

## 摘 要

当前 4S 店的维修、保养系统模式老旧，效率低，给社会造成了极大的经济损失，因此，急需更加科学便捷的管理模式使信息管理更加准确、方便、丰富，提高工作效率，该系统使用 JAVA 作为开发语言，设计采用了 B/S 结构，并使用 JSP 技术设计了动态页面，使用 MYSQL 数据库进行存储信息，采用 SSM 框架开发了一个汽车 4S 店的维修保养管理系统，汽车维修保养管理系统的主要用户分为管理员和用户以及维修员，管理员有用户管理、个人中心、汽车信息管理、维修管理、订单管理、历史订单管理、消息推送管理这些模块，用户则有主页，汽车信息，维修、保养服务预约，个人中心，订单管理功能，维修员则主要拥有个人中心和保养预约服务管理两个功能模块，通过对这些功能模块的设计，系统对当前 4S 店的经营模式极为吻合，能够大大提高汽车维修、保养管理效率。

**关键词：**服务管理系统；JSP 技术；MYSQL 数据库；B/S 结构

## ABSTRACT

With the arrival of the 21st century, the development of networked information management is becoming faster and faster, and people's understanding of informatization is also becoming richer. Therefore, the importance of information management is gradually increasing. A more scientific and convenient management model makes information management more accurate, convenient, rich, and efficient. The system uses JSP as a development technology to implement an automotive service management system, The main users of the automotive service management system are divided into administrators and users. Administrators have modules such as user management, personal center, automotive information management, repair management, order management, historical order management, and front desk homepage. Users have functions such as homepage, automotive information, repair and maintenance service appointment, personal center, and order management. Through the design of these functional modules, the process of the automotive service management system has been basically realized, The design of the system adopts a B/S structure and uses JSP to design dynamic pages. The backend uses SQL database to make it an excellent service management system.

**Keywords:** service management system; JSP technology; MySQL database; b/s structure

# 目 录

第 1 章 绪论 .....	1
1.1 课题背景及意义 .....	1
1.2 国内外研究现状 .....	1
1.3 本课题的主要工作 .....	3
第 2 章 系统开发环境 .....	4
2.1 java 简介 .....	4
2.2 Mysql 数据库简介 .....	4
2.3 JSP 技术介绍 .....	5
2.4 SSM 框架介绍 .....	5
第 3 章 系统分析 .....	6
3.1 可行性分析 .....	6
3.1.1 技术可行性 .....	6
3.1.2 操作可行性 .....	6
3.1.3 经济可行性 .....	6
3.1.4 法律可行性 .....	6
3.2 系统流程分析 .....	7
3.2.1 系统操作流程 .....	7
3.2.2 添加信息流程 .....	7
3.2.3 修改信息流程 .....	8
3.2.4 删除信息流程 .....	9
第 4 章 系统设计 .....	11
4.1 系统概述 .....	11
4.2 功能模块设计 .....	11
4.3 数据库设计 .....	12
4.3.1 数据库设计原则 .....	12
4.3.2 数据库实体 .....	12
4.3.3 数据库表设计 .....	15
第 5 章 系统实现 .....	21
5.1 登录注册功能模块 .....	21
5.1.1 注册模块 .....	21
5.1.2 登录模块 .....	23

5.2 用户功能模块.....	27
5.2.1 主页 .....	27

5.2.2 个人信息管理模块	28
5.2.3 车辆信息管理模块	29
5.2.4 维修保养预约模块	32
5.3 管理员功能模块	35
5.3.1 信息推送管理模块	35
5.3.2 用户信息管理模块	36
5.3.3 车辆信息查询模块	37
5.3.4 订单信息管理模块	38
5.3.5 维修配件管理模块	45
5.4 维修员功能模块	47
5.4.1 保养维修管理模块	47
<b>第6章 系统测试</b>	<b>51</b>
6.1 系统测试的意义	51
6.2 测试方法	51
6.2.1 注册测试用例	52
6.2.2 车辆信息添加测试用例	52
6.2.3 订单支付测试用例	52
6.3 测试分析	53
<b>结论</b>	<b>54</b>
<b>参考文献</b>	<b>55</b>
<b>致谢</b>	<b>56</b>

---

# 第 1 章 绪论

## 1.1 课题背景及意义

随着网络发展的越发先进，如今全世界的信息管理方式都很少采用纸质的方式了，都在向网络化转变，自从上世纪的九十年代起，我国各个企业甚至政府部门也开始采用网络系统来进行信息管理了。以前很少有地方用计算机管理系统，并且存在着诸多技术问题，这导致其发展缓慢，但是，随着科技的提升，各种功能层出不穷，大多数企业甚至政府部门都将管理信息与互联网系统紧密集成。

那么为什么要采用计算机来管理信息呢？其中有很多优点，比如纸质管理的成本低，保密程度更加安全，且能够存储的信息量要比纸质大得多，即使纸质等方式能够存储，也没有计算机查询更加快速、便捷，如此看，使用计算机管理的方式确实更加优秀，毕竟以往的存储方式弊端太多了，也没有权限设置来划分访问级别。

现如今，汽车的数量越来越多了，开车出行已经变成了常态化，汽车的销量猛猛上涨，可是后勤的维修、保养等服务却跟不上时代的发展。首先，现在人民群众对汽车相关的保养意识淡薄，这使得尽管社会上有着大量的汽车，但是因为损耗严重而造成了极大的经济损失。所以，决定开发一款基于 SSM 框架的汽车信息管理系统。

## 1.2 国内外研究现状

尽管我国在相关领域上取得的成就很大，但我国对相关系统的研究仍然落后于其他的先进国家。一系列相关的研究为现在的 4S 店管理软件的形成奠定了良好的基础，目前国内许多调查研究将客户与企业的关系分为三个核心：建立客户与企业的关系、维系客户与企业的关系、扫一扫关注微信号恢复客户与企业的关系，这个理念为如何做好一个 4S 店管理软件指明方向，要想做好 4S 店的管理必须把这些理念联系实际，在实际应用中体现出来。

2020 年，马芯在《基于微信平台的 4S 店智能管理系统的设计与实现》实现了一款微信平台上的 4S 店信息管理系统，他使用了移动应用开发技术、手机地图服务和 HTML 前端开发等技术，开发出了一套在微信平台上运行的汽车 4s 店的智能信息管理系统，这个系统有分为两个功能模块，分别是微信客户端和后台管理端两个部分。这个系统使用了 java 语言，还有一系列的前端技术如 HTML、CSS、JavaScript 和 Thymeleaf 等网页技术和后端的 SpringBoot 框架，使用这些技术实现了客户端的预约、个人中心、商家介绍、导航、联系模块、预约管理和紧急求助 7 个功能菜单。后台管理端完成了

---

4S 首页、预约中心、客户管理、财务结算和车品详情模块设计,用于管理员注册登录后对线上线下订单的管理和对汽车 4S 店数据信息的管理。其次,还使用了图形技术实现了一种识别车牌来认证身份的接口,这个功能能够通过识别车牌来进行车主信息匹配,完成了 VIP 会员车辆能够快速进入 4S 店和缴费等快速离店功能。经实验室功能测试和 4S 店实体店试运行,结果表明,系统的客户端和 4S 店管理后台均能达到预期功能效果且运行稳定。还有使用微信号实现汽车 4S 店的各种预约服务,这使得软件安装更加方便,PC 端的使用更加便捷,线上下相互结合,使用销售服务相分离的模式来营销,提高了 4S 店的汽车服务水平<sup>[1]</sup>。

2015 年,赵文静发表了《汽车 4S 店客户关系管理系统的设计与实现》,该文阐述了 4S 店是一种以“四位一体”为核心的汽车特许经营模式,包括整车销售(Sale)、零配件(Sparepart)、售后服务(Service)、信息反馈(Survey),它是一种个性突出的有形市场,具有渠道一致性和统一的文化理念,按照生产厂家的统一店内外设计要求建造,4S 店在提升汽车品牌、汽车生产企业形象上的优势是显而易见的。基于 SSH 和 MVC 框架为汽车 4S 商店开发管理系统,该系统允许存储和管理与汽车 4S 商店相关的大量数据信息,不同数据的分类、查询和统计,以及用户友好的用户界面。通过汽车 4S 商店的客户关系管理系统,提供功能性服务,与客户建立并保持一套有效的“一对一关系”,并伴随售前、售中和售后服务的三个阶段,以提高客户满意度,吸引和留住更多客户,再根据客户反馈及时分析市场需求,能够快速调整其产业结构和营销策略,以确保其竞争优势<sup>[2]</sup>。

2013 年,丁一在《汽车 4S 店服务管理系统的设计与实现》讲到国内汽车产业发展日新月异,同时竞争也非常残酷,大部分汽车配套服务行业都属于中小企业,信息化水平不高,行业之间的交流也多数局限于电话和传真等传统方式,造成了在日常生产,销售,以及服务中的一些难以避免的问题,导致企业生产效率不高,缺乏在信息网络时代的竞争能力。于汽车服务企业本身资金缺乏,信息化人才水平不高,而使用 ASP 技术架构信息管理系统,可以让企业客户通过支付较少的费用就能得到比较全面的信息化服务,因此应用 ASP 提供强大技术支持能快速开发出相应的网络应用系统,有效地满足当前汽车 4S 店服务的信息化要求。因此,汽车 4S 店服务管理系统应运而生。该系统的宗旨是实现服务商之间能够快速、准确的进行通信,并且对汽车服务方面相关的综合服务水平,对我国汽车服务业的现状和发展趋势进行了研究和探讨。研究了汽车 4S 店服务管理系统的应用与开发,分析并明确了汽车 4S 店里整体服务管理系统要求。最后设计了各种业务流程和系统的整体结构,完成了各个模块功能、数据库以及系统安全问题的详细设计,并加以实现;最后总结了汽车 4S 店服务管理系统的优缺点<sup>[3]</sup>。

接下来谈一谈马丁·福勒,在计算机方面,他是一位著名的作者和国际知名演说家,他在 2018 年的《Refactoring:Improving the Design of Existing Code,Addison-Wesley Professional》中提到重构概念,一言以蔽之,就是在不改变外部行为的前提下,有条不紊地改善代码。他的这个观点推动了管理系统的发展。国外向来不缺乏计算机相关人才,所以

---

其在汽车管理系统方面也很先进，同样有巨大贡献的还有厄马与在 2019 年编著的《Mycroft. Java 8 in Action: Lambdas, Streams, and Functional-Style

---

Programming》，通过新的 Stream API 及 Lambda 表达式等示例全面讲解 Java 8 新特性并为 Java 程序员开启函数式编程的大门。

国外信息管理系统的研究是相当先进的，因为国外相对中国来说，很早就开始了对信息管理系统的开发与研究。有很多的例子，就像 4s 店管理系统出现的初步是在上个世纪的八十年代左右，他们就已经开始了对客户信息、关系等的管理就行电子数据化管理。也因为计算机管理信息简便、安全等的优越性，服务管理系统就诞生了，并且在日积月累的开发中逐渐发展起来，如今已经不算是新生儿了，而是社会企业链上的一条至关紧要的纽带，征战在社会的信息管理难题上，这不仅使客户的满意度有所增长，并且随着电子商务的逐步崛起，出现了方方面面的网络经济结合的管理系统。

### 1.3 本课题的主要工作

现在，JAVA 语言已经成长的相当完善了，在各个领域的使用比重很大，因此开发语言选择了 JAVA，数据库使用 MYSQL，前端采用 JSP 编写网页，使用 JS 的功能可以实现各种各样的动态样式。开发框架采用 SSM 框架来简化系统的开发，上面所谈内容都是比较基础的部分，在下面会进行系统分析，然后更具分析数据再逐步进行系统设计，最后进行系统的开发与测试。

---

## 第 2 章 系统开发环境

### 2.1 JAVA 简介

首先就是介绍一下众所周知的 JAVA 了，如广大人民群众所知，JAVA 是一门使用起来及其便捷、迅速的软件，相较于 C++ 来说，开发难度下降了一整个等级，这使得尽管一个人做一套管理系统，耗费的精力也得到了大大的缩减。

这与 JAVA 的编程方式密不可分，首先要提到的就是 JAVA 是一门面向对象的语言，可能不了解的人会有点疑问，但是解释起来也非常方便，这里采用一个对比的手法来描述，请出老牌 ACE 编程语言 C++。JAVA 相对 C++ 来说，更加注重的是处理问题的人，也是处理问题的对象是谁，而不在乎如何去解决这个问题，就好比你去了一家餐馆，你关注的是谁给你做的饭，而不关注他是怎么做的饭，而 C++ 相反，C++ 更加注重做饭的过程而不是做饭的人，也就是做饭的结果。所以，这使得 JAVA 的使用更加方便。

而且采用 C++ 开发的话，会涉及到一系列复杂、难理解的概念比如指针和多继承等等，这些错综复杂的知识对程序开发是不友好的，而易用性更强的 JAVA 抛弃了这些概念，使程序员在编程的过程中更加游刃有余

### 2.2 MYSQL 数据库简介

接下来就该介绍一下本系统使用的数据库了。虽然说 MYSQL 的处理速度不是很快，毕竟是一个关系型数据库，更加注重实体间的关系，拥有者清晰的成员之间的体系，而且要开发的项目也并不是说像京东、淘宝那些大型应用，需要考虑的并发量千万甚至上亿。所以实际开发也并没有太困难，不需要配置集群，也不需要其他缓存，一个 MYSQL，几乎可以解决所有小应用的持久化问题，既然可以满足需求，又十分还用，还有什么理由不选择呢？

还有一点要提，就是为什么 MYSQL 数据库是必须的，因为要做一个信息管理系统，更加注重的是增加、修改、删除、查询等操作，也就是 CURD。MYSQL 又能完美完成这些操作，且使用 MYSQL，实现事务的原子性、一致性、持久性和隔离性也变得更加方便了。因为 MYSQL 支持查询语言，可以使用标准的 SQL 语法。

后面还有就是安全性和功能性，MYSQL 拥有灵活且安全性高的权限密码系统，而且支持多线程，合理分配 CPU 资源，还有 TCP/IP 和 JDBC 等多样化的连接方式，项目中使用的是 Alibaba 的 Druid 实现的。Druid 使用起来也是极其方便的，只需要一个 Properties 文件，就能写入所有连接所需的配置，再用反射机制加载即可。

---

## 2.3 JSP 技术介绍

JSP 技术是十分强大且好用的，无论是 JSTL 还是 JQuery，在结合这些技术后，可以达到各种各样的效果，因为能够做出各种令人眼前一亮的动态效果，比如使用 Javascript 脚本在前端做一些输入的格式校验，比如给一个密码输入框设定一个失去焦点事件，一旦鼠标离开输入框后，这个事件就会运行，开始检查密码的格式，就比如密码的长度不能小于 6，当用户输入的密码长度不符合时，再动态引入 Bootstrap 技术，弹出一个美观的弹出框，提示用户长度不符合要求，当然，既然提到了 JQuery，可以使用 JQuery 整合的 Ajax 来发送异步请求在后端进行验证，将用户输入的数据异步发送到后端，用后端接口验证再返回是否合理，如果用原生的 ajax 请求是十分麻烦的，毕竟要是去使用 XmlHttpRequest 的 Ajax 要写大量的代码。

而且实际上在前端也确实使用了这些技术，用 JSTL 进行集合的循环遍历，使用 JQuery 的异步请求增加用户体验，因为一刷新要刷新整个页面是让人很不舒服的，而且耗费流量。虽然现在前卫的技术是相当多的，但仍然有 JSP 的一席之地。

## 2.4 SSM 框架介绍

SSM 框架是一套方便开发项目的一套脚手架，能够大幅度的减少开发压力。说到 SSM 框架，就不得不提 SSM 的 MVC 模式，MVC 就是 M 的 Model 数据层，V 的 View 层和 C 的 Controller 层，其中 Model 层分为两个部分，分别是 Dao 层和 Service 层，也成为持久化层，主要处理的是数据库相关的增删改查，Controller 层又称接入层，给外界提供访问的各个入口，用 RequestMapper 等等注解实现，最后是视图层，也就是用户所看到的前端页面，其实算是四层结构。

SSM 主要的优点就是对很多组件的兼容性都是极好的，比如 SSM 就整合了 Mybatis，用 Mybatis 能够大幅度的减少冗余代码，能够减少程序员开发过程中的工作量。还有一些其他的功能性插件也都是十分好用的，就比如分页助手 PageHelper，能够帮助实现分页功能，以往是有三种分页方式的，有两种分别是前端和后端进行的，但其实都是假分页，要明白一件事，就是分页不仅仅是为了看起来更加方便，重要的是能够减少数据库查询的压力，在数据库的数据极为庞大，比如几百万条的消息时，如果你采用后端分页，将所有数据查出来然后进行筛选，最后传递到前端，是相当慢的，更别提前端分页掩耳盗铃的做法。实际上正确的分页只有一种就是基于数据库的，使用 Limit 命令，分页查询，但是这种查询方式也不省事，难免写大篇幅的代码，而这个 PageHelper 就能够帮助精简操作数据库时需要写的代码。当然不仅仅这些，在此就不多赘述了。

---

## 第 3 章 系统分析

### 3.1 可行性分析

本系统设计题目是基于 SSM 的汽车 4S 店维修保养管理系统，有着多个需要实现的模块，分别是个人中心、用户管理、汽车信息管理、汽车维修预约、汽车保养预约，还有相对基础的登录注册等功能。在确定这些目标功能后，下一步就是进行四方面的可行性分析。

#### 3.1.1 技术可行性

SSM 的汽车 4S 店维修保养管理系统要求能够管理汽车的型号、颜色、名称等信息，以及用户提交的维修、保养预约信息，并且尽可能的简化界面，方便操作和使用，前端采用的是 JSP，还运用了 Bootstrap 等技术，数据库采用 MySQL，再加上 JAVA 简单易用等特点，在开发上，拥有很高的可行性。

#### 3.1.2 经济可行性

因为考虑到要做的这个项目并非一个大项目，而是一个汽车 4S 店用小型信息管理项目，拥有的用户量并不是很多，并发量也不大，因此不用考虑集群的问题，这使得硬件和软件环境的要求都很低，市场买随处都能买到，成本低且便于维护，而且开发周期短，因此在经济上具有可行性。

#### 3.1.3 操作可行性

本系统是电脑端的类似官网后台一体的管理系统，在前端开发中，页面上会使用 Bootstrap 技术的导航头，使用导航头来明确区分各个功能，并且编写一个过滤器，按需进行页面跳转，也就是说只要登录对应的用户/管理员账号，就会自动跳转到对应的页面上去，因为 java 虚拟机的优点，完全可以在电脑端跑起来。因此本系统是可以进行开发的。

#### 3.1.4 法律可行性

本系统是自主开发的一个全新的汽车 4S 店信息管理系统，技术上可能会参照一些大牛的贴吧，但是绝对不会出现照抄的现象，本系统的代码都是开源的，不雷同任何人的项目成果，所以具有开发可行性。

如上所说，本汽车 4S 店维修管理系统在技术、经济、操作、法律这四个方面都是具有可行性的。

## 3.2 系统流程分析

在本系统的开发中，系统流程需求分为两个部分，分别是管理员方向和用户方向，但都是在一个页面上进行登录，所以首先就是登录，然后进行业务跳转，最后处理业务，尽可能的精简业务流程，使程序更加方便使用。

### 3.2.1 系统操作流程

当用户成功登录后，会自动跳转到对应的业务界面，进行操作。

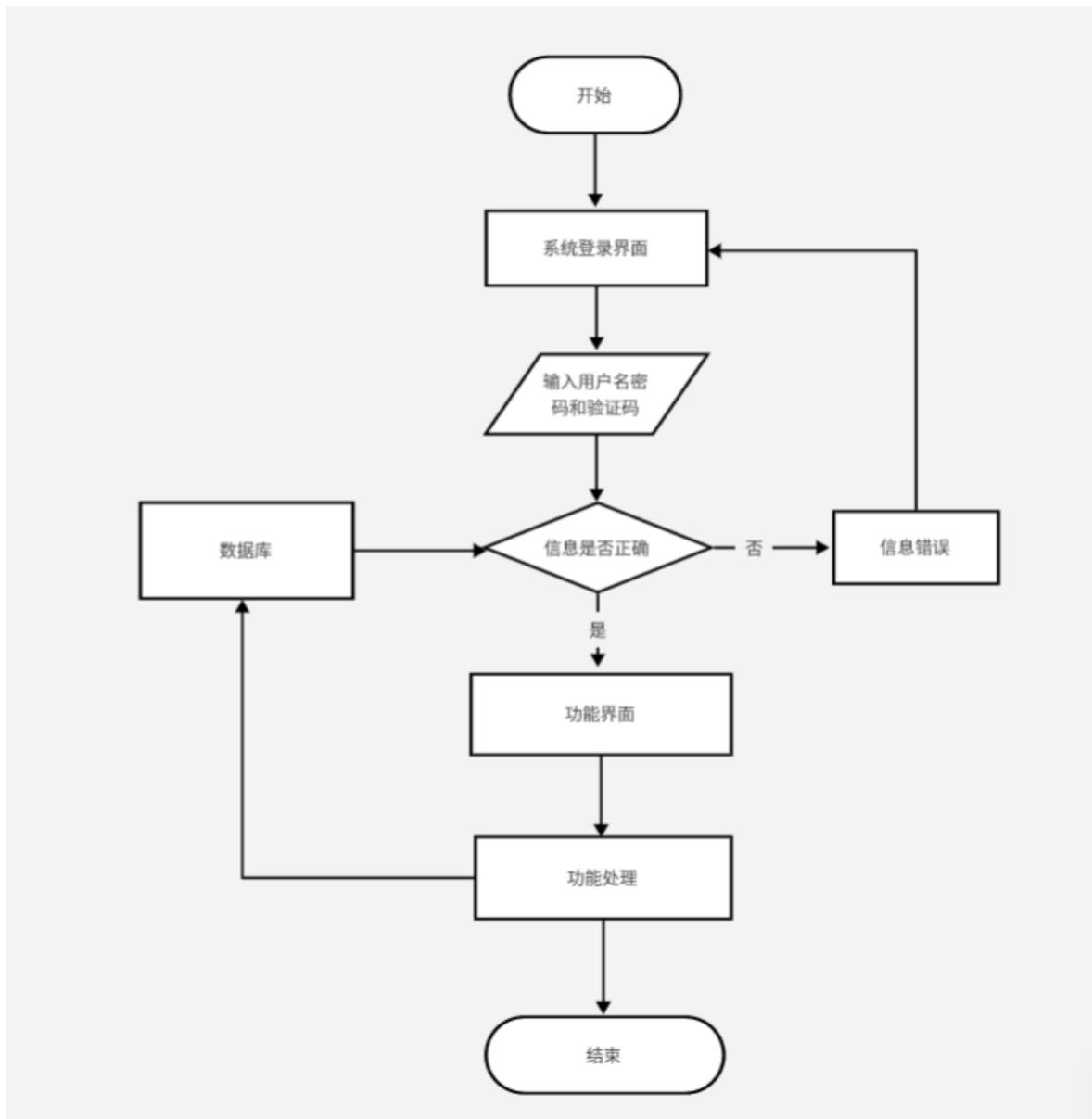


图 3.2 系统操作流程

### 3.2.2 添加信息流程

用户可以进行个人车辆信息的添加、也可以下单进行各种预约行为，可以根据自己的需求进行选择，最后系统读取用户信息添加到数据库。

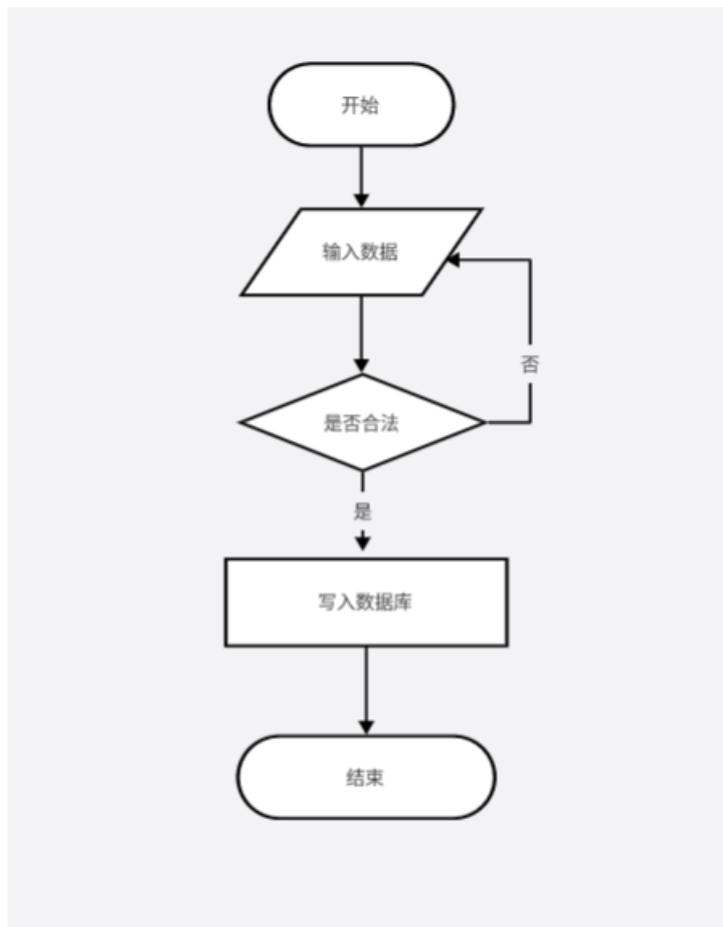


图 3.3 添加信息流程图

### 3.2.3 修改信息流程

管理员可以修改用户部分信息，用户也可以对自己的个人信息车辆等等信息进行修改，成功后进行持久化，失败不进行保存。

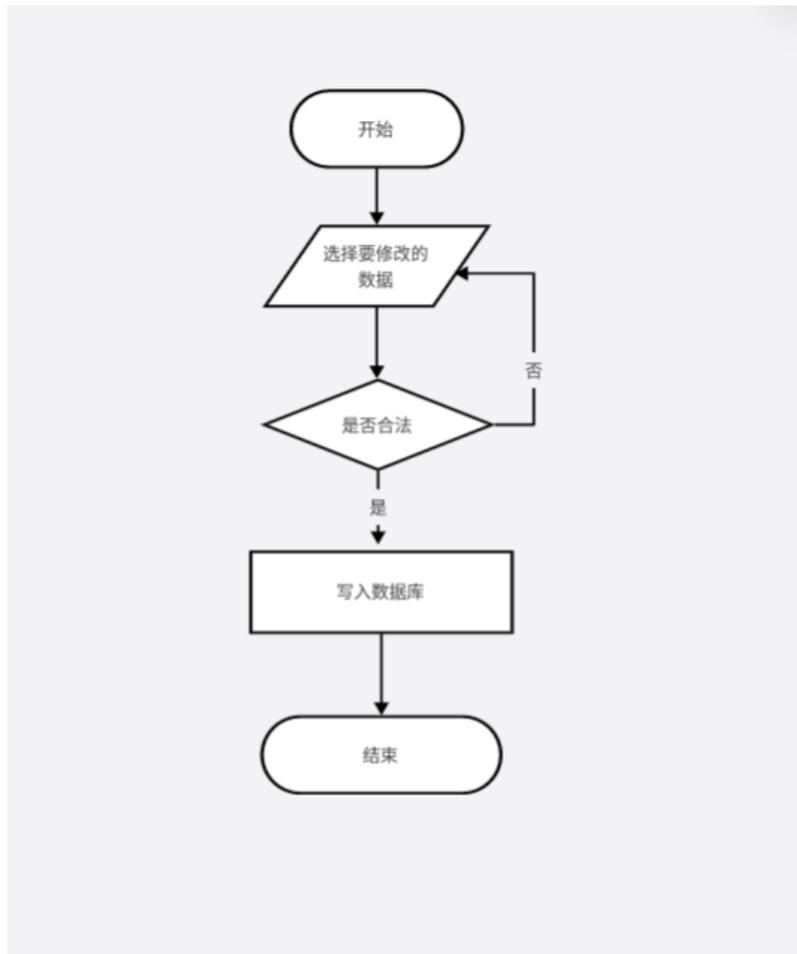


图 3.4 修改信息流程图

### 3.2.4 删除信息流程

管理员和用户都可以根据需求进行业务上的管理与删除，当选中删除条目后，就会进行删除，从数据库清除该条信息。

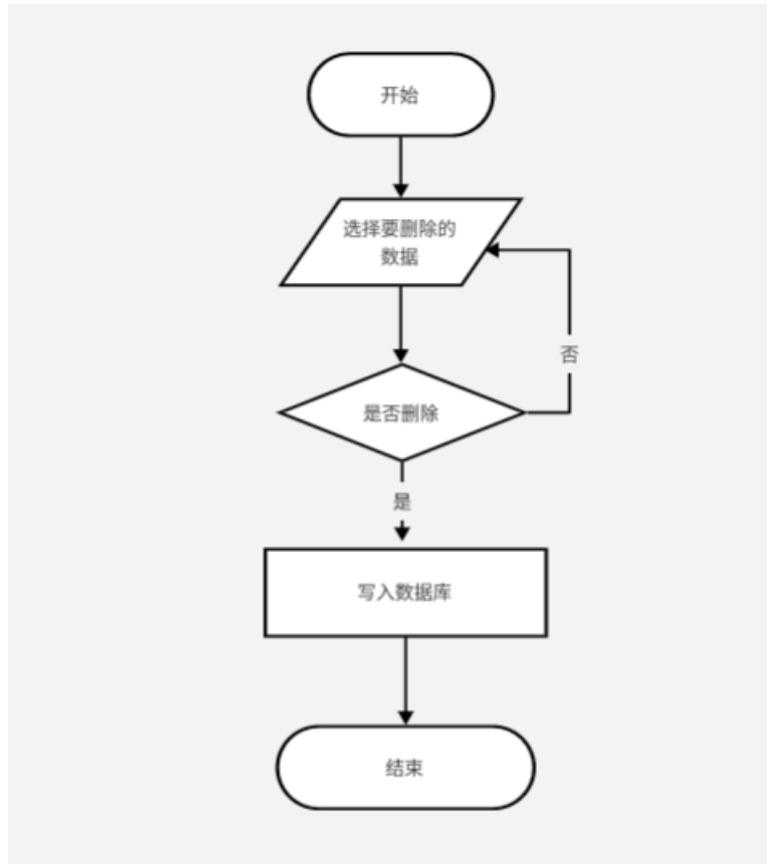


图 3.5 删除信息流程图

## 第 4 章 系统设计

### 4.1 系统概述

在完成对系统需求的分析后，就开始了系统的设计了，其中包括总体结构上的设计和比较详细的设计。其中总体设计指的是一个大方向，在确认大方向后，再去细致划分各个步骤，一步步完成各个模块，最终实现项目，当然本条目只有对系统的设计，系统的具体实现在实现条目上。

最终再进行各种比对，比如对性能、美观等等方向上，最终确定一个最好的设计计划。这一点是很重要的，更大的去贴近自己想要的系统设计，从而避免了在完成项目后发现项目的不足而需要重新设计开发的时间。

维修管理系统的工作原理如图所示：

图 4.1 维修管理系统的工作原理图

### 4.2 功能模块设计

要想一眼浏览系统的功能，就要规划一张架构图，系统架构图也是十分重要的，它能够很清晰的告诉需要实现的功能模块有哪些，也决定了整个系统的架构模式。是十分基础的一个产物，维修保养管理系统的整体架构设计如图：

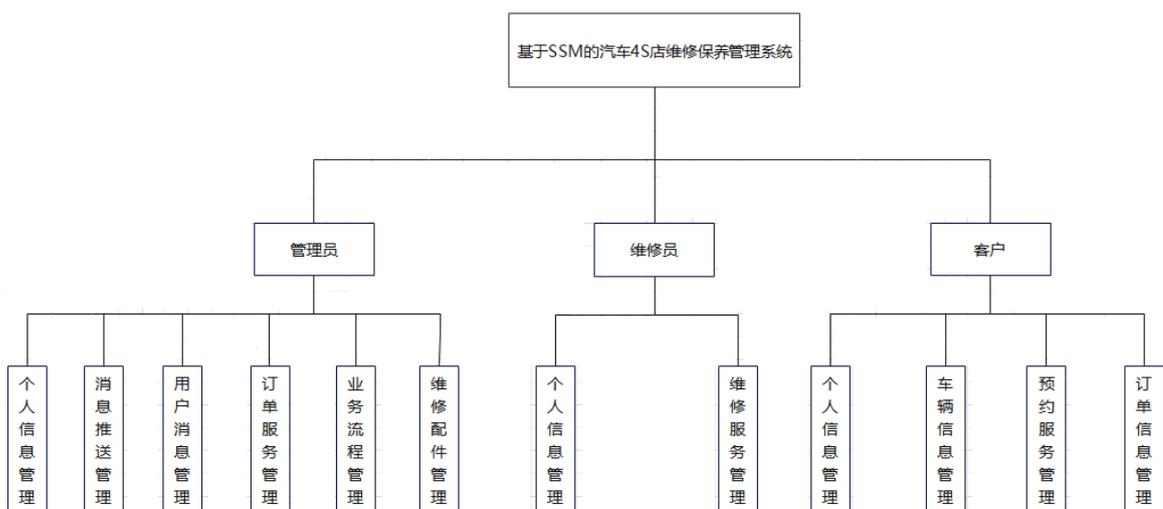


图 4.2 维修保养管理系统的整体架构图

### 4.3 数据库设计

想要进行数据的持久化，就要选择使用数据库，所以数据库设计是信息管理系统的基础，数据的选择与开发对整个系统的运行速度有着绝对影响。

#### 4.3.1 数据库设计原则

因为采用的数据库是 mysql 数据库，所以应该采用的数据库设计方案就是 java 实体对象对应数据库字段的方式来设计，使数据库中的字段名与 java 实体对象的属性一一对应，并且对驼峰命名等命名法进行处理。数据库在系统中担任着持久化的作用，决定着系统能否正常工作，因此要按照实体-关系的方式来设计。

#### 4.3.2 数据库实体

数据库实体对应着现实中物品的各种属性，比如汽车的颜色、型号，用户的姓名、性别等等，主要的任务就是对信息进行持久化，一些成员可能拥有一些较为复杂的属性，因此在分析阶段要进行区分，使这些属性与实体密切相关。

本系统的 E-R 图如下所示：

1.账号表（account）实体图：

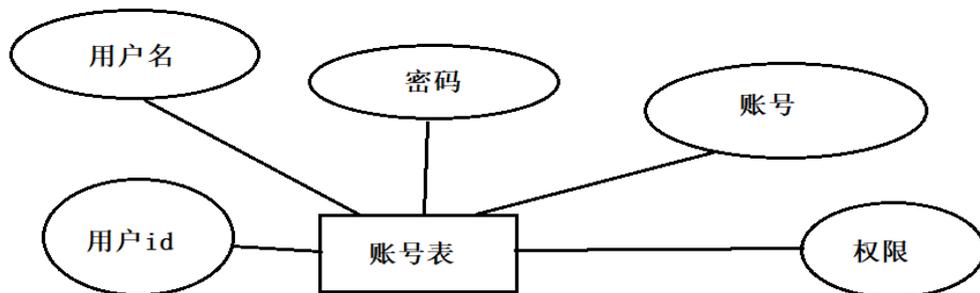


图 4.3 账号表实体图

2.汽车信息表（cars）实体图：

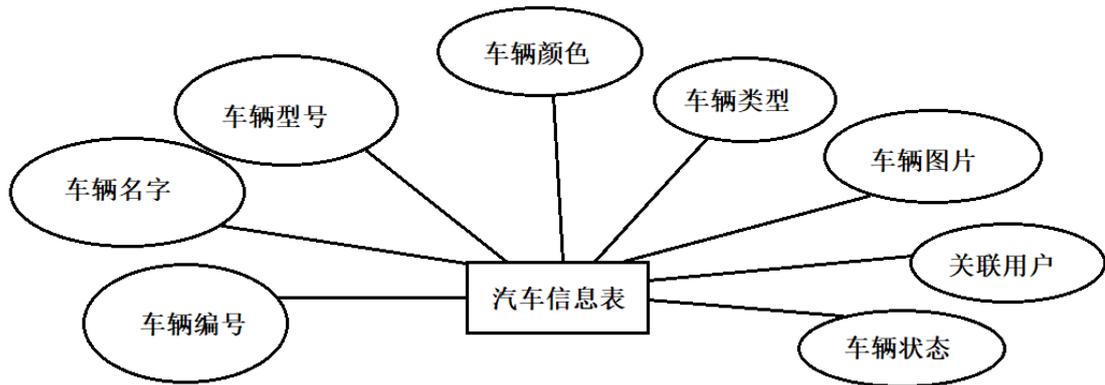


图 4.4 汽车信息表实体图

3.用户信息表 (users) 实体图:

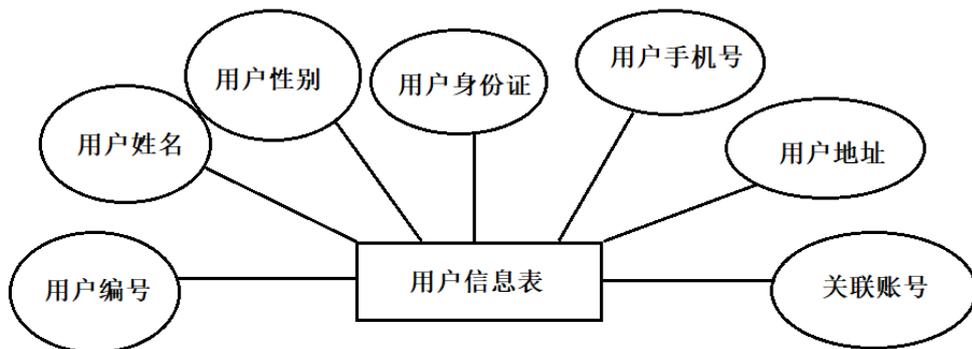


图 4.5 用户信息表实体图

4.订单表 (orders) 实体图:

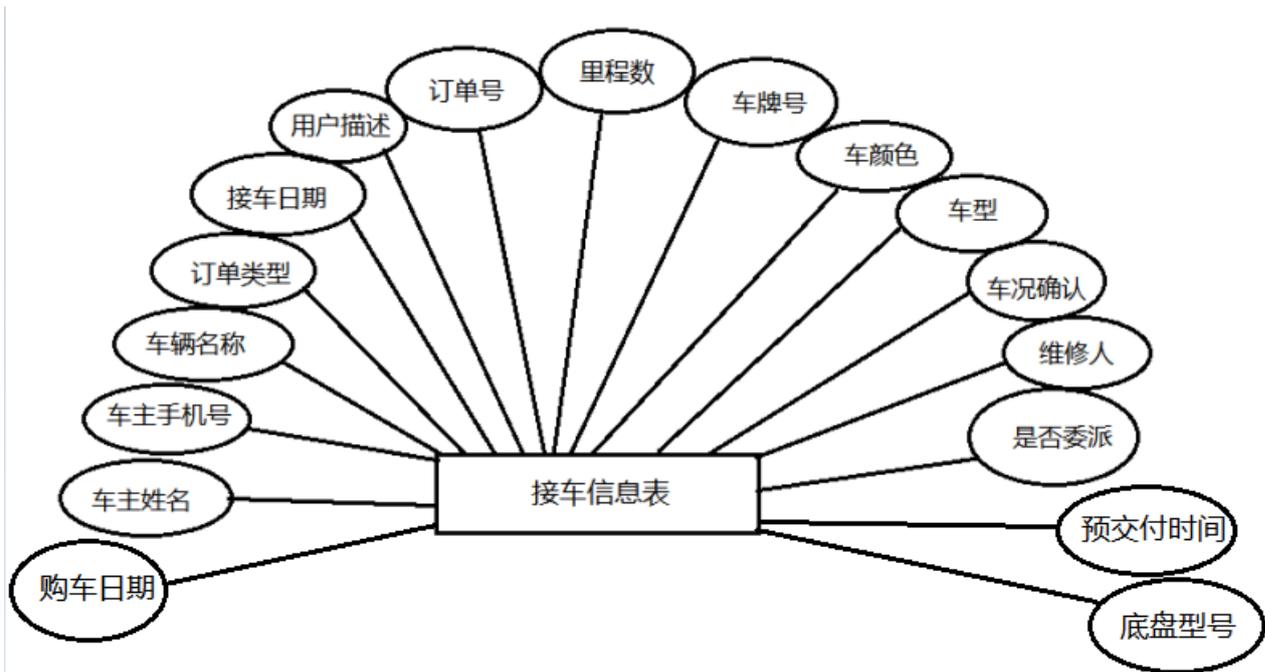


图 4.6 订单表实体图

5. 零部件信息表 (partspares) 实体图:

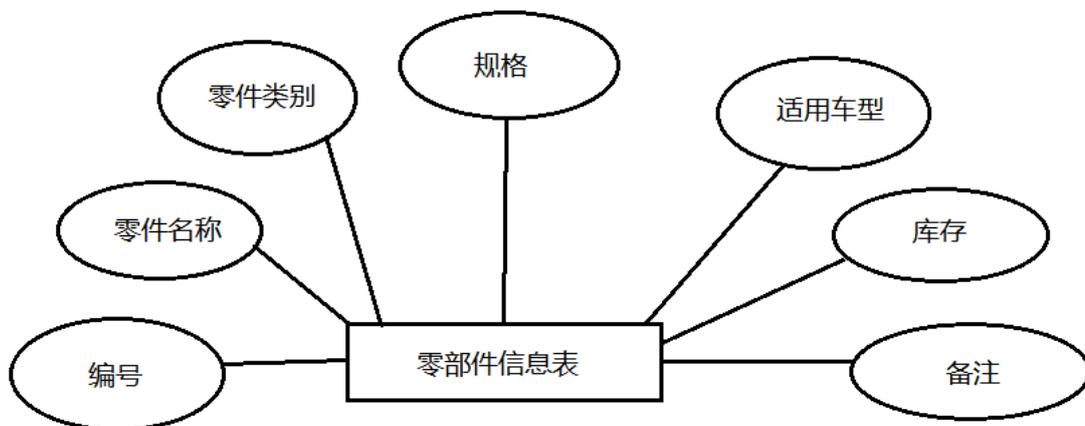


图 4.7 零部件信息表实体图

6. 接车信息表 (pickcar) 实体图:

图 4.8 接车信息表实体图

7. 维修保养服务信息表 (service) 实体图:

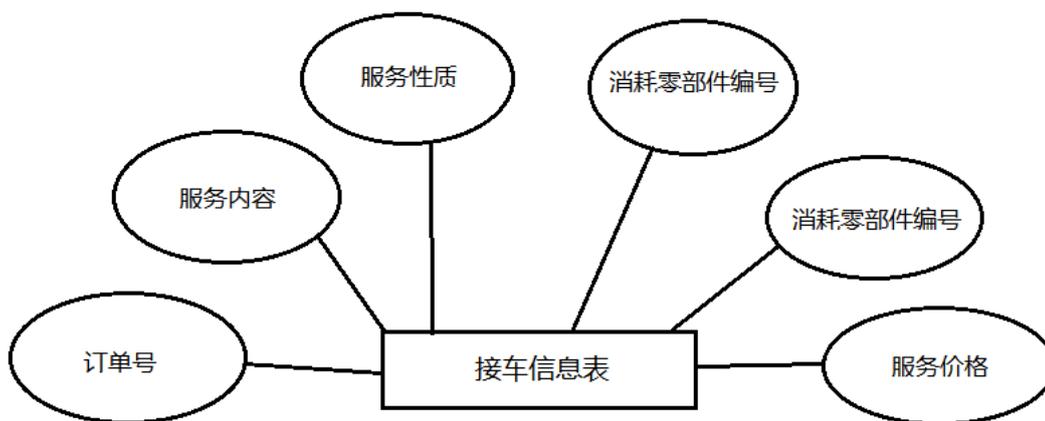


图 4.9 维修保养服务信息表实体图

8.服务项目表 (inventory) 实体图:

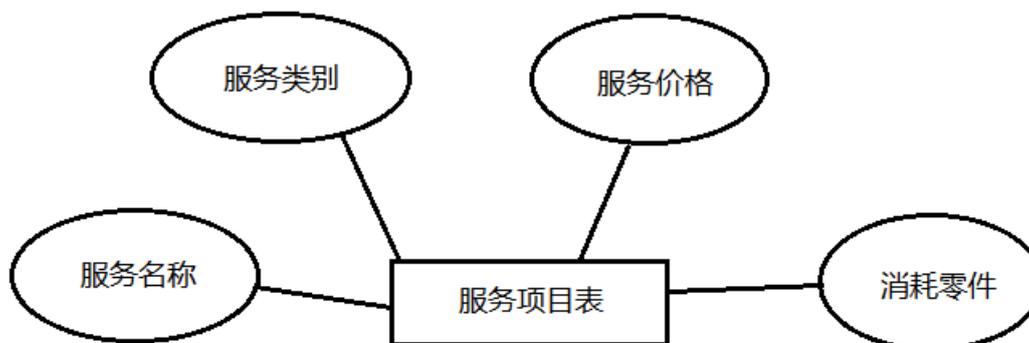


图 4.9 维修保养服务信息表实体图

### 4.3.3 数据库表设计

数据库的表信息属于设计的一部分，下面介绍数据库中的各个表的详细信息。

(1) 账号表：用来存储登录账户、密码等信息。

表 4.1 账号表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
Name	varchar	20	用户名		
password	varchar	20	密码		
aid	varchar	20	账号		

admin	tinyint	1	管理员认证	0
-------	---------	---	-------	---

(2) 汽车信息表：存储汽车相关信息。

表 4.2 汽车信息表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
c_name	varchar		汽车名称		
c_model	varchar	200	汽车型号		
c_color	varchar	200	汽车颜色		
c_type	varchar	200	汽车类型		
c_photo	varchar	200	汽车图片		
a_id	Int	200	关联账户		
c_status	varchar	200	汽车状态		

(3) 用户信息表：存储用户真实信息。

表 4.3 用户信息表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
u_name	varchar		用户姓名		
u_sex	varchar	200	用户性别		
u_id	varchar	200	身份证号		
u_phone	varchar	200	手机号		
u_address	varchar	200	用户住址		
a_id	Int		关联账户		

(4) 订单表：存储订单号、下单用户、订单状态等订单相关信息。

表 4.4 订单表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
o_id	varchar	20	订单号		
o_type	varchar	20	订单类型		
o_name	varchar	20	订单名称		
o_desc	varchar	100	订单描述		
o_date	date		下单日期		
a_name	varchar	20	关联账户名		
u_name	varchar	20	关联真实名		
u_phone	varchar	20	手机号		
a_id	int		关联账户		
o_handle	varchar	20	处理进度		
o_status	varchar	20	订单状态		

(5) 零部件信息表：记录零件的编号、规格、名称等零件相关信息。

表 4.5 零部件信息表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
number	varchar	20	配件编号		
name	varchar	20	配件名称		
type	Int	20	配件类型		
specifications	varchar	20	配件规格		
suit	varchar		使用车型		
inventory	int	20	库存		

notes	varchar	20	备注
-------	---------	----	----

(6) 接车信息表：记录接车相关信息，包括车主信息和汽车信息、委派人员等。

表 4.6 接车信息表

字段名称	类型	长度	字段说明	主键	默认值
id	bigint		主键	主键	
u_name	varchar	50	姓名		
u_phone	varchar	50	手机号		
c_name	Int	50	车名		
o_type	varchar	50	类型		
o_date	varchar	50	日期		
o_desc	int	50	描述		
o_id	varchar	50	订单号		
o_mileage	varchar	50	里程数		
c_number	varchar	50	车牌号		
c_color	varchar	50	车颜色		
c_model	varchar	50	车型		
c_situation	varchar	50	车况确认		
o_repairman	varchar	50	维修人		
c_buyday	date		购买日期		
c_pretime	date		预计完工时间		
c_under	varchar	50	底盘号		
o_status	varchar	50	是否委派		

(7) 维修保养服务信息表：主要记录服务内容和消耗的零件编号以及数量。

表 4.7 维修保养服务信息表

字段名称	类型	长度	字段说明	主键	默认值
id	int	11	主键	主键	
o_id	varchar	50	订单号		
s_service	varchar	50	服务内容		
s_nature	varchar	50	服务性质		
s_cost	varchar	50	服务性质		
s_cost_count	Int	11	消耗零部件编号		
s_money	double		服务价格		

(8) 服务项目表：主要记录提供给用户的保养、维修服务项目信息。

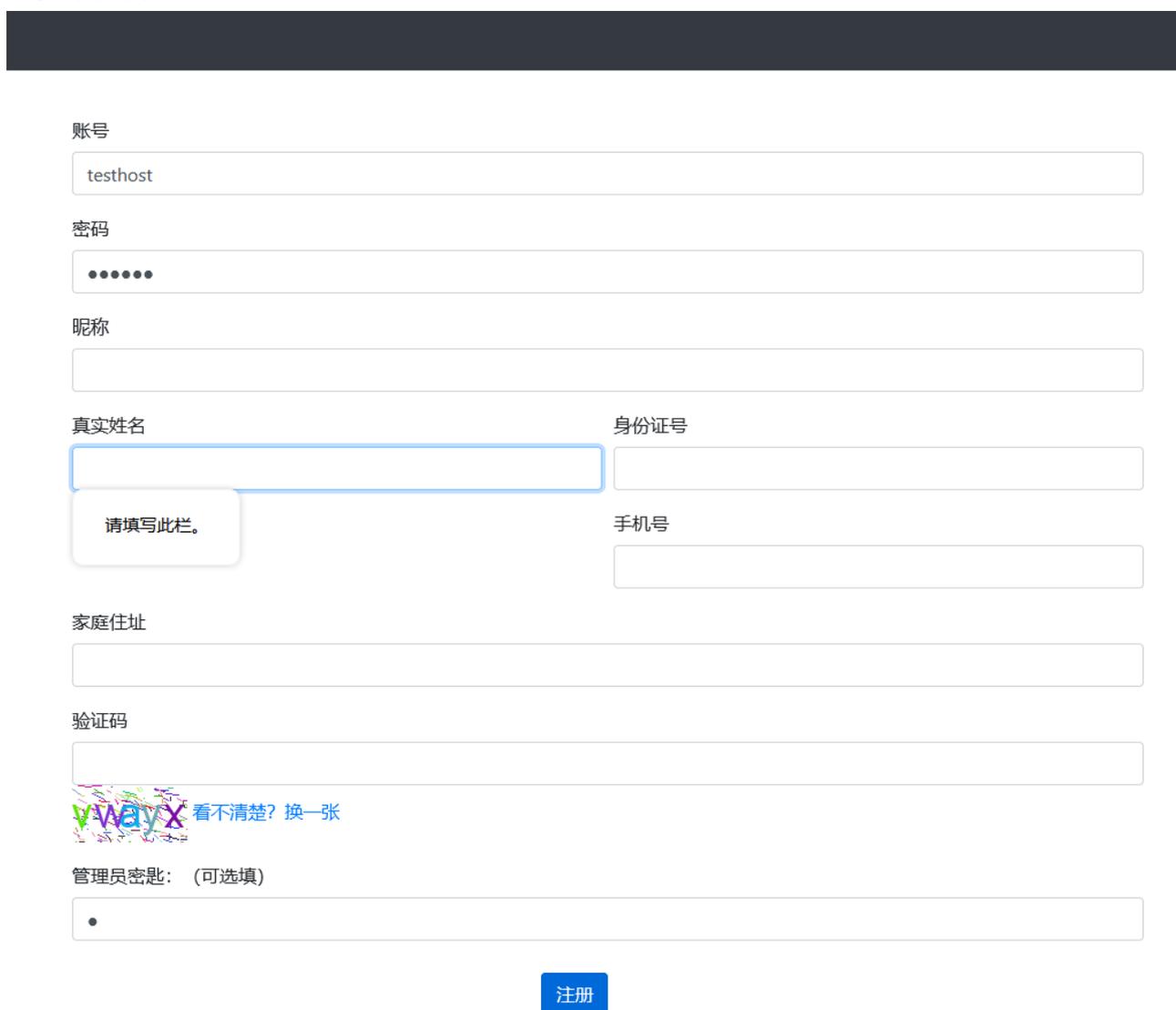
字段名称	类型	长度	字段说明	主键	默认值
id	int	11	主键	主键	
i_name	varchar	50	服务名称		
i_type	varchar	50	服务类别		
i_prise	double		服务价格		
i_part	varchar	50	消耗零件		

## 第 5 章 系统实现

### 5.1 登录注册功能模块

#### 5.1.1 注册模块

用户注册，需要填写账号、密码、用户昵称、真实姓名、性别、身份证号、手机号、家庭住址，以及需要验证验证码，如果需要注册管理员账号需要输入管理员密匙，即可注册成为管理员下面为注册图。



The image shows a user registration form with the following fields and elements:

- 账号 (Account): Input field containing "testhost".
- 密码 (Password): Input field with masked characters (dots).
- 昵称 (Nickname): Input field.
- 真实姓名 (Real Name): Input field with a tooltip that says "请填写此栏。" (Please fill in this field).
- 身份证号 (ID Card Number): Input field.
- 手机号 (Mobile Number): Input field.
- 家庭住址 (Home Address): Input field.
- 验证码 (Verification Code): Input field with a CAPTCHA image showing "wayx" and the text "看不清楚? 换一张" (Can't see clearly? Change one).
- 管理员密匙: (可选填) (Admin Key: (Optional)): Input field with a single dot.
- 注册 (Register): A blue button at the bottom.

图 5.1 用户注册

实现该功能的关键代码如下。

```
//注册界面入口
```

```
@RequestMapping("register_e")
```

```
public String register_e(){
    return "register";
}
```

如未按照正确格式填写信息，则会有对应的提示，如下图。

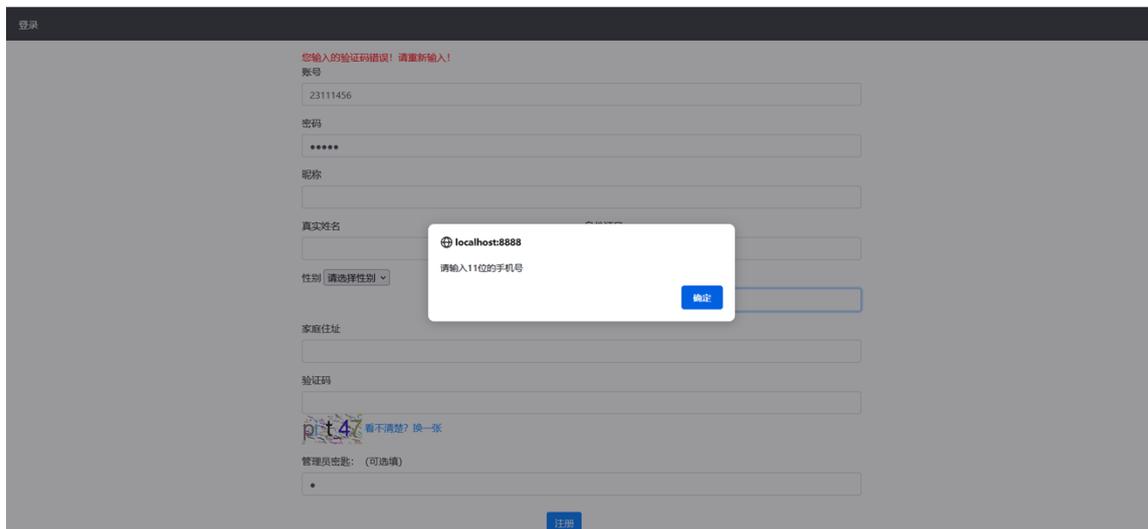


图 5.2 注册信息验证

实现代码在注册功能里。

在注册成功后，则会跳转到登录界面。

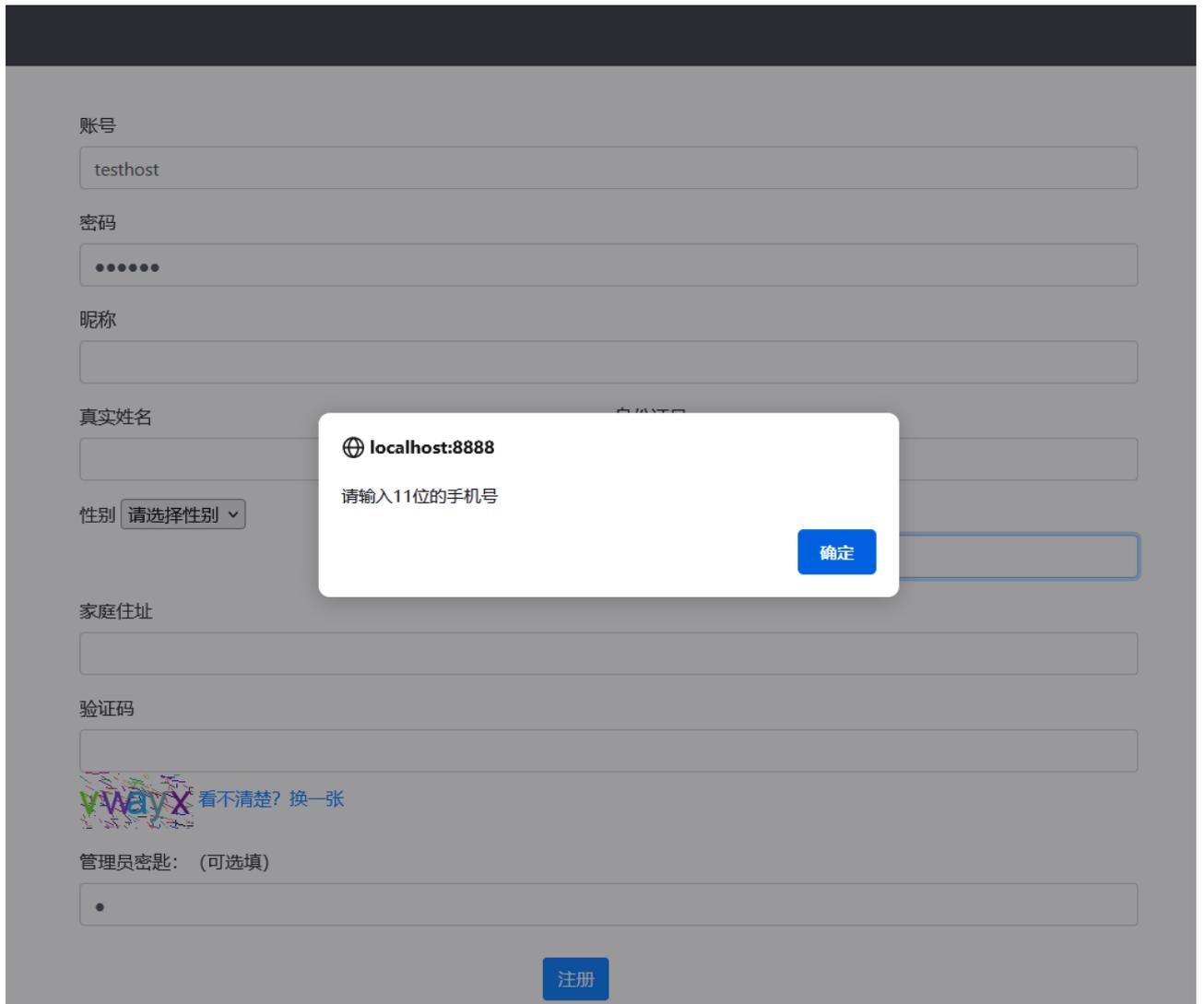


图 5.3 注册成功跳转

实现该功能的关键代码如下。

```
//注册业务实现
@RequestMapping("register_r")
public String register_r(LoginComplex loginComplex,Model model,HttpServletRequest
req){
    System.out.println(loginComplex.getAccount());
    System.out.println(loginComplex.getCode());
    System.out.println(loginComplex.getUsers());
    String code = loginComplex.getCode();
    System.out.println(code);
    HttpSession session = req.getSession();
```

---

```
String codereal = (String)session.getAttribute("code");
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/418005035102006050>