

毕业论文

基于SSH的小型企业商品销售管理系统

的设计与实现

|  |  |
| --- | --- |
| 院 别 | 经贸学院 |
| 专业名称 | 信息管理与信息系统 |
| 班级学号 | 2110201 |
| 学生姓名 | 张帅 |
| 指导教师 | 袁静波 |

2015年6月10日

基于SSH的小型企业商品销售管理信息系统的设计与实现

摘 要

随着信息经济时代的来临和经济全球化的进一步加快，企业的信息化水平己逐渐成为新经济时代衡量一个国家和地区综合实力的重要标志。现今，在我国许多大型企业和各种大型组织都已经迈入信息化阶段，而我国小企业大部分还在摸索阶段。大部分几乎还处于完全人工化管理阶段，其经营模式落后，管理水平差，这大大制约了我国小型企业的发展。商品销售管理系统可以明显改善小型企业的信息化水平，提高其管理水平和管理效率，提高企业的竞争力，有利于企业的长期发展。

根据小型企业信息化程度，提出了商品销售管理系统。对SSH框架及其相关技术、开发方式、开发环境进行了介绍，然后对商品管理系统进行了详细的系统分析和系统设计。本系统主要利用MyEclipse和MySQL两种工具，采用SSH框架，进行系统设计与实现。系统的主要模块有销售管理、客户与供应商管理、仓库管理、配送管理和会计管理，其中每一个模块都对应着小型企业的一个职能，实现了商品销售管理系统的系统目标。

关键词：商品销售管理系统，SSH，MyEclipse，MySQL

**The Design and Implementation of small Enterprises Commodity Sales Management Information System based SSH**

Author: Zhang Shuai

Tutor: Yuan Jingbo

Abstract

With the advent of the era of information economy and economic globalization further speeding up, the enterprise information level has become the new economic era to measure a country and region an important symbol of comprehensive strength. Nowadays, many big companies and all kinds of large organizations in our country has already entered information stage, and most of the small business is still in groping stage in our country. Most almost completely artificialization management is still in the stage, its management pattern backwardness, poor management level, which greatly restricted the development of small enterprises in our country. Merchandise sales management system can significantly improve the information level of small business, to improve the management level and management efficiency, improve the competitiveness of the enterprises, and is conducive to the long-term development of the enterprise.

According to small business informatization degree, commodity sales management system is proposed. On SSH framework and its related technology, development approach, development environment are introduced, then the merchandise management system has carried on the detailed system analysis and system design. This system mainly use two tools MyEclipse and MySQL, adopt the SSH framework, system design and implementation. Major modules of the system has the sales management, customer and supplier management, warehouse management, distribution management and accounting management, in which each module corresponds to a small business functions, to achieve the goal merchandise sales management system of the system.

**Key words:** Merchandies Sales Managermnt System,SSH,MyEclipse,MySQL

目 录

[1 绪论 1](#_Toc422490012)

[1.1 研究背景和研究意义 1](#_Toc422490013)

[1.2 国内外研究现状 1](#_Toc422490014)

[1.2.1国外研究现状 1](#_Toc422490015)

[1.2.2国内研究现状 2](#_Toc422490016)

[1.3 论文的设计方法与内容 2](#_Toc422490017)

[1.3 论文的组织结构 3](#_Toc422490018)

[2 相关技术介绍 4](#_Toc422490019)

[2.1 SSH框架简介 4](#_Toc422490020)

[2.2 SSH的相关技术 4](#_Toc422490021)

[2.3 SSH的开发方式 5](#_Toc422490022)

[2.4 SSH的开发环境 6](#_Toc422490023)

[2.5 本章小结 7](#_Toc422490024)

[3 系统分析 8](#_Toc422490025)

[3.1 系统目标 8](#_Toc422490026)

[3.2 业务流程分析 8](#_Toc422490027)

[3.3 功能需求分析 12](#_Toc422490028)

[3.4 用例图 13](#_Toc422490029)

[3.5 系统性能需求分析 16](#_Toc422490030)

[3.6 可行性分析 18](#_Toc422490031)

[3.7 本章小结 19](#_Toc422490032)

[4 系统设计 21](#_Toc422490033)

[4.1 功能结构图 21](#_Toc422490034)

[4.2 数据库设计 22](#_Toc422490035)

[4.2.1 数据库设计原则 22](#_Toc422490036)

[4.2.2 主题数据库 23](#_Toc422490037)

[4.2.3 数据库概念设计 24](#_Toc422490038)

[4.2.4 数据库逻辑结构设计 26](#_Toc422490039)

[4.3 本章小结 31](#_Toc422490040)

[5 系统实现与测试 32](#_Toc422490041)

[5.1 商品销售管理系统的实现 32](#_Toc422490042)

[5.1.1 员工登录界面 32](#_Toc422490043)

[5.1.2 root用户界面 34](#_Toc422490044)

[5.1.3 销售管理界面 35](#_Toc422490045)

[5.1.4 仓库管理 37](#_Toc422490046)

[5.1.5 会计管理 38](#_Toc422490047)

[5.1.6 客户和供应商管理 38](#_Toc422490048)

[5.1.7 配送管理 38](#_Toc422490049)

[5.2 商品销售管理系统的测试 40](#_Toc422490050)

[5.2.1 测试目的 40](#_Toc422490051)

[5.2.2 测试环境 40](#_Toc422490052)

[5.2.3 测试用例及结果分析 41](#_Toc422490053)

[5.3 核心代码 42](#_Toc422490054)

[5.4 本章小结 43](#_Toc422490055)

[结 论 44](#_Toc422490056)

[致 谢 45](#_Toc422490057)

[参考文献 46](#_Toc422490058)

[附 录 48](#_Toc422490059)

[附录A 48](#_Toc422490060)

[附录B 55](#_Toc422490061)

# 1 绪论

## 1.1 研究背景和研究意义

当今世界，随着经济全球化的加快，各行各业都在迅速发展，在不断的淘汰旧事物，也在不断地产生新事物，用“适者生存”来形容是最适合不过了。信息已经成为企业的一种重要的战略资源，在信息技术的推动下，现代社会正经历一个世界经济秩序和结构打破和重组的过程，这促使世界经济由物质主导型转变为信息主导型。现代企业既面临着旧的竞争秩序被迅速打破而带来的快速发展的机遇，又面临着信息经济和全球化所带来的威胁[1]。信息时代已经到来，大大加速了事物的发展与淘汰的速度，信息化程度已经成为企业的产生与淘汰的重要不容忽视因素[2]。一个国家的社会信息化程度 ,在很大程度上影响到人民生活的质量。我国是中小型企业的天下，其每年创造的GDP不容忽视，中小型企业在我国的经济发展中起着重要作用，而小型企业又是其中数量最大的一部分，所以提高小型企业的经济效率和经营模式，会改善我国的经济水平。在我国许多小型企业还采用旧的经营模式，所有工作均是由人来完成，这大大限制了小型企业的发展，所以改进小型企业的经营和管理模式迫在眉睫，也是大部分小型企业崛起的必由之路。改进则进步，不改则淘汰，只有跟上时代的进步了才会有发展的机会。

## 1.2 国内外研究现状

### 1.2.1国外研究现状

国外发达国家内小企业信息化建设都已经比较成熟，已经遍布几乎所有领域，几乎所有的公司都采用现代信息化管理。在美、日、英等国家，其小企业已经几乎完全依靠现代系统。美国是世界上电子信息技术和高科技创新的发源地，而且美国政府采取了很多措施支持中小企业发展信息化，例如，增加对小企业采购的比例。英国在上世界90年代末就在支持信息化建设，重点是中小企业，有绝大部分企业实施了电子商务。日本，一直重视企业信息系统的网络化，目前日本已经不再局限为建立信息系统，而是通过信息系统结合其他手段更好地进行信息资源管理[3-4]。

### 1.2.2国内研究现状

目前，我国小企业的工作还几乎完全依靠人力，几乎所有的工作都是由人完成，虽然所有的企业都有电脑，但是大部分企业仅仅是用于存储，只是会利用Microsoft Excel和Word等等简单的软件，还远远没有达到合理利用电脑的程度，不能及时反映所需的信息，尤其是从全局角度考虑，各个部门缺乏合理分配与合作。如果采用商品信息管理系统，就可以及时查询所需信息。例如，在仓库管理方面，可以及时查询商品的数量、价格、位置等等，可以及时采购数量较少的商品，还可以优化配送。在大企业采用物流信息管理系统，能够及时反映各种信息，对人员、车辆、商品进行合理的调度，涉及到仓库存储管理、运输和配送管理、财务管理、人力资源管理等等内容，通过计算机技术、通信技术、网络技术等手段，建立的物流信息化管理使物流信息的处理速度和传递速度大大提高，大大提高物流活动的效率和快速反应能力。中小企业是与所处行业的大企业相比人员规模、资产规模和经营规模都比较小的经济单位[5]。小企业虽然不能向大企业那样，做到完全信息化管理，但是仍有很大的提升空间。

小型企业由于数量巨大，但又资金有限，很难有统一的适合大部分企业使用的管理模式，因此针对各个企业开发自己的管理模式，应用属于自己的管理方法，才是正道，但是仍然有很多企业有很多共通之处。例如，几乎所有的企业都有自己的客户、供应商，大部分企业都有自己的产品和仓库和配送渠道，都有自己的财务管理，因此商品销售管理系统仍有很大的应用空间。

本次系统设计针对的是销售型的小型企业，主要业务一是面向企业型的大客户，由企业向公司发送的订货单，再由公司自己完成备货、送货等业务，这类业务订单数目较少，商品数量大，资金总额大，耗费时间长；二是面向个人的实体店部分，属于日常销售，有顾客直接上门，所有活动均是现场完成。这种业务，日常客户数量多，交易量小，但是交易总额也不容忽视。

## 1.3 论文的设计方法与内容

1、设计方法

本文依据软件工程方法进行研究设计，完成系统需求分析、系统设计、系统实现和系统测试等内容。通过阅读相关文献及实际调研，对商品销售管理系统进行业务流程分析及功能需求分析，并采用面向对象的方法进行用例图设计。在需求分析的基础上，进行系统功能结构设计及数据库设计，并基于SSH(Struts2+Spring+Hibernate)框架，使用MyEclipse和MySQL两种开发工具，数据库使用JDBC连接，采用JAVA语言、JSP技术、B/S模式对系统进行设计及编程实现。

2、研究内容

商品销售管理系统主要包括客户信息管理、供应商信息管理、销售管理、仓库管理、配送管理、会计管理等主要功能模块，其中每个模块都提供了最基本的查询、更新、删除、添加等功能。由于不同的模块由不同的人员完成，因此不同的登录用户，也有不同的权限。仓库管理和配送管理还可以记录并统计企业商品的出入库，及时准确地反映库存状况，而销售管理更能统计每种商品的销售数量与季度等的变化情况，为企业决策提供数据支持。

## 1.3 论文的组织结构

本论文共有五章，各章内容如下所述：

第1章绪论。主要介绍了商品管理系统的开发背景以及意义，即小型企业旧式管理模式到新式模式的转变，分析了信息化的优势；然后介绍了论文的主要研究内容、研究方法，最后是文章的组织结构。

第2章相关技术简介。介绍了SSH的框架简介、SSH框架的相关技术、开发方式以及开发环境。

第3章系统分析。具体是对系统目标说明；功能需求分析,并对系统各个模块做了详细的介绍；性能需求分析，从各个角度对系统作了评估，还进行了可行性分析，分析系统的可行性。然后给出了本系统的运行平台，最后是对本章内容的总结。。

第4章系统设计。主要就本系统的开发模式进行说明，接着进行功能设计和数据库设计，最后是本章小结。

第5章商品销售管理系统的实现与测试，还简单介绍了部分关键代码。主要是对系统界面的实现和对系统功能关键编码的实现部分进行了详细的介绍，还对系统测试作了介绍，最后对本章作了总结。

# 2 相关技术介绍

## 2.1 SSH框架简介

本文使用MyEclipse开发软件和MySQL数据库，基于SSH框架，进行系统设计与实现。下面对所需要的相关技术进行介绍。

SSH框架，是利用3种技术（Struts2+Spring+Hibernate）结合而成的一种非常流行的web应用程序开源框架。因为采用的是Struts2技术，所以又叫做S2SH框架。SSH框架分为3层：表示层、业务逻辑层、数据持久层，其中表示层一般是JSP页面；业务逻辑层，可以添加一些判定方法，用于连接Action控制器和DAO层；数据持久层指的是DAO层。SSH框架，能够使编程人员能够在短时间内构建出结构非常清晰、各层之间的耦合度低、可复用性好、维护非常方便的web项目应用程序[6-7]，其中Struts2是系统的整体基础架构，可以在struts的配置文件，控制业务跳转，即由JSP到Action控制器的跳转；利用Hibernate框架对持久层提供支持，spring通过Spring的依赖注入机制去管理Struts和Hibernate。简而言之，以Struts来实现系统的表示层,以Hibernate实现将数据持久化到数据库,以Spring来实现系统的业务逻辑层[6]。下面分别对Struts2、Spring、Hiberante进行相关介绍。

## 2.2 SSH的相关技术

1、Struts2简介

Struts2是Struts的第二代产品，是在Struts1和WebWork的技术基础上进行了合并的全新Struts2框架，与Struts1差别巨大，相对而言与WebWork差距反而很小，Struts2以WebWork为核心，采用在web.xml文件中配置Filter拦截器的机制处理用户请求，当在web.xml配置Filter拦截器后，每个JSP页面到Action控制器的跳转都会由Filter拦截，再由struts配置文件控制其跳转，所以Struts2实际是WebWork的更新产品。Struts2包含应用逻辑的横切面拦截器，基于注释的配置以减少和去除XML形式的配置文件，

功能强大的表达式语言，支持可更改、可重用 UI 组件的基于 MVC 模型的标签库[9]。

2、Spring简介

Spring是一个为了解决企业应用开发 的复杂性而创建的开源的框架。Spring使用基本的Javabean（封装了set和get方法）来完成以前由EJB完成的事情，它不单单用于服务器端的开发，从简单性、可测试性和松耦合的角度看，它可以帮助任何与Java有关的开发[10]。Spring是一个轻量级的控制反转(Inversion of Control，IoC)和面向切面编程(Aspect Oriented Programming，AOP)容器框架，通过在其配置文件中注入的方法，以及AOP的声明式事务管理，与Hibernate持久层框架整合，为企业提供一个轻量级的解决方案。在SSH开发中使用Spring的IOC容器负责创建和管理组件，包括前端的业务控制器（Action层），中间的业务逻辑组件（Service层），以及底层的DAO组件（DAO层），这些都将处于IoC容器的管理之下，从而大大降低各组件之间的耦合度。

3、Hibernate简介

Hibernate是一个开放源代码 的对象/关系映射框架，所谓“对象关系映射”（Object/Relation Mapping，ORM）指的是完成面向对象语言到关系型数据库的映射[11-12]，通过映射，我们可以像操作对象一样来操作我们的数据库。ORM能自动地将这些操作转换成SQL语句，从而完成对数据库的操作。Hibernate对JDBC进行了轻量级的封装，通过Hibernate支持，我们可以用ORM来对数据库进行各种操作。

ORM的基本规则有：类和表对应；类的属性和表的字段对应；类的实例和表中记录对应；一个类可以对应多个表，一个表可以对应多个类；数据库中的表可以无主键，对象必须设置主键；数据库中各表的关系映射为对象之间的关系；对象中属性的个数及名称可以和表中的字段个数及名称不一样[13]。

## 2.3 SSH的开发方式

下面来详细介绍SSH的运行方式。在表现层中，首先会通过JSP页面一系列完成操作，并负责接收请求（Request）和传送响应（Response），当JSP页面发送请求时，web.xml会拦截到请求，然后根据struts.xml里面的配置信息，将请求交给对应的action控制器处理。由于在Spring为Action控制器注入了业务逻辑组件，为业务逻辑层注入了数据持久层的（DAO）组件，在action业务层中，根据在Spring中注入的业务模型组件和持久层数据处理（DAO）组件，完成业务逻辑，并提供事物处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。Service 层（ 业务逻辑层）， 负责实现业务逻辑， 对 DAO对象进行模块封装[14]。在持久层中，会依赖Hibernate的对象化映射和数据库交互，处理DAO组件请求的数据，并返回处理结果。根据在持久层和业务逻辑层返回的结果，action也会返回不同的结果，然后strut.xml会根据不同的返回对象，控制其跳转到不同的页面。SSH框架的系统分层，如图2.1所示。



图2.1 “SSH框架”系统分层

## 2.4 SSH的开发环境

本次设计所使用的工具包括MyEclipse、MySQL和数据库管理工具Navicat。下面将一一介绍。

MyEclipse是在eclipse的基础上加上自己的插件开发而成的功能强大的企业级集成开发环境，主要用于Java、JavaEE以及移动应用的开发，它功能强大，支持Java、JSP、Servlet、SSH、JDBC等多项功能。MyEclipse是企业级的工作平台，是eclipseIDE的扩展，可以说MyEclipse是java web开发的神器。MyEclipse还提供了强大的快捷键功能，利用快捷键功能，开发者可以很方便的提高开发速度。例如，最常用的是，“alt+/”代码助手，可以帮助开发者完成代码的插入，还可以按照自己的喜好，设置快捷键位置。

本次毕业设计使用的是MyEclipse10，它还支持Java EE 6.0,、HTML5。MyEclipse10对检索功能以及错误查询功能比以前的版本更加强大，包括对js文件的错误查询。

MySQL是一款开放源码软件，非常受欢迎，是最好的关系数据库管理系统应用软件之一。它是一种关联数据库管理系统，它可以将数据保存在不同的表中，通过设定外键，将不同表中的数据联系起来，这样保存了数据的完整性，增加了数据存储的灵活性。

Navicat for MySQL，是一款专门为MySQL设计的数据库管理工具，Navicat使用了极为良好的图形用户界面，是开发者可以更加方便快捷的操作MySQL数据库。

## 2.5 本章小结

本章主要介绍了商品销售管理系统的相关技术，包括SSH框架简介，Struts2、Spring和Hibernate技术，介绍了SSH的开发方式，还简单介绍了其开发环境，包括MyEclipse、MySQL以及MySQL的界面工具Navicat。

# 3 系统分析

## 3.1 系统目标

需求是系统设计的基础[15]，对于设计商品销售管理系统的目的是希望通过此系统使得小型企业内部人员的日常工作效率可以得到显著的提升。商品管理系统仓库管理不仅能够记录并统计商品在仓库的出入记录，还可以及时准确地掌握商品库存的变动情况，销售管理可以准确的记录商品的日常销售和来自企业的订单，为企业的决策提供数据支持，同时还有会计管理部分为企业进行财务核算，准确的呈现资金的流动，将企业的商品销售管理、存货管理、运输管理三个经营业务更好地结合起来，达到数据共享、降低成本、提高效率、提高收益等目的，并且他能够及时反应采购成本，及时准确地了解采购情况和销售动态，便于调整营销策略，为管理者提供决策依据，加强了对商品的管理，加速了资金周转，提高了资金的利用率，使小企业能够确定自己资金的动向。它还包括客户信息管理模块和供应商信息管理模块，为企业的配送和采购提供支持。还有客户与供应商管理，方便企业及时联系客户与供应商，便于业务联系。另外，还有管理员模式，只有管理员可以注册新用户，员工只有在管理员注册过后，才可以登录。

本次设计希望通过系统可以达到以下目标：

（1）系统的可用性。系统能够满足需求还远远不行，还需要考虑系统的用户交互

界面设计，让开发出的系统具备易用性[16]。

（2）信息安全保密，保持数据的合理性。对用户输入的数据，系统会根据validate文件进行对数据进行检验，以达到减少人为的输入错误的目的。还为不同用户设置了不同的权限级别。

（3）系统能完成商品销售管理系统的基本功能，可以初步被小型企业使用，能够覆盖小型企业的大部分工作。

## 3.2 业务流程分析

项目开发的开始阶段，对商品销售管理进行业务流程调查，形成业务流程图，反映了系统的全貌和工作流程，属于系统分析的核心内容。

1、商品销售管理顶层业务流程图

商品销售管理顶层业务流程图，如图3.1所示。可以看出，商品销售管理系统包括销售管理、仓库管理、配送管理、会计管理、客户管理，五大模块，其中销售管理、仓库管理和配送管理是主要的管理模块。



图3.1 “商品销售管理”顶层业务流程图

2.“日常销售”业务流程图

商品销售管理系统的“日常销售”业务流程图，如图3.2所示。外部实体是个体客户，交易当面需要协商，商品价格会有轻微的浮动，依情况而定，业务处理包括，商品信息查询和销售录入。日常销售是面向个人的业务，一切交流都是面对面的。



图3.2 “商品管理日常销售”业务流程图

3、“客户与供应商”业务流程图

“客户与供应商管理”业务流程图，如图3.3所示。业务处理包括客户与供应商信息添加、查询、更新和删除 ，外部实体是销售管理员。由于客户信息和供应商信息比较简单，信息更新频率低，易于管理，所以放在同一个管理体系中。另外，其他职能的业务员也具有对客户与供应商管理的查询权限。



图3.3 “客户与供应商管理“业务流程图

4、“仓库管理”业务流程图

“仓库管理”业务流程图，如图3.4所示。处理业务包括仓库查询，采购和备货与入库4个，其中采购需要额外的采购员去完成。 销售管理和仓库管理是商品销售管理系统最复杂的两大模块。

5、“会计管理”业务流程图

商品管理的“会计管理”业务流程图，如图3.5所示。处理业务是财务管理，包括来自配送员的验收单，来自采购员的采购单，采购人员采购完成后带回的购货单，还有销售管理员的日常销售表。其中验收单是客户验收货物的需要签字的，具有法律效应，当配送员带回验收单后视为交易完成，购货单是会计人员用于核对采购商品价格，采购单是当会计核实采购单后，给采购员发放采购资金，与会计管理相关的业务主要是涉及资金的流动。



图3.4 “商品管理的仓库管理”业务流程图



图3.5 “商品管理的会计管理”业务流程图

6、“配送管理”业务流程图

“配送管理”业务流程如图3.6所示，处理业务包括制定配送优先序和配送，外部实体为客户，当客户验收货物后，需要返回一个具有客户签字的验收单。



图2.6 “商品管理的配送管理”业务流程图

## 3.3 功能需求分析

1、商品销售管理

日常销售时，企业将商品销售给客户，销售管理人员根据情况进行销售，现场确定销售价格，客户需要直接付款，付款后开收据，同时根据收据记录销售商品信息。非日常销售，即面向企业销售时，在客户发顶单之后，由销售管理员根据订单形成供货单，并发给仓管员进行备货，并进行订货存档，在配送员完成配送并返回验收单后将订货存档的信息导入销售表。

2、客户与供应商管理

客户管理即客户信息管理，主要是发送订单的客户，记录客户的单位、联系方式等信息。销售管理员在客户发送订单后，可以根据客户信息管理去完善供货单，日常联系时，也可以根据客户信息管理去查询联系方式等。若有新的客户产生，需要销售管理员去添加，客户产生变动，也可以根据情况进行删除、修改或更新。在日常销售中，也需要由销售管理员对个人客户信息进行更改，包括删除、更新、添加、查询。

3、仓库管理

仓库管理主要工作有两方面，一是由仓管员日常查询，根据查询条件查询是否有商品缺货或者商品数量过少，若有，则需要形成缺货单，由采购员去采购，并要形成采购单，采购后由仓管员去入库；二是在销售管理员将供货单发给仓管员后，由仓管员对订单上的商品进行查询，检查是否有缺货现象，若有就形成缺货单，交给采购员，去完成采购，若无缺货，就直接由仓管员去备货，形成出库单交给配送员。

4、会计管理

会计管理，主要是与资金相关的业务，包括根据采购员返回的验收单记录资金流出，根据配送员返回的验收单记录资金的流入，还有采购员返回的购货单，以修改商品售价，根据日常销售进行财务流入记录，还可以以周、月、年等为单位查询利润等情况，形成财务报表，便于进行决策和管理。会计管理因为涉及到资金的流动，因而具有较高的权限，可以查询由销售管理员录入的销售表。

5、配送管理

商品备货完成后由配送员根据出库单去进行配送，配送员根据出库单上面的相关信息完成配送，如果需要配送的任务包含多个企业或者单位，配送员还可以查询订单信息生成配送优先序完成配送。配送人员还需要对配送单进行审核，查看仓管员是否按要求进行备货，如有错误及时反映，防止在顾客验收时出现错误，从而导致需要配送员返回重新备货。配送到达后，需要客户在验收单上面签字，客户也可以在签收单上面提建议，便于改进服务质量，提高服务质量。

## 3.4 用例图

业务流程图采用的表达方式是图形，可以很好地表达业务功能之间的关系和流程，但是对功能的描述相对薄弱。下面将系统功能部分转换为用例图，详细说明了，人与业务之间的关系，即工作人员所负责的业务范围。

1、销售管理”与“客户与供应商管理”用例模型

“销售管理”与“客户与供应商管理”用例模型，如图3.7所示。显示了销售管理员需要完成的工作。



图3.7 “销售管理”和“客户管理”用例模型

2、“仓库管理”用例模型

“仓库管理”用例模型，如图2.8所示，主要工作有商品备货与入库、仓库商品查询、打印缺货单，这3类功能需要有仓库管理员完成，其中经理也可以参与其中。



图3.8 “仓库管理”用例模型

3、“采购管理”用例模型

“采购管理”用例模型，如图2.9所示。采购员主要完成采购工作，采购属于仓库管理的部分。



图3.9 “采购管理”用例模型

4、“配送管理”用例模型

“配送管理”用例模型，如图3.10所示。配送员主要工作是配送业务，当配送员配送完成后，还要带回验收单交给会计员。

5、“配送管理”用例模型

“会计管理”用例模型，如图3.11所示，是会计员需要完成的工作，会计员的工作是所有与资金流动相关的业务，比较杂，这里用财务管理统一描述。



图3.10 “配送管理”用例模型



图3.11 “会计管理”用例模型

6、“管理员”用例模型

管理员管理企业员工，企业员工只有在管理员注册后才能正常登陆。另外，管理员还具有其他所有员工的所有权限。“管理员”用例模型，如下图3.12所示。



图3.12 “管理员”用例模型

## 3.5 系统性能需求分析

系统的性能需求分析非常重要。好的性能需求分析能够使得软件项目又快又好的完成，如果需求分析的质量不高，可能会导致软件项目的延期或者严重的可能导致失败[17]。商品管理系统是一个适合为以销售为主营业务的企业服务的系统，是一个适用性比较广的信息管理系统。它功能比较全面，能够准确、及时、真实、适用的为小型企业工作。它可以大大提高企业的工作效率、管理水平和信息化程度，因此对系统的实用性、经济性、安全性、可靠性、可扩充可维护性等方面有较高的要求。

1、实用性要求

实用性是信息管理系统需要重点考虑的一项指标，因为这是企业所首先考虑的方面，也是企业最重视的指标。实用性决定了一个系统是否值得一个企业采用。实用性要求系统按企业内部工作管理分清层次与方向，不要将各个部门的工作混淆，但是也要求尽量完美体现各个部门之间的交流，更要使企业内部的数据管理更加方便与快捷。本次系统设计以满足企业进行仓库管理、销售管理、配送管理作为首要目标，着重完善仓库管理、销售管理、配送管理之间的联系，以实现客户信息管理和财务管理为次要目标。在用户界面设计中，尽量以操作简单、便于理解为目标。

2、安全性要求

系统安全性要求体现在应用软件安全性、数据库安全性、信息安全性及运行环境的安全性等方面。软件设计应做到使应用软件与数据分离，实现数据共享，防止数据丢失或破坏。数据库安全性要求保证数据库的安全，即保证数据不会被非法登录，因而数据库管理系统自身提供的安全性外，各应用系统还可通过分配权限、设置权限级别来区别不同用户对数据库操作来提高数据库的安全性。本系统针对不同的用户拥有不同的功能，只有具有一定权限的用户才能修改员工的信息[18]。普通用户只能维护及查看本人的一些基本信息，只有系统管理员（root用户）才具有所有权限[19]。运行环境的安全性是指要注意来自网络的非法侵入等，要注意尽量不要让系统所在的电脑直接连接网络。

3、可靠性要求

系统可靠性表示系统在规定的条件下和规定的时间内完成规定功能的能力。商品销售管理系统可靠性要求体现在信息的存储方面。为提高系统的可靠性，应使系统各个环节之间衔接更急紧密，尽量使用新的技术已被广泛认可并采用的技术。

4、可扩充可维护性

系统时，一般只是满足了当前使用者的需求，然而随着时间的变化，使用者会有新的需求，因而一个好的系统在设计之初就为系统的扩充，做好了准备，而且这样来算系统的维护所占的时间是最长的，有的系统维护时间甚至以年为单位计算。系统的设计与开发要有标准化合模块化的设计思想，使用合理的设计框架或者模型，当系统需要添加或者修改某个模块时，就会很方便，以尽量少产生不必要的麻烦。还要保证数据库表的可维护性，因为当使用者产生新的需求后，往往需要新的数据表存储。

5、经济性

经济性主要考虑系统运行发花费的成本，本系统硬件要求低、运行所占内存较小，对设备损害小，可长期使用。

本系统的运行平台windows7及windows7以上。

开发使用语言Java， 开发软件平台MyEclipse。

数据库管理系统软件MySQL。

## 3.6 可行性分析

随着世界范围内信息化浪潮的产生，世界范围内的所有行业开展了日益激烈的竞争，在中国各个行业都已经加快了前进的步伐，企业信息化程度已经成为评价一个企业是否有竞争力、是否具有发展前途的标准。积极推进先进完善、功能齐全的信息化模式，将是企业在中国竞争发展的大潮下，获得竞争优势的必要保证。在当前中国经济新常态，减少泡沫经济的要求下，提高企业信息化程度，改善企业经济效率显得日益迫切。

企业信息管理系统大大提高了一个企业的信息化程度，提高了企业管理效率和员工工作，节省了不必要的人力和物力，节省了开支，使企业分工更加明确，不同部门之间联系更加密切，工作更加细致，也使小企业家更加了解自己的公司。

1、系统开发必要性

随着信息技术的日益发展，我国各行各业都在逐渐走向信息化，有的行业信息化程度已经达到甚至超越了发达国家的同行业信息化水平，但是在我国也有行业信息化程度相对落后。目前小企业的信息化程度已经成为我国信息化发展 的阻碍，而且企业信息化程度也已经成为衡量 一个企业竞争力的重要因素。商品销售管理系统适合信息化水平较低的小企业，他可以大大提高小企业的管理效率和工作效率，节省了人力，提高了小企业的竞争力，也会促进我国信息技术的发展，提高我国整体信息化水平。

2、经济可行性

小企业，由于人员比较少，资金有限，一二一些人不愿把多余的钱投资在短期不会提高企业利润的方向。一个企业的信息化，是需要企业长期投资，逐渐发展起来的。但是商品管理系统适合大多数小企业，它比较简单，不需要企业太多投资，它页面简单，人机交互良好，只需要有简单的电脑基础就能完成，因此上手很快，相关人员完全不需要培训，仅仅只需要一些时间去熟悉业务。总而言之，商品销售管理系统占有和使用的经济资源非常低，仅仅只需要配备电脑，并装有需要的平台即可，当然还需要相关人员操作；采用商品销售管理系统后，可以提高工作效率，使工作更加明确，为企业决策提供依据。

3、技术可行性

商品销售管理系统的技术可行性从开发人员和使用人员两方面进行[20]。商品销售管理系统的系统开发，采用S2SH框架，使用Java语言，其中还涉及到JSP技术，JavaScript技术等，系统开发工具是MyEclipse，数据库采用MySQL，运行平台是windows，完全有能力去实现本次设计。在本次开发中，尽量采用以前用过的技术和方法，以便于满足项目开发需要，保证开发成本和时间。但是为了系统的发展，有些方面要尽量考虑以前没有用过的技术，例如，在本次开发中，登录验证时，使用validate文件验证，方便以后的更改。

4、操作可行性

系统设计应该遵循的原则是要可用性髙，上手快。所以，系统的界面设计一定要人性化，并且符合方便快捷的特性[21]。商品销售管理系统，在设计时，已经考虑到了用户操作方面。系统采用网页式，并充分考虑了管理人员，使其操作简便；实现了用户与数据库的交互，在数据录入时，还有对数据类型的检测功能，非常方便、准确、规范；界面简洁友好，具有易上手、灵活等优点。

5、社会可行性

商品销售管理系统，适合大多数小企业。当今中国，小企业的信息化程度很低，这在一些二三线城市，甚至小县城里非常明显。在中国，小企业无处不在。使用商品销售管理系统大大提高了小企业的工作效率，完全符合当今中国乃至世界的经济信息化大潮，有利于企业的成长。

## 3.7 本章小结

本章首先介绍了系统的设计目标，进而确定了系统要实现的功能范围。接着详细分析了商品销售管理的业务流程，画出了顶层业务、日常销售管理、仓库管理、配送管理、会计管理等的业务流程图。第三，又作了功能需求分析，对系统各个管理模块作了详细的介绍。第四，画出了用例图，详细说明了人与业务之间的关系，即由谁负责哪一样工作。第五，对系统性能作了详细分析，从实用性、安全性、可靠性、可扩充可维护性、经济性，考虑了系统的性能，并对系统的运行平台对了介绍。第六，为系统作了可行性分析，包括5个方面，即系统的开发必要性、经济可行性、技术可行性、 操作可行性、社会可行性。通过本章的分析，明确了系统开发的实现目标、业务流程、功能需求、性能需求等，为下一章系统设计作了准备。

# 4 系统设计

## 4.1 功能结构图

用户的需求具体体现在各种信息的提供、保存、更新和查询方面,这就要求数据库结构能充分满足各种信息的输出和输入[22]。根据需求分析，得到商品管理系统的功能结构图，如图4.1所示：



图4.1 商品管理的功能结构图

（1）商品销售管理系统的客户与供应商管理：包括客户与供应商信息的查询、删除、修改、更新。按客户编号（userid）的查询和删除，按客户姓名（username）的查询和删除。

（2）商品销售管理系统销售管理：包括商品信息更改、商品信息查询（包括按商品编号和商品名的查询）、销售录入、订货存档等等。

（3）商品销售管理系统配送管理：包括配送信息查询和更新、制定配送优先序。

（4）商品销售管理系统会计管理：包括更新销售数据库、财务核算，订单状态查询等。

（5）商品销售管理系统仓库管理：库存商品查询（按商品编号和商品名）、商品入库、更新商品数据库等。

（6）商品销售管理系统员工管理：员工注册、员工信息查询。

## 4.2 数据库设计

数据库是一种完成数据存储和操作的工具[23]。由于商品销售管理系统需要采集、存储及处理大量的数据信息，所以对于数据的管理就尤为重要[24]，一个系统的成功很大部分上取决于数据库是否合理的设计，如何设计出稳定、高效的数据库是每个系统设计人员需要着重考虑的，只有好的数据库设计才能使后期维护顺利进行[25]。数据库设计是信息管理系统中的主要部分，数据库是存放数据的仓库。数据库设计是指对于一个固定的系统环境，尽量设计出最优的、最合理的数据库结构，并且不会对固定的应用环境有更高的要求，达到合理有效的存储和管理数据信息，使用户满意。

数据库是一个集合，是一个为了能够使各层员工获取信息的，被整个企业共享的数据的集合。有了数据库企业可以更快获取信息，更快的做出合理决策，更好的使用企业资源。数据库的设计要满足数据库的信息录入、信息更新、信息删除等方面的需求。当企业想要查询信息时，会通过对系统某个模块的操作，对数据库内的数据进行筛选，以达到对数据的使用。

### 4.2.1 数据库设计原则

1、表中的记录有唯一的标识符

在设计数据库表的时候，应该有一个良好的习惯。用一个ID号唯一标识行记录，即ID作为标的主键，尽量不要通过名字、编号等字段对数据进行区分。每个表都应该有一个ID列，任何两条记录都不能共享同一个ID值，另外ID在值应该有数据库自动管理，而不能交给前台应用程序，否则的话很容易产生ID值不同意的状况。在本次设计中每个表都有ID列，并设其为主键，类型为自增（increment）。

本次数据库设计，还加入了行号。不同于ID，ID不能够自动维护。而行号可以自动变化，每次查询的第一条记录行号都为1。

2、数据库对象要有统一的前缀名

一个复杂的数据库设计，其对应的数据库表往往会以千记，而分清每张表的的作用或意义的难度会很大。这样，会使设计人员在数据库设计时，造成很大的困扰，甚至会减缓项目进度。本次毕业设计，虽然需要的表不多，但是仍然采用前缀名设计，以养成良好的习惯。每个管理模块的前缀名都要尽量不一样,尽量使设计人员看到表名，就理解其意义，即属于哪个管理模块的哪张表。

3、变量名的命名规则

（1）变量名采用小写英文字母

（2）变量命名尽量可以见词知意。

（3）变量名长度不超过30个字符。

### 4.2.2 主题数据库

主题数据库表如下所示：

**表4.1 主题数据库表**

|  |  |  |
| --- | --- | --- |
| 主题数据库 | 数据表 | 数据表的主要内容 |
| 客户信息管理 | 客户信息表 | 客户ID、姓名、联系电话、邮箱、所在单位等 |
| 供应商信息管理 | 供应商信息表 | 供应商ID、联系人、联系方式、地址等 |
| 销售管理 | 商品信息表 | 商品编号、商品名称、商品规格、数量等 |
| 日常销售存档 | 商品编号、商品名称、销售价格、数量等 |
| 订货存档 | 单位ID、商品名称、数量、联系人等 |
| 销售记录表 | 商品ID、商品名称、销售价格、数量等 |
| 商品编号表 | 商品编号、商品名称、商品规格、商品品牌等 |
| 仓库  管理 | 采购记录表 | 商品名称、商品数量、商品价格、购买单位等 |
| 缺货记录表 | 商品名称、商品规格、剩余数量等 |
| 出库记录表 | 商品名称、商品规格、剩余数量、客户单位等 |
| 商品底限表 | 商品编号、商品底限数量、剩余数量等 |

**续表4.1** **主题数据库表**

|  |  |  |
| --- | --- | --- |
| 主题数据库 | 数据表 | 数据表的主要内容 |
| 配送  管理 | 配送信息表 | 商品名称、数量、目的单位、联系人等 |
| 配送记录表 | 配送编号、配送日期、目的单位等 |
| 会计  管理 | 财务报表 | 采购花销、销售盈利等 |
| 日常结算表 | 日期、销售利润、盈利等 |

### 4.2.3 数据库概念设计

根据对系统的需求分析和系统设计，需要找出本系统的相关实体。所谓实体，即客观存在并可以相互区别的实物，实体可以是具体的人、事、物，也可以是抽象的概念或联系。本系统的实体有商品实体、商店实体、仓库实体、员工实体、供应商实体、客户实体、客户的等等实体。

由于实体属性较多，故不再列出，实体属性可以参照论文下面的数据库逻辑结构设计部分。

1、采购与入库”实体-联系图



图4.2 “采购与入库”实体-联系图

2、“出库”实体-联系图

出库”实体-联系图，如图4.3所示。下图是仓库管理部分的“出库”实体-联系图，与上图共同组成仓库管理整体实体-联系图。



图4.3 “出库”实体-联系图

3、“销售管理”实体-联系图

“销售管理”实体-联系图，如图4.4所示。表示的是顾客在实体店的交易活动，其中存档表示的是在交易成功后的活动。



图4.4 “销售”实体-联系图

4、“配送”实体-联系图

“配送”实体-联系图，如图4.5所示。



图4.5 “配送”实体-联系图

5、“客户与供应商信息”实体-联系图

“客户与供应商信息”实体-联系图，如图4.6所示。

图4.6“客户与供应商信息”实体-联系图

### 4.2.4 数据库逻辑结构设计

数据库的概念结构设计是根据企业需求，需要按照某种排列对数据进行存储，因而数据库的概念结构设计就是把企业工作所需的信息集合作为一条记录存储。由于数据库的存储效率直接影响到未来系统的运行效率，因此数据库物理结构的设计往往是以数据存储效率最优为设计出发点的。总而言之，数据库逻辑结构设计，反映了数据库中所有表的详细结构和属性内容，即是对数据库表的设计。

下面根据以上的E-R图和试实体分析，进行数据库表的详细分析。由于数据库表较多，所以有的部分会省略。

1、客户与供应商管理

包括客户信息表和供应商信息表，主键或外键为空表示，此属性不需要为主键或者外键。

**表4.2 客户信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 10 | 主键 | 否 |
| 客户编号 | Int | 4 | 外键 | 否 |
| 姓名 | Varchar | 4 |  | 否 |
| 联系电话 | Varchar | 11 |  | 是 |
| 邮件 | Varchar | 15 |  | 是 |
| 所在单位 | Varchar | 30 |  | 是 |
| 单位地址 | Varchar | 30 |  | 是 |

**表4.3 供应商信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 10 | 主键 | 否 |
| 供应商编号 | Int | 4 | 外键 | 否 |
| 联系人 | Varchar | 4 |  | 都 |
| 联系电话 | Int | 11 |  | 是 |
| 邮件 | Varchar | 15 |  | 是 |
| 主营业务 | Varchar | 30 |  | 是 |
| 所在单位 | Varchar | 30 |  | 是 |
| 单位地址 | Varchar | 30 |  | 是 |

销售管理包含的表比较多，主要有商品信息表、商品编号表、销售存档表（日常销售）、订货存档表、销售记录表等。

**表4.4 商品信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 数量 | Int | 4 |  | 否 |
| 底限数量 | Int | 4 |  | 否 |
| 单价 | Int | 4 |  | 是 |
| 供应商ID | Int | 4 |  | 是 |
| 备注 | Varchar | 50 |  | 是 |

**4.5 商品编号表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 商品名称 | Varchar | 10 |  | 否 |
| 商品规格 | Varchar | 10 |  | 否 |
| 商品品牌 | Varchar | 10 |  | 是 |
| 备注 | Varchar | 50 |  | 是 |

**表4.6 销售存档表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 价格 | Int | 5 |  | 否 |
| 数量 | Int | 4 |  | 否 |
| 总价 | Int | 5 |  | 否 |
| 购买方 | Varchar | 10 |  | 是 |
| 日期 | Date | 15 |  | 否 |
| 备注 | Varchar | 50 |  | 是 |

**表4.7 订货存档表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 价格 | Int | 5 |  | 否 |
| 数量 | Int | 4 |  | 否 |
| 总价 | Int | 5 |  | 否 |
| 购买方 | Varchar | 10 |  | 是 |
| 日期 | Date | 15 |  | 否 |
| 送货日期 | Date | 15 |  | 是 |

3、仓库管理

仓库管理包括采购记录表、出库记录表、商品底限表。其中采购记录表尤其重要，采购记录表里面的备注属性，可以作为采购失败商品标记。查询时按照备注是否为空查询，可以得到目前没有采购的商品，即供应商缺货。仓库管理的相关表如下：

**表4.8 采购记录表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 价格 | Int | 5 |  | 否 |
| 数量 | Int | 4 |  | 否 |
| 供应商ID | Int | 4 |  | 否 |
| 采购日期 | Date | 15 |  | 否 |
| 备注 | Varchar | 50 |  | 是 |

**表4.9 出库记录表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 出库ID | Int | 5 | 外键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 客户ID | Int | 5 |  | 否 |
| 数量 | Int | 5 |  | 否 |
| 日期 | Date | 15 |  | 否 |
| 备注 | Varchar | 50 |  | 否 |

商品底限表，用于仓管员采购时查询缺货或少货的商品种类，即当商品数量低于商品底限是，可以方便查询。

**表4.10 商品底限表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 | 外键 | 否 |
| 底限数量 | Int | 5 |  | 否 |
| 备注 | Varchar | 50 |  | 否 |

4、配送管理

**表4.11 配送信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 客户ID | Int | 5 |  | 否 |
| 出库ID | Int | 5 |  | 否 |
| 配送日期 | Date | 15 |  | 否 |
| 配送人 | Varchar | 4 |  | 否 |
| 备注 | Varchar | 50 |  | 是 |

5、会计管理

主要是根据销售存档表和订货存档表生成日常结算表。

**表4.12 日常结算表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键或外键 | 是否为空 |
| ID | Int | 5 | 主键 | 否 |
| 商品编号 | Int | 10 |  | 否 |
| 数量 | Int | 5 |  | 否 |
| 售价 | Int | 5 |  | 否 |
| 进价 | Int | 5 |  | 否 |
| 日期 | Date | 15 |  | 否 |
| 备注 | Varchar | 50 |  | 否 |

## 4.3 本章小结

本章主要是系统设计部分，首先画出了系统的功能结构图，介绍了商品销售管理系统主要功能。然后是数据库设计，说明了数据库设计原则，主题数据库、概念结构设计、逻辑结构设计。本次设计主要应用与小型企业，属于小型企业信息管理系统，因而数据库设计比较简单。

# 5 系统实现与测试

## 5.1 商品销售管理系统的实现

商品销售管理系统，针对的是销售型的小型企业，主要目的是提高小型企业的管理水平。商品销售管理系统的首页，如图5.1所示。



图5.1 商品销售管理系统首页

### 5.1.1 员工登录界面

商品销售管理系统，是有不同职务的人员共同登录使用的系统，因此不同的登录人需要具有不同的权限。即不同员工只具有在职位之内的权限。员工登陆界面，主要是对用户进行检查，排除非工作人员的非法登录，只有合法的用户在输入正确的情况下，才可以登录。

登录页面，需要实现以下功能。通过输入正确的用户名和密码才能进入商品销售系统。系统登录程序流程图，如图5.2所示。员工登陆页面，如图5.3所示。



图5.2 “系统登录”程序流程图



图5.3 “员工登录”页面

### 5.1.2 root用户界面

整个系统只有一个root用户，root用户拥有系统所有权限，并且root用户的权限不可修改。root用户可修改其他用户的权限。Root用户具有整个商品销售管理系统的所有权限。例如，可以查询员工登录信息，修改用户的权限，可以查询所有的数据。“管理员登录”页面，如图5.4所示。登陆成功页面，如图5.5所示。“管理员登录”使用流程图，如图5.6所示。



图5.4 “管理员登录”页面

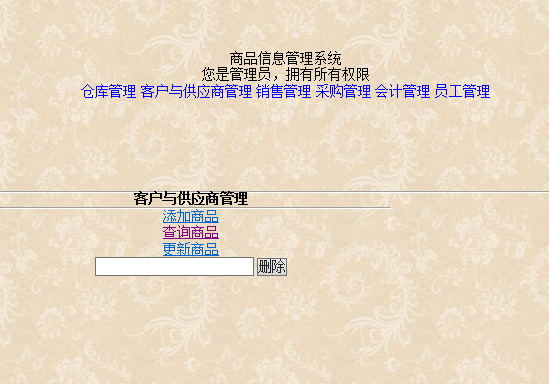


图5.5 管理员登录成功页面



图5.6“管理员登录”使用流程图

### 5.1.3 销售管理界面

商品销售管理，是由销售员进行登录与管理。销售员，只能登录销售管理模块，而且只具有查询、添加、删除已售商品的权限，登录员具有删除权限是因为当客户对某种商品不满意要求退货时，需要删除销售表内的对应商品，但是销售员不具有修改商品价格的权限。当有企业型客户发来订单时，由销售员录入，生成送货单，并存档。“销售管理“程序流程图和使用流程图，如图5.3所示。“销售管理”页面，如图5.7所示。

图5.7A “销售管理”使用流程图 图5.7B “销售管理”程序流程图

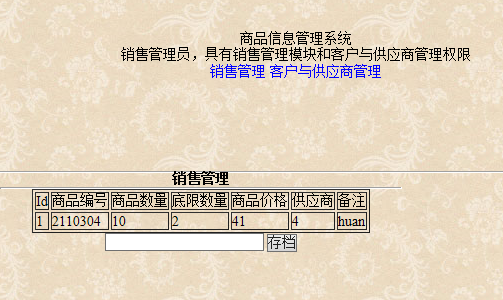


图5.7 “销售管理”查询页面

### 5.1.4 仓库管理

仓库管理，仓管员具有比销售员更大的权力，仓管员可以查询、添加、修改商品信息，另外仓管员还可以查询是否有新的送货单，这样便于为企业型客户备货，提高工作效率。由于“用户登录”程序流程，如图5.1所示，因此不在显示仓管员登录流程。“仓库管理”程序流程图和页面，如图5.8(A)和(B)所示。



图5.8“仓库管理”使用流程图



(a)“仓库管理”页面 图(b)“客户与供应商管理”页面

图 5.9 仓库管理

### 5.1.5 会计管理

会计管理，主要负责与金额相关的部分，因而具有比销售管理和仓库管理更高的权限。他拥有对销售管理、仓库管理、这两个模块的查询权限。当采购员需要采购时，他还负责审核采购单，并发放采购资金，当采购员采购完成后，还要审核购货单，如有商品进货价格变化，负责修改商品进价信息，同时要修改商品售价。“会计管理”程序流程图，如图5.5所示。



图5.10 “会计管路”使用流程图

### 5.1.6 客户和供应商管理

客户和供应商管理，主要用于联系客户和供应商，因为需要此类信息的员工较多，所以客户和供应商管理的权限限制较低。销售员和会计具有对客户和供应商管理的全部权限，包括客户和供应商信息的添加、删除、更新、查询，主要用于日常联系客户和核对相关信息。仓管员和采购员具有其查询权限。

### 5.1.7 配送管理

配送管理，主要工作是当仓库备货完成，并生成送货单后，由配送员去登录配送管理界面，修改送货状态（备货、配送、已签收），并生成配送优先序。“配送管理”程序流程图，如图5.6所示。

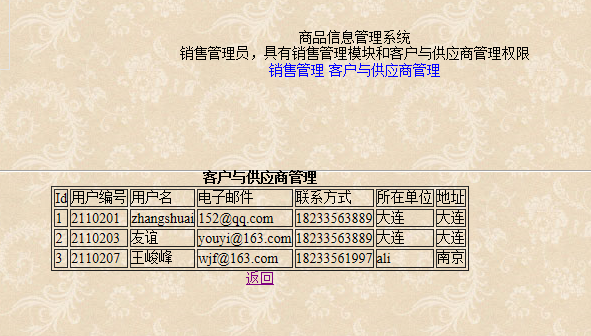


图5.11 “客户管理”查询结果



图5.11 “员工管理”界面对客户信息的更新



图5.6 “配送管理”使用流程图

## 5.2 商品销售管理系统的测试

### 5.2.1 测试目的

每一个系统，在由系统开发人员设计时，都会仔细考虑以尽量减少其中的漏洞，但是并不能完全杜绝这种错误，因而才有了系统测试这一工作，系统测试并不仅仅是为了证明程序正确的执行了预期的功能，而是为了发现在设计过程中没有考虑到的错误而测试的过程。系统测试的主要工作是，通过与系统的 需求定义作比较，发现系统设计与系统需求定义矛盾或者不符的地方，验证软件系统的功能和性能等满足其在系统设计初所制定的要求。系统与系统设计初的需求定义作比较，主要从几个方面去考虑。（1）要确定预期输出结果，用事先写出的程序预期输出结果去与程序运行输出结果作详细的比对，发现其中存在的错误。（2）检查系统中是否实现在设计初定义的所有需求，是否与设计初定义的需求有不符之处。（3）如果系统实现了，在需求定义中没有定义的需求，要仔细考虑是否合理。从测试方法来看：系统测试可以使用黑盒、手工、自动化、静态、动态方法测试。在系统开发过程中，总会由于各种原因，会在开发的各个阶段产生错误，所以测试活动应该贯穿整个开发过程中，尽早发现错误，消除隐患。测试工作，尽量不要由开发人员完成，因为开发人员会主观的认为自己的开发是完美的，会无意识的漏掉某些方面，也会限定测试思路和测试方向，有很大的局限性。

### 5.2.2 测试环境

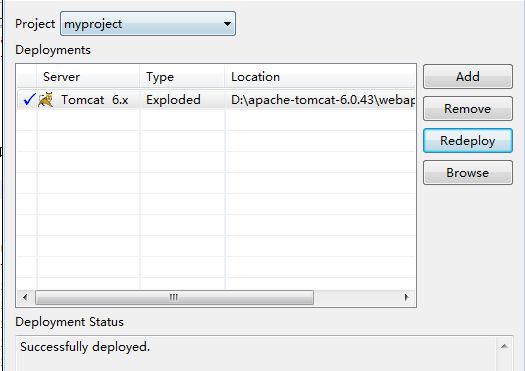


图5.7 MyEclipse部署

对系统进行部署是进行综合测试的必要步骤，包括连接数据、部署网站以及打开tomcat服务。首先是部署在Tomacat上。如上图5.7所示。

本次毕业设计，采用的是MyEclipse和MySQL数据库，运行环境是tomcat。

### 5.2.3 测试用例及结果分析

测试用例（Test Case）是为了检查系统实现后是否满足系统设计之初定的目标，而根据数据库编制的一系列测试数据，测试是为了是系统符合设计初衷，运行更加完美，更加符合用户需求。根据测试方案，不同模块设计需要不同测试用例。测试用例设计主要通过用例表格进行设计。根据功能测试，对系统各个模块进行测试用例设计。因为整个测试过程大同小异，所以只展示部分测试过程。

**表5.1 销售模块特性**

|  |  |  |
| --- | --- | --- |
| 测试功能编号 | 测试内容 | 测试优先级 |
| 1.1 | 添加商品 | 高 |
| 1.2 | 按商品名或商品编号查找商品 | 高 |
| 1.3 | 删除、更新商品 | 高 |
| 1.4 | 日常销售存档 | 高 |

**表5.2 销售模块测试用例**

|  |  |  |
| --- | --- | --- |
| 序号 | 测试需求 | 优先级 |
| 1.1.1 | 添加商品，正确输入数据，成功 | 高 |
| 1.1.2 | 添加商品，输入数据有误（某些数据类型有误），失败 | 中 |
| 1.1.3 | 添加已重复商品，添加失败 | 中 |
| 1.2.1 | 按商品名和商品编号查询，成功 | 高 |
| 1.1.2 | 输入不存在的商品名和商品编号，失败 | 中 |
| 1.3.1 | 更新商品信息，数据合理，成功 | 高 |
| 1.3.2 | 更新商品信息，数据不合理，失败 | 中 |
| 1.3.3 | 删除商品，成功 | 高 |
| 1.4.1 | 将查询的商品加入销售表，成功 | 高 |

## 5.3 核心代码

商品销售管理系统采用SSH框架完成，下面将介绍系统的重要代码。

（1）将SessionFactory交给Spring管理，为DAO组件的持久化提供支持。

<bean id=*"sessionFactory"* class=*"org.springframework.orm.hibernate3.LocalSessionFactoryBean"*>

<property name=*"configLocation"*>

<value>classpath:hibernate.cfg.xml</value>

</property>

</bean>

因为没有在Spring中配置数据源，而是由另外的Hibernate配置文件，因此需要将其导入到Spring配置文件中去。这样就可以直接管理SessionFactory了，而不需要另外的SessionFactory工具了。

（2）使用HibernateTemplate模板类在Spring中定义HibernateTemplate模板类，

并为其注入SessionFactory实例。

<bean id=*"hibernateTemplate"* class=*"org.springframework.orm.hibernate3.HibernateTemplate"*>

<property name=*"sessionFactory"*>

<ref bean=*"sessionFactory"*/>

</property>

</bean>

这样就把HibernateTemplate交给SessionFactory管理了。配置DAO组件，并注入HibernateTemplate实例。如此就可以直接在DAO层使用HibernateTemplate模板了。图5.8

是在DAO层使用HibernateTemplate的方法。在DAO层的查询方法结合了HQL语句和Iterator（迭代器）。利用Iterator迭代器可以将HibernateTemp返回的List<entity>转换成<entity>[]实体类型。

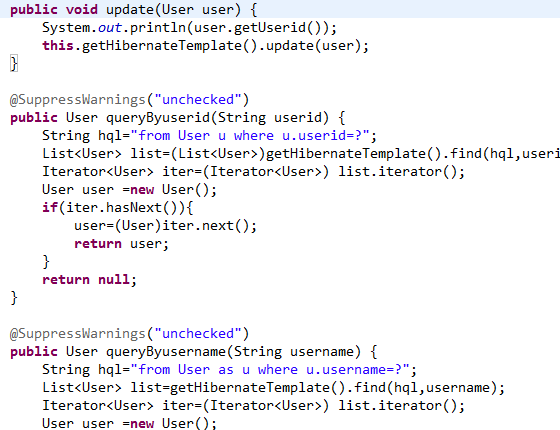


图5.8 DAO层HibernateTemplate实例

## 5.4 本章小结

本章是商品销售管理系统的实现、测试代码介绍部分。首先介绍了商品销售管理系统的功能实现，包括销售管理模块、仓库管理模块、客户与供应商管理模块、配送管理模块、会计管理模块。平台实现之后，就是平台测试，介绍了测试目的、测试环境、测试用例，并对测试结果进行了分析。系统测试是对系统每个模块进行测试信息的输入，检查是否有不合理的功能，从而实现了平台的稳健性，最后是部分核心代码介绍。

# 结 论

本文研究的内容是基于提高以销售为主的小型企业的信息化管理水平而设计和实现的商品销售管理系统。本文的系统界面主要通过JSP开发，后台数据库是MySQL，开发工具是MyEclipse，开发框架是S2SH。由于本系统是属于专为小型企业使用的管理系统，在开发之初，就制定了合理的企业需求，在开发过程中以系统需求分析为参考，这样不会有不必要的功能木块，也有利于后续的系统提升与扩充。

本文阐述了系统从提出需求到系统分析、系统设计、系统实现和测试的全过程，体现了如何将用户需求转化为具体的系统功能的过程。商品销售管理系统实现了系统设计之初的需求，包括销售管理、仓库管理、客户与供应商管理、配送管理、会计管理五大模块。以销售为主的小型企业采用本平台，实现了商品销售管理的信息化管理，对于改善企业信息化程度，节省企业资源有很大帮助。

本文的设计在也存在一些不足。例如，对于不同工作人员权限的设计，不能很好地表达每个员工的权限，系统设计所使用的新技术也比较少，为后期的维护带来了不便。系统界面不太好，有所欠缺。

# 致 谢

首先，我要衷心感谢我敬爱的导师袁静波老师。在整个毕业设计过程中，袁静波老师给我提了大量的有用建议，对我的论文给予了大量的支持和帮助，为本文的完成付出了很多心血。在我的整个论文完成工程中，袁老师耐心地指导，严格的要求，严谨的态度，深深鼓励了我。

同时，我还要感谢东北大学秦皇岛分校所有支持和帮助过我的老师和同学。在大学四年的生活中，我不仅仅学到了知识，更是收获了和老师同学们之间的深厚的友谊。转眼间四年时光，一晃而过，我即将结束我的大学生涯，走向社会，但是我的学习生涯还远远没有结束，社会将是我们的下一学习舞台。不管我身在何处，我都永远不会忘记与老师，与同学的美好时光，都会永远记得我是东秦的一员，都会记得我的老师和同学们，更加会记得我的母校：东秦。

最后，我要再次感谢我身边的所有人，是你们给我带来了温暖，时刻陪伴着我。

# 参考文献

[1] 张程远.信息化对企业竞争力的影响及其评价研究[D].中国海洋大学,2011.

[2] 施永裕,林子铭.不同国家和地区的社会信息化程度与人民生活质量的相关性分析[J]. 管理学报,2009,08:1019-1028.

[3] 付睿臣,毕克新,李唯滨. 企业信息化研究综述[J]. 科技管理研究,2009,12:238-240.

[4] 阳敏慧. 我国企业财务管理信息化研究[J]. 中国乡镇企业会计,2015,04:222-223.

[5] 李小文.中小企业移动信息化需求识别与引导机制研究[D].北京邮电大学,2012.

[6] 何信,杜江,庞海艳.基于SSH2框架的Web系统的设计与实现[J].统计与管理,2015,01:49-53.

[7] 王希望,孟祥书,王福顺.基于SSH架构的Web系统的开发方法[J].河北农业大学学报(农林教育版),2010,04:538-541.

[8] 王春超.基于J2EE架构的SSH组合框架的研究[D].长春理工大学,2010.

[9] 李倩倩.基于Struts+Spring+Hibernate框架的编目管理系统的研究与实现[D].中国地质大学（北京）,2013.

[10] 柴楠.基于SSH框架的人力资源系统的设计与测试[D].北京邮电大学,2012.

[11] 冯学军.基于SSH框架的Web网站设计与实现[D].长春理工大学,2010.

[12] 翟高粤.基于SSH框架的高校餐费管理系统设计与实现[J].计算机时代,2015,05:47-49.

[13] 梁德华.基于Hibernate4框架的Java访问数据库的设计与实现[J].山东工业技术,2014,23:146.

[14] 温小清,潘永才,刘海龙,汪标,罗雪姣.基于SSH2的电子监管平台的设计与实现[J]. 物联网技术,2015,01:64-66.

[15] 李都.商务会所商品销售管理系统的设计与实现[D].厦门大学,2014.

[16] 布赫.基于SSH的数字迎新系统的设计与实现[D].吉林大学,2013.

[17] 杨文韬.基于SSH框架的智能社区信息管理系统的设计与实现[D].中山大学,2013.

[18] 陈佳宝.瓷砖销售管理系统开发[D].云南大学,2013.

[19] 王健合.基于SSH框架的员工管理系统[D].南昌大学,2012.

[20] 汪文静.超市商品销售管理系统设计与实现[D].电子科技大学,2012.

[21] 袁春桃.基于SSH框架的企业人力资源管理信息系统设计与实现[D].厦门大学,2014.

[22] 胡峻榕,陆宇亮,温佩芝.普通高校篮球课程网络化教学的数据库设计研究[J].当代体育科技,2015,05:120-121.

[23] 刘华平.超市商品销售管理系统的设计与实现[D].电子科技大学,2013.

[24] 陈迪,赵骞. 高校网上评教系统中数据库的设计和评价算法[J]. 教书育人(高教论坛),2015,09:36-37.

[25] 孙守梅,何鑫,周坤,张业男. 基于SSH框架的图书管理系统设计[J]. 农业网络信息,2015,04:118-120.

[26] 潘斌,党庆忠,朱涛,赖波.The System Design and Implement Based on SSH Framework Technology[A]. 武汉大学、美国James Madison大学、美国科研出版社.Proceedings of International Conference on Engineering and Business Management（EBM2010)[C].武汉大学、美国James Madison大学、美国科研出版社:,2010:4.

[27] XIONG Jia,FAN Zhen,ZHANG Zengguang,ZHANG Senlin,LIU Meiqin. Design and Implementation of Silk Relic Information Systems via SSH[A]. 中国自动化学会控制理论专业委员会、中国系统工程学会.第三十三届中国控制会议论文集（D卷）[C].中国自动化学会控制理论专业委员会、中国系统工程学会:2014:6.

[28] 荣宏庆.我国中小企业信息化建设探索[A].Information Engineering Research Institute,USA.Proceedings of 2012 Third International Conference on Telecommunication and Information(TEIN 2012)[C].Information Engineering Research Institute,USA:,2012:4.

# 附 录

## 附录A

Design and Implementation of Silk Relic Information Systems via SSH

The silk relic information system based on SSH proposed in this paper is divided into four layers: persistence layer, DAO layer, business layer and presentation layer. Persistence layer is made up of persistent objects (PO) under the management of SessionFactory of Hibernate. The mapping from database table to PO makes object oriented access to database. DAO layer provides service for business layer by encapsulating the basic database CRUD operations. Business layer is the core logic of the system. It invokes DAO layer to implement business logic of the system. Presentation layer is made up of JSP pages and the MVC framework of Struts2. JSP page is responsible for collecting business data and presenting system state. The MVC framework carries out the separation of user interface and the model of system.

MySQL is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). Many of the world's largest and fastest-growing organizations rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software. The system proposed in this paper also chooses MySQL as its database. To simplify the analysis, only basic information of the silk relic and its related pattern information are taken in consideration. One of the database tables named basic, stores the ID, name, discovering time, and other basic information of silk. The pattern table stores the theme, picture path in the server, and its related basic number etc. One silk relic may relate to several patterns, thus the relationship between basic table and pattern table is one-to-many. Fig. 8 shows the detail of basic table and pattern table. Persistent classes are abstracted from database tables. Each table of the database corresponds to a persistent class.

The persistent class of basic table is Basic.class, and Pattern.class is related to pattern table. With low invasive design, Hibernate has no special requirements for persistent classes. The persistent classes are plain old Java objects (POJO). However, the following rules are still needed to obey in the development of persistent classes.

1) Non-parameter constructor: Each persistent class needs a non-parameter constructor. Hibernate invokes the Constructor.newInstance() method to create the instance of persistent class by using the related non-parameter constructor.

2) Identifying property: The identifying property is mapped to the primary key of the database table. The identifying property can be any type from basic types and the wrapper types of basic types.

3) Setter and getter methods: As for the basicName property ofBasic.class, the methods named getBasicName() and setBasicName() are needed. The default attribute access strategy for Hibernate is JavaBean. If necessary, the access strategy can be changed.

4) Overriding of equals() and hashCode(): If the persistent class is in the Set, the equals() method and hashCode() method need to be overridden. The common way to carry out these methods is comparing the identifying attribute. If the identifying attribute is equal, the instances of this class are related to the same row of the database table. Due to foreign key constraint, the relationship between persistent classes is presented by member variable of the class, and the member variable also needs setter and getter methods.

The mapping between persistent class and table of database is set in the configuration file of the persistent class. These characters are provided by Hibernate. Each member variable is mapped to an attribute of database table.

Based on Hibernate persistence layer, DAO component encapsulates the database operations, which is the common mode in Java enterprise application. With the use of DAO mode, each DAO component contains database access logic, and it completes basic CRUD operations on a database table. The implementation of DAO layer consists of three steps: firstly, design the interface of DAO component; secondly, implement the DAO interface in the form of class; finally, carry out the deployment of DAO component.

The design of DAO interface

Without any implementation, DAO interface only declares the CRUD methods. The implementation of these methods epends on the underlying persistence technology. With the se of DAO interface, the switch between different ersistence technologies such as traditional JDBC, ibernate used in this system and any other technology is uch easier. When the system needs to switch another ersistence technology, only another different DAO mplementation is needed and it is needless to make any hange of the other part of the system, which greatly mproves the scalability of the system.

Every DAO interface declares the methods mentioned above, and some other query methods are also needed to provide service for business layer.

The implementation of DAO

With abundant support for DAO based on any common persistence technology, Spring makes it easy to implement DAO component. As for Hibernate, Spring provides HibernateDaoSupport as the base class of DAO component. The instance of HibernateTemplate is easily got by passing the reference of SessionFactory to HibernateDaoSupport. The functions of HibernateTemplate are so powerful that can easily realize the majority of database operations. In the system, the DAO component must extend from HibernateDaoSupport and implements its related DAO interface. The business logic component works on DAO interface with no consideration of the implementation class.

The deployment of DAO

The DAO component is based on Hibernate and Spring, and is created and managed by Spring container. Spring container is responsible for the injection of SessionFactory which is the running basis of DAO component. In order to integrate with Hibernate, Spring provides lots of utility classes. By using the LocalSessionFactory class, the SessionFactory of Hibernate is attached to the Spring IoC container. Before configuring SessionFactory, the data source must be provided. The system uses C3P0 data source. Spring container is also responsible for the management of data source. The properties of the data source configuration and the corresponding meanings are shown in Table 1.

The configuration of SessionFactory includes data source, mapping files between persistent classes and database tables, and the properties of SessionFactory for Hibernate.

For the DAO component extended from HibernateDaoSupport, the injection of SessionFactory reference is needed. As all the DAO components need the reference of SessionFactory, the use of inheritance simplifies the configurations of the DAO components. Firstly, a DAO template is configured, and it is injected the reference of SessionFactory. After that, all the other DAO components are inherited from this DAO template. Thus all the DAO components are injected the reference of SessionFactory. The setSeesionFactory() method which is used to inject the SessionFactory reference, is provided by the parent class HibernateDaoSupport, so there is no need for DAO to provide the setSeesionFactory() method.

4.3 The Implementation of Business Laye

The silk relic information system uses three business logic components for the three roles of the system which are guest, common user and administrator. The three business logic components are GuestService, UserService, and AdminService. The business logic components encapsulate DAO components to provide service for Action. Because each business logic method involves multiple DAO operations and DAO is a single operation on database table, one business logic method may involve multiple records of database. The simplified use case diagram is shown in Fig. 9. Business logic components are responsible for achieving desired business methods of the system. The number of the business logic methods is the same as the requirement numbers of the system. Like DAO components, the deployment of business logic components is also provided by the dependency injection of

the Spring IoC container. The Spring transaction management is related to business logic methods. The single DAO operation is atomic, so it is needless to manage the DAO transaction. With tx namespace and aop namespace, the system can be easily configured for business transaction management. The <tx:advice…/> element under tx namespace is used to configure enhanced transaction processing, and the <aop:advisior…/> element configures automatic proxy.

4.4 The Implementation of Presentation Layer

In order to handle user request, the corresponding Action class must be provided. In application, Action is just the controller to invoke the method of business logic component, and the business logic is provided by business logic component. The following takes the example of adding silk relic information to introduce the implementation of presentation layer. To simplify the JSP page, only two attributes basicId and basicName are taken in consideration. The page mainly contains a form and the action attribute is set to the related Action class. The request of JSP page is intercepted by FilterDispatcher and then dispatched to the related Action. After the processing, a String type value is returned. The returned value is SUCCESS, FAILTURE, or any other values to indicate the processing result. According to the returned value, Action is configured to control the jump of JSP page after processing. The following shows the configuration of InsertDataAction.

5 Conclusions

The SSH architecture integrates Struts2, Spring and Hibernate, making full use of the MVC layered structure of Struts2, the seamless connection between frameworks based on IoC and AOP of Spring and the lightweight ORM of Hibernate. Compared with the traditional development based on JavaEE framework, the system based on SSH architecture has loose coupling, short development cycle and high maintainability. By analyzing the integration strategies of SSH, the silk relic information system based on SSH is designed and implemented in this paper. According to the different roles of the system, users are authorized to get access to some or all functions of the system. Although the requirements are always changing in our development process, the system based on SSH architecture is adaptive to the changes and the development difficulty does not increase significantly. The practice indicates that SSH architecture is an effective solution in developing web application.

译文：基于过SSH的丝绸文物信息系统的设计和实现

丝绸文物信息系统是基于SSH提出的，SSH分为四层：持久层、DAO层、业务层和表示层。持久层是由持久化对象(PO)的管理下的Hibernate SessionFactory。从数据库表映射到PO使得面向对象数据库的访问。DAO层为业务层提供服务通过封装的基本CRUD操作数据库。业务层是系统的核心逻辑。它调用DAO层来实现系统的业务逻辑。表示层是由JSP页面和Struts2的MVC框架。JSP页面负责收集业务数据和系统状态。MVC框架进行的分离用户界面和系统的模型。

MySQL是世界上的第二个最广泛使用的开源关系数据库管理系统(RDBMS)。许多世界上最大和增长最快的组织依靠MySQL来节省时间和金钱为他们的高容量网站、关键业务系统和打包软件。系统也选择了MySQL作为数据库。为了简化分析,只考虑丝绸遗迹及其相关模式的基本信息信息。每个数据库表存储都有基本的ID、名称、发现时间、丝绸和其他基本信息。模式表存储主题、图片路径在服务器及其相关基本数量等。一个遗迹可能与几个模式对应，因此基本表和模式表是一对多的关系。持久化类从数据库表抽象。每个表数据库的持久化类对应一个。

persistent类是普通Java对象(POJO)。然而,下面的规则仍然需要遵守持久化类的发展。

1)非参数构造函数:每个持久化类需要一个非参数的构造函数。Hibernate调用Constructor.newInstance()方法来创建持久化类的实例通过使用相关的非参数的构造函数。

2)标识属性:标识属性映射到数据库表的主键。识别属性可以是任何类型的基本类型和基本类型的包装器类型。

3)Setter和getter方法:至于basicName basic语言的属性。类的方法叫getBasicName()和setBasicName()是必要的。Hibernate是JavaBean的默认属性访问策略。如果有必要,访问策略可以改变。

4)覆盖的equals()和hashCode():如果持久化类的设置,equals()方法和hashCode()方法需要覆盖。执行这些方法的常用方法是比较识别属性。如果在确定属性相同的情况下,这个类的实例与数据库表的同一行。由于外键约束,提出的持久化类之间的关系是类的成员变量和成员变量也需要setter和getter方法。

数据库的持久化类和表之间的映射是在持久化类的配置文件设置。这些角色由Hibernate提供。每一个成员变量映射到数据库表的一个属性。

基于Hibernate持久性层,DAO组件封装数据库操作。使用DAO模式,每个DAO组件包含数据库访问逻辑,完成基本的CRUD操作在一个数据库表。DAO层的实现包括三个步骤:首先,设计DAO组件的接口;其次,实现DAO接口类的形式,最后,进行DAO组件的部署。

没有任何实现,DAO接口只声明了CRUD方法。这些方法的实现依赖于底层的持久性技术。DAO接口的集合，切换不同的持久性技术(如传统JDBC，本系统和其他技术中使用hibernate是如此容易。当系统需要切换另一个持久化技术，只需要另一个不同的DAO实现,不必做任何改变的其他系统的一部分，从而大大提高了系统的可伸缩性。

与丰富的基于任何常见的持久性技术支持DAO,Sprin使它容易实现DAO组件。至于Hibernate，Spring提供了HibernateDaoSupport 的DAO组件的基类。HibernateTemplate很容易得到的实例通过HibernateDaoSupport SessionFactory的参考。HibernateTemplate的功能非常强大,可以轻松实现绝大多数的数据库操作。系统中，DAO组件从HibernateDaoSupport教程，需实现相关DAO接口。业务逻辑组件DAO接口没有考虑工作的实现类。DAO的部署

基于Hibernate和Spring的DAO组件,并由Spring容器创建和管理。Spring容器负责注入SessionFactoryDAO组件的运行基础。为了与Hibernate集成,Spring提供了大量的实用工具类。通过使用LocalSessionFactory类,Hibernate SessionFactory被注入到Spring IoC容器中。在配置SessionFactory之前,必须提供数据源。系统使用C3P0数据源。Spring容器还负责管理数据源。SessionFactory的配置包括数据源、持久化类和数据库表之间的映射文件,和Hibernate SessionFactory的属性。

丝绸文物信息系统使用三个业务逻辑组件，系统的三个角色是客人,普通用户和管理员。这三个业务逻辑组件GuestService,UserService,AdminService。业务逻辑组件封装DAO组件提供的。因为每个业务逻辑方法涉及多个DAO操作和刀是一个操作在数据库表中,一个业务逻辑方法可能涉及多个数据库的记录。简化用例图是图9所示。业务逻辑组件负责实现所需的业务系统的方法。的数量相同的业务逻辑方法系统的需求数量。像刀组件,业务逻辑组件的部署也依赖注入提供的

为了处理用户请求,必须提供相应的Action类。在应用程序中,Action是控制器调用业务逻辑组件的方法，业务逻辑提供了业务逻辑组件。

SSH框架集Struts2、Spring和Hibernate,充分利用MVC Struts2的分层结构,基于国际奥委会和AOP框架之间的无缝连接的Spring和Hibernate的轻量级ORM。与传统的开发基于JavaEE框架相比,该系统基于SSH架构松散耦合、开发周期短和高可维护性。通过分析SSH的集成策略,基于SSH的丝绸文物信息系统设计和实现。根据系统的不同角色,用户有权获得部分或全部功能的系统。尽管需求不断变化在我们的发展过程中,该系统基于SSH架构是自适应变化和发展困难度并没有明显增加。实践表明,SSH架构在开发web应用程序是一种有效的解决方案

## 附录B

The System Design and Implement Based on SSH Framework Technology

The current B/S model of Web application system based on the traditional three layers structure of subdivided into five layers: user layer, presentation layer, business layer and persistence layer and database layer [1]. Traditional way of development, however, did not correspond to each layer of existing tools to quickly develop clear Web application system structure level. At present, China's telecommunications industry USES a variety of real-time billing solutions, to match the billing management system facing the management scope expands unceasingly, involving business increasingly complex challenges, an urgent need to an efficient, robust and able to work on demand platform support. In the J2EE platform, in recent years, there are three kinds of mature technology based on MVC pattern: Struts, Spring and Hibernate, they provide a rapid development tools with different levels of Web system. The organic combination of the three kinds of technology to SSH technology framework, not only can effectively improve the efficiency of system development, more can improve system security, stability and robustness.

SSH framework for its lightweight, open source, support for agile development was welcomed by the developer, he is short for the Struts + Spring + Hibernate. SSH will effectively integrate together the three architecture, its hierarchy is clear, to avoid the single frame structure under the high levels of coupling, makes it easier to system maintenance and extension. SSH framework, Web layer is mainly composed of Struts. Struts is essentially based on JSP Model2 implemented a MVC framework, can simplify the Web presentation layer development, realize the separation of logic and data, shorten the development time. The Spring framework with the Struts framework can perfect combination, and has unique feature in transaction management rely on processing, its architecture is based on the IOC container based on Bean properties. Spring is mainly responsible for processing the application business logic, business check, and transaction management, and management of the business layer object dependency. Hibernate is an O/R persistent object/relational mapping framework, fully USES the ordinary Java objects (pojos), to the lightweight JDBC encapsulation, can under the condition of the database connection is broken so the objects in the diagram according to the UI layer, the data can be updated and back to the data persistence layer and updated in the database. In general, three framework integration method is to use the Struts architecture as the system of the whole infrastructure (complete separation of the MVC), using the Hibernate framework to provide persistence layer support, business layer with Spring support.

In telecom billing system, according to user requirements, system to provide accurate, fast, display diverse, real-time statistics, billed as the user on the basis of other work. In the background database using stored procedures and JOB timer realize the data real-time statistics. Using B/S application mode in the browser to provide tables, pie charts and column chart to show the statistical results.

Business process analysis of the accounting management, the system is mainly to complete the business system management, data management, voice business query statistics, monitoring, alarm, etc.

Overall system is divided into 5 levels, the user layer: by dealing with the interaction with the user, the client browser mainly includes the JSP and HTML pages. The presentation layer: some View is related to the JSP page. Its data representation USES actionforms, Controller by the Action Servlet combining Struts2config. The XML configuration file and the Action class. Business layer: to achieve the required business method and the interface, system consists of a JavaBean and some Java classes. The core business logic encapsulated in the javabeans, and at the same time in the javabeans encapsulation and the interaction between Hibernate. Persistence layer: the persistence layer is mainly to realize the object/relational mapping, using Hibernate itself provided by the database connection pool technology. Database tier: the Oracle database management application data.

System development process, using the MyEclipse 6.5 as development tools, database backend Oracle 10. Development, with three kinds of framework in SSH integration is the basic method of set through the configuration file. To deploy the WEB server will use the UNIX operating system, use Tomcat6.0 as service program. First of all, based on the system UML model to realize the basic Java objects, and then to complete basic DAO interface, and give the Hibernate DAO implementation, DAO class in the Hibernate framework is used to realize the mapping between the Java classes and database and access and finally completed business logic by the Spring.

The presentation layer is responsible for handling the client and the business layer and data transmission between requests. This layer USES the Struts framework, mainly by the JSP page, actionforms, Action, Action Servlet and Struts2config XML common implementation. Traffic statistics module, for example, query statistics management input statistical conditions in JSP pages, JSP page will through these statistical condition request submitted to the corresponding interceptors, through its filtering statistical condition, and then to the corresponding Action, the Action call BO, BO call DAO, DAO from data meet the conditions for statistical data in the table, the data package in the object, and then return to the front desk interface layer, used to analyze the displayed to the user. 3.2.2 business layer.

First to build each function module corresponding management class to manage and maintain the business logic, and then modify the Spring configuration file, will manage the corresponding bean class to register configuration, in order to achieve the use of AOP mode for the purpose of transaction control. Spring in the configuration file configuration SessionFactory and HibernateTemplate, so as to realize Hibernate resource dependency injection. Configuration files will be JDBC DataSource, Hibernate SessionFactory resources such as search, respectively defined as different Bean, the application object access to resources you just need to reference through the corresponding Bean.

First need to create a persistent class, and then through a mapping XML format description document to achieve the object mapping for the tables in the database, the relationship between the relationship between the mapping for the table. The document suffix "HBM. XML". In order to control the Hibernate work, you must create a Hibernate. CFG. XML configuration files.

In the J2EE platform, good design pattern and framework technology can improve the development efficiency, ensure the quality of software development. Practice shows that application of SSH framework technology in information system development, can shorten system development.

译文：基于SSH框架的系统设计与实现

当前B/S模式的Web应用系统在传统的3层结构基础上细分为5层：用户层、表示层、业务层、持久层和数据库层[1]。然而，传统的开发方式没有与各层相对应的现成工具来快速开发出结构层次清晰的Web应用系统。目前，我国电信业采用了多种实时的计费解决方案，与之配套的计费管理系统面临着管理范围不断扩大、涉及业务日益复杂等挑战，迫切需要一个高效、健壮、能够随需应变的工作平台的支持。在J2EE平台下，近年有三种比较成熟的基于 MVC 模式的技术：Struts、Spring和Hibernate，它们提供了不同Web系统层次的快速开发工具。将这3种技术有机结合起来构建的SSH技术框架，不但可以有效提高系统开发效率，更能改进系统安全性、稳定性和健壮性。

SSH框架以其轻量级、开源，对敏捷开发的支持受到开发人员的欢迎，他是Struts+ Spring + Hibernate的简称。SSH将这3个架构有效整合在一起，其层次结构清晰，避免了单一的框架结构下各层的高耦合度，使得系统维护与扩展更为容易。SSH框架中，Web层主要由Struts实现。Struts实质上就是在JSP Model2的基础上实现的一个 MVC框架，可简化Web表示层开发，实现表示逻辑和数据的分离，缩短开发时间。Spring框架同Struts框架能够完美的结合，并且在事务管理依赖处理方面有着独到之处，其架构基础是基于Bean属性的IOC容器。Spring主要负责处理应用程序业务逻辑、业务校验和事务管理，同时管理业务层的对象依赖。Hibernate 是一个O/R（持久化对象/关系）映射框架，完全采用普通的Java对象(POJO)，对JDBC进行了轻量级的封装，这样就可以在数据库连接断开的情况下将对象图中的数据显示到 UI 层之上，这些数据也可以被更新和回送到数据持久层并在数据库里更新。一般说来，集成 3 种框架的方法是利用Struts架构作为系统的整体基础架构（完成MVC的分离），利用Hibernate架构来提供持久层支持，业务层用Spring支持。

在电信计费系统中，根据用户需求，系统要提供准确的、快速的、展示方式多样的、实时的话单统计数据，作为用户进行其他工作的基础。在后台数据库中使用存储过程和 JOB 定时器实现数据的实时统计。使用 B/S 应用模式，在浏览器端提供表格、饼图和柱图来展现统计结果。

对计费管理的业务过程分析得知，该系统主要需要完成的业务有系统管理、资料管理、语音业务查询统计、监控告警等。

系统总体上分为 5 个层次，用户层：通过客户端浏览器处理与用户的交互，主要包括JSP和HTML页面。表示层：其中View是一些相关的JSP页面。其数据表示采用ActionForm，Controller由Action Servlet结合Struts2config.xml配置文件和 Action 类组成。业务层：实现系统所需的业务方法和接口，由JavaBean以及一些Java类构成。核心业务逻辑封装到JavaBean中，同时在JavaBean里封装与 Hibernate的交互关系。持久层：持久层主要实现对象/关系的映射，使用Hibernate 自身所提供的数据库连接池技术。数据库层：采用Oracle数据库管理应用数据。  
 系统开发过程中，采用MyEclipse 6.5作为开发工具，数据库后台选用Oracle 10。开发时，完成SSH中三种框架整合的基本方法是通过配置文件来进行设置。部署时WEB服务器采用UNIX操作系统，用Tomcat6.0作为服务程序。首先根据系统UML模型实现基本的Java对象，然后完成基本的DAO接口，并给出Hibernate的DAO实现，采用Hibernate框架中的DAO类来实现Java类与数据库之间的映射和访问，最后通过Spring来完成业务逻辑。

表示层负责处理客户端和业务层之间的请求和数据传输。该层采用Struts框架，主要由JSP页面、ActionForm、 Action、 Action Servlet 和 Struts2config.xml 共同实现。以话务量统计模块为例，查询统计管理人员在JSP 页面中输入统计条件，JSP 页面将这些统计条件通过请求提交给对应的拦截器，通过它过滤统计条件，然后交给相应的 Action，Action调用BO，BO调用DAO，DAO从数据表中获取符合统计条件的数据，将这些数据包装在对象中，再一层一层的返回到前台界面，显示给用户分析使用。**3.2.2** 业务层实现。

首先构建每个功能模块相应的管理类来管理和维护其业务逻辑， 然后修改 Spring 配置文件，将管理类对应的 bean 进行注册配置，以达到对其使用 AOP 模式进行事务控制的目的。Spring 通 过 在 配 置 文 件 配 置 SessionFactory 和HibernateTemplate，从而实现Hibernate资源的依赖注入。配置文件中将JDBC DataSource、 Hibernate SessionFactory等资源的查找分别定义为不同的Bean，应用对象对资源的访问只需要通过相应的Bean进行引用即可。

首先需要创建持久化类，然后通过一个XML格式的映射描述文档来实现将对象映射为数据库中的表，将其关系映射为表之间的关系。该文档后缀名为“hbm.xml”。为了控制Hibernate工作，必须创建Hibernate.cfg.xml配置文件。

在J2EE平台下，好的设计模式和框架技术可以提高开发效率，保证软件开发质量。实践表明，应用SSH 框架技术进行信息系统开发，可以缩短系统开发。