**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

**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：

[1] 刘景良．我国物流企业发展现状及对策[J]．科学技术与工程，2006，6(20)[2] 邱圣鑫.企业物流供应管理系统的设计与实现[D].山东大学硕士学位论文，2005.03

[3]赵启兰.企业物流管理[M].北京:机械工业出版社，2011

[4]李伟辉.一汽国际物流中心物流管理系统改进方案研究[D].吉林大学硕士学位论文，2010

4 设计（论文）应完成的主要内容：

物流管理系统作为一个管理系统应该完成人员的管理，货物的管理，路线的管理，单据的管理，以及财务管理等功能，要控制用户权限，要能够修改密码。

5 提交设计（论文）形式（设计说明与图纸或论文等）及要求：

（1）纸质档毕业论文。

（2）刻录光盘一张，光盘内包含：开题报告、论文（电子档）、查重论文部分、查重报告、答辩演示PPT、代码。

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。

本系统综合考虑各方面的需求，确定了系统所需模块，设计了各个模块的功能，并结合目前流行的技术与框架，实现了这个基于Java的物流管理系统。系统主要实现了公司、客户、司机、货物一体化的管理。企业端主要功能包括客户、司机、员工的管理，货运单与运输合同的签发与结算，运输路线的管理，此外还有财务模块，统计公司的开支情况。司机端主要实现了货物运输路线的查看，以及对于货物的一些操作。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。此外还考虑了系统的非功能性需求，包括系统的易用性、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。

本系统使用B/S架构，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，面向接口编程增强程序可读性减少耦合，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。

使用上述技术实现了系统各个模块的功能。经过测试，该系统运行稳定，安全，性能优良，能够提高物流效率，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；Spring；框架

**ABSTRACT**

Under the current national conditions of rapid development of various industries in China, especially the Internet industry is developing more rapidly, traditional logistics management is difficult to meet the current needs, not only time-consuming and laborious, but also not efficient. With the rapid growth of the demand for logistics in all walks of life, the requirements for logistics efficiency are also increasing.

This system comprehensively considers the needs of all aspects, determines the required modules of the system, designs the functions of each module, and combines the current popular technology and framework to realize this logistics management system based on Java. The system mainly realizes the integrated management of the company, customers, drivers and goods. The main functions of the enterprise include the management of customers, drivers and employees, the issuance and settlement of freight orders and transportation contracts, the management of transportation routes, and in addition the financial module to count the company's expenditure. The driver side mainly realizes the view of the cargo transportation route and some operations on the cargo. The client is mainly to check the status of its own waybill, the transportation route of the goods, and the current position of the goods. In addition, the non-functional requirements of the system are also considered, including the ease of use, availability, security, reliability, and ease of maintenance of the system.

This article introduces the realization principle, operation mechanism and operation mode of the whole system from the technical level, data level and system level.

This system uses B / S architecture, Spring Boot as the main back-end framework, integrates Spring, Spring MVC, and Spring Data JPA framework, simplifies the back-end code with lombok, enhances program readability for interface programming, reduces coupling, and is easy to maintain. The toolkit encrypts and decrypts part of the data. The front-end uses layui admin to beautify the interface, and uses the native front-end language jQuery and Ajax for page control and front-end interaction. The database uses MySQL database.

Using the above technology has realized the function of each module of the system. After testing, the system is stable, safe, and has excellent performance. It can improve logistics efficiency, reduce corporate expenses, increase revenue, facilitate personnel management, and manage freight transportation. It is convenient for customers and drivers to communicate and understand cargo information in time. Basically achieve the purpose of this system research and design.

**Keywords:** Logistics management; Modularity; Java; Spring; Framework

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 2](#_Toc515387893)

[第二章 相关工具和技术介绍 5](#_Toc515387894)

[2.1 Java平台 5](#_Toc515387895)

[2.2 Spring系列框架 5](#_Toc515387898)

[2.3 前端相关技术 6](#_Toc515387901)

[2.4 数据库 7](#_Toc515387904)

[2.5 开发工具 7](#_Toc515387904)

[第三章 需求分析 9](#_Toc515387909)

[3.1 概述 9](#_Toc515387910)

[3.2 需求分析与非功能性需求 9](#_Toc515387911)

[3.3 可行性分析 10](#_Toc515387914)

[第四章 总体设计 17](#_Toc515387921)

[4.1 设计模式 13](#_Toc515387922)

[4.2 功能设计 13](#_Toc515387923)

[4.2.1 系统模块设计 13](#_Toc515387927)

[4.2.2 模块具体功能描述 14](#_Toc515387927)

[4.3 数据库设计 16](#_Toc515387924)

[4.3.1 数据库设计原则与实体图 16](#_Toc515387927)

[4.3.2 数据库表结构设计 1](#_Toc515387927)6

[第五章 详细设计 2](#_Toc515387936)5

[5.1 概述 2](#_Toc515387937)5

[5.2 登录模块的设计与实现 2](#_Toc515387938)5

[5.3 人员管理模块的设计与实现 2](#_Toc515387939)6

[5.4 货运单管理模块的设计与实现 2](#_Toc515387937)7

[5.5 运输合同管理模块的设计与实现 2](#_Toc515387938)8

[5.6 路线管理模块的设计与实现 30](#_Toc515387939)

[5.7 财务管理模块的设计与实现 31](#_Toc515387937)

[5.8 权限控制模块的设计与实现 32](#_Toc515387938)

[5.9 密码修改模块的设计与实现 33](#_Toc515387939)

[5.10 司机模块的设计与实现 33](#_Toc515387937)

[5.11 客户模块的设计与实现 33](#_Toc515387938)

[第六章 系统实现 35](#_Toc515387936)

[5.1 项目后端工程 35](#_Toc515387937)

[5.2 项目前端工程 38](#_Toc515387938)

[第七章 总结与展望 41](#_Toc515387941)

[参 考 文 献 43](#_Toc515387942)

[致 谢 45](#_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，吸引客户下单。

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。本系统着重于物流管理系统中的管理和物流，主要实现了对于货物运输的管理，对人员的管理，对路线的规划。本系统是个web应用，在移动平台和电脑上都能访问。本系统能够统一的管理各类人员，实时了解货物情况，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，同时方便客户、司机、企业信息的及时传递。

**1.2 课题研究的背景**

**（1）市场背景**

在这个互联网飞速发展的时代，各行各业都开始朝着数字化和信息化转型，物流行业也是如此。物流行业早已不适用以前那套手工操作的模式了。为了适应时代的潮流，物流管理系统是物流企业必要的也是急需的。一个优秀的物流管理系统能够很容易的打开市场的大门，因为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。

物流行业成为十大振兴产业在2009年，物流行业信息化的程度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。这就为本系统提供了市场。

1. **社会背景**

伴随着网络的普及，苏宁、淘宝等网上购物平台的兴起，大大刺激了人们的消费欲望。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，人员管理模块，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，最终实现的是一个高可用，易维护稳定的系统。

**1.4 物流管理系统的现状**

现在的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。但是国内起步较晚，市场上这种系统数量虽然众多，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给后来的开发者。同时市场上此类产品质量参差不齐，大部分系统目前还存在着产品功能单一、技术落后等问题，系统性能有待提高。

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巨大的物流市场。

**1.5 本文的主要内容**

本文内容概括如下。

第二章，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。

第三章，系统需求分析

本章是本系统需求分析和总体设计的过程，是开发编程之前的重要步骤。阐述了预计要实现的目标，分别设计。

1. 系统总体设计

从数据库的层面详细介绍了数据库的设计，以及表的设计。描述了数据库设计原则，以及具体的设计。

第五章，系统详细设计

本章对系统每个模块每个功能进行详细说明。

第六章，系统实现

本章是系统成果的展示，对于各界面进行展示与测试。

第七章，总结与展望

本章是论文的总结，包括阐述开发设计过程中遇到过的各种问题，系统存在的不足，对大学生活的总结。

1. **系统使用工具和技术简介**

**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应用开发以及移动端开发，主要写应用的服务器端。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Java虚拟机、Java应用编程接口组成。Java应用编程接口在此基础上还提供了一个独立在操作系统之外的标准编程接口，包括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，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的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，大部分配置都是自动装载的，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，spring boot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，是一个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中的动态代理类使用较为复杂。

Spring MVC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。MVC模式是指将应⽤程序分为 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应用开发。核心组件包括Dispatcher Servlet、Handle、Handle Mapping、Handle Interceptor、Handle ExecutionChain、Handle Adapter、ModelAndVewl、ViewResolver。主要过程是客户端请求被 Disptacher Servlet 接收，根据 Handler Mapping 映射到 Handler。⽣成 Handler和 Handler Interceptor。Handler 和 Handler Interceptor以Handler ExecutionChain 的形式⼀并返回给Disptacher Servlet。Dispatch Servlet通过Handler Adapter 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Dispatcher Servlet。Dispatcher Servlet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。Spring Data JPA支持使用JPQL查询语言和原生SQL。JPA主要有以下几个接口，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。在写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，不然需要自己在Query注解中写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，目的是提供一种更加简便的JavaScript开发设计模式，减少前端脚本开发的代码量，优化HTML文档、事件处理、与Ajax的交互及动画设计。jQuery基本兼容各种主流浏览器，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，它不是一种编程语言，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是前端模块化框架，能够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，因为前端界面开发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是反向代理服务器，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 需求分析与非功能性需求**

**（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。

**（4）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。

**3.3 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**

本物流管理系统是一个信息管理平台，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MySqll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**

本系统选用的设计模式是流行的MVC（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，在改动界面时，如果功能不变，则不用修改业务代码。同时MVC模式也使得代码维护变得更加简单方便。

**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所示。



**图4.1 管理员管理模块图**

**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，到达的可以结算。

**（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，到达的可以结算。

**（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 数据库设计概要与实体图**

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。

ER模型，全称为实体联系模型、实体关系模型或实体联系模式图（ERD）（英语：Entity-relationship model）由美籍华裔计算机科学家[陈品山](https://baike.baidu.com/item/%E9%99%88%E5%93%81%E5%B1%B1" \t "https://baike.baidu.com/item/%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E5%9B%BE/_blank)发明，是[概念数据模型](https://baike.baidu.com/item/%E6%A6%82%E5%BF%B5%E6%95%B0%E6%8D%AE%E6%A8%A1%E5%9E%8B" \t "https://baike.baidu.com/item/%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E5%9B%BE/_blank)的高层描述所使用的[数据模型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%A8%A1%E5%9E%8B" \t "https://baike.baidu.com/item/%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E5%9B%BE/_blank)或[模式图](https://baike.baidu.com/item/%E6%A8%A1%E5%BC%8F%E5%9B%BE" \t "https://baike.baidu.com/item/%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E5%9B%BE/_blank)。[2]

**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用户表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 | 用户编号 |
| groupId | int | 否 | 否 | 组编号 |

**（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（页面编号），groupId（组编号），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 | 页面编号 |
| groupId | Int | 否 | 否 | 组编号 |

**（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 | 职员编号 |
| 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（职员编号），userId（用户编号），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 | 职员编号 |
| userId | varchar(50) | 否 | 否 | 用户编号 |

**（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职员薪资表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 | 职员编号 |
| basicwage | double | 是 | 否 | 基础薪资 |
| Performancewage | double | 是 | 否 | 绩效薪资 |
| extrawage | double | 是 | 否 | 额外奖金 |
| date | date | 否 | 否 | 用户编号 |

**（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 | 客户编号 |
| 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 | 货运单编号 |
| sendid | varchar(50) | 是 | 否 | 发货客户编号 |
| 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 | 收货客户编号 |
| 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 | varchar(50) | 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 | 司机编号 |
| 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 | int | 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 |
| arriageId | 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 | int | 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();  
 }  
}

此外还有就是输入密码后程序需要对密码使用加密算法，加密后与数据库中的密码进行比较，结果一致才能登陆成功。

**5.3 人员管理模块的设计与实现**

**（1）客户管理**

管理员登录后先选择人员管理，然后找到客户管理，点击对应导航栏，就可以对客户信息进行管理。客户信息添加时，系统会为客户生成唯一的客户编号，客户编号以KH开头，同时会用客户编号同步注册系统用户，并自动归属于客户组，生成的用户默认密码为用Md5算法加密过的123456。Md5算法使用的是Hutool的工具类。在进行客户添加时，部分信息进行了正则表达式过滤，比如邮箱就必须输入邮箱格式的字符串，部分信息是下拉框直接选择就行无需手动输入如图5.2 静态下拉框所示。点击客户查询界面，则将分页查询出全部客户信息，可以在后面的操作中，删除客户信息，查看客户信息，以及修改信息。修改信息时无法修改客户的编号。删除客户时会同步删除该系统用户。部分代码如下。

String id = ***kh***+System.*currentTimeMillis*();  
customer.setCustomerId(id);  
**try** {  
 **customerDao**.save(customer);  
 User user = **new** User();  
 String pwd = SecureUtil.*md5*(**"123456"**);  
 user.setPassword(pwd);  
 user.setLoginId(customer.getCustomerId());  
 **userDao**.save(user);



**图5.2 静态下拉框**

**（2）司机管理**

管理员登录后先选择人员管理，然后找到司机管理，点击对应导航栏，就可以对司机信息进行管理。司机信息管理和客户信息管理一致，不过司机会被分配到司机组，会分配司机的权限。在进行司机信息添加时，也同步添加了车辆信息，至于是不是公司车这一栏，则是因为，有些司机用的是公司的车，需要付车辆费。

司机的编号是以SJ开头。

**（3）员工管理**

管理员登录后先选择人员管理，然后找到员工管理，点击对应导航栏，就可以对员工的信息进行管理。在对员工的信息进行录入时，不会同步生成系统用户。

修改员工信息时，员工编号无法改变。在查询员工信息页面，选中具体的员工点击查看详情可以看到员工信息，选择修改可以修改员工信息，选择删除则会删除员工，因为删除操作不可逆，所以会弹出提示框，点击确定后才能删除。

**5.4 货运单管理模块的设计与实现**

货运单是公司与客户签订合同，因此合同信息比较多，出于安全性和便利性考虑，发货客户和收货客户都必须在系统中登记，同时支持选择支付客户和支付方式，客户与公司关于各项费用需要约定好。货运单添加时，需要填写好货物的基本信息，对于方便计件的货物，需要登记数量，同时也要登记货物总体的长宽高，体积重量等信息，不方便计件的货物数量直接填1。添加合同时，合同状态默认为待发，不可改变。这样做是为了便于管理。

货运单查询页面能够查询四种状态的货运单以及全部运单，分别是待发、未到、已到、已结算。对于待发的运单，右侧操作中有查看详情、修改、删除三个按钮，选择查看详情会展示货运单的详细信息，选择修改则会进入货运单修改界面，大部分信息时不能修改的，其中最主要的要更改货运单的状态，状态在这里进行了限制只能选择待发和未到达。同时登记日期会默认变成当天，可以修改。演示效果如图5.3 货运单修改信息演示。对于货运单中客户信息写错的则需要将货运单删除，重新去添加。如果修改后，发现没有合适的司机运输，状态仍然是未出合同则能再次修改。将货运单的状态改为未到，此时选择未到合同，则能查看到未到的货运单。此页面操作只有，查看详情，可以查看货运单的详细情况。

选择已到合同，可以查看全部已经到达的未结算的货运单，对于已到运单，操作有结算操作和查看详细。选择结算，则会打开结算页面。结算页面会展示货运单的基本情况并将状态变为已结算。点击结算按钮则会根据货运单信息生成对应的结算信息，插入到收支表中。此时货运单状态变为已结算。

选择已结算合同可以查看已结算的全部合同。此界面只有一个查看详情操作。选择全部合同则可以查看所有的货运单信息，点击详情能查看货运单的详细信息。



**图5.3** 货运单修改信息演示

**5.5 运输合同管理模块的设计与实现**

本模块是企业与司机签订运输合同，委派司机根据货运单上的货物信息去送货。

添加运输合同页面，首先是选择货运单编号，这是一个动态下拉，当打开这个页面就会从服务上查找所有状态是未到的运输合同。当点击事件产发生后，部分信息也会自动从货运单中读取，无法手动输入。部分代码如下。

**$**.ajax({  
 **type**: **"get"**,  
 **url**: ***nginx\_url*** + **"/carriage/findWayBillByState1"**,  
 **async**: **false**,  
 success: **function** (result) {  
 **$**.each(result, **function** (i, item) {  
 **let** option = **"<option value='"** + item + **"'>"**;  
 option += item;  
 option += **"</option>"**;  
 **$**(**"#billId"**).append(option);  
 form.render(**'select'**);  
 });  
 }  
});

form.on(**'select(changeSend)'**, **function** (data) {  
 *// ajax* **$**.ajax({  
 **type**: **'get'**,  
 **url**: ***nginx\_url*** + **'/carriage/findWayBillByBillId/'** + data.**value**,  
 success: **function** (result) {  
 **$**(**"#billName"**).val(result.**billName**);  
 **$**(**"#send"**).val(result.**send**);  
 **$**(**"#sendPhone"**).val(result.**sendPhone**);  
 **$**(**"#receive"**).val(result.**receive**);  
 **$**(**"#receivePhone"**).val(result.**receivePhone**);  
 **$**(**"#exceptDate"**).val(result.**exceptDate**);  
 }  
 });  
});

前者是动态下拉的实现，后者则动态下拉点击自动赋值的实现。同样运输合同也是不能选择的，是未出合同。

接着是运输合同查看页面，同样能显示四种状态的运输合同。分别是未出合同，未到，已到，已结算。还有全部运输合同的展示。对于未出合同状态的运输合同能够修改，删除。对于已到的运输合同能够进行结算，结算时可以在备注中说明一些运输情况的信息。本页面所有选项卡下的子页面都有查看详情的功能，点击后能够查看运输合同的详细信息。

**5.6 路线管理模块的设计与实现**

首先设计的是基础城市的添加页面，生成一条运输路线需要多个坐标，因此在添加路线前，需要先添加基础城市信息。基础城市信息添加比较简单，就是一个坐标名，然后添加到数据库中就可以了。

接着是关联城市，就是这个坐标能够比较方便到达的坐标。设计之初的目的是在选择中途站点时能够通过关联的坐标连接出一条路径。由于技术原因并未实现此功能。

然后是路线的添加，这个是个人感觉最不好做的，因为中途经过的站点数量无法确定，因此就只能将中途站点的信息转变为一个字符串，然后在将数据存储。页面上也不好展示，因为是动态的，无法确定需要展示多少个中途站点。只能将从后端服务上的数据进行分割然后拼接网页字符串展示在界面上。部分代码如下所示。

form.on(**'submit(addRoute)'**, **function** () {  
 **let** startStation = $(**"#startStation"**).val();  
 ***console***.log(**"startStation: "** + startStation);  
 ***console***.log(**"passStation: "** + ***pass\_station***.toString());  
 $.ajax({  
 **type**: **'post'**,  
 **url**: ***nginx\_url*** + **'/route/addCityRoute'**,  
 **data**: {  
 **'startStation'**: startStation,  
 **'passStation'**: ***pass\_station***.toString(),  
 **'endStation'**: $(**"#endStation"**).val,  
 **'distance'**: $(**"#distance"**).val(),  
 **'fetchTime'**: $(**"#fetchTime"**).val()  
 },

选择路线信息可以查看所有的路线信息。

最重要的是为货物选择运输路线，选择路线时，输入根据起点和终点查询路线，因为同样的起点和终点可以有不同的路线，因此起点和终点不饿能作为唯一标识，需要额外添加一个路线编号，其中还需要存储货运单的信息，还有运输合同的信息，和当前位置，以及修改时间。路线选择界面如图5.4 路线选择图所示。



**图5.4** 路线选择图

**5.7 财务管理模块的设计与实现**

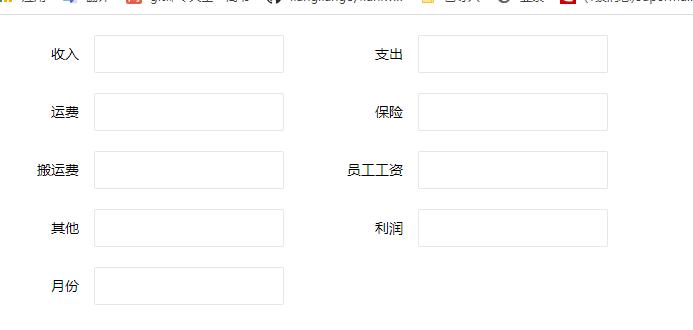
本模块的主要功能是统计当月收支情况。

首先是员工工资的录入，在报表时支出一项需要包括员工工资，因此需要录入员工的工资，员工工资也不是固定的，有基本薪资，绩效奖金和额外奖金组成。手动录入会比较方便。

此外还有特色的手工调账的功能，能对差错的收支进行调整，此外这一项能够支出杂费的收支情况，例如房租水电费等。

本月收支情况页面的数据并不是数据库中存储的，而是根据数据库中的数据计算出的。部分代码如下。String month = **""**+**new** Date().getMonth()+**new** Date().getYear();Income income = **new** Income();  
income.setCarriageFee(**incomeDao**.countcarriageFee(month));  
income.setIncome(**incomeDao**.counincome(month));  
income.setInsuranceFee(**incomeDao**.countinsuranceFee(month));

收支页面展示如图5.5 收支情况页面所示。



**图5.5** 收支情况页面

**5.8 权限控制模块的设计与实现**

本模块是专门控制用户权限的模块，主要做法是在数据库中定义用户表，组表，页面表，然后通过用户和组表，页面与组表将用户权限和页面展示关联起来。具体是将用户加入某一个组，然后页面与组表添加该组拥有的页面，每一条记录都是一个权限信息，都代表了某个组的用户拥有的使用某一个页面功能的权限。

本模块页面有添加用户组的页面，和查询用户组，同时也有权限更新的页面，选择对应的组，然后在界面上勾选，用户组所拥有的权限。如图5.6 用户组功能管理展示。



**图5.6** 用户组功能管理展示

**5.9 密码修改模块的设计与实现**

此模块分为两个，一个是管理员独有的可以修改所有用户密码的密码修改页面，还有一个是只能修改当前用户密码的页面。

只能修改当前用户密码的密码修改页面，只需要输入旧密码，和两次一样的新密码就可以，然后点击修改就行。

管理员独有的密码修改页面则需要选择用户，选择用户是一个动态下拉，不能随意输入。

其实这两者调用的是同一个服务，只是普通用户只展示修改自己密码的界面，用户编号是直接从浏览器的Cookie中获得的，而管理员的则是自己选择编号。

**5.10 司机模块的设计与实现**

司机模块中的页面是展示给司机看的，有运输合同查看模块，能够查看未到合同，已到合同，已结算合同和历史合同。司机模块下的这几个查询页面都只有查看详情的选项。这几个页面司机都只能看到自己接的货运单，这个也是借助浏览器的Cookie实现的。

司机能够查看路线信息，路线信息页面上分页展示所有的路线。

司机能够实时上报货物的位置，当司机到达目的地时能够将运输合同的状态改为已到。同时货运单的状态也会变为已到。当路线信息例如拥堵严重时，司机能够更改路线。

司机能够查看结算信息，包括未结算的单据和已结算的单据。点击详情则会跳转至收入详情页面。

**5.11 客户模块的设计与实现**

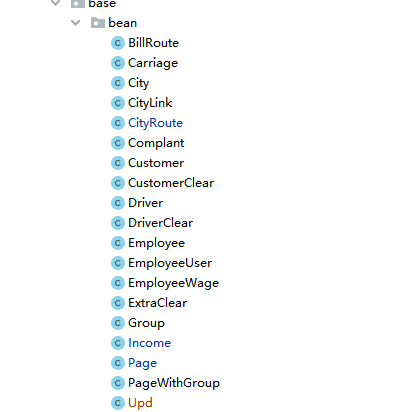
本模块专为司机设计，司机能够查看自己发出的货运单和收到的货运单，能够查看货物的路线信息，能够查看货物目前到达了什么位置。

客户能够对结算人是自己的已到达的运单进行结算。

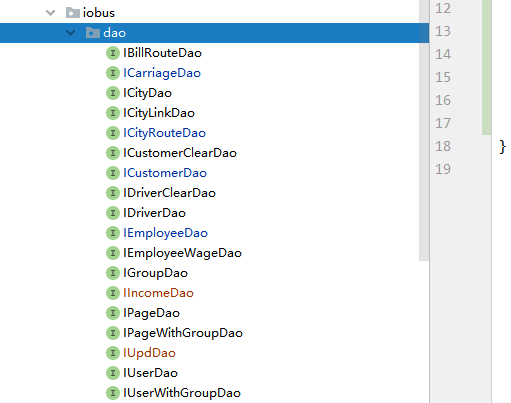
**第六章 系统实现**

**6.1 项目后端工程**

base包：里面有两个子包，一个是bean包，里面装的是JavaBean对象，需要添加@Bean注解标记为Bean，需要添加@Table注解标记这些实体类是哪些表的实体类，当对数据操作时，这些JavaBean对象就是表的记录。还有一个util包，里面定义了一些工具类。base包部分类情况如图6.1 base包部分类所示。

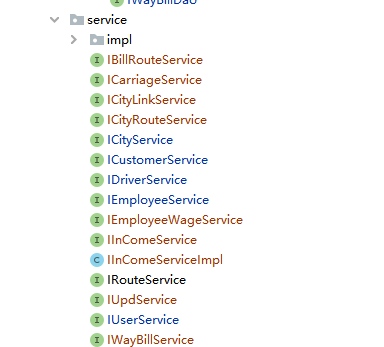


**图6.1** base包部分类

iobus包：里面放的是dao层对象，是直接操作数据库的类，base层的JavaBean对象当作这些对象的方法的参数。dao层的对象都继承了JpaRepository接口，可以直接调用一些常用的操作方法，不用自己写SQL。iobus包部分类如图6.2 iobus部分类所示。

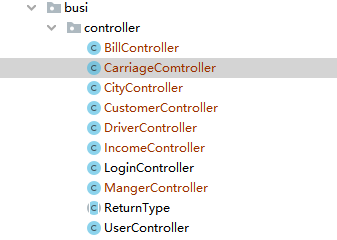
**图6.2** iobus包部分类

service包：里面放的是service接口和service的实现。在实现类中需要添加@Service注解才能自动注入，直接调用iobus里的dao对象操作数据。在进行修改、添加等操作时需要使用事务，要添加@Transactionl注解。部分类如图6.3 service包部分类所示。



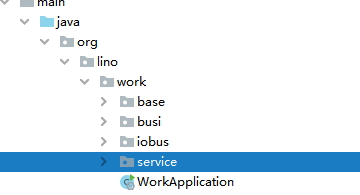
**图6.3** service包部分类

busi包：里面存放的是Controller对象，负责处理客户请求。里面调用的是service层的类和方法。在每个Controller中都需要使用注解标明请求的地址，部分类如图6.4 busi包部分类所示。



**图6.3** busi包部分类

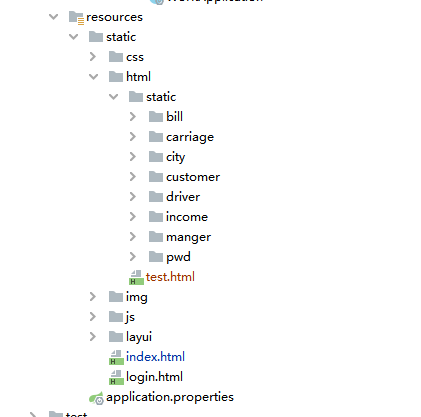
服务端包总体情况如图6.5 项目包图所示。其中的WorkApplication是springboot应用的启动类。



**图6.5** 项目包图

**6.2 项目前端工程**

整体结构如图6.6 前端目录结构所示。



**图6.6** 前端目录结构

css里面放的是css样式，html中存放的是前端界面，img是使用到的图片资源。js里面放的是JavaScript代码，layui是layui的库。

**第七章 总结与展望**

本系统对于笔者来说是一个很大的挑战。本系统使用的技术众多，不熟悉的需要一边翻文档一边开发。然后就是由于对物流的流程不清楚，导致设计的表结构改了多次，对于业务不太清楚只能慢慢摸索改进。缺乏经验，需求分析过于理想化，有些功能实现过于困难。

开发过程中问题最大的还是前端，对于界面设计不太在行，因此使用了模板化的框架Layui，但是还是遇到了许多问题。例如借助Ajax实现页面的动态刷新效果，就经常出错。对于下拉等需要拼接html字符串的，经常拼错。在使用Nginx做反向代理时也遇到了坑。还有就是前端跨域问题，因为没怎么遇到过，所以专门花时间去解决了这个问题。最最重要的是用户权限的控制，走了不少弯路。

最终花了几个月的时间，通过查资料，问同学问朋友，求助老师，最终实现了这个系统，第一次做工作量这么大的工程，当系统实现的时候是真的很开心。

本系统不尽人意的地方就是对于，货物运输路线的管理，并没有实时获得路况信息，而是手动录入的路线信息，而且写的路线信息过于宽泛。

**参 考 文 献**

[1] 廖圣清,易红发.大学生APP使用状况调查——基于上海的实证研究[J].暨南学报(哲学社会科学版),2016,3:86

[2] 张亚运.Android开发入门百战经典[M].北京:清华大学出版社,2017:1

[3] 韩文智,骆文亮.Android平台的移动APP开发方法与应用研究[J].四川理工学院学报(自然科学版),2015,28(3):23

[4] 王卓,包杰.嵌入式Linux系统及其应用前景[J].单片机与嵌入式系统应用,2004,5:5

[5] 高建,宋德强.使用Fragment来增强Android APP性能和用户体验[J].民营科技,2016,8:74

[6] 刘丽丽,高丽华.Java改变了什么[N].计算机世界,2005/09/19(A18)

[7] 董纪英,燕正伟,梁正玉.SQLite、MySQL、PostgreSQL关系型数据库管理系统比较[J].电脑编程技巧与维护,2014,14:55

[8] 张海藩,牟永敏.软件工程导论[M].北京:清华大学出版社,2013:55

[9] 施伯乐.数据库系统教程[M].北京:高等教育出版社,2008:17

[10] 朱少民.软件测试方法和技术[M].北京:清华大学出版社,2014:7

**致 谢**

从去年开始选题到现在完成系统设计与论文，中间发生了许多的事情。期间也更换过毕业设计题目。遇到过各种各样的困难，但在解决困难的过程中，自己也在逐渐的成长，这种慢慢变强的感觉真是太舒服了。

大学四年即将结束，我十分感谢我班上的同学，感谢他们四年的陪伴，感谢他们帮我做的这种事那种事。感谢那些曾经教过我的老师，是他们教会了我许多知识，感谢他们的付出。我十分感谢我的班主任刘桂开老师，每次都能提醒我们这个时间段该做什么。我也十分感谢我的毕业设计指导老师李志刚老师，他非常关心学生，学校一有什么事情马上就能通知大家，遇到问题时能够及时帮助笔者解决。在李老师的帮助下，笔者顺利完成了毕业设计与论文。