**存档日期:2018.05.25 存档编号:**





JIANGSU NORMAL UNIVERSITY

**本科毕业设计**

UNDERGRADUATE DESIGN

**设 计 题 目：**文具销售管理系统的设计与实现

**姓 名：**

**学 院：** 智慧教育学院

**专 业：**  计算机科学与技术（S）

**年 级 、 学 号：**

**指 导 教 师：** 孟倩

**江苏师范大学教务处印制**

**设计原创性声明**

本人郑重声明：所呈交的毕业设计，是在导师的指导下，独立进行研究所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本设计的研究成果不包含他人享有著作权的内容。对本设计所涉及的研究工作做出贡献的个人和集体，均已在设计中以明确的方式标明。本设计的知识产权归属培养单位。

**设计版权使用授权书**

本设计“文具销售管理系统的设计与实现”是本人在校期间所完成学业的组成部分，是在江苏师范大学教师的指导下完成的，因此，本人特授权江苏师范大学可将本毕业论文的全部或部分内容编入有关书籍、数据库保存，可采用复制、印刷、网页制作等方式将论文文本和经过编辑、批注等处理的论文文本提供给读者查阅、参考，可向有关学术部门和国家有关部门或机构呈送复印件和电子文档。本毕业论文无论做何种处理，必须尊重本人的著作权，署明本人姓名。

**文具销售管理系统的设计与实现**

**摘要**

计算机技术的飞快发展给人们的生活带来了很大的方便，传统的文具店是 采用人工管理方式，工作量大效率低。因此，开发出一套有效文具销售管理系统是有必要的，这样可以节省成本花费，提高工作效率，使商家和顾客完成双赢。

本系统以MyEclipse为前端开发工具,MySQL为后台数据库开发以及应用服务器Tomcat，此系统有两类使用者，一个是顾客，另一个是文具店管理员，两个不同的使用者，设置不同的权限，赋予不同的功能，因此本系统具有一定的稳定性和安全性；设计的界面清晰简单易懂，使用者不需要计算机知识也能够很好的使用。

该论文有图46幅，表8个，参考文献24篇。

关键词：文具销售管理 MyEclipse，MySQL，Tomcat

**Design and Implementation of Stationery Sales Management System**

**Abstract**

The rapid development of computer technology has brought great convenience to people's lives. The traditional stationery shop adopts manual management methods, and the workload is inefficient. Therefore, it is necessary to develop an effective stationery sales management system. This can save costs and increase work efficiency. Therefore, businesses and customers will achieve a win-win situation.

The system uses MyEclipse as a front-end development tool, MySQL as a back-end database development and application server Tomcat. This system has two types of users, one is a customer and the other is a stationery shop administrator. Two types of users set different permissions to give different Functionality. It has a certain degree of stability and security; design interface is simple and easy to understand, users can use without computer knowledge.

The paper has 46 pictures, 8 tables and 24 references.

**Key Words:**Stationery Sales ManagementMyEclipseMySQLTomcat

**目录**

[摘要 I](#_Toc514105140)

[Abstract II](#_Toc514105141)

[目录 III](#_Toc514105142)

[图清单 V](#_Toc514105143)

[表清单 VII](#_Toc514105144)

[1 绪论 1](#_Toc514105145)

[1.1课题意义 1](#_Toc514105146)

[1.2国内外研究现状 1](#_Toc514105147)

[1.3主要工作 1](#_Toc514105148)

[1.4本文研究内容及主要贡献 1](#_Toc514105149)

[2 系统运行环境 3](#_Toc514105150)

[2.1 JSP 3](#_Toc514105151)

[2.2 MySQL 3](#_Toc514105152)

[2.3 面向对象编程思想 4](#_Toc514105153)

[2.4 本章小结 4](#_Toc514105154)

[3 系统分析 5](#_Toc514105155)

[3.1 可行性分析 5](#_Toc514105156)

[3.2 需求分析 5](#_Toc514105157)

[3.3 本章小结 9](#_Toc514105158)

[4 系统设计 11](#_Toc514105159)

[4.1 系统功能设计 11](#_Toc514105160)

[4.2 数据库概念模型的设计 11](#_Toc514105161)

[4.3数据库逻辑结构的设计 14](#_Toc514105162)

[4.4本章小结 15](#_Toc514105163)

[5 系统实现及效果分析 17](#_Toc514105164)

[5.1 前台部分设计与实现 17](#_Toc514105165)

[5.2 后台部分设计与实现 25](#_Toc514105166)

[5.3 本章小结 31](#_Toc514105167)

[6 总结与展望 32](#_Toc514105168)

[6.1总结 32](#_Toc514105169)

[6.2展望 32](#_Toc514105170)

[毕业设计体会 33](#_Toc514105171)

[参考文献 34](#_Toc514105172)

[致谢 35](#_Toc514105173)

**图清单**

|  |  |  |
| --- | --- | --- |
| 图序号 | 图名称 | 页码 |
| 图3-1 | 用户系统流程图 | 6 |
| 图3-2 | 管理员系统流程图 | 7 |
| 图3-3 | 顶层数据流图 | 7 |
| 图3-4 | 一级数据流图 | 8 |
| 图3-5 | 二级数据流图 | 8 |
| 图3-6 | 数据字典 | 9 |
| 图4-1 | 前台功能模块图 | 11 |
| 图4-2 | 后台功能模块图 | 11 |
| 图4-3 | 用户实体图 | 12 |
| 图4-4 | 文具分类实体图 | 12 |
| 图4-5 | 文具实体图 | 12 |
| 图4-6 | 购物车实体图 | 12 |
| 图4-7 | 订单实体图 | 13 |
| 图4-8 | 管理员实体图 | 13 |
| 图4-9 | E-R图 | 13 |
| 图5-1 | 前台首页图 | 17 |
| 图5-2 | 用户注册图 | 17 |
| 图5-3 | 用户注册错误提示一图 | 18 |
| 图5-4 | 用户注册错误提示二图 | 18 |
| 图5-5 | 用户登录图 | 19 |
| 图5-6 | 用户登录错误提示图 | 19 |
| 图5-7 | 修改密码图 | 20 |
| 图5-8 | 文具列表图 | 20 |
| 图5-9 | 搜索文具图 | 21 |
| 图5-10 | 文具详细图 | 21 |
| 图5-11 | 购物车图 | 22 |
| 图5-12 | 订单准备图 | 22 |
| 图5-13 | 下单成功图 | 23 |
| 图5-14 | 支付页面图 | 23 |
| 图5-15 | 支付成功图 | 24 |
| 图5-16 | 订单列表图 | 24 |
| 图5-17 | 订单详细查看图 | 24 |
| 图5-18 | 订单详细取消图 | 25 |
| 图5-19 | 订单详细确认收货图 | 25 |
| 图5-20 | 管理员登录图 | 25 |
| 图5-21 | 后台主页图 | 26 |
| 图5-22 | 分类列表图 | 26 |
| 图5-23 | 添加1级分类图 | 27 |

**图清单（续）**

|  |  |  |
| --- | --- | --- |
| 图5-24 | 添加2级分类图 | 27 |
| 图5-25 | 修改1级分类图 | 28 |
| 图5-26 | 修改2级分类图 | 28 |
| 图5-27 | 文具列表图 | 29 |
| 图5-28 | 添加文具图 | 29 |
| 图5-29 | 编辑或删除图 | 30 |
| 图5-30 | 订单列表图 | 30 |
| 图5-31 | 订单详细页面图 | 31 |

**表清单**

|  |  |  |
| --- | --- | --- |
| 表序号 | 表名称 | 页码 |
| 表4-1 | goods数据库中的数据表及其相应功能 | 14 |
| 表4-2 | t\_user表 | 14 |
| 表4-3 | t\_category表 | 14 |
| 表4-4 | t\_stationery表 | 14 |
| 表4-5 | t\_cartitem表 | 15 |
| 表4-6 | t\_order表 | 15 |
| 表4-7 | t\_orderitem表 | 15 |
| 表4-8 | t\_admin表 | 15 |

|  |  |
| --- | --- |
|  |  |

**1 绪论**

1.1课题意义

在今天，每个行业的竞争非常激烈，每个企业现在最渴望的是可以提高效率质量。大家在平时的学习生活中，都能体会得到网络的发展，人们愈来愈倾向于去网上购物。

之前大家买文具都需要跑到实体文具店挑选，而现在出现的文具销售管理系统可以让顾客坐在自己家的电脑前输入想要买的文具，快速而准确的找到自己想要的文具，大大加快了查询速度。

该系统可以全天都可营业交易。用户能随时需要随时购买。文具销售管理系统的设计开发对全球市场有很大的好处，市场的效率提高了，对未来的发展有着一定的实际意义。

1.2国内外研究现状

信息社会的发展是由于国际互联网为其创造了条件，网上文具销售的发展在信息化程度较高的国家地区非常迅猛。上网人数队伍越来越多，使得在网上卖文具的市场空间有了非常大的发展，文具特征很容易识别和描述，因此这种商品类型是很理想的。网上卖文具这个是信息社会发展的必然要求，只要网络发展没有停止，网上卖文具的发展就不会停止。

在线购买成为现实是由于网络技术的应用遍及，目前已有很多文具销售系统，国内主要文具公司正在建立自己的计算机网上电子商务系统。Internet是一小撮科学家接触并使用的科学工具跟着信息技术的成长，它已成为许多人可以涉及使用的。

1.3主要工作

该系统是用于文具售卖的，它分为前台和后台两个模块，前台模块这个是让用户使用的，在用户注册登录后，就可以查看文具信息、自己的购物车信息和订单状态信息等；后台模块这个主要是管理员使用，管理员登上后，就可以管理文具和订单办理。用户和管理员他们有各自的权限，他们两个的权限实现的过程是不同的。本系统用java语言开发，MyEclipse 8.5、MySQL和Tomcat分别是本系统的前端、后台数据库开发工具以及应用服务器，实现代码，在设计完成后进行测试工作并更正一些错误。

1.4本文研究内容及主要贡献

本设计是对文具销售管理系统的开发，想要设计出一个适应现在网络时代的文具销售管理系统。本文的大致内容结构如下所示：

第1部分是绪论，它主要介绍的是这个课题的意义和研究的目的，通过分析国内外的研究现状，清楚系统是用来做什么的，并能够对课题研究设计有了更加的了解；

第2部分是系统运行环境，介绍本系统使用的开发语言以及使用的原因；

第3部分是系统分析，首先需要从经济技术及操作方面来对系统设计是否能够实现、制作的难度，进行一个大体的分析，只有在明确系统的制作具有确实的可行性后，然后才能进一步进行系统的需求分析，即要思考系统要实现的功能，最后画出本系统的流程图、数据流图和写出数据字典；

第4部分是系统设计环节，是把想法进行付诸实践的部分，将功能模块划分出来，设计数据库的表，画实体联系图等；

第5部分是展示系统功能运行结果的部分，并且通过截图和核心代码呈现出来；

第6部分是反思总结和展望的部分，思考自己在开发设计中的不足和写出对系统未来的期待和探索。

**2 系统运行环境**

2.1 JSP

2.1.1 JSP简介

JSP与HTML技术是不同的，HTML是静态的，由HTML语言编写，对于用户来说交互性差，功能少，所以就要使静态变成动态，JSP应运而生，通过它可以使网页变成动态，是java的一种，是servlet的一个简化。JSP（Java Server Pages）是一种动态页面技术，它的主要目的是将表示逻辑从Servlet中分离出来[1]。

2.1.2 JSP访问数据库

要用JSP建立动态的网页，就要连接数据库，因为静态网页和动态网页的一个很大区别，就是有无数据库。所以为了达到这一目的要使用JDBC技术，使用该技术，就能访问数据库了。

以下是JDBC做的事情，按顺序列出：

第一步需要进行的一个最基础的事情就是把网页和数据库建立实质性的连接和联系；

第二步使用SQL语句并放入数据库；

第三步对结果进行操作。

2.2 MySQL

2.2.1MySQL 简介

MySQL在本人的系统中有着非常重要的位置，它是本系统的基石，可以说它是所有的基石，它就是一种数据库，是关系型数据库，其实与其他的数据库进行比较，它也不是非常完美的，但是由于它使用起来简便，使用成本又低，如果是一些中小设计，它一般是大部分人的首选，它可以做的事情对于中小设计是完全足够的,特别是与JSP结合来建设网站是非常的方便[2]。

2.2.2 MySQL数据库特点

MySQL优点较多，有以下几点：

(1)不管有几个用户，它可以同时接纳他们访问；

(2)因为体积小安装、调试方便；

(3)符合人们的使用习惯，值得依靠；

(4)它所使用的语言就是SQL语言，也是深受人们欢迎。

2.3 面向对象编程思想

面向对象的设计是目前流行的程序开发技术的主要思想。用比较通俗易懂的话来说，面向过程的编程语言的书写方式是一个main函数，每个过程都是采用函数的形式进行书写，在main函数中进行调用，通过函数调用来实现整个程序设计。面向对象的设计主要是定义数据格式，N多数据项，所有操作都是通过对数据项进行操作，每个数据项都可以抽象成对象，编程思路都是通过对象的形式进行传递和操作。面向对象是从面向过程发展而来，面向过程的函数方式编码在发展的过程中，存在一些弊端。随着社会的发展，科学的进步，软件的需求越来越复杂，复杂度呈现几何形式的增长。面向过程在编写复杂项目的时候，代码的可读性变差，可维护性也变差，单个文件的代码行数直线上升，维护成本越来越高。面向对象的三大是封装、继承和多态。面向对象和面向过程都各有优点和缺点，各有自己的试用场合。面向过程主要优点是性能比面向对象高，系统开销比较小。面向对象的代码实现比较容易维护，容易复用和扩展，面向对象是用于封装继承多态特性，通过面向对象编程语言更容易编写低内聚高耦合的代码，性能相对面向过程略微降低。

2.4 本章小结

通过介绍它们的主要特点和相关优势，来阐述为什么要使用这些技术，它们会给系统带来什么样的界面设计、动态页面效果、存储多样的数据。因为这些技术都是目前使用频率相对较高的，而且相关的软件有免费使用的期限，在一定程度上节约了系统制作的成本，为本设计的开发提供了极大的便捷性和潜在的经济效益。

**3 系统分析**

3.1 可行性分析

由于传统的文具销售存在很多的不足，本系统根据这些不足来开发设计。商家一直以来对文具的管理都是在文具店里面用人工方式管理，这种方式在计算机技术发展非常快的时代里，对商家的管理没有多大的好处。本系统与传统文具店文具销售的流程相似，实现文具信息化，减轻文具店工作人员的工作量，文具店的管理人员可以利用本系统对文具店里的文具进行展示、新文具上架、编辑删除文具信息等操作。顾客可以通过本系统查询想要购买的文具，不需要来到实体文具店挑选，节省大量的时间。接下来我将从3个方面依次进行分析。

3.1.1经济可行性

本系统需要安装的有JDK、TOMCAT、MySQL和MyEclipse，这些都可以免费从相关官网下载安装，一般普通的电脑就可以稳定的运行。

3.1.2 技术可行性

本系统采用的是JSP技术和MySQL开发，Windows XP以上的操作系统。JSP技术功能强大，MySQL灵活易维护，这两个给设计者提供了很大的帮助，上手学习很快，有解决方案供系统开发者学习解决开发过程中遇到的问题。

3.1.3 操作可行性

用户打开浏览器，输入网址访问文具销售管理系统的网站，这个系统有效的保护了用户的信息。本系统在一般电脑上都可以运行，安装调试运行该系统不会改变任何一个原电脑的配置，用户和管理员根据本系统的提示信息就知道如何操作。

3.2 需求规格说明

网络的发展让越来越多的消费者不愿出门购买物品，面对种类繁多的文具店，浪费时间一个一个的寻找自己所需的文具。同时，传统实体文具店必须跟上大数据时代的潮流，跟上时代步伐，采取新的盈利渠道。

（1）使用者分为两类：管理员和客户，这两个使用者都有他们各自的使用权限，提高本系统的安全性。在文具销售管理系统中，管理员具有本系统的超级权限，客户的权限让他只享有部分功能，比如查询，他不能操作数据；

（2）管理员可以根据新到文具的种类进行分类处理，添加到系统对应的文具分类中，最重要的是管理员要及时处理顾客的订单信息；

（3）用户使用本系统前要先注册信息登录，登录后可以查询自己想要买的文具，也可以查询自己的所有订单信息。

3.2.1 系统流程图

系统流程图如图3-1和图3-2所示。



**图3-1 用户系统流程图**



**图3-2 管理员系统流程图**

3.2.2 数据流图

通过一个个的黑盒子表示每个部件，体现数据在各个实体或是组分之间的流动，它没有对数据进行任何处理或者加工，它是物理的数据流图。

顶层数据流图如图3-3所示。



**图3-3 顶层数据流图**

一级数据流图如图3-4所示。

**图3-4 一级数据流图**

二级数据流图如图3-5所示。



**图3-5 二级数据流图**

3.2.3 数据字典

数据流名称：管理员信息

简述：管理员拥有文具店整体的管理权限

数据流组成：管理员信息=管理员ID+管理员名称+密码

存放位置：数据库t\_admin表中

数据流名称：文具信息

简述：由管理员提供，供用户查看

数据流组成：文具信息=文具ID+文具名+定价+当前价+折扣+制造商+出厂时间

存放位置：数据库t\_stationery表中

数据流名称：用户信息

简述：用户是的系统的使用者

数据流组成：用户信息=用户ID+用户名+密码+邮箱

存放位置：数据库t\_user表中

数据流名称：分类信息

简述：对文具进行分类展示

数据流组成：分类信息=分类ID+分类名+分类的描述

存放位置：数据库t\_category表中

数据流名称：购物车信息

简述：购买的文具的具体细节

数据流组成：购物车信息=购物项ID+文具ID+数量

存放位置：数据库t\_cartitem表中

数据流名称：订单信息

简述：用户购买文具生成的具体订单细节

数据流组成：订单信息=订单ID+用户ID+下单时间+合计金额+订单状态+收货地址+订单号+数量+文具名+文具当前价+文具图片

存放位置：数据库t\_order表和t\_orderitem表中

**图3-6 数据字典**

3.3 本章小结

系统分析为接下来的系统设计提供坚实的基础，通过这些分析的结果，设计起来才会更加清晰、容易。比如说在最开始设计之前，要明确的定位这个系统有哪些功能，针对什么样的用户所设计的，系统这个整体的框架结构怎么规划，每一个部分都怎么具体实现。在对本系统有了一个非常全面的认识之后，开始考虑经费是否足够支撑该系统的开发和完善，开发人员的技术是否达到一定的要求，系统的操作是否符合日常的规律和人们的操作习惯。最后，进入更具体的对系统的需求分析，在这方面，首先需要对整个系统的模块进行合理的规划，画出系统流程图，有一个很清楚的脉络结构，也可以提高对接下来的代码编写的速度和效率。接着，需要对每一个实体或组分的数据传输和流动进行更细化的分析，完成相应的数据流图。最后，详细的分析一些数据流的名称、简述、存放位置，对它们有一个更清晰的认识。以此来完成对本系统一个客观而又具体的分析。

**4 系统设计**

4.1 系统功能设计

功能模块如图4-1和图4-2所示。

**图4-1 前台功能模块图**

**图4-2 后台功能模块图**

4.2 数据库概念模型的设计

信息管理系统需要能够储存和管理海量的数据，因此必须以数据库为基础才能实现这一功能。这次设计选择MySQL数据库来开发。

本次系统设计通过实体-联系方法来描述概念模型，也即E-R图。各个实体和E-R图如图4-3至4-9所示。



**图4-3 用户实体图**



**图4-4 文具分类实体图**



**图4-5 文具实体图**



**图4-6 购物车实体图**



**图4-7 订单实体图**



**图4-8 管理员实体图**

**图4-9 E-R图**

4.3数据库逻辑结构的设计

在数据库中构建数据库表goods，一共有7个表格，goods数据库中包含的数据表及其相应功能如表4-1所示。

**表4-1 goods数据库中的数据表及其相应功能**

|  |  |
| --- | --- |
| 数据表 | 功能 |
| t\_user | 存放用户的基本信息 |
| t\_category | 存放文具分类信息 |
| t\_stationery | 存放文具信息 |
| t\_cartitem | 存放购物车细节信息 |
| t\_order | 存放订单信息 |
| t\_orderitem | 存放订单细节信息 |
| t\_admin | 存放管理员信息 |

（1）t\_user表用于存放用户的基本信息，如表 4-2所示。

**表 4-2t\_user表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | Uid | char(32) | 用户ID | NOT NULL (Key) |
| 2 | Loginname | varchar(50) | 用户名 | NULL |
| 3 | Loginpass | varchar(50) | 用户密码 | NULL |
| 4 | Email | varchar(50) | 邮箱地址 | NULL |

（2）t\_category表用于存放文具分类信息，如表 4-3所示。

**表 4-3t\_category表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | cid | char(32) | 分类ID | NOT NULL (Key) |
| 2 | Cname | varchar(50) | 分类名称 | NULL |
| 3 | Pid | char(32) | 父分类ID | NULL |
| 4 | Desc | varchar(100) | 分类描述 | NULL |
| 5 | orderBy | int(11) | 序号，用来排序 | NOT NULL |

（3）t\_stationery表用于存放文具信息，如表 4-4所示。

**表 4-4 t\_stationery表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | Bid | char(32) | 文具ID | NOT NULL (Key) |
| 2 | Bname | varchar(200) | 文具名称 | NULL |
| 3 | Price | decimal(8,2) | 定价 | NULL |
| 4 | currPrice | decimal(8,2) | 当前价 | NULL |
| 5 | Discount | decimal(3,1) | 折扣 | NULL |
| 6 | Press | varchar(100) | 制造商 | NULL |
| 7 | Publishtime | char(10) | 出厂时间 | NULL |
| 8 | cid | char(32) | 所属分类ID | NULL |
| 9 | image\_w | varchar(100) | 大图路径 | NULL |
| 10 | image\_b | varchar(100) | 小图路径 | NULL |
| 11 | orderBy | int(11) | 序号，用来排序 | NOT NULL |

（4）t\_cartitem表用于存放购物车细节信息，如表 4-5所示。

**表 4-5 t\_cartitem表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | cartItemId | char(32) | 购物车细节ID | NOT NULL (Key) |
| 2 | Quantity | int(11) | 数量 | NULL |
| 3 | Bid | char(32) | 文具ID | NULL |
| 4 | Uid | char(32) | 用户ID | NULL |
| 5 | orderBy | int(11) | 序号，用来排序 | NOT NULL |

（5）t\_order表用于存放订单信息，如表 4-6所示。

**表 4-6 t\_order表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | Oid | char(32) | 订单ID | NOT NULL (Key) |
| 2 | Ordertime | char(19) | 下单时间 | NULL |
| 3 | Total | decimal(10,2) | 合计金额 | NULL |
| 4 | Status | int(11) | 订单状态 | NULL |
| 5 | Address | varchar(1000) | 收货地址 | NULL |
| 6 | Uid | char(32) | 用户ID | NULL |

（6）t\_orderitem表用于存放订单细节信息，如表 4-7所示。

**表 4-7t\_orderitem表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | orderItemId | char(32) | 订单细节ID | NOT NULL (Key) |
| 2 | Quantity | int(11) | 数量 | NULL |
| 3 | Subtotal | decimal(8,2) | 小计 | NULL |
| 4 | Bid | char(32) | 文具ID | NULL |
| 5 | Bname | varchar(200) | 文具名称 | NULL |
| 6 | currPrice | decimal(8,2) | 文具当前价 | NULL |
| 7 | image\_b | varchar(100) | 文具小图路径 | NULL |
| 8 | Oid | char(32) | 所属订单ID | NULL |

（7）t\_admin表用于存放管理员信息，如表 4-8所示。

**表 4-8 t\_admin表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型及长度 | 说明 | 备注 |
| 1 | adminId | char(32) | 管理员ID | NOT NULL (Key) |
| 2 | Adminname | varchar(50) | 管理员名称 | NULL |
| 3 | Adminpwd | varchar(50) | 管理员密码 | NULL |

4.4本章小结

本章可有效根据第3章数据流图和数据字典的内容，思考并分析文具销售管理系统具体可以实现哪些功能。在对系统的功能有一个明确的认识和规划后，开始设计数据库goods，建立所需功能的表7个，来实现对用户、文具、订单等一系列销售和管理文具的信息存放。鉴于对第3章的系统分析有了较为全面的规划，那么对接下来进入本章系统的设计，其实是提供了一个非常清晰的架构以及明确的思路。有了方向，下面具体功能的实现和设计就是一个主要的重难点，那么通过这章的设计分析，我们接下来需要进入系统的实现环节，这是项目的重中之重。

**5 系统实现及效果分析**

5.1 前台部分设计与实现

5.1.1 首页设计

前台首页如图5-1所示。

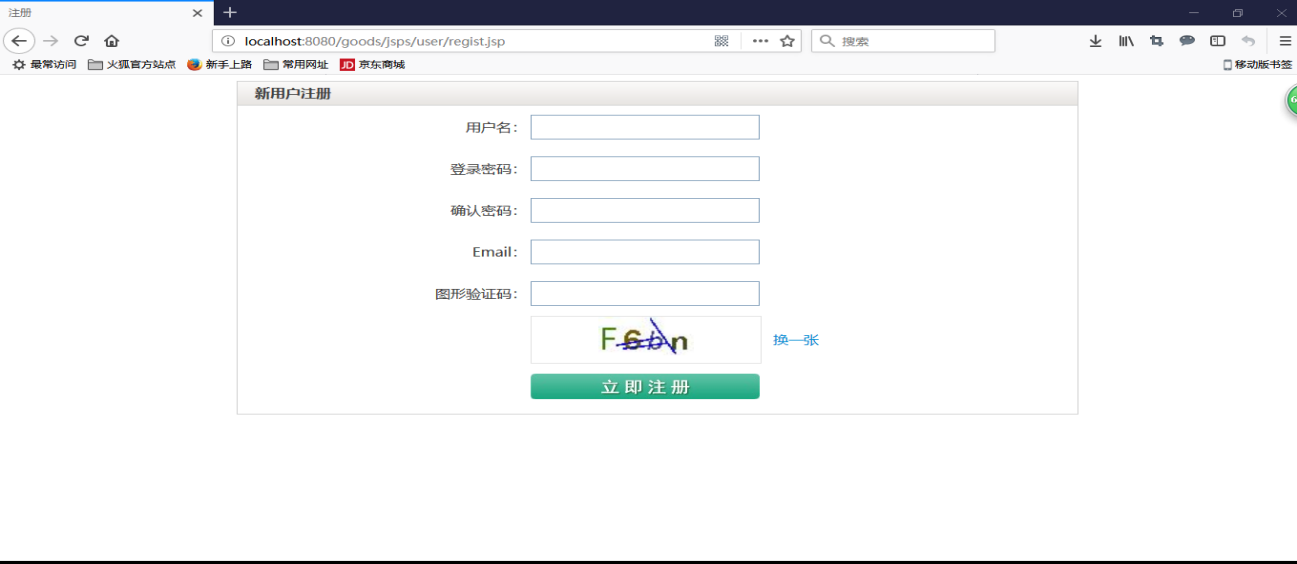


**图5-1 前台首页图**

5.1.2 用户模块

（1）用户注册

一般进入一个新系统，想要浏览更多的信息，需要进行新用户的注册。点击本系统的相关按钮可以快速进入注册页面。用户注册界面如图5-2所示。



**图5-2 用户注册图**

在注册过程中，输入的内容有格式或字符错误的话，在对话框的右边都会出现相对应的错误提示，用户需要按照提示填写相对应的内容。用户注册错误提示界面如图5-3和图5-4所示。



**图5-3 用户注册错误提示一图**



**图5-4 用户注册错误提示二图**

Java部分关键代码如下。

/\*\*

\* 添加User

\* @param user

\* @throws SQLException

\*/

publicvoid add(User user) throws SQLException {

String sql = "insert into t\_user values(?,?,?,?)";

Object[] params = {user.getUid(), user.getLoginname(), user.getLoginpass(),

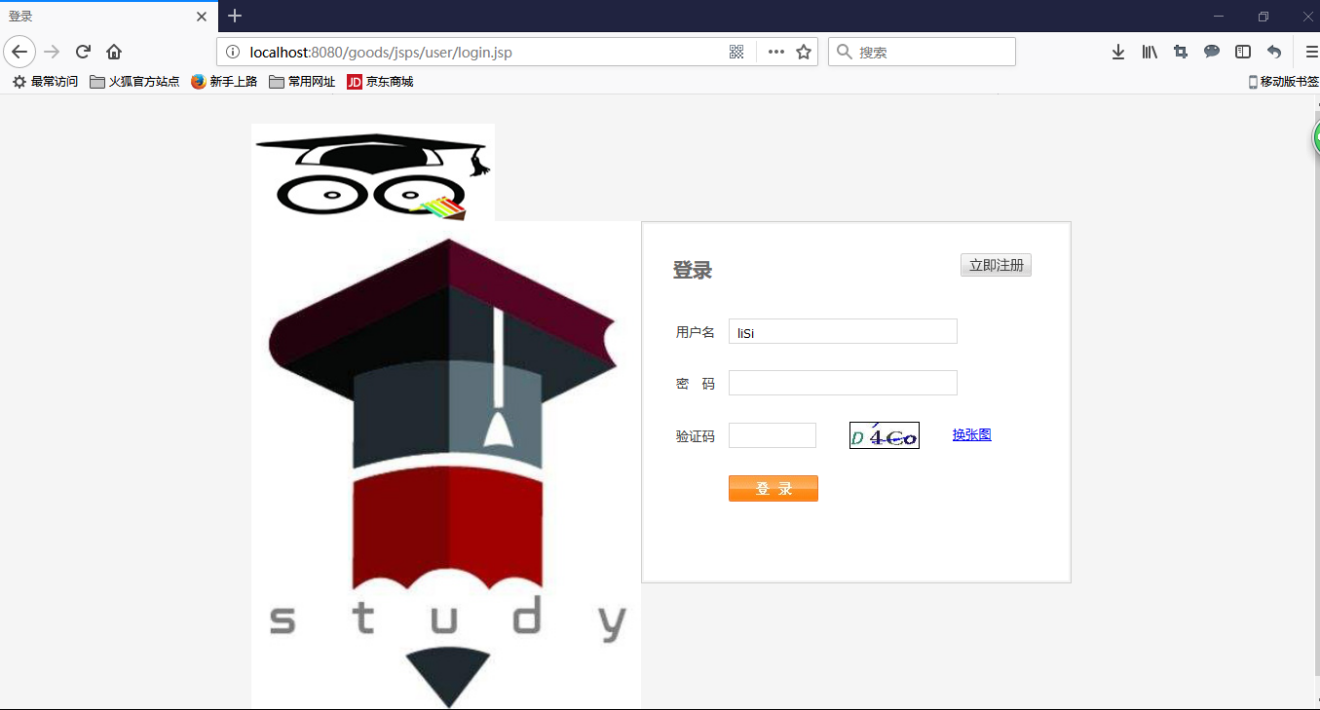
user.getEmail()};

qr.update(sql, params);

}

（2）用户登录

注册完成后，刷新系统，输入注册时所填写的相关信息，最后点击最下方按钮实现系统的登录。用户登录界面如图5-5所示。



**图5-5 用户登录图**

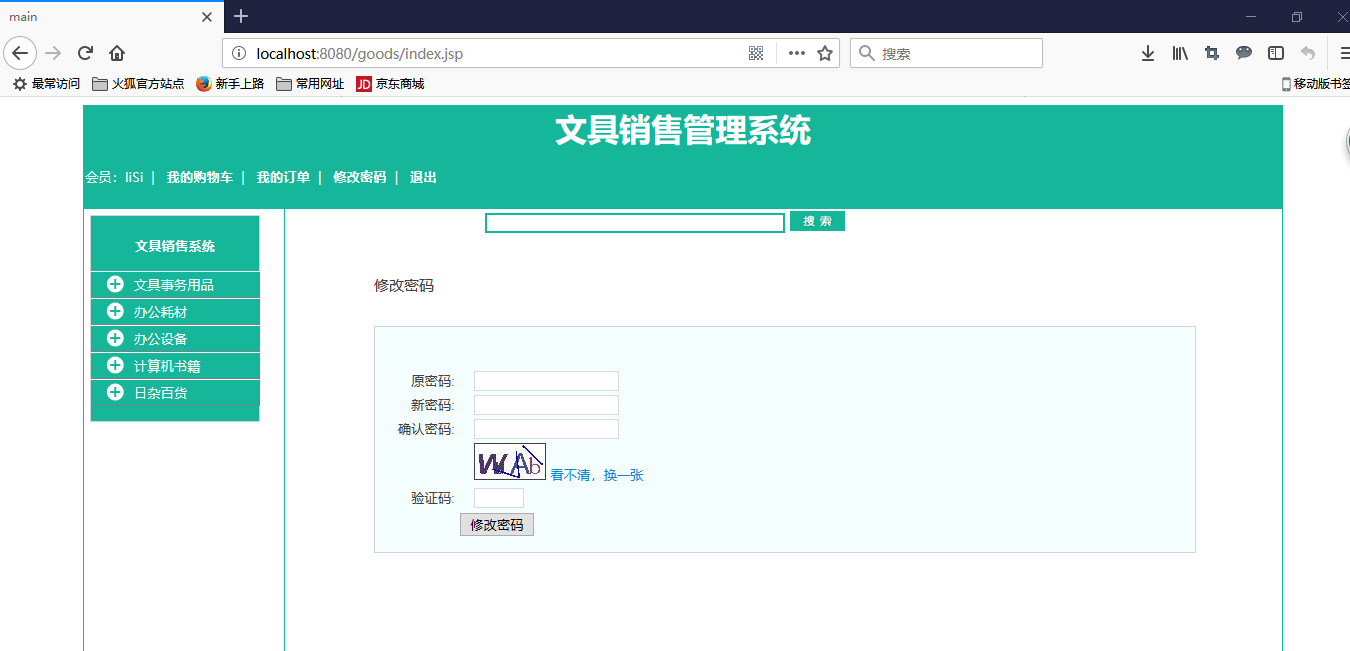
同样在登录时，这个填写错误的登录信息，也会出现相对应的提示。用户登录错误提示界面如图5-6所示。



**图5-6 用户登录错误提示图**

（3）修改密码

用户可根据自己的需要，对登录密码进行修改操作，点击相应按钮可实现。修改密码界面如图5-7所示。



**图5-7 修改密码图**

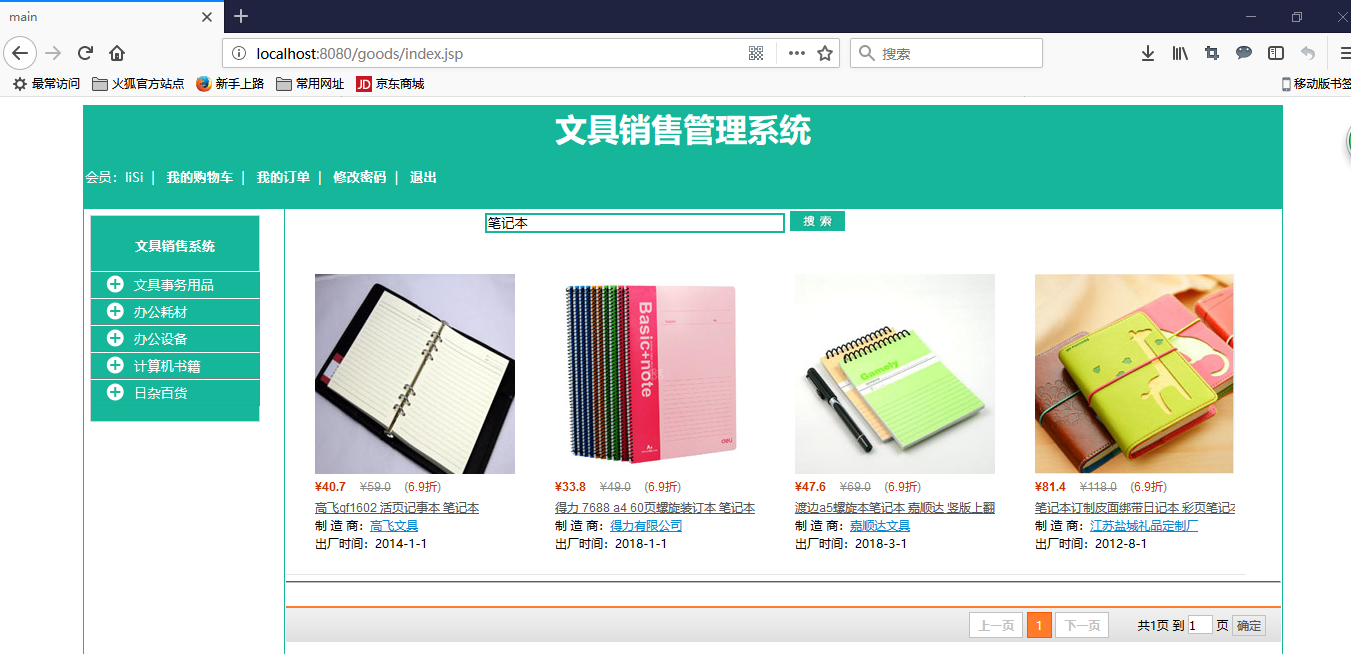
5.1.3 文具模块

（1）文具列表

点击系统中文具分类会显示详细的图文信息，用户可自行浏览，除此之外，顾客可以在文具列表上方搜索框中输入文具名关键字进行查询，以提供更多精确的选择。文具列表和搜索文具界面如图5-8和图5-9所示。



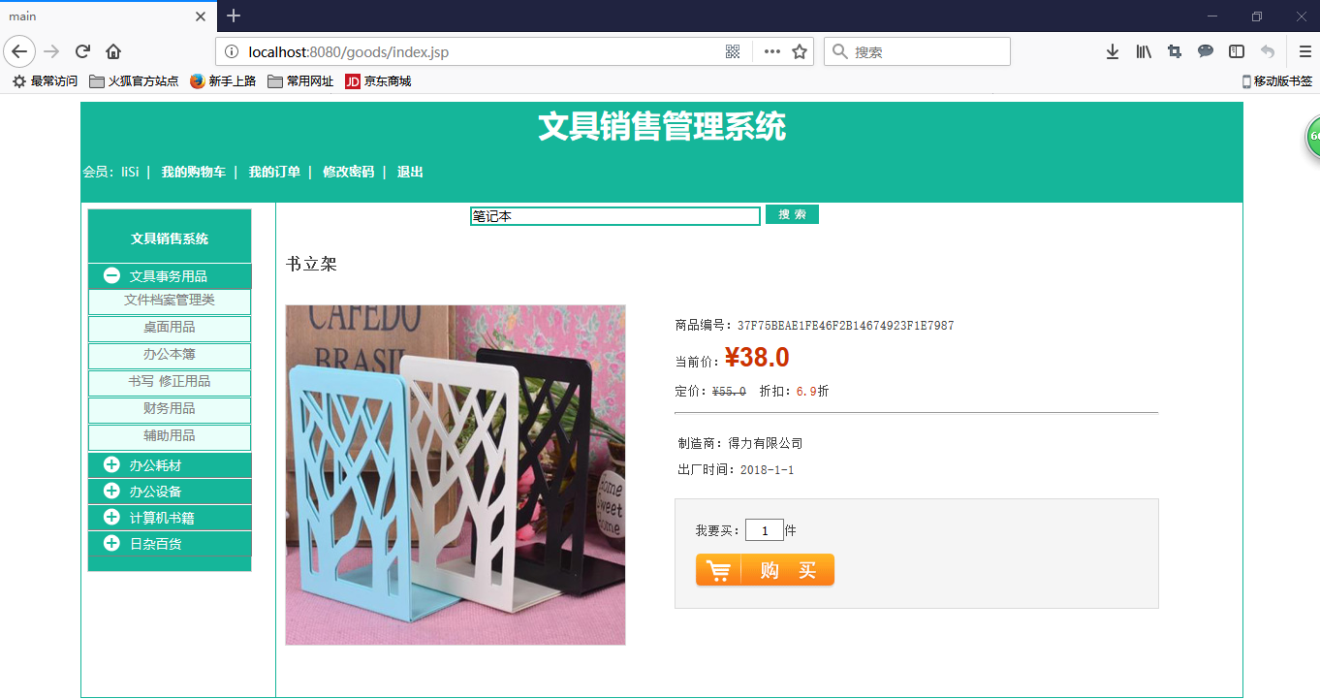
**图5-8 文具列表图**



**图5-9 搜索文具图**

（2）文具详细

点击某一个文具，在系统的窗口中将会显示该文具的详细信息：该文具实物的真实图片、管理员设置的价格、相关促销信息等，使消费者对该文具的性价比有一个直观的判断。文具详细界面如图5-10所示。



**图5-10 文具详细图**

5.1.4 购物车

选择自己想买的文具确定数量，添加到用户自己购物车中。用户也可以在首页“我的购物车”中查询自己的购物信息，每条信息会显示文具图片、文具名称、文具当前价、数量和小计，如果用户想再买多一点的文具或者少买一点的话，可以在此页面上修改数量，当然如果不想买这个文具了也可以删除它。购物车还有一个类似“云”的功能：当我们关闭该系统不再进行浏览时，数据信息不会凭空消失，在下一次打开并登录该系统后，仍然能够看到我们之前所存储的物品。购物车界面如图5-11所示。



**图5-11 购物车图**

Java部分关键代码如下。

/\*\*

\* 跳转添加页面

\*/

public String add() {

return*SUCCESS*;

}

5.1.5 订单模块

选择所需要购买的文具，在订单界面会出现文具的总额和详细信息，以及用户需要收货的地址，最后可选择提交订单，用户对所购买的物品进行总体金额的结算和支付。订单准备和下单成功界面如图5-12和图-13所示。

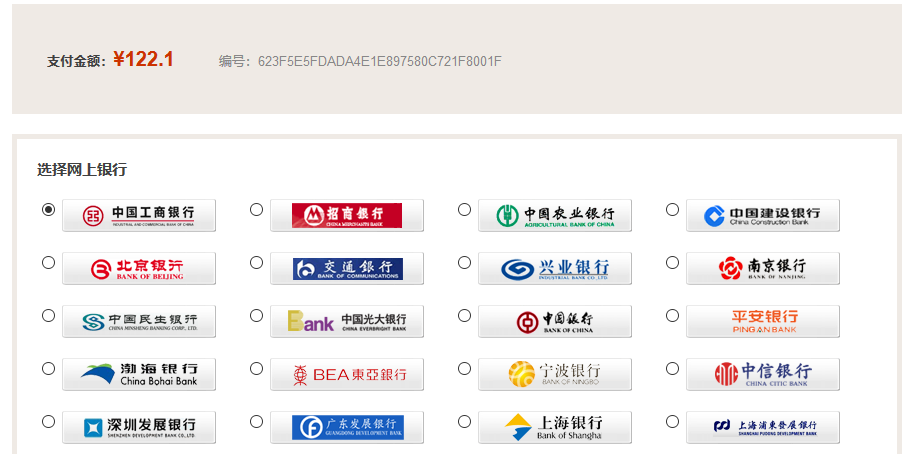


**图5-12 订单准备图**



**图5-13 下单成功图**

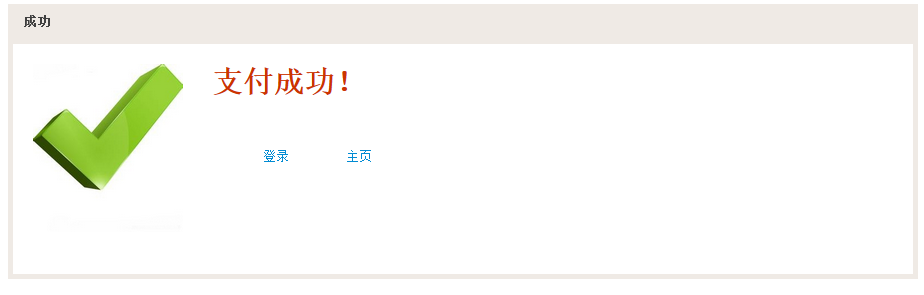
订单生成后，会自动生成订单编号以及购物的金额和收货地址，同时感谢用户对本系统的支持。支付页面如图5-14所示。





**图5-14 支付页面图**

在“支付”页面中选择自己可支付的银行卡，点击最下方按钮，用户就支付成功了。至此，就差不多完成线上购物的全部过程，等待收货即可。支付成功界面如图5-15所示。



**图5-15 支付成功图**

在本系统首页上部点击“我的订单”，可以知道自己的文具订单中有哪一些文具，它们处于什么样的状态，对于已下单的物品系统会自动显示下单时间以及还需要用户进行的操作。订单列表界面如图5-16所示。



**图5-16 订单列表图**

用户可根据自己的实际需要，在订单信息中，选择某个订单进行自己想要的操作，来实现对所需购买文具的信息确认和收货情况判断，对于不想购买的文具也可取消订单。订单详细查看、取消和确认收货界面如图5-17、图5-18和图5-19所示。



**图5-17 订单详细查看图**



**图5-18 订单详细取消图**

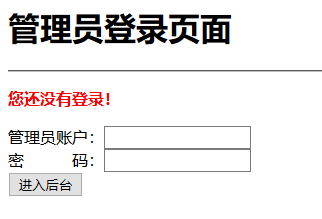


**图5-19 订单详细确认收货图**

5.2 后台部分设计与实现

5.2.1 管理员登录

管理员登录页面和后台主页如图5-20和图5-21所示。



**图5-20 管理员登录图**



**图5-21 后台主页图**

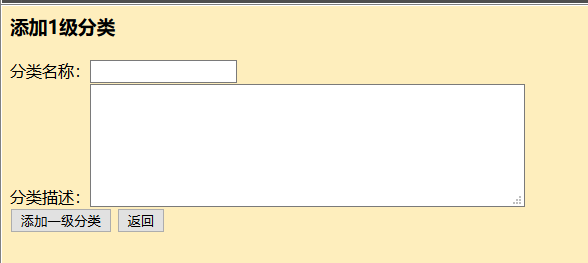
5.2.2 分类管理

分类管理是该系统的管理员对文具的分类进行的整体规划，只有管理员可以修改，管理员可根据具体情况以及用户的反馈来设计更为科学精确的分类框架。分类管理对文具销售系统至关重要，消费者可根据分类快速查找到需要购买的文具所在的分类，进行购物的选择。分类列表界面如图5-22所示。

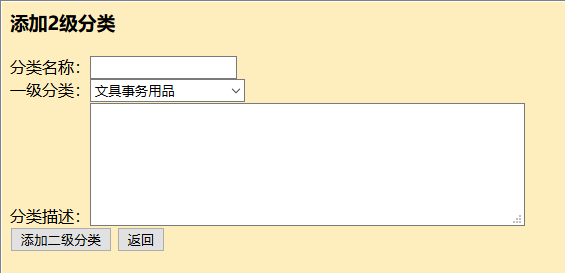


**图5-22 分类列表图**

管理员对文具的分类以及等级有清楚的认知，可根据实际情况或具体要求对文具的分类进行添加修改和删除，具体界面如图5-23和5-24所示。



**图5-23 添加1级分类图**



**图5-24 添加2级分类图**

Java部分关键代码如下。

/\*\*

\* 添加分类

\* @param category

\*/

publicvoid add(Category category) {

try {

categoryDao.add(category);

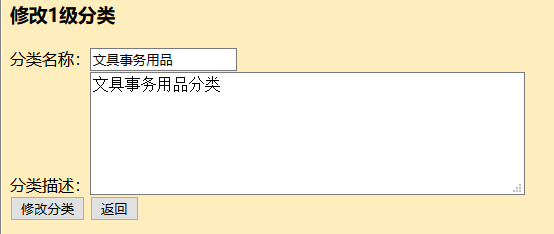
} catch (SQLException e) {

thrownew RuntimeException(e);

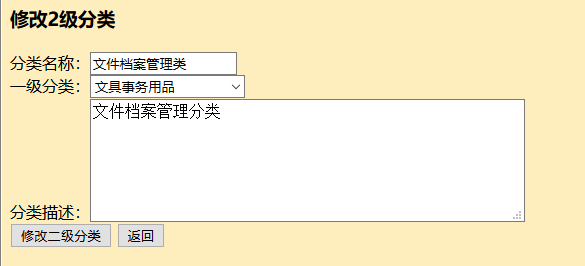
}

}

根据分类列表，可更好对分类进行管理，这样管理员可结合实际需要，修改设置不合理的分类以及删除不需要的分类。修改分类界面如图5-25和图5-26所示。



**图5-25 修改1级分类图**



**图5-26 修改2级分类图**

Java部分关键代码如下。

/\*\*

\* 修改分类

\* @param category

\*/

publicvoid edit(Category category) {

try {

categoryDao.edit(category);

} catch (SQLException e) {

thrownew RuntimeException(e);

}

}

5.2.3 文具管理

后台管理员的管理和操作权限在系统中占比较大，可以进行好多用户不能够实现的操作，这是有且仅有后台管理员才能做的事，比如对文具的管理，详情看图5-27。



**图5-27 文具列表图**

进入页面，会看到，点击它管理员可以让新文具上架。但要注意在添加新文具时对图片的尺寸、图片的扩展名、价格折扣是否是合法小数等进行校验，如果校验无误，添加文具会成功。添加文具界面如图5-28所示。



**图5-28 添加文具图**

底下的页面陈列着各种文具，可以随意轻易地查看文具的信息，并可以用管理员的权限进行各种操作，这类操作可以使文具的信息更加准确，方便了管理员管理，给用户提供了便利。编辑或删除文具界面如图5-29所示。



**图5-29 编辑或删除图**

Java部分关键代码如下。

/\*\*

\* 删除文具

\* @param bid

\* @throws SQLException

\*/

publicvoid delete(String bid) throws SQLException {

String sql = "delete from t\_stationery where bid=?";

qr.update(sql, bid);

}

5.2.4 订单管理

本系统还为管理员设置了可以按不同的订单状态来查询订单，这样可以方便管理员处理顾客的订单信息。订单列表界面如图5-30所示。



**图5-30 订单列表图**

系统还为管理员设置了另一种方法查询订单，在每个订单的后面会有“查看”、“取消”、“发货”，管理员可以根据不同的提示对订单进行处理，这样大大减少了管理员的工作量。订单详细页面如图5-31所示。



**图5-31 订单详细页面图**

5.3 本章小结

本章利用核心代码、功能截图、文字描述对本系统的各个模块功能进行介绍，主要分为前台部分设计与实现和后台部分设计与实现，都对其做了详尽的描述。用户能很好的进行登录，在管理员和用户的权限上有明显的区分，使消费者有较好的购买体验，管理员也能清晰的管理后台，系统功能也基本实现了。

**6 总结与展望**

6.1总结

时间总是过得很快，我从接触这个系统到现在已经有三个月，经过自己的努力，已经做出基本功能，主要有前台用户操作页面，实现新用户的注册以及老用户的登录、购买自己想要的文具、下订单等，后台管理员操作界面这个主要是用来进行文具分类管理、文具信息管理和顾客订单处理等。文具已经与我们的生活分不开了，可以说是息息相关，在以前，购买文具都要去文具店，而且文具店一般规模小，东西陈列杂乱，运用这个系统可以很好的解决这些问题。通过文具销售管理系统的开发，让我的java知识掌握的更牢固，但对于我来说，毕竟到大学才接触到这些技术的，不足的地方还是有的，这需要更多的努力。

6.2展望

本系统是一个小型的文具销售管理系统，只实现了基本的功能，还有一些功能需要改进完善：

（1）用户管理的信息没有考虑全面，只添加了几个重要的信息，并且要进一步加强用户信息的安全性；

（2）用户支付方面本系统使用的是模拟支付，未实现真正的支付，可以使用第三方支付平台来实现；

（3）管理员处理顾客订单要及时，在以后的设计中可以设置订单提醒的功能。

# 毕业设计体会

这次毕业设计开发我学到了一些经验：

（1）我们在做一个系统时，我们首先要做好分析，分析我们的这个系统要给谁使用，这个系统是用来做什么的，一开始不要一听到做系统就首先开始编程，不知道系统的功能是什么编程写的再好也是无用功；

（2）我们把系统分析做好后，我们可以写一些文档记录我们在编写程序代码时的步骤，这样我们在写代码时才不会没有条理；

（3）我们在写程序时要时刻谨慎，特别是小的细节如在不该有空格的地方加了空格，万一加了这个我们在程序出现错误的时候，想要找到那个错误会耗费我们很多的时间，所以平时在写的时候要时刻关注小的细节。

在这次毕业设计中，我主要是因为之前很少真正的接触系统设计开发的工作，在应用这个方面缺乏了很多经验，所以在设计开发中走了许多的弯路。虽然毕业设计已经结束了，但它对我的影响永远都在，这是个很宝贵的经验。从本次我的毕业设计中我学到了很多书本上没有的知识，老师的耐心指导我时刻铭记在心，老师的帮助提高了我的知识水平和解决实际问题的能力，我们一定要把自己学到的知识运用到现实中我们要面对的难题。

参考文献

[1]沈泽刚, 秦玉平. Java Web编程技术[M].北京: 清华大学出版社, 2010.

[2]王珊,萨师煊. 数据库系统概论[M]. 北京:高等教育出版社, 2006.

[3]耿祥义, 张跃平. JSP实用教程[M]. 北京: 清华大学出版社, 2003.

[4]麦斯阿塞克.需求分析与系统设计[M]. 北京: 机械工业出版社, 2009.

[5]张海藩.软件工程导论[M]. 北京: 清华大学出版社, 2007.

[6][埃克尔](http://baike.baidu.com/view/3600579.htm). Java编程思想[M].北京:机械工业出版社, 2007.

[7]徐明华. Java Web整合开发与项目实战——JSP、Ajax、Struts、Hibernate、Spring[M].北京:人民邮电出版社, 2009.

[8]王国辉. Java web开发技术方案宝典[M]. 北京: 人民邮电出版社, 2008.

[9]万欣. 电子商务概论[M]. 北京: 化学工业出版社, 2010.

[10]徐喆.销售管理信息系统的开发与设计[J].商场现代化,2009(05):34-35.

[11]邵振文,龙伟,杜仕冲.基于Web的中小型企业销售管理信息系统的设计与应用[J].中国制造业信息化,2006(07):7-9.

[12]张双喜.论办公自动化系统的发展与建设[J].山西焦煤科技,2005(03):10-12.

[13]李军怀,张景,耿国华.基于WEB模式的销售管理信息系统[J].计算机工程与应用,2001(04):99-102.

[14]李欢欢.基于JAVA 和WEB的网络管理系统[M].昆明:昆明理工大学出版社，2002.

[15]铁军.Web数据库技术[M].北京: 清华大学出版社,2004.

[16]张志刚.基于ASP 的网上购物系统设计与实现[M].吉林:吉林大学出版社，2012.

[17]Abraham Silberschatz, Henry F. Korth, S. Sudar shan. Database System Concepts[M]. McGraw-Hill, 2010.

[18]Peter Rob, Carlos Coronel .Database Systems[M]. Cengage Learning EMEA, 2008.

[19]J. Stanley Warford. Computer Systems[M]. Jones & Bartlett Publishers, 2009.

[20] Toby J. Teorey, Stephen Buxton, Lowell Fryman, Terry Halpin. Database design: know it all [M]. Morgan Kaufmann, 2009.

[21]John W. Satzinger. Systems Analysis and Design in a Changing World[M]. Cengage Learning EMEA, 2008.

[22]Alan Beaulieu. Learning SQLO'Reilly Series[M]. O'Reilly Media, I nc., 2009.

[23]Clare Churcher. Beginning SQL Queries: From Novice to Profession alApress Series[M]. Apress, 2008.

[24]Andy Oppel, Robert Sheldon. SQL: a beginner's guide BEGINNER'S G UIDE[M]. McGraw-Hill Professional, 2008.

**致谢**

**英文翻译资料**

A.英文原文

Data Design Elements

Like other software engineering activities, data design (sometimes referred to as data architecting) creates a model of data and/or information that is represented at a high level of abstraction (the customer/user’s view of data). This data model is then refined into progressively more implementation-specific representations that can be processed by the computer-based system. In many software applications, the architecture of the data will have a profound influence on the architecture of the software that must process it.

The structure of data has always been an important part of software design. At the program component level, the design of data structures and the associated algorithms required to manipulate them is essential to the creation of high-quality applications. At the application level, the translation of a data model (derived as part of requirements engineering) into a database is pivotal to achieving the business objectives of a system. At the business level, the collection of information stored in disparate databases and reorganized into a “data warehouse” enables data mining or knowledge discovery that can have an impact on the success of the business itself. In every case, data design plays an important role. Data design is discussed in more detail in Chapter 9.

Interface Design Elements

The interface design for software is analogous to a set of detailed drawings (and specifications) for the doors, windows, and external utilities of a house. These drawings depict the size and shape of doors and windows, the manner in which they operate, the way in which utility connections (e.g., water, electrical, gas, telephone) come into the house and are distributed among the rooms depicted in the floor plan. They tell us where the doorbell is located, whether an intercom is to be used to announce a visitor’s presence, and how a security system is to be installed. In essence, the detailed drawings (and specifications) for the doors, windows, and external utilities tell us how things and information flow into and out of the house and within the rooms that are part of the floor plan. The interface design elements for software depict information flows into and out of the system and how it is communicated among the components defined as part of the architecture.

There are three important elements of interface design: (1) the user interface (UI); (2) external interfaces to other systems, devices, networks, or other producers or consumers of information; and (3) internal interfaces between various design components. These interface design elements allow the software to communicate externally and enable internal communication and collaboration among the components that populate the software architecture.

UI design (increasingly called usability design) is a major software engineering action and is considered in detail in Chapter 11. Usability design incorporates aesthetic elements (e.g., layout, color, graphics, interaction mechanisms), ergonomic elements (e.g., information layout and placement, metaphors, UI navigation), and technical elements (e.g., UI patterns, reusable components). In general, the UI is a unique subsystem within the overall application architecture.

The design of external interfaces requires definitive information about the entity to which information is sent or received. In every case, this information should be collected during requirements engineering (Chapter 5) and verified once the interface design commences. The design of external interfaces should incorporate error checking and (when necessary) appropriate security features.

The design of internal interfaces is closely aligned with component-level design (Chapter 10). Design realizations of analysis classes represent all operations and the messaging schemes required to enable communication and collaboration between operations in various classes. Each message must be designed to accommodate the requisite information transfer and the specific functional requirements of theoperation that has been requested. If the classic input-process-output approach to design is chosen, the interface of each software component is designed based on data flow representations and the functionality described in a processing narrative.

In some cases, an interface is modeled in much the same way as a class. In UML, an interface is defined in the following manner [OMG03a]: “An interface is a specifier for the externally-visible [public] operations of a class, component, or other classifier (including subsystems) without specification of internal structure.” Stated more simply, an interface is a set of operations that describes some part of the behavior of a class and provides access to these operations.

For example, the SafeHome security function makes use of a control panel that allows a homeowner to control certain aspects of the security function. In an advanced version of the system, control panel functions may be implemented via a wireless PDA or mobile phone.

The ControlPanel class (Figure 8.5) provides the behavior associated with a keypad, and therefore, it must implement the operations readKeyStroke()and decodeKey (). If these operations are to be provided to other classes (in this case, WirelessPDA and MobilePhone), it is useful to define an interface as shown in the figure. The interface, named KeyPad, is shown as an <<interface>> stereotype or as a small, labeled circle connected to the class with a line. The interface is defined with no attributes and the set of operations that are necessary to achieve the behavior of a keypad.

The dashed line with an open triangle at its end (Figure 8.5) indicates that the ControlPanelclass provides KeyPadoperations as part of its behavior. In UML, thisis characterized as a realization. That is, part of the behavior of ControlPanel will be implemented by realizing KeyPad operations. These operations will be provided to other classes that access the interface.

B.原文的翻译

数据设计元素

与其他软件工程活动一样，数据设计(有时也称为数据架构)创建了一个数据和（或）信息模型，这些数据和（或）信息在高抽象层(客户/用户的数据视图)中表示。然后，将这个数据模型细化为更多的实现特定的表示，这些表示可以由基于计算机的系统进行处理。在许多软件应用程序中，数据的体系结构将对必须处理它的软件的体系结构产生深远的影响。

数据结构一直是软件设计的重要组成部分。在程序组件级别，数据结构的设计和操作它们所需的相关算法对于创建高质量的应用程序是必不可少的。在应用程序级别，将数据模型(作为需求工程的一部分)转换为数据库是实现系统的业务目标的关键。在业务级别，将存储在不同数据库中的信息集合并重新组织到“数据仓库”中，使数据挖掘或知识发现能够对业务本身的成功产生影响。在每个案例中，数据设计都扮演着重要的角色。第9章将详细讨论数据设计。

界面设计元素

软件的界面设计类似于一套房子的门、窗和外部设施的详细图纸(和规范)。这些图纸描述了门窗的大小和形状，以及它们的操作方式，在房子里的公用设施的连接方式(例如水、电、煤气、电话)，并分布在楼层平面图所示的房间里。他们告诉我们门铃在哪里，是否要使用对讲机来宣布访客的存在，以及如何安装一个安全系统。从本质上讲，门、窗和外部设施的详细图纸(和规范)告诉我们，房屋内和房间内的物品和信息是如何进出的，这些都是楼层平面图的一部分。软件的界面设计元素描述了系统中进出的信息，以及如何在定义为体系结构一部分的组件之间进行通信。

界面设计有三个重要元素:(1)用户界面(UI);(2)其他系统、设备、网络或其他生产者或消费者的外部接口;(3)各种设计组件之间的内部接口。这些接口设计元素允许软件在外部进行通信，并允许在软件体系结构中的组件之间进行内部通信和协作。

UI设计(越来越被称为可用性设计)是一个主要的软件工程操作，在第11章中被详细讨论。可用性设计包含了美学元素(如布局、颜色、图形、交互机制)、人体工程学元素(如信息布局和放置、隐喻、UI导航)和技术元素(如UI模式、可重用组件)。一般来说，UI是整个应用程序体系结构中唯一的子系统。

外部接口的设计需要关于发送或接收信息的实体的明确信息。在任何情况下，这些信息都应该在需求工程(第5章)中收集，并在接口设计开始时进行验证。外部接口的设计应该包含错误检查和(必要时)适当的安全特性。

内部接口的设计与组件级设计紧密结合(第10章)，分析类的设计实现表示所有操作和在不同类中的操作之间进行通信和协作所需的消息传递方案。每条信息都必须设计以适应必要的信息传递和要求的操作的具体功能要求。如果选择经典的输入-过程-输出方法，每个软件组件的接口都是基于数据流表示和处理描述中描述的功能设计的。

在某些情况下，接口的建模方式与类的方式大致相同。在UML中，接口是按照以下方式定义的[OMG03a]:“接口是对类、组件或其他分类器(包括子系统)的外部可见的(公共)操作的说明符，而不需要规范内部结构。”更简单地说，接口是一组操作，它描述类的某些行为，并提供对这些操作的访问。

例如，SafeHome安全功能使用一个控制面板，允许房主控制安全功能的某些方面。在系统的高级版本中，控制面板功能可以通过无线PDA或移动电话实现。

ControlPanel类(图8.5)提供了与键盘相关的行为，因此，它必须实现操作readKeyStroke()和decodeKey()。如果将这些操作提供给其他类(在本例中是WirelessPDA和MobilePhone)，那么定义一个如图所示的接口是很有用的。这个接口名为KeyPad，它被显示为一个<<接口>>原型，或者作为一个小的、带标记的圆与类连接在一起。接口定义为没有属性和实现键盘行为所必需的操作集。

在其末端的一个开放三角形的虚线(图8.5)表明，ControlPanel类作为其行为的一部分提供了KeyPad操作。在UML中，这被描述为实现。也就是说，ControlPanel的部分行为将通过实现KeyPad操作实现，这些操作将提供给访问接口的其他类。