目录

[第一章 绪论 2](#_Toc384027477)

[1.1 开发背景 2](#_Toc384027478)

[1.2 开发意义 3](#_Toc384027479)

[第二章 系统分析 3](#_Toc384027480)

[2.1 系统的需求分析 3](#_Toc384027481)

[2.2 系统开发设计思想 3](#_Toc384027482)

[2.3系统开发步骤 4](#_Toc384027483)

[2.4 系统的主要技术 4](#_Toc384027484)

[2.4.1 B/S系统的三层体系结构 4](#_Toc384027485)

[2.4.2 jeecg 智能开发平台 5](#_Toc384027486)

[2.5 系统的运行环境和开发平台 5](#_Toc384027487)

[2.5.1 硬件设备及操作系统 5](#_Toc384027488)

[2.5.2 系统开发工具 6](#_Toc384027489)

[2.5.3 开发工具简介 6](#_Toc384027490)

[第三章 系统设计 7](#_Toc384027491)

[3.1系统流程 7](#_Toc384027492)

[3.2 系统功能模块的划分 8](#_Toc384027493)

[3.2.1用户模块 8](#_Toc384027494)

[3.2.3 产品管理 8](#_Toc384027495)

[3.2.4供应商管理 8](#_Toc384027496)

[3.2.5 入库管理 9](#_Toc384027497)

[3.2.6 出货管理 9](#_Toc384027498)

[3.2.7 分部管理 9](#_Toc384027499)

[3.2.8 库存查看 9](#_Toc384027500)

[3.2.9 系统管理 9](#_Toc384027501)

[3.2.10 系统监控 9](#_Toc384027502)

[3.3数据库设计 9](#_Toc384027503)

[3.3.1数据库需求分析 9](#_Toc384027504)

[3.3.2数据库的逻辑设计 10](#_Toc384027505)

[第四章 系统实现 14](#_Toc384027506)

[4.1主要界面实现 14](#_Toc384027507)

[4.1.1用户登陆 14](#_Toc384027508)

[4.1.2 首页 15](#_Toc384027509)

[4.1.3 系统管理-部门管理 15](#_Toc384027510)

[4.1.4系统管理-用户管理 16](#_Toc384027511)

[4.1.5系统管理-角色管理 16](#_Toc384027512)

[4.1.6基础资料-货品信息 17](#_Toc384027513)

[4.1.6基础资料-供货商信息 18](#_Toc384027514)

[4.1.7基础资料-分公司信息 19](#_Toc384027515)

[4.1.8货品出入库-货品入库 19](#_Toc384027516)

[4.1.9货品出入库-货品出库 20](#_Toc384027517)

[4.1.10货品出入库-库存查询 20](#_Toc384027518)

[4.2主要功能程序的实现 21](#_Toc384027519)

[4.2.1 数据库连接的建立 21](#_Toc384027520)

[4.2.2货品添加 22](#_Toc384027521)

[4.2.3货品录入相关页面 25](#_Toc384027522)

[第五章 系统的测试 32](#_Toc384027523)

[5.1 软件测试的目的和原则 32](#_Toc384027524)

[第六章 总结 33](#_Toc384027525)

[参考文献 34](#_Toc384027526)

[**致谢：** 34](#_Toc384027527)

**摘要**

JAVA是前SUN公司（已被甲骨文收购）在九十年代中期推出的一门面向对象的编程语言，目前广泛应用于开发基于WEB的大中型应用系统。本设计综合应用了JAVA web技术的特点，同时结合jeecg智能开发平台和MySQL数据库技术，实现基于互联网的物品进销存管理系统。

【关键字】JAVA , jeecg, MySQL, 进销存

# 第一章 绪论

## 1.1 开发背景

如何吸引和留住人才，激发员工的创造行、工作责任感和工作热情已经成为关系企业兴衰的重要因素，人才已成为企业最重要的资产之一。“公正、公平、合理”的企业管理理念和企业管理水平的提高，是社会对库存管理系统有了更高的需求；同时由于个人电脑的普及，数据库技术、客户/服务器技术，特别是Internet/Intranet技术的发展，使的三代库存管理系统的出现才成为必然。库存管理系统的特点是从物品管理的角度出发，用集中的数据库将几乎所有与物品相关的数据统一管理起来，形成了集成的信息源。有好的用户界面，强有力的报表生成工具、分析工具和信息的共享使得物品管理人员得以摆脱繁重的日常工作，集中精力从战略的角度来考虑企业物品规划和政策。

## 1.2 开发意义

进销存管理系统能确保物畅其流，促使企业经营活动繁荣兴旺。不论什么企业，都要储备一些物资。以生产为主的企业，不储备一定的物资，不能维持其连续生产；服务性行业，也要备置某些需用的设备和服务用具；就连一般的事业单位，也要备有某些办公用品等。因此，各行各业都存在不同程度的库存管理业务。

# 第二章 系统分析

## 2.1 系统的需求分析

根据各大中小企业进销存工作流程与实际的需求和特色，本系统需满足以下几个方面的要求：

1. 提供对商品信息维护的功能
2. 提供供应商维护功能
3. 提供对进货渠道和进货数量的管理功能
4. 提供出货相关内容的维护功能
5. 提供库存查询和相关数据分析功能

## 2.2 系统开发设计思想

系统采用模块化程序设计方法，既便于系统功能的各种组合和修改，又便于未参与开发的技术维护人员补充、维护;

系统应具备数据库维护功能，及时根据用户需求进行数据的添加、删除、修改、备份等操作。

## 2.3系统开发步骤

进销存管理系统的建立与应用可以划分成总体规划、系统开发和系统运行三个阶段，其中系统开发阶段还可进一步分为系统分析、系统设计和系统实施等工作环节。上述各个阶段排列成一个严格的线性开发序列，在每个工作阶段均产生完整的技术文档作为下一阶段工作的指导和依据，每一阶段都应对文档进行评审，确信该阶段工作已完成并达到要求后才能进入下一阶段，同时在以后的工作中不能轻易改变前面经过评审的成果。

上述开发方式的主要优点是便于开发工作的组织和管理，并且可大大降低在线学习系统开发的复杂性。国内外许多系统开发的实例都证明这是一种行之有效的开发方式。

## 2.4 系统的主要技术

### 2.4.1 B/S系统的三层体系结构

在B/S的系统中，用户可以通过浏览器向分布在网络上的许多服务器发出请求。B/S结构极大的简化了客户机的工作，客户机上只需安装、配置少量的客户端软件即可，服务器将担负更多的工作，对数据库的访问和应用程序的执行将在服务器上完成。

在B/S三层体系结构下，表示层(view)、功能层(controller)、数据层(model)被分割成三个相对独立的单元：

第一层--表示层：Web浏览器

在表示层中包含系统的显示逻辑，位于客户端。它的任务是由Web浏览器向网络上的某一Web服务器提出服务请求，Web服务器对用户身份进行验证后用HTTP协议把所需的主页传送给客户端，客户机接受传来的主页文件，并把它显示在Web浏览器上。

第二层--功能层：具有应用程序扩展功能的Web服务器

在功能层中包含系统的事务处理逻辑，位于Web服务器端。它的任务是接受用户的请求，首先需要执行相应的扩展应用程序与数据库进行连接，通过SQL等方式向数据库服务器提出数据处理申请，后台数据库服务器将数据处理的结果提交给Web服务器，再由Web服务器传送回客户端。

第三层--数据层：数据库服务器

在数据层中包含系统的数据处理逻辑，位于数据库服务器端。它的任务是接受Web服务器对数据库操纵的请求，实现对数据库查询、修改、更新、删除功能，把运行结果提交给Web服务器。

从上面的分析中可以看出，B/S体系结构是把C/S结构的事务处理逻辑模块从客户机的任务中分离出来，由单独组成的一层来负担其任务，这样客户机的压力大大减轻了，把负荷均衡地分配给了Web服务器，于是由原来的C/S结构转变成B/S结构。

### 2.4.2 jeecg 智能开发平台

JEECG（J2EE Code Generation）是一款基于代码生成器的智能开发平台。引领新的开发模式(Online Coding模式->代码生成器模式->手工MERGE智能开发)， 可以帮助解决Java项目60%的重复工作，让开发更多关注业务逻辑。既能快速提高开发效率，帮助公司节省人力成本，同时又不失灵活性。

JEECG解决了大量的java开发中的重复性工作，简单功能由Online Coding配置出功能;复杂功能由代码生成器生成进行手工Merge; 复杂流程业务采用表单自定义，业务流程使用工作流来实现、扩展出任务接口，供开发编写业务逻辑。实现了流程任务节点和任务接口的灵活配置，既保证了公司流程的保密性，又减少了开发人员的工作量。

架构技术：Spring MVC+hibernate4+UI快速开发库+Spring JDBC+Highcharts图形报表+jquery+ehcache

**2.5 系统的运行环境和开发平台**

### 2.5.1 硬件设备及操作系统

服务器：inter Xeon e3（或与此相当的CPU）以上配置的高性能PC机，有网络接口卡（NIC），内存应在4g以上，硬盘在1T以上。数据库软件mysql，还可选用备份服务器。

操作系统：OpenSuSE

客户端：i3（或与此相当的CPU）以上配置的PC机，有网络接口卡（NIC），内存应在1g以上，硬盘在80G以上。Windows XP以上操作系统，安装Web浏览器。

网络：服务器和客户端应有网络连通。配置TCP/IP协议。

### 2.5.2 系统开发工具

电脑配置：thinkpad品牌机CPU: i5 内存：DDR 2G 硬盘：150G

操作系统：win 8

开发环境：Eclispse Kepler

Web服务平台：Tomcat 7.0

数据库：mysql 5.5

jdk：7.0

### 2.5.3 开发工具简介

(1) Eclipse概述

Eclipse企业级工作平台，利用它我们可以在数据库和J2EE的开发、发布，以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的J2EE集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持HTML, Struts, JSF, CSS, Javascript, SQL, Hibernate。

(2) Tomcat7.0概述

Tomcat7.0 服务器是一个免费的开放源代码的Web 应用服务器,Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、[Sun](http://baike.baidu.com/view/24856.htm) 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的[Servlet](http://baike.baidu.com/view/25169.htm) 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 7.0 支持最新的Servlet 3.0 和JSP 2.2 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

(3) mysql概述

MySQL是一个[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm)，目前属于[Oracle](http://baike.baidu.com/view/15020.htm)公司。MySQL是一种关联[数据库管理系统](http://baike.baidu.com/view/68446.htm)，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了[灵活](http://baike.baidu.com/view/474022.htm)性。MySQL所使用的SQL语言是用于访问[数据库](http://baike.baidu.com/view/1088.htm)的最常用标准化语言。MySQL软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](http://baike.baidu.com/view/394804.htm)这一特点，一般中小型系统的开发都选择MySQL作为[数据库](http://baike.baidu.com/view/1088.htm)。

# 第三章 系统设计

## 3.1系统流程

与本系统相关的角色包括：

系统管理员：管理系统用户、角色与权限、拥有系统全部权限，保证系统正常运行。

仓库管理员：负责查看库存状态和进行发货流程。

采购员：负责填写出入库单申请。

采购经理：负责对入库单进行审批。



图3.1 系统流程图

## 3.2 系统功能模块的划分

软件所实现的功能强弱是衡量一个软件的最根本的标准。经过对系统的需求分析和实际应用需求，确定了本子系统的功能模块如图3.2所示**：**

图3.2 系统功能模块图

### 3.2.1用户模块

用户包括为用户分配账号，修改密码，用户安全退出功能。此模块只能由系统管理员统一配置，用户不能自行注册，用户密码要进行加密处理。

### 3.2.3 产品管理

对需要入库的产品进行增删改查操作。

### 3.2.4供应商管理

对供应商进行增删改查操作

### 3.2.5 入库管理

先由采购员填写采购单，内容涉及产品相关信息，货源相关信息和数量，然后交由采购经理审批，审批通过后由采购员进行采购，采购完成后由库存管理员进行入库。整个采购流程中共分为4个状态，即：待审核、审核通过、驳回、已完成。

### 3.2.6 出货管理

由采购员填写出货单，内容涉及产品相关信息和数量，由采购经理进行审批，审批通过后出货。整个出货流程中共分为4个状态，即：待审核、审核通过、驳回、已完成。

### 3.2.7 分部管理

管理要分公司的一些基本信息。

### 3.2.8 库存查看

可以查看产品的库存状况，库存数量。

### 3.2.9 系统管理

进行用户角色权限分配，使不同的角色拥有不同的权限从而有效的使用系统进行 工作。

### 3.2.10 系统监控

记录登陆用户的登陆时间、推出时间及在系统内进行的所有操作。

## 3.3数据库设计

### 3.3.1数据库需求分析

数据库的设计在一个系统中的作用十分重要，数据库建立的是否完善直接影响到你个系统的实现。数据库的设计既要满足用户的需求又要尽最大可能的降低数据的冗余，尽可能降低数据间的依赖，将他们分离。在各种信息的提供，保存，更新和查询，这就要求数据库结构能充分满足各种信息的输出和输入，收集基本数据，数据结构，以及数据处理的流程，组成一份详尽的数据字典，为后面的具体设计打下基础．

### 3.3.2数据库的逻辑设计

逻辑结构设计的原则如下：

（1） 尽可能的减少数据冗余和重复。

（2） 结构设计与操作设计相结合。

（3） 数据结构具有相对的稳定性。

（4） 遵循数据库设计三范式。

基于以上设计原则，本系统设计了如下数据库：

全部都使用逻辑删除，为了便于开发，所以不添加任何表关联。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 管理员编号 |
| userName | varchar | 10 | N | N | 用户名 |
| Password | varchar | 100 | N | N | 密码 |
| Realname | Varchar | 50 | N | Y | 真实姓名 |
| Status | Int | 6 | N | Y | 状态 |
| Browser | Vachar | 20 | N | Y | 浏览器 |
| Signature | Blob |  | N | Y | 标志附件 |
| Userkey | Varchar | 200 | N | Y | 秘钥码 |
| Departid | Varchar | 32 | N | Y | 所属部门 |

表1 管理员信息表（T\_S\_BASE\_USER）

表2 供应商表（T\_B\_SUPPLIER）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 供应商编号 |
| SupplierName | varchar | 20 | N | Y | 供应商名称 |
| SupplierAddr | varchar | 26 | N | Y | 供应商地址 |
| SupplierTel | varchar | 15 | N | Y | 供应商电话 |
| SupplierEmail | varchar | 20 | N | Y | 供应商Email |
| SupplierNum | varchar | 20 | N | Y | 企业注册号 |
| ContactPerson | varchar | 10 | N | Y | 联系人 |
| PersonTel | varchar | 15 | N | Y | 联系人电话 |

表3 货品表（T\_B\_GOODS）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 货品编号 |
| GoodsName | varchar | 16 | N | Y | 货品名称 |
| GoodsType | varchar | 2 | N | Y | 货品类型 |
| CostPrice | decimal | 10 | N | Y | 成本价 |
| Remark | varchar | 100 | N | Y | 备注 |

表4 库存表（T\_B\_STOCK）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 库存编号 |
| GoodsId | varchar | 36 | N | Y | 货品编号 |
| TotalCount | varchar | 10 | N | Y | 数量 |

表5 入库单表（T\_B\_PURCHASE）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 入库编号 |
| GoodsId | varchar | 36 | N | Y | 货品编号 |
| GoodsName | varchar | 20 | N | Y | 货品名称 |
| GoodsNumber | varchar | 20 | N | Y | 货品批号 |
| GoodsCount | varchar | 10 | N | Y | 数量 |
| Price | decimal | 10 | N | Y | 单价 |
| Status | Char | 1 | N | Y | 状态 |
| SupplierId | varchar | 36 | N | Y | 供应商编号 |
| SupplierName | varchar | 20 | N | Y | 供应商名称 |
| ContactPerson | varchar | 10 | N | Y | 联系人 |
| PersonTel | varchar | 15 | N | Y | 联系人电话 |
| Remark | varchar | 100 | N | Y | 备注 |

表6 出库单表（T\_B\_SHIP）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 出库编号 |
| GoodsId | varchar | 36 | N | Y | 货品编号 |
| GoodsName | varchar | 20 | N | Y | 货品名称 |
| GoodsCount | varchar | 10 | N | Y | 数量 |
| Status | char | 1 | N | Y | 状态 |
| BranchId | varchar | 36 | N | Y | 分部编号 |
| BranchName | varchar | 20 | N | Y | 分部名称 |
| ShipDate | date | 10 | N | Y | 出库日期 |
| Remark | varchar | 100 | N | Y | 备注 |

表7分公司表（T\_B\_BRANCH）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | varchar | 36 | Y | N | 分公司编号 |
| BranchAdd | varchar | 20 | N | Y | 分公司地址 |
| BranchTel | varchar | 13 | N | Y | 分公司电话 |
| LeaderName | varchar | 10 | N | Y | 负责人姓名 |
| LeaderTel | varchar | 13 | N | Y | 负责人电话 |
| ZipCode | varchar | 8 | N | Y | 邮编 |

表8 部门表（T\_S\_DEPART）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | Varchar | 36 | Y | N | 部门编号 |
| Departname | Varchar | 100 | N | Y | 部门名称 |
| Description | Text |  | N | Y | 描述 |
| Parentdepartid | Varchar | 36 | N | Y | 上级部门编号 |

表9 日志表（T\_S\_LOG）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| id | Varchar | 36 | Y | N | 日志编号 |
| Browser | Varchar | 100 | N | Y | 浏览器 |
| Logcontent | Text |  | N | Y | 日志内容 |
| Loglevel | Int | 6 | N | Y | 日志级别 |
| Note | Text |  | N | Y | 登陆ip |
| operatetime | Date |  | N | Y | 操作时间 |
| operatetype | Int | 6 | N | Y | 操作类型 |
| Userid | Varchar | 36 | N | Y | 用户编号 |

表10 角色表（T\_S\_ROLE）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | Varchar | 36 | Y | N | 角色编号 |
| Rolecode | Varchar | 10 | N | Y | 角色码 |
| Rolename | Varchar | 100 | N | Y | 角色名称 |

表11 用户角色表（T\_S\_ROLE\_USER）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | Varchar | 36 | Y | N | 用户角色编号 |
| Roleid | Varchar | 36 | N | Y | 角色编号 |
| userid | Varchar | 36 | N | Y | 用户编号 |

表12 用户功能表（T\_S\_ROLE\_FUNCTION）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 是否空 | 说明 |
| Id | Varchar | 36 | Y | N | 用户功能编号 |
| operation | Varcahr | 100 | N | Y | 操作 |
| roleid | Varchar | 36 | N | Y | 角色编号 |

更多内容请具体查看init.sql文件

# 第四章 系统实现

## 4.1主要界面实现

本系统的UI设计主要采用easyUI，使得界面更加大气美观。

### 4.1.1用户登陆

输入正确的用户名、密码和验证码后进行登陆。



### 4.1.2 首页

登陆成功后进入首页面

****

图4.1首页效果

### 4.1.3 系统管理-部门管理

通过部门管理，对部门进行增删改查操作

****

图4.2部门管理

### 4.1.4系统管理-用户管理

在用户管理下面对用户进行增删改查

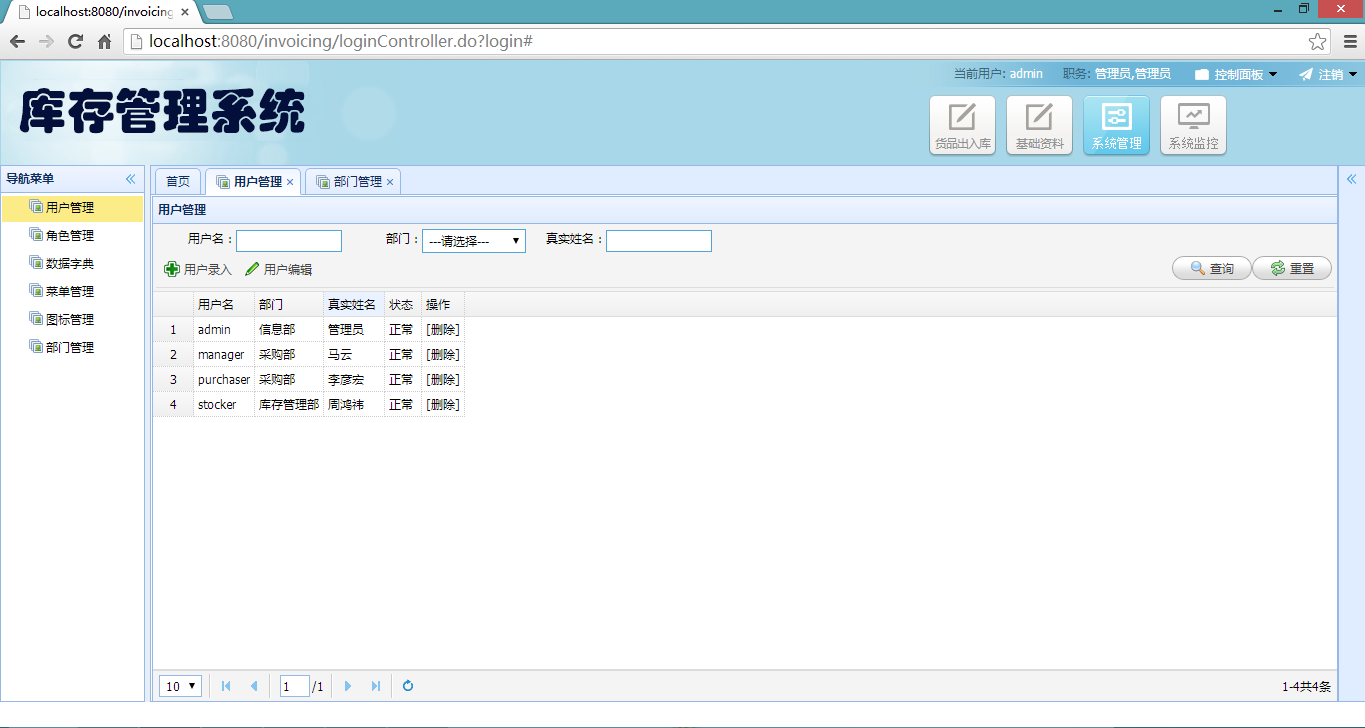
****

图4.3 用户管理

### 4.1.5系统管理-角色管理

进入角色管理模块，添加角色，库存管理员、采购员、采购经理。然后为其分配相应的页面和按钮访问权限。

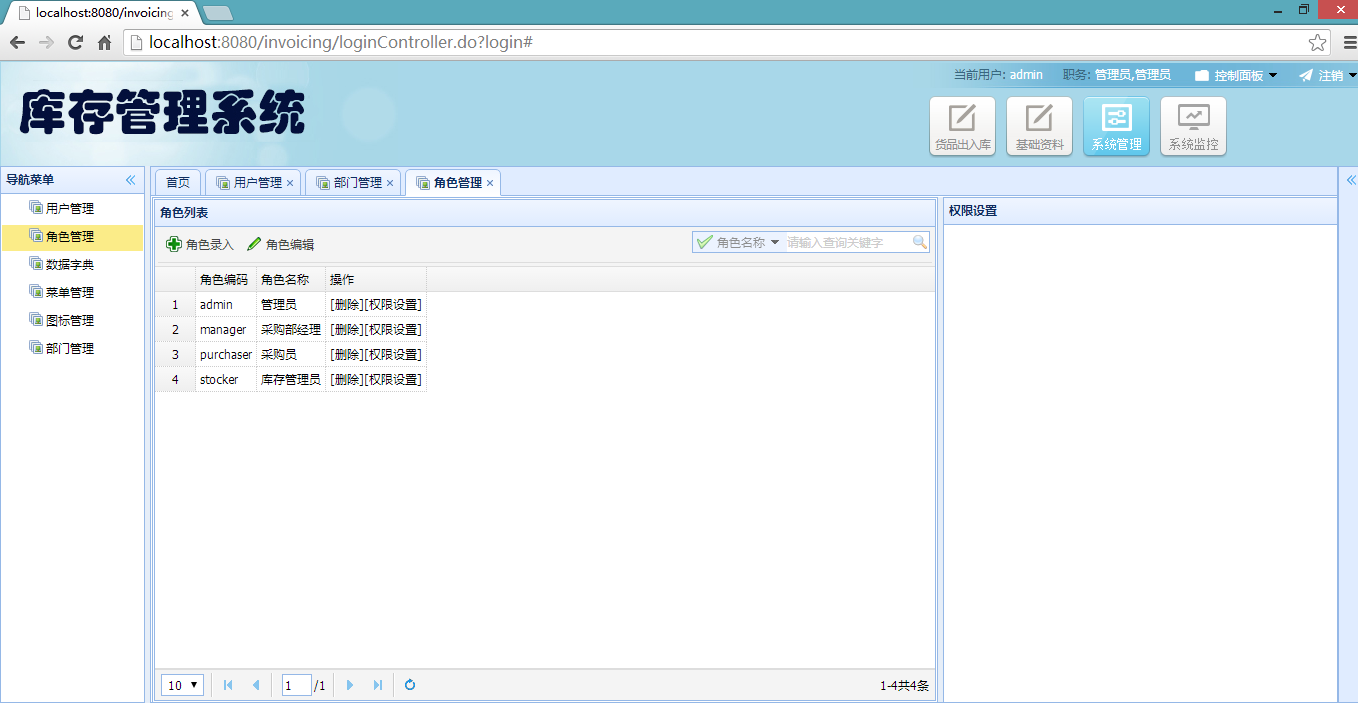


图4.4 角色管理

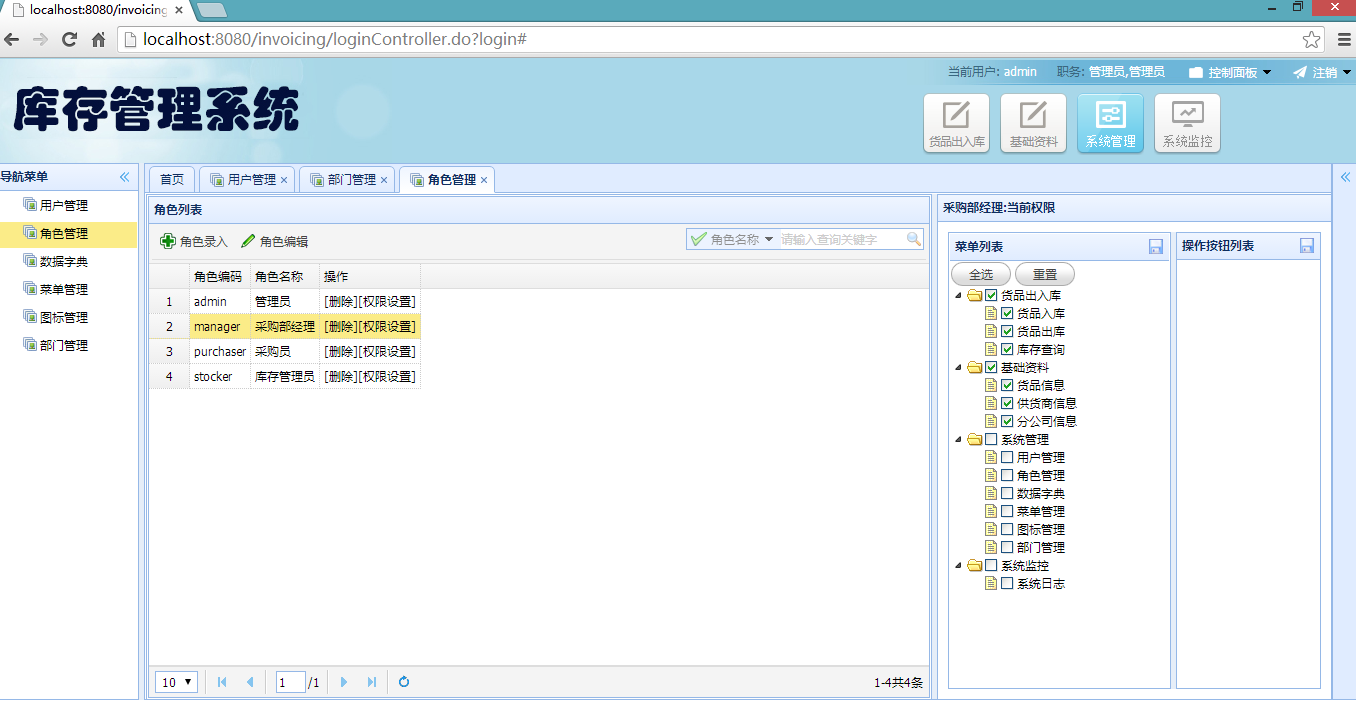
****

图4.5角色管理分配权限

### 4.1.6基础资料-货品信息

添加货品信息，并对其进行维护。

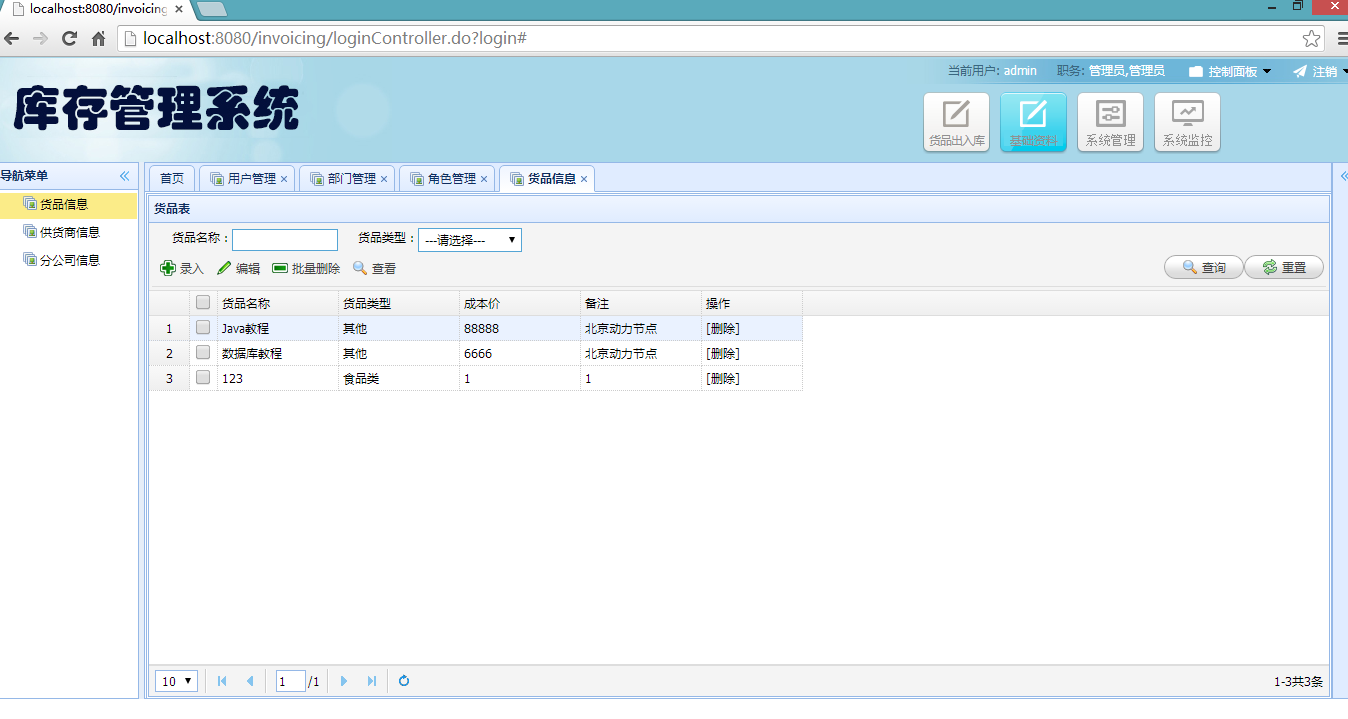


图4.6货品信息

### 4.1.6基础资料-供货商信息

添加供货商信息，并对其进行维护。

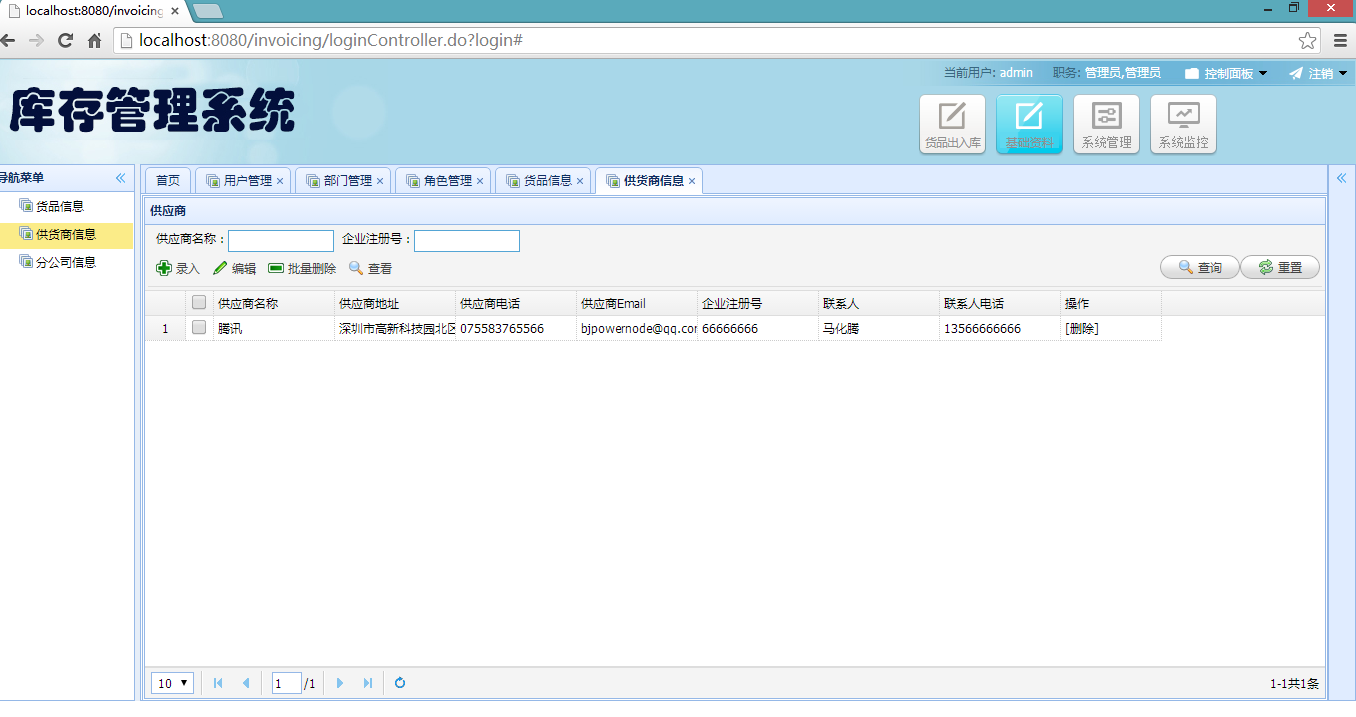


图4.7供货商信息

### 4.1.7基础资料-分公司信息

添加分公司信息，并对其进行维护。

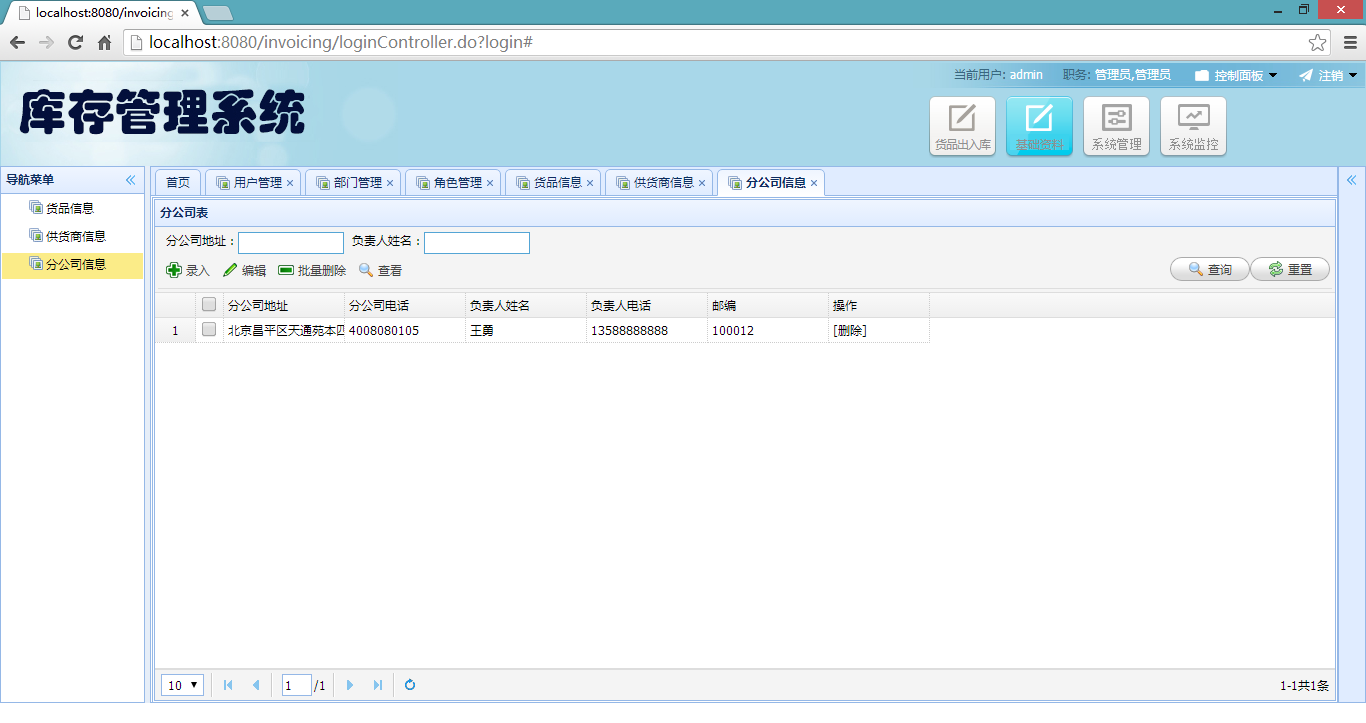


图4.8分公司信息

### 4.1.8货品出入库-货品入库

添加货品入库信息，并对其进行维护。

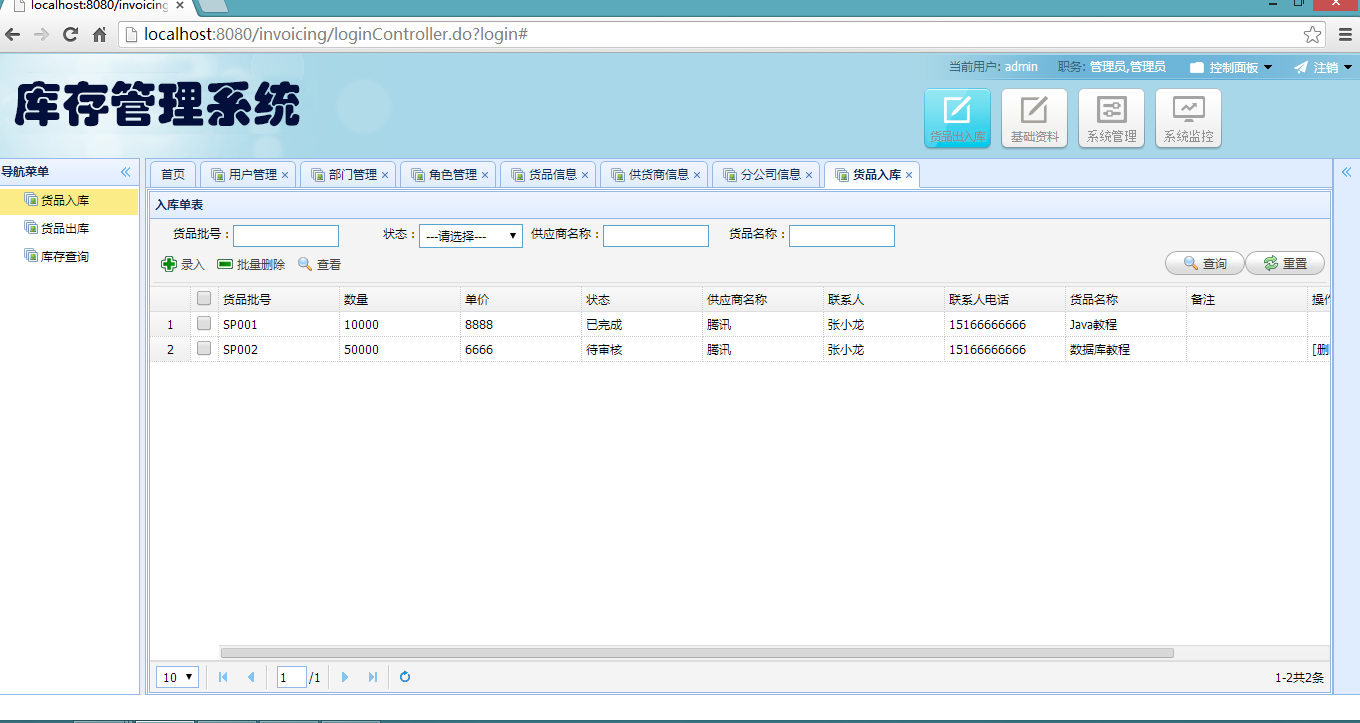


图4.9货品入库信息

### 4.1.9货品出入库-货品出库

添加货品入出信息，并对其进行维护。

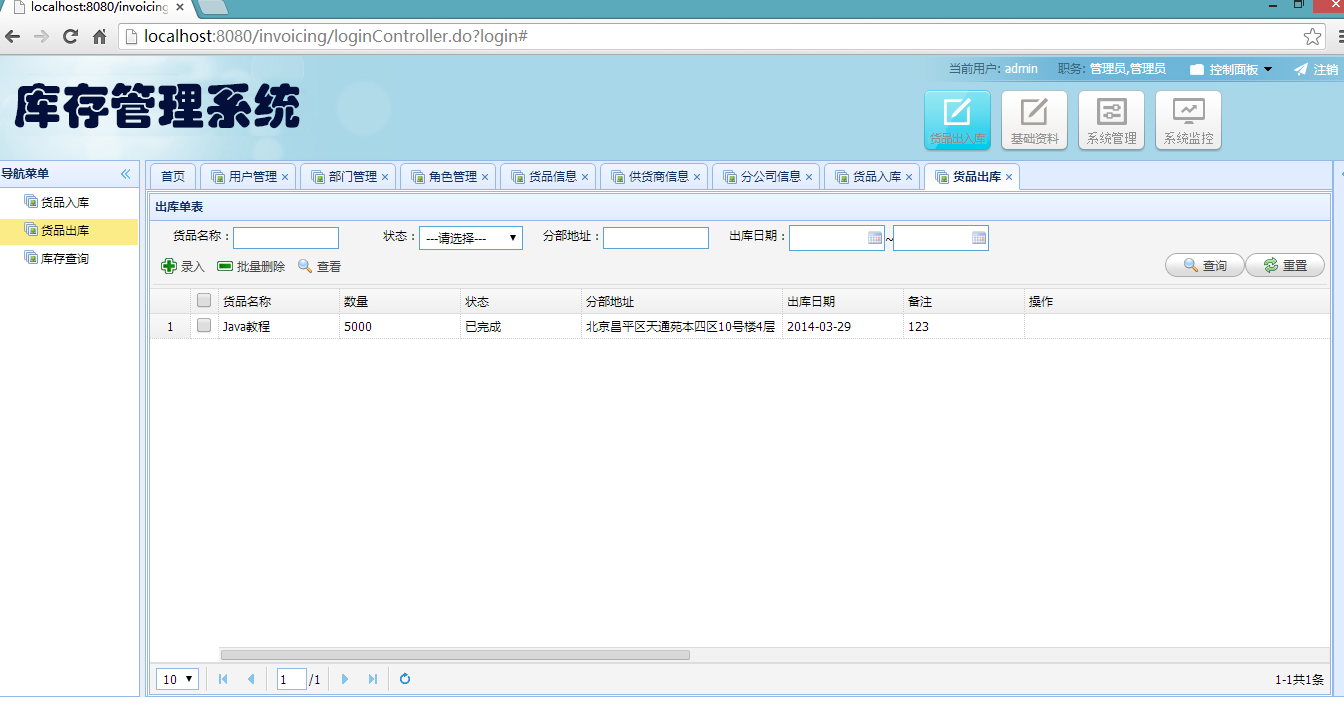


图4.10货品出库信息

### 4.1.10货品出入库-库存查询

查看库存信息

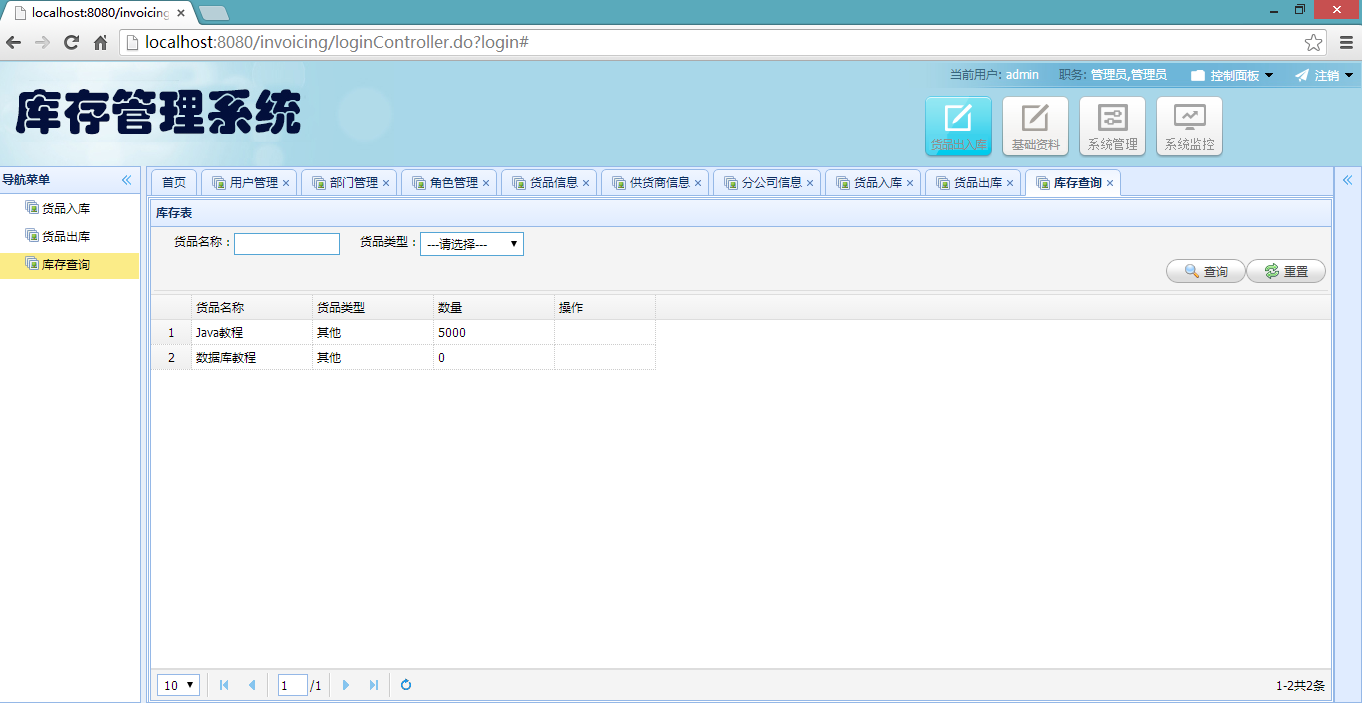


图4.11查看库存信息

## 4.2主要功能程序的实现

### 4.2.1 数据库连接的建立

本系统在数据库连接方面，使用统一的连接数据库的方法，引入相关的jar包之后，调用其中的类。通过dbconfig.properties配置文件读取数据库信息和登陆数据库帐户的用户名和密码，实现数据库的连接。具体的代码如下：

dbconfig.properties文件的内容如下：

|  |
| --- |
| hibernate.dialect=org.hibernate.dialect.MySQLDialect  validationQuery.sqlserver=SELECT 1  jdbc.url.jeecg=jdbc:mysql://localhost:3306/invoicing?useUnicode=true&characterEncoding=UTF-8  jdbc.username.jeecg=root  jdbc.password.jeecg=bjpowernode  jdbc.dbType=mysql |

Hibernate相关配置：

|  |
| --- |
| <property name=*"url"* value=*"${jdbc.url.jeecg}"* />  <property name=*"username"* value=*"${jdbc.username.jeecg}"* />  <property name=*"password"* value=*"${jdbc.password.jeecg}"* /> |

### 4.2.2货品添加

当用户打货品页面将货品信息填写好，点击“提交”按钮之后，在客户端浏览器首先对用户填入的信息进行校验，校验主要包括：用户在必填的选项下是否输入空的内容，是否符合该项所定义的格式要求。如不符合要求，该表单的不会被提交到服务器端去，这样有助于减轻服务器端的压力。

当用户填写的信息正确之后，页面由表单的Action属性确定将用户的请求发送到服务器下的TBGoodsController这个类下进行处理，当服务器获取到请求之后调用类TBGoodsController的doAdd方法,在doAdd方法中国调用tBGoodsService类中的save方法，在save方法中调用了CommonServiceImpl类中的save方法，然后将数据插入到数据库中。具体处理过程如下：

|  |
| --- |
| /\*\*  \* 添加货品表  \*  \* **@param** ids  \* **@return**  \*/  @RequestMapping(params = "doAdd")  @ResponseBody  **public** AjaxJson doAdd(TBGoodsEntity tBGoods,TBStockEntity tBStock, HttpServletRequest request) {  AjaxJson j = **new** AjaxJson();  message = "货品表添加成功";  **try**{  tBStock.setTotalcount("0");  tBGoodsService.save(tBGoods, tBStock);  systemService.addLog(message, Globals.*Log\_Type\_INSERT*, Globals.*Log\_Leavel\_INFO*);  }**catch**(Exception e){  e.printStackTrace();  message = "货品表添加失败";  **throw** **new** BusinessException(e.getMessage());  }  j.setMsg(message);  **return** j;  } |

|  |
| --- |
| **public** <T> Serializable save(T entity, TBStockEntity tBStock) {  Serializable t = **super**.save(entity);  tBStock.setGoodsid(t.toString());  **super**.save(tBStock);  //执行新增操作配置的sql增强  **this**.doAddSql((TBGoodsEntity)entity);  **return** t;  } |

|  |
| --- |
| /\*\*  \* 根据传入的实体持久化对象  \*/  **public** <T> Serializable save(T entity) {  **try** {  Serializable id = getSession().save(entity);  getSession().flush();  **if** (*logger*.isDebugEnabled()) {  *logger*.debug("保存实体成功," + entity.getClass().getName());  }  **return** id;  } **catch** (RuntimeException e) {  *logger*.error("保存实体异常", e);  **throw** e;  }  } |

保存成功之后将新添加的数据以不刷新页面的效果展示到页面中，并且提示保存成功，如果保存失败则提示失败信息。

### 4.2.3货品录入相关页面

货品内容在展示页面代码：

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>  <%@include file=*"/context/mytags.jsp"*%>  <div class=*"easyui-layout"* fit=*"true"*>  <div region=*"center"* style="padding:*1px*;">  <t:datagrid name=*"tBGoodsList"* checkbox=*"true"* fitColumns=*"false"* title=*"货品表"* actionUrl=*"tBGoodsController.do?datagrid"* idField=*"id"* fit=*"true"* queryMode=*"group"*>  <t:dgCol title=*"主键"* field=*"id"* hidden=*"false"* queryMode=*"single"* width=*"120"*></t:dgCol>  <t:dgCol title=*"货品名称"* field=*"goodsname"* hidden=*"true"* query=*"true"* queryMode=*"single"* width=*"120"*></t:dgCol>  <t:dgCol title=*"货品类型"* field=*"goodstype"* hidden=*"true"* dictionary=*"goodsType"* query=*"true"* queryMode=*"single"* width=*"120"*></t:dgCol>  <t:dgCol title=*"成本价"* field=*"costprice"* hidden=*"true"* queryMode=*"single"* width=*"120"*></t:dgCol>  <t:dgCol title=*"备注"* field=*"remark"* hidden=*"true"* queryMode=*"single"* width=*"120"*></t:dgCol>  <t:dgCol title=*"操作"* field=*"opt"* width=*"100"*></t:dgCol>  <t:dgDelOpt title=*"删除"* url=*"tBGoodsController.do?doDel&id={id}"* />  <t:dgToolBar title=*"录入"* icon=*"icon-add"* url=*"tBGoodsController.do?goAdd"* funname=*"add"*></t:dgToolBar>  <t:dgToolBar title=*"编辑"* icon=*"icon-edit"* url=*"tBGoodsController.do?goUpdate"* funname=*"update"*></t:dgToolBar>  <t:dgToolBar title=*"批量删除"* icon=*"icon-remove"* url=*"tBGoodsController.do?doBatchDel"* funname=*"deleteALLSelect"*></t:dgToolBar>  <t:dgToolBar title=*"查看"* icon=*"icon-search"* url=*"tBGoodsController.do?goUpdate"* funname=*"detail"*></t:dgToolBar>  </t:datagrid>  </div>  </div>  <script src = *"webpage/buss/goods/tBGoodsList.js"*></script> |

货品添加页面代码：

|  |
| --- |
| <%@ page language=*"java"* import=*"java.util.\*"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>  <%@include file=*"/context/mytags.jsp"*%>  <!DOCTYPE html>  <html>  <head>  <title>货品表</title>  <t:base type=*"jquery,easyui,tools,DatePicker"*></t:base>  <script type=*"text/javascript"* src=*"plug-in/ckeditor\_new/ckeditor.js"*></script>  <script type=*"text/javascript"* src=*"plug-in/ckfinder/ckfinder.js"*></script>  </head>  <body>  <t:formvalid formid=*"formobj"* dialog=*"true"* usePlugin=*"password"* layout=*"table"* action=*"tBGoodsController.do?doAdd"* tiptype=*"4"*>  <input id=*"id"* name=*"id"* type=*"hidden"* value=*"*${tBGoodsPage.id }*"* />  <table style="width: *600px*;" cellpadding=*"0"* cellspacing=*"1"* class=*"formtable"*>  <tr>  <td align=*"right"*>  <label class=*"Validform\_label"*>  货品名称:  </label>  </td>  <td class=*"value"*>  <input id=*"goodsname"* name=*"goodsname"* type=*"text"* style="width: *150px*" datatype=*"\*1-16"* class=*"inputxt"* />  <span class=*"Validform\_checktip"*></span>  <label class=*"Validform\_label"* style="display: *none*;">货品名称</label>  </td>  </tr>  <tr>  <td align=*"right"*>  <label class=*"Validform\_label"*>  货品类型:  </label>  </td>  <td class=*"value"*>  <t:dictSelect field=*"goodstype"* type=*"list"*  typeGroupCode=*"goodsType"* defaultVal=*"*${tBGoodsPage.goodstype}*"* hasLabel=*"false"* title=*"货品类型"*></t:dictSelect>  <span class=*"Validform\_checktip"*></span>  <label class=*"Validform\_label"* style="display: *none*;">货品类型</label>  </td>  </tr>  <tr>  <td align=*"right"*>  <label class=*"Validform\_label"*>  成本价:  </label>  </td>  <td class=*"value"*>  <input id=*"costprice"* name=*"costprice"* type=*"text"* datatype=*"n1-10"* style="width: *150px*" class=*"inputxt"*    >  <span class=*"Validform\_checktip"*></span>  <label class=*"Validform\_label"* style="display: *none*;">成本价</label>  </td>  </tr>  <tr>  <td align=*"right"*>  <label class=*"Validform\_label"*>  备注:  </label>  </td>  <td class=*"value"*>  <textarea id=*"remark"* name=*"remark"* maxlength=*"100"*></textarea>  <span class=*"Validform\_checktip"*></span>  <label class=*"Validform\_label"* style="display: *none*;">备注</label>  </td>  </tr>  </table>  </t:formvalid>  </body>  <script src = *"webpage/buss/goods/tBGoods.js"*></script> |

# 第五章 系统的测试

## 5.1 软件测试的目的和原则

在开发大型软件系统的过程中，需要面对错综复杂的问题，因此，在软件生存周期的每个阶段都不可避免地会产生错误。我们力求在每个阶段结束之前通过严格的技术审查，尽可能早的发现并纠正错误。测试的目的就是在软件投入生产性运行之前，尽可能多地发现软件中的错误。目前软件测试仍然是保证软件质量的关键步骤，它是对软件规格说明、设计和编码的最后复审。

尽管面向对象技术的基本思想保证了软件应该有更高的质量，但无论采用什么样的编程技术，编程人员的错误都是不可避免的，而且由于面向对象技术开发的软件代码重用率高，更需要严格测试，避免错误的繁衍。因此，软件测试在面向对象编程里更具有它的重要性。

与开发过程类似，测试过程也必须分步骤进行，每个步骤在逻辑上是前一个步骤地继续。大型软件系统通常由若干个子系统组成，每个子系统又由许多模块组成。因此，大型软件系统的测试的基本层次如图6-1所示[12]。



图6-1 测试的基本层次

(1) 单元测试 单元测试的用例从单元详细设计中导出。在单元测试中可以采用功能性测试和结构性测试两种。

(2) 集成测试 在这个测试步骤中所发现的往往是概要设计的错误。

(3) 系统测试 在这个测试步骤中所发现的往往是需求规格说明的错误。一般来说，系统测试是功能性测试，不是结构性测试。

# 第六章 总结

通过一个多月的系统开发，在线学习系统完成。在系统的开发过程中，以前学习感到很抽象的课程，如软件工程、数据库原理、面向对象程序设等变得清晰起来。感觉到这几门理论课程的重要性。

以前做一些应用的系统开发，随意性大，想到那里做到那里，没有统一的规划，一旦系统成型，出现错误返工修改工作量大，最终的系统可靠性不高。另外一直以来，进行程序设计均采用结构化开发方法，对系统的全貌难以在开发前看到，导致实现的系统不满足需求。

这次的毕业设计开发，主要有以下几点收获：

1．学习到了很多Java开发中需要的技术。

2．通过实践，提高了系统分析的能力，对数据流图、数据字典、系统流程图等系统设计工具的使用有了更深刻的认识。为以后的工作、学习的进一步提高打下了坚实的基础。

# 参考文献

[1] [美]Todd Cook. JSP从入门到精通[M].北京：电子工业出版社，2003. 1-385.

[2] Soren Lauesen.Software Requirements Styles and Techniques[M]. BeiJing: Publishing House of Electronics Industry,2002.10,25-90.

[3] Roger S.Pressman.Software Engineering A Practitioner’s Approach, 5th ed[M].北京：清华大学出版社，2001.1-589.

[4] Hans Bergsten. JavaServer Pages\_2nd Edition[M]. O'Reilly, August 2002，1-500.

[5] Kevin Duffey,Vikram Goyal. Professional JSP Site Design[M]. BeiJing: Publishing House of Electronics Industry,2002.1-75.

[6] Paul C.Jorgensen.Software Testing A Craftsman’s Approach(second Edition)[M].China Machine Press,2003,170-175.

**致谢：**

感谢学校以认真负责的态度为我们组织这次毕业设计以及毕业论文答辨。你们的严格把关，对我们来说有很大的压力。在这种压力下促使我们以最快的速度融入到软件开发理论的应用中去，通过这次煅炼，使我们的系统开发水平各方面都有了较大的进步。由此积累的经验将对我们以后的学习和生活产生帮助。

感谢我的指导老师，在毕业设计时帮助选题，在设计的过程中提出了许多有益的建议和意见，同时他先进的软件开发理念和思维方式也给我很大的启发。