

# 西南科技大学

Southwest University of Science and Technology

## 本科毕业设计（论文）



题目： 托付物流管理系统设计与实现

学生姓名： 冯 涛

学生学号： 20136623

专 业： 信息安全

指导教师： 杨雷老师

学院(部)： 计算机科学与技术

教务处制表

# 托付物流管理系统设计与实现

## 摘要

现代社会中,随着经济全球化的不断推进,信息技术的发展已经不断成为人类生活中不可或缺的一部分。尤其是 21 世纪计算机的普及,为我们的生活带来了很多的方便,在互联网的大环境中,计算机与人类生活的关系越来越密切。现阶段,网上购物的热潮以及市场的多元化,使物流行业得到了极大的发展,物流企业将速度决定效率为企业理念,把客户成为自己的上帝,提出以客户为核心的业务流程。所以,从上来看,性能优良的物流管理系统在当下这个经济高速发展的时代,有着广阔的前景。

本物流管理系统设计思路是基于 JSP 技术与 MVC 模式,系统以 Java 语言体系结构作为开发基础,以关系型数据库 Oracle 数据库为数据支持,我的开发工具是 MyEclipse,前端页面设计使用 JSP 动态页面技术, CSS + HTML + JQuery 相结合的开发体系结构。本系统具有优美的界面,操作简单,易维护的特点。

**关键字:** 物流管理, java, JSP, Oracle 数据库, JavaScript

# DESIGN AND IMPLEMENTATION OF TRUSTED LOGISTICS MANAGEMENT SYSTEM

## ABSTRACT

In modern society, with the continuous advancement of economic globalization, the development of information technology has become an indispensable part of human life. Especially in the 21st century, the popularity of computers for our lives has brought a lot of convenience, in the Internet environment, the computer and the relationship between human life more and more close. At this stage, the upsurge of online shopping and the diversification of the market, so that the logistics industry has been a great development, logistics companies will determine the efficiency of efficiency for the corporate philosophy, the customer to become their own God, put the customer as the core business processes. Therefore, from the point of view, excellent performance of the logistics management system in the current era of rapid economic development, has broad prospects.

The logistics management system design ideas are based on JSP technology and MVC model, the system to Java language architecture as the basis for the development of relational database Oracle database for data support, my development tool is MyEclipse, front-end page design using JSP dynamic page technology, CSS + HTML + JQuery combined development architecture. The system has a beautiful interface, simple operation, easy to maintain the characteristics.

**Key words:** Logistics management, java, JSP, Oracle database, JavaScript

# 目 录

|                                |    |
|--------------------------------|----|
| ABSTRACT.....                  | 2  |
| 第一章 绪论.....                    | 1  |
| 1.1 引言.....                    | 1  |
| 1.2 研究意义.....                  | 1  |
| 1.3 国内外研究现状及发展趋势.....          | 1  |
| 第二章 需求分析.....                  | 3  |
| 2.1 编写目的.....                  | 3  |
| 2.2 系统说明.....                  | 3  |
| 2.3 开发技术与开发工具介绍.....           | 3  |
| 2.3.1 JAVA 开发语言.....           | 3  |
| 2.3.2 JSP 技术.....              | 4  |
| 2.3.3 Oracle 数据库技术.....        | 4  |
| 2.3.4 MyEclipse 简介.....        | 5  |
| 2.3.5 Tomcat 简介.....           | 5  |
| 2.3.6 JavaScript 与前端技术.....    | 6  |
| 2.4 系统开发框架.....                | 6  |
| 2.4.1 Struts2 框架.....          | 7  |
| 2.4.2 Spring 框架.....           | 7  |
| 2.5 功能需求.....                  | 8  |
| 第三章 系统分析.....                  | 9  |
| 3.1 可行性分析.....                 | 9  |
| 3.2 系统基本功能分析.....              | 9  |
| 3.3 数据库物理设计.....               | 10 |
| 第四章 系统总体设计.....                | 12 |
| 4.1 系统功能结构.....                | 12 |
| 4.2 数据表关系设计.....               | 13 |
| 4.2.1 Power Designer 软件使用..... | 14 |
| 4.3 系统概念模型设计.....              | 14 |
| 第五章 系统模块设计.....                | 16 |
| 5.1 登录注册模块.....                | 16 |
| 5.1.1 登录界面与核心代码.....           | 16 |
| 5.1.2 注册界面与核心代码.....           | 19 |
| 5.2 主界面模块.....                 | 21 |
| 5.3 订单管理模块.....                | 22 |
| 5.4 线路管理模块.....                | 26 |
| 5.5 配送点管理模块.....               | 29 |
| 5.6 车辆管理模块.....                | 31 |
| 5.7 货物管理模块.....                | 33 |
| 第六章 配置文件设置.....                | 36 |

---

|                        |    |
|------------------------|----|
| 6.1 Struts2 配置文件 ..... | 36 |
| 6.2 Spring 配置文件 .....  | 37 |
| 6.3 MyBatis 配置文件 ..... | 38 |
| 第七章 系统测试.....          | 40 |
| 7.1 软件测试的原则和方法.....    | 40 |
| 7.2 单元测试.....          | 40 |
| 7.3 整体测试.....          | 43 |
| 第八章 结论.....            | 44 |
| 参考文献.....              | 45 |
| 致谢 .....               | 46 |

# 第一章 绪论

## 1.1 引言

现代社会中,随着经济全球化的不断推进,信息技术的发展已经不断成为人类生活中不可或缺的一部分。尤其是 21 世纪计算机的普及的情况之下,为我们的生活带来了很多的方便,在互联网的大环境中,计算机与人类生活的关系越来越密切。现阶段,网上购物的热潮以及市场的多元化,使物流行业得到了极大的发展,物流企业将速度决定效率为企业理念,把客户成为自己的上帝,提出以客户为核心的业务流程。所以,从上来看,性能优良的物流管理系统在当下这个追求快速发展的时代,有着宽阔的发展需求,随着个人消费对经济贡献的不断增大,消费需求将成为物流行业发展的主要推动力。以手机终端消费者为对象,个性化、多样化的物流体验将成为电子商务条件下消费者的核心需求,这其中首当其冲的就是快递服务。快递是指快递公司通过铁路、公路和空运等交通工具,对客户进行快速投递的物流活动,属于门对门服务,根据时间快慢和递送质量区别于其他运输方式。快递的高效运转是建立在完善的网络基础上的,且网络具有相当强的整合能力。现阶段看来开发一种基于 web 端的物流管理系统,是适应信息化发展的必然趋势。

本物流管理系统就是针对本阶段中小物流企业的实际需求而去开发的一款基于 web 端的物流管理系统。本系统能够帮助企业实现对物流过程的调度优化和动态监控,并实现对物流的管控,来提高物流速度,提供高效,实用的物流管理系统。

## 1.2 研究意义

随着现代企业信息化建设的提高,办公自动化,管理现代化和决策科学化,是现代公司办公系统的标志。并随着信息化的不断发展,计算机的使用早已深入人类社会的各个方面。随着人们对数据库的使用和依赖,更大大促使了信息技术的发展,在物流管理领域,开发基于 web 的物流管理系统,一方面可以提高办公效率,另一方面可以实现高效的物流管理,这些是物流管理进行现代化,标准化建设的必然趋势。本物流管理系统是一套基于 JAVA 后台技术和数据库技术的现代企业信息系统平台,现阶段,它并不能简单的认为是一套管理软件,更准确的说是提供了强大的应用支撑。在提高管理人员对信息处理的准确性,发挥系统的基本功能的前提下,并逐步提高工作效率,节省了人力成本。本物流管理系统是以 JAVA EE 为开发平台,java 为开发语言,以数据库 oracle 作为系统的后台操作。本系统采用三层模式开发,系统功能全部实现在后台完成,实现前台展示,用户可以通过文档等来实现对系统的操作。

## 1.3 国内外研究现状及发展趋势

在现有的研究中,物流指数已经成为社会发展的重要的数据标准,所谓物流指数就是指

根据各个地区经济发展状况、物流发展基础条件以及物流发展对环境的影响的系统性评价指标,是人们对物流行业发展进行综合诊断和物流行业管理的必要手段,是综合衡量一个地区物流业发展程度的重要指标。物流指数的研究发布有利于进一步加强与世界发达国家物流的交流、比较和研究;有利于推进物流行业的科学指导和规范管理;对政府部门宏观决策和物流相关企业微观经营都具有极大的帮助作用。

现今阶段,根据数据显示物流业归属于第三产业,然而物流服务业在其中有着很大的比重,根据数据,物流服务业占 2011 年第三产业的比重为 17.6%,是所有服务行业占比最大的一类。随着中国加入世界贸易组织,并在协议中承诺全面开放第三产业的承诺,物流服务作为第三产业中最重要的一部分,肯定会受到世界巨头(UPS,速贷,联邦快递)冲击,对我国现有的物流企业形成很大的竞争。如今,物流行业从当时的手工下单、手工核查数据的方式,转变成现如今的物流管理平台的信息化方式,进而大量的节省的人力支出,使得配送管理流程实现自动化、一体化。进入 2010 年后,随着互联网的全球化发展,出现了一种智能运输管理系统,它即是物流系统的一种,也标志这我国未来研究的方向。它的主要功能是采用信息处理、图形处理、通信、控制、电子、互联网的现今技术,使人力、车辆、道路更加协调地结合在一起,用来提高交通运输种的效率问题以及生产效率的综合管理系统。我国从 70 年代开始对电子信息技术在道路交通和轨道交通领域的发展以及应用,尽力了相应的电子信息技术、互联情报、交通工程、自动化控制的研究机构。尽管如此,由于研究对象的分散以及研究水平受社会环境所限,形成多数研究项目是针对交通运输的某一局部问题而进得的,缺乏一个综合性的、具有战略意义的研究项目恰恰是覆盖这些领域的一项综合性技术,也就是说可以通过智能运输系统将原来这些互不相干的项目有机的联系在一起,使公路交通系统的规划、建设、管理、运营等各方面工作在更高的层次上协调发展,使公路交通发挥出更大的效。

所以随着信息技术的发展,以及互联网全球化的趋势,未来物流行业发展将是朝着信息化、网络化、全球化发展。并且信息技术将是最主要的核心竞争力,信息技术、网络技术日益广泛用于物流领域,物流与电子商务日益融合。而互联网的出现则促使物流行业发生了革命性的变化,基于互联网的及时准确的信息传递满足了物流系统高度集约化管理的信息需求,保证了物流网络各点和总部之间以及各网点之间信息的充分共享。

## 第二章 需求分析

### 2.1 编写目的

为了明确所用用户的基本需求，更好的完善和改进对用户的了解，使系统更加量化个明晰开发所需的工作量和工作进度，特此编写需求说明书。本物流管理系统的主要读者是：

- 客户
- 需求分析人员
- 软件开发人员
- 软件测试人员
- 项目负责人

### 2.2 系统说明

托付物流管理系统主要为中小物流公司的实际需求而开发的一套基于 web 端的物流管理系统。系统的研发和部署能够实现企业对物流过程的优化调度和动态控制，快速的整合企业的物流业务，本物流管理系统在协助工作人员对日常物流管理和人员管理。通过本物流管理系统，物流公司可以实现对车辆、订单、货物、订单线路的动态管理：管理人员能更快地实现对订单的修改和添加；车辆管理人员可以对已有车辆进行查看和调度；财务人员可以通过本系统实现对运输费用的计算。

### 2.3 开发技术与开发工具介绍

#### 2.3.1 JAVA 开发语言

Java 是一门面向对象编程语言，吸收了 C++语言的各种所长，还丢弃掉了 C++里难以理解的多继承、指针等概念，所以 Java 语言具有功能强大和简单易用两个特征<sup>[1]</sup>。Java 语言作为静态面向对象编程语言的代表，极好地体现了面向对象理论的实用性，允许程序员以简便而优雅的思考方式进行复杂的编程过程。Java 具有简单性、面向对象、分布式、安全性、平台独立与可移植性、多线程、动态性等特点。Java 不仅可以编写桌面应用程序、Web 应用程序，还可以分布式系统和嵌入式系统应用程序等。

Java 语言的面向对象，相对于程序员来说，意味着要注意 java 开发中的数据和操纵数据的方法与方式，并不是用严格的过程来思考。在一个面向对象的系统中，数据类是数据和操作数据的方法集合，数据和方法一起描述对象的状态和行为<sup>[2]</sup>。



## 2.3.2 JSP 技术

JSP 是由 SUN 公司制定的一种服务器端动态页面生成技术。它使用 JSP 标签在 HTML 网页中插入 Java 代码。标签通常以<%开头以%>结束。Java Servlet 是 JSP 的技术基础，一般大型的 WEB 应用程序都需要 Servlet 与 JSP 两者的配合才能完成。JSP 包括两大部分：一是 html 内容，其中包括 html、css 样式、javascript 脚本，二是包括少量的 java 代码。JSP 通过网页表单获取用户输入数据、访问数据库以及其他数据来源、然后动态的创建网页。JSP 标签有很多种功能，比如访问数据库、记录用户选择信息、访问 JavaBeans 组件等，还可以在不同的网页中传递控制信息和共享信息。

### 2.3.2.1 JSP 技术优点

现阶段 JSP 技术的成熟使得它成为和 Java 一样不可不学的一种开发技术，则它的优点有以下几点<sup>[3]</sup>：

- JSP 代码跨平台：即一次编写，处处运行，众所周知，由于微软的垄断性，它的产品可移植性做的十分差，ASP 也不例外。
- JSP 组件跨平台：JSP 组件都是跨平台可重用的，企业 JavaBeans 组件可以访问传统的数据库，并能以分布式系统模式工作与 Solaris，UNIX 和 Windows 平台。
- 和 SSI 相比：SSI 无法使用表单数据、无法进行数据库连接。
- 健壮性与安全性：由于 JSP 页面使用的脚本语言和 Java 语言，因此，它就具有 Java 技术的所有好处，包括健壮的存储管理和安全性。

### 2.3.3 Oracle 数据库技术

Oracle Database，又名 Oracle RDBMS，简称 Oracle。一种关系数据库管理系统。他是在数据库领域一直处于领先地位的产品。Oracle 数据库系统是美国 Oracle（甲骨文）公司提供的以分布式数据库为核心的一组软件产品，是目前最流行的客户/服务器（Client/Server，C/S）或浏览器/服务器（Browser/Server，B/S）体系结构的数据库之一<sup>[4]</sup>。Oracle 数据库是目前世界上使用最为广泛的数据库管理系统，作为一个通用的数据库系统，它具有完整的数据管理功能；作为一个关系数据库，它是一个完备关系的产品；作为分布式数据库它实现了分布式处理功能。但它的所有知识，只要在一种机型上学习了 ORACLE 知识，便能在各种类型的机器上使用它。

本管理系统使用 Spring 技术连接数据库资源池，Spring 是轻型的开源对象关系映射框架，它对 JDBC 进行了轻量级的对象封装。spring 中的 JDBC 框架则可以为省去连接和关闭数据库的代码，我们只要关注我们想对数据库进行的操作即可。

连接数据库对根据不同数据库又不同的连接方式，本系统使用的数据库是 Oracle 数据库。则在 Spring 中配置 JDBC 如下图 3.1 所示：

```
<!-- 数据库连接池
    池子里放了若干个数据库连接, 有用户需要从池里面取, 用完了就放回来
-->
<!-- 数据源 -->
<bean id="dataSource"
class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver" />
    <property name="url" value="jdbc:oracle:thin:@127.0.0.1:1521:xe" />
    <property name="username" value="tao" />
    <property name="password" value="123" />
</bean>
```

图 3.1 Spring JDBC 连接配置

### 2.3.3.1 Oracle 数据库优点

Oracle 数据库由 oracle 公司开发, 面向互联网计算的, 支持关系对象型的, 分布式的数据库产品, 是一个高度集的互联网应用平台, 为企业数据存储提交高性能的数据管理系统。

Oracle 数据库的主要特点:

- 支持多用户、大事务量的事务处理
- 数据安全性和完整性控制
- 支持分布式数据处理
- 可移植性
- Oracle 数据库基于客户端/服务器技术

### 2.3.4 MyEclipse 简介

MyEclipse 是 j2ee 中 Eclipse 的插件集合, 是 j2ee 的集成开发环境, 包括了完备的编码、调试、测试和发布功能, 支持 HTML、CSS、Javascript、JSP、Spring、Struts、Hibernate 等。它具备了强大的汇编功能和广泛的插件支持。

### 2.3.5 Tomcat 简介

自从 JSP 发布之后, 推出了各式各样的 JSP 引擎。Apache Group 在完成 GNUJSP1.0 的开发以后, 开始考虑在 SUN 的 JSWDK 基础上开发一个可以直接提供 Web 服务的 JSP 服务器, 当然同时也支持 Servlet, 这样 Tomcat 就诞生了。Tomcat 是 jakarta 项目中的一个重要的子项目, 其被 JavaWorld 杂志的编辑选为 2001 年度最具创新的 Java 产品, 同时它又是 sun 公司官方推荐的 servlet 和 jsp 容器, 因此其越来越多的受到软件公司和开发人员的喜爱。servlet 和 jsp 的最新规范都可以在 tomcat 的新版本中得到实现。其次, Tomcat 是完全免费的软件, 任何人都可以从互联网上自由地下载。Tomcat 与 Apache 的组合相当完美。

## 2.3.6 JavaScript 与前端技术

### 2.3.6.1 JavaScript

JavaScript 是一种由浏览器解释并执行的脚本语言，它的主要作用是：

- 对数据进行基本验证,即对交付送给数据库的数据合法性进行判断,比如表单信息,登录注册验证等。
- 对网页进行操作,网页动画控制。
- 获取浏览器的基本信息,获取浏览器类型。
- Ajax 技术是 JavaScript 的核心。

### 2.3.6.2 JQuery

jQuery 是一个开源的、免费的、兼容多浏览器的 javascript 库, jQuery 能够使用户的 html 页面保持代码和 html 内容分离,使用户能更方便地处理 HTML、Events、实现动画效果,并且方便地为网站提供 Ajax 交互。

jQuery 核心特点如下:

- 具有独特的链式语法和简洁清晰的多功能接口。
- 丰富灵活的 css 选择器, 并可以进行选择器的扩张。
- 具有便捷的插件扩展机制和丰富的插件库。
- 可以兼容多种引擎的浏览器。

### 2.3.6.3 Bootstrap 框架

Bootstrap 是一种基于 HTML、CSS、Javascript 的前端框架,它简洁灵活,提供了优雅的 HTML 和 CSS 规范,兼容大部分 jQuery 插件,并包含了丰富的 Web 组件和 jQuery 插件,使开发者可以快速的搭建一个漂亮、功能完备的网站。

## 2.4 系统开发框架

在通常的 Java Web 开发中通常会用到 MVC 开发模式,MVC 全名是 Model View Controller,是模型(model)－视图(view)－控制器(controller)的缩写,一种软件设计典范,用一种业务逻辑、数据、界面显示分离的方法组织代码,将业务逻辑聚集到一个部件里面,在改进和个性化定制界面及用户交互的同时,不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

### 2.4.1 Struts2 框架

Struts2 是 Apache 软件基金下 Jakarta 项目的一部分, Struts2 是一个基于 MVC 设计模式的 Web 应用框架<sup>[5]</sup>, Struts2 是由传统的 Struts1、WebWork 两个经典的 MVC 框架发展起来, 无论是从 Struts2 设计的角度来看, 还是从 Struts2 的实际项目中的易用性来看, Struts2 都是一个非常优秀的 MVC 框架。与传统的 Struts1 相比, Struts2 允许使用普普通通的、传统的 Java 对象作为 Action; Action 的 execute() 方法不再与 ServletAPI 耦合, 因此更容易测试; 支持更多的师徒技术; 基于 AOP 思想的拦截器机制, 提供了极好的可扩展性; 更强大、更易用的输入验证; 整合 Ajax 支持。这些都是 Struts2 的巨大吸引力。

Struts2 是 Java 开发中使用技术最为重要的框架之一, 通常情况下, Struts2 的工作步骤为:

- Struts2 拦截器可以根据配置文件 Struts.xml 来确定所需调用的 Action
- 服务器根据请求来创建 ValueStack 栈对象, 并进行把 Action 放到栈顶。
- 服务器调用 Action 和相应的拦截器。
  - A. 调用 action 之前, 拦截器可对 ValueStack 中的 action 的数据进行操作。
  - B. 调用 action: 根据输入计算输出。
  - C. result 把 ValueStack 中的数据以某种形式向用户输出。
  - D. 拦截器做收尾工作。

Struts2 的工作原理如图 2.1 所示:

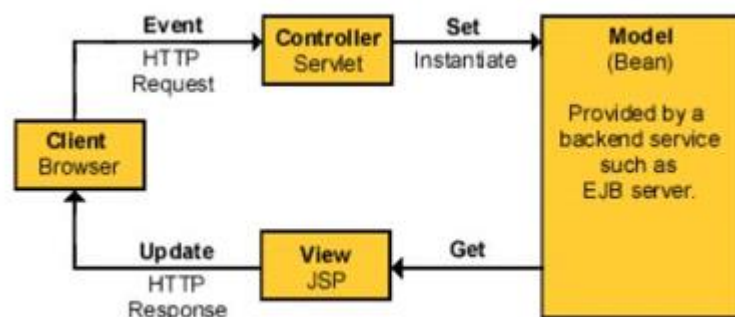


图 2.1 Struts2 原理图

### 2.4.2 Spring 框架

Spring 是一个轻量级控制反转 (IOC) 和面向切面 (AOP) 的开源容器框架<sup>[6]</sup>, Spring 是一个开源框架, Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架, 由 Rod Johnson 创建。简单来说, Spring 是一个分层的 JavaSE/EEfull-stack(一站式) 轻量级开源框架。

## 2.5 功能需求

本系统的主要功能需求有：用户管理模块，配送点管理模块，订单管理模块，线路管理模块，车辆管理模块，货物管理模块。

1. 用户管理模块需求：用户分为管理员用户和普通用户，管理员用户可以实现对订单的增加，对其他用户进行增加，删除，更改。
2. 配送点管理模块需求：配送点管理是由公司完成的，有配送点的浏览、添加、修改、删除。
3. 订单管理模块需求：订单管理是包括下订单、订单的修改、删除。每一个配送点的管理人员可以查询配送点的当前订单情况。查询的订单可以包括：今日提交订单，历史订单，未反馈订单等。
4. 线路管理模块需求：公司可以根据市场反应情况添加路线。系统选择添加路线，输入增加路线的名称，然后添加路线经过的配送点，并可以通过添加、移除、上移、下移等操作调整线路顺序。
5. 车辆管理模块需求：车辆管理是由公司负责的，公司负责购买所需车辆，以及车辆的维护；在该系统中我们只需要把车辆的增、删、改做完即可。
6. 货物管理模块需求：管理货物种类。

## 第三章 系统分析

在经济全球化和信息共享的时代，企业物流管理系统的信息技术也在不断的提升，从而使物流系统的开发要求也越来越严格。达到同样的目标可以采用很多不同的方式，通过对系统需求的分析，本物流管理系统选用 MyEclipse+Oracle 开发技术，前端页面使用 css+JavaScript+Jquery 实现理想的要求，我们在不同的方案要求中会提出不同的分析。

### 3.1 可行性分析

系统可行性分析的目的是用最小的代价在尽可能短的时间内确定问题是否能够解决，有效地避免对系统的额外投资浪费。托付物流管理系统的可行性主要体现在以下几个方面：

（1）技术可行性：本系统将采用的开发技术包括 HTML、CSS、Javascript、Java、Spring、Mybatis、Bootstrap，JSP 等，在这些技术都是经过实践的检验已成熟稳定，可以查阅大量的技术资料，并且使用的 Oracle 数据管理系统和 MyEclipse 开发工具被广泛运用，使得技术上是完全可行的。

（2）经济可行性：本系统的开发环境均使用现阶段开发人员常用的免费开源软件，并使用可视化编程工具以及高效的数据库管理系统，轻量级的后端 Spring 开发框架和流行的 Bootstrap 前端框架的使用也快速的完成系统的研发周期，使本系统在成本控制方面做到了完美。

（3）操作可行性：系统使用菜单式，实现用户与 Oracle 数据库之间的交互，Bootstrap 前端框架搭建的简洁友好的界面设计，使操作便捷，用户只需要掌握最基本的计算机操作规范和系统流程即可，不需要掌握其它技术知识，，所以系统具有操作可行性。

（4）社会可行性：本系统的研发可以简化并提高物流系统工作开展的流程，促进基层员工和企业管理之间的高效融合，并帮助基层员工减少大量工作量，有效提高其工作效率，因此，系统具有很强的社会可行性。

### 3.2 系统基本功能介绍

本物流管理系统主要有以下功能：

- 1) 实现用户的注册登录，对用户的增加、更改、删除操作。
- 2) 实现用户对信息的查询。
- 3) 实现用户对货物信息，订单信息，车辆信息，线路信息和配送点信息的管理的基本操作，如增加、删除、更改、查询。

### 3.3 数据库物理设计

对于一个信息系统，数据库建设和设计是最为重要的一环，数据库结构对系统效率影响非常大，数据库结构设计如果合理常规，将能非常大的提高数据存储的效率。本系统数据库中总共有 7 张表：user、station、goodstype、order、way、way\_detail 和 vehicle

**表 3-1 用户表 user**

| 列名           | 类型      | 字段大小 | 主键否 | 描述     |
|--------------|---------|------|-----|--------|
| id           | int     | 11   | 是   | 用户 id  |
| BELONGTO_IDR | int     | 11   | 否   | 配送点 id |
| ROLE_ID      | int     | 11   | 否   | 角色 id  |
| NAME         | varchar | 50   | 否   | 名字     |
| MOBILEPHONE  | varchar | 20   | 否   | 联系电话   |
| ADDRESS      | varchar | 200  | 否   | 住址     |
| USERNAME     | varchar | 50   | 否   | 用户名    |
| PASSWORD     | varchar | 20   | 否   | 密码     |

**表 3-2 配送点表 station**

| 列名             | 类型      | 字段大小 | 主键否 | 描述     |
|----------------|---------|------|-----|--------|
| ID             | int     | 11   | 是   | 配送点 id |
| STATIONNAME    | varchar | 50   | 否   | 配送点名称  |
| STATIONPHONE   | varchar | 20   | 否   | 固定电话   |
| STATIONADDRESS | varchar | 200  | 否   | 地址     |

**表 3-3 货物表 goodstype**

| 列名            | 类型      | 字段大小 | 主键否 | 描述      |
|---------------|---------|------|-----|---------|
| ID            | int     | 11   | 是   | 货物种类 id |
| GOODSTYPENAME | varchar | 50   | 否   | 货物种类名称  |
| REMARK        | varchar | 50   | 否   | 备注      |

**表 3-4 订单表 order**

| 列名              | 类型      | 字段大小 | 主键否 | 描述       |
|-----------------|---------|------|-----|----------|
| ID              | int     | 11   | 是   | 订单号      |
| OUTSETSTATIONID | int     | 50   | 否   | 始发配送点 id |
| DESTSTATIONID   | int     | 50   | 否   | 目的配送点 id |
| ORDERSTATUS     | varchar | 200  | 否   | 订单状态     |
| VEHICLEID       | int     | 11   | 否   | 承运车辆 id  |

续表 3-4

| 列名                   | 类型      | 字段大小 | 主键否 | 描述      |
|----------------------|---------|------|-----|---------|
| RECEIVEDATA          | date    | 50   | 否   | 收寄时间    |
| GOODSNAME            | varchar | 11   | 否   | 货物名称    |
| GOODSNUMBER          | int     | 11   | 否   | 货物数量    |
| GOODSTYPEID          | int     | 11   | 否   | 货物种类 id |
| GOODSTOTALWEIGHT     | int     | 11   | 否   | 货物总重量   |
| GOODSTOTALVOLUME     | int     | 11   | 否   | 货量总体积   |
| TOTALFARE            | int     | 50   | 否   | 费用      |
| SENDERNAME           | varchar | 20   | 否   | 寄件人名字   |
| SENDERMOBILEPHONE    | varchar | 200  | 否   | 寄件人电话   |
| SENDERADDRESS        | varchar | 20   | 否   | 寄件人地址   |
| SENDERZIPCODE        | varchar | 50   | 否   | 寄件人邮编   |
| ADDRESSEENAME        | varchar | 20   | 否   | 收件人名字   |
| ADDRESSEEMOBILEPHONE | varchar | 20   | 否   | 收件人电话   |
| ADDRESSEEADDRESS     | varchar | 50   | 否   | 收件人地址   |
| ADDRESSEEZIPCODE     | varchar | 20   | 否   | 收件人邮编   |
| REMARK               | varchar | 200  | 否   | 备注      |
| WAY_ID               | varchar | 20   | 否   | 线路 ID   |

表 3-5 线路表 way

| 列名            | 类型      | 字段大小 | 主键否 | 描述    |
|---------------|---------|------|-----|-------|
| ID            | int     | 11   | 否   | 线路 id |
| START_STATION | varchar | 50   | 否   | 起始点   |
| END_STATION   | varchar | 20   | 否   | 目的点   |
| NAME          | varchar | 200  | 否   | 线路名称  |

表 3-6 线路明细表 way\_detail

| 列名         | 类型  | 字段大小 | 主键否 | 描述       |
|------------|-----|------|-----|----------|
| ID         | int | 11   | 否   | 主键 id    |
| WAY_ID     | int | 11   | 否   | 线路 id    |
| STATION_ID | int | 11   | 否   | 途经配送点 id |
| SORT_ID    | int | 11   | 否   | 顺序       |

表 3-7 车辆管理表 vehicle

| 列名              | 类型      | 字段大小 | 主键否 | 描述    |
|-----------------|---------|------|-----|-------|
| ID              | int     | 11   | 是   | 车辆 id |
| VEHICLELICENSE  | varchar | 50   | 否   | 车牌    |
| VEHICLELOAD     | int     | 11   | 否   | 载重    |
| VEHICLECAPACITY | int     | 11   | 否   | 容量    |



## 第四章 系统总体设计

总体设计顾名思义就是对系统的各个模块进行设计和规划,又称为概要设计或初步设计。通过这个阶段的工作将划分出组成的系统的物理元素,如:程序、文件、数据库、人工过程、文档等等,但每个物理元素仍然处于测试阶段的黑盒阶段。总体设计阶段的重要任务就是<sup>[7]</sup>:

- A. 设计软件或系统的结构分布并分析其具体功能
- B. 确定系统中每个程序是由那些模块组成和实现
- C. 确定各个模块之间的关联性与依赖性

### 4.1 系统功能结构

通过对本物流管理系统的结构分析和结构梳理,我们确定了本系统的功能模块图,系统分为管理员权限,用户权限。下面是系统总的功能模块图:

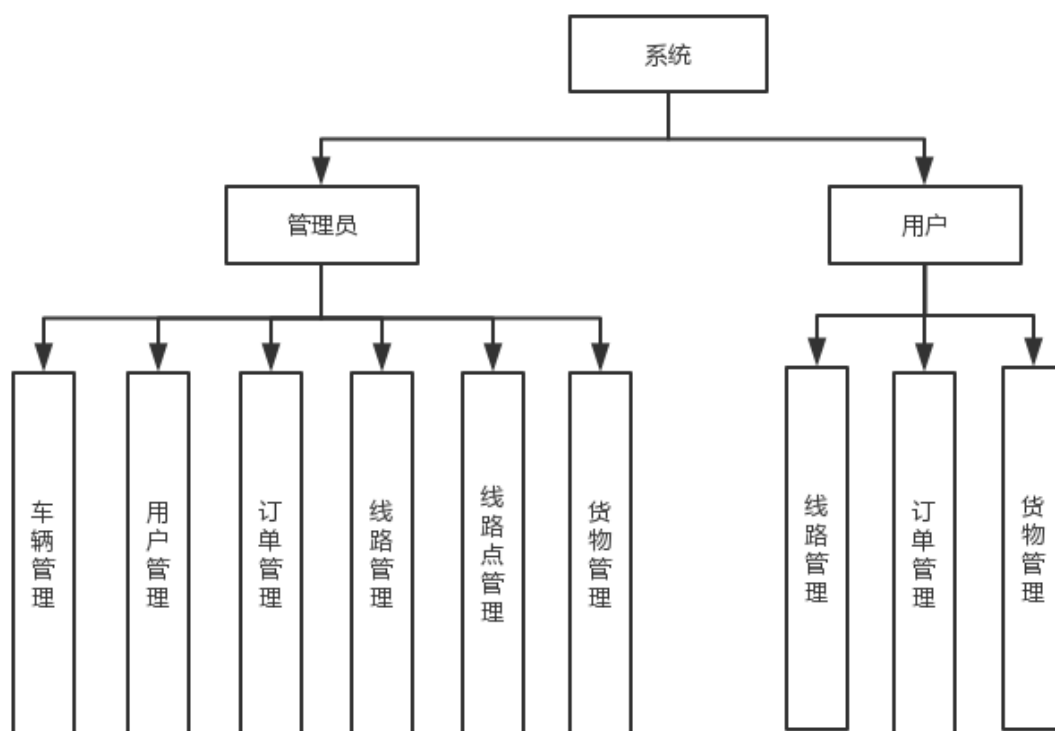


图 4.1 系统功能图

管理员主要功能: 对用户信息进行管理, 对用户进行增、删、改、查。查看车辆信息、订单信息、线路点信息、和货物信息的管理。管理员功能图如图 4.2 所示:

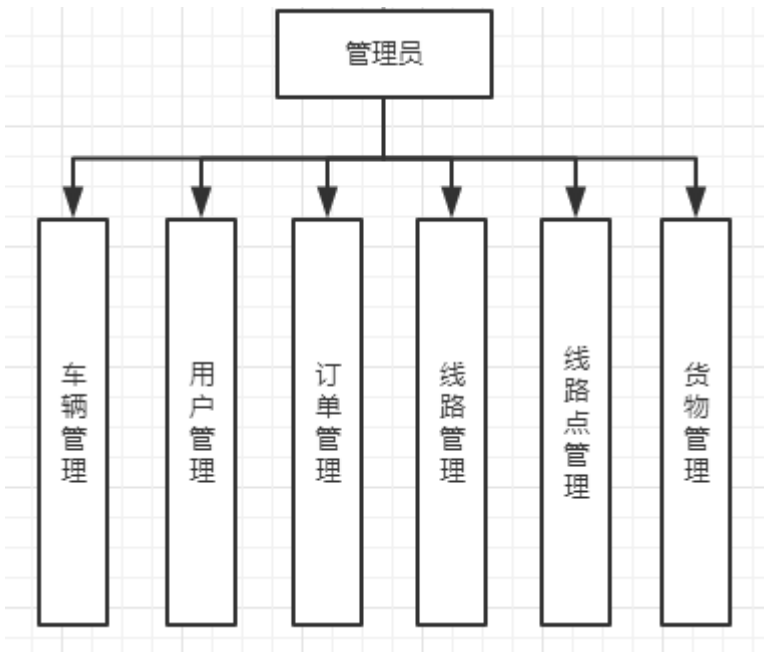


图 4.2 管理员功能模块

企业基层开票员主要有对订单信息的管理，如：开票、修改、删除。对线路信息的管理，对货物信息的管理。企业基层开票员的功能图如图 4.3 所示：

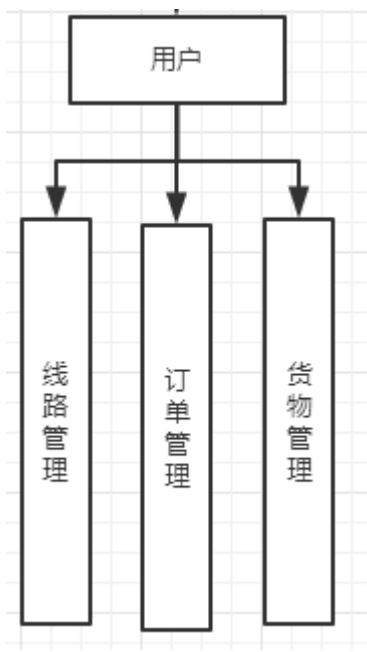


图 4.3 企业基层员工功能表

4.2 数据表关系设计

对于数据库设计遵循数据库设计的三大范式，每张表之间的关联关系的正确性是数据库

效率的关键,因此,数据表关系设计也成为数据库设计的重要环节,本系统中主要有七张表,如何去管理和关联这七张表。使其能更高效的使用,对数据的存储和获取是非常的重要,,数据库建设和设计是最为重要的一环,数据库结构对系统效率影响非常大,数据库结构设计如果合理常规,将能非常大的提高数据存储的效率。

4.2.1 Power Designer 软件使用

本系统的数据库各个数据表之间的关联性问题,主要使用 Power Designer 进行数据库关联性的设计,在数据库建模的过程中,运用 Power Designer 进行数据库设计,主要原因是 Power Designer 不但可以让人直观的理解模型,而且可以充分的利用数据库技术,优化数据库的设计。使用其来建立数据库连接,方便操作数据库。

使用基本的操作进行设计的数据库表关联图如图 4.4 所示:

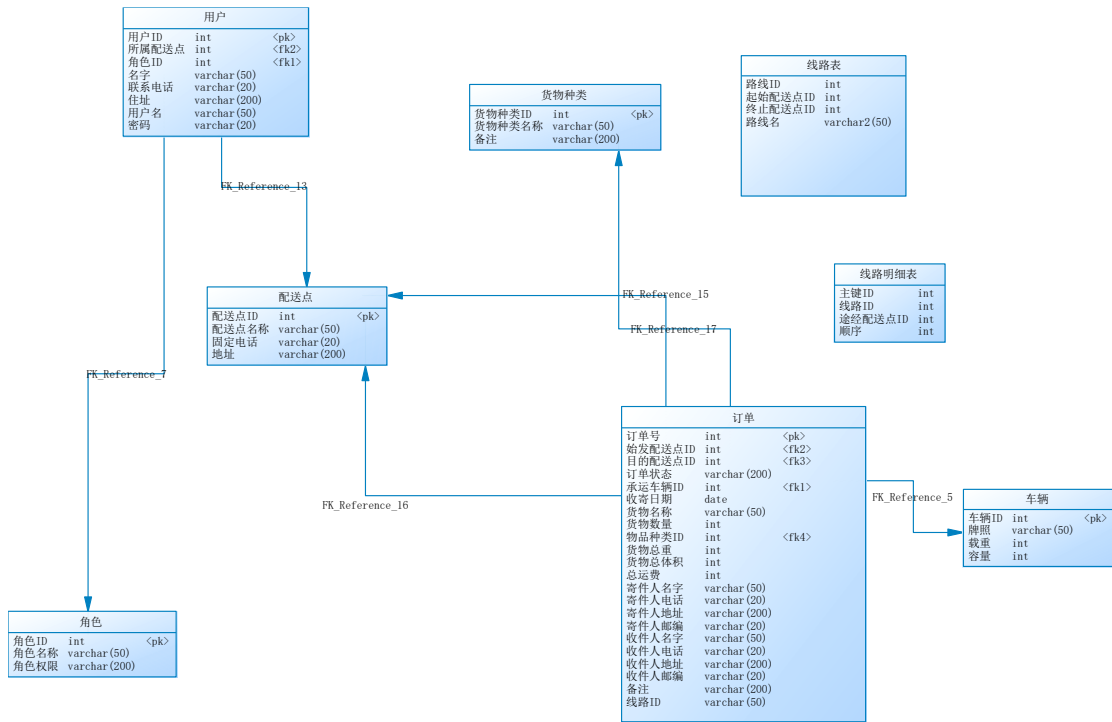


图 4.4 数据库表结构

4.3 系统概念模型设计

在信息系统设计中,我们通常需要用图表来表现各个实体之间的关系,E-R图是最常用的方法,它用来实现模型设计。E-R图可以用来表示数据库中各个成员组织情况,一般E-R图中主要包含(实体、联系和属性)。主要特点有简单易懂,方便掌握,直观表达。能够准确的反应现实世界的信息关联。

a) 企业基层操作员 E-R 图

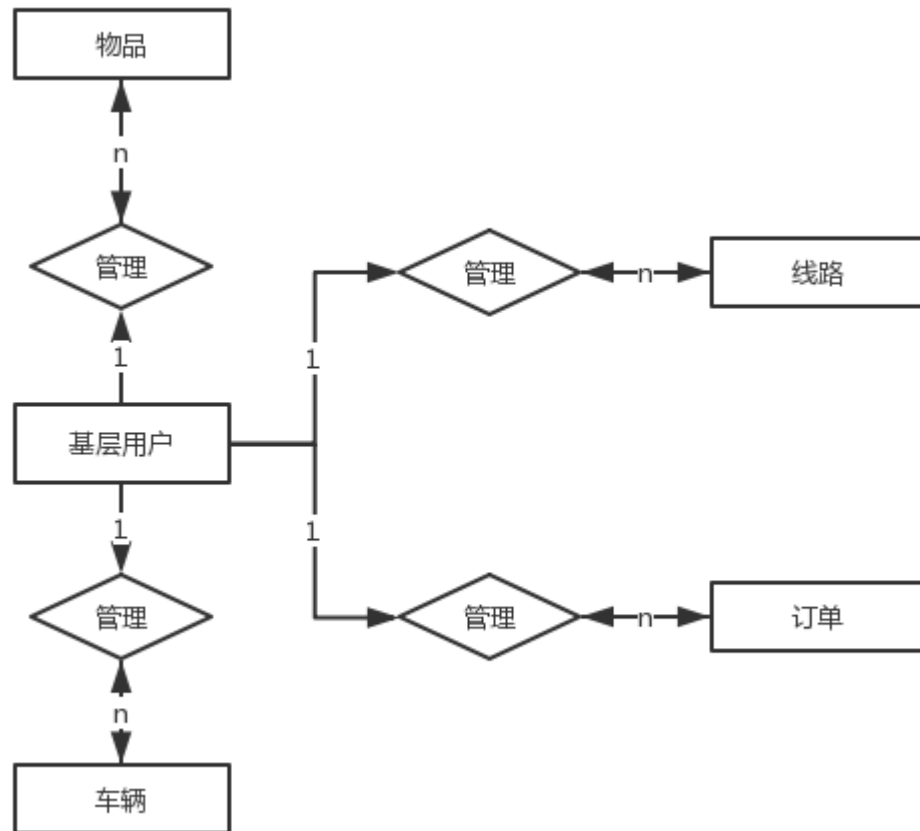


图 4.5 员工 E-R 图

## 第五章 系统模块设计

### 5.1 登录注册模块

这个模块主要分为两个核心点：（1）登录和注册使用前端流行的 css+JavaScript+jquery 插件写总体框架，（2）数据利用 strust2 进行传输实现和后台数据库的联系。

#### 5.1.1 登录界面与核心代码

登录界面如 5.1 图所示：



图 5.1 登录界面

登录核心代码：

一、前台代码

前台代码中主要有 jquery 控制代码，用于检测输入数据是否为数据库数据，如下：

```

$(".btn").click(function(){
    is_hide();
})
var u = $("#username");
var p = $("#password");
$("#submit").live('click',function(){
    if(u.val() == " || p.val() == "")
    {
        $("#ts").html("用户名或密码不能为空~");
        is_show();
        return false;
    }
    else{
        var reg = /^[0-9A-Za-z]+$/;
        if(!reg.exec(u.val()))
        {
            $("#ts").html("用户名错误");
            is_show();
            return false;
        }
    }
});
window.onload = function()
{
    $(".connect p").eq(0).animate({"left":"0%"}, 600);
    $(".connect p").eq(1).animate({"left":"0%"}, 400);
}
function is_hide(){ $(".alert").animate({"top":"-40%"}, 300) }
function is_show(){ $(".alert").show().animate({"top":"45%"}, 300) }

```

图 5.2 登录判断代码

前台 jsp 代码，主要作用搭建主题页面结构如下：

```

<div class="page-container">
    <h1>托付物流管理系统</h1>
    <form method="post" action="loginAction!login">
        <div>
            <input type="text" name="userInfo.username"
class="username" placeholder="Username" autocomplete="off"/>

```

```

        </div>
        <div>
            <input type="password" name="userInfo.password"
class="password" placeholder="Password" oncontextmenu="return false"
onpaste="return false" />
        </div>
        <!-- <button id="submit" type="button" name="submit">Sign
in</button> -->

        <div>
            <input type="submit" name="submit" value="登录系统"
id="submit"/>
        </div>
        <div class="type">
            <a href="<%=request.getContextPath() %>/JSP/User/register.jsp"
class="register">没有账号请注册</a>
        </div>
    </form>
    <div class="connect">
        <p>CERTAINTY IS OUR FOCUS</p>
        <p style="margin-top:20px;">专注你的托付</p>
    </div>

</div>
<div class="alert" style="display:none">
    <h2>消息</h2>
    <div class="alert_con">
        <p id="ts"></p>
        <p style="line-height:70px"><a class="btn">确定</a></p>
    </div>
</div>

```

图 5.3 登录 jsp 代码

## 二、后台 ACTION 代码

后代 ACTION 代码主要的作用是控制前后台数据的交换，你数据库和前台页面的信息更换。主要核心代码如下：

```

public String login(){

    //根据前台数据进行 request 请求
    HttpServletRequest req = ServletActionContext.getRequest();
    //UserInfoDao ud = new UserInfoDao();
    if(userInfoService.login(userInfo)){//判断输入信息是否正确

```

```

        HttpSession session = req.getSession();
        session.setAttribute("user", userInfo);
        return "main";
    }
    this.addActionMessage("登陆成功");
    return "login";
}

/**
 * 退出登录请求跳转
 * @return
 */
public String logOut(){
    HttpServletRequest req = ServletActionContext.getRequest();

    req.getSession().removeAttribute("user");
    return "login";
}

```

图 5.4 登录后台 ACTION

### 5.1.2 注册界面与核心代码

注册界面如 5.5 图所示：

请注册托付物流管理系统

姓名  请输入姓名

手机号码  仅用于发送服务开通与到期提醒以及紧急故障方便联系到您，绝对保密

地址： 请输入您的常用收货地址

用户名： 请输入用户名

设置密码  请使用6~20个英文字母（区分大小写）、符号或数字

[注册](#)

图 5.5 注册界面

注册界面核心代码：

(一) 前台核心代码，分为正则表达式的判断和页面结构

利用 JQuery 正则表达式来判断输入的数据是否符合数据库中数据类型。



```
function __changeUserName(of){
    var username=$('#'+of).val();
    if(of=='email'){
        if (username.search(/^[w\.,+-]+@[w\.,+-]+$/i) == -1) {
            //showToolTips('email_input','请输入正确的 Email 地址');
            return;
        }
    }else{
        if(username==" || !isMobilePhone(username)) {
            //showToolTips('mobile_input','请输入正确的手机号码');
            return;
        }
    }
}

function checkPwd1(pwd1) {
    if (pwd1.search(/^[6,20]$/i) == -1) {
        showToolTips('password1_input','密码为空或位数太少');
    }else {
        hideToolTips('password1_input');
    }
}

function check() {
    hideAllToolTips();
    var ckh_result = true;
    if ($('#email').val() == "") {
        showToolTips('email_input','邮箱不能为空');
        ckh_result = false;
    }
    if ($('#password1').val() == "") {
        showToolTips('password1_input','密码不能为空');
        ckh_result = false;
    }
    return ckh_result;
}

function checkMobilePhone(telephone) {
    if(telephone==" || !isMobilePhone(telephone)) {
        showToolTips('mobile_input','请输入正确的手机号码');
    }else{
        hideToolTips('mobile_input');
    }
}
```

图 5.6 正则表达式判断数据

(二) 后台核心代码  
后台中主要为添加数据如下所示：

```
public void setUserInfo(UserInfo userInfo) {
    this.userInfo = userInfo;
}
public String register(){
    userInfoService.add(userInfo);
    return list();
}
```

图 5.7 注册代码

5.2 主界面模块

主界面如图 5.8 所示：



图 5.8 主题界面

核心代码如下：

```
<!-- 顶部 -->
<div id="middle">
    <div class="left">
        <div id="my_menu" class="sdmenu">
            <div>
                <span>用户管理</span>
                <a href="loginAction!list">用户管理列表</a>
            </div>
            <div>
                <span>配送点管理</span>
                <a href="stationAction!list">配送点管理</a>
            </div>
            <div>
                <span>线路管理</span>
            </div>
        </div>
    </div>
</div>
```

```

    <a href="wayAction!list">线路管理</a>
</div>
<div>
    <span>订单管理</span>
    <!-- <a href="ordersAction!add">添加订单</a> -->
    <a href="ordersAction!list">订单列表</a>
    <!-- <a href="ordersAction!selOne">修改页面</a> -->
</div>
<div>
    <span>货物种类管理</span>
    <a href="goodsAction!list">货物种类管理</a>
</div>
<div>
    <span>车辆管理</span>
    <a href="vehicleAction!list">车辆管理</a>
</div>
</div>
<div>
    <div class="Switch"></div>
    <script type="text/javascript">
/*折叠框收缩*/
$(document).ready(function(e) {
    $(".Switch").click(function(){
        $(".left").toggle();

    });
});
</script>
</body>

```

图 5.8 主题界面核心代码

### 5.3 订单管理模块

订单模块是本系统中最为重要的两个模块之一，它主要的功能是，进行订单的展示，查询，模糊查询、修改、删除、添加。因为它在数据库设计中关联性是最为复杂，与多个表进行关联。

- 订单主界面如图 5.9 所示：

当前位置: [首页](#) / 订单列表管理

订单管理

+ 添加订单

订单日期:  订单状态: --请选择--

查询

| 订单号 | 始发配送点 | 目的配送点 | 订单状态 | 收货日期                | 货物名称 | 物品种类 | 总费用  | 寄件人姓名 | 收件人地址 | 收件人名字 | 寄件人地址 | 备注      | 操作                                    |
|-----|-------|-------|------|---------------------|------|------|------|-------|-------|-------|-------|---------|---------------------------------------|
| 69  | 4     | 5     | 已发货  | 2017-04-22 16:01:59 | 锤子手机 | 9    | 927  | 罗永浩   | 北京东城区 | 马云    | 杭州西湖边 | 基佬的激情   | <a href="#">删除</a> <a href="#">编辑</a> |
| 70  | 2     | 4     | 已收货  | 2017-04-22 16:04:38 | 猪    | 9    | 90   | 丁磊    | 网易养猪场 | 马化腾   | 企鹅总部  | 听说养猪致富快 | <a href="#">删除</a> <a href="#">编辑</a> |
| 77  | 1     | 2     | 已发货  | 2017-04-24 09:35:38 | 猫    | 1    | 283  | 三里阿克曼 | 省道    | 阿尔法   | 三大    | 无服务范围   | <a href="#">删除</a> <a href="#">编辑</a> |
| 79  | 1     | 8     | 运输中  | 2017-04-24 15:52:29 | 3233 | 1    | 4502 | 二维    | 而突然个人 | 厄齐尔群  | 个人提供  | 日本人体一天天 | <a href="#">删除</a> <a href="#">编辑</a> |

共1页,总4条记录,第1页 [首页](#) [上一页](#) [下一页](#) [末页](#)

图 5.9 订单主体界面

- 编辑界面如图 5.10 所示:

修改订单页面

订单日期:

始发配送点: 上海

目的配送点: 郑州

订单号:

线路明细: 郑州-昆明

承运车辆: 川C98746

订单状态: 已发货

| 货物名称                              | 货物数量                            | 货物总重量                           | 货物总体积                           | 物品种类            | 运费  |
|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|-----------------|---|
| <input type="text" value="锤子手机"/> | <input type="text" value="33"/> | <input type="text" value="21"/> | <input type="text" value="44"/> | <span>手机</span> | <div><input type="text" value="927"/> <input type="button" value="计算运费"/></div> |

寄件人 名字:

收件人名字:

寄件人 电话:

收件人电话:

寄件人地址:

收件人地址:

寄件人邮箱:

收件人邮箱:

备注:

提交

返回

[Debug]

图 5.10 编辑订单界面

- 添加界面如图 5.11 所示:

添加订单

订单日期:

线路: --请选择--

始发配送点: --请选择--

订单号:

目的配送点: --请选择--

承运车辆: --请选择--

订单状态:

| 货物名称                 | 货物数量                 | 货物总重量                | 货物总体积                | 物品种类ID               | 运费  |
|----------------------|----------------------|----------------------|----------------------|----------------------|---|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <span>--请选择--</span> | <div><input type="text"/> <input type="button" value="计算运费"/></div> |

寄件人 名字:

收件人名字:

寄件人 电话:

收件人电话:

寄件人地址:

收件人地址:

寄件人邮箱:

收件人邮箱:

备注:

添加订单

图 5.11 添加界面

订单管理模块的核心代码的后台 ACTION 中如图 5.12 所示:

```
/*
 * 获取 orders 存储在 list 中
 */
public String list() {

    HttpServletRequest req = ServletActionContext.getRequest();
    List<Way> waylist = wayService.getWay();//建立 waylist 存储 way 数据
    req.setAttribute("waylist", waylist);

    List<Vehicle> vehiclelist = vehicleService.getVehicle();
    req.setAttribute("vehiclelist", vehiclelist);

    List<Station> stationlist = stationService.getStation();

    req.setAttribute("stationlist", stationlist);

    List<Goods> goodsTypelist = goodsService.getGoods();
    req.setAttribute("goodsTypelist", goodsTypelist);

    List<Orders> list = ordersService.getAllOrders();

    System.out.println("qqqqqqqqqq"+list);
    req.setAttribute("list", list);
    /*
     * 分页查询
     */
    List<Orders> lists = ordersService.selList(orders,this.getCurPage());
    this.init(lists.size());
    //System.out.println("eiojaojaojsdjao"+ orders.getId());
    req.setAttribute("list", lists);
    if(orders!=null){//判断是否获取到 orders
        req.setAttribute("o", orders);
        orders=new Orders();
    }
    return "list";
}

/*
 *
 * 添加
 */
```

```
public String add(){
    ordersService.addOrders(orders);
    orders=new Orders();
    return "add";
}
//
public String selOne() {

    HttpServletRequest req = ServletActionContext.getRequest();
    List<Orders> list = ordersService.getOrders(orders.getId());
    req.setAttribute("list", list.get(0));

    List<Way> waylist = wayService.getWay();
    req.setAttribute("waylist", waylist);

    List<Vehicle> vehiclelist = vehicleService.getVehicle();
    req.setAttribute("vehiclelist", vehiclelist);

    List<Station> stationlist = stationService.getStation();
    req.setAttribute("stationlist", stationlist);

    List<Goods> goodsTypelist = goodsService.getGoods();
    req.setAttribute("goodsTypelist", goodsTypelist);
    orders=new Orders();
    return "update";
}
// 更新
public String update() {
    System.out.println(orders.toString());
    ordersService.update(orders);
    System.out.println(orders.toString());
    orders=new Orders();
    return list();
}

// 删除
public String del() {

    ordersService.del(orders.getId());
    orders = new Orders();
    return list();
}
//根据 id 获取数据, 并以 json 方式传输到前台
```

```
public void selWays() throws IOException{

    HttpServletResponse res = ServletActionContext.getResponse();

    //获取 list id
    List<Way> list = wayService.getStationId(id);

    System.out.println(list);
    List<Map<String,Object>> listStationId = new ArrayList<Map<String,Object>>();

    //设置 Map

    Map<String,Object> map = new HashMap<String,Object>();
    for(Way way : list){
        System.out.println("this is a test!");
        map = new HashMap<String,Object>();
        //put 所需要的数据
        map.put("startId", way.getStartStation());
        map.put("endId", way.getEndStation());
        map.put("startStationName",way.getStartStationName().getStationName());
        map.put("endStationName",way.getEndStationName().getStationName());

        listStationId.add(map);
    }

    //生成 json 键值对

    JSONArray js = JSONArray.fromObject(listStationId);

    res.setContentType("text/html;charset=utf-8");
    PrintWriter pw = res.getWriter();
    pw.write(js.toString());
    pw.flush();
    pw.close();

}
```

图 5.12 订单表后台核心代码

## 5.4 线路管理模块

本系统中线路管理的主要功能为：公司可以根据市场反馈情况添加路线。来进行选择添加路线，输入增加路线的名称，然后添加路线经过的配送点，并实现线路点的添加、移除、

上移、下移等操作调整线路顺序。

- 线路管理主界面如图 5.13 所示：



图 5.13 线路管理主界面

- 详细线路界面如图 5.14 所示：

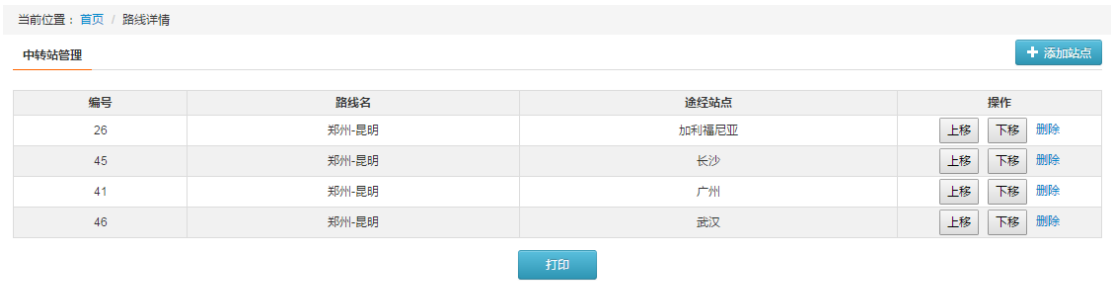


图 5.14 详细线路主界面

- 添加界面如图 5.15 所示：



图 5.15 添加线路界面

- 编辑界面如图 5.16 所示：



编辑路线

路线编号:

6

始发站:

加利福尼亚

终点站:

加利福尼亚

路线名称:

北京-广州

提交

取消

图 5.16 修改线路界面

- 添加站点界面如图 5.17 所示：

添加站点

站点名称:

北京

添加

取消

图 5.17 添加站点界面

核心代码如图 5.17 所示：

```
public String list(){

    List<Way_detail> list =way_detailService.getList(way.getId());
    HttpServletRequest req = ServletActionContext.getRequest();

    //设置 list

    req.setAttribute("list", list);
    req.setAttribute("wayid", way.getId());
    return "list";
}

public String add(){
    way_detailService.add(way_detail);
    way_detail = new Way_detail();
    return "add";
}

public String upWD(){

    Way_detail uWD=way_detailService.selUP(way_detail);
    System.out.println(uWD);
    String sid=uWD.getSortId();

    uWD.setSortId(way_detail.getSortId());

    way_detail.setSortId(sid);
    System.out.println(way_detail);
    System.out.println(uWD);
    way_detailService.update(way_detail);

    way_detailService.update(uWD);

    way_detail = new Way_detail();
    uWD=new Way_detail();
    return "update";
}
```

图 5.17 线路管理核心代码

## 5.5 配送点管理模块

本系统的配送点管理主要功能是：配送点管理是由省公司完成的，有配送点的浏览、添加、修改、删除。

- 配送点管理主界面如图 5.18 所示：



图 5.18 配送点主界面

- 配送点管理添加界面如图 5.19 所示：

添加配送点

站点名称:

联系方式:

地址:

添加

取消

图 5.19 添加配送点界面

- 配送点管理编辑界面如图 5.20 所示：

修改配送点:

站点编号:

5

站点名称:

郑州

联系方式:

666888

地址:

金水区

修改

取消

图 5.20 修改配送点

核心代码如图 5.21 所示:

```
public String list(){
    //获取 list 线路信息
    this.init(stationService.selCount());
    List<Station> list=stationService.selList(this.getCurPage());
    HttpServletRequest req = ServletActionContext.getRequest();
    System.out.println(list.size());

    System.out.println(list);

    req.setAttribute("list", list);
    return "list";
}
public String add(){
    System.out.println(station.toString());
    stationService.add(station);
    System.out.println(station.toString());
    station=null;
    return list();
}
public String del(){
    System.out.println("DELid:"+station.getId());
    stationService.del(station);
    System.out.println("DELid:"+station.getId());
    station=null;
    return list();
}
/**
 * 显示 List
 * @return
 */
public String update(){
    stationService.update(station);
    return list();
}
```

图 5.21 配送点管理核心代码

## 5.6 车辆管理模块

车辆管理的主要功能是: 车辆管理是由省公司负责的, 省公司负责购买车辆, 以及车辆的维护; 在该系统中我们只需要实现车辆的增、删、改。

- 车辆管理主界面如图 5.22 所示：

当前位置：[首页](#) / 车辆管理

车辆管理

+ 添加车辆

| 序号 | 车辆编号 | 车牌号     | 载重 | 容量 | 操作                                    |
|----|------|---------|----|----|---------------------------------------|
| 1  | 1    | 陕A89234 | 12 | 20 | <a href="#">修改</a> <a href="#">删除</a> |
| 2  | 2    | 川C98746 | 30 | 50 | <a href="#">修改</a> <a href="#">删除</a> |
| 3  | 3    | 苏H63526 | 20 | 30 | <a href="#">修改</a> <a href="#">删除</a> |
| 4  | 4    | 赣A67421 | 90 | 70 | <a href="#">修改</a> <a href="#">删除</a> |
| 5  | 10   | 桂R35621 | 22 | 22 | <a href="#">修改</a> <a href="#">删除</a> |

共5页, 总24条记录, 第1页 [首页](#) [上一页](#) [下一页](#) [末页](#)

打印

图 5.22 车辆管理主界面

- 车辆管理添加界面如图 5.23 所示：

添加车辆

×

车牌号:

载重:

容量:

添加

取消

图 5.23 车辆管理添加界面

- 车辆管理编辑界面如图 5.24 所示：

修改车辆信息

×

车辆编号:

10

车牌号:

桂R35621

载重:

22

容量:

22

修改

取消

图 5.24 车辆管理修改界面

核心代码如图 5.25 所示：

```
public String list() {  
  
    this.init(vehicleService.getTotalCount(vehicle));  
    // 初始化列表 List  
    List<Vehicle> list = vehicleService.getAllVehicle(vehicle, this.getCurPage());  
    HttpServletRequest req = ServletActionContext.getRequest();  
    // 获取 list  
    req.setAttribute("list", list);  
  
    return "list";  
}  
  
public String add() {  
    vehicleService.add(vehicle);  
    vehicle=null;  
    return list();  
}  
  
public String del() {  
    vehicleService.del(vehicle);  
    vehicle=null;  
    return list();  
}  
  
public String update(){  
  
    vehicleService.update(vehicle);  
    vehicle=null;  
    return list();  
}
```

图 5.25 车辆管理模块核心代码

## 5.7 货物管理模块

货物管理的主要功能是：管理货物种类，实现对货物的正，删，改。

- 货物管理模块的主题页面如图 5.26 所示：



图 5.26 货物管理主体页面

- 货物管理模块的添加页面如图 5.27 所示:

添加货物种类

货物种类:

描述:

添加

取消

图 5.27 货物管理添加页面

- 货物管理模块的修改页面如图 5.28 所示:

修改货物种类信息:

货物种类编号:

1

货物种类:

数码

描述:

数码设备

修改

取消

图 5.28 货物管理的修改页面

货物管理模块核心代码如下图 5.30 所示:

```
public String list(){
    this.init(goodsService.selCount());
    List<Goods> list=goodsService.selList(this.getCurPage());
    HttpServletRequest req = ServletActionContext.getRequest();
    System.out.println(list.size());
}
```

```
        req.setAttribute("list", list);  
        return "list";  
    }  
    public String add(){  
  
        goodsService.add(goods);  
        goods=null;  
        return list();  
    }  
    public String del(){  
        goodsService.del(goods);  
        goods=null;  
        return list();  
    }  
    public String update(){  
        goodsService.update(goods);  
        goods=null;  
        return list();  
    }  
}
```

图 5.30 货物管理核心代码



## 第六章 配置文件设置

### 6.1 Struts2 配置文件

Struts2 框架中使用包来管理 action，避免了 Servlet 在 web.xml 中难以管理的与维护的局面。包的作用和 java 中的类包是非常类似的，它主要用于管理一组业务功能相关的 action，在实际应用中，我们应该把一组业务功能相关的 action 放在同一个包下。配置包时必须指定 name 属性，该 name 属性值可以任意取名，但必须唯一，如果其他包要继承该包，必须通过该属性进行引用，包的 namespace 属性用于定义该包的命名空间，命名空间作用为访问该包下的 action 路径的一部分，见示例。namespace 属性可以不配置，如果不指定该属性，默认的命名空间为“ ”。

本管理系统中的 Struts2 配置文件分为四个模块管理，最后进项整体的合并如下图 6.1 所示：

```
<struts>
<include file="st.xml"/>
<include file="fengt.xml"/>
<include file="l_struts.xml"/>
<include file="flt.xml"/>
</struts>
```

图 6.1 整体 Struts2 文件

A. st.xml Struts2 配置文件如下图 6.2:

```
<struts>
  <package name="st" extends="struts-default">
    <action name="vehicleAction!*" method="{1}" class="vehicleAction">
      <result name="list">/JSP/Vehicle/vehicle_sel.jsp</result>
    </action>
    <action name="goodsAction!*" method="{1}" class="goodsAction">
      <result name="list">/JSP/goods/goodsType.jsp</result>
    </action>
  </package>
</struts>
```

图 6.2 st.xml 配置文件

B. fengt.xml Struts2 配置文件如下图 6.3:

```
<struts>
  <package name="fengt" extends="struts-default">
    <action name="ordersAction!*" method="{1}" class="ordersAction">
      <result name="list">/JSP/Bill/order.jsp</result>
      <result name="add" type="redirectAction">

```

```

        <param name="actionName">ordersAction!list</param>
    </result>
    <result name="update">/JSP/Bill/orders_update.jsp</result>
</action>
</package>
</struts>

```

图 6.3 fengt.xml 配置文件

## 6.2 Spring 配置文件

spring 配置文件是用于指导 Spring 工厂进行 Bean 生产、依赖关系注入（装配）及 Bean 实例分发的“图纸”。Java EE 程序员必须学会并灵活应用这份“图纸”准确地表达自己的“生产意图”。Spring 配置文件是一个或多个标准的 XML 文档，applicationContext.xml 是 Spring 的默认配置文件，当容器启动时找不到指定的配置文档时，将会尝试加载这个默认的配置文档。

本物流系统的 Spring 配置文件如下图 6.4 所示：

```

<!-- 自动的扫描指定的包下面的类 -->
<context:component-scan base-package="com.dao,com.service,com.action"/>
<!-- <context:component-scan base-package="com" /> -->

<!-- 数据库连接池
    池子里放了若干个数据库连接,有用户需要从池里面取,用完了就放回来
-->
<!-- 数据源 -->
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
    <!-- <property name="url" value="jdbc:oracle:thin:@10.10.84.39:1521:xe" /> -->
    <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe" />
    <property name="username" value="tao" />
    <property name="password" value="123" />
</bean>

<!-- 配置 Mybatis -->
<!-- spring 和 MyBatis 完美整合, 不需要 mybatis 的配置映射文件 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <!-- 自动扫描 mapping.xml 文件 -->
    <property name="mapperLocations"
value="classpath:com/mapper/*.xml"></property>
    <!-- 自动配置别名 -->
    <property name="typeAliasesPackage" value="com.pojo"></property>
    <property name="configLocation" value="classpath:mybatis-
configuration.xml"></property>
</bean>

```

```

<!-- DAO 接口所在包名, Spring 会自动查找其下的类 -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="com.mapper" />
    <property name="sqlSessionFactoryBeanName"
value="sqlSessionFactory"></property>
</bean>

<!-- 声明式事务 -->
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>

<!-- 配置方法的隔离级别 -->
<tx:advice id="txAdvice" transaction-manager="transactionManager">
    <tx:attributes>
        <tx:method name="get*" read-only="true"/>
        <tx:method name="save*" propagation="REQUIRED" read-only="false"/>
        <tx:method name="add*" propagation="REQUIRED"/>
        <tx:method name="update*" propagation="REQUIRED"/>
        <tx:method name="del*" propagation="REQUIRED"/>
        <tx:method name="test*" propagation="REQUIRED"/>
    </tx:attributes>
</tx:advice>

<!-- 配置事务切点 -->
<aop:config>
    <aop:pointcut id="txPoint" expression="execution(* com.web.service.*.*(..))"/>
    <aop:advisor advice-ref="txAdvice" pointcut-ref="txPoint"/>
</aop:config>

```

图 6.4 Spring 配置文件

### 6.3 MyBatis 配置文件

本物流系统在设计过程中需要用到一个插件，就是分页插件，所以进行了分页插件的配置。

配置方法如下图 6.5 所示：

```

<configuration>

    <settings>
        <setting name="lazyLoadingEnabled" value="true"/>
    </settings>

```

```
<setting name="aggressiveLazyLoading" value="false"/>
</settings>
<!-- 配置分页插件 -->
<plugins>
  <plugin
interceptor="com.github.miemiedev.mybatis.paginator.OffsetLimitInterceptor">
    <property name="dialectClass"
value="com.github.miemiedev.mybatis.paginator.dialect.OracleDialect"/>
  </plugin>
</plugins>
</configuration>
```

图 6.5 MyBatis 配置文件

## 第七章 系统测试

### 7.1 软件测试的原则和方法

每个软件在开发过程中都会遇到不可预知的问题，在每一个开发周期当中都会残生不可回避的错误和问题，为了能够更好地消除各个方面的问题，避免以后工作中造成不必要的损失，软件在投入使用之前都将必须经过全面和规范的测试。

软件系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足需求分析报告中的定义，并找到与需求规格说明书中不相符且与之矛盾的地方，从而提出更加完善的方案。系统测试发现的问题，必须根据手册进行调试和改正。

### 7.2 单元测试

功能测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，测试的独立单元必须在与程序的其他部分相隔离的情况下进行单元测试。单元测试不仅仅是作为编码是否有错误的一种辅助手段在一次性的开发过程中使用，单元测试必须相对来说是可重复性的，无论是在软件编码修改，或者是移植到新的运行环境进行二次开发过程中，都必须进行测试。因此，所有的软件测试行为都必须在整个软件系统的生命周期中进行维护。下面将列出本系统的模块测试及结果。

用户注册测试结果信息如表 7-1 所示：

表 7-1 注册测试信息表

| 测试信息        | 测试数据              | 测试结果           |
|-------------|-------------------|----------------|
| 用户名、密码、确认密码 | 已存在的用户名           | 提示“用户名存在”，不能登录 |
| 用户名、密码、确认密码 | 用户名正确，密码不一致       | 提示“密码不一致”      |
| 用户名、密码、确认密码 | 用户名、密码、确认密码<br>合法 | 提示注册成功         |

用户登录测试结果信息如表 7-2 所示：

表 7-2 用户登录测试信息表

| 测试信息   | 测试数据        | 测试结果            |
|--------|-------------|-----------------|
| 用户名、密码 | 用户名正确，密码不匹配 | 提示“登录失败”，返回登录界面 |
| 用户名、密码 | 用户名正确，密码正确  | 提示“登录正确”        |

用户列表测试模块如表 7-3 所示：

表 7-3 用户列表测试模块

| 功能名称   | 输入                                  | 期望输出                   | 测试结果   |
|--------|-------------------------------------|------------------------|--------|
| 查看用户列表 | 选择查询条件，点击【查询】                       | 显示用户列表信息，              | 符合期望输出 |
| 修改用户信息 | 选择列表中某用户，点击【修改】，并在修改页输入修改的内容，点击【修改】 | 系统返回到用户列表页面，反之，提示出错信息。 | 符合期望输出 |
| 删除用户   | 选择列表中的某用户，点击【删除】                    | 刷新用户列表                 | 符合期望输出 |
| 添加用户   | 输入添加用户账号、密码等信息，点击【保存】               | 显示保存结果，刷新用户列表。         | 符合期望输出 |

线路管理测试模块如表 7-4 所示：

表 7-4 线路管理测试模块

| 功能名称      | 输入                                   | 期望输出                  | 测试结果   |
|-----------|--------------------------------------|-----------------------|--------|
| 查看线路列表    | 选择查询条件，点击【查询】                        | 显示线路查询信息，             | 符合期望输出 |
| 修改线路信息    | 选择列表中某条线路，点击【修改】，并在修改页输入修改的内容，点击【修改】 | 系统返回到线路信息表，反之，提示出错信息。 | 符合期望输出 |
| 删除线路      | 选择列表中的某条线路，点击【删除】                    | 刷新线路列表                | 符合期望输出 |
| 添加线路      | 输入添加线路的开始和结束点，以及线路名，点击【保存】           | 显示保存结果，刷新线路列表。        | 符合期望输出 |
| 上下移动中间线路点 | 点击上移动或者下移动线路点                        | 选中的线路点发生移动            | 符合期望输出 |

订单管理测试模块如表 7-5 所示：

表 7-5 订单管理测试模块

| 功能名称   | 输入                                   | 期望输出                   | 测试结果   |
|--------|--------------------------------------|------------------------|--------|
| 查看订单列表 | 选择查询条件，点击【查询】                        | 显示订单列表信息，              | 符合期望输出 |
| 修改订单信息 | 选择列表中某张订单，点击【修改】，并在修改页输入修改的内容，点击【修改】 | 系统返回到订单列表页面，反之，提示出错信息。 | 符合期望输出 |
| 删除订单   | 选择列表中的某张订单，                          | 刷新订单列表                 | 符合期望输出 |

|      |                    |                |        |
|------|--------------------|----------------|--------|
|      | 点击【删除】             |                |        |
| 添加订单 | 输入添加订单的所需信息，点击【保存】 | 显示保存结果，刷新订单列表。 | 符合期望输出 |

配送单管理测试模块如表 7-6 所示：

表 7-6 配送点管理测试模块

| 功能名称    | 输入                                   | 期望输出                    | 测试结果   |
|---------|--------------------------------------|-------------------------|--------|
| 查看配送点列表 | 选择配送点管理                              | 页面刷新，显示配送点列表信息，         | 符合期望输出 |
| 修改配送点信息 | 选择列表中个配送点，点击【修改】，并在修改页输入修改的内容，点击【修改】 | 系统返回到配送点列表页面，反之，提示出错信息。 | 符合期望输出 |
| 删除配送点   | 选择列表中的某配送点，点击【删除】                    | 刷新配送点页面列表               | 符合期望输出 |
| 添加配送点   | 输入添加的配送点信息，点击【保存】                    | 显示保存结果，刷新配送点列表。         | 符合期望输出 |

货物管理测试模块如表 7-7 所示：

表 7-7 货物管理测试模块

| 功能名称   | 输入                                  | 期望输出                   | 测试结果   |
|--------|-------------------------------------|------------------------|--------|
| 查看货物列表 | 选择货物管理                              | 页面刷新，显示货物列表信息，         | 符合期望输出 |
| 修改货物信息 | 选择列表中个货物，点击【修改】，并在修改页输入修改的内容，点击【修改】 | 系统返回到货物列表页面，反之，提示出错信息。 | 符合期望输出 |
| 删除货物   | 选择列表中的某货物，点击【删除】                    | 刷新货物页面列表               | 符合期望输出 |
| 添加货物   | 输入添加的货物信息，点击【保存】                    | 显示保存结果，刷新货物列表。         | 符合期望输出 |

车辆管理测试模块如表 7-8 所示：

表 7-8 车辆管理测试模块

| 功能名称   | 输入                                 | 期望输出                   | 测试结果   |
|--------|------------------------------------|------------------------|--------|
| 查看车辆列表 | 选择车辆管理                             | 页面刷新，显示车辆列表信息，         | 符合期望输出 |
| 修改车辆信息 | 选择列表中车辆信息，点击【修改】，并在修改页输入修改的内容，点击【修 | 系统返回到车辆列表页面，反之，提示出错信息。 | 符合期望输出 |

|      |                  |                |        |
|------|------------------|----------------|--------|
|      | 改】               |                |        |
| 删除车辆 | 选择列表中的某车辆，点击【删除】 | 刷新车辆页面列表       | 符合期望输出 |
| 添加车辆 | 输入添加的车辆信息，点击【保存】 | 显示保存结果，刷新车辆列表。 | 符合期望输出 |

## 7.3 整体测试

完成基本的单元测试后，系统测试将真正的进入部署测试环节，主要的步骤有以下几步：

- 1) 首先，对系统进行部署，添加到 web 服务器中，本系统使用的是 Tomcat7.0.
- 2) 之后，启动 web 服务器。
- 3) 最后，在浏览器中输入访问地址，访问系统，进行系统的各个功能的测试，检验系统是否符合需求分析报告的妖气，经过多轮的实验和反复验证，发现并记录系统中存在的问题和问题并及时修改与完善，最终实现系统的完美交付。



## 第八章 结论

经过从选题到毕业设计代码书写完成,再到毕业论文定稿,每一个环节都在用心去感受,毕业设计的完成也就标志着我们大学生活即将结束,这个管理系统用了差不多两个半月的时间,在这其中的快乐与辛苦可能只有自己能体会,一路过来学会了很多也感触良多。

首先,在一个系统的开发初期,需要一个完善的需求分析,设计数据库和设计每个模块应该具备的功能,然后建立模块之间的整体架构,实现每个模块的基本功能。当基本功能完成之后,接下来就要设计界面,调试每个色素、距离,优化每个细节。再不断的完善一些基本功能,最终完成整个物流系统的开发设计。因为时间短,任务重。只有学会合理分配才能更高效的完成需求。所以也在期间学习到了系统开发流程以及如何更高效的完成一个任务的能力。

其次,就是在具体的程序实现环节,编写代码实现相关功能时,当问题出现时,如何去处理。本次物流系统开发,Java 类中学会了如何设置断点调试,因为以前基本不愿意去用设置断点。前端开发中利用 JavaScript 中的 alert 输出信息来查找可能的错误,总之,解决问的方法比以前更加丰富了,在一个系统的开发中要学会代码的优化。在错误检查阶段,需要的可能是耐心和细心。

毕业设计对我们毕业生来说不仅仅是一个任务,可能是我们人生中不可或缺的一次经历和磨练,它能使我们获得了只是面的扩充,基本技能的提升,更重要的可能的就是积累了经验。我相信毕业后工作后,这次的毕业设计将会影响我的每次开发和每行代码的书写。

## 参考文献

- [1] 霍斯特曼.Java 核心技术 卷一 基础知识, 机械工业出版社, 2007
- [2] Bruce Eckel.Thinking in Java (第四版), 机械工业出版社, 2007.
- [3] 刘晓华.JSP 应用开发详解, 电子工业出版社, 2007
- [4] 何明/何倩颖.Oracle 快速 Web 应用开发-从实践中学习 Oracle Application Express, 电子工业出版社, 2010
- [5] 孙卫琴.精通 Struts:基于 MVC 的 Java Web 设计与开发, 电子工业出版社, 2004
- [6] 陈夫真.基于 SSM 的某高校教室管理信息系统的设计与实现[D].江苏省: 苏州大学,2012.
- [7] 宋雨/赵文清.软件工程, 电子出版社, 2006
- [8] 孙卫琴.Tomcat 与 Java Web 开发技术详解[M]. 第 2 版. 电子工业出版社, 2009
- [9] 刘腾红, 孙细明.信息系统分析与设计[M]. 北京: 科学出版社, 2003 年
- [10] 单东林/张晓菲.锋利的 jQuery[M].魏然人民邮电出版社 2009-6
- [11] (美)David.Flanagan. JavaScript 权威指南(第六版)[J].机械工业出版社 2012.3
- [12] 张洪伟.JSP 网络开发技术与整合应用, 清华大学出版社, 2006
- [13] 李海生、郭跃.软件测试技术案例教程[M].清华大学出版社, 2012 年 4 月第一版
- [14] 黄平主编.数据库技术.北京: 机械工业出版社, 2002.

## 致谢

时光荏苒，岁月如梭。转眼间四年大学时光就要过去，我就要对陪伴我四年的母校说再见了。在有限的大学时光中，我收获良多。不仅掌握了很多的科学文化知识，还懂得了许多做人的准则和规则。首先要感谢我的指导老师——杨雷老师，他对事物的精益求精，勤奋好学，热爱探索的精神深深的激励和感染着我，在实践学习过程中，让我的专业能力不断提高。在他的帮助下，我克服了很多学习中的困难，在我一步步完成毕业的几个月里，杨雷老师的严格知道对我的毕业设计有很大的帮助，不至于我迷失方向。我跟杨老师学到的并不仅仅是这个物流管理系统如何去实现，更重要的将是在今后的工作中当我遇到不会的问题，我如何去克服。毕业设计对我来说不再是一份作业。最后我会带着老师们这四年里对我的所有教诲走向我人生的下一个阶段，用我最好的状态去面对未来更大、更强的挑战。

其次我想去感谢我在毕业设计过程中帮助和支持我的老师和朋友们，谢谢你们对我的支持和关心，让我在四年的学习生涯中干道充实和快乐，能够向着自己既定的目标前进。