武汉纺织大学

计算机与人工智能学院

Java开发课程设计报告

2020 ~2021学年第2学期

|  |  |
| --- | --- |
| 课程设计题目 | 基于SSM框架的客户信息管理系统 |
| 学生姓名 | 王德明 |
| 指导教师 | 黄俊杰 |
| 评定成绩 |  |
| 填写时间 | 2021.07.10 |
|  | |

摘 要

由于时代的变迁，科技的快速发展，我们现在正处于一个资源泛滥的社会，凡事都离不开资源。对于资源的整合，如果单靠人力去整理，会显得特别麻烦且容易乱，为此我设计了一个客户信息的管理系统。

客户信息管理系统能够简单且高效的管理各类客户的信息，不同的管理员拥有不同的客户资源信息，并且通过加入一些功能来使管理员更加方便的管理客户信息。

该系统后端使用的是Java语言，前端使用了Html+CSS+JS，以Mysql为关系型数据库平台，MVC设计模式为层级，Spring+SpringMVC+Mybatis为框架设计了一个小型的客户信息管理系统。

关键词：Java; Html; CSS; MySql; Spring+SpringMVC+MyBatis

**ABSTRACT**

Due to the changes of the times and the rapid development of science and technology, we are now in a society full of resources, and everything is inseparable from resources. For the integration of resources, if we only rely on manpower to sort out, it will be particularly troublesome and easy to mess, so I designed a customer information management system.

Customer information management system can simply and efficiently manage all kinds of customer information, different administrators have different customer resource information, and by adding some functions to make administrators more convenient to manage customer information.

The back end of the system uses Java language, the front end uses HTML + CSS + JS, takes MySQL as the relational database platform, and spring + spring MVC + mybatis as the framework to design a small customer information management system.

**Keywords:**Java; Html; CSS; MySql; Spring+SpringMVC+MyBatis

目 录

[1. 引言 4](#_Toc1573)

[1.1 客户信息管理系统开发背景 4](#_Toc15759)

[1.2 系统开发的目的 4](#_Toc32189)

[2. 系统技术概括 4](#_Toc22178)

[3. 代码实现及功能截图 7](#_Toc22938)

[3.1 底层代码 7](#_Toc23020)

[3.1.1 model层中的对象层 7](#_Toc26291)

[3.1.2 model层中的数据访问层 8](#_Toc23255)

[3.1.3 model层中的服务层 9](#_Toc9202)

[3.1.4 model层中的控制层 9](#_Toc14652)

[3.1.5 view层 9](#_Toc17401)

[3.1.6 Servlet过滤器 10](#_Toc11939)

[3.1.7 数据库操作 10](#_Toc19959)

[3.2 运行截图 10](#_Toc4661)

[4. 总结 17](#_Toc24127)

[5.参考文献 18](#_Toc29059)

1. 引言

1.1 客户信息管理系统开发背景

由于客户的信息越来越多，用纸来记录显得特别臃肿，且随着信息的增多，人为的记录信息就很难进行查找、整理、存储。

1.2 系统开发的目的

为了解决这个问题，我开发了一个信息管理系统，可以随时的访问服务器来进行系统的操作，把所有的客户信息都整理到服务器上，不同的管理员有着不同的客户信息，分门别类的放置在数据库中进行持久化操作，这样一来不仅解决了存储困难的问题，而且还让管理员能够快速的找到需要的客户信息，大大的提升了信息管理工作的效率。

2. 系统技术概括

2.1 SpringMVC

Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。Spring 框架提供了构建 [Web](https://baike.baidu.com/item/Web/150564" \t "https://baike.baidu.com/item/spring%20MVC/_blank) 应用程序的全功能 MVC 模块。使用 Spring 可插入的 MVC 架构，从而在使用Spring进行WEB开发时，可以选择使用Spring的Spring MVC框架或集成其他MVC开发框架，如Struts1(现在一般不用)，Struts 2(一般老项目使用)等等。

2.2 HTML

HTML的全称为超文本标记语言，是一种[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/5964436" \t "https://baike.baidu.com/item/HTML/_blank)。它包括一系列标签．通过这些标签可以将网络上的[文档](https://baike.baidu.com/item/%E6%96%87%E6%A1%A3/1009768" \t "https://baike.baidu.com/item/HTML/_blank)格式统一，使分散的[Internet](https://baike.baidu.com/item/Internet/272794" \t "https://baike.baidu.com/item/HTML/_blank)资源连接为一个逻辑整体。HTML文本是由HTML命令组成的描述性[文本](https://baike.baidu.com/item/%E6%96%87%E6%9C%AC/5443630" \t "https://baike.baidu.com/item/HTML/_blank)，HTML命令可以说明[文字](https://baike.baidu.com/item/%E6%96%87%E5%AD%97/612910" \t "https://baike.baidu.com/item/HTML/_blank)，[图形](https://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2/773307" \t "https://baike.baidu.com/item/HTML/_blank)、[动画](https://baike.baidu.com/item/%E5%8A%A8%E7%94%BB/206564" \t "https://baike.baidu.com/item/HTML/_blank)、[声音](https://baike.baidu.com/item/%E5%A3%B0%E9%9F%B3/33686" \t "https://baike.baidu.com/item/HTML/_blank)、[表格](https://baike.baidu.com/item/%E8%A1%A8%E6%A0%BC/3371820" \t "https://baike.baidu.com/item/HTML/_blank)、[链接](https://baike.baidu.com/item/%E9%93%BE%E6%8E%A5/2665501" \t "https://baike.baidu.com/item/HTML/_blank)等。

[超文本](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC/2832422" \t "https://baike.baidu.com/item/HTML/_blank)是一种组织信息的[方式](https://baike.baidu.com/item/%E6%96%B9%E5%BC%8F/3616191" \t "https://baike.baidu.com/item/HTML/_blank)，它通过超级链接方法将文本中的文字、图表与其他[信息媒体](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E5%AA%92%E4%BD%93/8400627" \t "https://baike.baidu.com/item/HTML/_blank)相关联。这些相互关联的信息媒体可能在同一文本中，也可能是其他文件，或是地理位置相距遥远的某台计算机上的文件。这种组织信息方式将分布在不同位置的信息资源用随机方式进行连接，为人们查找，检索信息提供方便。

2.3 JS

JavaScript（简称“JS”） 是一种具有函数优先的轻量级，解释型或即时编译型的[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80/9845131" \t "https://baike.baidu.com/item/JavaScript/_blank)。虽然它是作为开发[Web](https://baike.baidu.com/item/Web/150564" \t "https://baike.baidu.com/item/JavaScript/_blank)页面的[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80/1379708" \t "https://baike.baidu.com/item/JavaScript/_blank)而出名，但是它也被用到了很多非[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8/213911" \t "https://baike.baidu.com/item/JavaScript/_blank)环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式、声明式、[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0/301912" \t "https://baike.baidu.com/item/JavaScript/_blank)式编程范式。

JavaScript在1995年由[Netscape](https://baike.baidu.com/item/Netscape/2778944" \t "https://baike.baidu.com/item/JavaScript/_blank)公司的Brendan Eich，在网景导航者浏览器上首次设计实现而成。因为Netscape与[Sun](https://baike.baidu.com/item/Sun/69463" \t "https://baike.baidu.com/item/JavaScript/_blank)合作，Netscape管理层希望它外观看起来像Java，因此取名为JavaScript。但实际上它的语法风格与[Self](https://baike.baidu.com/item/Self/4959923" \t "https://baike.baidu.com/item/JavaScript/_blank)及[Scheme](https://baike.baidu.com/item/Scheme/8379129" \t "https://baike.baidu.com/item/JavaScript/_blank)较为接近。

JavaScript的标准是[ECMAScript](https://baike.baidu.com/item/ECMAScript /1889420" \t "https://baike.baidu.com/item/JavaScript/_blank)。截至 2012 年，所有浏览器都完整的支持ECMAScript 5.1，旧版本的浏览器至少支持ECMAScript 3 标准。2015年6月17日，ECMA国际组织发布了ECMAScript的第六版，该版本正式名称为 ECMAScript 2015，但通常被称为ECMAScript 6 或者ES2015。

2.4 Mysql

数据库分为关系型数据库和非关系型数据库。

关系型数据库指的是采用了关系模型来组织数据的数据库，拥有关系，元组，

属性，域，关键字，关系模式等概念。关系型数据库比较直观，更容易被理解，

使用起来较为方便，易于维护。但关系型数据库性能欠佳，在处理大量关系复杂

的数据时，效率很低。我们常用的关系型数据库有 Oracle，Microsoft SQL Server， MySQL，PostgreSQL，DB2，Microsoft Access，SQLite 等非关系数据库指的是指与关系型数据库相对的非关系型的，分布式的数据库。它以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，不局限于固定的结构。

非关系型的数据库结构简单，以节省一点的时间和开销。但它仅限处理一些简单的没有复杂关联关系的数据库，不适合持久存储海量数据。由于非关系型数据库都是针对某些特定的应用需求出现的，因此，不同非关系数据库的特长也有所不同，主要分为以下几类：面向高性能并发读写的 key-value 数据库，主流代表为 Redis，Amazon DynamoDB；面向海量数据访问的面向文档数据库，主流代表为MongoDB；面向搜索数据内容的搜索引擎，主流代表为 Splunk，Solr；面向可扩展性的分布式数据库，主流代表为 Cassandra，HBase。

MySQL是典型的关系型数据库，它由瑞典 MySQL AB 公司开发，在被收购之后属于 Oracle 旗下产品。与其他关系型数据库相同，MySQL 将数据保存 在不同的表中，确保效率并提高了活性。MySQL 使用 C 和 C++编写，并使

用了多种编译器进行测试，保证了源代码的可移植性，使得它能够支持大多数

的操作系统。

MySQL 所使用的操作语言 是用于访问数据库的最常用标准化语言 SQL 语

言。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一

般中小型网站的开发都选择 MySQL 作为网站数据库。虽然 MySQL 与其他的

大型数据库比有所不足，但对于个人开发者，MySQL 数据库包含了个人开发

所需的全部功能，并且它是免费开放的，降低了学习成本。也正是出于上述原

因，我在房屋租赁系统中选择了 MySQL 数据库为项目的数据库。

2.5 MyBatis

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Ordinary Java Object,普通的 Java对象)映射成数据库中的记录。

Mybatis有以下特点：

简单易学：本身就很小且简单。没有任何第三方依赖，最简单安装只要两个jar文件+配置几个sql映射文件易于学习，易于使用，通过文档和源代码，可以比较完全的掌握它的设计思路和实现。

灵活：mybatis不会对应用程序或者数据库的现有设计强加任何影响。 sql写在xml里，便于统一管理和优化。通过sql语句可以满足操作数据库的所有需求。

解除sql与程序代码的耦合：通过提供DAO层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试。sql和代码的分离，提高了可维护性。提供映射标签，支持对象与数据库的orm字段关系映射

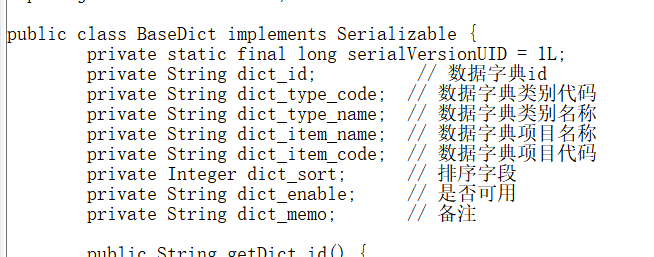
提供对象关系映射标签，支持对象关系组建维护提供xml标签，支持编写动态sql。

3. 代码实现及功能截图

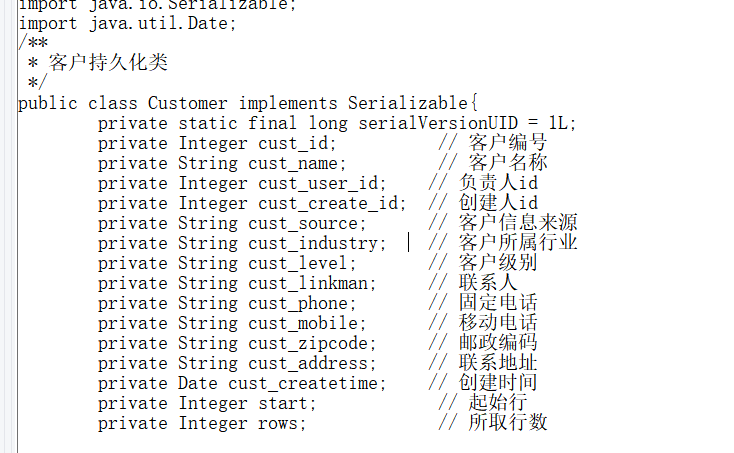
3.1 底层代码

3.1.1 model层中的对象层

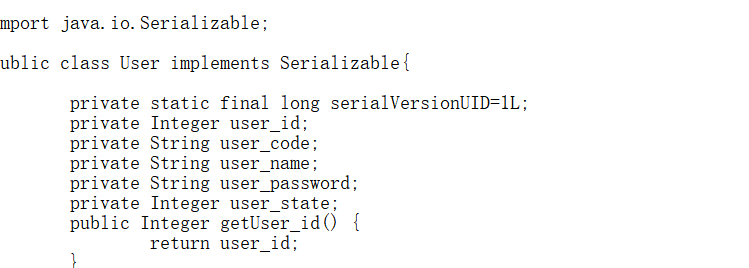
首先我在这层分别创建了三个对象类，分别为BaseDict类，客户持久化类Customer，用户类User，分别对应了数据库中的三张表，后面的一系列操作都是由这个些对象的实例跟数据库进行交互，代码如下三图：



1.1 BaseDict类



1.2 Customer类



1.3 User类

3.1.2 model层中的数据访问层

访问层中，我依次创建了BaseDict，Customer，User类对于数据库具体操作的访问类，以及由Mybatis映射出来的xml文件，里面涉及到了对应的逻辑操作，使sql语句的耦合度降低，可以让我们有更多的时间去判断实现的逻辑问题，

其中最主要功能代码是在[Customer.xml](报告里涉及到的代码链接到此处/CustomerDao.txt)(超链接)文件中，里面是对于客户信息的操作，也是本系统主要的功能部分，有各种增删改查，并且通过嵌套来更加灵活的使用Mybatis技术，但是这个时候我们会发现有的地方会出现Sql注入问题，这也是本系统的Bug部分之一。

3.1.3 model层中的服务层

服务层中首先我创建了三个基础对象的对应的逻辑操作服务接口，其次创建了这三个接口对应的实现类，这三个实现类分别发出不同的功能服务请求，然后通过数据访问层的具体代码来实现系统所要求的功能。BaseDictService接口的实现类中有一个功能，是根据Typecode的值来进行查找相关信息。UserService接口的实现类中的功能就是根据控制层传入的账户密码来进行对数据库的匹配，这个匹配的结果可以作为登录的条件来进行对登录功能的实现。CustomerService的实现类进行了四个对于用户信息的操作， findCustomerList方法根据分页的行数和其它的客户信息来进行一个整体搜索，createCustomer方法根据上一层传来的参数进行包装，然后传入到数据访问层完成对客户信息的添加，getCustomerById就是根据客户ID来查询客户信息，updateCustomer就是将传入前的客户信息对象根据传入后信息进行修改，完成客户信息修改的功能，deleteCustomer就是简单的删除一个用户信息。

3.1.4 model层中的控制层

控制层中主要就是和视图层进行交互，将视图层的数据获取到，然后通过addAttribute放在session中供其它层来使用，requestmapping注解就是根据里面的URL来判断到底执行什么请求方法，这一层中就充分体现了SSM框架中的依赖注入思想。

3.1.5 view层

View层就涉及到了前端的页面设计，我使用的是div+css来设计的，首先将要传送的数据可以放入form表单中，然后通过关键词submit来做出一个提交按钮控制提交，点击按钮之后就可以将表单中的信息以你想要的方式，如get/post形式进行传送，在servlet过滤器下对控制层进行数据输入，视图页面也用到了JS技术以及EL表达式等让jsp里面处理更多逻辑，这里就不一一举例了。

3.1.6 Servlet过滤器

  在Servlet中，如果要定义一个过滤器，则直接让一个类实现javax.servlet.Filter接口即可完成，此接口定义了三个方法， 在三个方法中，最重要的是doFilter()方法，此方法定义了ServletRequest,ServletResponse和FilterChain三种参数，其中前两个参数表示，过滤器可以完成对任意协议的过滤操作。在使用过滤器之前需要在web.xml文件中进行配置之后再才能使用。

3.1.7 数据库操作

在前面的dao层中用到了Mybatis，更新客户功能中，对于每个条信息都有一个判断，如果为空那么就不执行添加那一项，查询的话我用到了一个模糊查询，默认状态下是查询所有的客户，若添加了一些条件，那么就用where来实现，里面嵌套的<if>就很好的取舍了判断的条件。

3.2 运行截图

**首先这是登录界面**

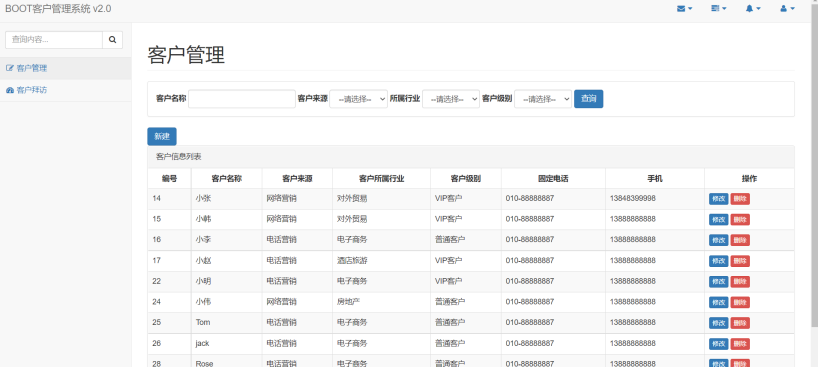
**如果账号和用户名有一个为空，就会提示账户或用户名不能为空**

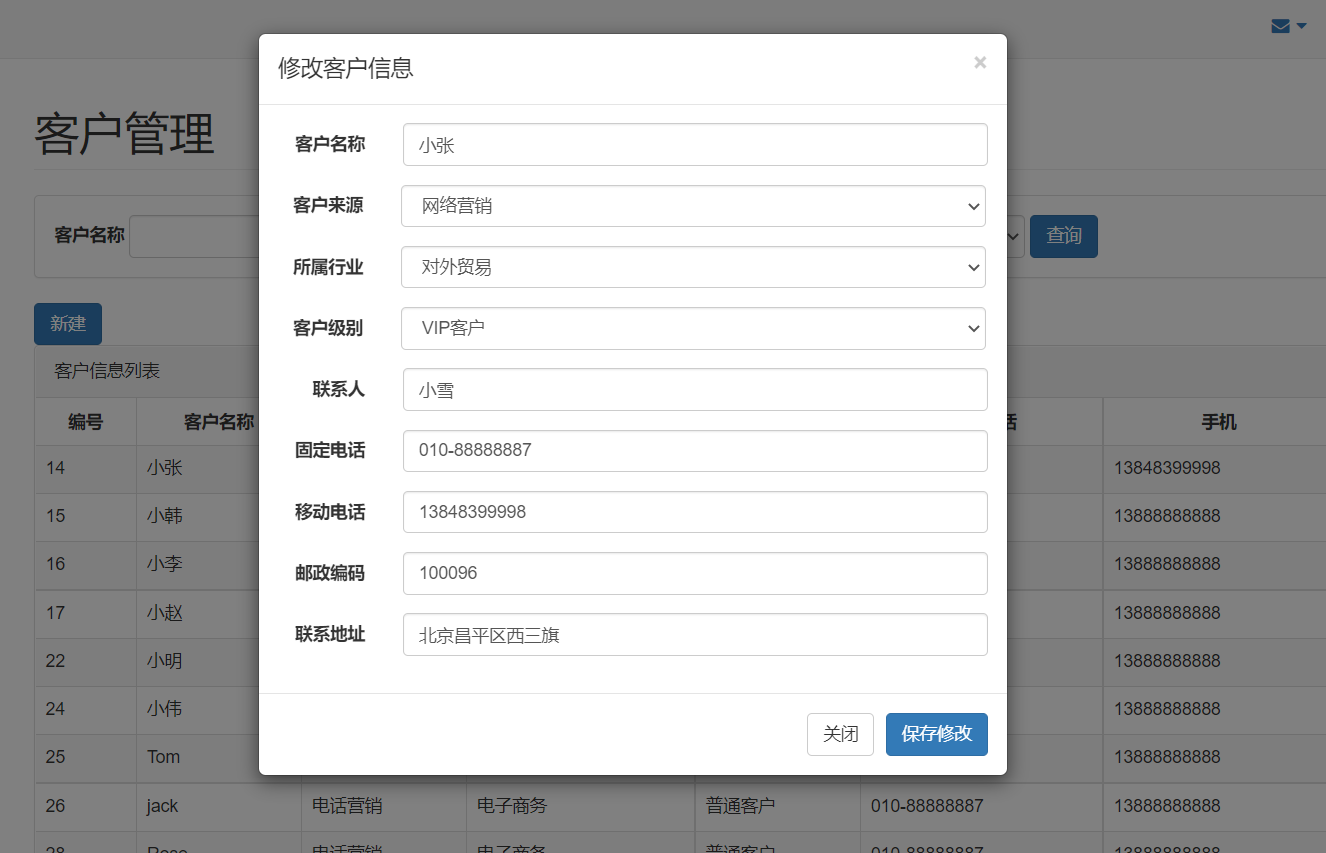


**当都不为空时就进行对数据库匹配，若密码错误，则会出现密码错误的红色字体，直到第n次输入成功**

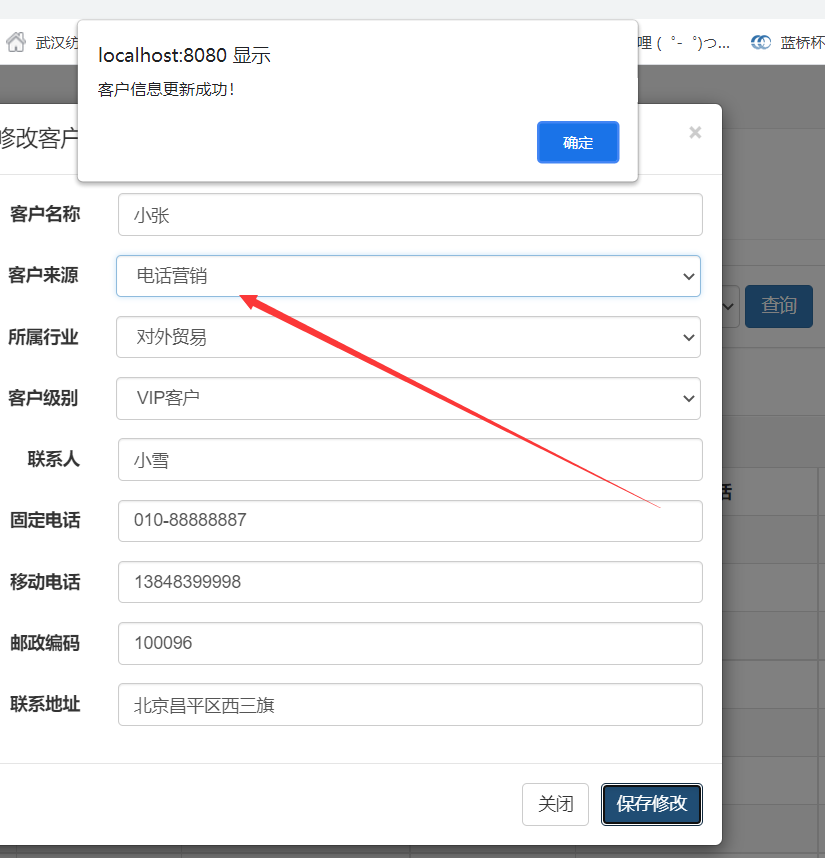


**登录成功后就进入了主界面，同时显示所有的客户信息**



**找到某位客户信息点击修改，会将这位客户的信息显示到屏幕中的小框内**

**信息修改完成后点击保存修改就会提示一个修改成功**

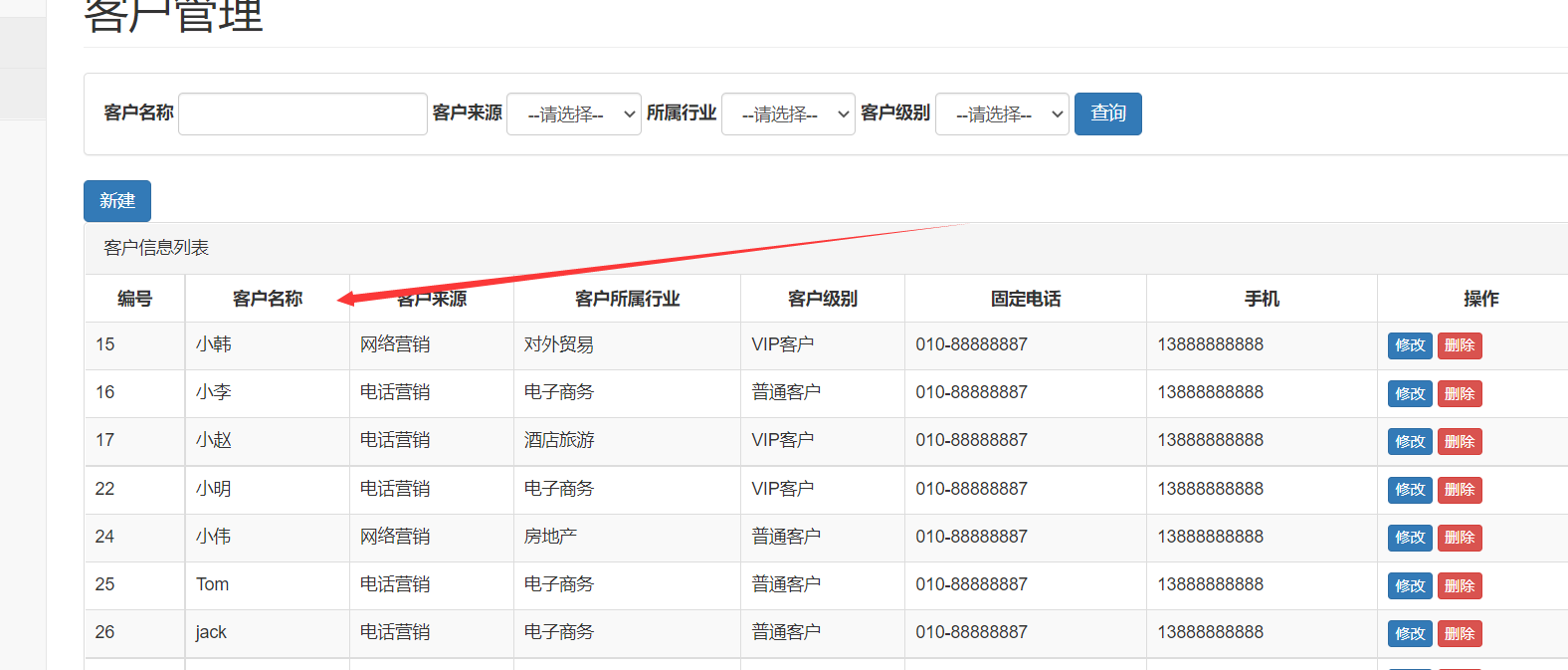


**修改成功之后数据就会发生改变，但此时的客户名称变成了问号，这是本系统中的一个Bug**

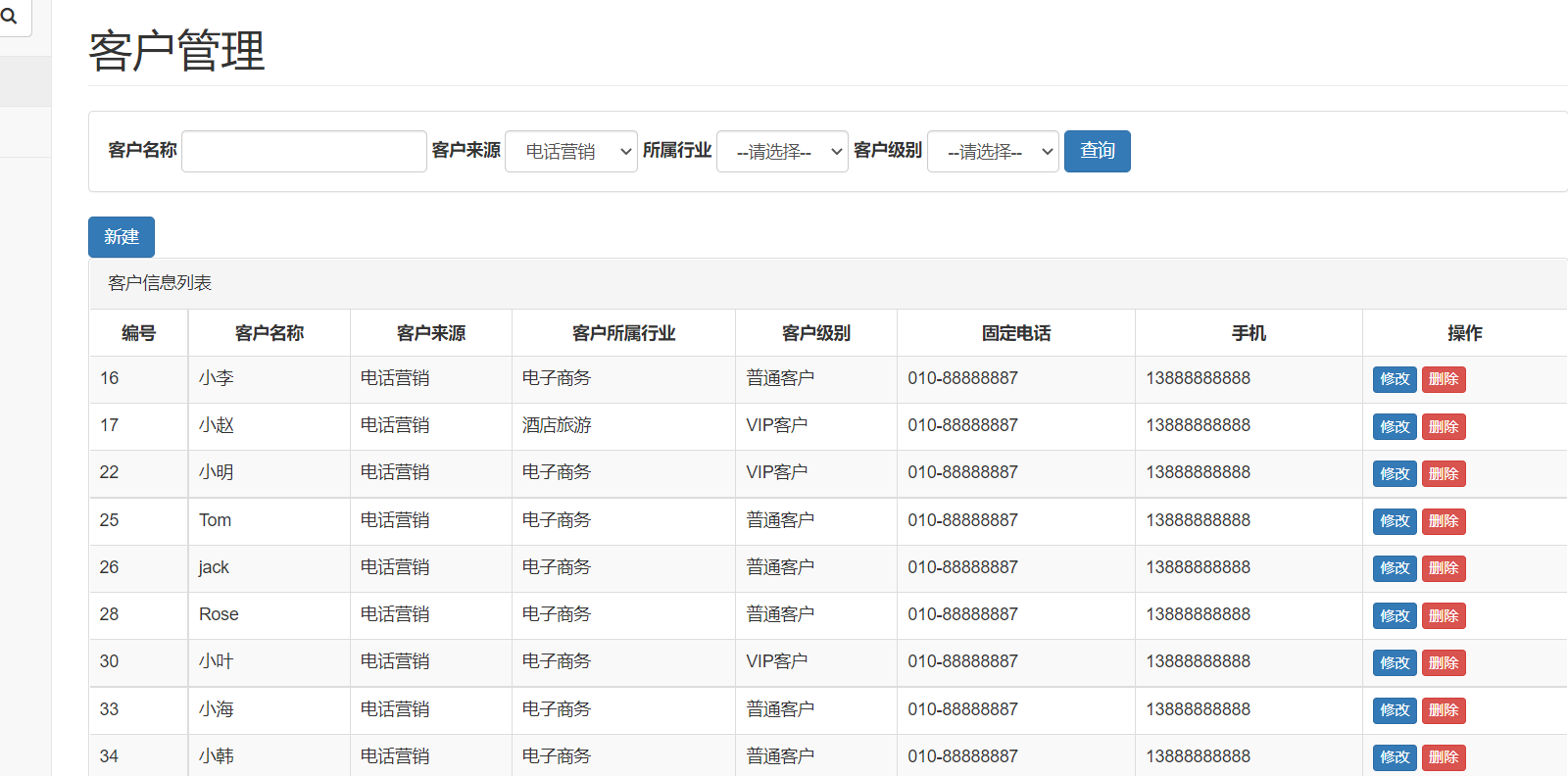


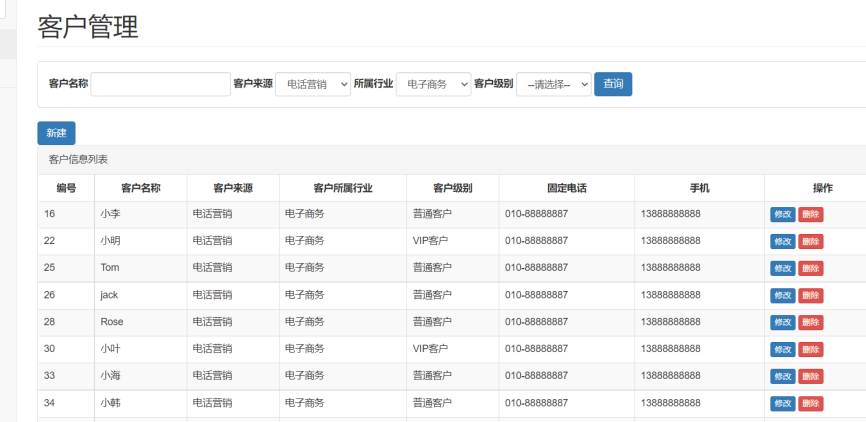
**选中一个客户信息点击删除，就会弹出一个框，向你再次确认是否删除**

**删除成功后列表的信息少了一位**



**中间的查询框支持单条件查询也支持多条件查询，如图**

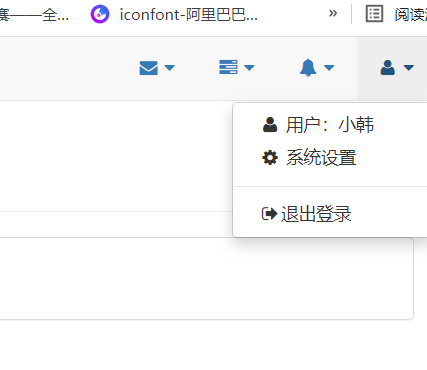
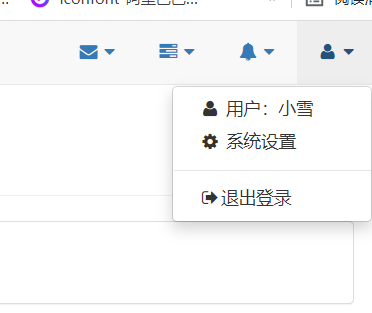




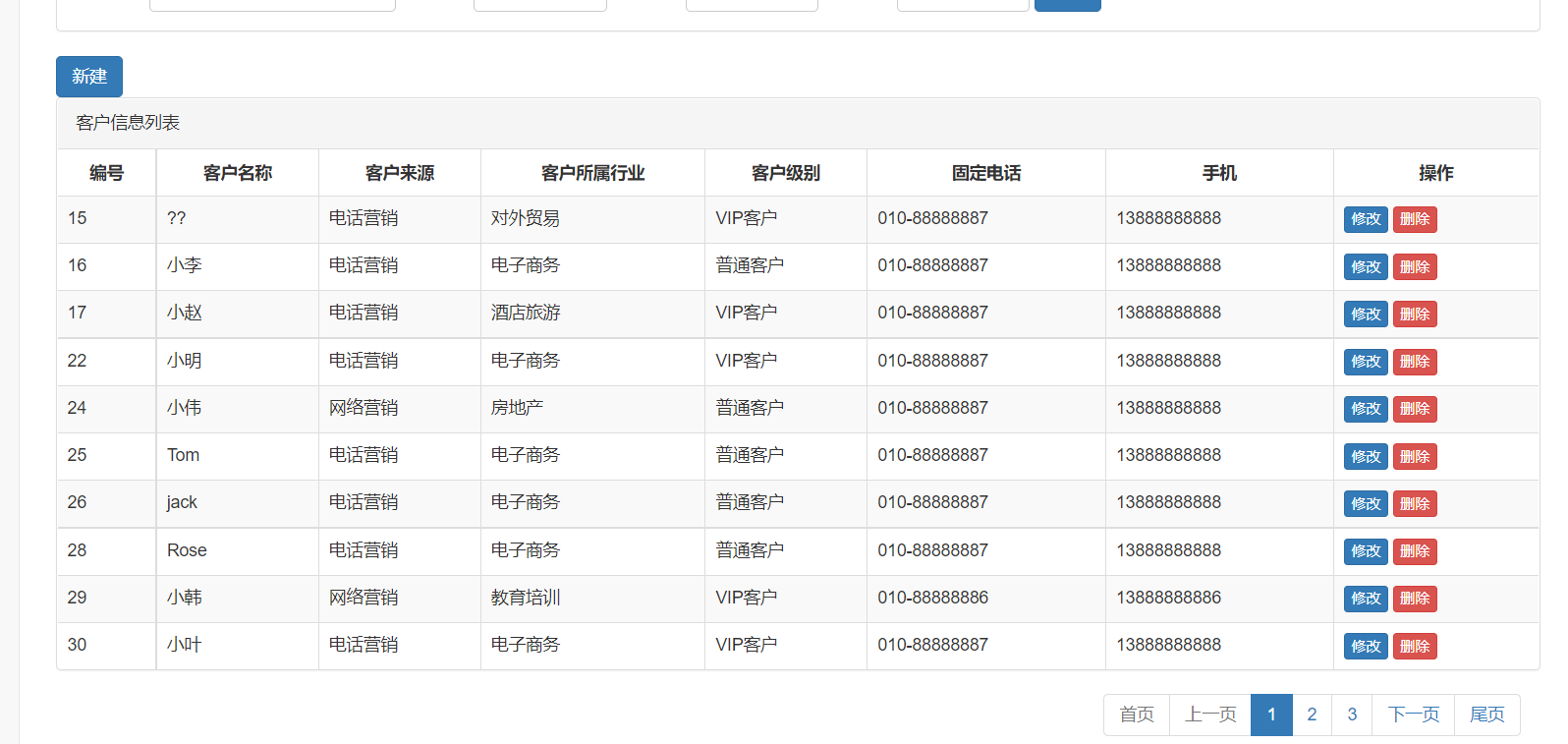


**根据名称查询按道理是会出现对应的信息，但是没有。**

**不同的用户登录，右上角显示不同的用户姓名**



**查询的时候可以进行分页显示，每页显示不同的数据**



**点击首页就跳到第一页**

****

**点击尾页就跳到最后一页，且如果所有的数据总共只有一页的话，那么上下翻页就不能使用，但是点击尾页之后出现了bug**

****

4. 总结

通过这学期的javaweb学习，我学到了很多的东西，比如MVC三层架构，三层的分工明确，将一个大项目转换成三层，每一层都处理它们对应的逻辑，特别方便。Mybatis可以使sql语句实现低耦合，而mybatis内部又可以进行嵌套的衔接，使用起来特别灵活，springmvc框架就可以实现面向切面变成，将每一层涉及到的东西都放在一个文件里面供它们使用，同时我也用到了servlet拦截器，这个可以按指定的方式拦截你想要拦截的请求，在项目设计期间，我也碰到了一些Bug，比如在根据客户名查询信息的时候就怎么也查询不到，而且修改了某一位客户信息之后，他的名字就变成了问号，而其它的字段却没有出现异常，我翻阅了很多资料但仍不能解决问题，我觉得应该是跟这个字段的设计有关系，因为其它的字段也都是中文，唯独这个才会乱码，这是其中一个bug，另一个bug就是当点击尾页的时候，前面出现了-1，这个bug可能是页面设计的时候出了问题，由于前端页面代码量庞大，不容易找到在哪里修改，虽然这个项目有一点瑕疵，但是大部分都能完美运行，并且可以学到很多知识，感谢老师这学期对我的指导！今后我会更加努力的学习java方便的相关知识的。

5.参考文献

[1]中国报告大厅.房屋租赁行业定义及分类,2015-12-08.

http://m.chinabgao.com/k/fangwuzulin/21100.html

[2]百度百科.O2O 模式,2020-02-10.

https://baike.baidu.com/item/O2O/8564117?fr=aladdin

[3]Casillas J C. O2O Business Patterns and Developing Strategies [J]. International

Business Review, 2015, 5(3): 465-479.

[4]百度百科.bootstrap,2020-04-10.

https://baike.baidu.com/item/Bootstrap/8301528

[5]百度百科.jquery,2019-10-18.

https://baike.baidu.com/item/jQuery/5385065?fr=aladdin