**湖 南 科 技 大 学**

**毕 业 设 计（ 论 文 ）**

|  |  |
| --- | --- |
| **题目** | **基于Java的物流管理系统的设计与实现** |
| **作者** | **罗福良** |
| **学院** | **计算机科学与工程** |
| **专业** | **信息安全** |
| **学号** | **1605030307** |
| **指导教师** | **李志刚** |

二〇二〇 年 五 月 二十五 日

**湖 南 科 技 大 学**

**毕业设计（论文）任务书**

计算机科学与工程 院 信息安全 系（教研室）

系（教研室）主任:（签名） 年 月 日

**学生姓名: 罗福良 学号: 1605030307 专业: 信息安全**

1 设计（论文）题目及专题： 基于Java的物流管理系统的设计与实现

2 学生设计（论文）时间：自 2020 年 5 月 15 日开始至 2020 年 6 月 7 日止

3 设计（论文）所用资源和参考资料：

4 设计（论文）应完成的主要内容：

5 提交设计（论文）形式（设计说明与图纸或论文等）及要求：

6 发题时间： 年 月 日

指导教师： （签名）

学 生： （签名）

**湖 南 科 技 大 学**

**毕业设计（论文）指导人评语**

[主要对学生毕业设计（论文）的工作态度，研究内容与方法，工作量，文献应用，创新性，实用性，科学性，文本（图纸）规范程度，存在的不足等进行综合评价]

**指导人：** （签名）

年 月 日

**指导人评定成绩：**

**湖 南 科 技 大 学**

**毕业设计（论文）评阅人评语**

[主要对学生毕业设计（论文）的文本格式、图纸规范程度，工作量，研究内容与方法，实用性与科学性，结论和存在的不足等进行综合评价]

**评阅人： （签名）**

**年 月 日**

**评阅人评定成绩：**

**湖 南 科 技 大 学**

**毕业设计（论文）答辩记录**

**日期：**

**学生： 学号： 班级：**

**题目：**

**提交毕业设计（论文）答辩委员会下列材料：**

**1 设计说明书/ 论 文 共 页**

**2 设计（论文）图 纸 共 页**

**3 指导人、评阅人评语 共 页**

**毕业设计（论文）答辩委员会评语：**

[主要对学生毕业设计（论文）的研究思路，设计（论文）质量，文本图纸规范程度和对设计（论文）的介绍，回答问题情况等进行综合评价]

**答辩委员会主任：** （签名）

**委员：** （签名）

（签名）

（签名）

（签名）

**答辩成绩：**

**总评成绩：**

**摘 要**

在我国目前各个行业都快速发展的国情下，尤其是互联网行业发展更加快速，传统的物流管理很难满足当下的需求，不仅费时费力，效率也不高。随着各行各业对物流的需求的快速增长，同时对物流效率的要求也越来越高。这时一个好的物流管理系统对于一个第三方物流企业来说就尤为重要，不仅能提高物流企业的效率，方便物流公司对于物流方面以及人员方面的管理，对于客户的体验也有很大的改善。

在对赣部分小型物流企业进行调查后，结合目前的实际情况进行了系统的可行性分析以及需求分析，确定了系统所需模块，以及各个模块的功能，结合目前流行的技术与框架，以及个人的熟练程度，实现了这个基于Java的物流管理系统。系统主要实现了管理员端也就是企业端，还有客户端，以及司机端。企业端主要包括客户、司机、员工的管理以及货运单与货运合同的签发与结算，还有就是运输路线的管理，此外还有统计模块，统计公司的开支情况。司机端主要就是查看货物的运输路线，然后实时上报自己的位置，以及到货后及时通知买卖方进行货物验收，还有就是查询自己的运单情况。客户端主要是查看自己的运单状况，以及货物的运输路线，以及目前货物位置。

对于系统的这些功能性需求以及非功能性需求包括易用性、可用性、安全性、可靠性、易维护性，本文分别从技术层面、数据层面以及、系统层面整体介绍了

整个系统的实现原理、运行机制、以及操作方式。

综合上述需求，本系统决定基于B/S架构，采用spring boot作为主要后端框架，整合spring、spring MVC、spring data JPA框架，借助lombok简化后端代码，基于接口开发增强可读性减少代码耦合，易于维护，借助hutool工具包进行部分数据的加解密。前端使用layui admin美化界面，使用原生前端语言jQuery以及Ajax进行页面控制以及前后端交互。数据库采用mysql数据库。

在采用以上技术，以及对于系统的需求分析及可行性分析的基础上，对系统进行设计，实现了对应的功能。经过测试，该系统运行稳定，安全，性能优良，能够提高相关行业人员的效率，减少企业的支出，增加收入，便于人员管理，以及货物运输的管理，方便客户与司机及时沟通了解货物信息。基本达到本系统研究与设计的目的。

关键词：物流管理；模块化；Java

**ABSTRACT**

The number of our foreign exchange reserves has experienced quick growth over the past thirty years，from breaking 100 billion U.S. dollar in 1996 to breaking 1.5 trillion U.S. dollar in 2007. The so high-speed increase and gigantic scale，are a double-edged sword for our country development of the economy. As a developing country，China definitely needs a certain amount of foreign exchange to ensure external payments，interfere in foreign exchange market and promote the country credit. But excessive exchange cover scale may produce problems in management. At first，this paper introduces the development of our country foreign exchange reserves，analyses the reasons for quick growth of foreign exchange reserves，illustrates the negative effects of excessive foreign exchange reserves，reveals the problems in foreign exchange reserve; Then analyses and comprises the management systems of foreign exchange reserves in five counties; At last，makes recommendations for our country’s management of foreign exchange reserves.

**Keywords:** foreign exchange reserves; international comparison; meanings

目 录

[第一章 绪 论 1](#_Toc515387888)

[1.1 物流管理系统的概述 1](#_Toc515387889)

[1.2 课题研究的背景 1](#_Toc515387890)

[1.3 课题研究的目的和意义 2](#_Toc515387891)

[1.4 物流管理系统的现状 2](#_Toc515387892)

[1.5 本文的主要内容 3](#_Toc515387893)

[第二章 相关工具和技术介绍 5](#_Toc515387894)

[2.1 android平台 5](#_Toc515387895)

[2.1.1 android系统架构 5](#_Toc515387896)

[2.1.2 四大组件和Fragment 5](#_Toc515387897)

[2.2 开发语言 6](#_Toc515387898)

[2.2.1 Java语言 6](#_Toc515387899)

[2.2.1 xml语言 6](#_Toc515387900)

[2.3 android开发工具 6](#_Toc515387901)

[2.3.1 android Studio工具 6](#_Toc515387902)

[2.3.2 Adobe PhotoShop CS6 6](#_Toc515387903)

[2.4 系统主要技术介绍 7](#_Toc515387904)

[2.4.1 SQLite数据库 7](#_Toc515387905)

[2.4.2 GreenDao 框架 7](#_Toc515387906)

[2.4.3 MpandroidChart框架 7](#_Toc515387907)

[2.4.4 BuftterKnife框架 7](#_Toc515387908)

[第三章 需求分析和总体设计 9](#_Toc515387909)

[3.1 概述 9](#_Toc515387910)

[3.2 需求分析 9](#_Toc515387911)

[3.2.1 用户需求 9](#_Toc515387912)

[3.2.2 可行性分析 9](#_Toc515387913)

[3.3 系统总体设计 10](#_Toc515387914)

[3.3.1 功能设计 10](#_Toc515387915)

[3.3.2 界面设计 11](#_Toc515387916)

[3.4 数据库设计 13](#_Toc515387917)

[3.4.1 数据库设计概要 13](#_Toc515387918)

[3.4.2 实体图的设计 13](#_Toc515387919)

[3.4.3 数据库表的设计 15](#_Toc515387920)

[第四章 系统详细设计 17](#_Toc515387921)

[4.1 概述 17](#_Toc515387922)

[4.2 数字键盘编写 17](#_Toc515387923)

[4.3 底部导航栏 17](#_Toc515387924)

[4.4 用例图绘制 18](#_Toc515387925)

[4.5 界面的详细设计 18](#_Toc515387926)

[4.5.1 首页模块 19](#_Toc515387927)

[4.5.2 账单模块 20](#_Toc515387928)

[4.5.3 统计模块 21](#_Toc515387929)

[4.5.4 我的模块 21](#_Toc515387930)

[4.6 功能的详细设计 23](#_Toc515387931)

[4.6.1 个人记账 24](#_Toc515387932)

[4.6.2 金额信息 27](#_Toc515387933)

[4.6.3 图表显示 28](#_Toc515387934)

[4.6.4 登录注册 28](#_Toc515387935)

[第五章 系统实现与测试 31](#_Toc515387936)

[5.1 概述 31](#_Toc515387937)

[5.2 项目包类的介绍 31](#_Toc515387938)

[5.3 系统实现与演示 32](#_Toc515387939)

[5.4 系统测试与分析 39](#_Toc515387940)

[第六章 总结与展望 43](#_Toc515387941)

[参 考 文 献 45](#_Toc515387942)

[致 谢 47](#_Toc515387943)

**第一章 绪 论**

**1.1 物流管理系统的概述**

传统物流行业对于各类凭证例如货运单、运输合同之类的都是采用纸质凭证的方式进行登记、结算，纸质文件容易丢失损坏，难以管理，凭证出问题补办手续繁多耗时长，企业保管大量纸质文件，清查困难，而且也容易受自然条件的影响。手工方式效率低下，难以统计各方面的信息。

在过去的几十年里使用这种方式没有太大的问题，因为物流量不大，对效率的要求没有那么高。但是在如今这个互联网高速发展的时代，各行各业对物流的需求急剧增大，对于物流的效率要求也变得更高，使用传统全手工模式无法满足现在的要求。要想在激烈的物流市场竞争中不被淘汰就得跟上时代的潮流，物流企业必须高度的关注各种信息，及时准确的掌握物流资源的信息以及客户的相应要求，对于客户的需求需要选择更加合理的路线，以便提高效率，控制成本，以及吸引客户。

因此一个跟随时代发展脚步的物流管理系统就变得尤为重要，本设计就是在这种背景下实现的。本系统着重物流管理系统中的管理和物流，主要是对于货物运输的管理，以及对人员的管理，还有就是对于路线的规划。本系统基于web平台，在移动平台和电脑上都能访问。使用这个系统能够统一的管理人员情况，以及货物情况，能够对员工进行管理，能够对客户和司机信息进行录入，能够在系统上填写相应的单据，能够统计收支情况，能够安排路线，相应人员无需在进行大量的手工操作，能够大大的减少纸质文件数量，也方便客户和司机。

**1.2 课题研究的背景**

**（1）市场背景**

这是一个互联网飞速发展的时代，各行各业都开始进行数字化以及信息化，物流行业也是如此。随着网络的普及，以及网上购物的兴起，物流行业早已不适用以前那套手工模式了。因此为了适应时代的发展，物流管理系统是物流企业必要的也是急需的。此时一个优秀的物流管理系统就很容易打开市场的大门，因为这个物流管理系统的缺口数是巨大的。访问国家交通运输物流公共物流平台，能够发现上面很多的物流管理系统，功能还是不够丰富。

物流行业入主十大振兴产业在2009年，可见物流行业信息化的程度还是不够高，物流行业还在起步阶段或者说还在走向成熟的过程中，像那些新兴的物流企业都有比较完善的物流管理系统。但是对于一些更小型的物流企业来说，他们不需要那么复杂的系统，他们要的是更加简洁的系统。这就为这个系统提供了市场机会。

1. **社会背景**

随着网络的普及，以及天猫、淘宝等网上购物平台的兴起，大大刺激了人们的消费欲望。在人们网上大肆购物的同时，也刺激了物流行业的发展，对于物流行业的企业来说，物流量相比以前变得巨大，对于物流的管理也比以前变得艰难，而一个物流管理系统就能很好的为企业解决这个问题。对于企业来说他们需要一个方便管理员工以及物流信息的系统，对于司机来说他们需要一个接货的平台，对于客户来说他们需要一个高效率的物流企业来帮他们运输货物。

**1.3 课题研究内容**

在这个物流管理系统的实现过程中，首先是研究了一些物流企业的需求，接着进行了可行性分析。最终决定基于B/S架构使用Java语言进行编写，采用流行的mvc模式，数据库选用MySQL数据库。基于此本系统实现了用户的登录模块包括管理员、司机、以及客户的登录，对于员工的管理，对于司机的管理，对于客户的管理，发货模块，派单模块，收支模块，路线管理模块，司机端，以及为客户设计的客户端。在这些模块实现的同时，也对这些模块进行了相应的测试，最终的是一个高可用，易维护稳定的系统。

**1.4 物流管理系统的现状**

现在市场中存在着各种各样形形色色的物流管理系统，功能界面各不相同，服务对象也各不相同，因为国内起步较晚，虽然市场上这种系统数量众多，但是并没有出现具有绝对领先优势的品牌，这就留下了大量的机会。同时市场上产品质量参差不齐，目前还存在着产品功能单一、技术落后等问题，只要做的系统足够好就能参与竞争，对系统性能要求没有非常严格。

据不完全统计国内有实力提供物流管理系统的软件公司不到500家，这远远无法满足，国内的物流市场。

**1.5 本文的主要内容**

本文内容概括如下。

1. 绪论

介绍物流管理系统的概述，市场背景，社会背景，课题研究内容，物流管理系统的现状。

1. 相关工具和技术介绍

介绍实现系统的过程中主要所使用的工具和技术，阐述这些工具和技术在现实世界的功能和应用，结合本课题阐述它们起到的作用。

1. 系统需求及可行性分析

是软件工程的需求分析和总体设计过程，是开发编程之前要实现的一项重要步骤。针对界面和数据库等分别阐述预计实现的目标，分别设计。

第四章，系统总体设计

继第三章总体设计后进行界面和功能的详细设计，包括逻辑和控件。

1. 系统演示及测试

是系统完成后的效果展示，阐述完成系统所用的重要包类，以及实现后的效果，对实现结果的基本测试。

1. 总结与展望

是论文的结束语，记录设计和论文实现过程中的感想，对大学的重要人物表达感谢。

1. **系统使用工具和技术简介**

Java语言是一种简单的、跨平台、分布式的、健壮的、安全的、完全面向对象的语言，Java语言对技术人员友好，技术多文档多，在大部分场景都适用。本章主要介绍了Java语言的特点，web的特点，spring系列框架，jQuery，Ajax等技术还有idea、navcat等工具。

**2.1 Java平台**

Java是一种高级的计算机编程语言，拥有跨平台、面向对象、面向泛型编程等特点，在高版本中还提供了函数式编程，广泛运用于企业级web开发以及移动端开发，主要写服务器端。Java主要由四方面组成：Java编程语言、Java类文件格式和Java虚拟机、Java应用程序接口(Java API)。

Java平台由jvmJava虚拟机还有apiJava应用编程接口组成。Java应用编程接口为在此基础上提供了一个独立在操作系统外的标准接口，可分为基本部分和扩展部分。在硬件或操作系统平台里安装一个Java平台（Jre Java运行时环境）之后，Java程序就可以运行。Java平台几乎嵌入了所有的操作系统。因此Java程序可以只编译一次，就能运行在各种系统中。

Java支持面向接口编程这就是经典的面向对象思想，因此用Java语言写的程序可以有很好的扩张性以及维护性。同时Java文档众多，技术也多，因此几乎能用Java语言开发各种各样的程序。

**2.2 spring系列框架**

Spring Boot框架是由Pivotal团队提供的一个全新框架，它设计的目的是用来简化新Spring应用的初始搭建以及开发过程。传统的spring应用配置文件繁多，但是使用spring boot整合后大部分情况下就只需要一个配置文件，同时因为大部分配置都是自动装载的，因此只要配置少量的设置就能启动。此外不使用自动配置也可以通过自己写配置类的方式来实现手动配置。spring boot项目自带tomcat可以不用额外配置tomcat，同时支持打jar包而不像传统web项目只能打war包，springboot项目大部分是基于注解开发的。

Spring框架，主要目的是使JavaEE开发变得更加简单，是一个企业级开发框架，是软件设计层面的框架，优势在于可以将应用程序进行分层，开发者可以自主选择组件。核心是面向切面编程（AOP）还有控制反转（IOC）/依赖注入（DI）。控制反转是指使用IOC容器来统一管理对象的生命周期、依赖关系等。面向切面编程是指通过预编译的方式和运行期间动态代理实现程序功能统一维护的一种技术，是oop的延续，利用aop可以减少业务逻辑各部分间的耦合。Spring框架提供了Ioc容器，同时也提供了易于编写的动态代理类，不像原生Java中的动态代理使用较为复杂。

SpringMVC是⽬前主流的实现 MVC 设计模式的企业级开发框架，Spring 框架的⼀个⼦模块，⽆需整合，开发起来更加便捷。什么是MVC模式？将应⽤程序分为 Controller、Model、View三层，Controller接收客户端请求，调⽤ Model⽣成业务数据，传递给 View。Spring MVC 就是对这套流程的封装，屏蔽了很多底层代码，开放出接⼝，让开发者可以更加轻松、便捷地完成基于MVC模式的Web开发。核心组件包括DispatchServlet、Handle、Handle Mapping、HandleInterceptor、HandleExecutionChain、Handle Adapter、ModelAndVewl、ViewResolver。客户端请求被DisptacherServlet接收。根据 HandlerMapping映射到Handler。⽣成Handler和HandlerInterceptor。Handler和HandlerInterceptor以HandlerExecutionChain的形式⼀并返回给DisptacherServlet。DispatcherServlet通过HandlerAdapter调⽤Handler的⽅法完成业务逻辑处理。Handler 返回⼀个ModelAndView给DispatcherServlet。 DispatcherServlet将获取的ModelAndView 对象传给ViewResolver视图解析器，将逻辑视图解析为物理视图View。ViewResovler返回⼀个View给DispatcherServlet。DispatcherServlet根据View进⾏视图渲染（将模型数据Model填充到视图View。Spring MVC流程⾮常复杂，实际开发中很简单，因为⼤部分的组件不需要开发者创建、管理，只需要通过配置⽂件的⽅式完成配置即可，真正需要开发者进⾏处理的只有Handler、View。

Spring-data-jpa是spring生态中的一员，jpa是一套标准不是产品，Spring-data-jpa它是在JPA规范下提供了Repository层的实现，底层使用的是全自动orm框架Hibernate的JPA技术实现。主要是可以使用jpql查询语言，有以下几个接口，CrudRepository：提供了一些简单的crud操作的方法。PagingAndSortingRepository：增加了分页和排序的方法。SimpleJpaRepository:提供了方法的实现。写dao层时可以直接继承接口，不用自己实现。注意作为自动化框架方法名必须符合命名规范才能识别，不然需要自己写sql。

**2.3 前端相关技术**

jQuery是一个高效、快速、简洁的Javascript代码库，封装了很多原生的JavaScript方法，提供一种简便的JavaScript开发设计模式，旨在减少前端脚本开发的代码里，优化HTML文档、事件处理、于Ajax交互及动画设计。jQuery基本兼容各种主流浏览器，因此也是一个十分流行的前端框架。jQuery生态良好，文档众多，也不断在推出新版本，是前端框架很好的选择。

Ajax是动态jQuery和xml的简写，它并不是一种新的编程语言，仅仅是一种新的编程技术，使用它可以精简代码，开发出更高效、更简洁且交互性更强的程序，比如说动态下拉使用Ajax就非常方便。Ajax使用JavaScript在web服务器和web浏览器之中交互交换数据。而且是在幕后与后台服务器交互，不用用户整个提交，可以实现网页数据的动态效果。Ajax也兼容各种主流浏览器，且程序独立于浏览器和平台，因此也是跨平台的。它也简化了前端开发的难度。

Layui经典前端模块化框架，使页面比原生的页面更加美观，layui更多的面向后端的开发人员，因为前端页面并不是大部分后端开发人员熟悉的，layui内置模块众多因此基本能满足大部分需求。

**2.4 数据库**

MySQL关系型数据库管理系统是一个成熟的关系型数据库管理系统，而且是开源的并且有社区版和商业版，大部分情况下社区版都适用。MySQL使用的sql语言是标准化的数据库访问语言。MySQL效率高，基于红黑树的索引机制使MySQL的开发效率高。MySQL支持事务，并且支持行级锁而不是统一表锁，使效率更高。支持多个引擎可以自由选择。

**2.5 开发工具**

Intellij IDEA使Java编程语言开发的集成开发环境，尤其在智能化代码助手方面，例如代码提示，重构，JavaEE支持，整合版本控制工具git和svn，spring应用快速创建，集成maven，整合junit等。开发spring运用时可以不用去spring官网创建spring项目，直接就能在工具上操作。提供了大量快捷键，还能自己定义各种快捷键，还有文档注释模板。

navcat是一款MySQL可视化工具，可以便捷的操作数据库。

Nginx反向代理服务器，可以解决部分前端跨域问题，同时可以使用它将ip地址映射成域名。

**第三章 需求分析**

**3.1 概述**

需求分析的任务是确定系统必须完成哪些工作，对目标系统提出完整的、准确、清晰、具体的要求[1]。它是软件开发计划阶段的重要组成部分，为了促进软件开发的更加规范，需求分析务必得做的谨慎。需要综合考虑到用户所需功能，开发时间、开发成本、以及开发技术难度等，功能不宜过多，能满足客户需求就可以。但是一定得保证客户的需求得到满足，包括功能性需求和非功能性需求。

本章从软件设计层面描述了用户的需求，以及界面的需求。

**3.2 需求分析**

**3.2.1 用户需求**

**（1）总体需求**

传统的物流行业都是采用手工的方式记录各种凭证，和客户签订货运合同使用纸质凭证，和司机签订运输合同也是人工纸质凭证，结算时也是要人去现场结算。丢失凭证，处理很麻烦，公司保存凭证也不方便。当需要报表时，以及进行会计核算时，难以清点众多纸质文档。

因此，设计一个各个方面都能使用电子凭证的物流系统就尤为重要。将各类凭证存在数据库中，既方便查找、结算、又便于保存，还能随时备份数据，防止数据丢失。

**（2）功能需求**

根据物流公司的需求，本物流管理系统应包含的功能有以下几点。首先是企业端也就是管理员所需要的功能，有用户的添加删除修改查询，用户包括公司职员、司机以及客户。路线的管理，路线的增加删除修改查询，以及为货物选择运输路线。对资金的管理，包括司机的结算、客户的结算、员工工资、手工调账。

与客户签订货运单，查看各种状态的货运单，与司机签订运输合同，查看各种状态的运输合同，修改密码。客户端需要的说查看货运单信息，查看货物路线，与目前位置，接收货物，货运单结算，修改密码，投诉。司机端主要时查看运输合同，查看货物路线，上报货物位置信息，查看收入信息，修改密码。

**（3）界面需求**

界面上需要完成的界面比较多，主要有登录页面，然后就是管理模块，路线规划模块，货运单模块，货运合同模块，司机模块，客户模块，资金模块，每个模块下面有若干个界面。

**3.2.2 非功能性需求**

应该尽可能的保证系统运行的稳定性，减少系统宕机次数，减少宕机时恢复服务的时间，因为是个web项目，当出现故障时重启就行，因此可以满足基本要求。当数据量增大时可以调整为分布式架构。应该要保证关键数据的安全例如，用户的口令就应该加密存储。对用户权限要区分，进行相应的访问控制。要保持程序的可读性，易用性，可维护性，程序应该添加适当的注释，界面上的提示应该通俗易懂，表单内容应该见名知意，同时要面向接口基于注解编程。

**3.2.3 可行性分析**

除却用户已提出的需求外，可行性分析也是需求分析中不可缺少的一环，可行性分析主要从其他非实际需求方面分析软件，例如开发需要的成本多少，开发的技术选择以及技术的难度，还有就是综合考虑社会因素，看这个软件是不是合理、合法、合乎道德的。因此可行性分析就应该从尽量多的方面考虑设计方案是否可行。

**（1）经济可行性分析**

首先开发本系统的软件IntellijIDEA是需要获取正版授权的，但是在校生可以凭借学校邮箱免费获取教育版凭证，数据库MySQL开源社区版免费获取，navcat软件试用期足以开发完，及时到期激活成本也不高。因此本软件开发经济成本不高，主要是花时间。市面上此类软件价格并不低廉，且此类系统市场缺口巨大，付出的成本远小于收益，因此本系统在经济上是可行的。

**（2）技术可行性分析**

本系统主要采用Java语言开发，此前曾经系统的学习过此项技术，基础知识基本掌握，即使遇到困难也能很快找到资料去解决。框架使用并不是很熟练，不过可以看教学视频，而且并不用去寻根溯源只需要使用，因此也灭有困难。前端虽然没怎么学过，但是也多多少少接触过一些，因此也能快速上手，遇到困难也能很快解决。所以本系统在技术上也没有太大的问题。

**（3）社会环境可行性分析**

最后就是社会环境可行性分析，本系统基于帮助物流企业迈向信息化自动化的目的开发，系统中不涉及到违反道德、违反法律的内容，整个系统设计的目的都是为了企业更好的发展，变现出一种积极、正能量的态度。因此在社会环境可行性分析上也没有问题。

**第四章 总体设计**

本物流管理系统时一个信息管理平台，总体是基于B/S架构的，使用nginx作为反向代理服务器，使用spring框架作为服务端框架，使用的是MVC模式，系统的数据库采用mysql数据库，旨在为用户提供一个可靠、可用、稳定、高性能的物流管理系统。

**4.1 设计模式**

提到mvc模式相信只要是程序员都不会陌生，本系统选用的设计模式就是流行的模型-视图-控制器模式。视图是指用户看到的界面，模型则表示的是企业的数据和业务逻辑，控制器则是接受用户的请求调用对应的视图和模型去完成请求。将业务逻辑、数据、以及界面分离，这种分层模式有利于复杂程序的管理，可以使程序更加简洁，同时在改动界面，如果功能不变，则不用修改业务代码。

同时这种模式也使得代码维护起来更加简单方便。

**4.2 功能设计**

**4.2.1 系统模块设计**

本系统基于对物流企业相关业务的分析，实现了能够协助企业相关人员更加方便的处理大部分的物流业务。在不同的用户登录后会根据权限的不同看到不同的页面，管理员登陆后能够进行人员、路线的管理，同时也能管理货运单和货运合同，能够查看财务情况，能够手工调账。客户登录能够看到自己的货运单情况，

能够查看路线，接收货物，投诉司机。司机登录能看到自己运输合同的情况，能够查看路线，能够上报自己的位置。完整的系统模块图如图4.1、图4.2、图4、3所示。



**图4.1 管理员管理模块图**



**图4.2 司机模块图**

****

**图4.3 客户模块图**

**4.2.2 模块具体功能描述**

**（1）登录模块**

在首页输入账号密码后点击登录按钮，如果密码正确，就会根据用户所在的用户组分配相应的权限，分配对应权限的页面，如果密码错误则提示密码错误。

**（2）修改密码模块**

如果是管理员登陆，则点击修改密码可以修改本用户的密码，需要验证旧密码，新密码需重复输入两次，当两次一样时，修改成功，需要重新登陆。选择左侧导航栏中的系统管理模块下的用户密码修改则可以修改其他用户的密码。

如果是其他用户登录则点击右侧修改密码只能修改自己的密码，需要验证当前密码，需要重复输入两次新密码，当两次新密码一样，提交后则密码修改成功，之后会返回首页，需要重新登陆。

**（3）人员管理模块**

管理员登陆后，点击左侧系统管理后，选择员工管理，则可以选择对应的页面，对员工的信息进行增加、删除、查询、更新。

选择司机管理后，选择对应的页面就可以对司机的信息进行增加、删除、查询、更新。

选择客户管理后，选择对应的页面就可以对司机的信息进行增加、删除、查询、更新。

选择权限管理后，选择对应的页面就可以进行用户组管理，包括用户组的增加、删除、查询、修改，同时还能根据用户组，分配对应的权限。

**（4）路线管理模块**

管理员登陆后，点击左侧导航栏的路线管理之后，先选择城市页面，进行基础城市的添加，之后选择路线管理，对路线的信息进行增加、删除、修改以及展示，同一个起点和终点可以有不同的运输路线，运输时间和运输距离都有差距。

**（5）货运单管理模块**

管理员登陆后，点击左侧导航栏的货运单管理之后，选择货运合同可以进行货运合同的签订，然后可以查看各种状态的货运合同，可以根据客户的需求选择合理的路线等，未分发的货运单可以修改、删除。

**（5）运输合同管理模块**

管理员登陆后，点击运输合同管理，可以将已签订的货运单生成的运输合同进行发放，根据货物及客户的要求选择合理的车去配送，同时选择路线，未分发的合同可以修改，删除。

**（6）财务管理模块**

管理员登录后，选择财务管理模块，可以查看收支情况，对于发生错账等情况，可以进行手工调账。

**（7）司机端模块**

司机登录后，选择运输合同模块可以查询自己的运输合同情况，以及实时更改正在配送的货物状态。可以查询路线信息，然后根据路线信息将自己的大概位置上报。可以查看自己的收支情况。

**（8）客户端模块**

客户登录后，选择查看自己的运单信息，也可以查看正在配送的货物的路线，以及大概的实时位置，可以进行货运单的结算。

**4.3 数据库设计**

**4.3.1 数据库设计概要**

数据库设计是指根据客户需求，在某个具体的数据库系统中，设计数据库结构和建立数据库以及表的过程。设计原则一一对一原则，在软件开发过程中，需要遵循一对一的原则，此原则能够尽可能的减少数据维护问题的出现，在保证维护工作正常进行的同时能够降低维护的难度。在这个过程中需要避免较大的数据以及杂乱的数据出现，不然会增加工作难度，使工作进度推进缓慢。原则二独特命名原则，这项原则可以防止重复命名的出现，同时通过命名规范能够对大部分表见名知义，能够使开发变得简单，同时也能锻炼程序员的命名能力，培养程序员良好的命名习惯。三是双向使用原则，包括事务使用原则以及索引功能原则。通过前者能够使得非事务模块数据提供基础保障。后者则可以加快查询速度，能够基于多列灵活排序。

利用这些数据库设计原则可以节约内存资源、cpu资源，能够提高软件的运行效率，能够减少软件出现故障的概率，同时也便于管理和维护。

**4.3.2 数据库表结构设计**

**（1）用户表（user）**

用户表由三个字段组成，loginId（登录账号），password（登录密码），ifOnline（在线标志），用户表user结构如表4.1 所示。

**表4.1费用数据表user**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| loginId | varchar(50) | 否 | 是 | 登录账号 |
| password | varchar(50) | 否 | 否 | 登录密码 |
| ifOnline | Bit | 否 | 否 | 在线标志 |

**（2）组表（group）**

组表由三个字段组成，id（自增主键），groupName（组名），desc（描述），组表结构如表4.2 所示。

**表4.2组表group**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| id | Int | 否 | 是 | 自增主键 |
| groupName | varchar(50) | 是 | 否 | 组名 |
| Desc | varchar(50) | 否 | 否 | 组权限描述 |

**（3）页面表（page）**

页面表由三个字段组成，id（自增主键），pageFunction（页面功能），pageName（页面名称），组表结构如表4.3 所示。

**表4.3组表page**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| id | Int | 否 | 是 | 自增主键 |
| pageFunction | varchar(50) | 是 | 否 | 页面功能 |
| pageName | varchar(50) | 否 | 否 | 页面名称 |

**（4）用户与组表（userWithGroup）**

用户与组表由三个字段组成，id（自增主键），userId（用户Id），groupId（组Id），组表结构如表4.4 所示。

**表4.4组表userwithgroup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| id | Int | 否 | 是 | 自增主键 |
| userId | varchar(50) | 是 | 否 | 用户id |
| groupId | Int | 否 | 否 | 组id |

**（5）页面与组表（pageWithGroup）**

页面与组表由三个字段组成，id（自增主键），pageId（页面Id），groupId（组Id），组表结构如表4.5 所示。

**表4.5组表pagewithgroup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| id | Int | 否 | 是 | 自增主键 |
| PageId | varchar(50) | 是 | 否 | 页面id |
| groupId | Int | 否 | 否 | 组id |

**（6）职员表（employee）**

职员表由六个字段组成，employeeId（自增主键），employeeName（职员姓名），department（部门），position（职位），gender（性别），birthday（生日）组表结构如表4.6 所示。

**表4.6职员表employee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| EmployeeId | varchar(50) | 否 | 是 | 职员id |
| employeeName | varchar(50) | 是 | 否 | 职员姓名 |
| Department | varchar(50) | 否 | 否 | 部门 |
| Position | varchar(50) | 是 | 否 | 职位 |
| Gender | varchar(50) | 是 | 否 | 性别 |
| Birthday | Date | 是 | 否 | 生日 |

**（7）职员用户关联表（employeeUser）**

职员用户关联表由三个字段组成，Id（自增主键），employeeId（职员id），userId（用户Id），职员用户关联表结构如表4.7 所示。

**表4.7职员用户关联表employeeUser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 自增主键 |
| employeeId | varchar(50) | 是 | 否 | 职员id |
| userId | varchar(50) | 否 | 否 | 用户Id |

**（8）职员薪资表（employeeWage）**

职员新资表由六个字段组成，Id（自增主键），employeeId（职员id），basicwage（基础薪资），performancewage（绩效工资），extrawage（额外），date（发薪日期）职员薪资表结构如表4.8 所示。

**表4.8职员用户关联表employeeWage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 自增主键 |
| employeeId | varchar(50) | 是 | 否 | 职员id |
| Basicwage | Double | 是 | 否 | 基础薪资 |
| Performancewage | Double | 是 | 否 | 绩效薪资 |
| Extrawage | Double | 是 | 否 | 额外 |
| Date | Date | 否 | 否 | 用户Id |

**（9）客户信息表（customer）**

客户信息表由九个字段组成，customerId（客户id），customername（客户名），email（邮件），address（地址），customertype（客户类型），linkman（联系人），

Linkmanmobile（联系人电话），businesslicense（机构号），idcard（身份证），客户信息表结构如表4.9 所示。

**表4.9客户信息表customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| CustomerId | varchar(50) | 否 | 是 | 客户id |
| Customername | varchar(50) | 是 | 否 | 客户名 |
| Email | varchar(50) | 是 | 否 | 邮件 |
| Address | varchar(50) | 是 | 否 | 地址 |
| Linkman | varchar(50) | 是 | 否 | 联系人 |
| Customertype | varchar(50) | 是 | 否 | 客户类型 |
| Linkmanmobile | varchar(50) | 是 | 否 | 联系人电话 |
| Businesslicense | varchar(50) | 是 | 否 | 机构号 |
| Idcard | varchar(50) | 是 | 否 | 身份证号码 |

**（10）货运单表（waybill）**

货运单表由27个字段组成，货运单表结构如表4.10 所示。

**表4.10货运单表waybill**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| BillId | varchar(50) | 否 | 是 | 货运单id |
| Sendid | varchar(50) | 是 | 否 | 发货客户id |
| Send | varchar(50) | 是 | 否 | 发货客户 |
| SendAddress | varchar(50) | 是 | 否 | 发货地址 |
| Sendphone | varchar(50) | 是 | 否 | 发货人电话 |
| Reciverid | varchar(50) | 是 | 否 | 收货客户id |
| Reciver | varchar(50) | 是 | 否 | 收获客户 |
| Reciveraddress | varchar(50) | 是 | 否 | 收货地址 |
| Reciverphone | varchar(50) | 是 | 否 | 收货人电话 |
| Paycustomer | varchar(50) | 是 | 否 | 支付客户 |
| Paytype | varchar(50) | 是 | 否 | 支付类型 |
| Billname | varchar(50) | 是 | 否 | 货物名 |
| Length | Double | 是 | 否 | 货物长 |
| Width | Double | 是 | 否 | 货物宽 |
| Height | Double | 是 | 否 | 货物高 |
| Weight | Double | 是 | 否 | 货物质量 |
| Number | Double | 是 | 否 | 货物数量 |
| Startdate | Date | 是 | 否 | 开始配送时间 |
| Exceptdate | Date | 是 | 否 | 预计到达时间 |
| Freight | Double | 是 | 否 | 运费 |
| Insurance | Double | 是 | 否 | 保险费 |
| State | varchar(50) | 是 | 否 | 货运单状态 |
| Isclear | Date | 是 | 否 | 结算标志 |
| Arrivedate | Date | 是 | 否 | 到达日期 |
| Desc | varchar(50) | 是 | 否 | 货物描述 |
| Err | varchar(50) | 是 | 否 | 差错信息 |
| Writedate | Date | 是 | 否 | 签发日期 |
| Cleardate | Date | 是 | 否 | 结算日期 |

**（11）司机表（driver）**

司机表由19个字段组成，司机表结构如表4.11 所示。

**表4.11司机表driver**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Driverid | varchar(50) | 否 | 是 | 司机id |
| Drivername | varchar(50) | 是 | 否 | 司机姓名 |
| Idcard | varchar(50) | 是 | 否 | 身份证编号 |
| Phone | varchar(50) | 是 | 否 | 电话 |
| Email | varchar(50) | 是 | 否 | 邮箱 |
| Gender | varchar(50) | 是 | 否 | 性别 |
| Birthday | varchar(50) | 是 | 否 | 生日 |
| address | varchar(50) | 是 | 否 | 地址 |
| Driverlicence | varchar(50) | 是 | 否 | 驾驶证 |
| Cartype | varchar(50) | 是 | 否 | 车类型 |
| Carno | varchar(50) | 是 | 否 | 车牌号 |
| Volume | varchar(50) | 是 | 否 | 载物空间 |
| Weight | Double | 是 | 否 | 货物长 |
| Length | Double | 是 | 否 | 货物宽 |
| Width | Double | 是 | 否 | 货物高 |
| Height | Double | 是 | 否 | 货物质量 |
| Remark | Double | 是 | 否 | 货物数量 |
| Number | Date | 是 | 否 | 被投诉次数 |
| Iscompany | Bit | 是 | 否 | 是否为公司车 |

**（12）运输合同表（carriage）**

运输合同表由25个字段组成，运输合同表结构如表4.12 所示。

**表4.12货运单表carriage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| CarriageId | varchar(50) | 否 | 是 | 运输合同号 |
| Billid | varchar(50) | 是 | 否 | 货运单号 |
| Billname | varchar(50) | 是 | 否 | 货物名称 |
| Driverid | varchar(50) | 是 | 否 | 司机编号 |
| Drivername | varchar(50) | 是 | 否 | 司机姓名 |
| Phone | varchar(50) | 是 | 否 | 司机电话 |
| Iscompany | Bit | 是 | 否 | 公司车标志 |
| truckFee | Double | 是 | 否 | 搬运费 |
| Freight | Double | 是 | 否 | 运费 |
| Insurance | Double | 是 | 否 | 保险费 |
| Cashpledge | Double | 是 | 否 | 押金 |
| Carfee | Double | 是 | 否 | 车辆费 |
| State | varchar(50) | 是 | 否 | 状态 |
| writeDate | Date | 是 | 否 | 登记日期 |
| Isclear | Bit | 是 | 否 | 结算标志 |
| Send | varchar(50) | 是 | 否 | 发货人 |
| sendAdress | varchar(50) | 是 | 否 | 发货人地址 |
| Sendphone | varchar(50) | 是 | 否 | 发货人电话 |
| Reciver | varchar(50) | 是 | 否 | 收货人 |
| reciverAdress | varchar(50) | 是 | 否 | 收货人地址 |
| reciverPhone | varchar(50) | 是 | 否 | 收货人电话 |
| Startdate | Date | 是 | 否 | 开始日期 |
| Exceptdate | Date | 是 | 否 | 期望到达日期 |
| Arrivedate | Date | 是 | 否 | 到达日期 |
| paytype | varchar(50) | 是 | 否 | 支付方式 |

**（13）客户结算表（customerClear）**

客户结算表由9个字段组成，客户结算表结构如表4.13 所示。

**表4.13客户结算表customerClear**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | varchar(50) | 否 | 是 | 编号 |
| Customerid | varchar(50) | 是 | 否 | 客户编号 |
| waybillId | varchar(50) | 是 | 否 | 货运单编号 |
| Freight | varchar(50) | 是 | 否 | 运费 |
| Insurance | varchar(50) | 是 | 否 | 保险费 |
| Money | Double | 是 | 否 | 总费用 |
| Isclear | Bit | 是 | 否 | 结算标志 |
| Cleardate | Date | 是 | 否 | 结算日期 |
| Writedate | Date | 是 | 否 | 登记日期 |

**（14）司机结算表（driverClear）**

司机结算表由13个字段组成，司机结算表结构如表4.14 所示。

**表4.14司机结算表driverClear**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | varchar(50) | 否 | 是 | 编号 |
| Driverid | varchar(50) | 是 | 否 | 客户编号 |
| CarriageId | varchar(50) | 是 | 否 | 货运单编号 |
| Freight | Double | 是 | 否 | 运费 |
| Insurance | Double | 是 | 否 | 保险费 |
| truckFee | Double | 是 | 否 | 搬运费 |
| Cashpleaget | Double | 是 | 否 | 押金 |
| carFee | Double | 是 | 否 | 车辆费 |
| Err | Double | 是 | 否 | 扣款 |
| Money | Double | 是 | 否 | 总费用 |
| Isclear | Bit | 是 | 否 | 结算标志 |
| Cleardate | Date | 是 | 否 | 结算日期 |
| Writedate | Date | 是 | 否 | 登记日期 |

**（15）杂费结算表（extraClear）**

杂费结算表由6个字段组成，杂费结算表结构如表4.15 所示。

**表4.15杂费结算表extraClear**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| SubjectName | varchar(50) | 是 | 否 | 科目名 |
| Money | Double | 是 | 否 | 费用 |
| Type | varchar(50) | 是 | 否 | 类型 |
| Remark | varchar(50) | 是 | 否 | 详细描述 |
| Writedate | Date | 是 | 否 | 登记日期 |

**（16）城市表（city）**

城市表由2个字段组成，城市表结构如表4.15 所示。

**表4.16城市表city**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| City | varchar(50) | 是 | 否 | 城市 |

**（17）城市关联表（citylink）**

城市关联表由3个字段组成，城市关联表结构如表4.17 所示。

**表4.17城市关联表citylink**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| Cityid | Int | 否 | 否 | 城市 |
| Linkcity | Int | 否 | 否 | 可达城市 |

**（18）路线表（cityroute）**

路线表由6个字段组成，路线表结构如表4.18 所示。

**表4.18路线表cityroute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| startStation | varchar(50) | 否 | 否 | 起点 |
| endStation | varchar(50) | 否 | 否 | 终点 |
| passStation | varchar(50) | 是 | 否 | 中途站点 |
| Distance | Double | 是 | 否 | 距离 |
| Fetchtime | Date | 是 | 否 | 时间 |

**（19）货物路线表（billroute）**

货物路线表由7个字段组成，货物路线表结构如表4.19 所示。

**表4.19路线表billroute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| billId | varchar(50) | 否 | 否 | 货运单编号 |
| start | varchar(50) | 否 | 否 | 起点 |
| end | varchar(50) | 否 | 否 | 终点 |
| pass | varchar(50) | 是 | 否 | 中途站点 |
| Now | varchar(50) | 是 | 否 | 目前位置 |
| Signdate | Date | 是 | 否 | 登记时间 |

**（20）投诉表（complont）**

投诉表由5个字段组成，投诉表结构如表4.20 所示。

**表4.20投诉表complant**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许空 | 是否主键 | 备注 |
| Id | Int | 否 | 是 | 编号 |
| billId | varchar(50) | 否 | 否 | 货运单编号 |
| driverId | varchar(50) | 否 | 否 | 司机编号 |
| Desc | varchar(50) | 否 | 否 | 描述 |
| Money | varchar(50) | 是 | 否 | 损失 |

**第五章 系统详细设计**

**5.1 概述**

详细设计继第四章后，将根据每个设计每个表的情况具体化设计过程。

本章的内容包含各个模块的详细设计与实现。

**5.2 登录模块的设计与实现**

用户通过输入用户名和用户密码进行登录验证，当验证通过时，将根据用户所属用户组拥有的权限展示对应的页面。如果输入密码错误则需要重新输入。登录界面如图5.1 所示。



**图5.1 登录界面图**

登录模块的设计重点不在于界面而在于登录是权限的控制，需要根据使用者输入的用户名去找对应的用户，然后去找对应用户所属的组拥有的页面权限，展示拥有权限的界面。因此权限的控制是在前端的js上实现的。部分js代码如下

**let *login*** = ***$***.cookie(**"loginId"**);  
**let *funArray*** = [];  
***$***.ajax({  
 **type**: **'get'**,  
 **url**: ***nginx\_url*** + **'/selectPage/'** + ***login***,  
 **async**: **false**,  
 **dataType**: **'json'**,  
 success: **function** (result) {  
 ***funArray*** = [];  
 ***$***.each(result, **function** (i, item) {  
 ***funArray***.push(item.pageId);  
 })  
 }  
});  
  
**for** (**let *i*** = 1; ***i*** <= 9; ***i***++) {  
 **if** (***$***.inArray(***i***, ***funArray***) == -1) {  
 *$*(**"#function\_"** + ***i***).remove();  
 }  
}

**5.2 人员管理模块的设计与实现**