仲恺农业工程学院

**期末大作业**

**设计报告**

( 2020-- 2021年度第2学期)

名 称： Java应用技术

题 目： 客户关系管理系统的设计与实现

院 系： 计算科学学院

班 级： 信息与计算科学184

学 号： 201821314402，201821314409

学生姓名： 林华萍（50%）、王佳萍（50%）

指导教师： 吴东庆

设计周数： 2

成 绩：

2021年6月

**目 录**

[一、设计概要 1](#_Toc437157273)

[二、设计正文 2](#_Toc437157274)

[1 需求分析 2](#_Toc437157275)

[2 概要设计 4](#_Toc437157276)

[3 详细设计 14](#_Toc437157277)

[4.调试分析 20](#_Toc437157278)

[5.测试结果 20](#_Toc437157279)

[6.用户使用说明 32](#_Toc437157280)

[三、设计总结或结论 43](#_Toc437157281)

[参考文献 44](#_Toc437157282)

[附录（其他必要资料，不需要源代码） 44](#_Toc437157283)

# 一、设计概要

客户是公司最宝贵的资源，为了更好的发掘老客户的价值，并开发更多新客户，XX公司决定实施客户关系管理系统。希望通过这个系统完成对客户基本信息、联系人信息、交往信息、客户服务信息的充分共享和规范化管理；希望通过对潜在客户、客户开发过程的追踪和记录，提高新客户的开发能力；希望在客户将要流失时系统及时预警，以便销售人员及时采取措施，降低损失。并希望系统提供相关报表，以便公司高层随时了解公司客户情况。

目前。国内大多数企业的CRM系统还是几年前开发的基于C/S架构的老旧系统，这类系统有较多缺点：功能模块及处理流程繁琐，信息处理效率底下、数据保密性差，必须安装软件才能使用，为了能够在所有操作系统都可以安全、高效的使用，必须为每一种操作系统做适配开发，这不仅不方便随时随地的查看管理客户信息，还使得系统的开发、维护变得更加复杂。这种系统会浪费大量的物力以及人力资源元，已无法适应时代的发展。

又由于传统客户关系管理系统基本采取人工记录的方式添加信息，人工记录方法存在以下问题：

①部门人员信息沟通不及时，会有信息混乱现象发生；

②岗位工作人员变化快，收录的信息格式多种多样，信息收录质量没有统一标准；

③由于数据零散，在收集资料与查询时出现困难；

④人工记录导致数据信息不能及时更新，难以保证数据的时效性。

综上，如何高效的存储客户信息、管理信息和处理信息是企业需要亟待解决的问题，在此基础上对客户信息数据进行分析，找出潜在的问题即可避免企业客户流失。本设计采用B/S结构，从而保证了系统的易维护性和易升级性。采用三层体系结构，整个系统结构层次分明，结构清晰，使得整个系统的维护工作更加简单方便。

本设计采用的后端设计框架为Spring Boot+MyBait，前端用Vue，其中 UI使用Element，网络请求使用Axio，数据库使用MySQL，编译工具使用的是IDEA，作图工具使用的是EA，用到的数据结构有数组、列表、哈希等等。

需要实现的功能模块有：按区域建立客户档案、客户分类管理包含增删查改功能、客户服务管理主要是跟进服务的记录的管理和客户投诉、员工绩效考核，对考核结果进行查询、客户访谈服务包括需求计划录入和汇总以及报告管理、客户合同管理、客户销售分析（交易额按时间分析，趋势分析和客户间差异化分析）等几大功能模块。以此来做到较好的数据收集管理，同时也能做到合理的数据统计、分析和展现。

可行性分析：本系统采用较为成熟的技术 B/S 模式，由浏览器、服务器和数据库组成，该设计过程中所用的开发技术比较常见，之前对这些开发技术有了一定的理解和使用，对于不熟知的技术在大量的图书或者网络上都可以进行学习。所需硬件也非常简单；该设计涵盖了层次化的特点。系统界面简洁明了，在系统运行时，管理员的职能就是根据自己的权限去完成各个模块和各个模块之间的实现和应用。系统采用了导航菜单的样式，可实现数据库—角色的交互。简洁、方便和大方是其操作界面的主要特点。普通的开发者和管理者在不需要掌握数据库知识的前提下也可进行功能模块的选择。该系统在操作上很容易被使用的。本系统由自己开发调试，所使用硬件设备为自己的计算机，软件都是免费下载使用的，所以在经济上是可行的。该系统的设计存在一定难度，在分析理解功能模块的部分可能不够符合需求。对于一些新的技术比如之前不熟悉的JFreeChart技术和Jasperreports打印技术，需要一定时间学习和使用。不过整体来说，实现该系统是没问题的，不管是技术上还是经济上都是可行的。

# 二、设计正文

## 1 需求分析

* 1. 功能需求分析

客户关系管理系统开发最重要的是实现客户信息的规范化管理、自动化管理和系统化管理，节约企业的人力资源成本、管理成本和时间周期，以提高整体工作效率，结合企业实际需求，这次设计的系统主要包含的功能模块有如下几点。

客户管理：负责管理客户信息，包括用户信息的增加、删除、修改和查询，其中是根据区域建立的客户信息，可以查询全部客户信息，也可以按字段进行查询，实现对用户的分类管理。这里客户分类管理的类型，可以是按区域也可以看客户对应的状态进行分类管理。

客户服务管理和客户投诉：该管理系统的核心，主要功能是把客户与客户相关的任务（比如说客户下的订单或者客户的投诉）以及负责该客户的员工相关联，在系统中以跟进历史服务来体现，以及投诉信息。对于每位客户能做到专人负责，对每位客户的相关信息等进行统一的流程化管理。

员工以及经理的绩效考核：设计好考核标准，根据不同的表现打分，同时也根据投诉情况和员工接单情况设置相应的加分项和扣分项，对考核信息进行维护以及查询考核结果。

客户访谈服务：这个模块主要对应的是需求计划的管理以及访谈日程表的维护，对于不同的服务，进行需求计划的录入，需求计划的汇总等。

客户合同管理：服务分为几个不同的服务状态，其中一个服务状态就是签订合同，合同管理就是包含，对合同的录入以及查询等操作。

客户销售分析：主要有四种分析，交易额按时间分析，员工绩效排名，客户间差异化分析，这里主要分析的是不同工作来源的差异化。还有一个是趋势分析。可以据此看出对应月份客户来源的趋势。

根据功能需求分析，得到分两大管理模块的数据流图，如图1、图2所示

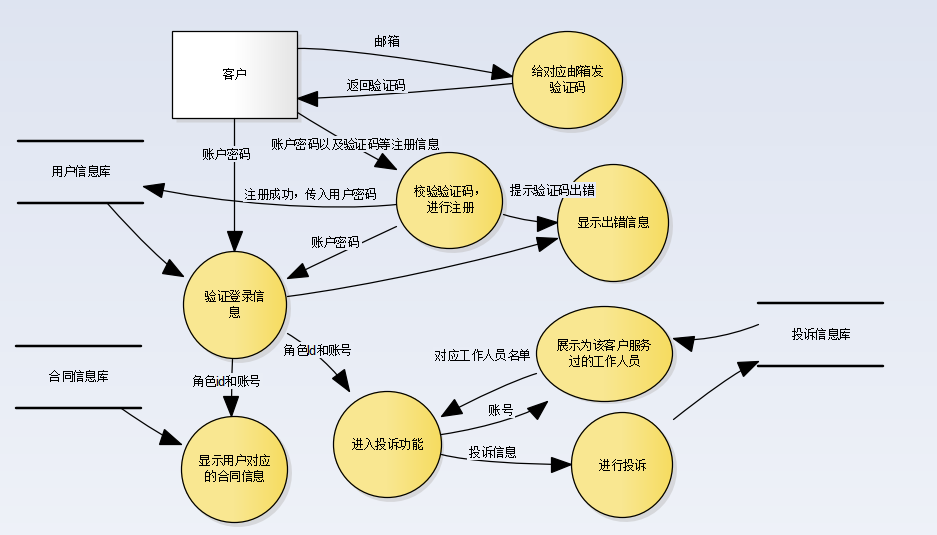


图1 客户模块

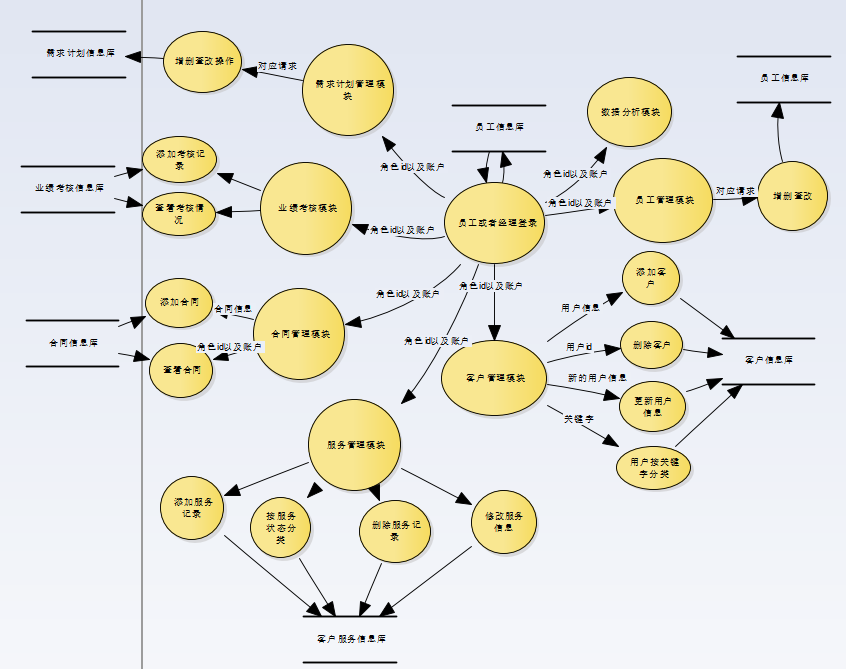


图2 员工及经理模块

* 1. 性能需求分析

安全性：在网络非常发达的社会，我们更加注重信息的安全，尤其是客户信息的安全，本系统采用的是前后端分离实现，这样就需要考虑安全性问题，需要设置拦截器，本系统设置了token,发送请求的时候都需要带上请求头部这样才能访问，这样可以能够提高安全性能。

可靠性：它包含稳定性和可用性，要保证管理系统的稳定性就是保证系统出错率要低，页面响应速度要快，注意好这些问题，系统才能做到较好的用户体验。

## 2 概要设计

2.1总体功能设计

本系统主要有三个角色设计，分别是客户，员工，经理，根据不同的角色，设置不同的权限。整体包含的功能模块有：客户管理、服务管理、合同管理、员工管理、需求计划管理、业绩考核管维护、数据统计分析等7个功能模块。可满足日常客户资源维护、销售数据分析、潜在和有价值的客户分析等需求。对各个功能模块有相对应的增删查改等操作。下面就三种用户角色对应的功能模块进行分析。

2.1.1客户功能模块

客户可进行注册和登录操作，然后进入系统首页，对于客户有对应两个功能模块进行操作，一个是合同管理模块，客户仅限于可以查看自己对应的合同信息。另一个是投诉功能，可以选择投诉为自己服务过的员工，匿名投诉。该客户功能模块的结构图，如图3所示。

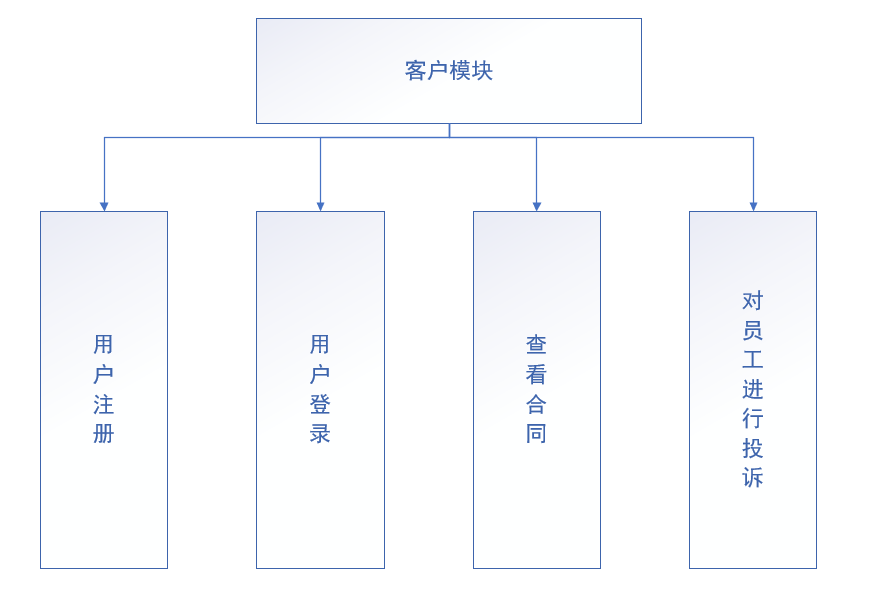


图3 客户功能模块结构图

2.1.2员工功能模块

员工可进行登录和注册操作，同样进入系统首页，员工进来可进行客户管理、服务管理、合同管理、需求计划管理、业绩考核维护、数据统计分析等6大模块进行操作，只是部分模块的一些功能无法实现。对于客户管理、服务管理以及需求计划管理，都能查到对应跟自己有关的数据，（不是全部数据），增删查改也是对自己有关的数据进行处理。对于合同管理自己只有查看对应的数据和添加合同，业绩以及数据分析部分也是只有查看功能。该员工功能模块的结构图，如图4所示。

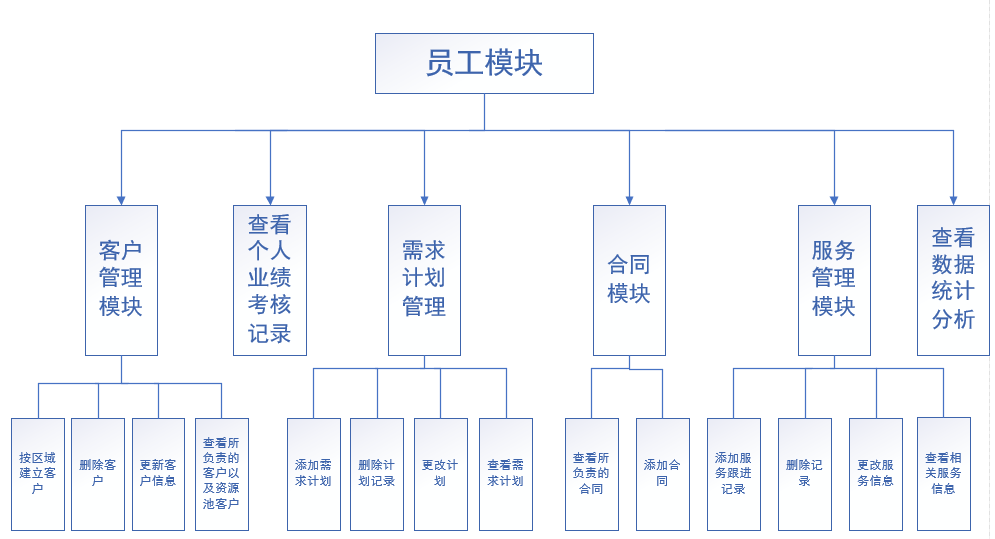


图4 员工功能模块的结构图

2.1.3经理功能模块

经理拥有员工对应的所有数据，只是权限更多了些，所有查看的数据，可以查看全部人的信息，相较于员工功能模块还多了一个小的功能模块，就是对员工的管理，增删查改，同时在业绩考核那一块，也多了对每个员工进行考核的功能。经理功能模块的结构图，如图5所示。

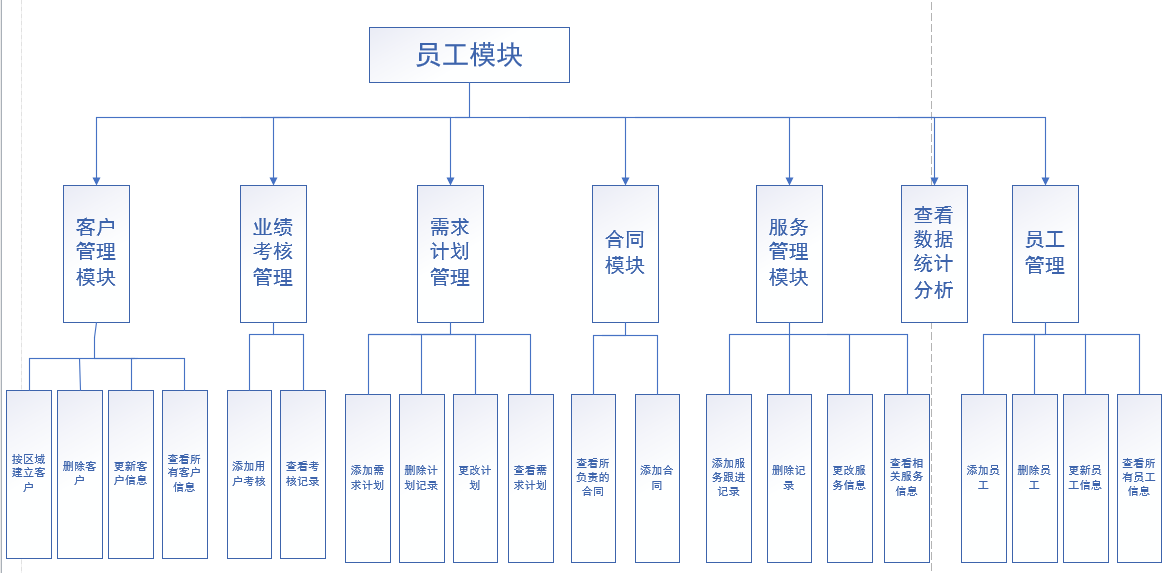


图5 经理功能模块的结构图

2.2 数据库设计

2.2.1 数据库概念模型设计

本系统用MySQL配合Navicat制作，对该CRM管理系统进行分析可知该系统包含8个实体。分别有客户、员工、服务、业绩、合同、角色、计划、投诉等实体，各个实体的主要属性如下图所示：

1. 客户实体，包含的属性有客户id、客户名、密码、工作类型、性别、年龄、所属类型、电话、地区等等

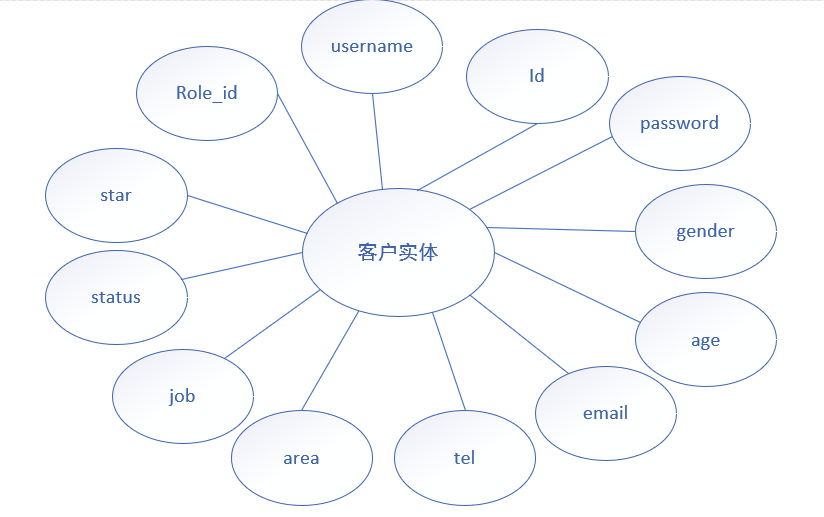


图6 客户实体

1. 员工实体包含的属性有员户id、员户名、密码、部门、性别、年龄、角色id等等

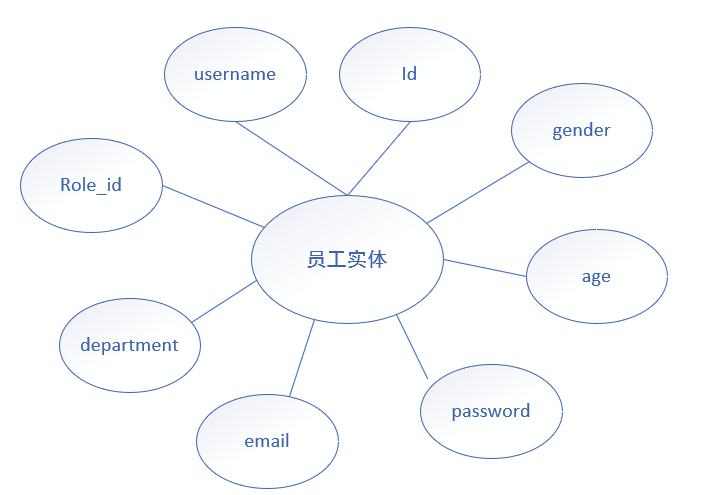


图7 员工实体

1. 服务实体，其包含的属性有服务开始时间、服务内容、服务结束时间、服务类型、服务结果、客户名、服务进程、服务负责人等等

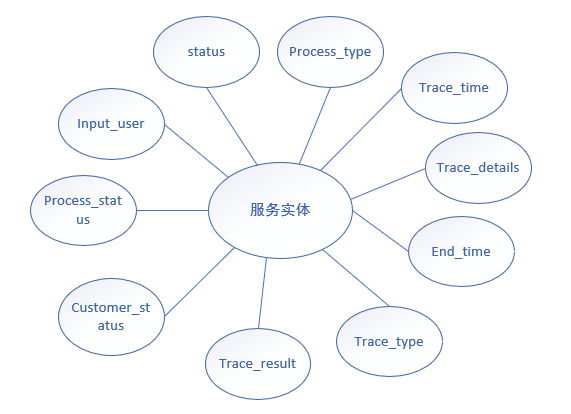


图8 服务实体

1. 业绩实体包含的属性有：工作能力、工作态度、环境行为、纪律、加分项、扣分项、评定时间、评定人、被评定人等属性。

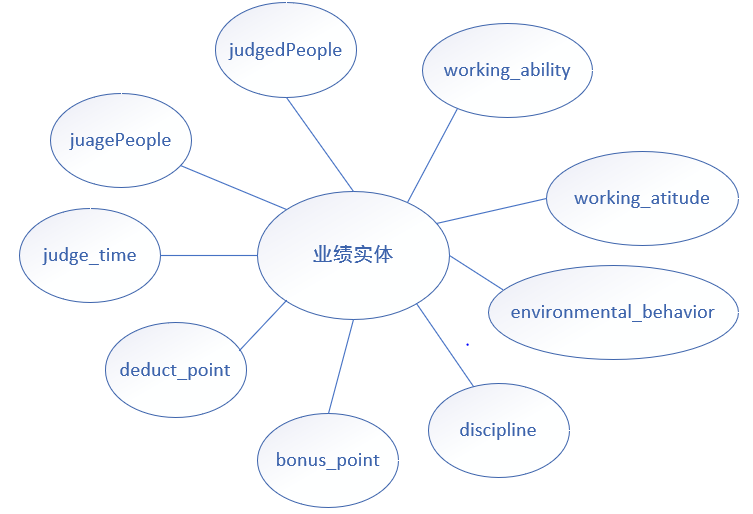


图9 业绩实体

1. 合同实体，包含的属性有：合同id、服务名称、甲方、乙方、负责人、创建时间、截止时间、交易金额、支付方式、金额状态、职业类型、跟进服务等等

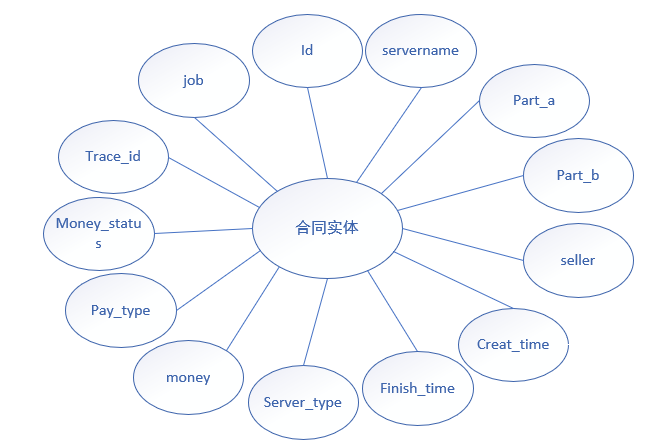


图10 合同实体

1. 计划实体包含的属性有，服务id、服务名称、标题、沟通方式、计划细节、开始时间、截止时间、对应服务id

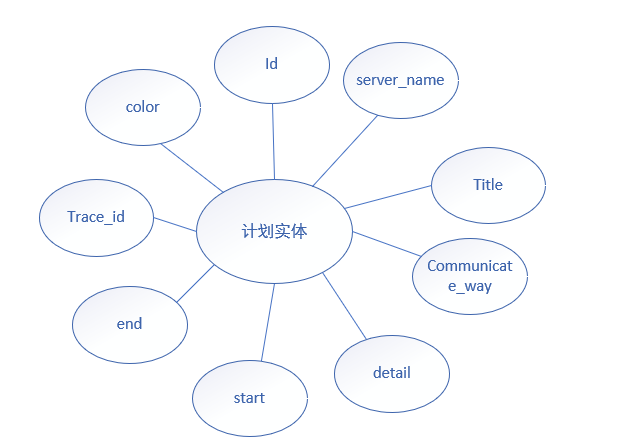


图11 计划实体

1. 投诉实体的属性有：投诉id、客户名、投诉内容、投诉对象、投诉时间、分值等

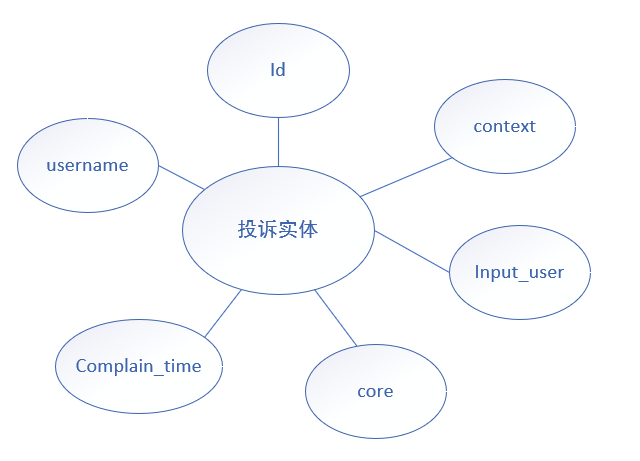


图12 投诉实体

2.2.2 数据库逻辑结构设计

本系统的数据库名称为crm\_system该数据库中包含8个表，主要的表有7个。分别有客户表、员工表、服务表、需求计划表、投诉表、合同表、业绩考核表等，各表的数据结构详细设计如下所示：

表1 user数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 客户id |
| username | varchar | 64 | Not null | 否 | 客户名 |
| password | varchar | 255 | default | 否 | 客户密码 |
| gender | varchar | 8 | default | 否 | 性别 |
| email | varchar | 32 | default | 否 | 邮箱 |
| age | int | 11 | default | 否 | 年龄 |
| tel | varchar | 128 | default | 否 | 电话 |
| seller | varchar | 11 | default | 否 | 负责人 |
| input\_user | varchar | 11 | default | 否 | 创建人 |
| area | varchar | 255 | default | 否 | 区域 |
| job | varchar | 255 | default | 否 | 工作类型 |
| source | varchar | 255 | default | 否 | 客户来源 |
| input\_time | date | 0 | default | 否 | 录入时间 |
| status | varchar | 255 | default | 否 | 客户状态 |
| star | int | 1 | default | 否 | 星标标签 |
| role\_id | int | 11 | default | 否 | 角色id |

表2 employee数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 员工id |
| username | varchar | 64 | Not null | 否 | 员工名 |
| password | varchar | 255 | default | 否 | 员工密码 |
| gender | varchar | 8 | default | 否 | 性别 |
| email | varchar | 32 | default | 否 | 邮箱 |
| age | int | 11 | default | 否 | 年龄 |
| department | varchar | 255 | default | 否 | 部门 |
| hire\_date | date | 0 | default | 否 | 注册时间 |
| input\_user | varchar | 11 | default | 否 | 创建人 |
| state | varchar | 255 | default | 否 | 状态 |
| role\_id | int | 11 | default | 否 | 角色id |
| performance | double | 255 | default | 否 | 最新业绩 |

表3 trace数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 服务id |
| trace\_time | date | 0 | default | 否 | 创建时间 |
| trace\_details | varchar | 255 | default | 否 | 服务细节 |
| end\_time | date | 0 | default | 否 | 结束时间 |
| trace\_type | varchar | 255 | default | 否 | 服务类型 |
| trace\_result | varchar | 255 | default | 否 | 结果 |
| customer\_name | varchar | 11 | default | 否 | 客户名 |
| process\_status | int | 0 | default | 否 | 服务状态 |
| input\_user | varchar | 255 | default | 否 | 创建人 |
| status | varchar | 255 | default | 否 | 客户类型 |
| type | varchar | 255 | default | 否 | 交流方式 |

表4 plan数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 计划id |
| server\_name | varchar | 255 | default | 否 | 服务名称 |
| title | varchar | 255 | default | 否 | 标题 |
| communicate\_way | varchar | 255 | default | 否 | 沟通方式 |
| detail | varchar | 255 | default | 否 | 细节 |
| start | date | 0 | default | 否 | 开始时间 |
| end | date | 0 | default | 否 | 结束时间 |
| trace\_id | int | 11 | default | 否 | 服务id |
| color | varchar | 255 | default | 否 | 颜色 |

表5 complain数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 投诉id |
| username | varchar | 255 | default | 否 | 默认匿名 |
| context | varchar | 255 | default | 否 | 投诉原因 |
| input\_user | varchar | 255 | default | 否 | 投诉对象 |
| core | int | 255 | default | 否 | 分值 |
| complain\_time | date | 0 | default | 否 | 时间 |

表6 contract数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 合同id |
| servername | varchar | 255 | Not null | 否 | 合同名 |
| part\_a | varchar | 255 | default | 否 | 甲方 |
| part\_b | varchar | 255 | default | 否 | 乙方 |
| seller | varchar | 255 | default | 否 | 负责人 |
| creat\_time | date | 0 | default | 否 | 创建时间 |
| finish\_time | date | 0 | default | 否 | 结束时间 |
| server\_type | varchar | 255 | default | 否 | 服务类型 |
| money | int | 11 | default | 否 | 金额 |
| pay\_type | varchar | 255 | default | 否 | 支付方式 |
| money\_status | varchar | 255 | default | 否 | 金额状态 |
| trace\_id | int | 11 | default | 否 | 服务id |
| job | varchar | 255 | default | 否 | 工作类型 |

表7 judge数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 是否为空 | 是否主键 | 说明 |
| id | int | 11 | Not null | 是 | 业绩id |
| working\_ability | int | 10 | default | 否 | 工作能力 |
| work\_attitude | int | 10 | default | 否 | 工作态度 |
| environmental\_beh | int | 10 | default | 否 | 环境行为 |
| discipline | int | 10 | default | 否 | 纪律 |
| bonus\_point | int | 10 | default | 否 | 加分项 |
| deduct\_point | int | 10 | default | 否 | 扣分项 |
| performance | double | 255 | default | 否 | 总成绩 |
| judge\_time | date | 0 | default | 否 | 评定时间 |
| judge\_people | varchar | 255 | default | 否 | 评定人 |
| judged\_people | varchar | 255 | default | 否 | 被评定人 |

## 3 详细设计

**3.1功能模块算法与数据结构分析**

本系统整体包含的功能模块有：客户管理、服务管理、合同管理、员工管理、需求计划管理、业绩考核管维护、数据统计分析等7个功能模块。对各个功能模块有相对应的增删查改等操作。用到的数据结构有，数组，列表，哈希等等。下面对几个重要的功能模块进行分析。

3.1.1 客户管理模块

客户管理模块主要操作有，按区域建立客户，查看用户信息，按id删除客户，修改用户信息等等。这一模块用到的数据结构，主要是用到List处理分页查询。前期用模糊查询到的数据或者直接查询全部数据之后，根据页码大小以及所在页码进行分页。这里是对查询到的对象用List包装，然后根据页码，得到起始下表位置，获取相对应的子列表sublist。以及用到Map和HashMap，String、double两个类型对应的数组结构。来处理封装按照5种客户状态查询出来的数据得到不同用户状态的占比的数据返回给前端需要的操作数据。首先构建一个Map对应关键字和数据类型分别未string和Object。然后创建客户状态的字符串数组，创建浮点数数组来存放占比。一个简单的for循环，根据5中类型查到不同数据得到不同的占比，对应关键字和数组放入map中，进行封装。相关代码实现如下：

public List<User> queryusersByArray(Integer pageNo, Integer pageSize, String keyword) {  
 List<User> user = userMapper.queryusersByArray(keyword);  
*// 从第几条数据开始* int firstIndex = (pageNo - 1) \* pageSize;  
*// 到第几条数据结束* int lastIndex = pageNo \* pageSize;  
 if(lastIndex>user.size()){  
 lastIndex=user.size();  
 }  
 return user.subList(firstIndex, lastIndex);

public Result getTypeData(){  
 Map<String,Double> map=new HashMap<>();  
 String[] value={"流失","开发失败","潜在客户","正式客户","资源池客户"};  
 double[] num=new double[5];  
 int all=userService.selectNum(null);  
 for(int i=0;i<value.length;i++){  
 num[i]=userService.getTypeData(value[i]);  
 map.put(value[i],num[i]/all\*100);  
 }  
 return ResultUtil.*success*("true",map);  
}

3.1.2 员工管理模块以及服务模块

员工管理模块主要操作有，添加员工信息，模糊查看员工信息，按id删除员户，修改用户信息等等。服务管理模块也是具有增删查改的功能。这两个模块用到的数据结构同上面一样，主要是用到List处理分页查询。相关代码实现如下：

public List<Employee> selectEmployeePage(QueryDTO queryDTO) {  
 List<Employee> employees = employeeMapper.selectEmployeePage(queryDTO.getKeyword());  
*// 从第几条数据开始* int firstIndex = (queryDTO.getPageNo() - 1) \* queryDTO.getPageSize();  
*// 到第几条数据结束* int lastIndex = queryDTO.getPageNo() \* queryDTO.getPageSize();  
 if(lastIndex>employees.size()){  
 lastIndex=employees.size();  
 }  
 return employees.subList(firstIndex, lastIndex);  
 }

List<Trace> traces = traceMapper.selectTracePage(queryDTO.getKeyword(),queryDTO.getFromTime(),queryDTO.getToTime(),queryDTO.getUsername(),queryDTO.getRole\_id());  
*// 从第几条数据开始* int firstIndex = (queryDTO.getPageNo() - 1) \* queryDTO.getPageSize();  
*// 到第几条数据结束* int lastIndex = queryDTO.getPageNo() \* queryDTO.getPageSize();  
 if(lastIndex>traces.size()){  
 lastIndex=traces.size();  
 }

3.1.3 数据统计分析管理模块

这一模块主要是对数据的一个处理和包装，根据前端需要的数据类型，对查找到的数据进行封装，返回给前端进行展示。

首先第一部分是要得到近六个月份，每个月份对应的销售金额，把月份和对应金额的数据进行封装返回给前端。这里用到的数据结构是String、double两种类型的数组以及HashMap，首先根据系统当前时间获得该时间对应的月份，然后用for循环得到近6各月份第一天的日期。然后再用for循环得到每个月份对应的金额，根据关键字和数据类型放入map进行封装返回数据。对应代码实现如下：

public Result getMoneyData(){  
 Map<String,Object> map=new HashMap<>();  
 Calendar cale = null;  
 SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");  
 cale = Calendar.*getInstance*();  
 cale.add(Calendar.*MONTH*, 0);  
 cale.set(Calendar.*DAY\_OF\_MONTH*, 1);  
 String[] date=new String[6];  
 date[5]=format.format(cale.getTime());  
 for(int j=1;j<date.length;j++){  
 try {  
 cale.setTime(format.parse(date[6-j]));  
 cale.add(Calendar.*MONTH*, -1);  
 date[5-j]=format.format(cale.getTime());  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
  
 }  
 double[] money=new double[6];  
 for(int i=0;i<money.length;i++){  
 money[i]=contractService.selectMoneyByDate(date[i]);  
 }  
 map.put("name","sfd");  
 map.put("xData",date);  
 map.put("yData",money);  
 return ResultUtil.*success*("查询成功",map);  
}

第二个需要处理的数据有，根据不同的行业得到对应行业数量。用到的数据结构是List<Map<String,Object>这样的类型，以及String类型的数组。因为前端要求封装方式是list类型，里面又是一个个的对象，所以采用这样的封装方式。对应代码实现如下：

String[] jobs={"互联网","金融","教育","服务","餐饮"};  
List<Map<String, Object>> list = new ArrayList<Map<String, Object>>();  
for(int i=0;i<jobs.length;i++){  
 Map<String,Object> map=new HashMap<>();  
 map.put("value",userService.selectByJob(jobs[i]));  
 map.put("name",jobs[i]);  
 list.add(i,map);  
}

第三个相对简单，只需要用到map数据类型，直接将查询到的数据对象放入map中即可。代码实现如下

Map<String,Object> map=new HashMap<>();  
map.put("xData",employeeService.getUsername());  
map.put("yData",employeeService.getPerformance());

最后一个数据处理相对复杂些，需要得到的数据是，不同行业，对应近六个月的销售金额。这个的处理，前期获得近六个月份的数据String数组类型，然后不同行业也是String数组类型，double数组存放对应金额。接着还用到了List、Map等类型，双重for循环对数据进行处理。具体实现代码如下：  
String[] jobs={"互联网","金融","教育","服务","餐饮"};  
int temp = 0;  
List<DataDTO> list=new ArrayList<>();  
Map<String,Object> map=new HashMap<>();  
for(int i=0;i<jobs.length;i++){  
 DataDTO dto=new DataDTO();  
 double[] dd=new double[6];  
 for(int j=0;j<date.length;j++){  
 dd[j]=contractService.selectMoneyByDateAndJob(jobs[i],date[j]);  
 }  
 dto.setData(dd);  
 dto.setName(jobs[temp]);  
 temp = temp +1;  
 list.add(dto);  
}  
map.put("leg",jobs);  
map.put("xData",date);  
map.put("yData",list);

**3.2功能模块实现细节与接口细节体现**

该系统的设计是前后端分离的设计，前端向后端发送数据请求，后端根据传入的参数，进行逻辑判断以及对数据进行封装返回给前端，以下是设计期间，对于不同的功能模块，前端与后端间接口的实现细节，对应每个模块，前端发送传入什么数据，对应后端处理之后返回的数据进行应该汇总，如表8接口汇总表

表8 接口汇总表

|  |  |  |
| --- | --- | --- |
| 传入参数 | 接口 | 返回结果 |
| email | <http://localhost:8082/send> | 发送成功与否 |
| username,password,email,code,role\_id | <http://localhost:8082/checkCore> | 验证是否通过 |
| username、password、role\_id | <http://localhost:8082/api/login> | 是否成功登录 |
| pageNo、pageSize、keyword | <http://localhost:8082/user/list> | 按关键字查询的所有客户对象 |
| 客户对应的信息 | <http://127.0.0.1:8082/user/add> | 添加成功与否 |
| id | <http://127.0.0.1:8082/user/delete> | 删除成功与否 |
| 整个对象对应的信息 | <http://localhost:8082/user/update> | 修改成功与否 |
| 无 | <http://localhost:8082/user/selectStar> | 返回对应星标客户 |
| role\_id,username,pageNo,pageSize | <http://localhost:8082/trace/list> | 返回对应角色可看到的所有服务对象 |
| 服务对像的相关信息 | <http://localhost:8082/trace/add> | 添加成功与否 |
| id | <http://localhost:8082/trace/deleteByI> | 删除成功与否 |
| 整个对象新的数据 | <http://localhost:8082/trace/update> | 更新成功与否 |
| username | <http://localhost:8082/selectInput_user> | 与该用户交接过的负责人名单 |
| 投诉的相关信息 | <http://localhost:8082/complain/add> | 添加成功与否 |
| 无 | <http://localhost:8082/plan/selectAll> | 计划表对应所有信息 |
| 整个对象新的数据 | <http://localhost:8082/plan/updatePlan> | 更新成功与否 |
| id | <http://localhost:8082/plan/deleteById> | 删除成功与否 |
| 计划的相关信息 | <http://localhost:8082/plan/addPlan> | 添加成功与否 |
| date,username | <http://localhost:8082/show> | form1对应投诉记录，form2对应接单记录 |
| 业绩考核内容 | <http://localhost:8082/judge/add> | 添加成功与否 |
| 无 | [http://localhost:8082//judge/selectAll](http://localhost:8082/judge/selectAll) | 返回所有数据 |
| 合同相关数据 | <http://localhost:8082/contract/add> | 添加成功与否 |
| role\_id,username,pageNo,pageSize | <http://localhost:8082/contract/list> | 返回对应角色可看到的所有合同对象 |
| id | <http://localhost:8082/selectByTrace_id> | 返回对应id的合同内容 |
| 无 | <http://localhost:8082/getMoneyData> | 近6月总的销售额 |
| 无 | <http://localhost:8082/getJudgeData> | 绩效前6对应员工的绩效 |
| 无 | <http://localhost:8082/getJobData> | 不同行业的数量 |
| 无 | <http://localhost:8082/getJobMoneyData> | 不同行业不同月份对应交易额 |
| role\_id,pageNo,pageSize,keyword | <http://localhost:8082/employee/list> | 返回对应角色可看到的所有员工对象 |
| 员工信息 | <http://localhost:8082/employee/add> | 添加是否成功 |
| 改动后的员工信息 | <http://localhost:8082/employee/update> | 更新是否成功 |
| Id | <http://localhost:8082/employee/deleteById> | 删除员工是否成功 |
| 无 | <http://localhost:8082/user/getTypeData> | 5客户类型对应数量 |
| 无 | <http://localhost:8082/trace/selectTop5ByDate> | 服务表最新的5条数据 |

几个重要模块对应的测试用例，对应输入内容和期望输出的内容。

1. 注册登录注册模块，如表9

表9 登录注册测试用例表

|  |  |
| --- | --- |
| 输入内容 | 期望输出 |
| Email | 验证码是否发送成功 |
| username password core role\_id | 用户名是否重复，验证码是否正确，注册是否成功 |
| username password role\_id | 用户名是否重复，密码是否正确，登录成功与否 |

1. 客户管理模块，如表10

表10 客户管理测试用例表

|  |  |
| --- | --- |
| 输入内容 | 期望输出 |
| 对应用户的信息，username,age,area,tel等等 | 用户添加成功与否 |
| 用户id | 用户删除成功与否 |
| pageNo，pageSize,keyword | 按关键字查询到的所有客户对应页码的对象 |
| 修改后的用户信息 | 用户信息更新是否成功 |

1. 合同管理模块，如表11

表11 合同管理测试用例表

|  |  |
| --- | --- |
| 输入内容 | 期望输出 |
| 新的合同相关信息，比如甲方乙方等信息 | 合同添加是否成功 |
| pageNo，pageSize,role\_id | 对应角色可看到的合同信息 |

1. 业绩考核管理模块，如表12

表12 业绩考核测试用例表

|  |  |
| --- | --- |
| 输入内容 | 期望输出 |
| 对员工各项考核指标所打的分数，以及评定人等信息 | 添加考核记录是否成功，返回该员工总成绩 |
| 员工id | 对应员工业绩考核记录 |

下面以两个主要功能模块客户管理模块和注册登录模块为例，分析得到这两个功能模块对应的流程图，如图13所示

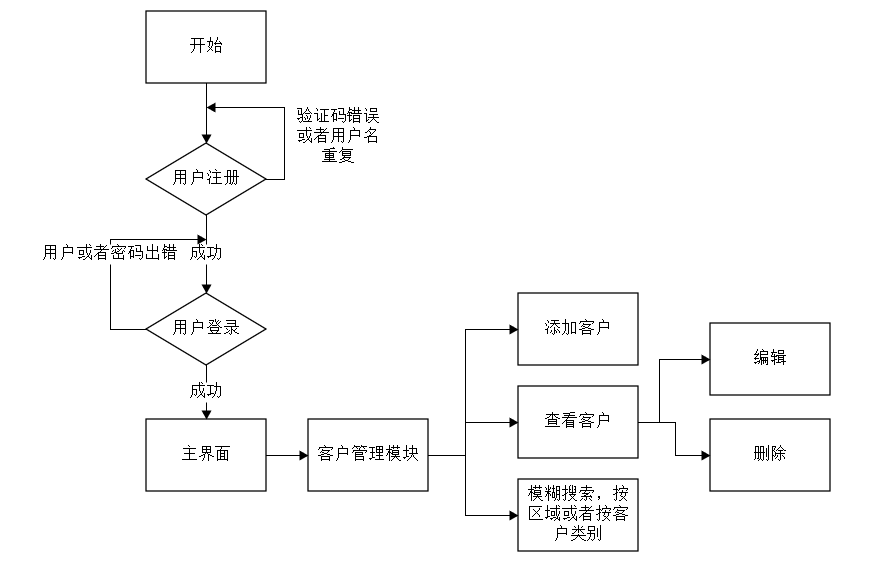


图13 客户管理模块流程图

## 4.调试分析

### 4.1 跨域问题

本项目使用了前后端分离的技术，所以涉及到了跨域的问题，系统会提示been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource，这表示前后端的数据交互出现了跨域的问题，引起这个问题的原因主要是浏览器出于安全考虑，限制访问本站点以为的资源，这里我们可以通过多种方式进行解决跨域，其中包括JSONP、CORS、反向代理等。

在本次设计中使用了CORS来解决跨域问题，主要是当跨域发送请求时，浏览器会自动在请求头上添加 Origin 字段，用于说明本次请求来自哪个源（协议 + 域名 + 端口）。当后端发现发送该请求的页面所在域不在白名单中，则会返回一个正常的HTTP回应。浏览器发现，这个回应的头信息没有包含Access-Control-Allow-Origin字段，就知道出错了，从而抛出一个错误，被 XMLHttpRequest 的 onerror回调函数捕获。如果在白名单中，则会在响应头添加Access-Control-Allow-Origin字段，即跨域成功，数据能成功的获取。

### 4.2 Vue页面刷新问题

在Vue2中，其对数据的双向数据绑定是利用ES5 的一个 API Object.definePropert()对数据进行劫持，结合发布订阅模式的方式来实现的，这样就会存在数组长度或者对象属性的增加或者修改存在不能及时更新的问题，在页面中也会出现无法渲染正确数据的问题。在本次设计的员工、客户等管理界面，对其增加或者删除操作都会存在无法实现页面刷新的问题，这里想到了两个解决办法，第一种是对数据进行深拷贝，在对应的（副本）数据上修改后再重新对页面进行赋值；第二种是使用Update来强制更新数据，本次设计使用了第二种方式，及时的通知界面去更新页面的数据以实现数据的更新。

### 4.3 axios传参问题

对于前期数据的交互，出现了前端发送带参数的axios请求后端接受到的数据为null的情况，这种情况是因为前端发送的数据格式与后端接受的数据格式不一致，这里有两种解决方法，都是实现对前端参数进行封装，第一种是利用$qs对参数进行序列化处理供后端接收，第二种是发送FromData格式的数据来实现后端对传递参数的获取，本次设计采用的是第二种解决方式。

### 4.4 后端

前期使用Spring Boot导入依赖经常出错。几乎每次运行都会出错，经查找，有时是网络问题导致，有时是依赖的版本冲突问题，有时是存放依赖的文件夹里存在空的目录所致，刚开始就是边学习边改正，遇到导入依赖错误问题，依据错误提示信息进行相应改正，有时只是没有刷新导致，这刷新Maven可以解决，有时则需要点击complie重新注入依赖来解决问题。

token由于这是新学的东西，对于这个还不是很熟悉，百度学习了这个之后，调整了代码，发现还是运行不了，原来是对这个点不是很懂，继续百度学习，发现后端在需要token验证的时候方法上需要添加注解@token。

查询全部数据存在问题，根据不同的角色，查出对应不同的数据，刚开始写的分页查询接口是按传入的页码和关键字进行查询，调用了MyBatis封装好的分页查询会出现一个问题，就是它并不是按照先查询了全部信息后再进行分页，而是在这个页码上有没有数据，这样会出现的问题就是，可能第一个页没有数据，第二页有数据这样的情况，后面自己用List来封装，先根据关键字查询，返回一个List，然后再以传入的页码，和大小获取subList。这样就不会出现刚才的问题。

对于其他查询操作也会出现各种问题，比如当查询每个人对应的合同的时候，其实可能会对应多个数据，如果返回的类型resultType还是设计成一个对象的话，就会报错，后面调试才知道，错误提示说预期返回回数据应该为1，而我是多条数据。后面的处理也是把返回类型包装成一个List，里面包含多个对象。还有一个是判断查询的数据是否为空， 对于返回值是为空的情况需要做判断，前期没有判断，会直接报错，然后要判断是否为空，对于List类型的，不能直接List==null来判断，而是要以List.size==0来判断。根据是否为空，包装不同的消息返回给前端，这样才能较好的处理，有较好的用户体验。

还有一个问题就是Spring Boot集成MyBatis，有一些固定约定，前期在属性命名方面，我在数据库和类上都采用的驼峰式命名，发现数据库的操作会报错，提示要是下划线的形式命名，我就一并都改成两个单词间用下划线的方式连接的命名方式，不再报错，但是到后期才发现数据库查询操作是查询不到对应的这个数值的，经过百度和各种测试才发现。原来在类中，对应的属性值是需要以驼峰式命名的，在数据库则是以下划线的形式。这些应该算是固定的了，不然就要用注解去批注别名。

## 5.测试结果

### 5.1 注册、登录功能

一、注册测试

1、测试数据：

email：2466921236@qq.com

username：whh

password:111

2、测试结果：





（二）登录测试

1、测试数据：

email：2466921236@qq.com

username：whh

password:111

2、测试结果：

注册结果：

注册失败：用户名重复

注册成功：返回登录界面



### 5.2 客户管理功能

一、客户添加测试

1、测试数据：

创建客户：

客户名称：www

客户行业：金融业

客户区域：华北地区-天津

客户状态：流失

创建时间：2021-07-19

客户邮箱：2466xxx@qq.com

星标客户：是

客户来源：线上



1. 测试结果：

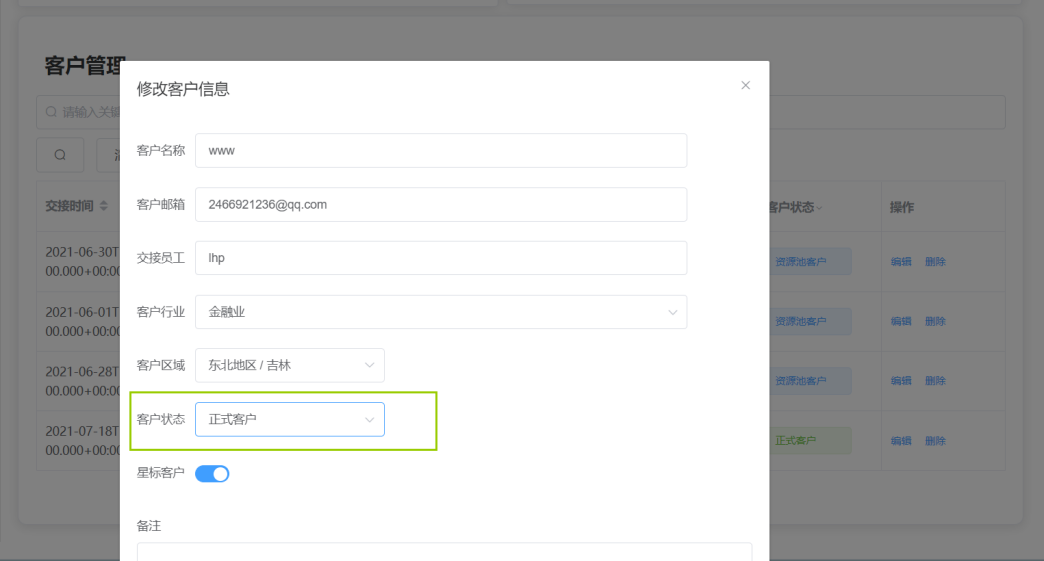
在客户管理面板可以查看到对应添加的客户信息



1. 客户编辑测试

1、测试数据：

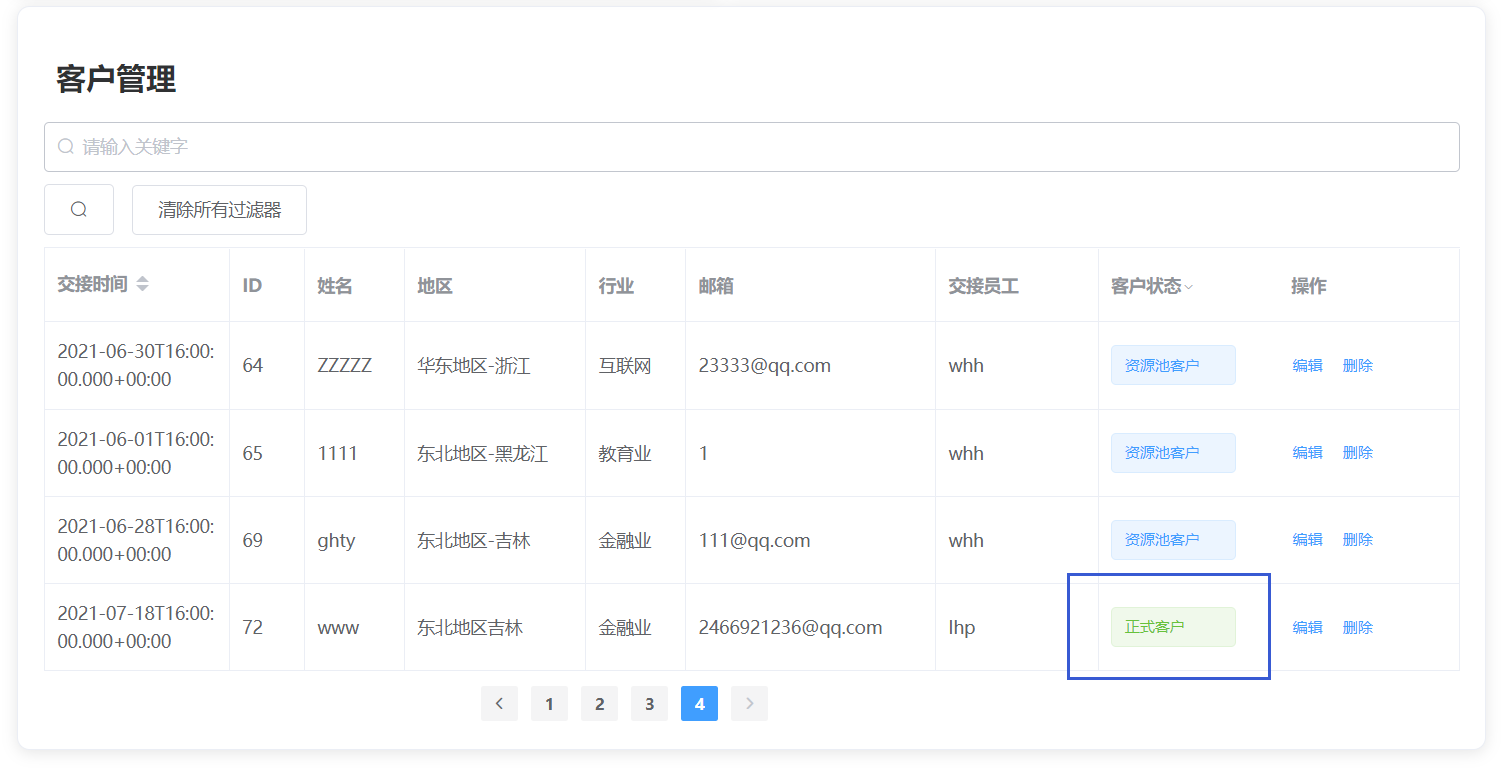
修改客户状态为正式客户



2、测试结果：

修改成功：将信息同步到表格中

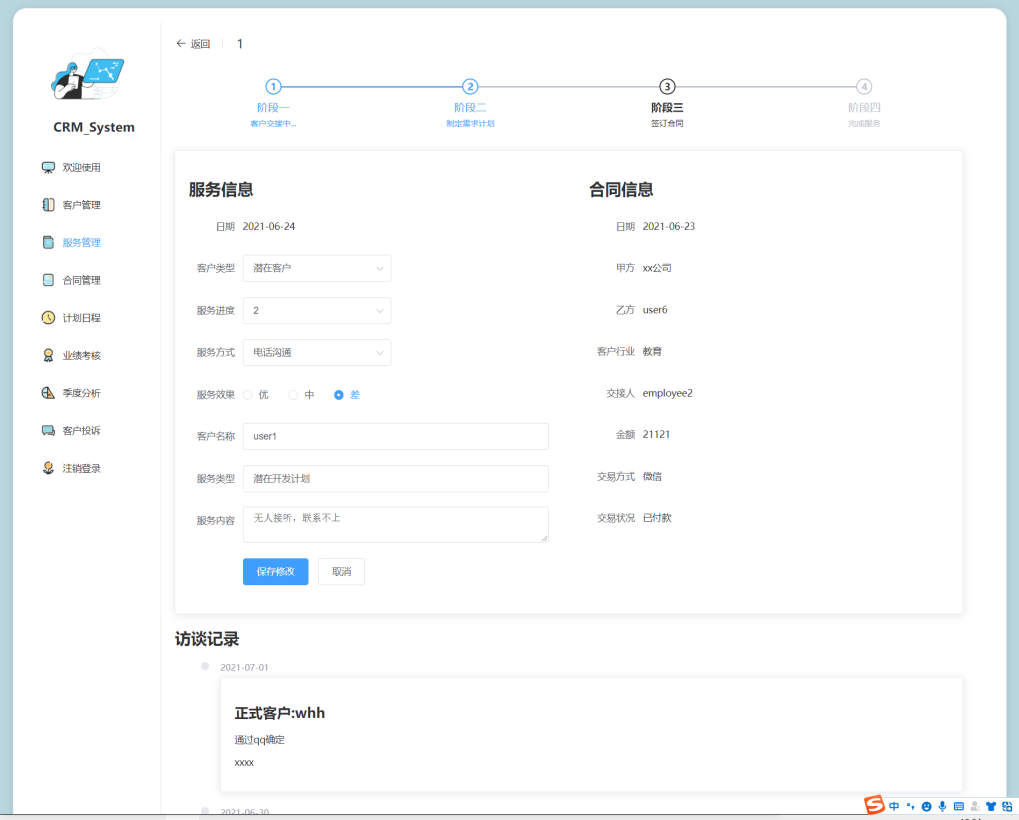
修改失败：弹出提示信息



### 5.3服务管理功能

1. 查看服务及对应的合同

根据所选服务查看对应的服务详情页信息



二、为没有签订合同的服务添加合同

1、测试数据：根据合同表单信息进行填写相关数据并提交

日期：2021-07-01

甲方：test

乙方：testB

客户行业：金融业

交接人：lhp

金额：50000

交易方式：现金

交易状况：已支付



2、测试结果：

添加合同成功，将填写表单的模块改为展示合同信息的模块



### 5.5 需求计划管理功能

一、为服务添加需求计划/日程

1、测试数据：选择服务后填写新建的计划内容

选择服务：1

客户名称：www

沟通方式：电话

需求概要：加急

详细计划：加急

截止时间：2021-07-14



2、测试结果：将对应的计划更新到日历表中



### 5.6 图表功能

1、测试数据：

月份：2021-07-01

2、测试结果：返回当月所在季度对应的业绩信息



### 5.8 考核功能

一、员工资料修改测试

1、测试数据：

员工姓名：employee1 => 修改为 eee

所属部门：财务部

员工邮箱：1452588824@qq.com

员工性别：男



2、测试结果：



二、考核信息添加测试

1、测试数据：

考核时间：2021-07-01

考核员工：employee2

考核分数：

工作能力[40%]1-10分

10

工作态度[20%]1-10分

10

环境行为[10%]1-10分

10

纪律[10%]1-10分

10

加分项[20%]1-10分

10

扣分项[10%]1-10分

10

备注

考核经理：lhp



2、测试结果：

提交考核后会将对应的表单信息提交到数据库中在此点击考核则会显示已考核并列出考核的分数详情，且将提交按钮置为不可选。



## 6.用户手册

本系统分为三个操作端，分别为客户端、员工端、经理端，不同的登录端表示不同的身份，从而实现了三种功能操作的权限。

### 6.1 客户端

### **6.1.1 客户登录**

客户登录该系统首先需要有一个对应的账号，可以通过登录界面的注册按钮进入注册页面，如图6-1、6-2所示为该系统的总体注册登录界面。用户可以通过该界面进行注册对应的登录端以及登录操作。



图6-1 系统登录页面



图6-2 系统注册页面

### 6.1.2 客户操作

如图6-3所示为客户登录后呈现的界面，其中，客户可以实现查看合同以及客户投诉的功能。点击查看合同会呈现出对应名下（即该客户签订）的合同界面，对应的可以查看到具体的合同信息，如合同的签订甲乙方（这里的乙方即为当前登录）、签订时间、交易方式、交易状态等都会在这里进行相应的记录。



图6-3 用户登录主界面



图6-4 合同界面

### 6.1.3 客户投诉

客户对某项服务过程中的员工不满意可以在此界面进行投诉，后台会根据接待过当前登录客户的记录将对应的员工列出供客户选择想投诉的对象，并在此处说明理由以及选择对应的不满意程度，系统会将此处的投诉信息及分数进行记录，供经理查询以及考核。



图6-5 客户投诉界面

### 6.1.4 退出登录

用户可以通过注销登录的菜单实现退出系统的登录，同时会将对应的记录信息进行清除并返回到登录界面。

### 6.2 员工端

### 6.2.1 员工操作界面

员工在系统登录界面登录后，会跳转到如图6-6的界面，其中员工的操作包括：客户管理、服务管理、合同管理、计划日程。

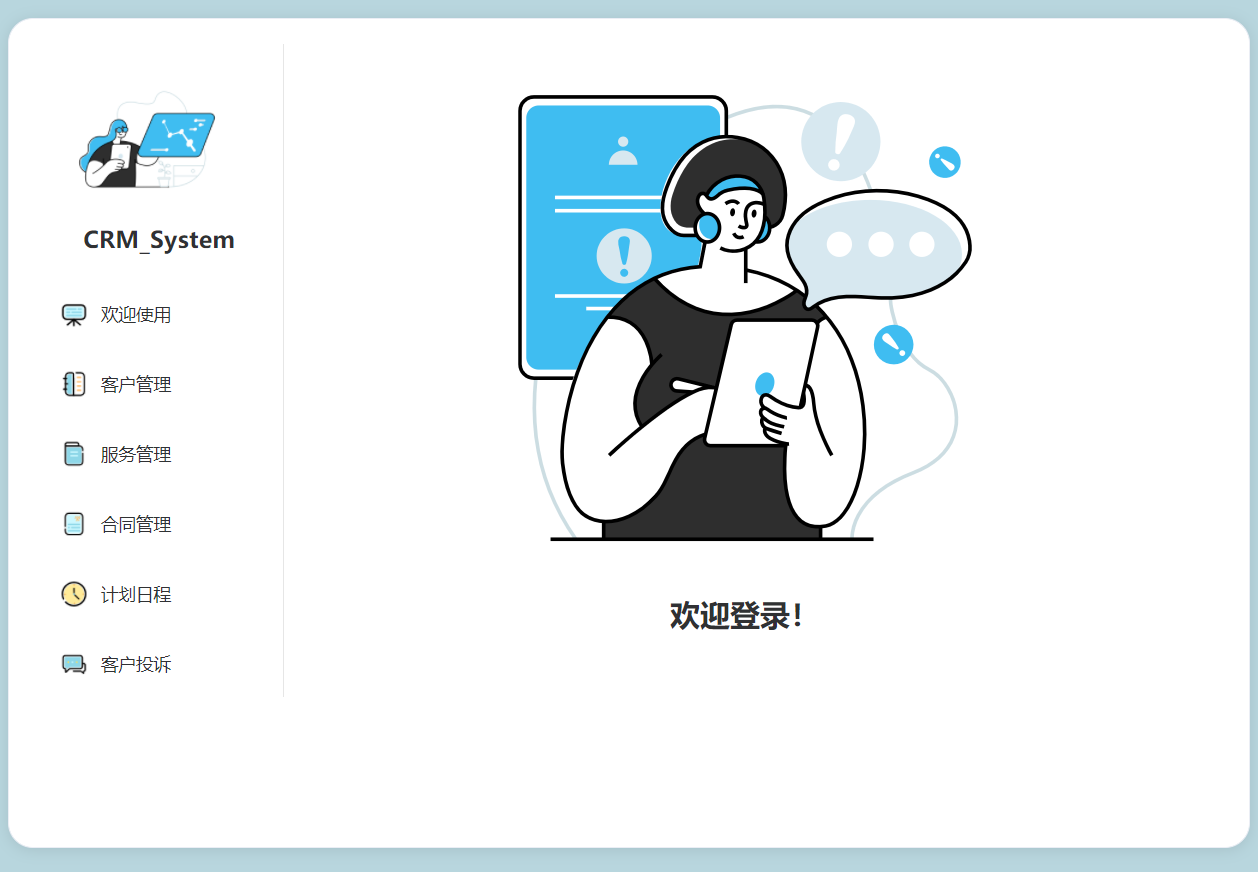


图6-6 员工登录界面

### 6.2.2 客户管理

员工可以点击导航栏的客户管理进入管理界面，如图6-7为员工管理的界面，其中第一个模块实现的是客户的创建，对于新接待的客户可以将其对应的录入系统保存，其中需要填写的信息有客户名称、客户行业（教育业、互联网、金融业、服务业、餐饮业）、客户区域（华东地区、华南地区等）、客户状态（流失、资源池、正式、开发失败、潜在客户）、对应的创建时间即接待时间，可选的字段有星标客户、客户来源、备注。填写对应信息后会系统会根据当前登录的用户即员工进行创建客户（默认其为接待员工，后期可对其进行移交）。

在主页右侧的星标客户模块可以查询到对应的星标客户信息，这里展示的仅为重要程度较高的前10名星标客户。

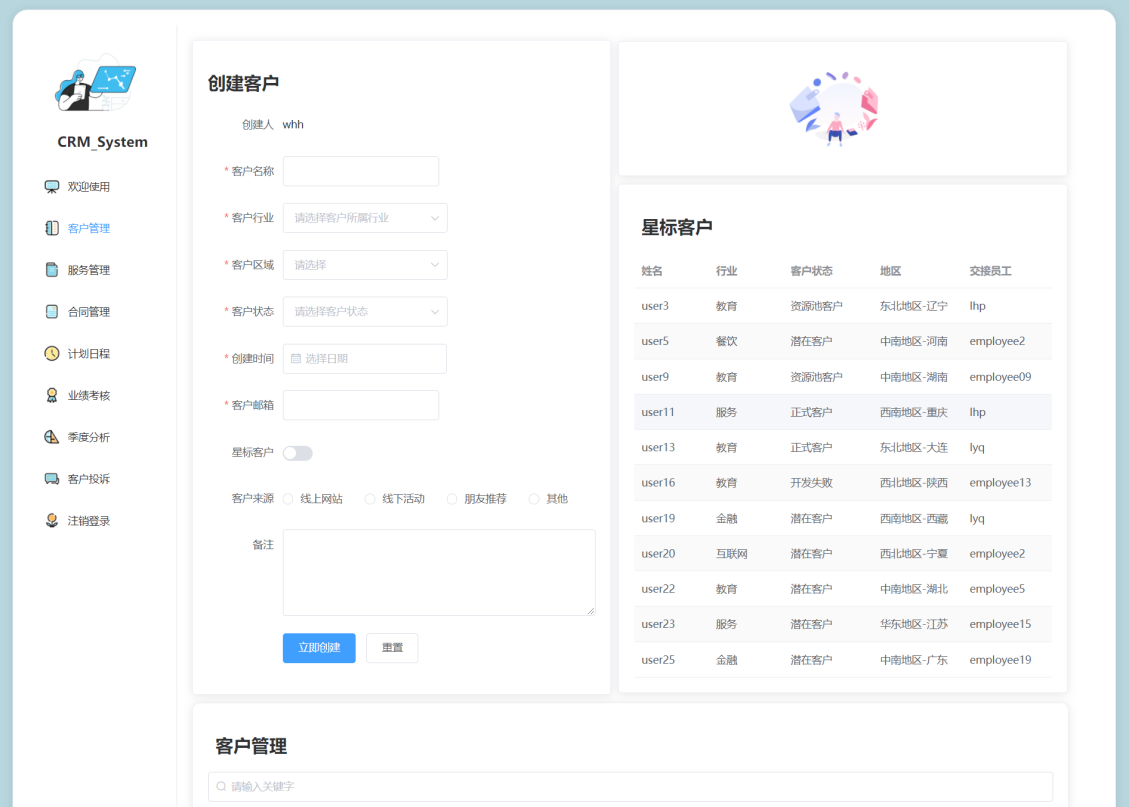


图6-7 客户管理界面

在此页面中的下方会有对应的客户信息表，可以在该表处对客户进行管理，这里会对客户信息进行分页处理（客户数量较多），此时可以查看到客户的各种信息，如交接时间、客户ID、客户姓名、客户行业、客户邮箱、交接员工、客户状态等。通过搜索框可以对客户的姓名进行模糊搜索，便于员工的管理。接着在客户状态栏可以实现对客户状态的筛选，可以筛选出符合状态的多种客户，在操作栏可以实现客户的编辑以及删除。

点击编辑按钮会弹出客户的信息编辑框，如图6-9所示，对表单进行编辑后即可对对应的信息进行修改。点击删除按钮，弹出提示框，点击确定实现对客户的删除。

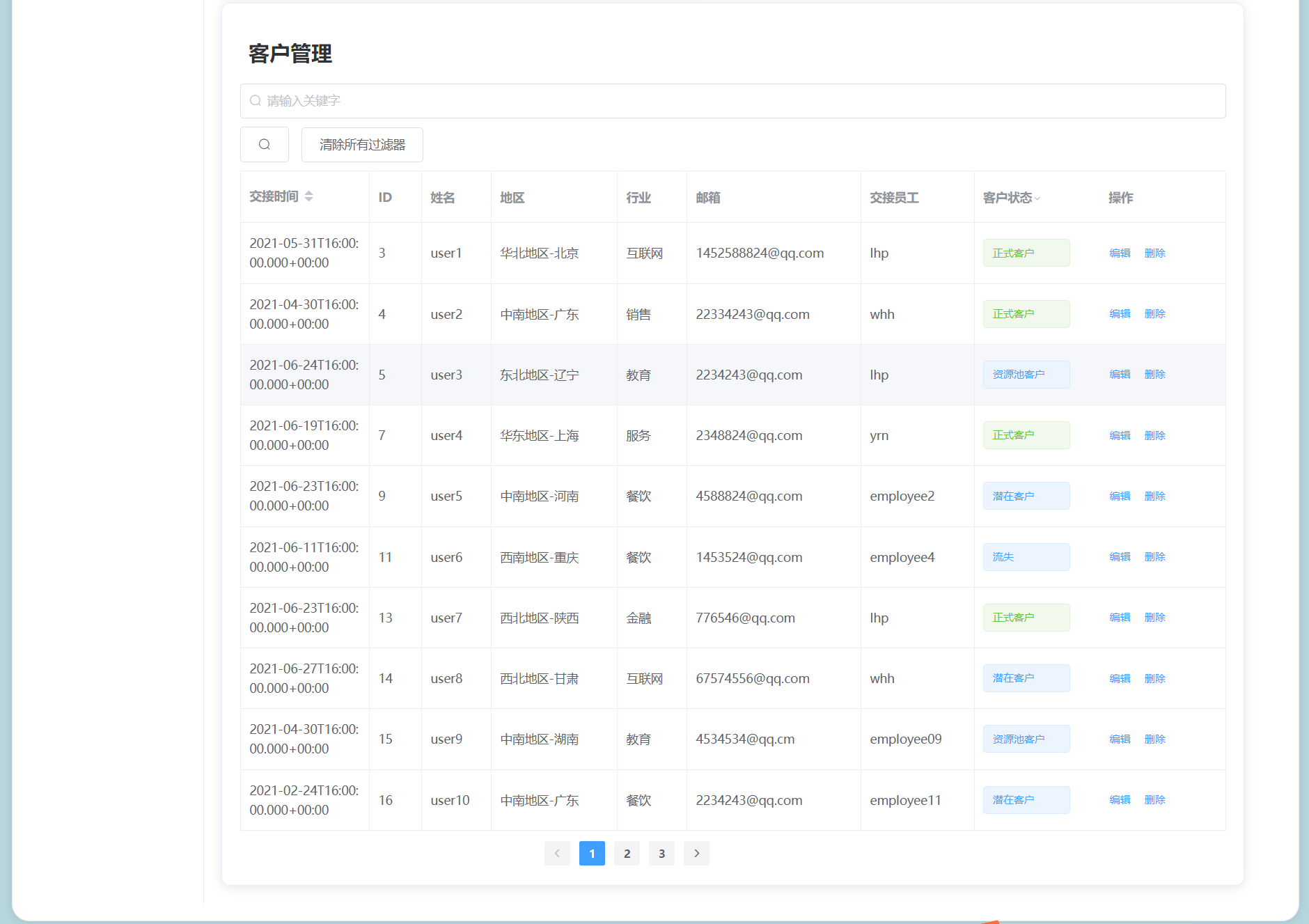


图6-8 客户管理界面

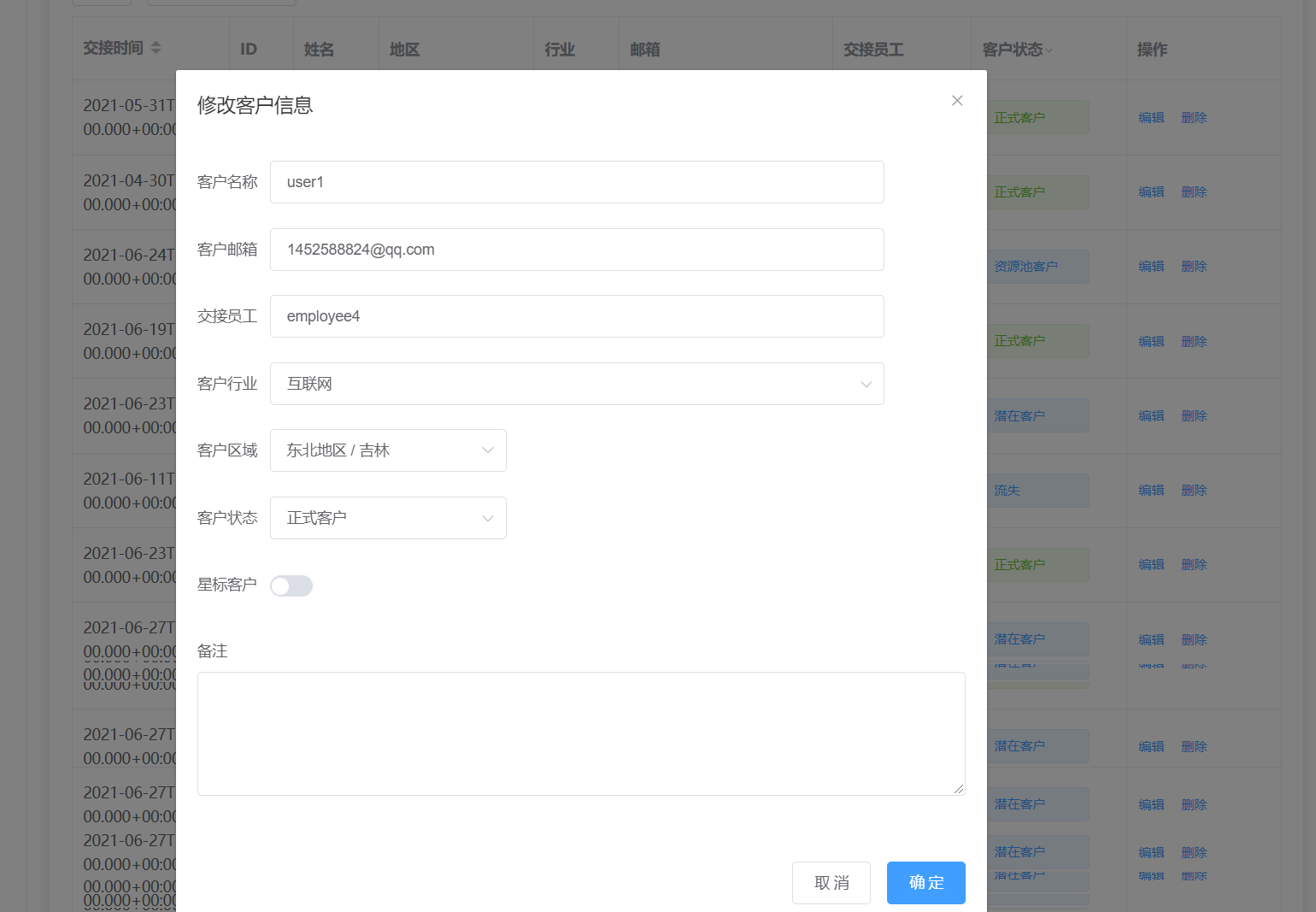


图6-9 客户管理界面

### 6.2.3 服务管理

员工选择服务管理的导航，可以跳转到服务管理界面，对应包含新增服务记录、统计客户状态、服务管理的模块。其中客户状态反应的是当前客户数据中的不同客户状态对应的数量，利用图表以更加直观的形式将其呈现到页面上。而新增服务记录对应服务管理模块，对应的填写新增的表单信息后，点击提交按钮会把对应的字段填充到服务管理中。点击删除按钮可以对应的删除选中的服务记录，实现服务的更新删除。



图6-10 服务管理界面

点击查看按钮，则会跳转到详情界面，如图6-11所示。其中会将选中的服务信息对应的赋值到详细页面供修改，这里的顶部会有一个对应的服务进度条，员工可以对服务信息进行修改以实现进度即服务状态的更新。右侧的合同信息为该服务对应签订的合同，若没有签订合同，那么界面会对应的展示合同的新增窗口，如图6-12 所示为合同的提交表单，由于合同的特殊性，只可对合同进行未有合同的服务增加合同，已有合同服务的查看合同的功能。如图6-11所示，对应的访谈历史中会对应的展示按时间排序的访谈记录，即客户通过某种沟通方式向员工提出的某种需求或服务类型，这里只展示了最新的查看数据。

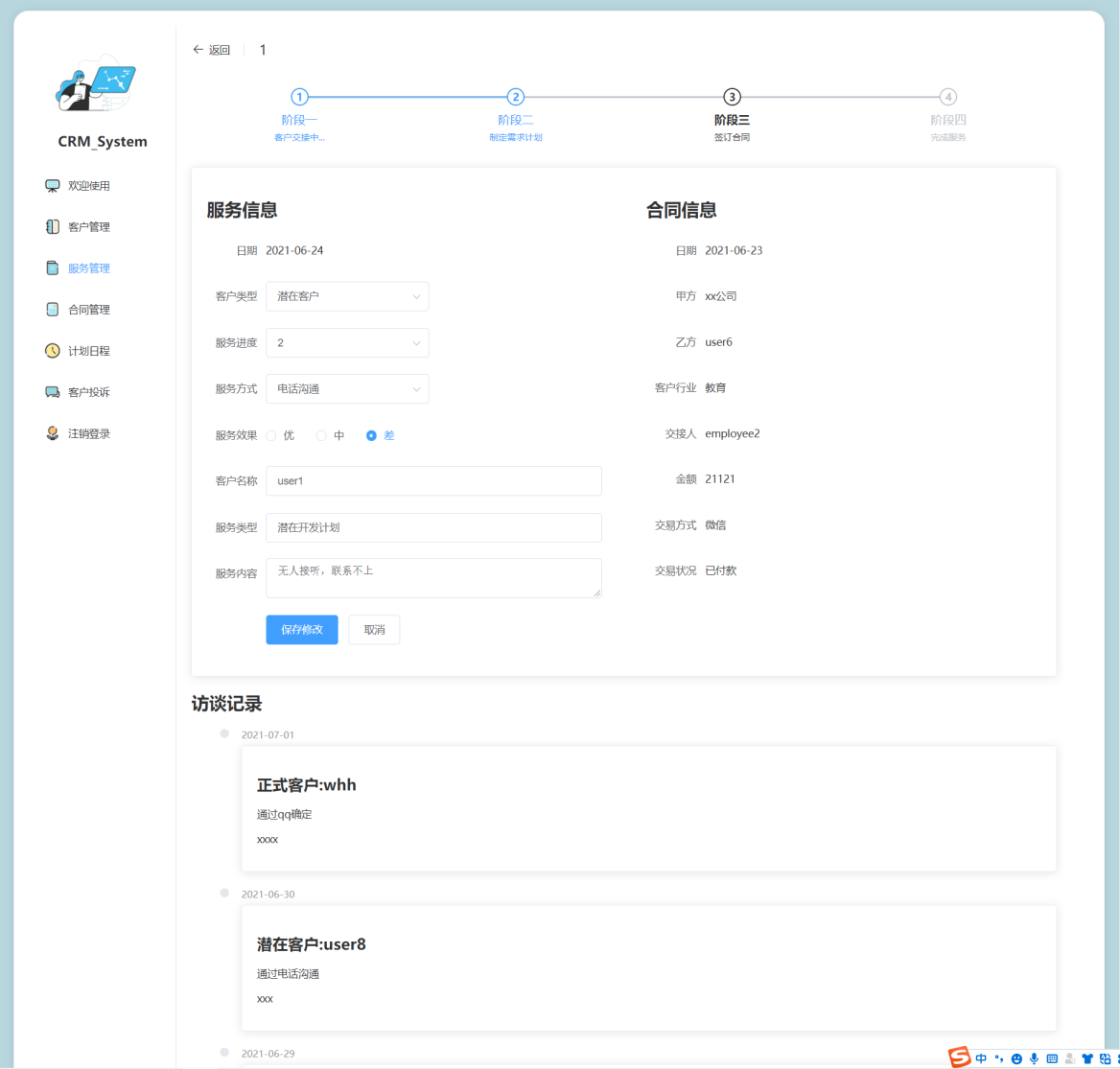


图6-11 服务详情界面

### 6.2.4 合同管理

员工可以在该界面查看到与之相关的合同信息，如图6-12所示，点击拓展按钮可以查询到对应的合同详情，由于合同的特殊性，只可对合同进行未有合同的服务增加合同，已有合同服务的查看合同的功能。



图6-12 合同管理界面

### 6.2.5 计划日程

在此模块中员工可以按照步骤对客户或者服务所需对应的增加需求计划或日程规划，如图6-13所示为步骤一，选择对应的一项服务，然后可以点击下一步实现对计划的新增，如图6-14所示，填写对应的表单信息后，即可将对应的需求计划填充到日程中，其中日程模块如图6-15所示。



图6-13 新建计划步骤一



图6-14 新建计划步骤二

日程模块中以日历的形式呈现了对应的需求计划等信息，可以较为直观的供员工来安排对应的时间以及即时的反馈客户。

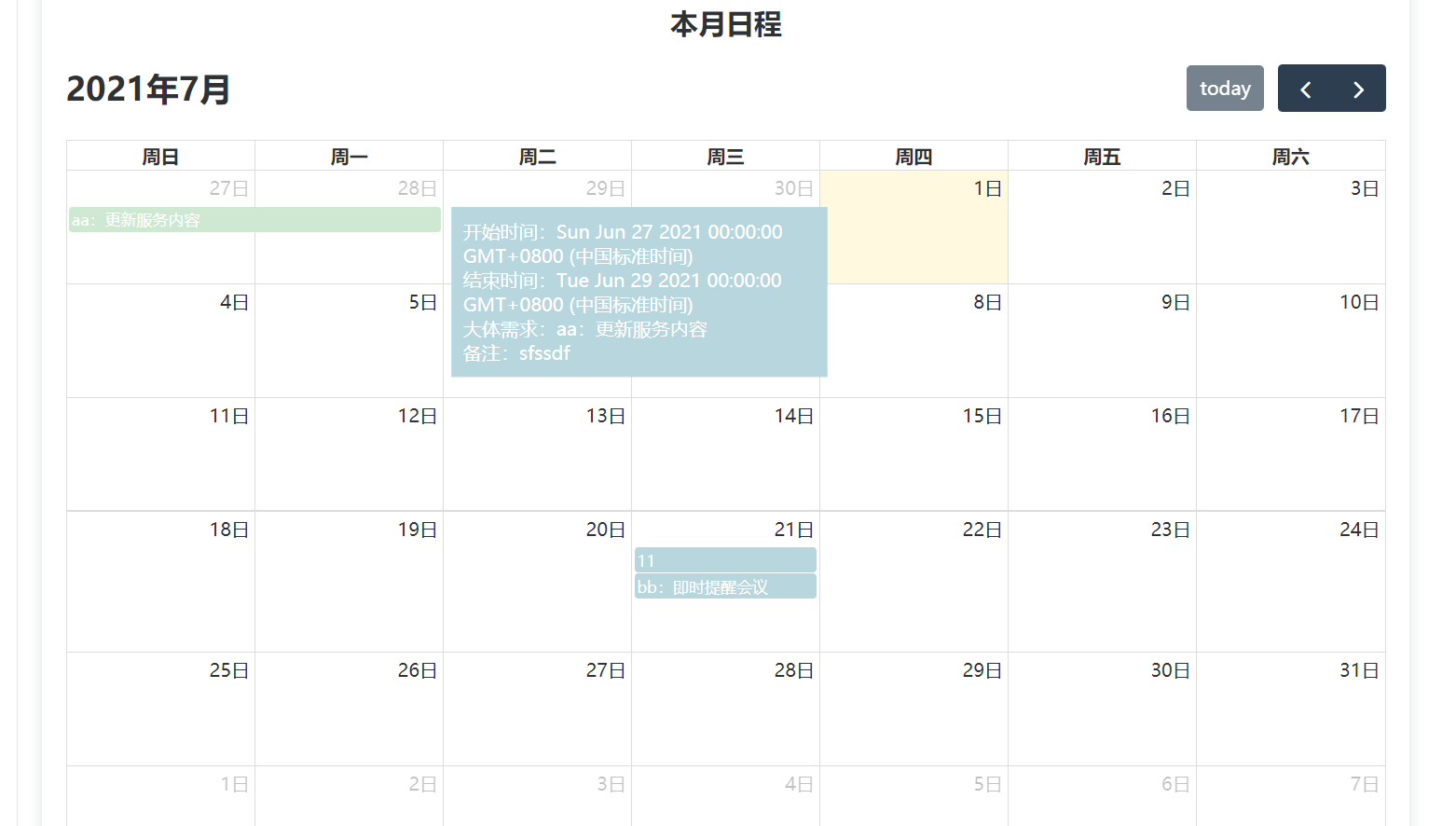


图6-15 日程计划表

### 6.3 经理端

### 6.3.1 业绩考核

经理登录该系统后，除了以上的功能，还可以对员工进行考核。点击业绩考核，进入考核界面，如图6-16所示，其中还能在此查询、修改、移除员工的等操作，与上述步骤大致相同，点击编辑后即可以表单形式提交要修改的员工信息，实现员工信息的修改。

对于考核表单，经理可以按照考核内容对员工的业绩进行考核评定，其中考核经理固定为登录该系统的人员。在考核表单中对应的对其工作能力、工作态度等进行分值的评定，后台会根据其对应的占比情况对员工当月的业绩作出考核评定，从而实现基本的员工绩效考核。

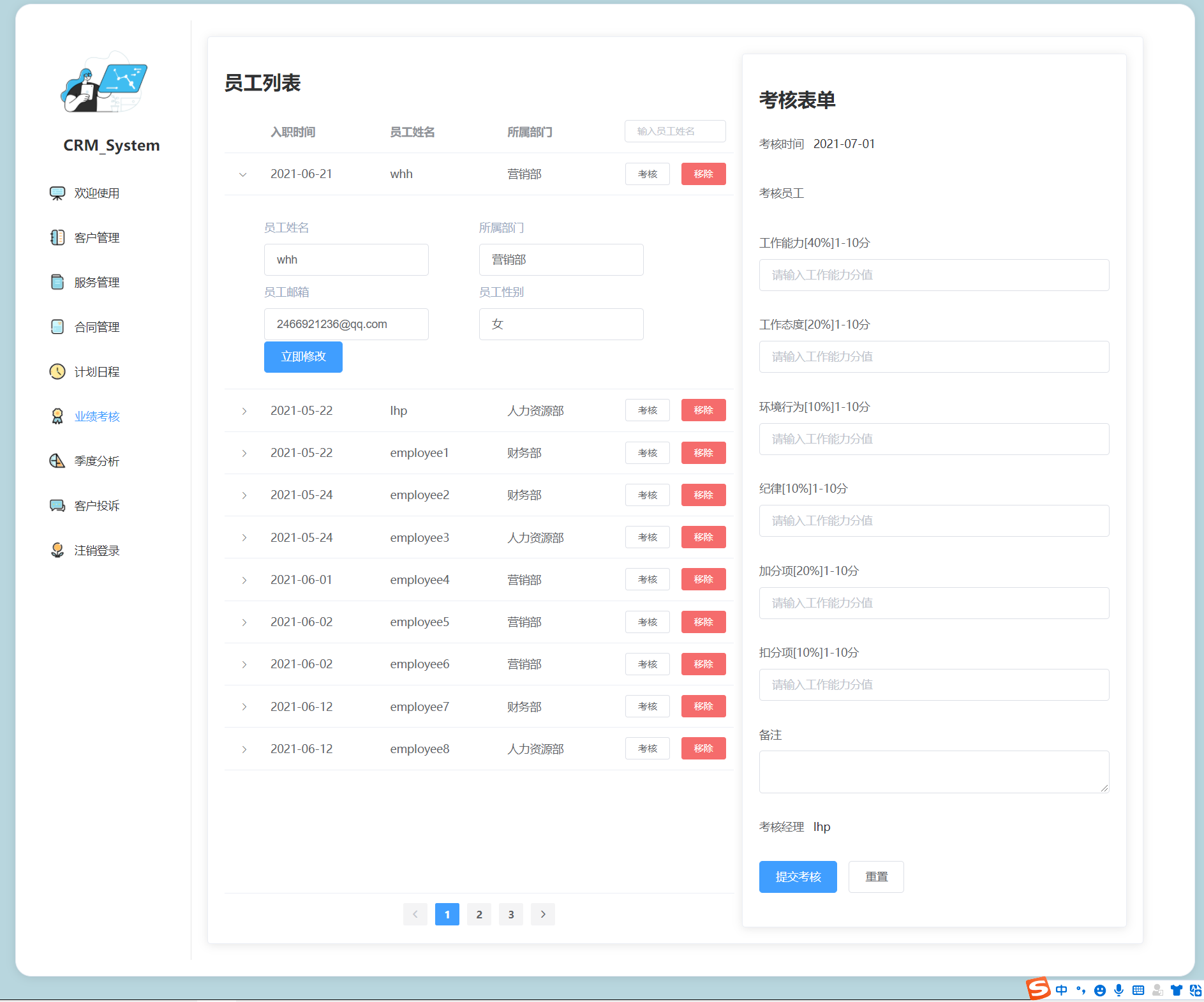


图6-16 业绩考核界面

### 6.3.2 季度分析

经理可以点击季度分析才当进入季度的业绩分析界面，如图6-17所示。在此界面中包含四个模块，分别为业绩排行、销量分析、客户差异化分析、趋势分析。

其中第一个模块销量分析总结了本公司最近6个月的销售情况，这里是根据合同表中已签订的合同作为销量的评判依据。

第二个模块为业绩查询，这里是根据每个员工所负责的客户中签下的合同作为销量依据，经理对员工的考核作为评判标准，附加条件有客户投诉的扣分值按照一定的比例来汇总成最后的业绩分数指标以供展示评分最高的6条员工数据。

第三个模块从客户的角度出发来作为图表的数据，根据客户的行业不同来对应的查看占比情况，从该图中可以查询到客户中不同来源的客户所占的比例，比较直观的展示了客户不同来源的行业间差异。

第四个模块是对每个客户行业、对应不同的月份所生成趋势图表，在这个模块中，可以查询到根据行业的不同，每个月对应的销量数据也存在着一定差异，而对经济上的直观变化也可以更好的通知管理层及时调整企业规划。



图6-17 季度分析界面

# 三、设计总结或结论

系统的设计和开发，是一项非常复杂且任务繁重的工作。在事先的研发和设计过程中，对其进行深入具体的系统分析显得尤为重要，这决定着系统研究和设计的价值和意义。CRM系统可以协助企业实现对既有消费者的管理，避免消费者大规模流失，也为企业经营管理工作提供参考，使得各个部门之间形成良好的沟通与交流，优化企业组织结构以及工作流程。在企业客户关系管理方面具有强大的管理功能和优势，为企业提升客户管理效率、优化客户关系、转变营销模式以及降低营销成本提供了便捷的系统平台支撑。

该系统采用B/S的结构来设计，相较与采用C/S的结构设计来说，安全性能提高了很多。保证了系统的易维护性和易升级性。采用三层体系结构，整个系统结构层次分明，结构清晰，使得整个系统的维护工作更加简单方便。

对于本次设计，其主要优点是对于多种增删查改的操作使用了父、子组件进行页面的设计，减少了嵌套页面的使用，实现了页面数据的高效展示以及减少操作数量，便利了用户的操作，并以比较直观的形式将数据展示在图表中，使用了Echart组件对业绩模块的数据进行分析处理，提高了经理等管理人员对数据的直观展示，此处的前后端交互还利用Vuex来保存token实现校验信息，获取用户的登录状态从而实现页面的跳转。其中还有很多不足之处，比如对应身份验证这一块设想使用AOP来验证其角色权限，实现真正的权限控制；在功能上可以拓展报表功能来打印对应的业绩信息以实现数据生成报表；对于页面的渲染刷新机制这里使用了Update的生命周期来判断，可能给存在更优于页面加载的方法来实现数据的刷新，如使用Vue3等。对于该系统其他敌方的改进，或许可以添加一些未实现的功能，比如在客户端那里，可以提供不同部门的一些服务人员的相关信息，可以让客户跟员工在系统中进行沟通，咨询等，对于员工业绩考核部分，也可以设计业绩报表的生成。

# 参考文献

1. 张兴华.CRM客户信息管理系统的设计与实现[J].山西电子技术,2021(03):59-61.
2. 周洁莹.企业客户信息管理系统设计及实现探讨[J].通讯世界,2016(02):98-99.
3. 杨勤.基于B/S架构的企业客户信息管理系统设计与实现[D].厦门大学,2017.
4. 李阳.基于B/S架构的客户信息管理系统设计与实现[D].电子科技大学,2013.
5. 苑旭东.客户信息管理系统的设计与实现[D].吉林大学,2012.
6. 张旭刚,张昕,高若寒.基于Spring Boot与MyBatis框架构建动态读写分离模型[J].微型电脑应用,2021,37(02):84-86+98.
7. 王志文.Vue+Elementui+Echarts在项目管理平台中的应用[J].山西科技,2020,35(06):45-47.
8. 许华琳.基于Spring Boot的代码自动生成系统设计的分析[J].电子技术与软件工程,2021(10):30-31.
9. 张兴华.CRM客户信息管理系统的设计与实现[J].山西电子技术,2021(03):59-61.
10. 张峰.应用SpringBoot改变web应用开发模式[D].科技创新与应用,2017-08-18．
11. 冷霞.实施客户关系管理系统不断提升企业竞争力[J].中国商论,2020(23):129-130

# 附录