

论文题目 基于 Spring 框架的商务债权管理系统的研究与设计

专业学位类别	工 程 硕 士
学 号	201192231164
作 者 姓 名	陈 昇
指 导 教 师	汤 羽 教 授

分类号

汤羽

密级

UDC ^{注1}

学 位 论 文

基于 Spring 框架的商务债权管理系统的研究与设计

(题名和副题名)

陈 昇

(作者姓名)

指导教师

汤 羽

教 授

电子科技大学

成 都

陈剑雄

高 工

厦门网络科技有限公司

福 建

(姓名、职称、单位名称)

申请学位级别 硕士

专业学位类别 工 程 硕 士

工程领域名称

软 件 工 程

提交论文日期 2014.3.25

论文答辩日期 2014.5.24

学位授予单位和日期 电子科技大学

2014 年 6 月 24 日

答辩委员会主席

评阅人

注 1: 注明《国际十进分类法 UDC》的类号。

RESEARCH AND DESIGN OF FINANCIAL-CLAIM MANAGEMENT SYSTEM BASED ON THE SPRING FRAMEWORK

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Major: Master of Engineering

Author: Chen Sheng

Advisor: Tang Yu

School : School of Information and Software Engineering

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期：_____ 年 _____ 月 _____ 日

摘要

现代商品经济社会，交易形式多种多样，并且不断创新，信用交易、信托交易等交易形式不断推陈出新，形式越来越灵活。随着科技地不断进步，电子信息技术的广泛应用、金融服务功能的不断完善，无纸化办公，网络办公必然成为一种趋势，越来越多的企业选择使用网络软件，使得员工能够不受地域空间的限制，方便快捷地参与工作。其中，经过身份验证即可使用的办公系统逐渐成为企业级应用软件的主导产品。

从古至今，债权就在商品经济市场中扮演重要的角色，随着经济的发展，传统的销售模式已经不能满足商品经济的发展，在当今的商品交易中，信用销售占据销售的比例越来越大，这一现状的副产品便是企业的债权管理，债权管理就像防火墙一样，在一定程度上隔离风险，正因为如此，债权管理对于一个公司起到盾的作用，被受人们关注，同时债权管理也趋向于网络化，本文正是基于此大背景，对现阶段的商务债权现状进行了阐述，并分析了现有的商务债权管理工作上的不足。在此基础上，通过收集资料以及实地实例调查确定债权管理工作中的各个环节的功能和需求，对商务债权管理系统进行设计与分析，力求开发一个符合现代办公要求的商务债权管理系统。本文设计的系统是基于J2EE的Spring框架，采用MVC设计模式，引用Struts和Hibernate，采用Struts、Spring和Hibernate三种著名的框架进行分层架构模式设计，并以B/S模式进行服务器和浏览器页面的开发，以出售大型机械的用户对债权的管理为例，开发债权管理系统，该系统与传统的系统相比，界面更加美观，功能更加完善，更加能适应现代化网络办公的要求，可以大大的提高员工的工作效率，为企业节省资源。

由于债权管理系统所需要处理的信息量往往是“大数据”乃至“海量数据”，这些数据的管理往往具有高难度、安全性低、容错性低等难点，传统的信息处理技术不能很好的解决这些问题，因此如何更有效的收集、分析和处理“大数据”，就成了一个亟待解决的问题。针对这一问题，本文引入数据挖掘技术，采用Apriori算法对数据进行关联分析，力求从中挖掘出潜在价值的信息，以对债权进行有效的管理。

关键字：债权，J2EE，Spring框架，B/S，MVC

ABSTRACT

Modern commodity economy, there are a variety of transaction forms, and innovation, constantly credit transactions, trust transactions and transactions in the form, the form of increasingly flexible. With the technology to progress, the extensive application of electronic information technology and the constant improvement of the financial services functions, paperless office, network office is bound to become a trend, more and more companies choose to use network software so that employees can not the geographical space restrictions, convenient involvement. Authenticated to use office systems gradually become the leading products of the enterprise-level application software.

In ancient times, the claims on the commodity economy market plays an important role, along with economic development, the traditional sales model can not meet the development of commodity economy, in today's commodity trading, credit sales account for an increasing proportion of sales, this status quo is a byproduct of corporate debt management, debt management as firewalls, to a certain extent, it quarantines the risk, and as such, debt management company for a shield to play the role of being subject to people's attention,debt management tend to network Based on this background, the stage of Commerce claims status quo, and analysis of the current the lack of some business debt management work. On this basis, through the collection of data as well as field instance of the function and needs of the investigation of all aspects of the work of determining claims management, business debt management system design and analysis, and strive to develop a business debt management system in line with modern office requirements. In this paper, the design of the system is based on J2EE Spring Framework, Using the MVC design pattern,citing Struts and Hibernate, using Struts, Spring and Hibernate frameworks three kinds of famous layered architecture model design, and B / S mode for server and browser page development,the sale of large machinery dealer claims management, for example, compared with the traditional system, the interface is more beautiful, more perfect, more able to adapt to the requirements of a modern network office, you can greatly improve the efficiency of the staff, to save resources for the enterprise.

Claims management systems need to deal with the amount of information is

often a "big data" and even "the vast amounts of data", the management of these data often have a difficult, low-security, fault tolerance and low difficulty, the traditional information processing technology is not good to solve these problems, how to more effectively collect, analyze and deal with "big data", became a solved problem. To solve this problem, this paper introduces data mining technology Apriori to analysis data, and strive to dig out the potential value of the information, and effective management of claims.

Keywords: claims, J2EE, Spring framework, B/S, MVC

目录

第一章 引言	1
1.1 研究背景	1
1.2 研究现状与意义	2
1.3 本文的研究工作	4
1.4 本文的组织结构	4
第二章 系统需求分析	5
2.1 问题描述	5
2.2 可行性分析	5
2.3 需求分析	6
2.3.1 功能性需求分析	6
2.3.2 非功能性需求分析	7
2.4 系统详细设计	7
2.4.1 新增功能设计	12
2.4.2 删除功能设计	16
2.4.3 修改功能设计	17
2.4.4 查询功能设计	18
2.5 本章小结	20
第三章 系统总体设计	22
3.1 设计原则	22
3.2 架构设计	23
3.2.1 债权的相关概念	23
3.2.2 MVC 架构	24
3.2.3 Spring 框架	26
3.2.4 Struts 和 Hibernate 框架	28
3.2.5 MySQL 数据库和 B/S 结构	29
3.3 模块设计	31
3.3.1 基础信息管理模块	31
3.3.2 债权查询及统计模块	32
3.3.3 财务催收管理模块	33
3.3.4 系统管理模块	33

3.4 数据库设计	35
3.4.1 E-R 图	35
3.4.2 表设计	38
3.4.3 数据库连接	43
3.5 基于数据挖掘的债权数据分析模型设计	44
3.5.1 数据挖掘相关技术	44
3.5.2 Apriori 算法	45
3.5.3 债权数据分析模型设计方法	46
3.6 本章小结	49
第四章 系统实现	50
4.1 界面设计	50
4.1.1 系统登录界面设计	50
4.1.2 系统主界面设计	51
4.2 基础数据管理模块	52
4.2.1 客户信息管理	52
4.2.2 销售信息管理	54
4.3 债权查询及统计模块	55
4.3.1 债权明细查询	55
4.3.2 债权统计查询	55
4.4 财务催收管理模块	56
4.4.1 还款	56
4.4.2 催收记录查询	57
4.4.3 法务单据	57
4.5 系统管理	58
4.6 本章小结	58
第五章 系统测试	59
5.1 系统测试的步骤	59
5.2 系统测试的方法	59
5.3 测试用例	60
5.3 本章小结	61
第六章 总结与展望	62
致谢	63
参考文献	64

第一章 引言

1.1 研究背景

随着科技的进步，科学技术作为第一生产力，极大的推动着商品经济的发展，在现阶段，无纸办公已经成为一种潮流，在给企业带来经济效益的同时，也增加了交易方式，信用交易、信托交易等新型交易方式的出现，

针对交易方式的多样性，以高成本商品为例，许多大型机械设备、高科技设备由于造价高昂，会导致交易不便，而分期付款、融资租赁等交易方式可以有效地解决买家资金不够无法承担特别是许多小成本公司前期投入过大造成资金运转不流通的问题，同时也能帮助卖家提高销售额，减少库存。因此，以分期付款、融资租赁等方式完成高成本商品交易，对于买卖以及用户三方来说，都能在一定程度上促进交易，但是同时也会产生大量的债权问题。

据调查研究表明，在标的权利为银行存款、赊卖贷款债权、保险金、贷款债权、工程垫支款等场合中，债权为压倒性的选择^[1]，债权是卖家与买家在一定的信用基础上进行的交易，并不仅仅是签订合同，后期的债权管理才是债权的核心内容，买家需按要求偿还债权，卖家需要追回剩余的期款，在这一过程中，方便快捷的工具不仅方便卖家管理债权，还能方便买家偿还债权。在企业中，往往需要设立专门的债权管理部门，对债权进行管理。

对于出售大型机械的用户来说，交易的同时会产生极大量的数据信息需要管理和维护。比如用户、厂家的基本信息、销售产品的记录、买家的详细信息，如公司名称、公司法人代表、公司地址、电话、营业执照，产品代号，销售数量等等信息。同时，对于同一个客户来说，可能并不只是交易一台设备，也可能并不是只用一种形式进行交易，因此会产生大量的交易数据。对于以分期付款、融资租赁等方式交易出去的机械设备来说，会产生还款日期、还款金额、拖欠款项等等的债权数据。大量的数据如果仅仅使用纸张或者传统的文本文件存储，会对查阅和管理造成极大的麻烦^[2]。

因此，一个用于管理各项信息的商务债权系统软件会对储存、管理、维护和分析数据起到极大的帮助。一个债权信息系统软件，由用户操作，输入客户公司的各种资料，每台设备的交易信息数据，各台设备的债务信息数据，系统将所有的数据信息保存于数据库，在用户的登录系统时从数据库调出展示，随时可以查看各项数据，界面清晰，操作简便，是一个解决债权信息管理的有效途径，采用一定的方式对数据进行归类处理，不仅可以方便查阅，同时通过对归类数据的分

析，可以发现同类型的客户特点，从而挖掘潜在的客户。

1.2 研究现状与意义

在当今的商务活动中，企业往往有债权性投资，调查表明还有部分企业仍在使用传统的文件管理方式，通过给文件编号，按时间和首字母分类的方法管理债权。即使大量自动化设备的引进，在一定程度上大大简化卷宗的查找流程，但是效率仍然很低，并且报表的制作需要花费大量的人力物力，很难实现批量检索，这对于债权的管理是个极大的负担，因此一个高效的管理系统必不可少，在这个系统中，工作人员可以管理海量的债权，从中追踪债务人信息，发放通知以及账单。

2004年，戴发文从商业银行的不良金融债权入口，结合国内外的相关研究结果，在不良金融资产管理这一研究领域，对我国商业银行体系内不良资产的管理进行重点关注，建立起一套既借鉴国际经验，又适合我国国情的理论体系和逻辑体系^[3]。2008年，壮琼芳针对债权债务的复杂性、紧迫性、艰巨性，一些企业集团存在“新官不理老账”甚至“老官不理旧账”的现象，通过分析企业集团债权形成的原因，提出债权管理的相关策略^[4]。2009年，韩麟在文中指出企业债权的概念，分析收账对象的重点考虑因素，并从建立合理的信用标准、建立符合实际的信用条件、建立强有力的收款政策等方面来改革和加强企业债权管理，是企业债权管理的理论指南^[5]。2011年，段琳琳在敏捷软件开发过程的基础上，开发了一个债权管理系统，以国有四大银行在贷款业务中涉及的抵押、质押为背景，提出债权管理的重要性，并加以实践，开发相应的系统，该系统实现了录入数据的功能，以方便处置人员查看处置报表对未处置资产的处置估价作参考^[6]。同年，王海鹏在“债权回收管理系统的设计与实现”一文中，针对日本的债权债务管理实际情况，提出一个债权回收管理系统，通过该系统，工作人员可以方便的管理债权数据，并可以通过自动生成报表和各种通知，发放账单等来追踪债务人员信息，在开发语言上，分别使用了C#和JAVA语言，不统一，不便于数据和系统的交互、维护以及更新^[2]。

综合分析当前现状存在着如下不足：

①多数的企业对于债权管理依然过分依赖于传统的纸质文件管理，对交易数据的整理，耗费大量的人力物力，即使有开发相对应的债权管理系统，也只是用于简单的记录，并不能处理大量的数据存储需求。

②用户各自的系统统一管理性差，而有些系统又是早期开发的数据管理系统，就像是一个个数据孤岛，所以难以做到不同系统间的数据交互。

综上所述，鉴于现阶段的不足，我们提出能基于“云”概念模式下的商务债权管理系统，并对相关数据进行分析。

基于当前商务债权管理的研究现状，本课题的研究目标为：

- 1、针对企业的商务债权管理，设计相应的管理系统，能提高企业的管理效率。
- 2、在了解债权信息 and J2EE 的基础上，开发了基于 Spring 框架的商务债权管理系统。
- 3、通过数据挖掘技术，对债权进行分析，挖掘潜在的客户，为企业带来一定的效益。

商务债权管理系统的开发可以有效的解决商品交易中存在的问题，极大的方便了交易多方，同时提高债权管理效率，降低债权管理成本，符合发展的要求，为企业带来长期利益。本研究课题的建设具有很大的实用价值和意义，具体如下：

- 1、系统的使用可以减少债权管理人员的繁重业务，提高债权管理效率。

债权，作为企业的一项投资，往往是企业资金流动的脉搏，为管理债权，企业会花费大量的人力进行管理，传统的文件管理往往需要进行修订，按字母或时间进行查询，对于账单的发放更需要员工加班加点完成，当面对海量数据时，计算机的优势凸显出来，债权管理系统基于计算机的高运算速度、大存储量，可以方便的对债权进行管理，在一定程度上减少员工的工作量，提高债权管理效率。

- 2、通过本系统，可以及时发现未还期项，便于追踪遗漏款项

面对海量数据，人力往往会花费太久的时间进行整理，这会导致未还债权遗漏，导致债权追踪的困难，该系统设计了提醒通知功能，不仅能提醒买家还贷，而且还能提醒卖家未还债权项，方便卖家采取一定的措施以免买家逃债。

- 3、实现企业的高效率办公以及现代化办公，加快资金流动速率

债权管理作为企业管理的一个重要部分，其效率能影响整个企业的工作效率，甚至影响企业的生存，在当今世界，时间就是金钱，现代化的办公是大势所趋，现代化的一个重要标志就是计算机的大量使用，本系统基于计算机的基本功能，在此基础上进行设计，具有一定的现实意义。

- 4、为债权管理的研究提供理论上的借鉴

本研究在对国内外的债权管理现状的基础上提出，针对现状的不足，设计实现了适合商务活动的债权管理系统，对传统的一般系统进行了创新性和探索性的改进，为进一步研究债权管理提供了一定的借鉴。

1.3 本文的研究工作

商务债权管理是一个富有实际意义的工作，本文首先介绍了商务债权管理的背景，分析了商务债权管理的研究现状，并指出现阶段商务债权管理存在的不足以及开发与企业接轨的商务债权管理系统的可行性和重要性。

为确保系统的使用性，本文详细地分析了开发一个商务债权管理系统所需要的核心技术，如Mvc下的Spring框架，并将其用于本系统的开发，考虑实际应用中的相关数据存储要求，本文将引入数据挖掘的相关技术，力求确保系统的安全性、实用性并从大量的记录中挖掘有用的数据，以便对客户进行管理。

1.4 本文的组织结构

本论文围绕商务债权管理系统的研究与设计，主要包括以下几个方面的研究工作：

第一章 分析课题的研究背景、当前的研究现状，说明当前存在的问题，在此基础上指出本文工作的研究意义。

第二章 描述系统的相关问题，对系统的可行性进行研究，分析系统的功能性需求和非功能性需求，说明服务层代码是如何完成对数据库中数据的增删改查功能的详细设计过程，分析数据维护主系统等数据表的设计与主要功能模块的详细实现过程。

第三章 包括分析了系统设计涉及到的相关技术，并分析给出了系统各部分的功能模块图

第四章 介绍系统各个模块的实现。

第五章 介绍了系统的测试方法和相关用例。

第六章 总结商务债权管理系统的特特点，分析了研究的发展方向，指出进一步的工作方向。

最后给出了致谢和参考文献。

第二章 系统需求分析

2.1 问题描述

(1) 主要功能:

通过调查商务债权的基本情况,以仓单质押贷款业务为实例,深入调查货物入库、质押监管、出库提货等各个环节的信息,管理用户相应的业务数据,对债权信息进行维护,实现企业办公的现代化、自动化、信息化以及高效化。

(2) 问题性质:

商务债权管理系统是以人为本,利用当代先进的科学技术以及科学理论,对仓单质押进行管理,达到提高工作效率,为企业带来长久利益为目的。

(3) 系统开发的目的:

该系统主要解决的是针对传统的纸质管理信息的不便,利用先进的网络技术,对商务债权进行现代化的管理操作,以方便用户对商品和客户的管理,实现一个数据库管理系统软件,以方便员工通过该系统高效的对业务进行管理。

(4) 系统的组成:

商务债权管理系统是典型的浏览器管理系统,其开发主要包括前段各个模块的开发和后端海量数据维护的数据库的建立两部分。前者要求要能在总体上满足操作简单、功能完善等特点,后者要求能建立起完善的数据库系统,数据要具有完整性、符合实际情况且易于维护等特点。

商务债权管理系统以仓单质押为应用背景,主要包括以下四大模块:基础信息管理、债权信息查询及统计、财务催收管理和系统管理。

2.2 可行性分析

本文的可行性分析从必要性和可能性两方面着手:

(1) 必要性

随着商品交易的增多,商务债权对于一个企业来说,其附加的信息也越来越错综复杂,当前主要采用的是简单的办公软件对业务数据进行简单的管理,部门之间、业务双方之间基本没有实现信息的共享,很多工作仍然采用的是手工统计,纸上作业这种工作模式,不仅效率低下,而且浪费企业资源,更不利于企业的发展,结合当前债权管理的现状,以现实为基础,针对企业债权管理的工作流程,开发一个高效且简便的商务债权管理系统是企业现代化办公的关键所在。

（2）可能性

当代科技快速发展，编程技术日新月异，1995 年 sun 公司提出的 java 编程语言和 Java 平台，被广泛应用于个人 PC、数据中心和互联网等领域，在编写跨平台应用软件更为成熟，在此基础上，本文以仓单质押系统为背景，在对其进行全面需求分析的基础上，分别对 Web 层、业务逻辑层、数据持久层进行了详细设计。针对项目特点，提出了一个通用模式的轻量级架构：Struts2 实现 MVC 模式并完成表示层功能；Hibernate 负责数据的持久化工作，具有一定的可行性。

综上所述，开发一个商务债权管理系统是十分必要的且可行的。

2.3 需求分析

2.3.1 功能性需求分析

（1）界面分析

用户界面（UI）设计在人与计算机之间搭建了一个有效的交流媒介，在设计时，首先要识别用户、任务和环境需求，一旦用户任务被确定，则可以通过创建和分析用户场景来定义一组用户界面对象和动作。

用户登录界面具有识别用户身份，保护系统数据不轻易被盗取的功能，为避免泄露问题，所以这些对象都在登陆系统是都需要进行身份验证，且身份验证的安全性需要提高。

在管理员后台的界面中，管理员可以对数据进行处理，查询统计和修改数据问题。为使用户查询客户信息应该快捷方便、操作简单和界面直观，数据库所设置的表格需要丰富多样，并需要提高对查询数据进行分类统计，设计快速查询模块。关于数据修改问题，要做到信息数据种类和数量上容易扩充和修改。

（2）数据管理分析

在 J2EE 项目中，与数据库的交互方式目前流行的主要有：Hibernate, Spring, JDBC, ibatis 等，本系统采用的是将 Hibernate 集合在 Spring 框架中，采用的是 MySQL 数据库，在后台数据的管理上，保证数据录入方式的多样性，可以直接通过 SQL 语句进行插入数据，也可以在数据表或者先在 Word、excel 表中导入数据，并可以直接对数据表中的数据进行增删改查等基本操作。

（3）系统基本功能分析

本系统是以仓单质押管理为背景的，在实际应用中，作为用户的用户，可以对客户的基本信息进行添加修改等，对销售信息进行实时的更新管理操作，并对债权的明细和财务进行相应的管理，能保证系统的基本功能能符合现实需求，具

有可操作性，流畅性和简洁性^[7]。

2.3.2 非功能性需求分析

本项目开发拟采用的设计语言为 java，数据库服务器为 MySQL Server5.5，辅助的开发工具有：Myeclipse。java 是一门优秀的面向对象开发设计语言，对于 WEB 应用程序的开发，其有着自身独特的优势，如封装、继承和多态。可大大缩短项目的开发周期，开发出的代码执行效率高，也降了应用程序开发的难度。

系统软件均可在 Windows98 , Windows ME, Windows2000 以及 Windows xp 下安装使用，其中还贷和信息查询端只需浏览器即可。

2.4 系统详细设计

本系统使用 Spring 框架进行系统的框架搭建，很好的对代码进行了分层管理，并实现全部相应模块的功能。其中涉及到如页面展示、代码与数据库交互、数据传输等方面的实现。展示层的页面获取数据传输至控制层，由控制层中相应的控制器选择后台逻辑代码的处理器，再转至服务层，服务层从数据持久层中取出已经与数据库交互完成后获得的对象并进行逻辑处理，封装至视图对象，返回页面，由 Spring 的视图解析器进行解析并在页面上展示。而其中，最核心的业务处理代码是服务层的逻辑代码，而所进行的操作基本是对数据库中数据进行的增删改查。

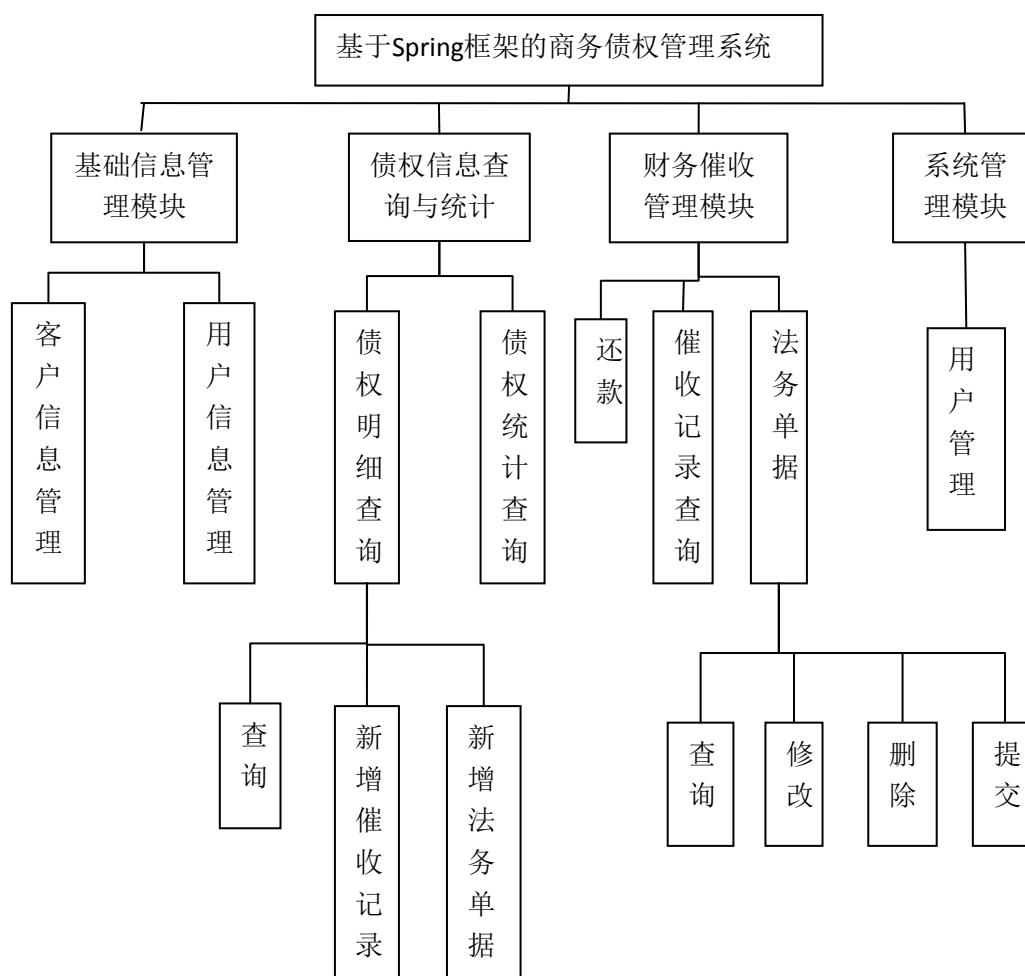


图 2-1 为系统总体框架设计图

商务管理系统以仓单质押为背景，包括数据库设计和应用程序两部分，其中应用程序由基础信息管理、债权信息查询及统计、财务催收管理和系统管理四大模块组成。

图 2-1 给出了商务债权管理系统的总体功能框架图。

从图 2-1 可以看出该系统主要由基础信息管理模块、债权信息查询与统计模块、财务催收管理模块和系统管理模块组成，其中基础信息管理模块用于管理客户和用户的信息，对其进行增删改查的操作；债权信息查询与统计模块用于债权信息明细和债权信息统计的查询以及管理；财务催收管理模块用于对交易财务的催收以及法务的管理；系统管理模块主要是用户的管理，本系统的主要功能是在实际的交易中，能记录交易，并对债权进行管理，通过设置标签，催收债权的方式主要是通过短信或者邮箱的形式，对买家进行债务催收通知，以防过期或者遗漏，同时具有自动性，另外可以通过本系统，记录买家的信用指数，通过数据挖掘技术，分析买家类型，避免信用度差的买家，同时发现潜在的有信用的买家。

设计一个系统的最开始使用用例图来帮助系统建模。用例图是由参与者，用例以及之间关系所构成的视图，用来描述系统的功能一目了然。使用用例图对系统进行建模会使得系统模块和功能十分清晰，方便模块化的处理，有利于有效的编程。

系统的主界面框架设计应如图 2-2 所示：



图 2-2 系统主界面框架

系统后台管理存储信息，系统前台界面显示系统功能，如用户登录认证、整机销售信息、催收记录、法务单据等信息。

系统管理员拥有本系统的最高管理权限，可以对代理用户进行增删改查操作，并且能对系统进行更新、维护等操作。

图 2-3 为系统管理员的功能用例图。

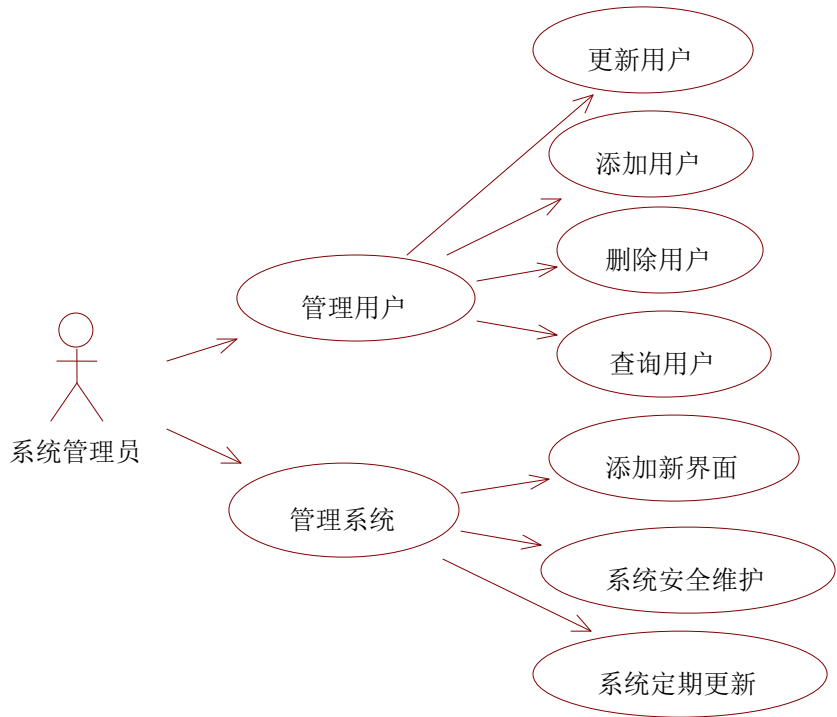


图 2-3 系统管理员用例

图 2-4 为用户管理员的用例图：

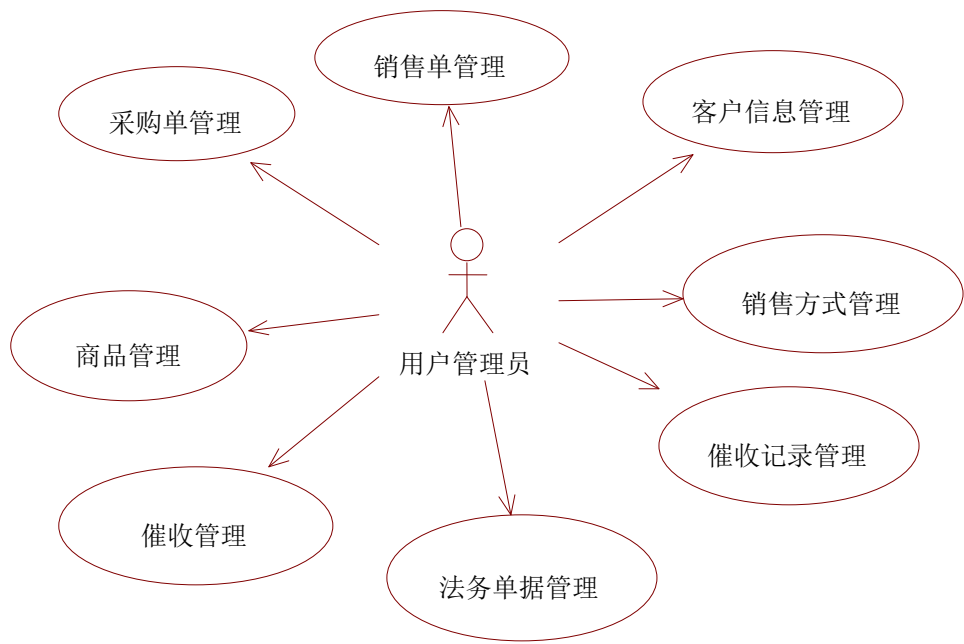


图 2-4 用户管理员

用户是主要操作系统的人员，主要负责客户信息的管理、和销售有关的管理，比如销售方式，销售单的管理等，同时也负责管理采购单和库存商品，并能将采购单和商品库存的消息实时传递给客户分析员，另外根据系统的功能需求，用户用户还需要对债权进行管理，主要通过管理催收和法务单据，以及催收记录等。

图 2-5 为客户分析员的用例图：

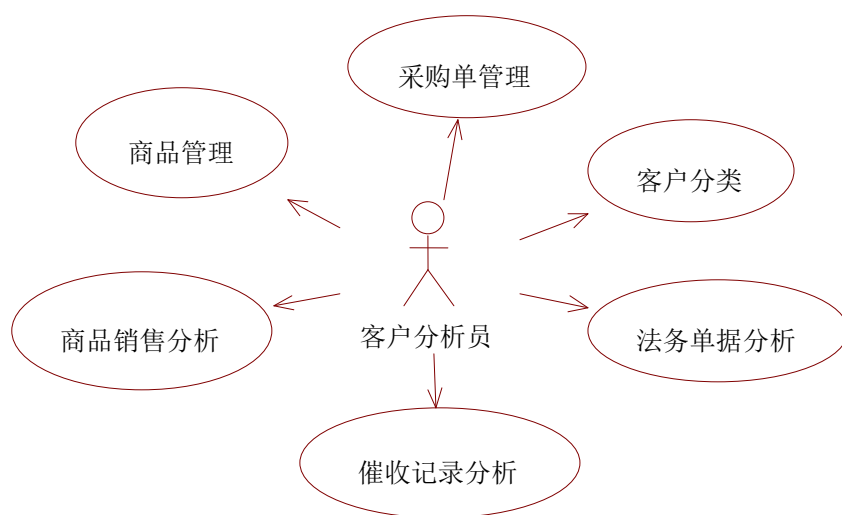


图 2-5 客户分析员

由图 2-5 可知，客户分析员主要完成的是数据挖掘的相关工作，对客户数据、销售的法务单据和催收记录进行分析，从而将客户分类，以便找出能长期稳定合作的客户，同时对商品销售分析，确定采购单中商品的种类和数量，以适应市场需求。

图 2-6 为系统总体框架的用例图：

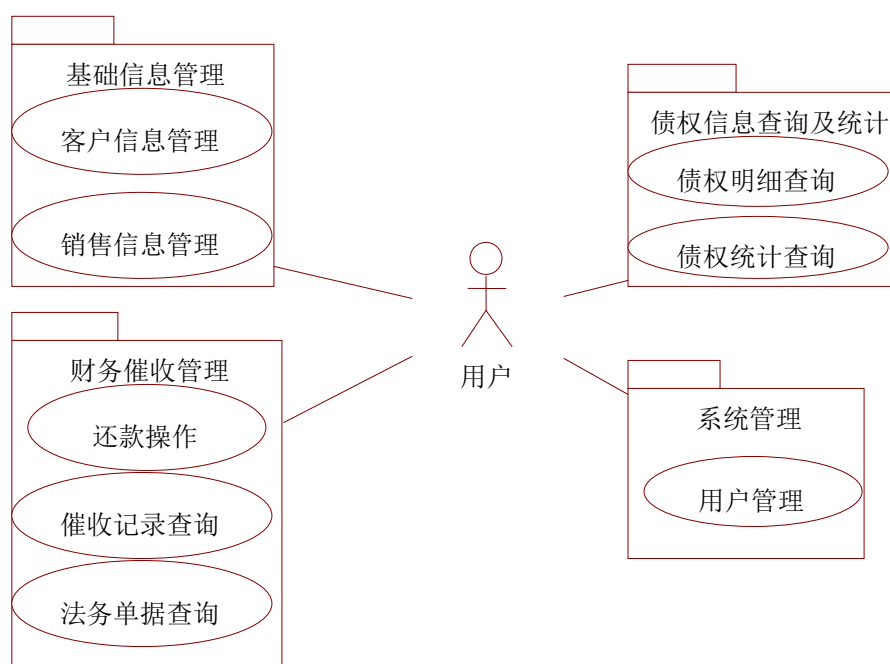


图 2-6 系统用例图

下面使用基础信息管理中的客户信息代码来说明服务层代码是如何完成对数据库中数据的增删改查功能的。

本系统主要涉及到四个功能模块，分别是新增功能模块、删除功能模块、修改功能模块、查询功能模块。

2.4.1 新增功能设计

2.4.1.1 模块简述

本模块包括整机新增，用户新增，客户新增和交易信息新增等四大块。

(1) 整机新增主要针对的是整机的进货管理以及信息新增，整机的信息包括产品类型、产品型号、产品价格、产品生产日期等信息，下图是整机进货的活动图。

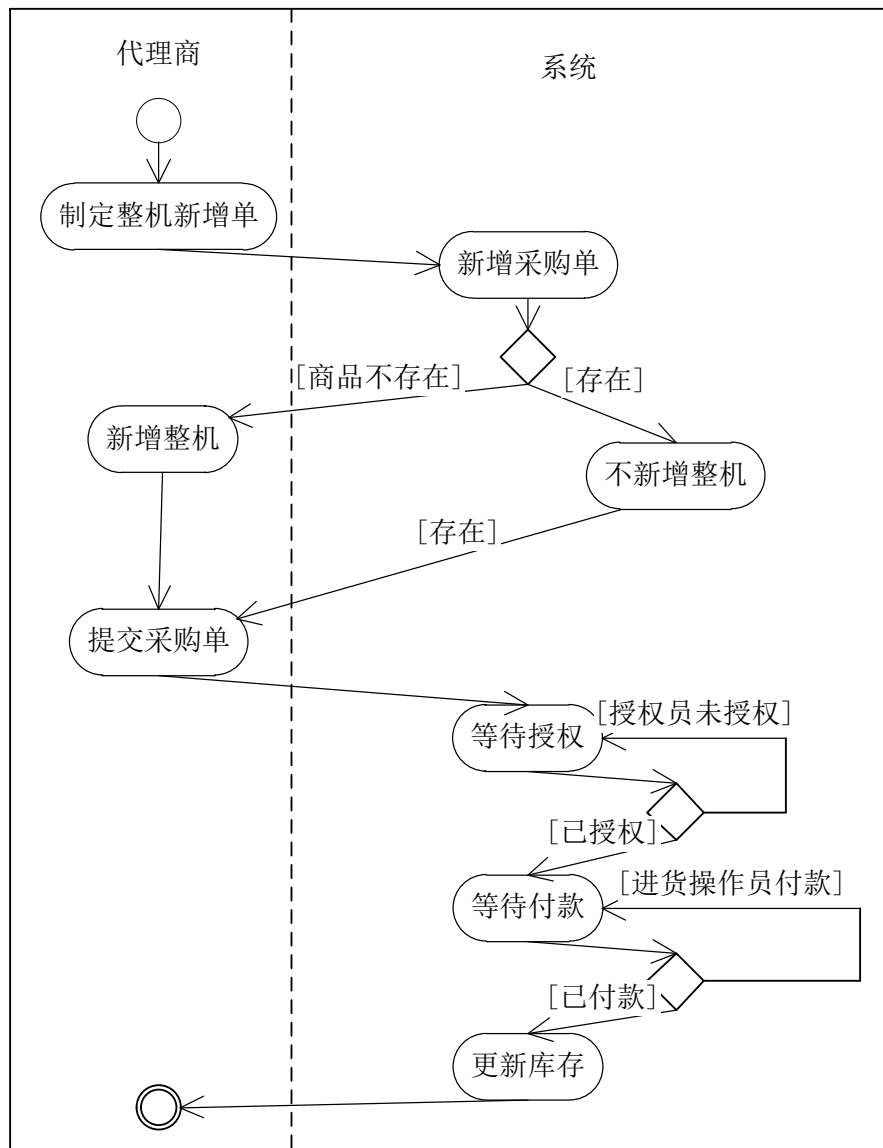


图 2-7 整机进货活动图

(2) 用户新增主要针对的是用户信息的新增管理功能，用户的信息涉及到用户ID、用户名字、用户密码、用户性别、用户类型、用户电话、用户邮箱、用户地址等详细信息，下图为用户新增管理的流程图。

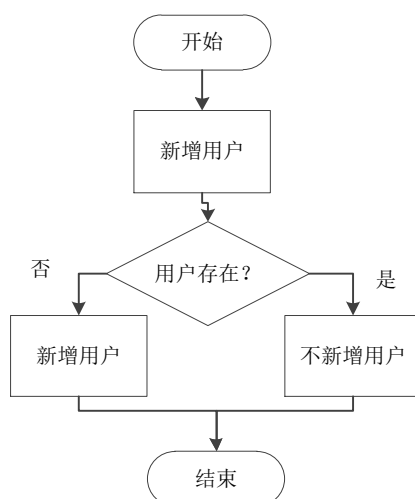


图 2-8 用户新增流程图

(3) 客户新增主要针对的是客户信息的新增管理功能，客户的信息主要包括客户ID、客户名称、客户类型、客户性别、证件号证据复印件、证件类型、户口地址、户口薄、照片、结婚证、配偶照片、配偶身份证复印件、配偶证件类型、配偶户口薄、配偶单位职务、配偶照片、工作地点、邮编、EMAIL、联系电话等详细信息。下图为客户新增管理的流程图。

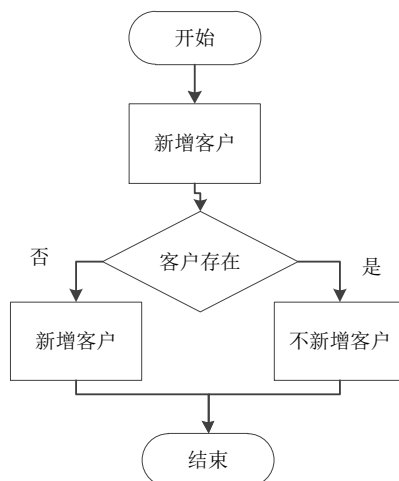


图 2-9 客户新增流程图

(4) 交易信息新增主要针对的是整机销售信息、还款记录、催还记录和法务单据的新增管理功能，其中，整机销售信息包括整机销售ID、客户ID、整机编号、产品类型、产品型号、销售日期、首付款、合同价等信息；信用销售期数还款记录包括还款记录ID、客户ID、整机销售ID、还款类型、应还金额、应还日期、实际还款金额、实际还款日期、逾期原因、状态、组织机构ID等信息(还款信息

还和客户信息、整机相关联) ;c催还记录包括记录ID、整机销售ID, 邮件短信发送历史, 同时还包含催收时间、催收人、催收类型和催收内容, 通过这些信息以及方式, 达到催收的效果; 法务单据包括法务单据ID、整机销售信息ID、法务编号、申请内容、生成法务单据时间、提交申请时间、单据状态等, 对法务单据的管理主要是为进销存货提供参考。

2.4.1.2 模块详细设计

模块详细设计以用户管理为例, 下图为级别管理和用户管理的活动图:

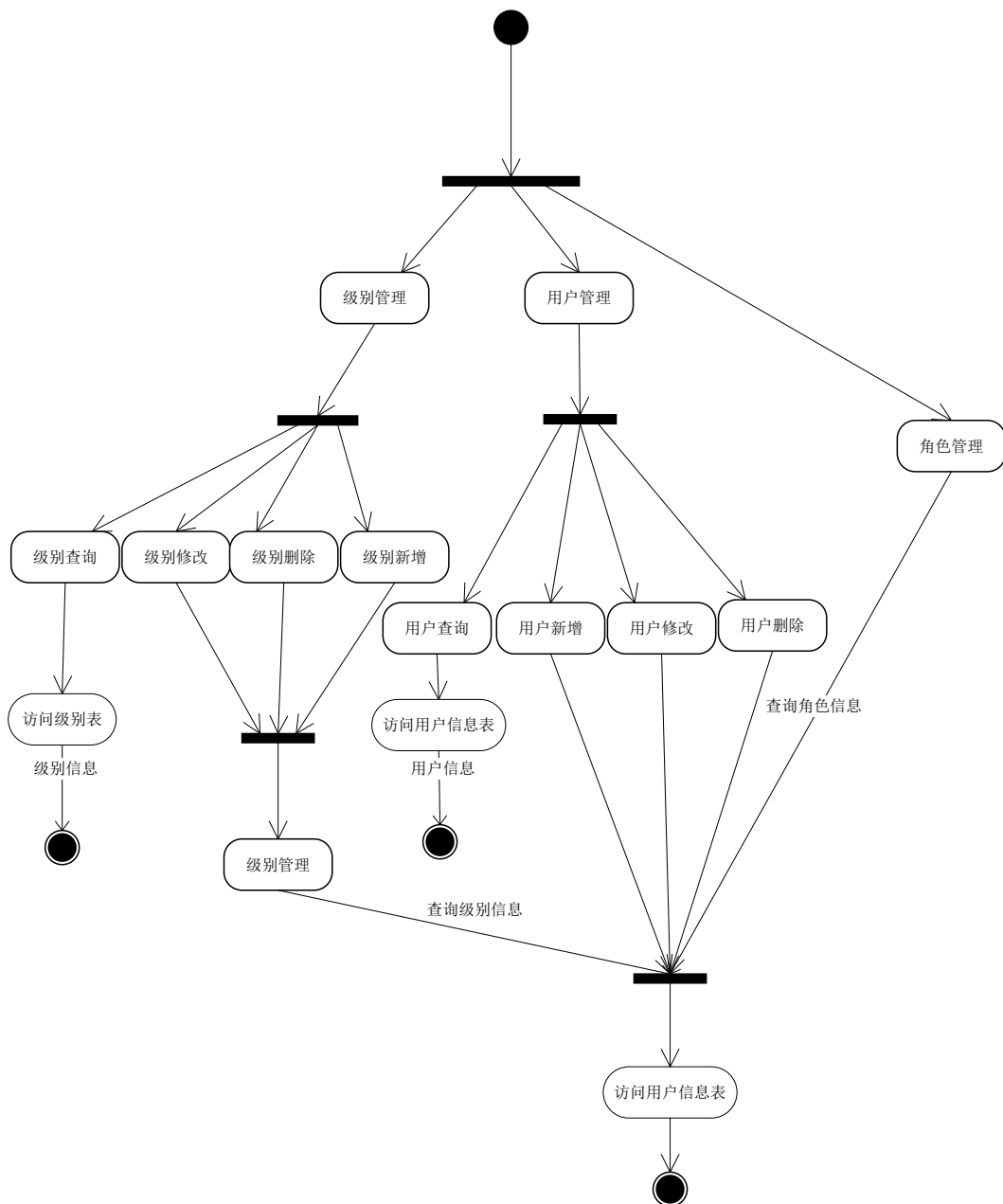


图2-10 级别管理和用户管理的活动图

其中用户添加的代码如下所示:

```
@RequestMapping(value="/addUser")
public ModelAndView addUser (@ModelAttribute addUser,HttpServletRequest
request){
    long addStatus=0;
    User login_user=
        (User)request.getSession().getAttribute(AuthenticationFilt
r.AUTH_INFO);
    addUser.setCloudId(login_user.getCloudId());
    addStatus=userservice.doAddUser (user);

    Map deMap=new HashMap();
    deMap.put("as", addStatus);
    MappingJacksonJsonView jsonView=new MappingJacksonJsonView();
    return new ModelAndView(jsonView,deMap);
}
```

新增用户信息代码

在页面中, 触发新增用户信息之后转到控制层, 在控制层中执行所对应的方法中的代码。页面上的相应内容由 Spring 框架自行封装进一个 HttpServletRequest 对象中, 控制层直接从该对象中提取数据, 再由 Spring 的数据持久层与数据库进行连接并自动映射表中字段, 对数据库进行新增操作, 返回新增操作的结果至页面, 页面由返回结果进行相应的响应, 显示是否添加成功。由于使用 Spring 框架, 所以在页面中相应的属性的文本框的 name 属性需要与数据库中字段名使用相对应的命名方式, 这样才能让 Spring 框架的解析器能够读出是对应的属性, 不会导致无法写入数据库中^[31]。

2.4.2 删除功能设计

2.4.2.1 模块简述

本模块包括整机删除, 用户删除, 客户删除和交易信息删除等四大块。

四大分块所包含的信息与2.1.1中所描述的信息相同。

2.4.2.2 模块详细设计

模块详细设计以删除客户信息为例，如以下代码段所示：

```
@RequestMapping(value="/deleteCustomer")
public ModelAndView deleteMachinesales(@RequsetParam("id") long
custId) {
    boolean ifDel=false;

    Customer customer=new Customer();
    customer.setId(custId);
    ifDel=customerservice.doDeleteCustomer(customer);

    Map deMap=new HashMap();
    deMap.put("ifDel", ifDel);
    MappingJacksonJsonView jsonView=new MappingJacksonJsonView();
    return new ModelAndView(jsonView,deMap);
}
```

删除客户信息代码

在页面上触发删除客户信息之后，页面发送请求至控制层，由控制层进行识别并转发至服务层，服务层代码执行相应方法中的代码。页面上展示数据时会将该条记录在数据库中的 ID 取出并存放于页面上，在请求删除时，将该条数据的 ID 与请求一同发送至服务层，服务层根据 ID 对数据库相应的数据进行删除操作，并返回相应的处理结果给页面，页面根据所返回的处理结果进行展示。数据库中，每条记录都有其对应的 ID，使用 ID 查找数据库中的数据最为方便，因此在删除对应数据时使用 ID 作为索引快速定位数据库中数据并进行删除操作^[32]。

2.4.3 修改功能设计

2.4.3.1 模块简述

本模块包括整机修改，用户修改，客户修改和交易信息修改等四大块。

四大分块所包含的信息与2.1.1中所描述的信息相同。

2.4.3.2 模块详细设计

模块详细设计以修改客户信息为例，如以下代码段所示：

```
@requestMapping(value="/modifyCustomer")
```

```

public ModelAndView modifyCustomer(@ModelAttribute Customer
customer,HttpServletRequest request){
    boolean modifyStatus=false;

    User
login_user=(User)request.getSession().getAttribute(AuthenticationFilt
r.AUTH_INFO);
    customer.setCloudId(login_user.getCloudId());
    modifyStatus=customerservice.doModifyCustomer(customer);

    Map deMap=new HashMap();
    deMap.put("ms", modifyStatus);
    MappingJacksonJsonView jsonView=new MappingJacksonJsonView();
    return new ModelAndView(jsonView,deMap);
}

```

修改客户信息代码

在页面上触发修改客户信息后，页面将数据与请求发送至控制层，由控制层根据请求转发至服务层，并执行相应方法中的代码。与删除操作相同，传回服务层时会将对数据在数据库中的ID一同传回，根据ID快速定位至对应条目，并将页面上所修改的数据更新至数据库，并返回处理结果，页面根据处理结果进行相应的展示。

2.4.4 查询功能设计

2.4.4.1 模块简述

本模块包括整机查询，用户查询，客户查询和交易信息查询等四大块。
四大分块所包含的信息与2.1.1中所描述的信息相同。

2.4.4.2 模块详细设计

模块详细设计以查询客户信息为例，如以下代码段所示：

```

@RequestMapping(value="/list/data")
    public ModelAndView listCustomer(DataGridModel dgm, Customer
customer,HttpServletRequest request){

```



```

    User login_user=(User)request.getSession().getAttribute(Authent
icationFilter.AUTH_INFO);
    List<Customer> customerList=null;
    long totalCustomer=0;
    customer.setCloudId(login_user.getCloudId());
    //获取客户 list 和客户总条数
    customerList=customerservice.doQueryCustomerList(customer,dgm);
    totalCustomer=customerservice.totalCustomer(customer);
    //将查询结果封装成指定的 map
    Map mnMap=new HashMap();
    mnMap.put("total", totalCustomer);
    mnMap.put("rows", customerList);
    //使用 jackson 将 map 转换为 json 对象
    MappingJacksonJsonView jsonView=new MappingJacksonJsonView();
    return new ModelAndView(jsonView,mnMap);
}

```

查询所有客户信息代码

在页面加载时需要展示所有已有客户的信息，因此在加载时就需要从数据库取出所有客户信息的列表。页面发送请求至控制层，控制层转发至服务层，服务层执行相应方法中的代码。通过数据持久层与数据库交互取出所有客户的信息，并且封装成 Map 并封装入 ModelAndView 中传回页面。Spring 框架使用内部解析器分析解码传回的 ModelAndView 对象并进行数据展示。

```

@RequestMapping("/detail")
public ModelAndView custInfo(@RequestParam("id") long custId){
    Customer customer=new Customer();
    Map custMap=new HashMap();
    customer=customerservice.customerInfo(custId);

    custMap.put("custInfo", customer);
    MappingJacksonJsonView jsonView=new MappingJacksonJsonView();
    return new ModelAndView(jsonView,custMap);
}

```

查询单一客户信息代码

在选中某一客户查看其详细信息时，只需要展示所选客户信息即可，因此不同于查询所有客户信息，另有方法查询单一客户信息。同理在页面触发请求后将该用户 ID 与请求发送至控制层，控制层转发至服务层，服务层执行相应方法中的代码。根据 ID 从数据库中快速定位并取出对应数据，封装如 ModelAndView 对象中，再由 Spring 框架的解析器进行解析并展示。

2.5 本章小结

本章通过问题描述简要介绍了商务债权管理系统的主要功能、问题性质、开发目的和系统组成；从软件工程的角度，阐述了该系统的可行性；并从功能性和非功能性两个方面进行需求分析，完成系统的需求分析。

本章介绍了系统的详细设计，介绍了系统的四大功能模块的核心结构和关键代码，最后给出了代码的整体框架。

根据前文介绍的Struts、Spring和Hibernate三种著名的框架进行分层架构模式，将代码整体框架分为表示层、业务层和持久层进行设计。

图2-11为表示层代码结构图：

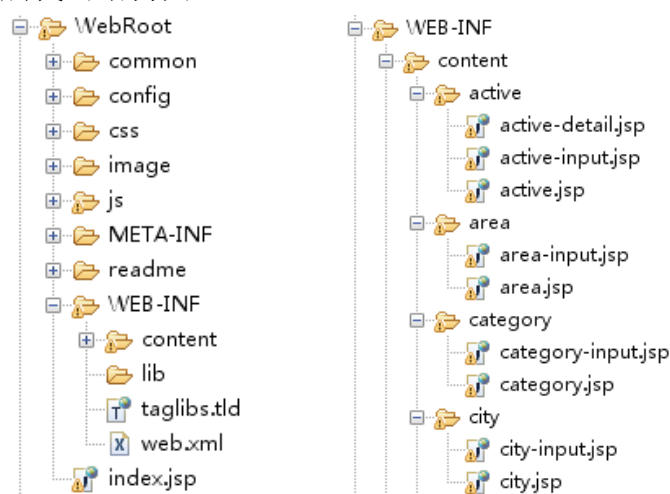
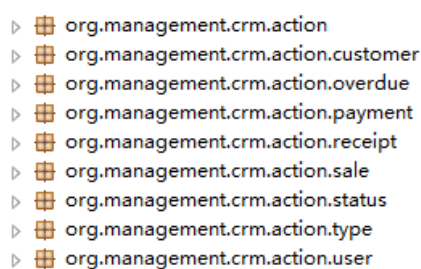


图 2-11 表示层

图2-12为业务层的代码结构图：



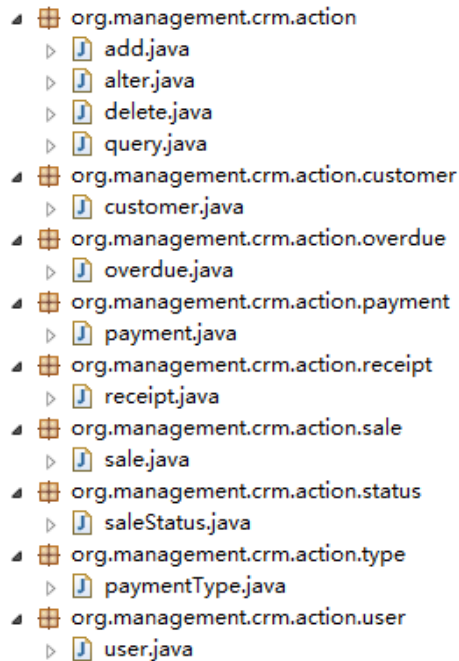


图 2-12 业务层

图2-13为持久层代码结构图:

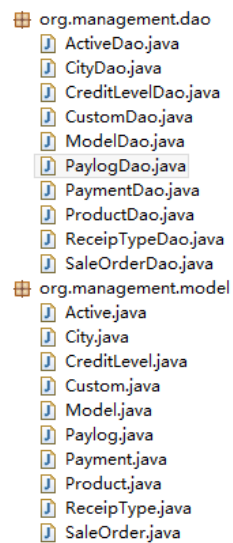


图 2-13 持久层

第三章 系统总体设计

本文前面分析了债权管理的研究现状，认识到开发商务债权管理系统对于企业债权管理工作的重要性，在此基础上，我们选择的是 B/S 的开发模式以及 Spring 框架，采用 JAVA 语言编写，设计实现了基于仓单质押背景下的债权管理系统，本章节将对该系统的设计和实现过程做详细的阐述。

为了使程序开发能有序的进行，在程序开发之前，需要先进行需求分析，其中包括系统界面和数据库设计分析，系统的具体开发将按照系统的总体框架图进行。本文开发的债权管理系统可以实现办公室人员对债权的网络化管理和对客户的管理，催收债权和实时更新债权的工作，力求与实际相结合，且具有查询、管理等方面的功能，旨在推动企业办公的现代化，提高员工的工作效率，为企业减少管理费用。

3.1 设计原则

（1）可靠性和稳定性

可靠性意味着该系统在测试运行过程中避免可能发生故障的能力，且一旦发生故障后，具有的解脱和排除故障的能力。在设计方面，采用容错设计。在设备选型方面，为保证软、硬件的可靠性，采用相对成熟的技术和产品。

（2）方便管理和维护

在不改变系统运行的情况下，用户可以对系统进行增删改查等操作；易维护是应用系统成功与否的重要因素，它包含两层含义：故障易于排除，日常管理操作简便。

（3）容易理解

在系统可修改和可维护之后，提出系统的大众性，不仅要具有相关文档清晰可读的特点，而且要求系统本身具有简明的结构。

（4）实用性

设计本系统，应首先考虑本系统的功能和实际商务活动中的需求是否一致。

（5）开放性和规范性

为保证各供应商产品的协同运行，同时考虑到投资者的长远利益，并结合相关的国际标准或工业标准执行。另一方面，由于本系统是一个综合性系统，在系统设计和建设初期应着手参考各方面的标准与规范，并且应遵从该规范各项技术规定，并做好系统的标准化设计与管理工作。

(6) 可测试性

本系统应该具有可测试性，通过设计测试集合，确保系统得到全面的检测。

3.2 架构设计

3.2.1 债权的相关概念

债权是得请求他人为一定行为（作为或不作为）的司法上权利。属于权利义务相对原则，相对于债权人债务，即必须为一定行为（作为或不作为）的司法上义务。因此债之关系本质上即为一司法上的债权债务关系，债权和债务都不能单独存在，否则即失去意义。和物权不同的是，债权是一种典型的相对权，只在债权人和债务人之间发生效力，原则上债权人和债务人之间的债之关系不能对抗第三人。

企业债权管理指在公司与客户开始接触到与客户终止合作，贯穿于这个过程中，企业采取防止坏账损失的一系列管理方法，管理涉及的范围主要包括企业经济活动中发生的暂收、预收、应收、应付等。

债权管理对于一个企业来说，是非常重要的，原因有以下两个：一、在现代交易中，全款销售占有所有销售中的比例越来越小，其地位慢慢被信用销售取代；二、在信用销售的流程中，产品的售出只是销售流程的开始，债权管理通过对回款风险的管理控制，以保证整个销售流程的安全完成，保证全部贷款按时回收。

常用的债权管理手段有：

(1) 防范风险

防范风险可以有效的预防和避开债权销售中的风险，在销售之前，通过对新老客户进行调查，充分分析客户的信用程度，避开高风险、信用度低的客户，并在签订合同时，严格按照合同和客户签约，减少不必要的纠纷和麻烦。

(2) 控制风险

根据签订的合同，企业可以采用一定的方式督促客户还款，在本系统中，对于分期付款的用户，主要采用电话、邮箱以及上门督促的方式，并对客户的还款程度进行记录，将客户分为优质的可信赖的客户和拖欠客户，以便于未来的长期合作。

(3) 利用风险

对于企业来说，哪里有风险哪里就有商机，虽然债权销售有一定的风险，但是这种销售同时会给企业带来大量的商机，设置优良、精干的债权管理部门，充分利用风险，化风险为商机。

目前企业通常采用的债权管理方式有：对业务员实施回收贷款的奖惩制度、将客户进行划分信用等级、设定信用额度等，虽然能达到一定的效果，但是所耗费的人员众多，管理松散，缺乏一定的标准。

3.2.2 MVC 架构

MVC 是模型(Model), 视图(View)和控制 (Controller)的缩写，其目的是实现 Web 系统的职能分工，如图 3-1 所示：

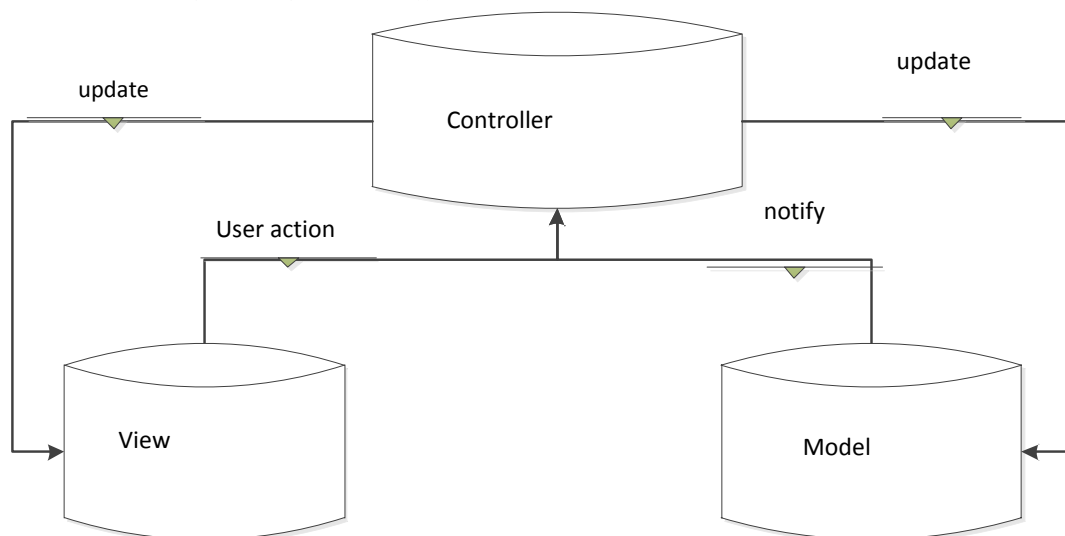


图 3-1 MVC 概念图

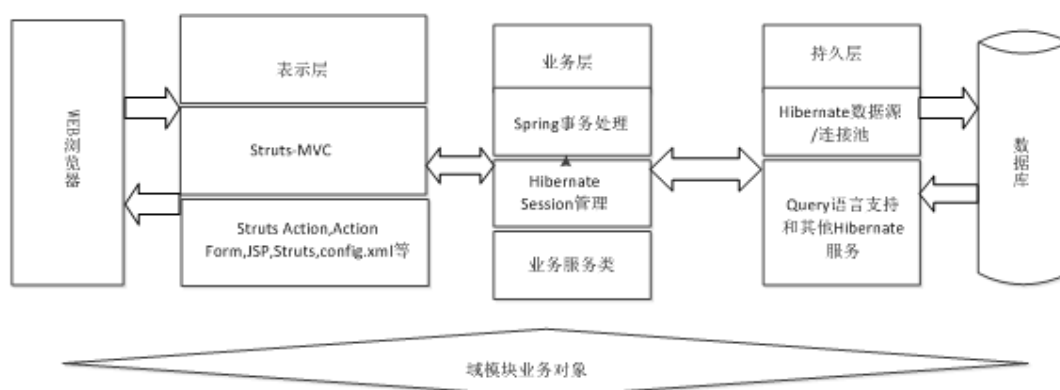
MVC 是一个设计模式，能强制使应用程序的输入、处理和输出分开。模型、视图、控制器作为 MVC 应用程序的核心部件，在各自处理自己的任务的同时，也互相合作，相辅相成。

Controller 层是 Model 与 View 之间沟通的桥梁，它可以分派用户的请求并选择恰当的视图以用于显示，同时它也可以解释用户的输入并将它们映射为模型层可执行的操作。

Model 层实现系统中的业务逻辑，用于表示企业数据和业务规则，拥有最多的处理任务，通常可以用 JavaBean 或 EJB 来实现，由于模型与数据格式无关，一个模型可以被多个视图共用，可以减少代码的重复性。

View 层用于与用户的交互，通常用 JSP 来实现，MVC 可以为应用程序处理很多不同的视图，并且在视图中没有涉及到真正的处理，视图只是作为一种输出数据并允许用户操纵的方式。

MVC 的设计模式如图 3-2:



3-2 MVC 设计模式

表示层主要用于向客户端用户提供 GUI 交互，包含代码、用户交互 GUI，数据验证等，允许用户在显示系统中输入和编辑数据，同时为系统提供数据验证功能。

业务层用于执行业务流程和制定数据的业务规则，包含业务规则处理代码，即程序中与业务相关专业算法、业务政策等，主要面向业务应用，为表示层提供业务服务。

持久层负责与数据库管理系统（如数据库）之间的通信，包含数据处理代码和数据存储代码。

MVC 的框架有多种类别，本文主要用到 Spring 框架及其相关技术。

图 3-3 为 Spring 响应过程：

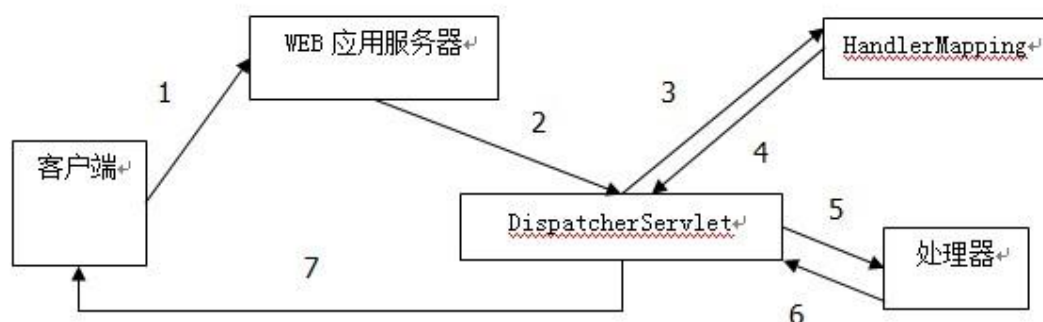


图 3-3 Spring 响应过程

从图中可以看出，从接受请求到返回响应，Spring MVC 通过一个前端 Servlet 接受所有的请求，DispatcherServlet 就是这个前端 Servlet。

过程由客户端发出的 HTTP 请求开始，Web 应用服务器接受到请求之后，如果在 DispatcherServlet 中有匹配的请求映射路径，则 Web 容器就会将该请求转交

给 DispatcherServlet。DispatcherServlet 接受到请求之后，根据请求信息，及 HandlerMapping 的配置找到处理请求的处理器。并由 Spring MVC 框架级接口进行调用相应的处理器。处理器完成业务逻辑的处理后返回一个 ModelAndView 给 DispatcherServlet，ModelAndView 包含了视图逻辑名和模型数据信息。DispatcherServlet 解析 ModelAndView，将视图对象解析出来，得到真实的视图对象 View，DispatcherServlet 使用这个 View 对象对 ModelAndView 中的模型数据进行视图渲染，返回响应消息给客户端。客户端得到响应消息，并展示。这就是一个完整的 Spring MVC 的工作流程。

3.2.3 Spring 框架

Spring 是一个开源框架，它由 Rod Johnson 创建。它是为了解决企业应用开发的复杂性而创建的。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 Java 应用都可以从 Spring 中受益。

Spring 是一个轻量级的控制反转 (IoC) 和面向切面 (AOP) 的容器框架。Spring 框架如图 3-4 所示：

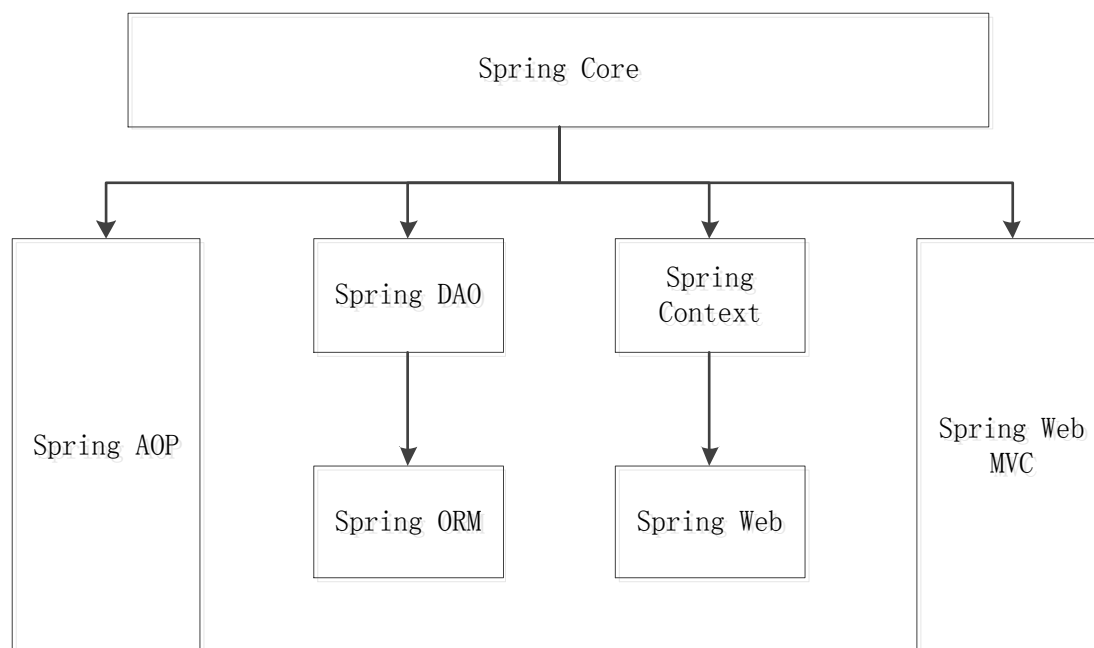


图 3-4 Spring 框架图

3.2.3.1 IOC 容器

IoC (Inverse of Control: 控制反转) 是 Spring 容器的内核，AOP 等功能都基于 IoC。IoC 包涵很多内涵，如代码解耦、设计模式、代码优化等等问题。

许多应用的业务逻辑都需要多个类之间的彼此合作来达成，这导致了每个对象都需要与其依赖的对象进行引用。如果这个获取过程需要通过代码得以实现，则将导致代码的高度耦合亦或实现困难。

IOC 也可称为“依赖倒置原理” (Dependency Inversion Principle)。IOC 模式是建立在面向对象的基础之上的一种用来解决组件之间依赖关系、配置及生命周期的设计模式。在以往的程序设计中，上层模块都是在代码中声明下层模块的实例或直接调用下层模块的方法。一旦下层模块的方法改变，相应的上层模块的代码也需相应的修改，造成了上层模块依赖于下层模块。而把上层模块中的方法取出来定义成接口，上层面向接口编程，就产生了依赖倒置。

通过框架把程序的运转流程由客户端程序中剥离出来，就是控制反转。IOC 的原理就是使用控制反转，当对象被创建的时候，由控制系统中所有对象的外界实体，将所依赖的对象的引用传递给它。也可以说是依赖被注入对象之中。依赖注入实现将上层模块和下层模块实现联系在一起，把框架的流程和功能结合在一起。IOC 采用的最基本技术就是“反射”，也就是根据给出的类名来生成对象。这种方式的编程可以在对象生成的时候才决定要生成哪一种对象。

3.2.3.2 AOP 技术

AOP 是 Aspect Oriented Programming 的简称，译为面向切面编程。按照软件重构的思想，如果多个类中存在相同代码，则应该考虑定义抽象类，将重复代码提取到抽象类中。而同时，也会出现不同的类有相同的代码结构却没有相同的代码段，这种情况就导致了无法通过抽象父类的方法消除重复性横切代码的结果。而 AOP 则通过横向抽取机制为这类例子找到了解决方法，AOP 将分散在各个业务逻辑代码中的相同代码，通过横向切割的方式抽取到一个独立的模块中，使得代码一目了然，清晰易懂。AOP 中引入了例如：连接点、切点、增强、目标对象、引介、织入、代理、切面等概念。

(1) 连接点：程序执行到某个特定位置：如类的初始化前、初始化后、方法抛出异常后等边界性质的特定点就被称为连接点。Spring 支持方法连接点，即在方法调用前后、抛出异常时这些连接点织入增强。

(2) 切点：AOP 通过切点定位特定的连接点。通过 `org.springframework.aop.Pointcut` 接口进行描述，使用类和方法作为连接点的查询条件，Spring AOP 的规则解析引擎解析切点所设查询条件，找到对应的连接点。

(3) 增强：增强是指在目标连接点上织入的一段程序代码。在找到对应的连接

点后，执行织入代码，达到增强的逻辑。

(4) 目标对象：增强逻辑的织入目标类就是目标对象。在 AOP 的帮助下，横切逻辑上的代码可以由 AOP 动态织入到特定的连接点上，使得目标对象只需要实现非横切逻辑的代码即可。

(5) 引介：引介是一种特殊的增强，它为类添加一些属性和方法。通过 AOP 的引介功能，我们可以为一个类实现并没有实现的某个接口的实现逻辑，让业务类成为接口的实现类。

(6) 织入：织入是将增强逻辑添加到目标对象上的过程。AOP 将目标类、增强或者引介通过 AOP 连接到一起。

(7) 代理：一个类被 AOP 织入增强后，就产生除了一个融合了原类和增强逻辑的代理类。

(8) 切面：切面由切点和增强组成，Spring AOP 是负责实现切面的框架，它将切面所定义的横切逻辑织入到切面所指定的连接点中。

3.2.3.3 Spring 的 DAO 概念

DAO (Data Access Object) 是用于访问数据的对象。在 UserDao 中定义访问 User 数据对象的接口方法，业务层通过 UserDao 操作数据，并使用具体持久技术实现 UserDao 方法，这样业务层和具体持久化技术就实现了解耦，这是典型的 DAO 应用实例。

DAO 层抽象出来，我们就可以很容易的构造模拟对象，方便代码的测试。例如团队开发时，底层代码还没有实现，抽象出 DAO 层后，展示层程序员只需要手动定义一个模拟的 DAO 对象就可以进行展示层的实现，并不需要等到底层代码完全实现过后再进行开发。Spring 本质上希望以统一的方式整合底层的持久化技术。Spring 提供了统一的异常体系，则可定义出与具体实现技术无关的 DAO 接口，用以整合到相同的事务管理中。

3.2.4 Struts 和 Hibernate 框架

在建立软件系统的高层结构中，常用 Struts、Spring 和 Hibernate 三种著名的框架进行分层架构模式设计，如图 3-5 所示：

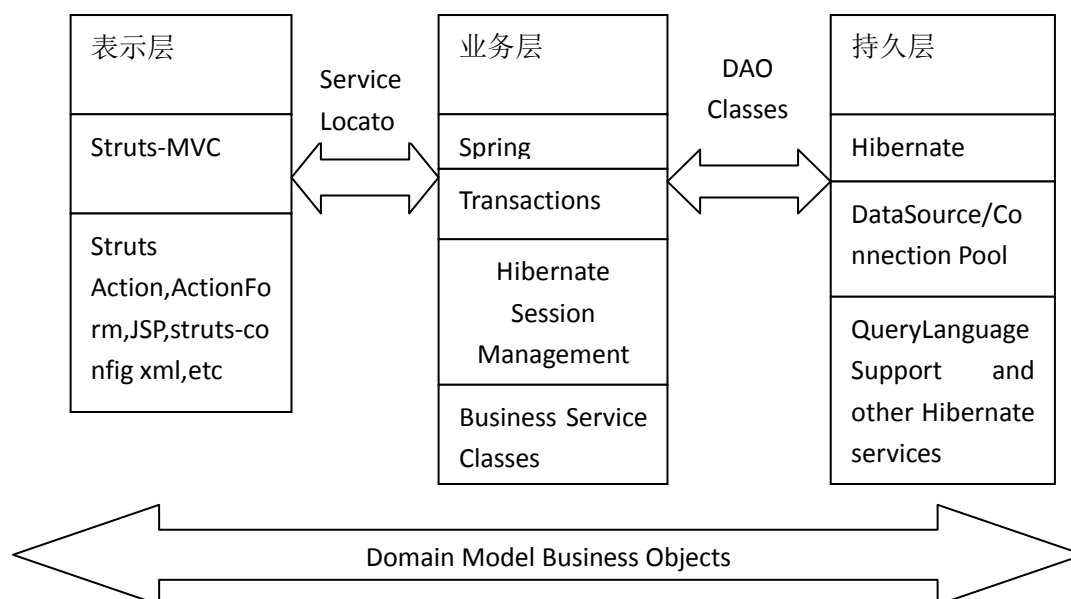


图 3-5 Struts-Spring-Hibernate 架构

一般来讲，一个典型的 Web 应用的前端应该是表示层，Struts 主要负责的是：管理用户的请求，并做出相应的响应；提供一个流程控制器，委派调用业务逻辑和其他上层处理；处理异常；为显示提供一个数据模型；用户界面的验证等。

持久层是典型的 Web 应用的后端，Hibernate 为 Java 提供持久化机制和查询服务，其持久化对象是基于 POJO(Plain Old Java Object) 和 Java 集合 (collections) 的，Hibernate 主要负责的是：如何查询对象的相关信息；如何存储、更新、删除数据库记录；支持数据库的父表/子表关系、事务处理、继承和多态等[8-9]。

3.2.5 MySQL 数据库和 B/S 结构

3.2.5.1 MySQL

由于开源数据库具有免费使用、配置简单、稳定性好、性能优良等特点，使其在中低端应用中占据很大的市场份额，MySQL 就是开源数据库的杰出代表。20 世纪 90 年代开发的关系型数据库——MySQL，是一款免费开源、小型的数据库管理系统。随着该数据库功能的不断完善、性能的不断提 高、可靠性的不断增强，被广泛应用于 Internet 上的中小型网站中。

MySQL 是一个快速的、多线程、多用户和健壮的 SQL 数据库服务器，MySQL 服务器支持关键任务、重负载生产系统的使用，也可以将其嵌入到一个大配置的软件中去，相比于其他数据库管理系统，MySQL 具有以下优势：

- (1) MySQL 是一个关系型数据库管理系统
- (2) MySQL 是开源的
- (3) MySQL 服务器是一个快速的、可靠的和易于使用的数据库服务器
- (4) MySQL 服务器工作在客户/服务器或嵌入式系统中
- (5) 有大量的 MySQL 软件可以使用^[15]

3.2.5.2 B/S 结构

C/S 和 B/S 是目前网络软件运行的两种主流模式，两种模式的合理结合会起到相得益彰的效果，因为 B/S 和 C/S 有它们各自的优势，侧重表现于不同的方面。C/S 模式主要是在以局域网为基础的运行环境下展开应用的，它受到地域的限制。B/S (Browser/Server, 浏览器/服务器) 模式又称 B/S 结构。它是随着 Internet 技术的兴起，对 C/S 模式应用的扩展。在这种结构下，用户工作界面是通过 IE 浏览器来实现的。B/S 模式最大的好处是运行维护比较简便，能实现不同的人员，从不同的地点，以不同的接入方式（比如 LAN, WAN, Internet/Intranet 等）访问和操作共同的数据^[16]。

基于 B / S 的 Internet 技术，有三项优势是明显的：

(1) 相当稳定的技术平台。在 C/S 时代，用户最苦恼的问题莫过于面对的是众多互不兼容的技术平台。每种技术平台，从用户应用稳定性要求来看，都可以说是昙花一现，给用户留下了无奈与苦恼。而 B/S 的技术保证了它技术的高度稳定性。

(2) 本质上是一种客户机技术。在 C/S 模式下，每种应用的主要部分是在客户机侧。别的不说，光是在众多的客户机侧安装和维护多种应用的多个版本，就叫人受不了。而 B/S 技术不同，千万个用户只是通过浏览器这个统一界面，可享受到无限丰富的，永远在不断变化和发展着的信息服务。Intranet 原则上取消了所有在客户机侧的维护工作。

(3) 提供了异种机、异种网、异种应用服务的联机、联网。通过 Intranet 以技术统一访问异种数据库，已成为现实。从 ISDN 到 ATM，再到现在大行其道的 TCP/IP，正是由于 Internet 的盛行，宽带 IP 将会成为真正的、统一服务的(数据、语音、多媒体)网络基础。

在实际应用中，对于使用软件进行管理所得带的交易数据来说，一个很重大的问题就在于软件的安装。对于一半的办公软件来说，需要用户在使用电脑上安装软件，并将数据逐一输入至软件中进行保存，这样每次使用都只能在一台电脑上使用，若更换电脑则需要重新安装软件并导入数据，会给使用者带来极大

的麻烦。而基于 B/S 结构，整个软件可以在一台服务器上运行和维护，使用者只需要登陆在系统首页，通过身份验证后即可使用系统，数据保存在服务器的数据库，省去了用户重复导入数据的繁琐过程，也降低了软件的使用条件，将软件的应用范围扩大到最大。

对于使用者来说，使用此类系统不需要在不同环境使用时都进行软件的安装和调试，不需要在每个不同环境的软件中导入初始数据，也不会因为电脑损坏或者误操作等情况的发生而丢失重要的数据。在服务器上运行的程序相对来说有较高的稳定性，很少出现死机的问题，就解决了紧急情况需要使用却无法使用的尴尬局面，也很好的对数据进行了私密的保护和保存。使用浏览器就可以进行软件的使用同时也让软件的上手度降到了极低的程度，使用任何连接到互联网的电脑都将可以使用此软件，并且并不需要专业的培训即可简单操作，方便了各种受群对软件的需求。同时在登录时的身份验证也可进行权限设置对查看特定数据的权限进行限定，方便了不同层级使用者的各种需求。

在本系统的设计中，拟采用 B/S 模式相结合的方式，实现通过浏览器进行信息查询、记录浏览。

3.3 模块设计

本系统主要由四大模块组成：基础信息管理、债权信息查询及统计、财务催收管理和系统管理。

图 3-6 为简易总功能模块图：

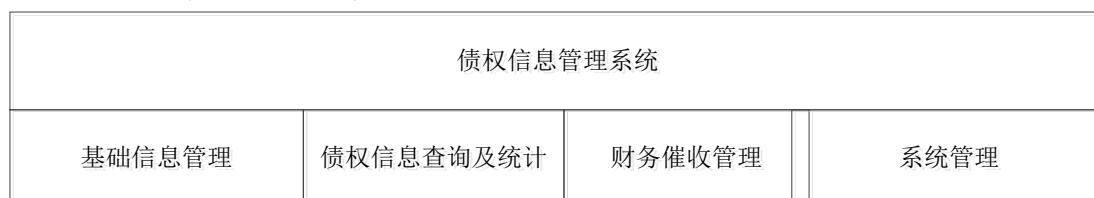


图 3-6 简易总功能模块图

基础信息管理模块：主要用于用户对客户信息和销售信息的录入及管理。

债权信息查询及统计模块：主要用于用户对已有债权信息进行查询和统计管理，并提供相应的催收记录和法务单据查询功能。

财务催收管理模块：主要用于用户对已有债权进行还款登记、催收记录查询、法务单据查询等功能。

系统管理模块：主要用于用户对下属用户帐户分配。

3.3.1 基础信息管理模块

图 3-7 为基础信息管理模块图:

基础信息查询							
客户信息查询				销售信息查询			
新增	修改	查询	删除	新增	修改	查询	删除

图 3-7 基础信息管理模块

基础信息管理模块是帮助用户管理客户信息和销售信息的模块。主要分为客户信息管理和销售信息管理两个小模块。

客户信息管理模块负责管理买家的基本资料，并为每一个客户分配一个 ID，一个 ID 对应一个客户，以方便用户和客户登陆系统进行相关的操作。客户信息管理根据需求对客户的基础信息提供增删改查。用户可以在相应页面对客户进行新增、查询、修改和删除操作，完成对客户基础信息的管理功能。

销售信息管理模块，主要是系统的管理人员对销售的信息进行管理，为每一个销售出去的整机创建一个 ID，以方便管理。销售信息管理模块提供销售信息的录入、修改、查询和删除功能。由于债权销售方式的特殊，在销售信息录入的时候需要输入相应的销售模式，因此在新增时需要用户手动选择销售方式同时产生相应的债权数据并由用户选择相应的还款方式。

3.3.2 债权查询及统计模块

图3-8为债权查询及统计模块图:

债权查询及统计			
债权明细查询			债权统计查询
查询	新增催收记录	新增法务单据	按客户分类债权信息查询

图 3-8 债权查询和统计

债权查询和统计模块主要有债权明细查询和债权统计查询两个小模块。主要

用于用户对债权信息的查询和管理。

债权明细查询中列出所有销售订单所产生的债权信息，用户可以根据需求进行查询。同时，逾期未还的债权信息提供添加催收记录功能，可以帮助用户记录是否已经对该债权条目进行催收。同时对逾期过于严重，催收无果的债权信息提供新增法务单据选项。

债权明细查询主要是包括对客户的债权信息进行查询，和对商品所涉及的债权进行查询；债权统计查询主要是对客户某一时间段，比如一个季度，所有债权的统计，和对所有商品的债权统计。通过债权统计，可以清晰的显示商品的债权情况，结合相应的数据挖掘技术，明确商品的销售情况，以及客户的信用状况。

债权统计查询中将所有债权信息按照客户分类，使得用户可以一目了然的了解每个客户所属的债权信息。

3.3.3 财务催收管理模块

图3-9为财务催收管理模块图：

财务催收管理						
还款		催收记录查询	法务单据			
查询	还款	查询催收记录	查询	修改	删除	提交

图 3-9 财务催收管理模块

财务催收管理模块主要分为还款、催收记录查询、法务单据三个小模块。为用户提供债权管理的相应功能。

在还款模块中用户可以查看到所有尚有未还款项的债权条目，并可以手动添加还款记录。

催收记录查询模块中将所有已经进行录入过催收记录的整机及所有催收记录显示出来，方便用户查询。

法务单据模块中提供对法务单据的查询、修改、删除和提交功能。法务单据是对逾期情况严重的公司进行法务通知并采取法律途径解决债务问题的单据。设立此模块可以方便用户查询对客户发出的法务单据记录。

3.3.4 系统管理模块

图3-10为系统管理模块图：



图 3-10 系统管理模块

系统管理模块主要提供用户对登陆用户的管理。用户可以在此为除自己以外的人添加身份验证帐户以登陆本系统，并可以对所添加的用户进行信息的查询、修改和删除功能。

整个系统的大致模块如上所述，对于不同用户可能会出现不同的具体需求条目，需要对系统作出相应的功能修改或增减。

图3-11为系统的类图：

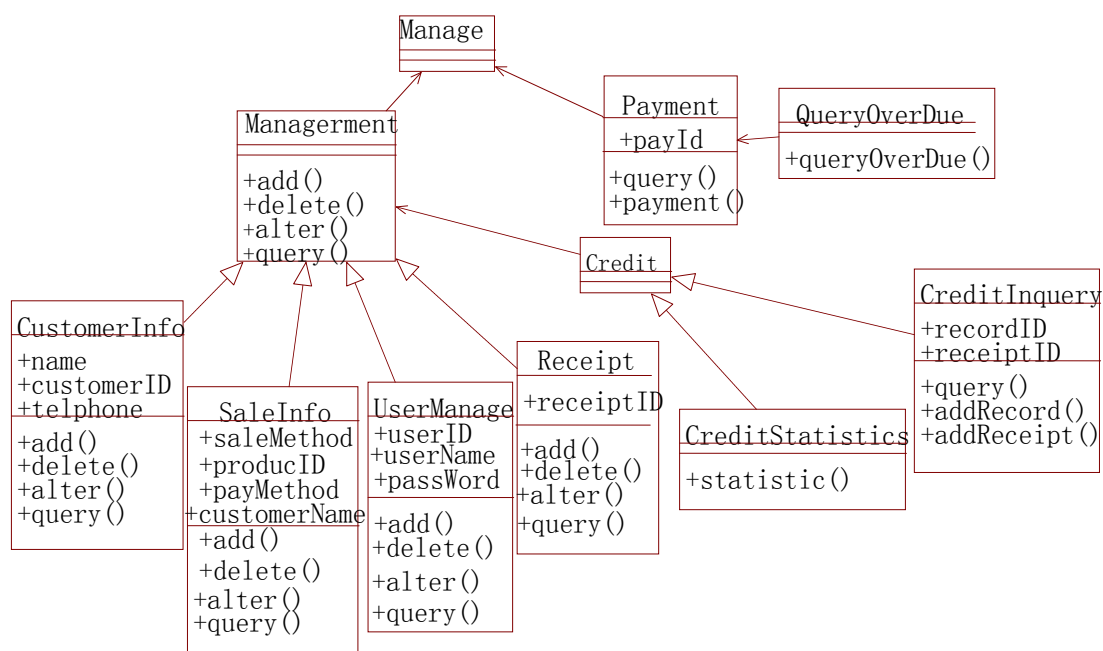


图 3-11 类图

采用starUML将类图生成的代码如下图3-12所示：

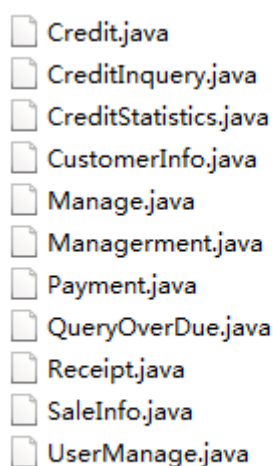


图 3-12 初始代码

3.4 数据库设计

对于数据的保存，都是使用数据库软件来进行，本系统选用的是Mysql数据库。数据库采用一定的数据模型，不仅仅描述了数据本身的特点，还描述了数据之间的联系，同时会使得所存储的数据冗余小，易修改，易扩充，与程序有较高的独立性。使用数据库还可以对数据进行统一的管理和控制，提供了数据的安全性、完整性以及并发控制。因此选择使用数据库软件而不是普通的文件进行数据的保存。

3.4.1 E-R 图

在数据模型构建中，用户可以管理交易的信息，并通过催收记录以及法务单据对客户的债权进行管理，用E-R图来建立数据库的概念模型，下面对部分实体内容进行简易介绍：

在图3-13给出了用户和整机销售以及客户之间的ER图，一个用户可以管理多个客户的信息，也可以管理多个整机的销售信息，但是一个客户和一个整机销售只能对应一个用户、客户和整机通过销售关系起来。

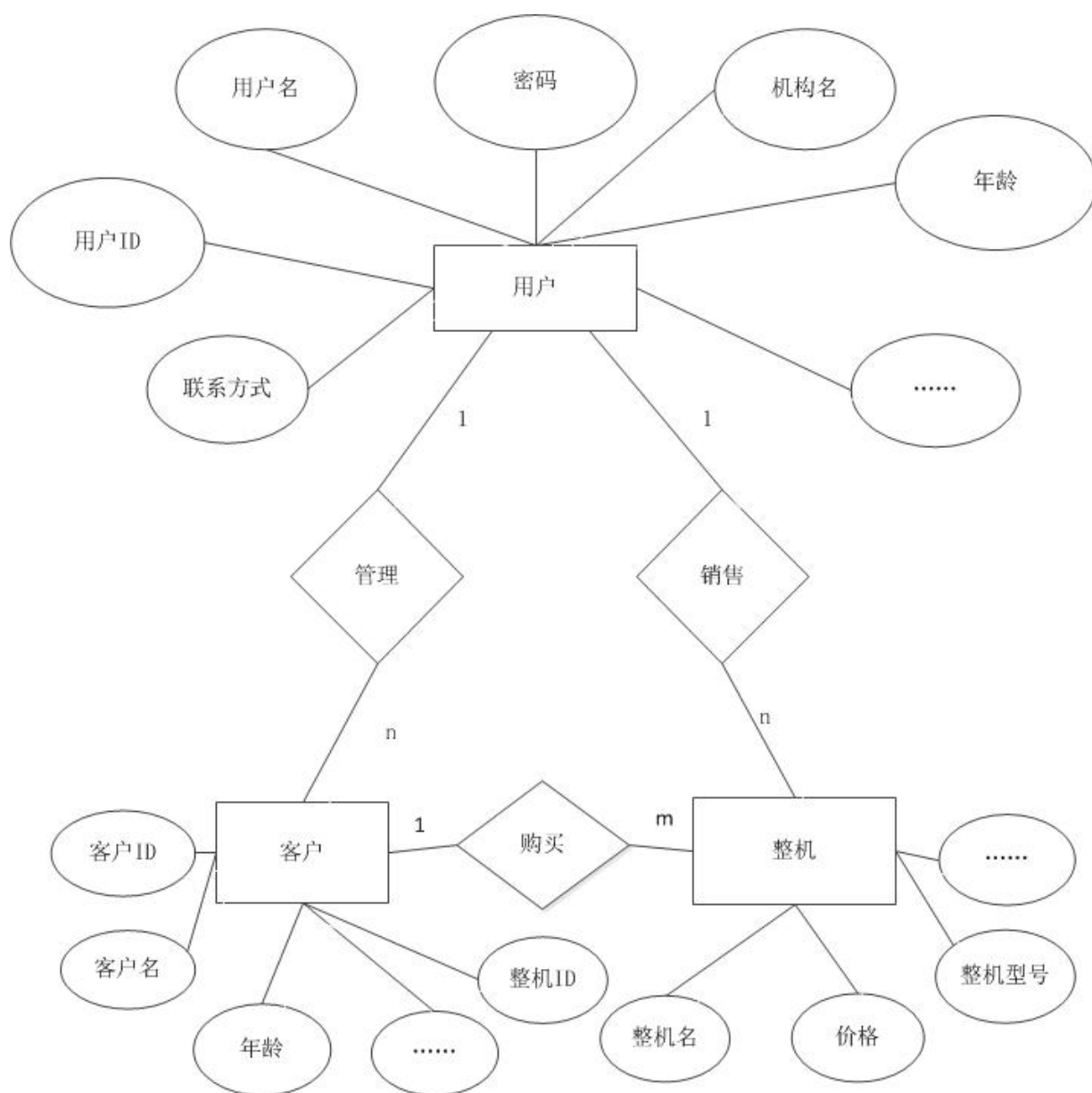
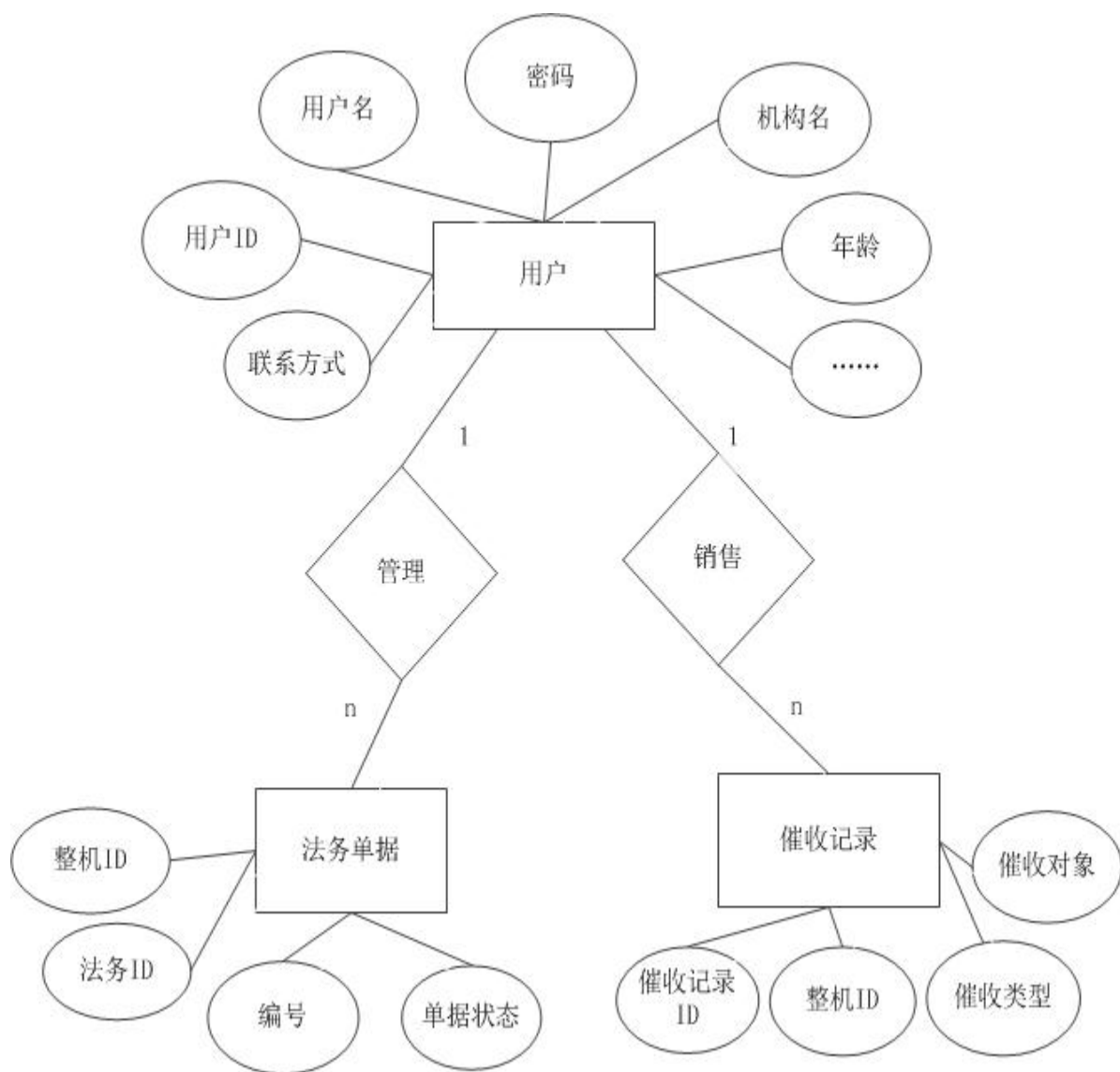


图 3-13 用户-客户-整机

图3-14和图3-15给出了用户对债权进行管理的信息：



3-14 用户债权管理 1

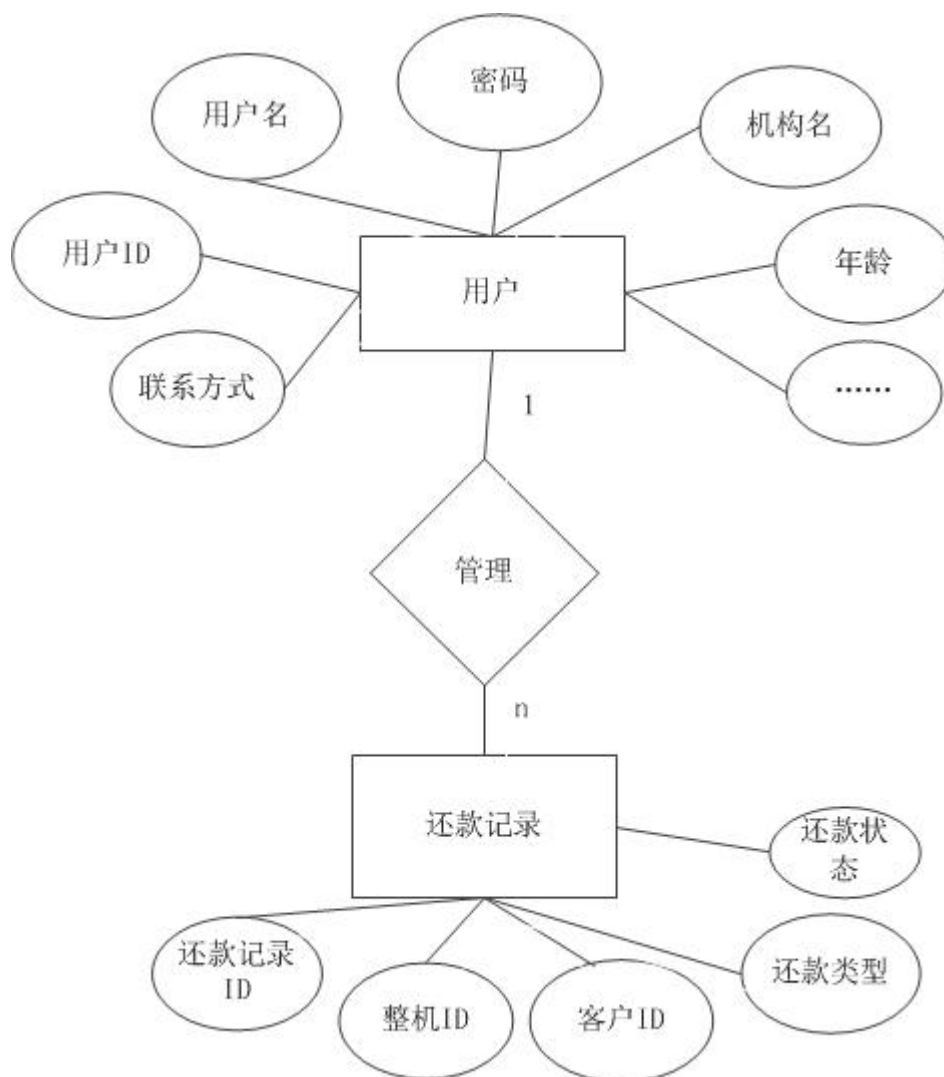


图 3-15 用户债权管理 2

在图3-14和图3-15的债权管理ER图中可以看出，一个用户可以管理多个法务单据，多个催收记录和多个还款记录，但是一个法务单据，一个催收记录和一个还款记录只能对应一个用户。

3.4.2 表设计

以下详细说明几个主要数据库表的设计：

表3-1 为系统用户信息表：

表 3-1 系统用户信息

字段名	注释	类型	长度	是否主键	是否不为空
User_id	用户ID	bigint	20	是	是

User_name	用户名字	varchar	128		
User_pwd	用户密码	Int	20		
User_sex	用户性别	Int	11		
User_type	用户类型	Int	11		
User_tel	用户电话	Varchar	128		
User_email	用户邮箱	Varchar	128		
User_addr	用户地址	varchar	128		

表 3-2 为员工信息表:

表 3-2 员工信息

字段名	注释	类型	长度	是否主键	是否不为空
Staff_id	员工ID	bigint	20	是	是
Staff_name	员工名字	varchar	128		
Staff_pwd	员工密码	Int	20		
Staff_sex	员工性别	Int	11		
Staff_type	员工类型	Int	11		
Staff_tel	员工电话	Varchar	128		
Staff_email	员工邮箱	Varchar	128		
Staff_addr	员工地址	varchar	128		
Staff_perf	员工业绩	Int	20		

表3-3为客户信息表:

表 3-3 客户信息

字段名	注释	类型	长度	是否主键	是否不为空
id	客户ID	bigint	20	是	是
Cust_name	客户名称	varchar	128		
Cust_type	客户类型	Int	11		
Cust_sex	客户性别	Int	11		
Per_id_card	证件号	varchar	64		
Per_id_card_pic	证件复印件	varchar	1024		
Identify_type	证件类型	Int	11		
Residence_addr	户口地址	varchar	128		
Residence_bl	户口簿	varchar	1024		

Photo	照片	Text			
Marriage_cert	结婚证	Varchar	1024		
Mate_id_card	配偶照片	Varchar	1024		
Mate_id_card_pic	配偶身份证复印件	Varchar	1024		
Mate_identify_type	配偶证件类型	Int	11		
Mate_residence_bl	配偶户口簿	Varchar	1024		
Mate_job	配偶单位职务	Varchar	1024		
Mate_photo	配偶照片	Varchar	1024		
Work_position	工作地点	Varchar	255		
Cust_post	邮编	Varchar	32		
Cust_email	EMAIL	Varchar	128		
Cust_tel	联系电话	Varchar	128		
Mate_tel	配偶电话	Varchar	128		
Wp_condition	单位职务证明	Varchar	1024		
Income_cert	收入证明	Varchar	1024		
Tax_reg_cert	纳税凭证	Varchar	1024		
Cust_dirt	土地使用权证	Varchar	1024		
Cust_house	房屋所有权证	Varchar	1024		
Cust_car	车辆	Varchar	1024		
Cust_bank	存款	Varchar	1024		
Cust_stock	股权证明	Varchar	1024		

在客户信息表中，将客户ID设为主键，并且不为空，通过客户信息的具体表项，掌握客户的基本信息，以对客户进行管理。

表3-4为整机销售信息表：

表 3-4 整机销售信息

字段名	注释	类型	长度	是否主键	是否不为空
Id	整机销售信息ID	Bigint	20	是	是
Cust_id	客户ID	Bigint	20		
Cm_num	整机编号	Varchar	64		
Cm_type	产品类型	Int	11		
Model	型号	Varchar	128		
Engine_num	发动机编号	Varchar	128		

Chs_num	车架编号	Varchar	128		
Warranty_num	保修卡编号	Varchar	128		
Sale_date	销售日期	Date			
Serv_center	指定特约服务中心	Varchar	255		
Sales_type	销售方式	Int	11		
Curr_price	现款价	Decimal	(10, 2)		
Intrest	利息总额	Decimal	(10, 2)		
First_pay	首付款	Decimal	(10, 2)		
Settl_price	合同价	Decimal	(10, 2)		
Maturity_len	信用销售期数	Int	11		
Pay_limit_type	还款周期类型	Int	11		
Pay_money_type	还款类型	Int	11		
Pay_later	首付垫付金额	Decimal	(10, 2)		
If_pay_later	是否首付款垫付	Smallint	6		
Pay_later_limit	首付款垫付期限	Int	11		
Pay_repay_info	是否已经产生债权数据	Int	11		
Pro_id	生产商ID	Bigint	20		
Agency_id	用户ID	Bigint	20		
Lease_id	融资租赁ID	Bigint	20		
Earnest_money	保证金	Decimal	(10, 2)		
Finance_fee	融资款手续费	Decimal	(10, 2)		
Premium	保险费	Decimal	(10, 2)		
Risk_control_id	风控描述ID	Bigint	20		
Status	整机销售状态	Int	11		
Cloud_id	组织机构ID	Bigint	20		

在整机销售信息表中，将整机销售信息ID设为主键，同时表4-2中还包含客户ID，用于记录整机对应的买家，以方便管理。

表3-5为还款记录表：

表 3-5 还款记录

字段名	注释	类型	长度	是否主键	是否不为空
id	还款记录ID	Bigint	20	是	是
Cust_id	客户ID	Bigint	20		
Cm_id	整机销售ID	Bigint	20		
Repay_type	还款类型	Int	11		
Repay_money	应还金额	Decimal	(10, 2)		
Repay_date	应还日期	Date			
Repay_amount	实际还款金额	Decimal	(10, 2)		

Real_rp_date	实际还款日期	Date			
Overdue_reason	逾期原因	Varchar	512		
Status	状态	Int	11		
Cloud_id	组织机构ID	Bigint	20		

表3-5为还款记录表，将还款记录ID设为主键，且不为空，同时包含客户ID和整机销售ID，将还款记录和客户信息以及相应的整机关联起来，不是独立的表。

表3-6为催收记录表：

表 3-6 催收记录

字段名	注释	类型	长度	是否主键	是否不为空
id	Id	Bigint	20	是	是
Cm_id	整机销售ID	Bigint	20		
Email_history_id	邮件发送历史ID	Bigint	20		
Msm_history_id	短信发送历史ID	Bigint	20		
Collection_time	催收时间	Datetime			
Collectioner	催收人	Bigint	20		
Collection_type	催收类型	Int	11		
Collection_content	催收内容	Varchar	512		

表3-6中，将ID作为主键，且不为空，关联整机销售ID，包含邮件短信发送历史，同时还包含催收时间、催收人、催收类型和催收内容，通过这些信息以及方式，达到催收的效果。

表3-7为法务单据表：

表 3-7 法务单据

字段名	注释	类型	长度	是否主键	是否不为空
Id	法务单据 ID	Bigint	20	是	是
Cm_id	整机销售信息 ID	Bigint	20		
Legal_num	法务编号	Varchar	128		
Legal_content	申请内容	Text			
Legal_create_ti	生成法务单据时间	Datetime			
Legal_duel_time	提交申请时间	Datetime			
Legal_status	单据状态	Int	11		

表3-7为法务单据表，在该表中，法务单据ID为主键，且不为空，关联整机销售信息ID，包含了法务编号、申请内容、单据状态等内容，以生成完整的法务

单据。

3.4.3 数据库连接

由于本系统是采用Struts、Spring和Hibernate三种著名的框架进行分层架构模式设计的，数据库连接操作方面不是采用JDBC技术，而是采用了基于OR-Mapping（对象关系映射）的Hibernate框架。其基本思想就是将数据库中的每张表映射成一个JavaBean对象，如图4-3所示。这样就避免了大量编写sql语句，然后通过JDBC操作数据库。我们只需通过操作实体Bean对象，然后里面Hibernate将该实体Bean对象提交，将Bean的修改更新到数据表中。系统是通过以下方式达到连接数据库：

首先先获得连接数据库的参数数据，如下：

```
jdbc.driver = com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost/crm?useUnicode=true&characterEncoding=utf-8
jdbc.username = management
jdbc.password = 123
```

然后通过配置文件利用spring内置的DBCP数据库连接池建立数据源对象：

```
< bean id= "dataSource" class="org.apache.commons.dbcp.BasicDataSource"
destroy-method = " close " >
    <property name="driverClassName" value="{jdbc.driver}" />
    <property name="url" value="{jdbc.url}" />
    <property name="username" value="{jdbc.username}" />
    <property name="password" value="{jdbc.password}" />
    <property name="initialSize" value="5" />
    <property name="maxActive" value="100" />
    <property name="maxIdle" value="30" />
    <property name="maxWait" value="1000" />
    <property name="poolPreparedStatements" value="false" />
    <property name="defaultAutoCommit" value="false" />
</bean>
```

最后再利用spring对hibernate进行配置，将数据表跟每个实体bean进行一一映射，并配置hibernate的sessionFactory所要用的数据源对象。

```
<bean id="sessionFactory"
```

```

class="org.springframework.orm.hibernate3.annotation.AnnotationSession
FactoryBean">
    <property name="dataSource" ref="dataSource" /><!--数据源对象-->
    <property name="namingStrategy">
        <bean class="org.hibernate.cfg.ImprovedNamingStrategy" />
    </property>
    <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">${hibernate.dialect}</prop>
            <prop key="hibernate.show_sql">${hibernate.show_sql}</prop>
            <prop key="hibernate.format_sql">${hibernate.format_sql}</prop>
        </props>
    </property>
    <property name="packagesToScan" >
        <list>
            <!--将数据表跟实体bean进行一一映射-->
            <value>org.jingf.invoicing.model</value>
            <value>org.jingf.crm.model</value>
        </list>
    </property>
</bean>

```

利用已经配置过的hibernate的sessionFactory对象操作实体bean操作数据库的数据。

3.5 基于数据挖掘的债权数据分析模型设计

3.5.1 数据挖掘相关技术

随着数据库技术的迅速发展，人们积累的数据越来越多，这些数据的背后隐藏着许多重要的信息，这时人们希望对其有更高层次的分析，在本系统中，由于交易的增多，导致债权信息的激增，在高效实现对数据的录入、查询、统计等功能的基础上，人们更加希望能发现这些数据中存在的关系和规则。

20 世纪 80 年代后期，出现了一门新的交叉性学科—数据挖掘，该学科融合了人工智能、数据库技术、模式识别、机器学习等众多领域的理论和技术，是人们

长期对数据库技术进行研究和开发的结果。数据挖掘作为一种技术，它的生命周期正处于沟坎阶段，需要人们花费一定的时间和精力去研究开发，使其逐步成熟，最终被广泛大众所接收^[17-18]。

数据挖掘作为数据库中知识发现的核心步骤，其中数据仓库是整个数据挖掘过程的基础，20 世纪 80 年代中期，数据仓库之父 W.H. Inmon 在《建立数据仓库》(building the data warehouse)^[19]一书中定义了数据仓库的概念，并在之后给出更为精确的定义：数据仓库是在企业管理和决策中面向主题的、集成的、时变的以及非易失的数据集合。

数据挖掘的应用于人类生产和生活的各个方面，其交叉产业标准过程(cross-industry standard process for data mining, CPISP-DM)是由 SPSS(Statistical Product and Service Solutions)、NCR(National Cash Register)和 Daimler Chrysler 三家公司于 1996 年制定的标准，它强调了数据挖掘的商业价值。CPISP-DM 参考模型包括：商业和数据理解、数据准备、建立模型、模型的评估和发布。

简而言之，数据挖掘就是采用一定的技术对数据进行“取其精华，去其糟粕”的过程。

数据挖掘按是否需要以时间为参数，根据其数据的历史和当前值去预测未来以及对当前数据存在的事实做规范性描述，可以分为以下五个模式：

1、分类模式(Classification):通过构造分类函数把具有某些特征的数据项映射到某个给定的类别上。

2、聚类模式(Clustering):将数据项分为多个类或簇，并按照“最小化类间的相似性，最大化类内的相似性”原则对数据项进行处理。

3、回归模式(Regression):与分类模式采用离散预测值不同，回归模式采用的是连续的预测值，将非线性的问题通过变量变换转换为线性问题，进行分析。

4、序列模式(Sequential):基于时间或其他序列经常发生的规律或者趋势，进行建模分析。

5、关联模式(Association):研究数据项之间的关联规则，同一事件中出现不同项之间的相关性。其中最著名的关联规则算法是 1994 年，由 Agrawal 等提出的 Apriori 算法，也是本文后续工作要用到的算法。

3.5.2 Apriori 算法

数据库中具体的一个物品定义为项目，并把一组项目的集合被称为项目集，对项目的操作被称为事务，例如一次购物。

支持度是指在数据库事务中存在某类项目集数占数据库中的比例。

置信度是指在项目集中存在某类项目集 A，其中有些同时还存在另一类项目集 B，存在某类项目集中还存在另一些项目集数（即存在 A 和 B）占存在某类项目集 A 的比例。例如在 {面包} 为 5，而 {面包，牛奶} 为 3，置信度=3/5，就是说买面包的人可能有 60% 的人会买牛奶。

频繁项目集：项目集的支持度超过定义的支持度阈值（最小支持度），其中频繁项目集的子集也一定是频繁项目集。

若事务 T，其中 $X \subseteq T$ ， $Y \subseteq T$ ， $X \cap Y = \phi$ ， $X \cup Y \subseteq T$ ，X、Y 也为项目集；则形如 $X \Rightarrow Y$ 的规则称关联规则；若关联规则的支持率和置信度分别大于或等于用户指定的最小支持率（minsupport）和最小置信度（minconfidence），则称关联规则为强关联规则；否则称关联规则为弱关联规则。

项目集中只有 k 个项目的称为 k 项目集，并用 C_k 表示，将数据库中所有超过最小支持度的 k 项目集的集合称为 K-频繁项目集，并 L_k 表示。

Agrawal 等人 1993 年在 “Mining Association rules between sets of items in large database” 提出关联规则方法，并于 1994 年在 “fast Algorithms for Mining Association rules in large database” 提出 Apriori 关联规则算法方法^[20-25]。

Apriori 算法的重要性质是频繁项集的非空子集都是频繁的项集，这一性质被广泛应用于寻找频繁项集中。

Apriori 算法的步骤：

第一步：简单统计所有含一个元素的项目集出现的频率，并找出不小于最小支持度的项目集，即 L_1 。

第二步：循环处理，在 L_1 中确定 C_2 ，并从中寻找所有含有两个元素的出现的频率不小于最小支持度的项目集，即 L_2 。

第 k 步：根据第 k-1 步生成的 (k-1) 维最大项目集产生 k 维候选项目集 C_k ，然后通过对数据库的搜索，找出频率不小于最小支持度的项目集，作为 L_k 。

3.5.3 债权数据分析模型设计方法

在对债权数据进行分析时，取其数据库中的部分客户和整机类型进行分析，为方便分析过程，A 代表挖掘机，B 代表压路机，C 代表装载机，D 代表推土机，E 代表起重机，定义最小支持度 minsupport_count=2，即 0.5，对数据库 D 中的数据利用 Apriori 算法进行如下处理：

表 3-8

TID	D中的项目
T1	A, B, C, E
T2	A, B, C
T3	C, D
T4	A, B, E

对表3-8进行处理：

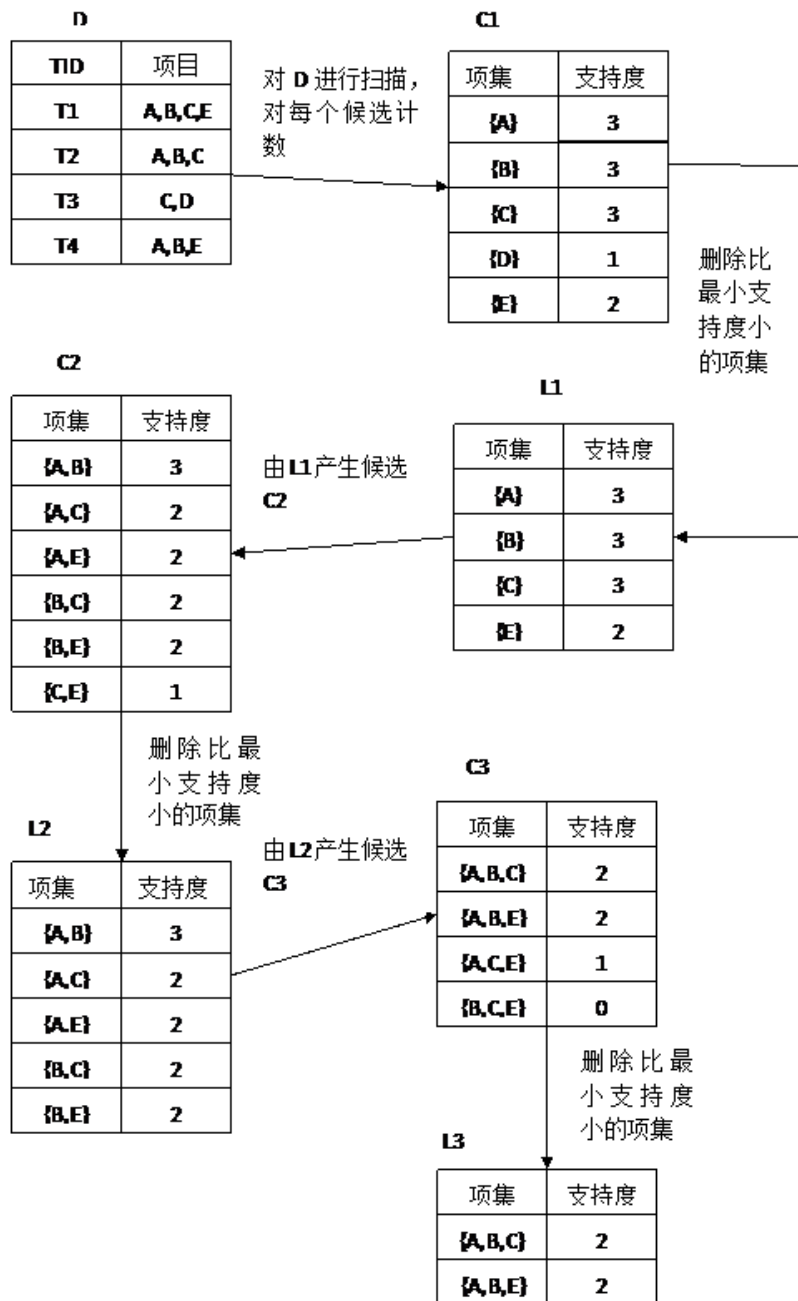


图 3-16 债权数据项集分析过程

由图3-16中可以得出L1, L2, L3对应的支持度如表3-9, 表3-10, 表3-11所示:

表 3-9 L1 支持度

1_项集 (L1)	支持度计数	支持度
{A}	3	0.75
{B}	3	0.75
{C}	3	0.75
{E}	2	0.5

表 3-10 L2 支持度

2_项集 (L2)	支持度计数	支持度
{A, B}	3	0.75
{A, C}	2	0.5
{A, E}	2	0.5
{B, C}	2	0.5
{B, E}	2	0.5

表 3-11 L3 支持度

3_项集 (L3)	支持度计数	支持度
{A, B, C}	2	0.5
{A, B, E}	2	0.5

该过程分析结果与编程结果一致, 编程结果如图3-17所示:

```
挖掘频繁关联规则: {[E]=[A], [B], [A, B]], [A]=[B]], [B]=[A]], [B, C]=[A]], [A, C]=[B]], [E, B]=[A]], [E, A]=[B]]}
候选频繁2-项集: {[E, C], [A, B], [B, C], [A, C], [E, B], [E, A]}
挖掘频繁2-项集: {[A, B]=0.75, [B, C]=0.5, [A, C]=0.5, [E, B]=0.5, [E, A]=0.5}
挖掘频繁1-项集: {[E]=0.5, [A]=0.75, [B]=0.75, [C]=0.75}
挖掘频繁3-项集: {[E, A, B]=0.5, [A, B, C]=0.5}
挖掘频繁项集: {1=[E], [A], [B], [C]], 2=[A, B], [B, C], [A, C], [E, B], [E, A]], 3=[E, A, B], [A, B, C]}
```

图 3-17 Apriori 编程结果

通过频繁集产生关联规则, 设用户的最小置信度 $minconf=0.7$, 则由公式一可以计数最小置信度:

$$\text{置信度}(A \Rightarrow B) = P(B|A) = \frac{\text{support_count}(AB)}{\text{support_count}(A)} \quad \text{公式一}$$

经过计算，我们可以得出如表 3-12 所示的结果：

表 3-12 由频繁集得到的关联规则

关联规则	支持度	置信度
$\{A\} \Rightarrow \{C, B\}$	2	$2/3$
$\{A\} \Rightarrow \{B, E\}$	2	$2/3$
$\{B\} \Rightarrow \{A, C\}$	2	$2/3$
$\{B\} \Rightarrow \{A, E\}$	2	$2/3$
$\{C\} \Rightarrow \{A, B\}$	2	$2/3$
$\{E\} \Rightarrow \{A, B\}$	2	$2/2$
$\{A, B\} \Rightarrow \{C\}$	2	$2/3$
$\{A, B\} \Rightarrow \{E\}$	2	$2/3$
$\{A, C\} \Rightarrow \{B\}$	2	$2/2$
$\{A, E\} \Rightarrow \{B\}$	2	$2/2$
$\{B, C\} \Rightarrow \{A\}$	2	$2/2$
$\{B, E\} \Rightarrow \{A\}$	2	$2/2$

由表3-12可以得出，购买E（起重机）的一定会购买A（挖掘机）和B（压路机），而且A（挖掘机）和B（压路机）一定会同时购买。

3.6 本章小结

本章是本文的理论组成部分，是商务债权系统开发的理论依据。本章介绍了商务债权管理系统开发需要研究和应用相关技术，使用 Hibernate 框架进行数据库层的封装，java 语言实现开发，采用 Spring 框架进行开发，数据库使用 MySQL，并采用了 B/S 结构设计，接着介绍了商务债权系统的相关功能设计和数据库设计，并提出了基于数据挖掘的债权数据分析模型^[25-30]。

第四章 系统实现

4.1 界面设计

4.1.1 系统登录界面设计

图4-1为系统的登陆界面：



图 4-1 登录界面

在系统部署成功后，进入如图5-1所示的登入界面。登录界面主要用于判别使用者的身份是否有在系统中录入，如果数据库中存在该用户的用户名与密码，则可以登录成功并使用系统，若不存在，则无法使用系统。

相应的代码如下：

```
@Override
public void actionPerformed(ActionEvent e) {
    if(userName.equals(text1.getText())&&passwrod.equals(text2.getText()))
    {
        JOptionPane.showMessageDialog(null, "登陆成功误", "提示",
        JOptionPane.INFORMATION_MESSAGE);
    }
    else {
```



```

JOptionPane.showMessageDialog(null,          "错误",          "提示",
JOptionPane.ERROR_MESSAGE);
        text1.setText("");
        text2.setText("");
    }
}
});
//以上用于判断用户是否正确
public String image() {
    BufferedImage img = new BufferedImage ( 58 , 20 , BufferedImage.
    TYPE_INT_RGB ); //得到该图片的绘图对象
    Graphics g = img.getGraphics(); Random r = new Random(); Color c = new
    Color(200, 150, 255);g.setColor(c);// 填充整个图片的颜色
    g.fillRect(0, 0, 58, 20);// 向图片中输出数字和字母
    StringBuffer sb = new StringBuffer(); char[] ch =
    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789".toChara
    rArray();
    int index, len = ch.length;
    for (int i = 0; i < 4; i ++) {
        index = r.nextInt(len);
        g.setColor(new Color(r.nextInt(88), r.nextInt(188), r.nextInt(255)));
        // 输出的字体和大小
        g.setFont(new Font("Arial", Font.BOLD | Font.ITALIC, 12));
        //写什么数字, 在图片的什么位置画
        g.drawString("" + ch[index], (i * 15) + 3, 15);
        sb.append(ch[index]);
    } session.put("piccode", sb.toString());
    ImageIO.write(img, "JPG", response.getOutputStream());}
//以上为验证码图片的关键代码

```

4.1.2 系统主界面设计

图4-2为系统主界面:

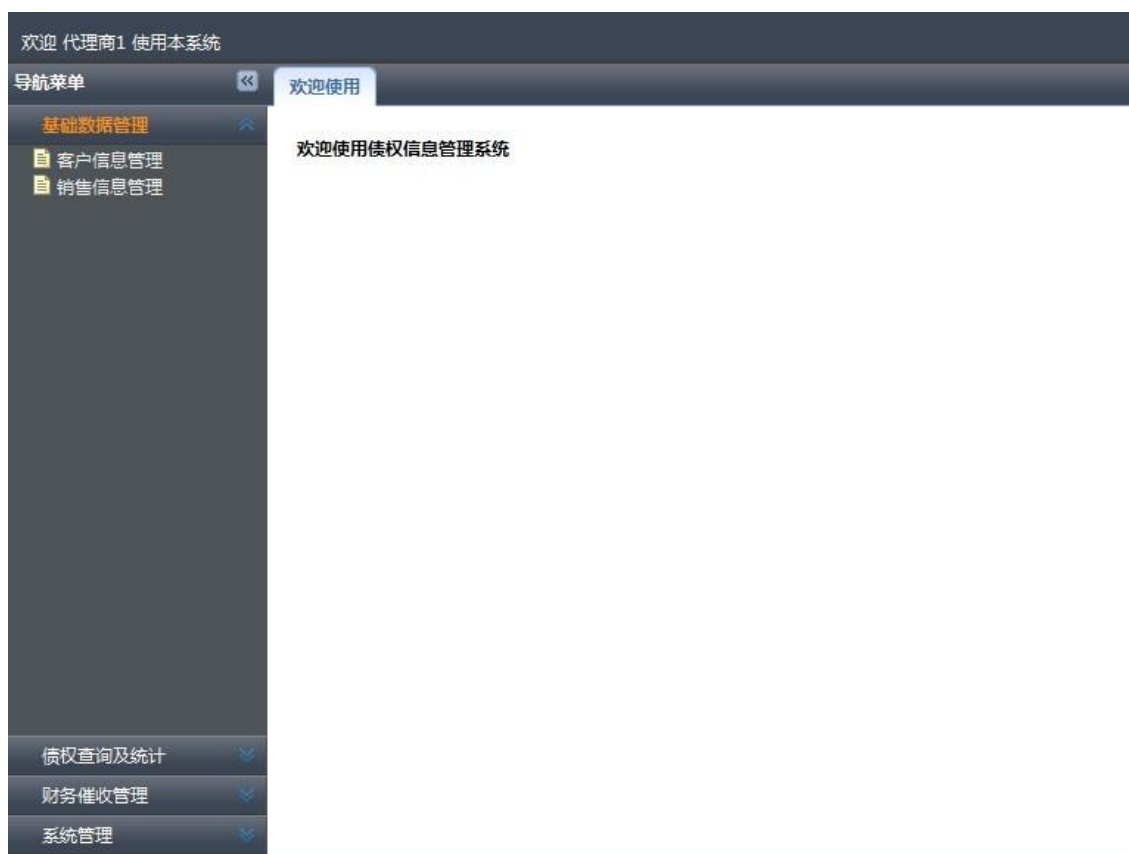


图 4-2 系统主界面

在图 4-1 的身份验证成功后，用户用户可以登录到系统，并跳转到如图 4-2 的主界面页面。主界面主要分为两个部分，左侧为功能列表，包含基础数据管理、债权查询及统计、财务催收管理和系统管理等四个模块，并这设计下拉选项用于展示功能并指向跳转，右侧用于展示所请求页面，在右侧打开的窗口中使用系统。

4.2 基础数据管理模块

4.2.1 客户信息管理

以用户用户身份登录成功进入系统之后选择客户信息管理模块，即会显示如图 4-3 所示的客户信息页面。



图 4-3 客户信息管理界面

在图4-3中可以看到客户列表以及每个客户的部分信息，同时用户用户可以通过新增、修改、删除和查询按钮对客户进行增删改查等操作。例如选中客户1，并点击详细信息按钮，可以查看所选客户的详细信息，如图4-4所示。



图 4-4 客户详细信息查看界面

而点击新增按钮，则会出现如图4-5的界面，用户可以在此界面完成添加新客户动作。

客户信息

基础信息

客户类型：

个人

* 客户名称：

性别：

男

女

* 个人证件类型：

身份证

* 个人证件号码：

配偶电话：

户口地址：

邮编：

Email：

配偶证件类型：

身份证

配偶证件号：

配偶单位职务：

* 移动电话：

身份证明

* 身份证复印件：

上传

0

* 照片：

上传

0

配偶身份证复印件：

上传

0

配偶照片：

上传

0

* 户口簿复印件：

上传

0

* 结婚证复印件：

上传

0

配偶户口簿复印件：

上传

0

资产证明

单位职务证明：

上传

0

收入证明：

上传

0

纳税凭证：

上传

0

土地使用证：

上传

0

图 4-5 客户信息管理界面

4.2.2 销售信息管理

在左侧的功能菜单中，点选销售信息管理后会进入到销售信息管理模块，在此可以对整机销售信息进行管理：包括新增销售信息、查看详细信息、修改和删除销售信息等。

图4-6为销售信息管理界面：

销售信息管理 ×

销售信息列表

新增

生成债权数据

详细信息

修改

删除

查询

	整机编号	产品类型	客户名称	销售方式	债权数据生成状态
1	1001	挖掘机	客户1	分期付款	已生成
2	1002	挖掘机	客户1	分期付款	已生成
3	1003	压路机	客户2	分期付款	已生成
4	1004	装载机	客户2	分期付款	已生成
5	1005	挖掘机	客户2	分期付款	已生成
6	1006	挖掘机	客户2	分期付款	已生成
7	123	挖掘机	客户1	分期付款	已生成

10

第 1 页 共 1 页

当前显示 1 - 7 条记录 共 7 条记录

图 4-6 销售信息管理界面

在新增销售信息时根据所选销售方式可以生成债权数据，如若没有在添加的时候生成债权数据，可以在列表中选取并生成债权数据。

54

4.3 债权查询及统计模块

4.3.1 债权明细查询

图4-7为债权明细查询界面：

债权详细查询 >

债权详细列表

查询

新增催收记录

新增法务单据

查看整机信息

查看客户信息

查看还款信息

序号	整机编号	客户名称	整机类型	销售方式	应还金额	已付金额	已还金额	未还金额	销售期数	已还期数	逾期期数	逾期金额	催收次数
1	1002	客户1	挖掘机	分期付款	¥1,100.00	¥900.00	¥900.00	¥200.00	2	0	2	¥200.00	0
2	1003	客户2	压路机	分期付款	¥1,000.00	¥800.00	¥0.00	¥1,000.00	4	0	2	¥300.00	0
3	1001	客户1	挖掘机	分期付款	¥1,100.00	¥100.00	¥0.00	¥1,100.00	10	0	2	¥220.00	4
4	1004	客户2	装载机	分期付款	¥1,200.00	¥800.00	¥0.00	¥1,200.00	4	0	2	¥300.00	0
5	123	客户1	挖掘机	分期付款	¥92,000.00	¥10,000.00	¥0.00	¥92,000.00	3	0	0	¥0.00	0
6	1005	客户2	挖掘机	分期付款	¥400.00	¥800.00	¥0.00	¥400.00	4	0	0	¥0.00	0
7	1006	客户2	挖掘机	分期付款	¥1,200.00	¥800.00	¥0.00	¥1,200.00	4	0	0	¥0.00	0

10 < < 第 1 页 共 1 页 > > >

当前显示 1 - 7 条记录 共 7 条记录

图 4-7 债权明细查询界面

在主界面的左侧菜单栏中，点选债权明细查询及统计模块下的债权明细查询，会显示如图4-7所示的界面，用户可以在此界面查询债权明细记录。债权信息有还款期限，在还款期限内未还上的定义为逾期债权，逾期债权可以进行催收。债权详细列表提供了录入催收记录的功能，同时逾期过于严重可发法务单据进行交涉并通过法律途径解决债权问题，此模块同时实现了新增法务单据的功能。

4.3.2 债权统计查询

图4-8为债权统计查询界面：



序号	客户名称	应还金额	逾期期数	已还金额	未还金额	逾期金额
1	客户1	¥94,200.00	4	¥900.00	¥93,300.00	¥420.00
2	客户2	¥3,800.00	4	¥0.00	¥3,800.00	¥600.00

图 4-8 债权统计查询界面

在主界面的左侧菜单栏中，点选债权明细查询及统计模块下的债权统计查询，如图4-8所示的债权统计查询界面，改界面主要提供给系统使用者清晰且详细

的债权统计数据。

债权统计查询页面主要为用户将同一个客户下的所有整机的债权信息统计起来并集中显示在一起，让用户更为方便查看债权统计数据，并便于统计客户的还款情况，将客户进行分类。

如图4-9为债权统计查询的详细界面：

债权统计查询												
债权信息列表												
查询 查看债权信息												
	客户名称	应还金额	逾期期数	已还金额	未还金额	逾期金额						
1	客户1	¥94,200.00	4	¥900.00	¥93,300.00	¥420.00						
	整机编号	整机类型	销售方式	应还金额	首付金额	已还金额	未还金额	期数	已逾期数	逾期期数	逾期金额	催收次数
1	1002	挖掘机	分期付款	¥1,100.00	¥900.00	¥900.00	¥200.00	2	0	2	¥200.00	0
2	1001	挖掘机	分期付款	¥1,100.00	¥100.00	¥0.00	¥1,100.00	10	0	2	¥220.00	4
3	123	挖掘机	分期付款	¥92,000.00	¥10,000.00	¥0.00	¥92,000.00	3	0	0	¥0.00	0
5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2												

4.4.2 催收记录查询

图4-11为催收记录查询界面：



图 4-11 催收记录查询界面

点击主界面左侧财务催收管理模块下的催收记录查询，用户可以进入如图 4-11 所示的催收记录查询界面，在界面中可以看到催收类型、催收人、催收对象以及时间和内容等详细内容。

4.4.3 法务单据

图4-12为法务单据界面：



图 4-12 法务单据界面

点击主界面左侧财务催收管理模块下的催收记录查询，用户可以进入如图4-12 的法务单据界面，在此可以对法务单据进行查看、修改、删除和提交操作。

4.5 系统管理

图4-13为系统管理界面：

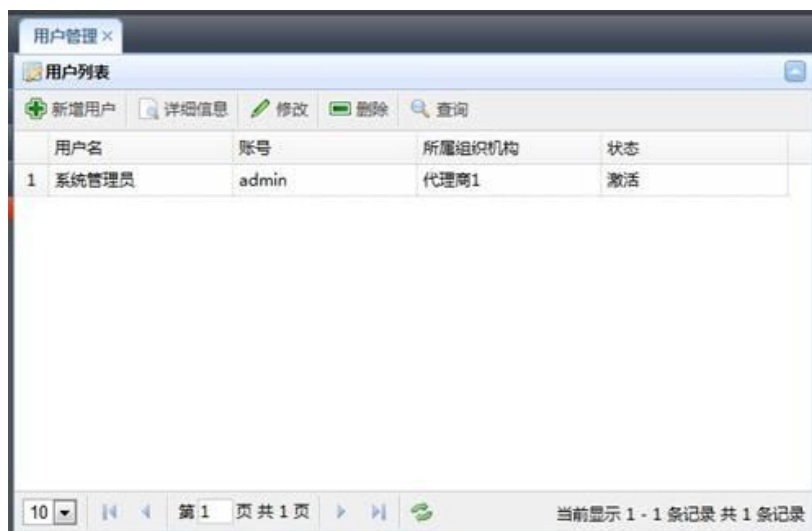


图 4-13 系统管理界面

点击主界面左侧菜单栏中的系统管理，用户可以进入如图4-13所示的系统管理界面，在该界面对用户进行的创建、修改、删除、查询等操作。

4.6 本章小结

本章详细介绍了商务债权管理系统的界面，系统的使用步骤。

第五章 系统测试

5.1 系统测试的步骤

系统测试是整个系统开发和管理的过程中必不可少的步骤，可以渗透到整个系统的开发过程中^[34-40]。在系统开发期间，主要采用白盒测试的方法，对代码进行调试，检查代码的正确性，并加入负载压力测试，度量系统的可扩展性和性能；在系统的所有的代码编写完成之后，主要采用的是黑盒测试方法，通过编写测试用例进行测试。设计测试用例，需要对系统的需求十分了解，是由系统测试进程中测试前提条件、测试步骤、测试结果等信息编辑组成的文档。通过使用系统测试用例对商务债权管理系统的所有功能进行测试，能最大限度地测出系统存在的bug，以便使整个系统更好、更稳定。活动步骤如图 5-1：

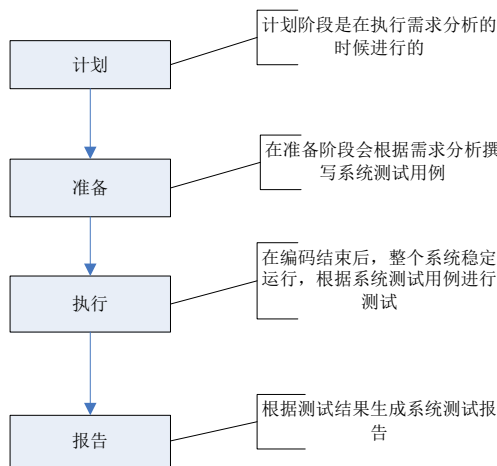


图 5-1 系统测试的活动步骤

通过完整的活动步骤完成，系统测试会形成一个测试报告文档，报告中包含系统存在缺陷、未实现的功能和新出现的问题等。开发人员会依据该报告进行新一轮的开发计划，经过反复多次的系统测试才能使系统趋于稳定。

5.2 系统测试的方法

本系统测试采用的白盒测试方法主要是在程序编码阶段，通过设计test用例，测试代码的正确性和无误性。

本系统测试采用的黑盒测试方法，是在系统功能已经知道的情况下，通过对每个功能的测试，查看功能是否正常。在测试时，不用考虑程序的内部逻辑和内

部流程，只需要把程序当做一个看不到内部的黑盒子，在程序的接口进行测试，只关注功能是否符合需求规格说明书，程序能否在外部信息完整的情况下恰当地获取输入的数据而生成正确的输出。

5.3 测试用例

测试用例主要包括测试用例 ID、测试用例名称、测试等级、测试用时、测试过程组成，其中测试过程还包括测试项目、测试步骤、预期结果和实测结果等内容。表 5-1 是一个测试用例表

表 5-1 测试用例表

测试用例 ID	ZQGL-功能测试-01		
测试用例名称	新增功能测试		
测试等级	1	测试用时	30m
测试过程			
测试项目	测试步骤	预期结果	实测结果
用户新增功能测试	1、登陆系统	可以在债权管理系统中添加新的用户	可以在债权管理系统中添加新的用户，并提示“用户添加成功”
	2、在主界面的左侧点击“系统管理”		
	3、在右侧界面的左上角，点击“新增”，添加新的用户		
客户新增功能测试	1、登陆系统	可以为债权管理系统添加新的客户	可以载债权管理系统总添加新的客户，并提示“客户添加成功”
	2、在主界面左侧点击“基础数据管理”		
	3、在基础数据管理的下拉选项中，选中“客户信息管理”		
	4、在右侧界面的左上角，点击“新增”，添加客户		
货物新增功能测试	1、登陆系统	可以为债权管理系统添加新的整机信息	如果是新的整机类型，可以直接添加新的整机信息，并提示“整机添加成功”；如果不是新的整机类型，添加之后，提示“整机更显成功”
	2、在主界面左侧点击“基础数据管理”		
	3、在基础数据管理的下拉选项中，选中“销售信息管理”		
	4、在右侧界面，选中“整机信息”		

	5、在新的界面中，点击“新增”，添加整机		
销售信息新增功能测试	1、登陆系统	可以为债权管理系统添加新的销售信息	可以添加新的销售信息，并提示“新增销售信息成功”
	2、在主界面左侧点击“基础数据管理”		
	3、在基础数据管理的下拉选项中，选中“销售信息管理”		
	4、在右侧界面，选中“销售信息”		
	5、在新的界面中，点击“新增”，添加销售信息		
催收记录新增功能测试	1、登陆系统	可以为债权管理系统添加新的催收记录	可以添加新的催收记录，并提示“新增催收记录成功”
	2、在主界面左侧点击“财务催收管理”		
	3、在右侧界面中，点击“新增”，添加催收记录		

5.3 本章小结

本章主要描述了系统测试流程和要点，包括系统测试的步骤，测试使用方法和详细阐述了系统测试用例的具体内容。在系统开发过程中，系统测试起着承上启下的作用，它能够保证系统功能的全面性、可靠性和系统的性能指标。

第六章 总结与展望

随着商品经济的发展，办公自动化成为必然趋势，商品交易的快速化和跨区域化，使传统的商品经济无法满足人们的需求，方便、快捷成为新的市场发展趋势，正是由于这一主流趋势，结合实际整机销售的债权信息，结合当前已有的电子商务系统和相关的电子管理信息系统，在已有的商务债权信息的大背景下，提出了开发商务债权系统的可行性和实用性，设计详细的需求分析，结合当前主流的电子商务开发语言 JAVA 进行设计研究，商务债权管理系统的开发具有一定的现实意义，本文从先进性、稳定性、实用性和便捷性等多个方面展开研究开发工作，详细阐述了商务债权开发的背景意义，解释了债权的概念和方式，并在此基础上提出了系统设计的相关需求。

本课题对当前市场已有的债权系统进行研究与分析，提出基于 Spring 框架的商务债权管理系统，在 B/S 开发模式下，开发拟采用的设计语言为 java，数据库服务器为 MySQL Server5.5，辅助的开发工具有：Myeclipse。这样的设计既能满足实际的功能需求，又能保证信息资源共享的高效性与安全性。

该系统主要注重数据的管理，将整机销售、代理商和客户的实际情况结合，对系统的基础数据管理模块、债权查询及统计模块、财务催收管理模块和系统管理模块进行了详细的设计和分析，并采用经典的软件测试方法——黑盒测试，设计了详尽的测试用例，发现系统软件均可在 Windows98 , Windows ME, Windows2000 以及 Windows xp 下安装使用，其中还贷和信息查询端只需浏览器即可，证实了系统的稳定性和可操作性。

本文所研究并开发的商务债权管理系统已经依照需求完成了，但是由于各种条件的限制，还有很多问题需要做进一步的研发，虽然本系统已经投入实际应用，但是由于客户需求的不断更新，商务债权管理系统还需要根据实际情况不断完善系统功能。由于目前主要针对的是 PC 操作，以目前各种移动设备 APP 快速发展的形势，开发 APP 成为必然趋势。应加强对系统使用角色的分级，进而加强用户权限管理。目前此系统是一个独立的系统，可以和其他系统结合使用，比如客户关系管理系统等。

致谢

本文的编写得到许多人的帮助，在此我向他们致以诚挚的谢意。

首先感谢汤羽、陈剑雄老师。在毕业设计作品的完成过程中，他们给予了技术上的指导和精神上的鼓励，在我论文的选题、立题、等过程中提供许多宝贵意见，无论在课程的认识过程、课题的研究，特别是汤羽老师给了我热忱的关怀和耐心的指导。

其次感谢部门同事，他们为我提供了宝贵的建议以及经验，指导我如何分析问题，在实践方面给予了我很多帮助和提供了很多宝贵的意见，让我少走了很多弯路，从中学到了很多东西，顺利完成实验。

再者，我要感谢我的父母，是他们在疲惫时给予了我巨大的关心和鼓励，让我更加的有动力去完成这次设计，今后我也将在你们的鼓励 and 希望下努力工作。

最后，谨向百忙之中审阅本文的老师们表示衷心的感谢！

参考文献

- [1] [日]椿寿夫.《担保物权法》.法律文化社,1991年版,第219页
- [2] 王海鹏.债权回收管理系统的设计与实现[D].山东:山东大学,2011,20-60
- [3] 戴发文.我国商业银行不良金融债权管理研究[D].江西:江西财经大学,20042-10
- [4] 壮琼芳.论企业集团的债权管理[J].时代经贸(学术版),2008,6(20):1-10
- [5] 韩麟.论企业的债权管理[J].湖南商学院学报,2009,16(4):3-8
- [6] 段琳琳.基于敏捷软件过程的债权管理系统开发[J].微计算机应用,2011,32(1):1-6
- [7] 郑人杰、马素霞译.《软件工程》[M].机械工业出版社,2011年,10-100
- [8] 刘宁,陆荣国,缪万胜. MVC体系架构从模式到框架的持续抽象进化阴[J]. 计算机工程, 2008: 107-110.
- [9] 王映辉,王英杰,王彦君. 基于MVC的软件界面体系结构研究与实现[J]. 计算机应用研究, 2004: 188-190.
- [10] 孙卫琴. 精通Struts: 基于MVC的JavaWeb设计与开发[M]. 北京: 电子工业出版社, 2004: 43-44.
- [11] 黄华. 框架技术在Web系统开发中的应用阴[J]. 微机发展, 2005: 77-79.
- [12] 张娅,杜友福,张南平. 用Struts建立MVC模式的Web应用阴[J]. 微机发展, 2003: 105-107.
- [13] 梁立新.《基于Struts-Spring-Hibernate的Java应用开发》(第二版)[M].电子工业出版社,2008年,150-200
- [14] 计文柯.《Spring技术内幕》[M].机械工业出版社,2010年,5-30
- [15] 崔洋、贺亚茹.《MySQL数据库应用从入门到精通》[M].中国铁道出版社,2012年,5-5
- [16] 孙莉,李树刚,陶莹,陈华斌.基于B/S模式的质量信息管理系统设计与实现[J].上海交通大学学报.2010(44):175-178.
- [17] Agrawal.R.Datamining:Crossingthe.chasm[R/OL].<http://www.almaden.ibm.com/cs/people/ragrawal/papers/kdd99-chasm.ppt>,2002-11-20
- [18] Han J W,Micheline K. 数据挖掘概念与技术[M].范明, 孟晓峰译.北京: 机械工业出版社, 2001.
- [19] Inmon W H.Buidling the data warehouse[M].3rd ed.New York:John Wiley & Sons Inc,2002.31-145.
- [20] Codd.E.F,Codd.S.B,Salley.C.T.Beyond,decision,support[J].Computer,World,1993,27(30):87-89.
- [21] Shearer C.The CRISP-DM model:The new blueprint for data mining[J].Journal of Data

- Warehousing,2000,5(4):13-22.
- [22] 陆建江, 张文献.关联规则挖掘的基本算法[J].计算机工程,2004(15): 34-35
- [23] 荣冈, 刘进锋, 顾海杰. 数据库中动态关联规则的挖掘[J].控制理论与应用,2007(1):127-131
- [24] 武建华, 宋擒豹, 沈均毅, 谢建文. 基于关联规则的特征选择算法[J].模式识别与人工智能,2009(2): 256-262
- [25] 何军, 刘红岩, 杜小勇. 挖掘多关系关联规则[J].软件学报,2007(11):2752-2765
- [26] Ben Alex. Acegi Security Reference Documentationl. 0. 3[EB / PDF]. 2006
- [27] Sun Microsystem. Java Platform, Enterprise Edition(J2SE)Specification, versionl. 4 final Release[EB / PDF]: Sun Microsystem, 2003
- [28] Derek C. Ashmore. The J2EE Architect'S Handbook: How to be a Successful Technical Arechitect for J2EE Application[M]. DVT Press, 2004
- [29] Craig Walls, Ryan Breichenbach. Spring inAction[M]. 人民邮电出版社. 2006. 4
- [30] Paul Hammant,Aslak Hellsoy . Constructor Injeetion[EB / OL] . [http : / / www. picecontainer. org / Constructor+Injcction](http://www.picecontainer.org/Constructor+Injcction). 2004
- [31] 黎小红. 基于 Spring 框架应用的权限控制系统的研究和实现[J]. 计算机与信息技术. 2006
- [32] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman: Role-Based Access Control Models. IEEE Computer 29(2): 38-47(1996)
- [33] Longhua Zhang, Gail-Joon Ahn and Bei-Tseng Chu. A Role-Based Framework for Role-Based Delegation . In Proceedings of ACM Symposium on Access Control Models and Technologies(SACMAT), Chantilly, VA, May 34, 200 1.
- [34] 董晓霞.相邻因素组合测试用例集的最优生成方法[J].计算机学报,2007(02)
- [35] 杨玲萍;韩阳.基于功能点分析测试设计充分性模糊评判建模[J].计算机工程与应用, 2007(03)
- [36] 张永强;陈永革;姚立新.军用软件的测试与实践方法[J].火力与指挥控制,2006(09)
- [37] 朱海燕.关于两两测试的研究[J].计算机工程与设计,2006(15)
- [38] 董晓霞.软件测试工程化的研究和实践[J].计算机工程与设计,2006(11)
- [39] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman: Role-Based Access Control Models. IEEE Computer 29(2): 38-47(1 996)
- [40] Longhua Zhang, Gail-Joon Ahn and Bei-Tseng Chu. A Role-Based Framework for Role-Based Delegation. In Proceedings of ACM Symposium on Access Control Models and Technologies(SACMAT), Chantilly, VA, May 34, 200 1.
- [41] E. and R. Sandhu. Framework for role-based delegation models. In 16 Annual Computer Security Applications Confercnce,New Orleans, Louisiana, December 2000.

- [42] Ezedin Barka and Ravi Sandhu. A Role-based Delegation Model and Some Extensions. Proceedings of 16th Annual Computer Security Application Conference, Sheraton New Orleans, Dec. 11-15, 2000.
- [43] Zhao Qingsong, Sun Yufang etc. , Configuring Role-based Access Control Model to Enforced Chinese Wall Security Policy, Proceedings of the Sixth International Conference for Young Computer Scientists, Volume I, Hangzhou, China, International Academic Publishers, World Publishing Corporation, Beijing, China, Oct 2001, pp206-209.