服务商订单管理系统

|  |  |
| --- | --- |
| 专业 | Web应用开发 |
| 姓名 | 李幸阜 |
| 学号 | 1804240326 |
|  | |

武汉纺织大学

计算机与人工智能学院

**2021年6月**

摘要

信息技术日新月异,电商服务平台需求也日益增长,如何以较短的周期开发出高效优质的服务商订单管理系统将具有很大的研究价值。设计的服务商订单管理系统基于开发效率较高的Spring MVC + Spring + Mybatis(SSM)框架,采用SOA系统架构和B/S3层结构模式,使用MySQL数据库进行开发,通过Maven项目管理工具以及Dubbox分布式治理方案设计实现。该系统主要分为3个子系统:订单管理子系统、统计子系统、服务商管理子系统。系统能使运营商更好的管理和维护自己的平台,为服务和被服务的商家以及用户带来了极大的使用便利。通过对系统的详细设计、编码实现与多次测试,结果证明;该系统性能稳定,能有效的缩短30%的开发周期,减小20%的代码量,并且便于维护。

关键词：服务商订单管理系统，SSM框架，MySQL，B/S模式

ABSTRACT

With the rapid development of information technology and the increasing demand of e-commerce service platform, how to develop an efficient and high-quality order management system for service providers in a short period of time will be of great research value.Based on the Spring MVC + Spring + Mybatis (SSM) framework with high development efficiency, the service provider order management system is designed and developed using the SOA system architecture and B/S3 layer structure mode, using MySQL database, and implemented through Maven project management tool and Dubbox distributed governance scheme.The system is mainly divided into three subsystems: order management subsystem, statistics subsystem and service provider management subsystem.The system enables operators to better manage and maintain their own platform, bringing great convenience to service and service providers and users.Through detailed design, coding implementation and multiple tests of the system, the results prove that;The system has stable performance, can effectively shorten the development cycle by 30%, reduce the code load by 20%, and is easy to maintain.

Keywords: Service provider order management system, SSM framework, MySQL, B/S mode

目录

[1. 需求分析 4](#_Toc77105142)

[1.1.功能需求分析 4](#_Toc77105143)

[1.1.1.功能描述 4](#_Toc77105144)

[1.1.2.服务商管理模块 4](#_Toc77105145)

[1.1.3.订单管理模块 5](#_Toc77105146)

[1.1.4.统计模块 5](#_Toc77105147)

[1.2.非功能需求 6](#_Toc77105148)

[1.3.系统环境 6](#_Toc77105149)

[2. 系统设计 7](#_Toc77105151)

[2.1.软件架构设计 7](#_Toc77105152)

[2.1.1、视图层 7](#_Toc77105153)

[2.1.2、模型层 8](#_Toc77105154)

[2.1.3、控制层 9](#_Toc77105155)

[2.2.功能模块设计 9](#_Toc77105156)

[2.2.4、订单管理子系统 9](#_Toc77105157)

[2.2.5、统计子系统 11](#_Toc77105158)

[2.2.6、服务商管理子系统 12](#_Toc77105159)

[3. 数据库分析与设计 14](#_Toc77105160)

[3.1.数据需求分析 14](#_Toc77105161)

[3.2.数据库设计 15](#_Toc77105162)

[4. 收获 16](#_Toc77105163)

[4.1.Redis 16](#_Toc77105164)

[4.2.SQL通用语句分类： 16](#_Toc77105165)

[4.3.Maven 16](#_Toc77105166)

[4.4.SSM 17](#_Toc77105167)

[参考文献 18](#_Toc77105168)

# 需求分析

## 1.1.功能需求分析

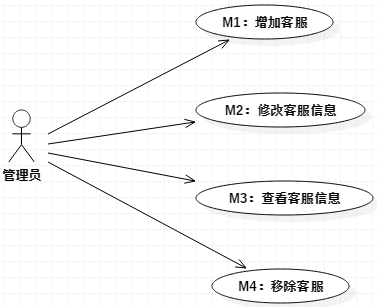
### 1.1.1.功能描述

设计一个客服订单管理系统，该系统包含三个模块，分别是订单管理模块、服务商管理模块、统计模块，服务商管理模块实现客服的增删改查。订单管理模块实现订单的增删改查。统计模块实现订单统计功能。

### 1.1.2.服务商管理模块

在服务商管理模块中，管理员可以增加和移除客服、修改客服信息、查看客服信息。

该模块的用例图如图1-1所示,该模块的用例概述如表1-1所示



【图1-1】服务商管理模块用例图

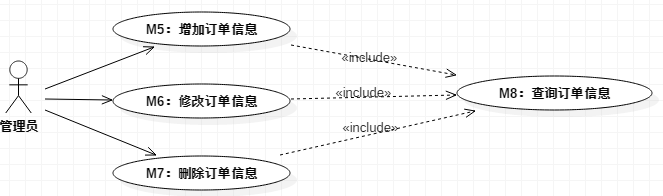
用例简述：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例ID | 用例名称 | 角色 | 优先级 |
| M1 | 增加客服 | 管理员 | 1 |
| M2 | 修改客服信息 | 管理员 | 1 |
| M3 | 查看客服信息 | 管理员 | 1 |
| M4 | 移除客服 | 管理员 | 1 |

【表1-1】服务商管理模块用例简述

### 1.1.3.订单管理模块

在服务商管理模块中，管理员可以增加订单信息、删除订单信息、修改订单信息、查询订单信息，其中增删改订单都需要查询订单信息。实验室预约管理模块用例图如图1-2所示,实验室预约管理模块用例概述如表1-2所示



**【图1-2】订单管理模块用例图**

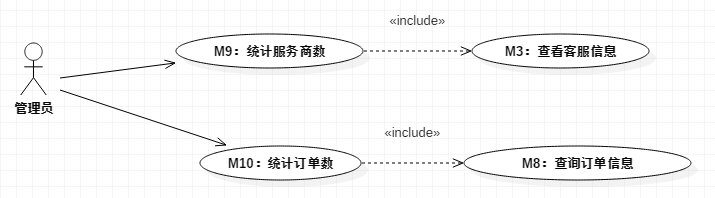
用例简述：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例ID | 用例名称 | 角色 | 优先级 |
| M5 | 增加订单信息 | 管理员 | 1 |
| M6 | 修改订单信息 | 管理员 | 1 |
| M7 | 删除订单信息 | 管理员 | 1 |
| M8 | 查询订单信息 | 管理员 | 1 |

【表1-2】服务商管理模块用例简述

### 1.1.4.统计模块

统计模块包含统计服务商个数和统计订单个数。管理员可以查看服务商个数和统计订单个数。实验室预约管理模块用例图如图1-3所示,实验室预约管理模块用例概述如表1-3所示。



**【图1-3】实验室预约统计管理模块用例图**

用例简述：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例ID | 用例名称 | 角色 | 优先级 |
| M9 | 统计服务商数 | 管理员 | 1 |
| M10 | 统计订单数 | 管理员 | 1 |

**【表1-3】统计模块用例简述**

## 1.2.非功能需求

系统应当以创造优良的用户体验为宗旨。

第一，要有简便的操作流程，简便的流程会使用户操作简单。

第二，要有美观的UI设计，这样用户才会感觉效果好。

第三，系统要对用户请求快速响应，系统对用户请求处理速度快，用户体验才可能流畅。

## 1.3.系统环境

设备环境：PC或移动端上Chrome浏览器、Edge浏览器、火狐浏览器。

开发使用数据库：Redis 、MySQL

开发语言：后端使用Java(SSM框架)，前端使用html、css、js

服务器和工具：Tomcat9.0、maven 、Navicat for MySQL、Intellij IDEA、Chrome开发工具



# 系统设计

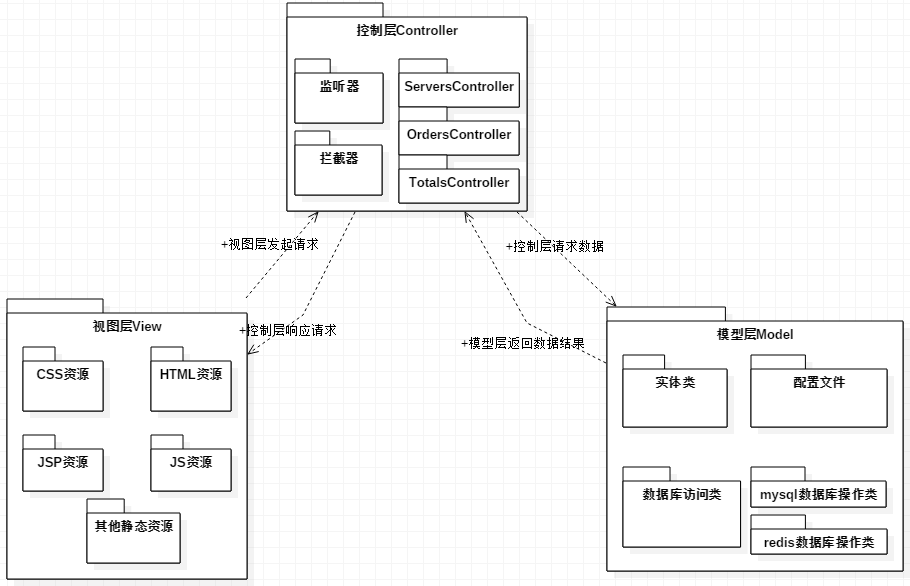
## 2.1.软件架构设计

该服务商订单管理系统采用MVC系统架构，MVC 模式代表 Model-View-Controller模式。这种模式用于应用程序的分层开发，使其前后端分离，这样便于维护、管理以及开发。

Model（模型）层用来查询或修改由控制层发起的请求访问的数据，并且返回结果。

View（视图）层会提交数据给控制层，并把控制层反馈的数据结果可视化。

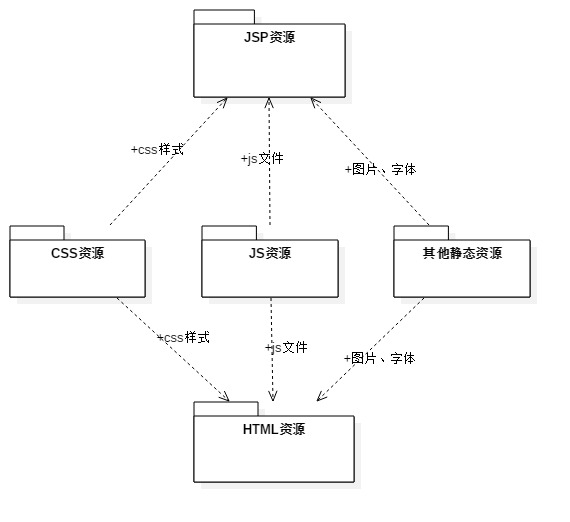
Controller（控制）层用来调度View和Model层，Controller层通常是接收并处理View层的请求并对Model层发出数据访问请求。



**【图2-1】服务商订单管理系统设计包图**

### 2.1.1.视图层

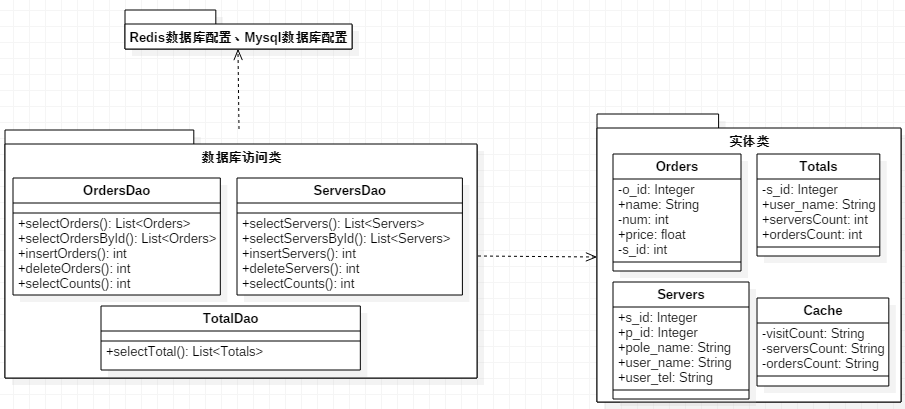
在JS包和CSS包中分别存放了JavaScript文件和CSS文件，在静态资源包中存放了图片和字体用来美化界面。在JSP包中存放了需要进行动态数据可视化的页面，而静态页面全部放在HTML包中。视图层设计包图如图2-2所示。



**【图2-2】视图层设计包图**

### 2.1.2.模型层

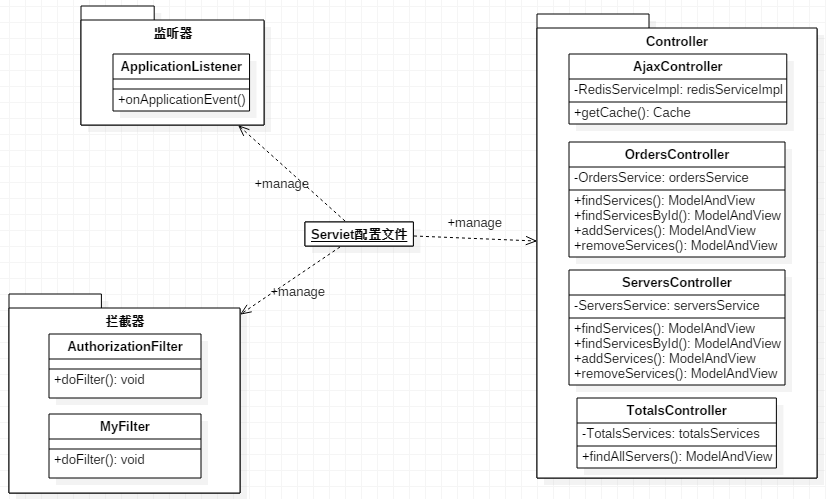
模型层包含数据库操作类、数据库访问类以及实体类，数据库操作类采用JDBC规范，并通过数据库配置文件中的的配置信息进行管理，数据库访问类定义了对不同对象进行相应的数据访问操作，实体类包括Orders（订单）类，Servers（服务商）类，Totals（统计）类，Cache（缓存信息）类， 模型层设计包图如图2-3所示。



**【图2-3】模型层设计包图**

### 2.1.3.控制层

控制层用于接收并处理视图层的请求并对模型层发出数据访问请求，之后再根据模型层返回的数据结果对视图层进行响应，他们负责用户各个功能流程的处理和控制，并对数据进行处理。整个控制层都通过Servlet的配置文件进行管理.控制层包图如图2-4所示。



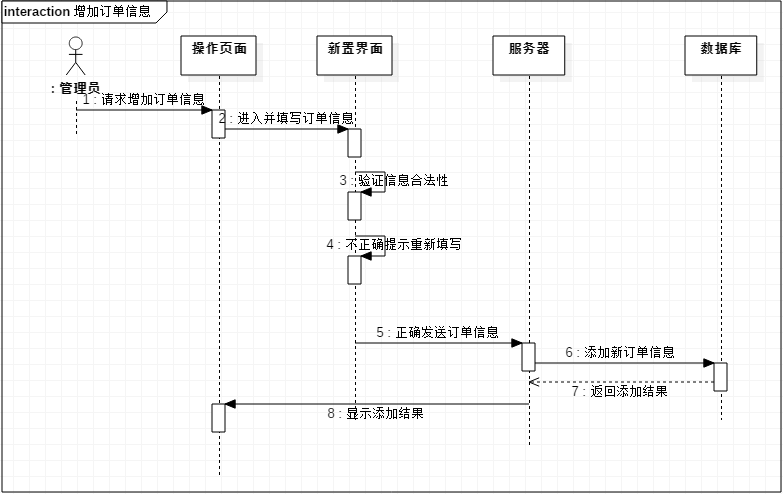
**【图2-4】控制层设计包图**

## 2.2.功能模块设计

### 2.2.4.订单管理子系统

2.2.4.1、增加订单信息

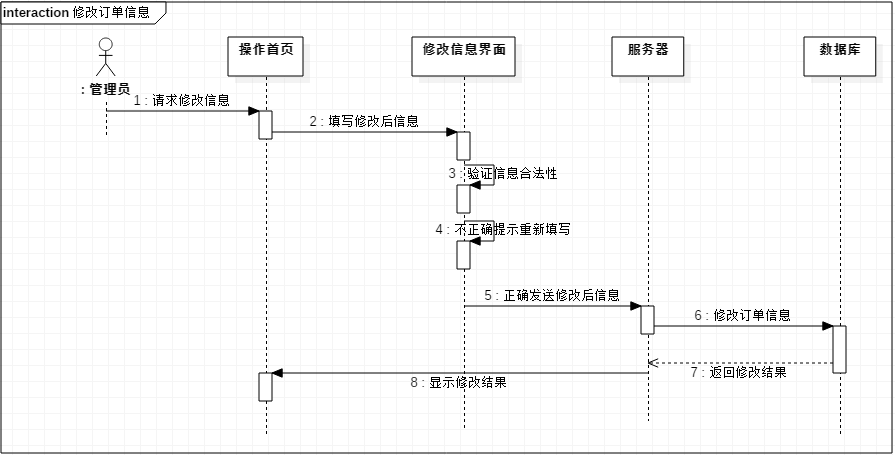
管理员在首页选择增加订单信息，管理员填写新订单信息，系统检查信息格式是否正确，服务器在数据库中添加新订单信息，返回添加结果给服务器，然后系统显示添加后的结果。其时序图如图2-5所示。



**【图2-5】增加订单信息时序图**

2.2.4.2、修改订单信息

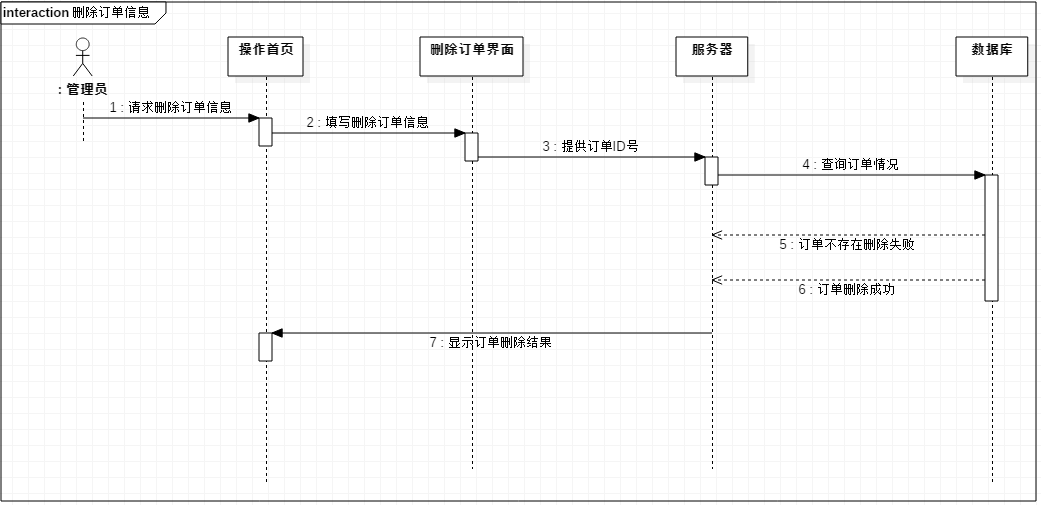
管理员在首页选择修改订单信息，管理员填写信息，系统检查信息格式是否正确，服务器在数据库中修改对应订单信息，返回修改结果给服务器。其时序图如图2-6所示。



**【图2-6】修改订单信息时序图**

2.2.4.3、删除订单信息

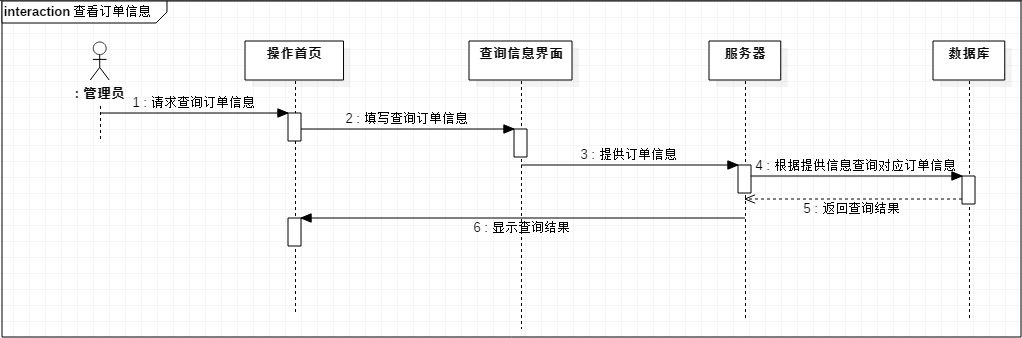
管理员在首页选择删除订单信息，之后填写要的删除订单，系统将要删除的订单信息提供给服务器，服务器在数据库中查询对应订单的记录，订单若不存在则返回删除失败，否则返回订单删除成功给服务器，返回删除订单结果。其时序图如图2-7所示。



**【图2-7】删除订单信息时序图**

2.2.4.4、查看订单信息

管理员选查订单信息，输入要查询订单的信息，系统提交订单信息给服务器，服务器在数据库根据订单信息查询对应订单的信息，返回结果给系统。时序图如图2-8所示。

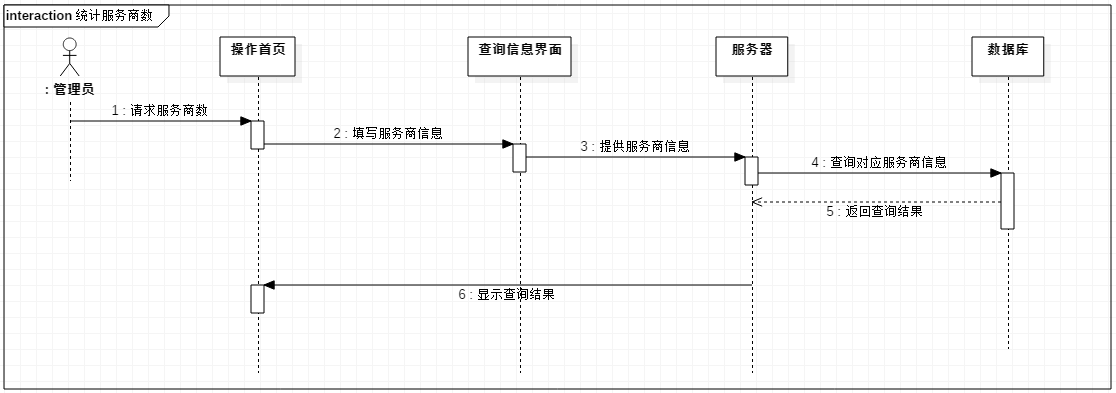


**【图2-8】查看订单信息时序图**

### 2.2.5.统计子系统

2.2.5.1、统计服务商数

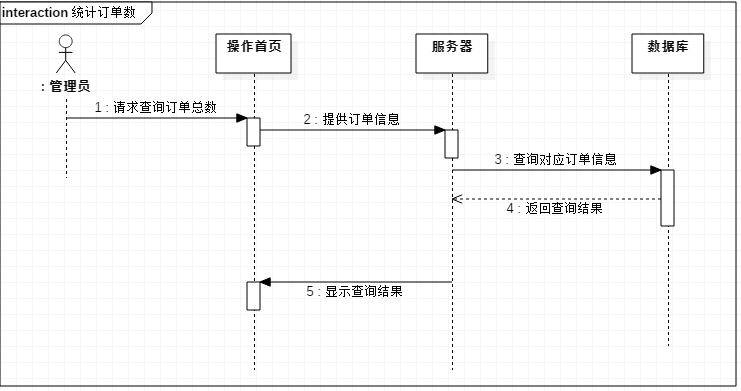
管理员选择统计服务商数，系统会将请求交给服务器，再让数据库查询服务商总数，查询成功后让服务器返回结果给系统，然后系统显示查询结果。其时序图如图2-9所示。



**【图2-9】统计服务商数时序图**

2.2.5.2、统计订单数

管理员选择统计订单数，系统会将请求交给服务器，再让数据库查询订单总数，查询成功后让服务器返回结果给系统，然后系统显示查询结果。其时序图如图2-10所示。

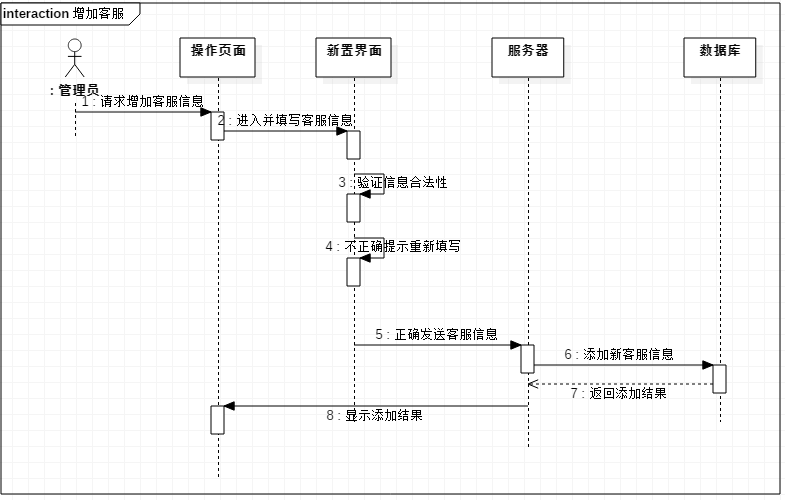


**【图2-10】统计订单数时序图**

### 2.2.6.服务商管理子系统

2.2.6.1、增加客服

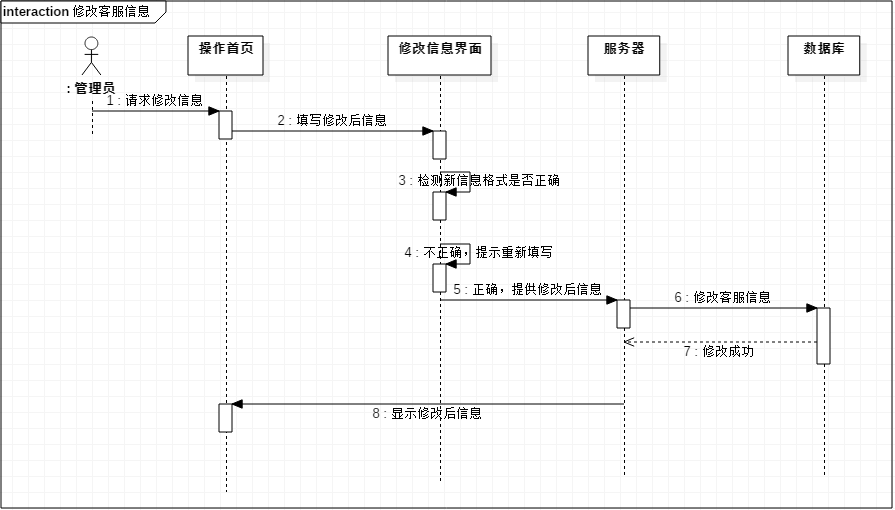
管理员在首页选增加客服信息，管理员填写客服信息，系统检查信息格式是否正确，服务器在数据库中添加新客服信息，返回添加结果给服务器。时序图如图2-11所示。



**【图2-11】增加客服时序图**

2.2.6.2、修改客服信息

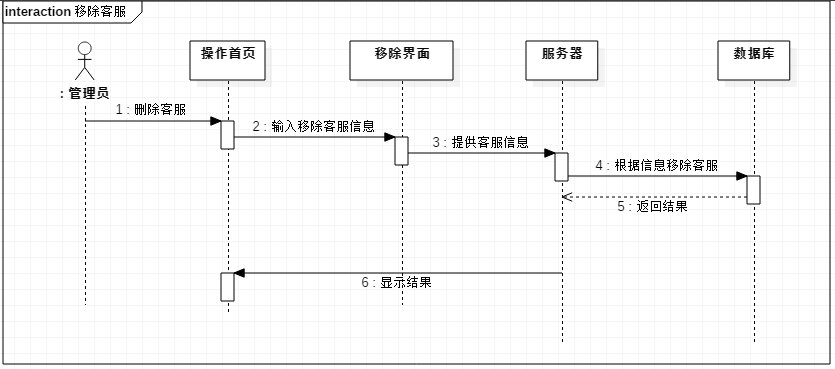
管理员在首页选修改客服信息，管理员填写新的信息，系统检查信息格式是否正确，服务器在数据库修改对应客服信息，返回修改结果给服务器。时序图如图2-12所示。



**【图2-12】修改客服信息时序图**

2.2.6.3、移除客服

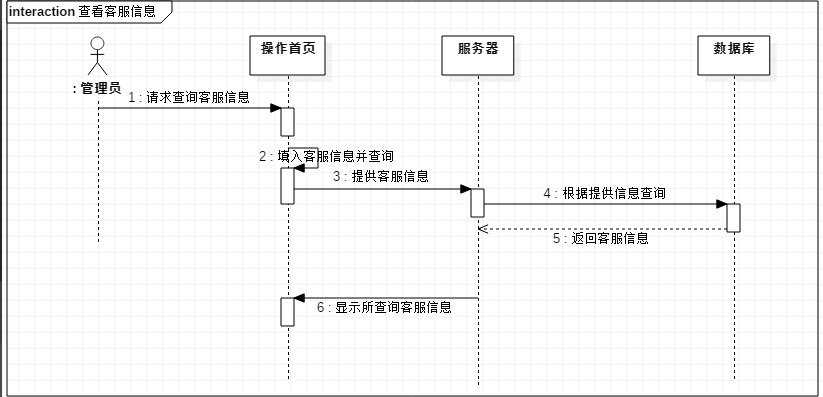
管理员在首页选移除客服，管理员填写要移除客服信息，系统将要信息提供给服务器，服务器在数据库中查询对应客服信息后系统返回结果。其时序图如图2-13所示。



**【图2-13】移除客服时序图**

2.2.6.4、查看客服信息

管理员选查看客服信息后输入要查询客服信息，系统提交客服信息给服务器，服务器在数据库中查询客服信息后返回结果给系统。其时序图如图2-14所示。



**【图2-14】查看客服信息时序图**

# 数据库分析与设计

## 3.1.数据需求分析

①记录服务商基本信息E-R图如图3-2所示，服务商信息类图如3-3图所示。

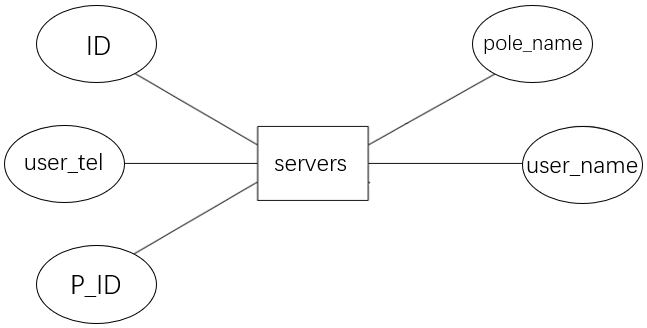
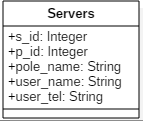
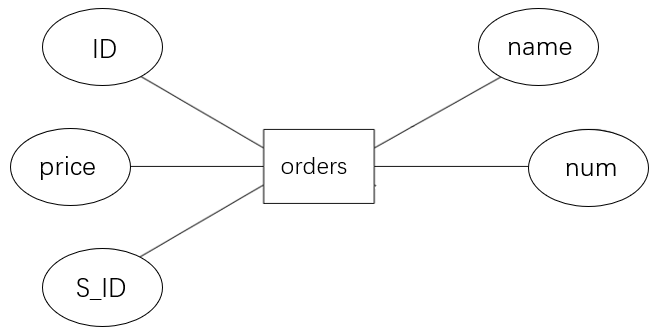
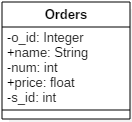
 

图3-2 服务商基本信息E-R图 图3-3 服务商信息类图

②记录订单基本信息E-R图如图3-4所示,订单基本信息类图如图3-5所示。 

**图3-4 订单基本信息E-R图 图3-5 订单基本信息类图**

## 3.2.数据库设计

该系统的数据库是基于MySql创建，包含了服务商表（如表3-8）、订单表（如表3-9）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型（长度） | 非空 | 说明 |
| s\_id | ID号 | int | 是 | 主键 |
| p\_id | 上级id | int | 否 |  |
| pole\_name | 角色 | Varchar(15) | 是 |  |
| user\_name | 名称 | Varchar(11) | 是 |  |
| user\_tel | 电话 | Varchar(255) | 是 |  |

**表3-8 服务商表 servers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型（长度） | 非空 | 说明 |
| o\_ld | ID号 | int | 是 | 主键 |
| name | 名称 | Varchar(6) | 否 |  |
| num | 数量 | int | 是 |  |
| price | 价格 | decimal | 是 |  |
| s\_id | 所属服务商 | int | 是 |  |

**表3-9 订单表 orders**

# 收获

## 4.1.Redis

**Re**mote **Di**ctionary **S**erver是一个用C语言编写的、开源、基于内存运行并支持持久化NoSQL数据库，它可用作数据库、缓存、消息中间件MQ。他有以下**特点：**1、支持数据持久化，可以将内存中的数据保持在磁盘中，重启时可再次加载。2、支持多种数据结构，Redis不仅支持简单的key-value类型的数据，同时还提供list，set，zset，hash等数据结构的存储。3、支持数据备份，即master-slave模式的数据备份。

## 4.2.SQL通用语句分类：

DQL(数据查询语言)：select，查询语句

DML(数据操作语言)：insert、delete、update，对表中的数据进行增删改

DDL(数据定义语言)：create、drop、alter，对表进行增删改

DCL(数据控制语言)：grant授权，revoke撤销权限等

TCL(事务控制语言)：commit提交事务，rollback回滚事务

## 4.3.Maven

maven的生命周期:就是maven构建项目的过程，清理，编译，测试，报告，打包，安装，部署都对应一个个生命周期阶段。

maven的命令:maven独立使用，通过命令，完成maven的生命周期的执行。maven可以使用命令，完成项目的清理，编译，测试等等。

mvn clean:清理( 会删除原来编译和测试的目录，即target目录，但是已经install到仓库里的包不会删除)

mvn compile:编译主程序(会在当前目录下生成- -个target,里边存放编译主程序之后生成的字节码文件)

mvn test-compile:编译测试程序(会在当前目录下生成- - 个target,里边存放编译测试程序之后生成的字节码文件)

mvn test:测试( 会生成一个 目录surefire-reports,保存测试结果)

mvn package:打包主程序(会编译、编译测试、测试、并且按照pom.xml配置把主程序打包生成jar包或者war包)

mvn install:安装主程序(会把本工程打包，并且按照本工程的坐标保存到本地仓库中)

mvn deploy:部署主程序(会把本工程打包，按照本工程的坐标保存到本地库中，并且还会保存到私服仓库中,还会自动把项目部署到web容器中)。

## 4.4.SSM

界面层： 和用户打交道的， 接收用户的请求参数， 显示处理结果的。（jsp ，html ，servlet）

业务逻辑层： 接收了界面层传递的数据，计算逻辑，调用数据库，获取数据

数据访问层： 就是访问数据库， 执行对数据的查询，修改，删除等等的。

三层对应的包：

1.界面层： controller包 （servlet）

2.业务逻辑层： service 包（XXXService类）

3.数据访问层（持久层）： dao包（XXXDao类）

三层中类的交互：

用户使用界面层--> 业务逻辑层--->数据访问层（持久层）-->数据库（mysql）

三层对应的处理框架：

1.界面层---servlet---springmvc（框架）

2.业务逻辑层---service类--spring（框架）

3.数据访问层---dao类--mybatis（框架）

框架特点：

1. 框架一般不是全能的， 不能做所有事情

2. 框架是针对某一个领域有效。 特长在某一个方面，比如mybatis做数据库操作强，但是他不能做其它的。

3. 框架是一个软件。

参考文献

[1] UML面向对象建模与设计(第2版),人民邮电出版社。

[2] runoob.com MVC模式<https://www.runoob.com/mvc-pattern.html>

[3]宋锐星,朱小勇,胡琳琳,王昭.HTTP媒体传输方式简述[J].网络新媒体技术,2020,9(03):61-67.

[4] Layui经典模块化前端 UI 框架<https://www.layui.com>