

企业合同管理信息系统设计与实现

摘 要

随着科技发展和社会进步，特别是计算机的一个广泛的普及，计算机应用领域逐渐从大规模科学计算转向大规模海量数据处理以及 workflow 事务管理，这导致在桌面计算机的核心，数据库管理体系为环境管理的大型交易处理和 workflow 管理和其他应用程序的信息系统的开发，尤其是在与各项目管理中的应用时，日益受到重视。

中国的信息产业近年来发展迅速，手工处理数据的管理方法是不合时宜的，使用 IT 技术来提高服务质量和水平是当务之急。

该系统是专为企业的进行合同管理开发的。随着经济的发展，企业正朝着规模化，规模化发展，并且中小型企业，产品的多样化，这也使得签署合同在这种情况下大幅增加，依靠手工处理这些合同不仅显得力不从心，而且很容易出错。

具体任务是设计一个企业合同数据管理系统，以电脑代替人来执行一系列新的合同操作，如增加，删除旧的合同，修改，查询操作。设计的指导思想是为用户着想，典雅的界面，尽可能操作简单，作为一个实际运行系统具有良好的容错性，误用时，会向用户发出警告并且及时更正。计算机的资源是有限的，所以对功能强大的 JSP 的充分利用，并且使用 struts 以及 hibernate 架构，使得对系统资源的使用尽可能多的减少，系统使 SQLServer2005 进行数据存储。

关键词：合同管理 jsp struts hibernate SQLServer2005

Design and Implementation of Enterprise Contract Management Information System

Abstract

With technological development and social progress, in particular, a widely popular computer, computer applications gradual shift from large scale scientific computing mass data processing and workflow transaction management, which led to the core of the desktop computer, the database management system for the environmental management of large transaction processing and workflow management and other applications of information systems development, especially with the project management application, and more attention.

China's information industry has developed rapidly in recent years, manual processing of data management is outdated, the use of IT technology to improve service quality Volume and level of management is a priority.

The system is designed for industrial manufacturing enterprises for the development of contract management. With economic development, companies are moving in size, scale development and small and medium enterprises, product diversification, which makes the contract signed by a significant increase in this case, relying on hand not only appeared to be inadequate to deal with these contracts, but prone to error.

Specific task is to design an industrial contract data management systems to computers instead of people to perform a series of new contract operations such as add, delete the old contract, modify, query operation. Design guidelines for the sake of users, elegant interface, as simple, as a practical operating system has a good fault tolerance, misuse, it will warn the user and promptly corrected. The computer's resources are limited, so powerful, full use of JSP, and

use struts and hibernate framework, making use of system resources as much as possible to reduce the system to SQLServer2005 for data storage.

Keywords: contract; management; jsp; struts; hibernate; SQLServer2005

目 录

1 引言	1
1.1 企业合同管理系统论述	1
1.2 开发环境及运行环境	1
1.2.1 开发环境	1
1.2.2 运行环境	1
1.3 系统可行性分析	2
1.3.1 技术可行性	2
1.3.2 硬件可行性	2
2 需求分析	3
2.1 功能分析	3
2.2 角色分析	4
3 开发环境及开发工具的介绍	5
3.1 简单介绍 Eclipse 和 MyEclipse	5
3.2 Servlet 技术	5
3.2.1 Servlets 工作原理及概述	5
3.2.2 特点及应用	6
3.3 Struts 技术	7
3.3.1 概述	7
3.3.2 整体结构概述	7
3.3.3 Struts 优点	7
3.4 Hibernate	8
3.4.1 Hibernate 数据持久化实现	8
3.4.2 使用 hibernate 编程步骤	8
3.4.3 Hibernate 控制的事务	9
3.4.4 Hibernate 的优点	9
4 数据库设计	11
4.1 Customer 表	11
4.2 Contract 表	11

4.3 Staff 表	12
5 系统总体设计	13
5.1 设计思想	13
5.2 框架设计	13
5.3 系统界面要求	14
5.4 系统模块的划分:	15
5.5 主要系统模块的具体实现	16
5.5.1 职工或管理员的登录模块	16
5.5.2 添加职工模块	17
5.5.3 职工或管理员的修改和删除模块	18
5.5.4 查询客户信息	19
5.5.5 合同信息的录入	20
5.5.6 合同信息的签署	22
6 系统的测试	23
6.1 输入测试	23
6.2 部分功能单元测试	25
7 总结与体会	27
8 致 谢	28
9 参考文献	29

1 引言

1.1 企业合同管理系统论述

随着信息时代的到来，信息管理技术受到越来越多企业的高度重视，逐步成为现代化的发展不可缺少的一部分。

《企业合同管理信息系统》是一个计算机信息管理系统，其中包括合同所有方面的操作。企业产品制造合同的成立，是一个企业在市场经营的销售是最重要的部分。因此，利用计算机信息技术管理是企业的实际需要。

随着经济和科学技术的进步，电脑业务，迅速发展的计算机和通信技术为基础的信息系统在一个蓬勃发展的时期迅速发展，信息管理系统领域正在发生重大变化。在企业管理的计算机应用的普及，计算机和文件企业合同管理中的应用势在必行。

随着企业间竞争日趋激烈，企业管理体系的要求也越来越高，《企业制造合同管理信息系统》在商业基础上运作，按照统筹操作的逻辑思维，销售合同，销售交付，成果和历史数据紧密衔接在一起。

本系统使用开发工具 MyEclipse，结合实际合同文档管理，按照大部分企业的制度，进行实际的需求分析。

1.2 开发环境及运行环境

1.2.1 开发环境

开发工具：MyEclipse 7.5

数据库：Microsoft SQL Server 2005

文档工具：Microsoft Office Word 2003

1.2.2 运行环境

操作系统：Microsoft Windows 7, Microsoft Window XP

1.3 系统可行性分析

1.3.1 技术可行性

本次开发采用 MyEclipse 作为编程工具,使用 Java 语言实现系统功能。Java 语言是由 SUN 公司开发的一种功能强大、简单、先进,面向对象的全新程序设计语言。Java 继承了 C++的优点,也抛弃 C++的一些弊端,具有 C++语言的强大功能,使用起来又非常简单易懂。

Java 开发的程序具有“编写一次到处运行”的巨大能力。而且,借助其 JavaAPI 及其编程语言上的 Java 虚拟机,它已产生一种分布信息的崭新模式。这种模式叫做 Java 企业计算,正在帮助各企业以各种不同的方法取得竞争优势。网管和控制已大大简化。软件分配基本上免费的,而且立即可实现。电子贸易已获得。占有成本大幅度降低。信息和应用软件到处可存取

Microsoft SQL Server 2005 建立于成熟的和强有力的关系模型上,是当今首选的数据存储和检索模型。Microsoft SQL Server 2005 除了具有与其它数据库类似的特征外,还有许多独到之处。这要归功于 SQL Server 2005 与 Windows NT 操作系统的紧密继承。因此,后台有强大的 SQL Server 2005 数据库支持,能保证系统开发的顺利进行。

1.3.2 硬件可行性

本网站的设计是在 WINDOWS XP 系统上进行的,即便如此设计出的动态网页也同样可以用于其它 WINDOWS 系统中。

2 需求分析

2.1 功能分析

根据实际合同管理系统的行为规范，具体功能包括：

客户管理：。有客户的注册，修改和删除功能,以及客户名，地区进行查询。

合同管理：。有合同的录入，修改和删除，另外按时间，客户，客户地区，以及按合同的进度进行查询；

职工管理：。有职工的添加，修改和删除功能，另外有查询所有的职工。

功能描述如图 2-1：

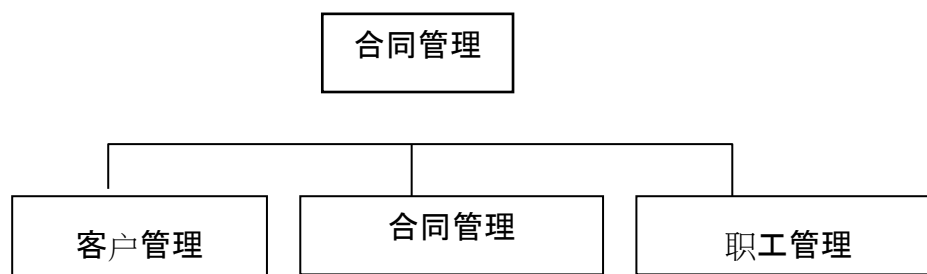


图 2-1 功能描述图

1) 客户管理

a) 客户基本信息的维护：

- 客户信息的录入
- 客户信息修改
- 客户信息删除

b) 客户信息的查询：

- 按地区查询
- 按客户名称查询

2) 合同管理

a) 合同的基本编辑

- 合同的录入
- 合同的修改

b) 合同的维护

- 合同录入后，交付上级管理人员、部门审核，负责人签名
- 合同变更，相关管理人员对合同内容进行变更，负责人签名

- 合同执行进度管理，对合同的执行情况进行记录
- c) 合同统计查询管理
 - 合同订单的查询（按时间、客户、客户地区查询）
 - 按照合同的完成情形查询(审核, 未审核)
- 3) 职工管理
 - a) 职工基本信息的维护：
 - 职工信息的录入
 - 职工信息修改
 - 职工信息删除
 - b) 客户信息的查询：
 - 集体职工查询

2.2 角色分析:

根据实际情况，本系统分为三种角色：客户，职工，管理员。

客户：

委托职工或者管理员进行合同的办理；

职工：

受客户委托进行合同办理，合同的管理，录入，修改，查询，删除；

管理员：

受客户委托进行合同办理，合同的管理，录入，修改，查询，删除；
对职工进行管理，职工的添加，修改，查询，删除。

3 开发环境及开发工具的介绍

3.1 简单介绍 Eclipse 和 MyEclipse

Eclipse 是一个开源的基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务建设的发展环境而设置插件组件。幸运的是，Eclipse 附带了一个标准的插件，包括 Java 开发工具（Java 开发工具，JDT 的）。这种平等和一致性并不仅限于 Java 开发工具。

尽管 Eclipse 是使用 Java 语言的发展，但它的使用并不限于 Java 语言；例如，支持诸如 C/ C + +中，COBOL 和艾菲尔编程语言，如插件已经可用，或预计将介绍。Eclipse 框架和软件的发展也可以被用来作为其他应用程序类型，比如内容管理系统的基礎上独立。

MyEclipse 是 Eclipse 的插件，一个强大的集成开发环境，支持编码，配置，测试和调试。Genuitec 已经发布了 MyEclipse 的企业 Workbench 5.0 对，更聪明，更快，更容易和更便宜的 J2EE 工具的新版本诞生了。它为个人和企业开发的价格是非常有吸引力。这是一个重量级的 J2EE IDE 的市场主体。通过增加双向 UML 建模工具，用于 JSP/ Strutsdesigner 所见即所得，直观的 Hibernate/ ORM 工具，Spring 和 Web 服务支持，以及新的 Oracle 数据库开发，MyEclipse 的 5.0 继续为客户提供全面的产品行业。Genuitec 总裁马希尔马斯里说：“今天，MyEclipse 已经提供了意料之外的价值。其中的每个功能在市场上单独的价格都比 MyEclipse 要高。但是，我们承诺为顾客提供全面并且可以买得起的解决方案。按照这个传统，我们的顾客将继续享受年度订购活动的好处，该活动提供了所有发布功能的入口以及伴随 MyEclipse5.0 一起的专业技术支持。”

3.2 Servlet 技术

3.2.1 Servlets 工作原理及概述

Servlets 是用 Java 语言实现的独立于协议和平台的服务器端组件，它动态扩展了支持 Java 的服务器，它可以被插入到支持 Java 的 Web 服务器中以提供定制服务，包括：

- 增加新的特性服务

- 页面内容的动态改变 (Runtime changes)
- 页面显示的动态改变 (Runtime changes)
- 对新的标准协议的支持(例如 FTP)
- 对用户定制协议的支持

Servlets 在服务器内部工作，它们不需要图形用户接口，作为 Java 程序组件，它们根据需要在服务器内部被部署。而使用 Servlets 的客户多种多样，既有简单的 HTML 表单，也有复杂的 Java Applets。Servlets 被设计在请求/应答 (Request/Response) 处理模式下工作，在该模式下，客户向服务器发送请求消息，服务器则通过发送应答消息予以回复。用户定制协议在通常情况下，请求和其相关应答可以反映请求时刻客户和服务器的状态，而客户和服务器之间连接的状态无法在多个请求/应答之间保存，而 Servlet 所提供的会话 (Session) 机制则可以在多次请求/应答之间保存会话信息。Servlets 有效地扩展了基于请求/应答的服务器 (例如 Web 服务器) 的功能。Java Servlet API 包括若干 Java 接口 (Java interfaces) ，全面定义了主机服务器与 Servlets 之间的连接。Servlet API 属于标准 Java 开发包 (JDK) 扩展，被包含在如下 Java 包中： javax.servlet . javax.servlet.http Servlets API 包括了很多有用的特性，如会话跟踪 (Session tracking)、表单数据、共享数据 (Shared data)、国际化字符 I/O，初始化参数采集 (Form data parsing) 数、请求代理 (Request delegation) 等。

3.2.2 特点及应用

Servlets 是对 Java 环境的一个功能强大的补充，具有快速、安全、可靠和 100% 纯 Java 的优点。因为 Servlets 被插入到已经存在的服务器中运行，可以有效利用大量现成代码和技术，所有有关网络连接、协议协商、类装载等工作都可以由服务器代劳，极大地减少了重复工作。Servlet API 提供了服务器和 Servlets 之间的紧密连接，它们允许为服务器增加新的协议支持。Servlet API 包中已经实现了对 HTTP 的很好支持。从本质上说，任何遵循请求/应答工作模式的协议都可以通过一个 Servlet 来实现, 包括：

- SMTP
- POP
- FTP

目前主要的 Web 服务器都实现了对 Servlets 的支持，而越来越多的其他类型的应用程序服务器也开始实现对 Servlets 的支持。由于 HTTP 是最为常用的协议之一，Servlets 在基于 HTTP 的系统中得到了最为广泛的应用。相对于其他服务器端技术如 CGI、Web 服务器 API 扩展

(NSAPI、ISAPI、Apache Modules) 等, Servlets 有如下优势:

- 由于使用了不同的处理模型和处理机制, Servlets 在速度和效率上高于 CGI, 在功能和实现灵活性上也要强于 CGI, 有取代 CGI 的趋势。
- Servlets 使用了被许多 Web 服务器产品支持的标准 API, 具有较强通用性。
- Servlets 具有 Java 编程语言的所有优点, 包括易于开发、平立性等。
- Servlets 可以访问基于 Java 平台和 Java 技术的大量 API, 能很好地和目前主流体系结构 (如 J2EE) 融合, 具有高灵活性和高扩展性, 适用于构造大型复杂的应用系统。

3.3 Struts 技术

3.3.1 概述

struts 是基于当前的一些相关标准的开发技术 (servlet, xml...) 基础之上, 提供了一个开发框架的开放源码项目, 由于我们开发 web 应用的复杂度随着系统的复杂度的要求越来越来复杂。特别是在代码重用, 代码移植、代马可插扒等问题上出现了许多重复开发、维护困难等。Struts 这个框架鼓励我们应用 model2 的 MVC 开发模型。当然我们在基于其上的开发中不一定要进行完全遵守。当前 struts 比较稳定的发行版时 1.0, 已经有了 1.1 的测试开发版。

3.3.2 整体结构概述

在到处可以找到关于 MVC 结构和 struts 结构的图形。这里我就不对结构图进行描述。只站在对 J2EEWEB 应用的相关标准了解的基础来对 struts 整个框架进行一描述。

其实 struts 框架完全是基于 J2EE 相关技术和标准之上的。Struts 提供一个 servlet 来作为控制器对客户端的相应请求进行转接给后台的模型。它为了使各个部分能够分离, 提供相应的配置文件 (struts-config.xml) 进行请求和处理类的映射。整个类的映射完全来自 java 技术的类反射和相应的请求过滤技术。通过标记扩展来实现相应的视图和模型的联系。

3.3.3 Struts 优点

Struts 跟 Tomcat、Turbine 等诸多 Apache 项目一样，是开源软件，这是它的一大优点。使开发者能更深入的了解其内部实现机制。

除此之外，Struts 的优点主要集中体现在两个方面：Taglib 和页面导航。Taglib 是 Struts 的标记库，灵活动用，能大大提高开发效率。另外，就目前国内的 JSP 开发者而言，除了使用 JSP 自带的常用标记外，很少开发自己的标记，或许 Struts 是一个很好的起点。

关于页面导航，我认为那将是今后的一个发展方向，事实上，这样做，使系统的脉络更加清晰。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处。尤其是当另一批开发者接手这个项目时，这种优势体现得更加明显。

缺点：

Taglib 是 Struts 的一大优势，但对于初学者而言，却需要一个持续学习的过程，甚至还会打乱你网页编写的习惯，但是，当你习惯了它时，你会觉得它真的很棒。

Struts 将 MVC 的 Controller 一分为三，在获得结构更加清晰的同时，也增加了系统的复杂度

3.4 Hibernate

3.4.1 Hibernate 数据持久化实现

1. hibernate 是一种 ORM(object relation mapping, 对象关系映射)框架，所谓的对象关系映射，通俗的说，就是把 JAVA 对象保存到关系型数据库中，即进行持久化的框架。

2. hibernate 进行持久化首先要建立对象到关系数据库的映射。

3. hibernate 根据定义的映射规则将对象持久化（保存到）数据库存中。这就实现了对象持久化。

3.4.2 使用 hibernate 编程步骤

1, 配置环境, 加载 hibernate 的 jar 文件, 以及连接数据库连接使用的 jar 文件, 并配置 CLASSPATH 环境变量。

2, 写 hibernate 所需的配置文件, hibernate.cfg.xml , Xxxxx.hbm.xml

3, 写 POJO 类

4, 调用 hibernate API。

1)使用 Configuration 对象的 buildSessionFactory() 方法创建 SessionFactory 对象

2)使用 SessionFactory 对象 openSession() 方法创建 Session 对象。

3)使用 Session 的相应方法来操作数据库, 将对象信息持久化到数据库。

3.4.3 Hibernate 控制的事务

事务保证原子操作的不可分, 也就是操作的同时成功或同时失败。

```
Transaction tran=session.beginTransaction();
```

```
tran.commit();
```

```
tran.rollback();
```

以上是事务对象的方法, 来实现对事务的支持。

hibernate 的事务隔离级别

hibernate 的事务隔离级别和 JDBC 中大致相同。

设置时要在 hibernate.cfg.xml 配置

```
<property name="hibernate.connection.isolation">4</property>
```

1, 读未提交的数据 (Read uncommitted isolation)

2, 读已提交的数据 (Read committed isolation)

4, 可重复读级别 (Repeatable read isolation)

8, 可串行化级别 (Serializable isolation)

hibernate 的锁 (悲观锁, 乐观锁)

悲观锁是由数据库本身所实现的, 会对数据库中的数据进行锁定, 也就是锁行。

LockMode.UPGRADE, 修改锁, 在 get() 方法中加上这个设置作为第三个参数。

LockMode.NONE 无锁机制

LockMode.READ 读取锁

LockMode.WRITE 写入锁，不能在程序中直接使用

还可以使用 `Session.lock()` `Query.setLockMode()` `Criteria.setLockMode()` 方法来设置锁

3.4.4 Hibernate 的优点

(1) 对象/关系数据库映射(ORM)

它使用时只需要操纵对象，使开发更对象化，抛弃了数据库中心的思想，完全的面向对象思想

(2) 透明持久化(persistent)

带有持久化状态的、具有业务功能的单线程对象，此对象生存期很短。这些对象可能是普通的 JavaBeans/POJO，这个对象没有实现第三方框架或者接口，唯一特殊的是他们正与（仅仅一个）Session 相关联。一旦这个 Session 被关闭，这些对象就会脱离持久化状态，这样就可被应用程序的任何层自由使用。（例如，用作跟表示层打交道的数据传输对象。）

(3) 事务 Transaction(org.hibernate.Transaction)

应用程序用来指定原子操作单元范围的对象，它是单线程的，生命周期很短。它通过抽象将应用从底层具体的 JDBC、JTA 以及 CORBA 事务隔离开。某些情况下，一个 Session 之内可能包含多个 Transaction 对象。尽管是否使用该对象是可选的，但无论是使用底层的 API 还是使用 Transaction 对象，事务边界的开启与关闭是必不可少的。

(4) 它没有侵入性，即所谓的轻量级框架

(5) 移植性会很好

(6) 缓存机制，提供一级缓存和二级缓存

(7) 简洁的 HQL 编程

4 数据库设计

由于合同系统功能复杂，并且要存取的数据较多，因此简单地使用文件夹来存储数据库和管理显然是不合适的。所以本系统使用 MySQL 数据库来存储数据。根据系统需求分析需要创建以下的数据表来存储数据。

4.1 Customer 表

Customer 表用来保存客户注册的一些基本信息，并且同时保存一些 staff 信息，设计出 Customer 表如表 4-1:

表 4-1 Customer 表

字段名称	数据类型	说明
id	int	用户 id
name	varchar	用户名
Phone	varchar	电话号码
Fox	Varchar	传真
postcode	Varchar	电子邮件
Sex	Varchar	性别
idcNum	Varchar	身份证
regDate	Date	注册时间
address	Varchar	详细地址
Staff_if	Int	职工或管理员 ID

4.2 Contract 表

Contract 表用来保存合同的一些基本信息，和客户的一些信息，设计出 Contract 表如表 4-2:

表 4-2 Contract 表

字段名称	数据类型	说明
Id	Int	合同 ID
name	Varchar	合同名称
Description	Varchar	合同描述
submitDate	Date	录入时间
Content	Varchar	合同内容
Status	Int	状态
Customer_id	int	合同对应的客户 Id

4.3 Staff 表

Staff 表用来保存用户的一些基本信息，设计出 Staff 表如表 4-3:

表 4-3 Staff 表

字段名称	数据类型	说明
Id	int	职工或者管理员 Id
name	Varchar	用户名字
Password	Varchar	用户密码
flag	Varchar	权限

5 系统总体设计

5.1 设计思想:

越来越重视现代科学和技术的企业，只有科学技术的企业为了获取更大的效益。合同是商品经济的产物，是商品交换的法律表现形式。现代企业的经济关系，主要是进行通过合同的形式。

因此，合同管理，建立现代企业制度的法律的律师，一个有效的管理可以推迟合同的重要方面，现在是成功还是失败，是企业管理的一个重要因素。

企业往往订立合同，并考虑如何管理合同，合同用户的信息化进程。提高合同管理经营是健康的重要标志

设计原理图如下图:

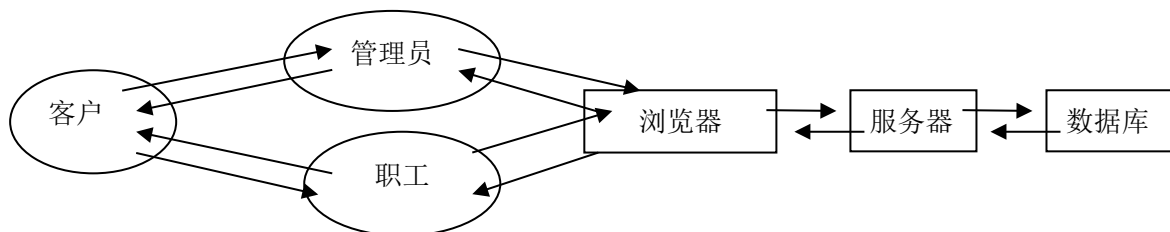


图 5-1 设计原理图

5.2 框架设计:

web 开发三层结构:

交互层，表现数据，收集数据，也就是视图层，接受请求，通过处理层完成相应的响应。(V, C) 处理层，完成数据的处理，也就是业务层 (M)。

MVC 架构模式强行的业务的表现和业务的处理分开，最大限度得减少软件各层之间的偶合性，软件可以更好维护和扩展。当你改变业务的表现时不会影响到业务的处理，扩展业务时不会影响到业务的表现。

StrutsMVC 框架

Struts 是使用 MVC 的实现模式二来实现的，也就是以控制器为核心。Struts 提供了一些组件使用 MVC 开发应用程序：

Model: Struts 没有提供 model 类。这个商业逻辑必须由 Web 应用程序的开发者以 JavaBean 或 EJB 的形式提供

View: Struts 提供了 action form 创建 form bean，用于在 controller 和 view 间传输数据。此外，Struts 提供了自定义 JSP 标签库，辅助开发者用 JSP 创建交互式的以表单为基础的应用程序，应用程序资源文件保留了一些文本常量和错误消息，可转变为其它语言，可用于 JSP 中。

Controller: Struts 提供了一个核心的控制器 ActionServlet，通过这个核心的控制器来调用其他用户注册了的自定义的控制器 Action，自定义 Action 需要符合 Struts 的自定义 Action 规范，还需要在 struts-config.xml 的特定配置文件中进行配置，接收 JSP 输入字段形成 Action form，然后调用一个 Action 控制器。Action 控制器中提供了 model 的逻辑接口。

MVC 结构的框架如下：

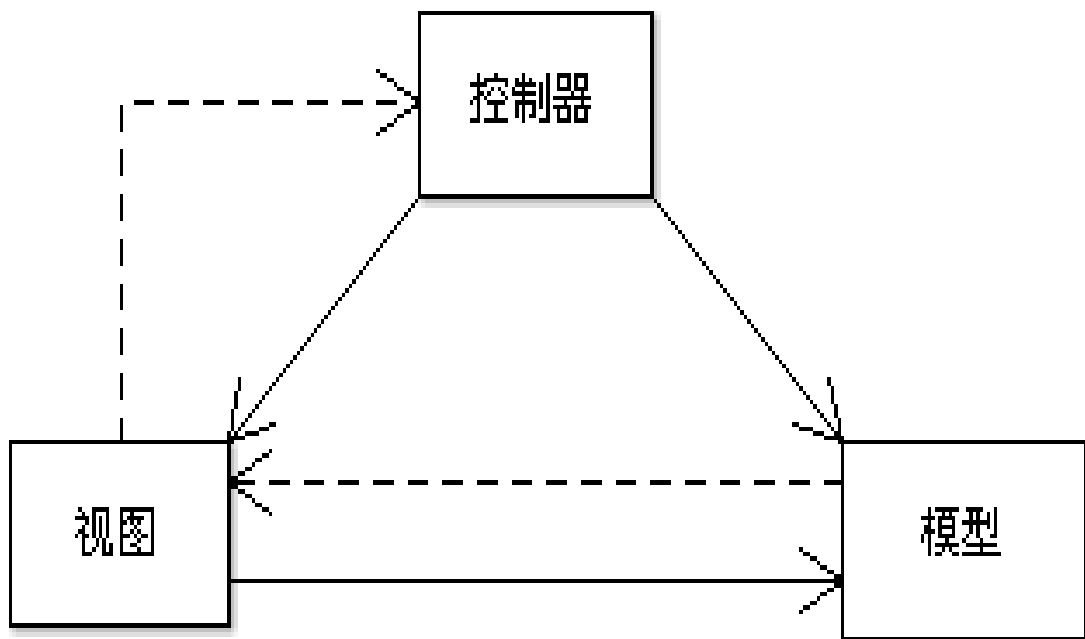


图 5-2 MVC 架构图

5.3 系统界面要求

系统的客户端和服务端初始界面假定用户是未登录，主页面显示登陆窗口，通过初

始界面提供的统一的用户登录接口，登录进入系统，因此系统主页面应该能自动进行身份判别，按照访问者的身份自动进入其相应的功能权限的管理

设计如下：

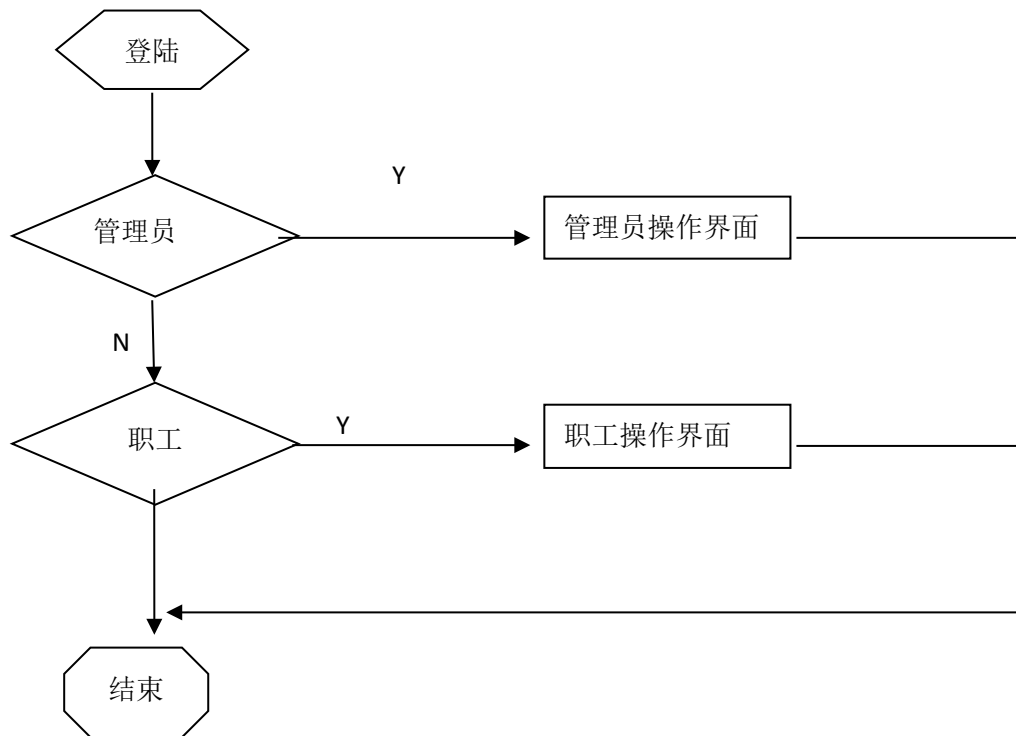


图 5-3 系统设计图

5.4 系统模块的划分：

合同管理系统的核心内容就是方便公司内部能够有效的对客户的需求进行管理，以得到广大新老客户的光顾，从而让企业壮大。

系统的总体用例图如下：

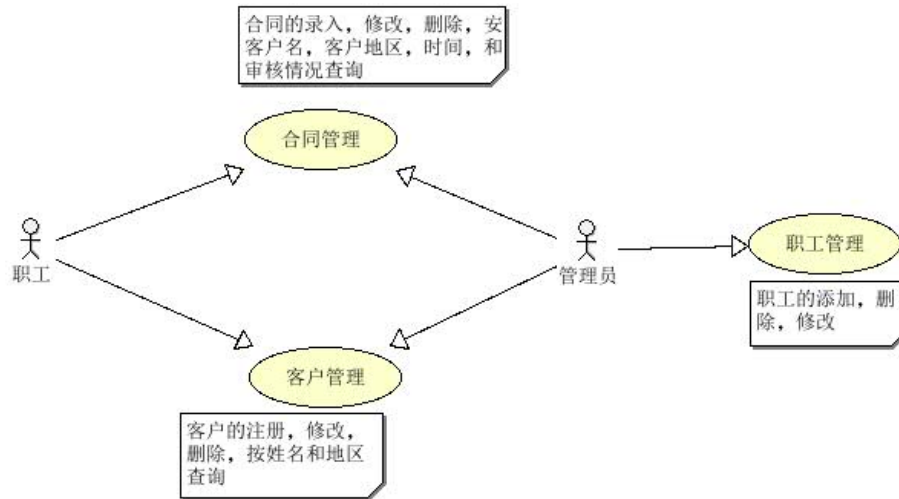


图 5-4 系统总体用例图

5.5 主要系统模块的具体实现

5.5.1 职工或管理员的登录模块

职工或管理员的登录模块一个 login 连接，默认为职工或管理员的登录界面，系统根据帐号和密码以及权限设置登录到不同的页面，若职工权限进入客户管理和合同管理模块，若管理员权限进入客户管理，合同管理以及职工管理模块。

步骤：

1. 输入用户和密码，选择用户权限即职工或管理，点“登录”；
2. 根据用户名和密码，系统进行数据库连接，若没有找到，返回错误信息：用户名，密码或用户权限错误返回当前登录页面；
3. 若找到用户名和密码，根据权限设置，登录系统操作页面，若 a 登入管理员页面，若 s 登入职工管理页面。以下为关键代码：

```

public ActionForward login(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {

    StaffForm sform = (StaffForm) form;
    String name = sform.getName();
    String password = sform.getPassword();
    String flag = sform.getFlag();
    Staff staff = new Staff();
    staff.setName(name);
    
```

```

staff.setPassword(password);

StaffService ss = new StaffServiceImpl();
Staff sta = ss.findStaff(name, password);

if (sta != null)
{
    if (flag.equals(sta.getFlag()))
    {
        request.getSession().setAttribute("sta", sta);
        return mapping.findForward("login");
    }
    else if (flag.equals(sta.getFlag()))
    {
        request.getSession().setAttribute("sta", sta);
        return mapping.findForward("login");
    }
    else
    {
        request.setAttribute("errors", "用户名、密码或权限不对");
    }
}
else
{
    request.setAttribute("errors", "用户名、密码或权限不对");
}
return mapping.findForward("back");
}

```

5.5.2 添加职工模块

添加职工模块管理员进入了管理员模块进行添加职工。

步骤:

1. 点 职工管理→添加职工，进入添加职工页面；
2. 根据管理员的要求添加另一个管理员和职工，新创的管理员具有和该管理员一样的权限，输入要创建的用户名和密码以及新的职工类型；
3. 点注册系统根据输入的帐号访问数据库判断是否存在，如果存在，返回错误信息，如果帐号不存在，注册成功，提示成功信息，返回添加职工页面。管理员可以继续添加职工。 以下分别为添加职工如图 5-1 所示：



常熟理工学院 陆参

图 5-1 员工添加图

5.5.3 职工或管理员的修改和删除模块

管理员进入管理员界面后进行职工信息的删除和修改。

步骤:

1. 点 所有职工进入职工的修改和删除页面;
2. 如果点修改跳到所选职工信息修改页面, 对职工的信息进行修改, 点修改, 系统根据数据对该职工的信息进行修改(先找到该用户, 然后更新), 返回当前页面, 提示修改成功信息; 如果点删除删除当前所选的职工, 当前所选职工信息已经不存在了返回到当前页面; 以下为关键代码:

修改部分:

```
public ActionForward modifyCustomer(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) {

    CustomerService cs=new CustomerServiceImpl();
    CustomerForm cform=(CustomerForm) form;
    int sid= Integer.parseInt(request.getParameter("sid"));
    String name=cform.getName();
    String address=cform.getAddress();
    String fox=cform.getFox();
    String sex=request.getParameter("sex");
    String postcode=cform.getPostcode();
    String idcNum=cform.getIdcNum();
```

```

String phone=cform.getPhone();

Customer c=cs.findCustomerById(sid);
if(c!=null) {
    c.setAddress(address);
    c.setFox(fox);
    c.setPhone(phone);
    c.setPostcode(postcode);
    c.setIdcNum(idcNum);
    c.setSex(sex);
    cs.modifyCustomer(c);
    request.setAttribute("cinfo", "修改成功!");
    request.setAttribute("c", cs.findCustomerById(sid));
} else{
    request.setAttribute("cinfo", "修改失败");
}

return mapping.findForward("modifyC");
}

```

删除部分:

```

public ActionForward delCustomers(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) {

    CustomerService cs=new CustomerServiceImpl();

    int cid= Integer.parseInt(request.getParameter("sid"));

    cs.removeCustomer(cs.findCustomerById(cid));

    return mapping.findForward("list");
}

```

5.5.4 查询客户信息

查询客户信息职工或管理员进入登录页面，对客户进行按姓名查询。

步骤:

1. 点客户查询 →按姓名查询，输入姓名，点查询；
2. 系统根据所提供的姓名这个信息对数据库进行查询，若找到这个客户所有对应的信息，跳转到另一个显示界面，显示该用户的信息；
3. 点客户查询 →按地区查询，输入地区名，点查询；

4. 系统根据所提供的姓名这个信息对数据库进行查询，若找到这个地区所有对应的信息，跳转到另一个显示界面，显示该地区的信息表单；以下为操作所涉及的关键代码：

```
public ActionForward findCustomersByName(ActionMapping mapping, ActionForm
form,
    HttpServletRequest request, HttpServletResponse response) {

    CustomerService cs=new CustomerServiceImpl();
    CustomerForm cform=(CustomerForm) form;
    String name=cform.getName();
    Customer customer=cs.findCustomerByName(name);
    if(customer!=null){
        request.setAttribute("customer", customer);
    }else{
        request.setAttribute("cinfo", "用户不存在");
    }
    return mapping.findForward("list");
}

public ActionForward findCustomersByAddr(ActionMapping mapping, ActionForm
form,
    HttpServletRequest request, HttpServletResponse response) {
    CustomerForm cform=(CustomerForm) form;
    String address=cform.getAddress();
    CustomerService cs=new CustomerServiceImpl();
    List<Customer> cuss=cs.findCustomerByAddress(address);
    if(cuss!=null&&cuss.size()!=0){
        request.setAttribute("cstm", cuss);
    }else {
        request.setAttribute("cinfo", "这个地区没有客户!!!");
    }
    return mapping.findForward("list");
}

public ActionForward findAllCustomers(ActionMapping mapping, ActionForm
form,
    HttpServletRequest request, HttpServletResponse response) {

    CustomerService cs=new CustomerServiceImpl();
    List<Customer> cstm=new ArrayList<Customer>();
    cstm=cs.findall();
    if(cstm!=null&&cstm.size()!=0){
        request.setAttribute("cstm", cstm);
    }else{
```

```

        request.setAttribute("cinfo", "没有一个客户");
    }
    return mapping.findForward("list");
}

```

5.5.5 合同信息的录入

合同信息的录入职工或管理员进入合同管理页面，对合同进行管理。

步骤：

1. 点添加合同显示添加合同页面，录入合同页面指定的一些信息，点提交；
2. 系统根据合同名对数据进行查询，如果有该合同名，返回当前页面，提示录入失败；若没有，在当前提示录入成功，以下为页面操作所涉及的页面及关键代码：

```

public ActionForward addContract(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response
    ) throws Exception{
    ContractService cons=new ContractServiceImpl();
    CustomerService cuss=new CustomerServiceImpl();
    ContractForm cform=(ContractForm) form;
    String cus_name=request.getParameter("cus_name");
    String con_name=cform.getName();
    String content=request.getParameter("content");
    System.out.println(content.length()==0);
    String description = cform.getDescription();
    int status=cform.getStatus();
    Contract con=cons.findContractsByName(con_name);
    if(con!=null){
        Customer cus=cuss.findCustomerByName(cus_name);
        if(cus!=null){
            con.setCt(cus);
            cus.addContract(con);
            request.setAttribute("cinfo", "为指定的客户添加已存在的合同成功!");
        }else if(!cus_name.equals("")&&cus==null){
            request.setAttribute("cinfo", "对不起,该客户不存在,请先为其注册!");
        }else if(cus_name.equals("")){
            request.setAttribute("cinfo", "该合同名已存在,请用其他合同名!");
        }
    }
    else {
        Customer cus=cuss.findCustomerByName(cus_name);
        if(cus!=null){
            Contract c=new Contract();
            c.setContent(content);
            c.setName(con_name);

```

```

        c.setDescription(description);
        c.setStatus(status);
        c.setSubmitDate(new java.sql.Date(new
java.util.Date().getTime()));
        c.setCt(cus);
        cons.addContract(c);
        cus.addContract(c);

        request.setAttribute("cinfo", "为指定的客户添加新合同成功!");
    }else if(!cus_name.equals("")&&cus==null){
        request.setAttribute("cinfo", "对不起,该客户不存在,请先为其注册!");
    }else if(cus_name.equals("")&&(!content.equals(""))){
        Contract c=new Contract();
        c.setContent(content);
        c.setName(con_name);
        c.setDescription(description);
        c.setStatus(status);
        c.setSubmitDate(new java.sql.Date(new
java.util.Date().getTime()));
        cons.addContract(c);
        request.setAttribute("cinfo", "添加新合同成功!");
    }else{
        request.setAttribute("cinfo", "请仔细填写合同内容!!");
    }
}
return mapping.findForward("addCon");
}

```

5.5.6 合同信息的签署

管理员进入合同管理页面，对合同进行签署。

步骤：

1. 到查询合同页面，根据姓名查询合同，若该客户不存在提示没有该客户，若有客户而没有合同，提示该客户没有合同，若该客户有合同得到按姓名查询的合同页面；
2. 点审核修改合同的签署状态；效果图如图(5-6)所示

管理目录	合同录入	合同查找	所有合同			
合同管理						
客户管理	按客户查找	按地址查找	按时间查找		按审核查询找	
员工管理						
退出系统						
	序号	合同名	合同描述	合同状态	修改操作	删除操作
	1	test01	test	未签署	修改	删除

图 5-6 修改合同签署状态图

6 系统的测试

在每个模块设计完成之后，对系统进行单元测试，确定其能否达到功能需求所预期的目标。单元测试完成后，对系统进行各种综合测试。经过反复的测试，工业制造企业合同管理系统基本达到功能需求的要求。下面给出部分测试的图例。

6.1 输入测试

当输入框为空时，进行页面提交。如图 6-1 6-2 所示

企业合同管理信息系统

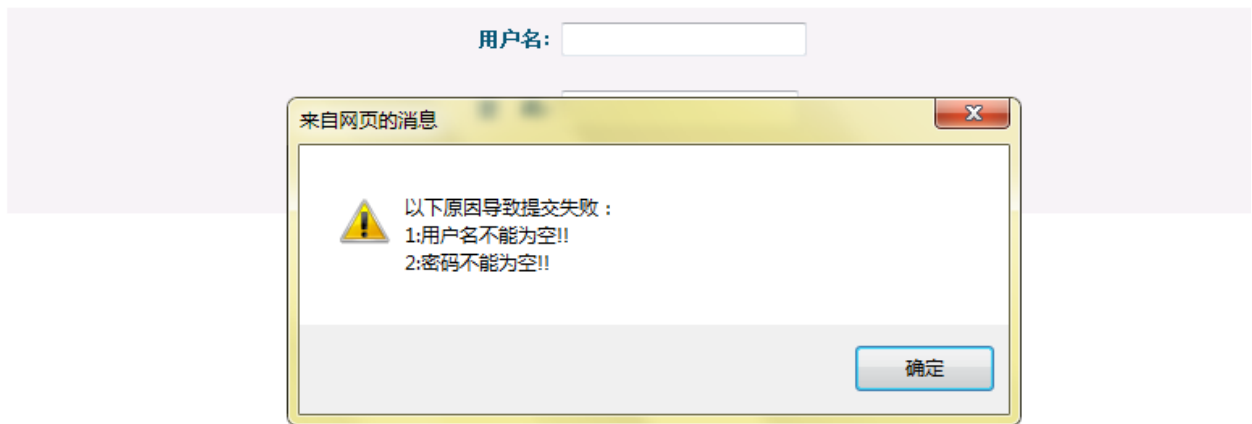


图 6-1 登录界面输入为空图



图 6-2 合同添加页面输入为空图

当输入为字符串，符号等内容时，依旧进行正常匹配。如图 6-3 所示。



图 6-3 使用字符串，符号登录图

6.2 部分功能单元测试

合同录入界面 客户方不存在时。如图 6-4 所示



图 6-4 非关联客户录入

查询客户，合同等，均按照全匹配查询。

使用客户名 1 进行查找，如图 6-5。

管理目录	合同录入	合同查找	所有合同	
合同管理				
客户管理	按客户查找	按地址查找	按时间查找	按审核查询找
员工管理	没有这个客户			
退出系统				

常熟理工学院 陆参

图 6-5 不支持模糊查找图

使用客户名： 客户 001 进行查找，如图 6-6

管理目录	合同录入	合同查找	所有合同			
合同管理						
客户管理	按客户查找	按地址查找	按时间查找	按审核查询找		
员工管理						
退出系统						
	序号	合同名	合同描述	合同状态	修改操作	删除操作
	1	test01	test	未签署	修改	删除

常熟理工学院 陆参

图 6-6 全匹配查找图

7 总结与体会

这次的毕业设计是 JSP / Struts 的基于 WEB 的合同管理系统的开发与设计，该系统的总体格局是 MVC，Struts 框架。在具体使用时，它的功能是实现合同管理，添加和删除客户资料以及员工资料，签署合同，合同管理。本次毕业设计是我大学生活的知识，全面的，使我对软件设计和开发的全过程，有了较为直接的认识。

首先，系统的需求分析，分析系统，实现什么系统应设计成若干模块划分功能，系统应该如何实现具体的功能；分析完成后，整体设计，框架设计，数据库设计，考虑到计算机相关专业知识，完善问题。然后分解成一个个大问题和小的问题，然后逐一解决。

这次毕业设计是对我大学四年所学知识的一个综合，从中我熟练掌握了软件设计的思想及其开发的全过程。

首先要进行系统的需求分析，分析系统要达到什么功能，系统要划分为几个模块来设计，系统的具体功能应当如何去实现；分析完之后，进行总体设计，框架设计，数据库设计，考虑其涉及到的计算机专业知识，把问题细化，把大的问题划分为小的问题，然后逐个进行解决。

毕业设计开始阶段，由于自己初次接触 JSP/Struts 作具体的项目系统，在这方面知

道的很少，作的过程中遇到了很多 BUG 难调，通过查阅相关书籍以及在互联网上进行相关资料的搜索，逐渐开始对所接触的事物有了一定地了解和把握。

本次的项目制作，很多困难的出现，只是给了我自学的机会，以及自己知识的增长。一次次的 BUG，也使自己的代码调试能力得到了大幅度的提升。

同时感觉到软件开发的乐趣，软件开发是自我挑战，实现自我，表现自己能力的一个途径。现在看到代码就有点莫名的冲动，有点一天不敲代码，心里痒痒的感觉，见到代码异度兴奋。底层的代码，所体现的视觉感，总让人感到一个好奇感。几句代码怎么就能表现出人们所想的视觉效果呢。所以在我以后的软件开发的路途中，我会克服困难，坚决书写自己的理想。

个人的实践能力有限，缺乏经验，系统可以基本满足合同管理，但是系统的功能还是过于简单并且不是很完美的，这个是个人的能力不强造成的。希望老师和同学可以提出建议和指导。

8 致 谢

毕业设计是大学四年的最后一门课程，也是综合运用所学知识的课程。平常上课学习的都是理论知识，然而亲自动手实践的机会很少，这次毕业设计为我提供了一个很好的机会，可以让我将所学的理论知识运用到实践中，是检验理论知识学习成果的最好方法，例如：在运用到 Struts 时，遇到很多 BUG，经过仔细的调试，才能运行通过，所以说实践是实现理论的法宝。在本次毕业设计过程中，要特别感谢我的指导老师Z老师，在整个设计过程中对我们要求很严格，从而能顺利完成系统的设计以及论文的书写。每次和Z老师见面时，他都会及时地提出我们设计过程中的不足，指导我们如何去解决问题，从中我们也学到了解决问题的方法。设计过程中老师教授的知识，我们将终身受用。

9 参考文献

- [1] TODD COOK ,JSP 从入门到精通[M]. 北京：电子工业出版社, 2003。
- [2] 金林樵. 网络数据库技术及应用[M] .北京：机械工业出版社，2002。
- [3] 李盛恩，王珊. 数据库基础与应用[M] .北京：人民邮电出版社，2002。
- [4] 吴以欣，陈小宁. JavaScript 脚本程序设计[M] .北京：北京人民邮电出版社[M], 2005。
- [5] 郝玉龙. J2EE 编程技术[M]. 北京：清华大学出版社, 2005。
- [6] 金勇华. JAV 网络高级编程[M]. 北京：人民邮电出版社，2001。
- [7] 张大治，王欢. JSP 实用教程[M]. 北京：清华大学出版社，2006。
- [8] 周桓，王殊宇 . JSP 项目开发全程实录[M]. 北京：清华大学出版社 2008。
- [9] Java 编程入门经典. 梅兴文译. 第 4 版. 北京：人民邮电出版社 2010。
- [10] Java 编程思想第四版【美】Bruce Eckerl 机械工业出版社 2007。
- [11] J2EE 构建企业系统专家级解决方案 【美】Paul J. Perrone, et al. 译者：张志伟 谭郁松 张明杰 清华大学出版社 2001