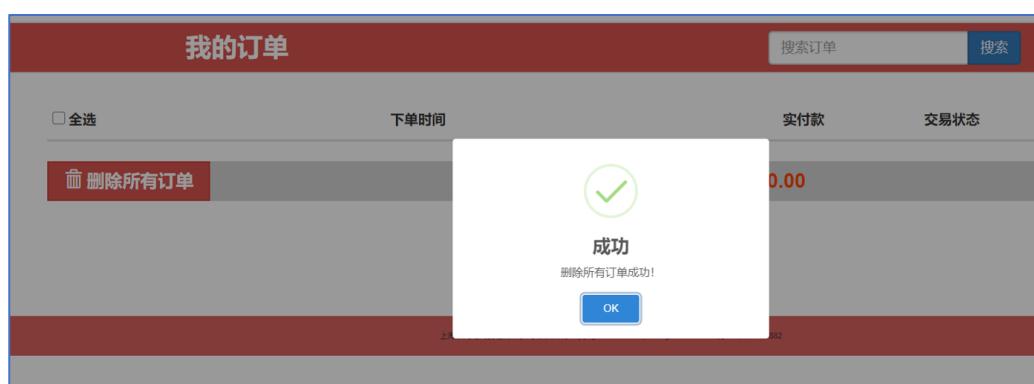
9) 容错提示功能



当购物车是空的时候，若点击“清空购物车”则会进行警告提示；



当订单中点击“删除所有订单”的时候，给出模态对话框警告提示，提示小心操作。



并给出“成功”操作提示。



订单页面中没有订单时，点击“删除所有订单”按钮，会给出相应的警告提示。



在主页没有登录的状态下，若是点击“购物车”，则会给出“请先登录”的提示操作。

登录成功、注册成功、登录失败等情况已经在前面展示，都会给出相应的提示操作，指导用户正确使用该系统。

7.3.4 关键数据结构

1. Java Bean(对应数据表结构中的字段)

1) 品牌种类 Bean

```
public class BrandType {
    private int id;
    private String name;
```

2) 购物车商品项 Bean

```
public class CartItem {
    Goods goods;
    private Integer quantity;
```

3) 商品 Bean

```
public class Goods implements Serializable {  
    |  
    private Integer id;//编号  
    private String code;//商品编码  
    private Integer gid;//商品类型  
    private Integer bid;//品牌  
    private String name;//商品名  
    private String description;//简介  
    private Double price;//单价  
    private String picture;//商品大图  
    private Integer inventory;//库存  
    private Integer status;//处理状态  
    private String shop;//销售店铺  
    private String location;//销售地点  
    private String img2;//第一张小图  
    private String img3;//第二张小图  
    private String img4;//第三张小图
```

4) 商品种类 Bean

```
public class GoodsType {  
    private int id;  
    private String name;
```

5) 订单 Bean

```
public class Order implements Serializable {  
    private Integer id;//订单编号  
    private Integer uid;//用户信息 (匹配用户id)  
    private String time;//下单时间  
    private Double price;//订单总价  
    private String status;//订单状态  
  
    //一个订单信息表可能含有多个订单，每个订单有各自的明细，  
    private List<OrderDetail> orderDetailList;  
    //所以用list来表示订单明细 (一对多)  
  
    //一个订单只属于一个用户  
    private User user;
```

6) 订单明细 Bean

```
public class OrderDetail implements Serializable {  
    private Integer id;//订单明细项编号  
    private Integer oid;//订单信息(匹配订单信息id)  
    private Integer gid;//商品信息(匹配商品id)  
    private Integer quantity;//商品数量  
  
    //一个订单明细项只能包含一个商品  
    private Goods goods;  
  
    //一个订单明细项只能属于唯一的订单编号  
    private Order order;
```

7) 订单明细项 Bean

```
public class OrderDetailItem {  
    private Integer gid;  
    private String code;  
    private String picture;  
    private String description;  
    private Double price;  
    private Integer quantity;
```

8) 用户 Bean

```
public class User implements Serializable {  
    private Integer id;  
    private String username;  
    private String password;  
    private String email;  
    private String phoneNo;  
    private String realName;  
    private String age;  
    private String sex;  
    private String address;  
    private Date registerDate;
```

上述 Bean 都是和表结构一一对应的 Java Bean。

7.3.5 关键技术

1. 数据库操作 DAO

1) UserDao

```
@Repository("userDao")
public interface UserDao {

    @Insert("insert into user(username,password,email,phoneNo,realName,age,sex,address,registerDate) " +
        "values(#{username},#{password},#{email},#{phoneNo},#{realName},#{age},#{sex},#{address},#{registerDate})")
    public void addUser(User user);

    @Select("select * from user where username like #{username} and password = #{password}")
    public List<User> findUsers(User user);

    @Select("select * from user where username like #{username}")
    public List<User> findUser(String username);
}
```

addUser:向数据库中添加用户，用于注册；

findUsers: 向数据库表中查询用户名和密码相同的用户，用于登录；

findUser:查询用户名相同的用户，用于注册时检验用户名是否已经注册；

2) GoodsDao

```
@Repository("goodsDao")
public interface GoodsDao {

    /**
     * 得到所有商品列表
     *
     * @return
     */
    @Select("select *from goods")
    public List<Goods> findAll();

    /**
     * 根据商品ID查询商品信息
     *
     * @param gid
     * @return
     */
    @Select("select * from goods where id = #{gid}")
    public Goods findGoodsById(int gid);
}
```

findAll:查询所有商品，用于商品展示；

findGoodsById:通过商品 id 查询该商品信息；

3) OrderDao

```

@Repository("orderDao")
public interface OrderDao {

    @Select("select * from order_ where uid = #{uid}")
    public List<Order> findOrderByUid(int uid);

    @Select("select * from order_ where id in (select max(id) from order_)")
    public Order findLastOrder();

    @Update("delete from order_ where id=#{oid}")
    public void deleteOrderById(int oid);

    @Update("insert into order_(uid,time,price,status_) values(#{uid},#{time},#{price},#{status})")
    public void addOrder(Order order);
}

```

`findOrderByUid`: 查询指定用户的所有订单，用于在订单页面展示；

`findLastOrder` :查询最后一次添加进订单表的订单项，用于和商品明细中的 oid 对应；

`deleteOrderById`: 通过订单 id 删除订单，主要用于在订单页面删除订单；

`addOrder`: 在数据库表中插入订单；

4) OrderDetailDao

```

@Repository("orderDetailDao")
public interface OrderDetailDao {
    /**
     * 由订单信息表的id得到订单的详细信息
     *
     * @param oid 订单明细项编号
     * @return 订单明细集合
     */
    @Select("select * from order_detail where oid=#{oid}")
    public List<OrderDetail> findOrderDetailById(int oid);

    /**
     * 查看订单明细
     *
     * @param orderDetail 订单明细项
     */
    @Update("insert into order_detail(oid,gid,quantity) values(#{oid},#{gid},#{quantity})")
    public void addOrderDetailItem(OrderDetail orderDetail);
}

```

5) GoodsTypeDao

```
@Repository("goodsTypeDao")
public interface GoodsTypeDao {

    /**
     * 得到所有商品种类
     *
     * @return
     */
    @Select("select * from goods_type")
    public List<GoodsType> findAll();
}
```

findAll:得到所有商品类型

6) BrandTypeDao

```
@Repository("brandTypeDao")
public interface BrandTypeDao {
    /**
     * 得到所有的品牌类型
     *
     * @return
     */
    @Select("select * from brand_type")
    public List<BrandType> findAll();
}
```

findAll: 查询所有品牌类型;

以上的 Dao 没有实现类，只需要通过 MyBatis 注解即可实现，在内部提供了实体类的封装，因此无需考虑得到结果集合后的对象封装情况。

2. 视图层操作 Controller

1) UserController

```

@RequestMapping("/login")
public @ResponseBody
User login(@RequestBody User user, HttpServletResponse response) {
    User loginUser = userService.login(user);
    System.out.println(loginUser);
    if (loginUser != null) {
        Cookie userCookie = new Cookie("name", "LoginUser", loginUser.getUsername());
        //将登录用户名存储在cookie中
        userCookie.setPath("/");
        response.addCookie(userCookie);
        return loginUser;
    } else {//继续输入
        return null;
    }
}

```

Login: 登录，用户在时是通过 ajax 请求来进行的，因此不能将登录用户的对象信息存进 Session 或 Request 域，此处将用户名存进 cookie，在前台进行用户名抽取，还可用作数据库按照 username 查找记录，可以有用处。

```

String checkExists(@RequestParam String username) {
    System.out.println(username);
    //requestBody requestParam 此处不同 前者传过来的是对象 后者传过来的是json的值
    List<User> userList = userService.checkExists(username);
    Map<String, Boolean> registerMsg = new HashMap<>();
    ObjectMapper mapper = new ObjectMapper();
    String jsonData = null;
    if (userList.size() == 1) {//用户已存在
        registerMsg.put("valid", false);
        try {
            jsonData = mapper.writeValueAsString(registerMsg); //将map转为json返回ajax
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    } else {//用户不存在
        registerMsg.put("valid", true);
        try {
            jsonData = mapper.writeValueAsString(registerMsg); //将map转为json返回ajax
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    }
    System.out.println(jsonData);
    return jsonData;
}

```

CheckExists: 此处是检查数据库表中是否有 username 与注册时输入的相同，为了提供实时反馈，使用 ajax 请求的方式传数据，将 username 作为请求体传入即可，之后还须注意的就是，在给 ajax 返回数据时，需要返回 json 格式的数据，因此需

要 objectmapper，将 map 数据转换为 json，在返回给前端页面即可，否则前端接收不到。

```
@RequestMapping("/logout")
public void logout(HttpServletRequest request, HttpServletResponse response) throws IOException {
    Cookie cookies[] = request.getCookies();

    for (Cookie cookie : cookies) {
        cookie.setMaxAge(0);
        cookie.setPath("/");
        response.addCookie(cookie);
    }
    request.getSession().invalidate(); //清除会话
    response.sendRedirect(request.getContextPath() + "/index.jsp");
}
```

Logout: 登出，此处要注意销毁 cookie，删除 session 中的所有数据。

2) RegisterController

```
@RequestMapping("/register")
public @ResponseBody
String register(@ModelAttribute("registerUser") User user) {
    System.out.println(user);
    boolean registerUser = userService.register(user);
    System.out.println(user);
    Map<String, String> registerMsg = new HashMap<>();
    registerMsg.put("registerMsg", "注册成功！");
    ObjectMapper mapper = new ObjectMapper();
    String jsonData = null;
    try {
        jsonData = mapper.writeValueAsString(registerMsg); //将map转为json返回ajax
    } catch (JsonProcessingException e) {
        e.printStackTrace();
    }
    System.out.println(jsonData);
    return jsonData;
}
```

Register: 注册，此处也是前端传来的 ajax 请求，里面携带 User 对象，是在执行该方法之前预先处理过的，再插入数据库之后，返回执行成功的信息，通过 objectmapper 转为 json 数据，返回给 ajax 请求，作为响应体。

```

    /**
     * /注册前先执行 该函数必须单独和其关联函数一个controller\
     * 否则会造成其他函数执行时的错误
     * 此处只是先为用户注册添加日期信息
     * ajax请求传递参数
     *
     * @param registerUser
     * @param map
     */

```

```

    @ModelAttribute
    public void register_(@RequestBody User registerUser, Map<String, User> map) {
        System.out.println(registerUser);
        registerUser.setRegisterDate(new Date());
        map.put("registerUser", registerUser); //将user存进map 在register中取出来
    }

```

Register_: 注册前预处理，由于该函数有 `ModelAttribute` 注解，所以在该 Controller 里面的方法执行之前都必须执行该方法，此处要慎重使用，应该尽量使其和关联它的方法处于同一个 Controller，别的不关联的写在别的 Controller。此处是需要处理注册日期，因此先对 ajax 中的请求体 User 对象预处理，使其 `registerDate` 字段有值，然后插入数据库。

3) OrderController

```

public void addOrder(String username, HttpServletRequest request, HttpServletResponse response) throws IOException {
    System.out.println(username);
    Order order = new Order();
    //取得当前系统时间
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    order.setTime(simpleDateFormat.format(new Date()));
    List<User> userList = userService.checkExists(username); //其实为1
    order.setUser(userList.get(0));
    order.setUserId(userList.get(0).getId());
    order.setPrice((Double) request.getSession().getAttribute("totalPrice"));
    order.setStatus("未处理");
    orderService.addOrder(order);
}

```

```
Map shopcart = (Map) request.getSession().getAttribute("shopcart");//获得购物车
Iterator it = shopcart.keySet().iterator();
Order order1 = orderService.findLastOrder();
System.out.println(order1.getId());
try {
    while (it.hasNext()) {
        Object key = it.next();
        CartItem cartItem = (CartItem) shopcart.get(key);
        //设置订单明细
        OrderDetail orderDetail = new OrderDetail();
        orderDetail.setGoods(cartItem.getGoods());
        orderDetail.setGid(orderDetail.getGoods().getId());
        orderDetail.setOrder(order1);
        orderDetail.setOid(order1.getId());
        orderDetail.setQuantity(cartItem.getQuantity());
        //添加订单明细项
        orderDetailService.addOrderDetailItem(orderDetail);
    }
} catch (Exception e) {
    e.printStackTrace();
}
//已经生成订单，所以删除购物车
```

addOrder: 添加订单，向数据库 order_ 表中填加一个订单项；

addOrderDetailItem: 向数据库表 order_detail 中添加一个订单明细项；

此处因为生成了订单，因此需要将 Session 中的 shopcart 项删除，即清空购物车。

4) CartController

```

@RequestMapping("/addToCart")
public String addToCart(Integer gid, HttpServletRequest request, Model model) {
    System.out.println(gid);
    Map shopcart = (Map) request.getSession().getAttribute("shopcart");
    Goods goods = goodsService.findGoodsById(gid); //通过商品id 获取商品
    System.out.println(goods);
    if (shopcart == null) { //如果购物车不存在，则创建购物，并存入到session中
        shopcart = new HashMap();
        model.addAttribute("shopcart", shopcart);
    }
    //如果存在购物车则判断要添加的商品是否已在购物车
    CartItem cartItem = (CartItem) shopcart.get(goods.getId());
    cartItem.setGoods(goods);
    System.out.println(cartItem);
    if (cartItem != null) { //商品已在购物车，则更新其数量即可
        cartItem.setQuantity(cartItem.getQuantity() + 1);
    } else { //创建一个商品项进入购物车
        CartItem newCartItem = new CartItem(goods, quantity: 1);
        shopcart.put(goods.getId(), newCartItem);
    }
    System.out.println(shopcart);
}

```

addTocart: 添加到购物车，此处是利用了 Session 域对象，将 shopcart(HashMap)直接放在其中即可，前端页面利用 EL 表达式取出。

```

@RequestMapping("/updateGoodsQuantity")
public String updateGoodsQuantity(Integer gid, Integer quantity, HttpServletRequest request) {
    Map shopcart = (Map) request.getSession().getAttribute("shopcart");
    Goods goods = goodsService.findGoodsById(gid); //通过商品id 获取商品
    CartItem cartItem = (CartItem) shopcart.get(goods.getId());
    cartItem.setQuantity(quantity);
    return "cart";
}

@RequestMapping("/deleteCartItem")
public String deleteCartItem(Integer gid, HttpServletRequest request) {
    Map shopcart = (Map) request.getSession().getAttribute("shopcart");
    shopcart.remove(gid);
    return "cart";
}

@RequestMapping("/clearShopcart")
public String clearShopcart(HttpServletRequest request) {
    Map shopcart = (Map) request.getSession().getAttribute("shopcart");
    shopcart.clear();
    return "cart";
}

```

updateGoodsQuantity: 更新商品数量，它是在 cart.jsp 中进行了修改商品数量的操作触发的，需要从 session 域中得到 shopcart 属性，然后得到购物车商品项 CartItem，再取出 Goods，修改数量即可。

deleteCartItem: 删除购物车商品项，它是在点击 cart.jsp 页面的商品项右侧的删除按钮触发的，需要获得 session 域中的 shopcart 属性然后移除 gid (key) 即可；

clearShopcart: 清空购物车，在 cart.jsp 页面点击“清空购物车”按钮触发，同样需要获得 session 域中的 shopcart 属性，直接 clear() 即可。

8 心得体会

本次课程设计对我来说，是一个比较大的考验，同时也是一次历练，通过本次课设的设计，我对 Java 这门语言有了全新的认识，因为一开始有 C/C++ 语言的基础，从 Java SE 学起来还算比较顺手，并且能够用心去体会两种语言之间的微妙的关系，几乎 Java 零基础的我，边看视频，边进行实践操作，对每天看的内容有基础的巩固。这样可以预防遗忘。在“速成”JavaSE (基础部分) 之后，就开始转向了真正与本次项目相关的 JavaWeb 学习，在大致了解其套路之后，便开始根据已有的经验，找到与本项目相关的知识内容着重展开学习，对于其中一些简单的知识就直接跳过。同时也必须伴有实践环节，通过练习不断的巩固。

在经过为其 10 天左右的 Javaweb 学习之后，便又开始转向了关键环节 SSM 框架的学习，这块的内容虽然没有前两者那么多，但大多是以前面的知识为基础，底层实现方式还是依据前面学过的知识，所以前面知识的透彻理解十分重要，在学习 SSM 框架的时候，真实感受到了框架的强大和便捷之处，可以帮我们省去好多工作。在学习单个框架的过程中，我先学的是 MyBatis，接下来分别是 Spring, SpringMVC，后者是从前者中分离出来的，控制视图层，在学习过程中过，我不断领会老师讲的内容，以及其中的关键之处，对配置文件.xml 的作用有了详细的了解，而且其中的逻辑联实际和不用框架时是一致的，只是框架都帮我们做了。

最后便是三个框架的整合，对于整合，是对三个框架的综合应用，又提升了一个层次，理解又需加深一步。但是随着时间推移，前面学习的框架有些知识已经忘了，所以定期的复习也是有必要的，这个项目必然是个持久战。

在学习完成之后，便开始了漫长的项目开发之路，装软件，配环境，问题一波未平一波又起，过程跌宕起伏，但好在自己坚持下来都克服了这些问题，查阅资料，回顾知识，总算有一个比较完满的结果。本次开发中，我对 java 中的某些概念更加明晰，对 MVC 前后端分离设计的内涵也有了掌握和深度理解。尤其是 EL 表达式，JSTL 表达式以及 Session, Request 等 jsp 中的知识有了详细的了解和掌握。对 SSM 框架的学习应用实践也颇有感触和收获。

9 参考文献

- [1] AbrahamSilberschatz, Korth H , Sudarshan S , et al,数据库系统概念[M],高等教育出版社, 2006.
- [2]李昂,体育用品管理系统的应用与实现,工程硕士学位论文,2015.
- [3]萨师煊,数据库系统概论 (第三版) [M],高等教育出版社, 2000.
- [4]黑马程序员 JavaSE,JavaWeb 视频教程
- [5]SSM 框架源码