Web大作业

课程名称： 订单执行过程管理-执行确认管理

学 院： 计算机科学与工程学院

专 业： 物联网16 班 级： 二班

组 别： 刘东临&史邵冲

2018 年 11 月 23 日

# 实验目的

BootDo面向学习型的开源框架，简洁高效，减少过渡封装，展现技术本质!

# 实验要求

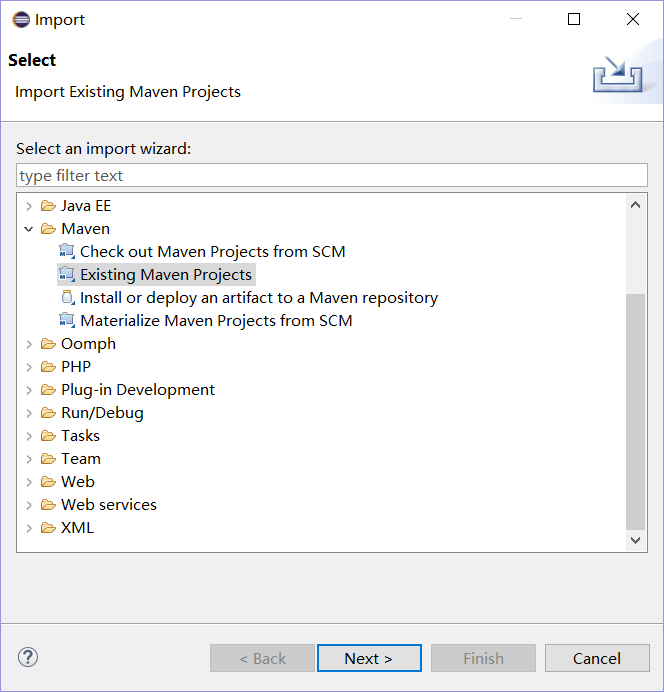
订单执行过程管理

执行确认管理：美容师每做完一次，管理员确认一次，自动减少一次，可修改次数。

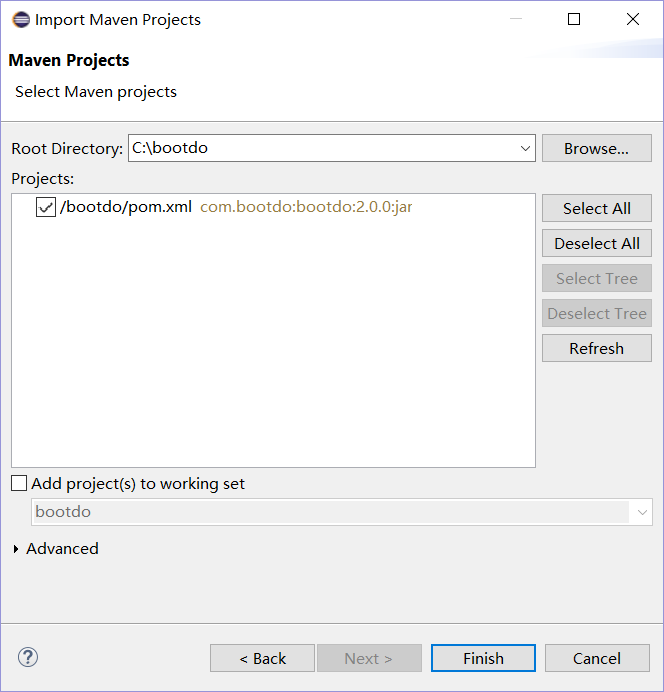
# 实验内容

导入项目

1. 从eclipse选择File->Import->Maven->Existing Maven Projects



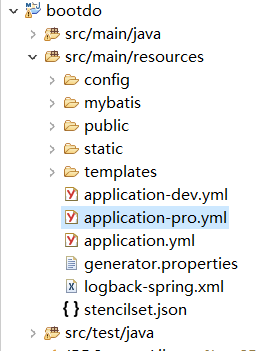
1. Next->Browse->bootdo



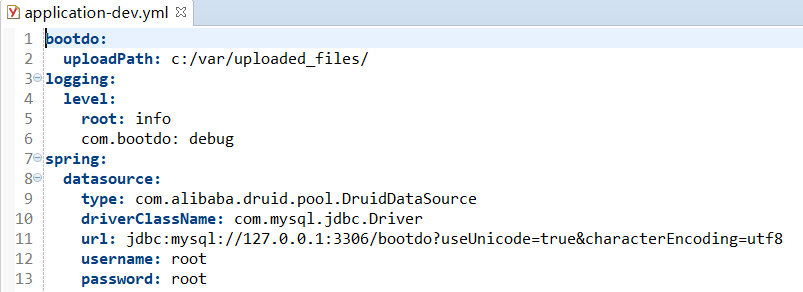
1. 等待项目导入，完成

启动项目

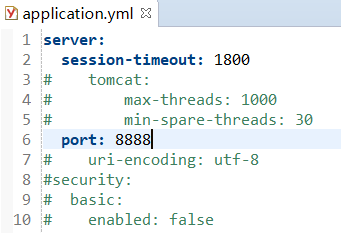
1. 修改配置文件



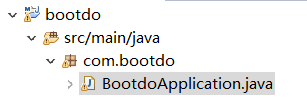
Application-dev.yml修改数据库访问的链接和文件上传的路径



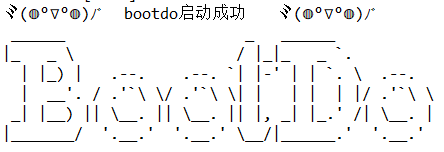
Application.yml修改端口号



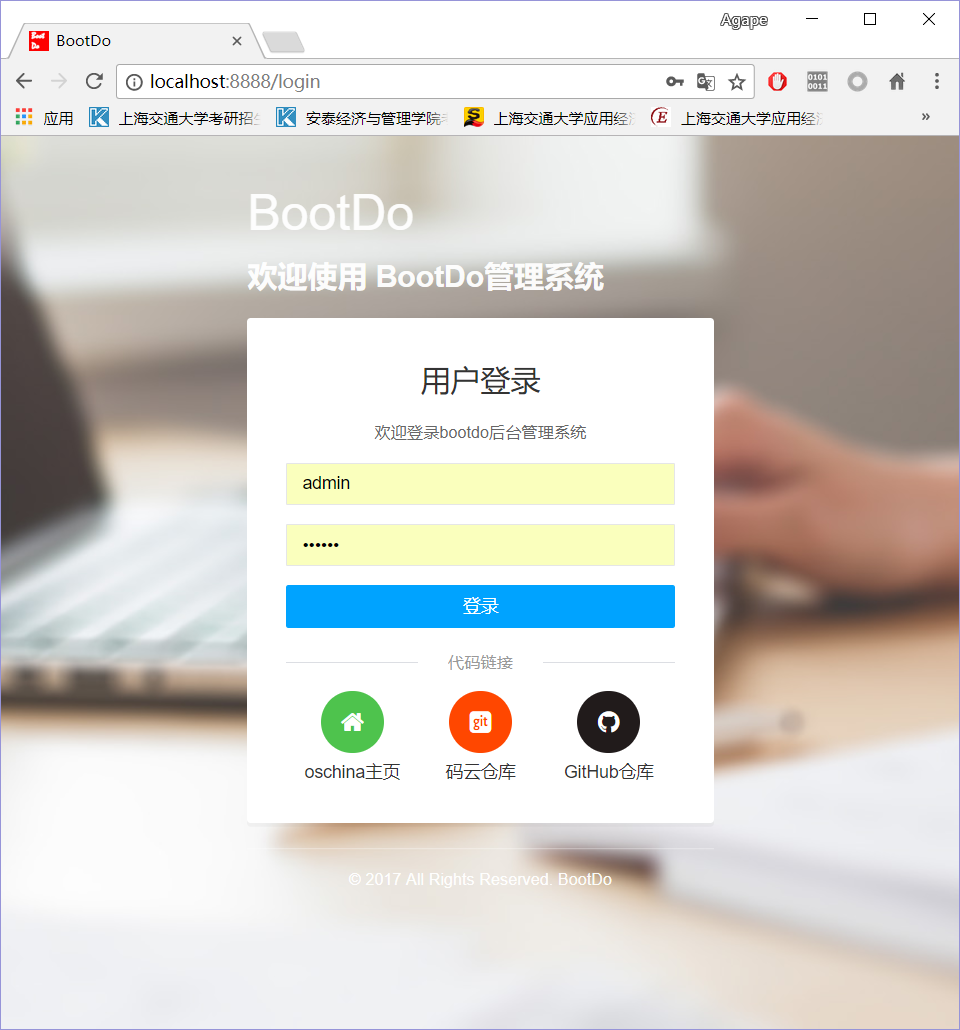
1. 找到BootdoApplication运行Run As Java Application



1. 启动成功



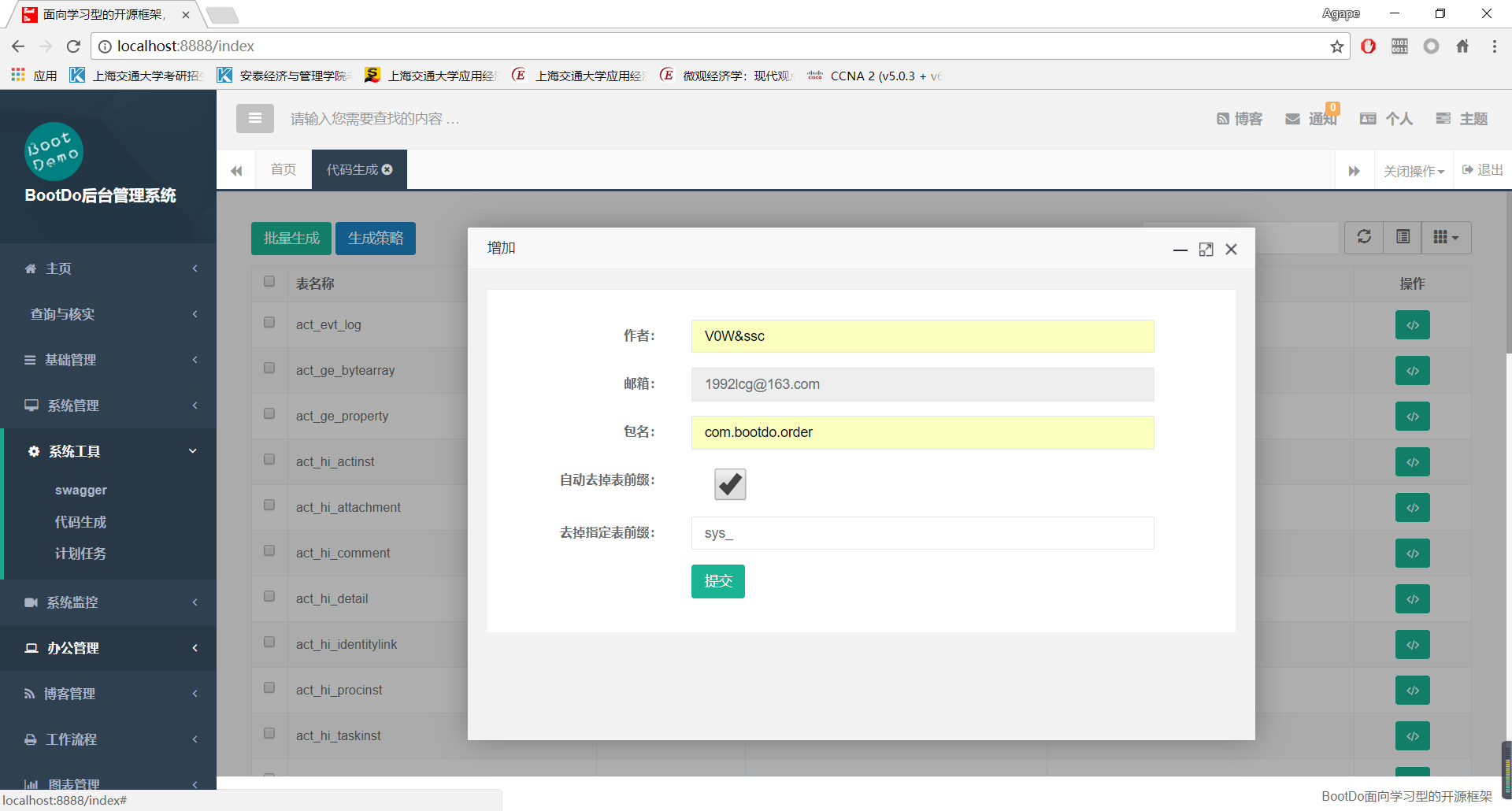
1. 登录<http://localhost:8888/login>访问后台



代码生成

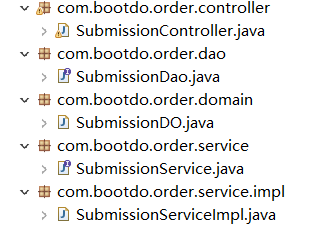
1. 配置生成策略

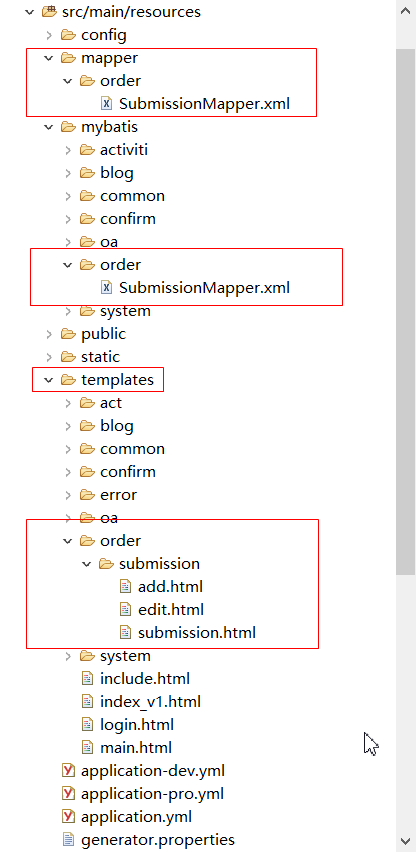
打开代码生成模块，点击生成策略，指定作者、邮箱、包名等代码注释信息



1. 生成代码并复制到项目中

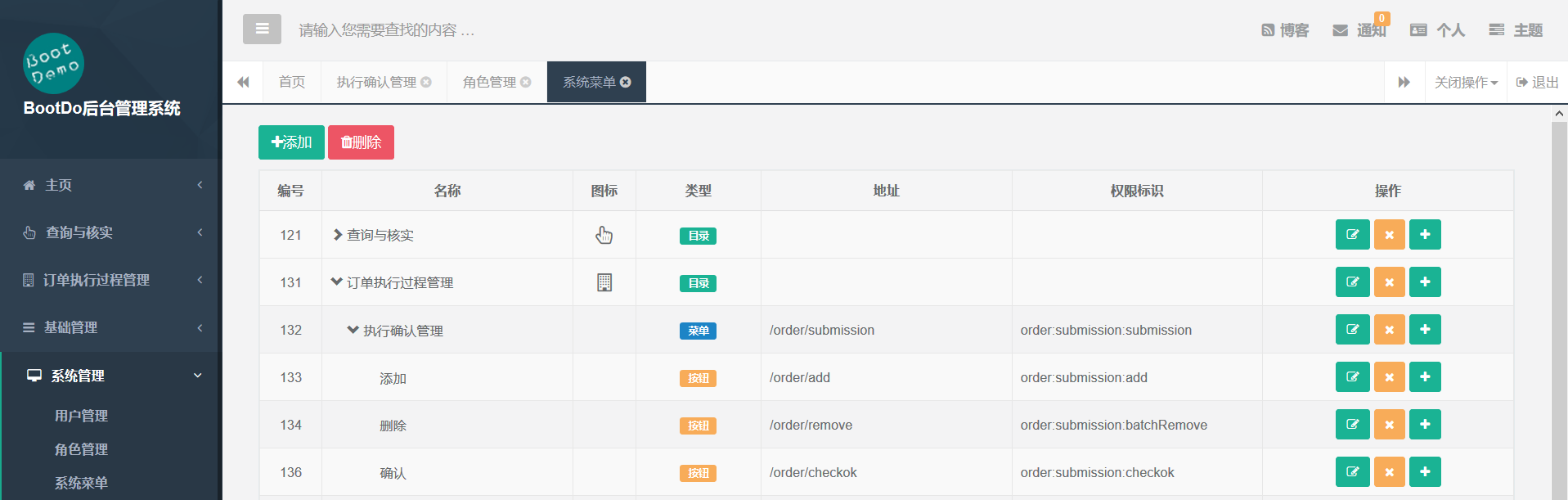






1. 设置菜单和权限

打开系统管理-系统菜单模块，设置菜单按钮等信息



打开系统管理-角色管理模块，给指定角色添加打开该模块的权限



# 实验代码

这里只给出关键部分代码。

静态文件,通过js，将ajax数据传到Controller。

1. /src/main/resources/static/js/appjs/order/submission/submission.js代码（实现交互功能和数据传递功能）：

columns : [

{

checkbox : true

},

{

field : '客户编号',

title : '客户编号'

},

{

field : '员工编号',

title : '员工编号'

},

{

field : '执行日期',

title : '执行日期'

},

{

field : '消费次数',

title : '消费次数'

},

{

title : '操作',

field : 'id',

align : 'center',

formatter : function(value, row, index) {

var e = '<a class="btn btn-primary btn-sm '+s\_edit\_h+'" href="#" mce\_href="#" title="编辑" onclick="edit(\''

+ row.客户编号

+ '\')"><i class="fa fa-edit"></i></a> ';

var d = '<a class="btn btn-warning btn-sm '+s\_remove\_h+'" href="#" title="删除" mce\_href="#" onclick="remove(\''

+ row.客户编号

+ '\')"><i class="fa fa-remove"></i></a> ';

var f = '<a class="btn btn-success btn-sm'+s\_checkok\_h+' href="#" title="确认" mce\_href="#" onclick="checkok(\''

+ row.客户编号

+ '\')"><i class="fa fa-check"></i></a> ';

return e + d + f;

}

} ]

});

}

**function** checkok(客户编号){

layer.confirm('确定要确认选中的订单记录？', {

btn : [ '确定', '取消' ]

}, **function**() {

$.ajax({

url : prefix+"/checkok",

type : "post",

data : {

'客户编号' : 客户编号

},

success : **function**(r) {

**if** (r.code==0) {

layer.msg(r.msg);

reLoad();

}**else**{

layer.msg(r.msg);

}

}

});

})

}

在/src/main/java/com/bootdo/order/controller/SubmissionController.java中，处理相应的请求和数据（主要实现用户编号的用户，消费次数减一的功能）：

/\*\*

\* check OK

\*/

@PostMapping( "/checkok")

@ResponseBody

@RequiresPermissions("order:submission:checkok")

**public** R checkok(String 客户编号){

SubmissionDO smd = submissionService.get(客户编号);

Integer cnt = smd.get消费次数();

**if** (cnt > 0) {

cnt -= 1;

}**else**{

**return** R.*error*();

}

smd.set消费次数(cnt);

**if**(submissionService.update(smd)>0)

**return** R.*ok*();

**return** R.*error*();

// if(submissionService.checkok(客户编号)>0){

// return R.ok();

// }

// return R.error();

}

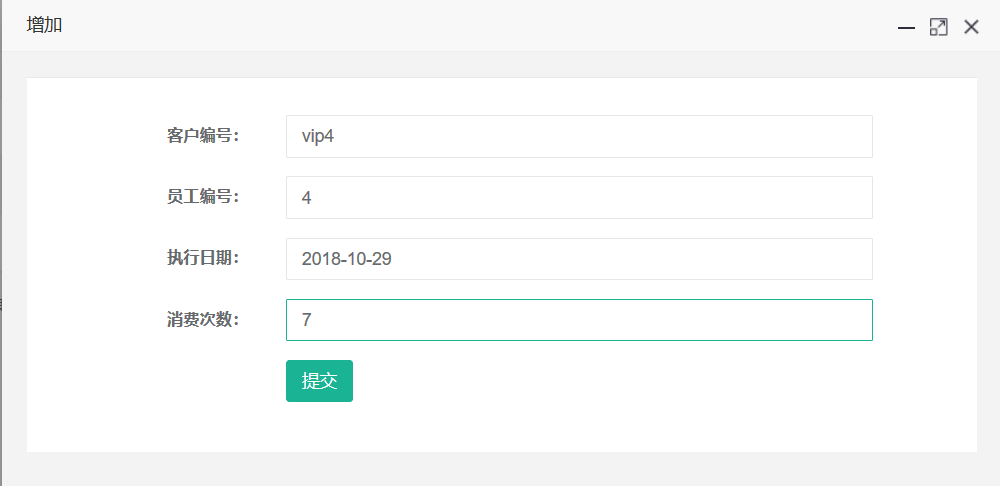
}

# 实验结果

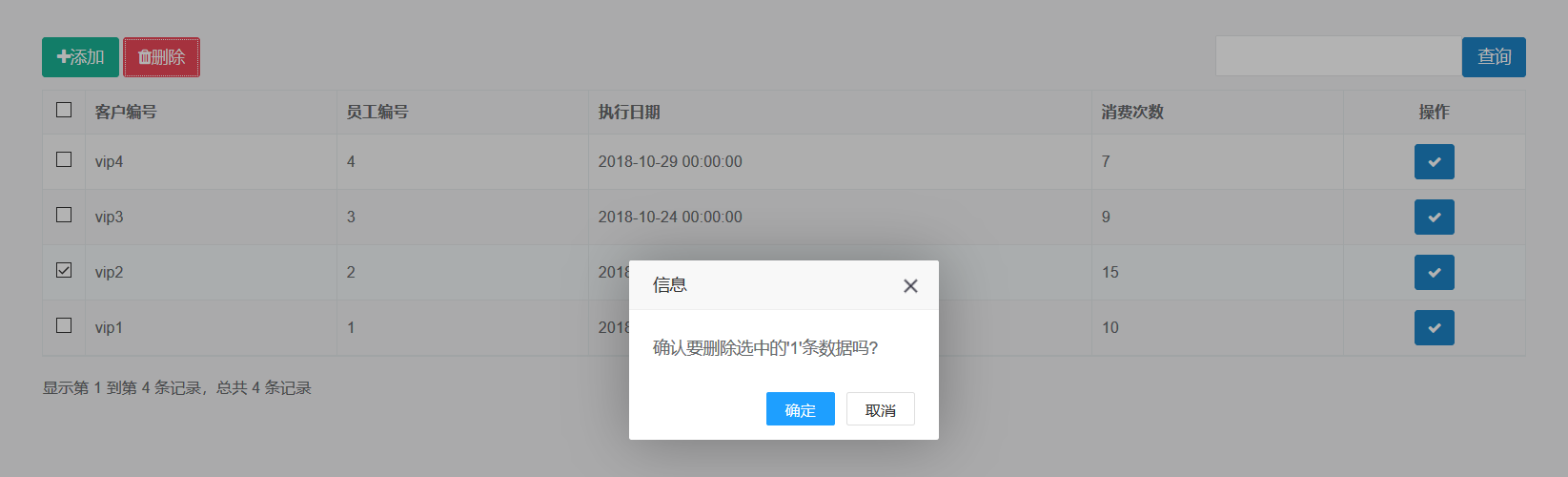
1. 总体概况预览



1. 点击添加按钮，可添加客户信息

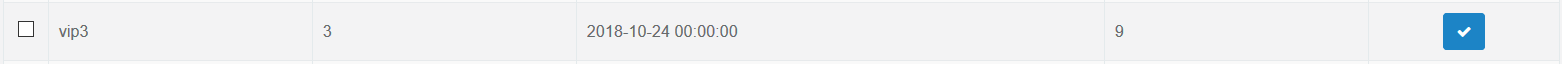


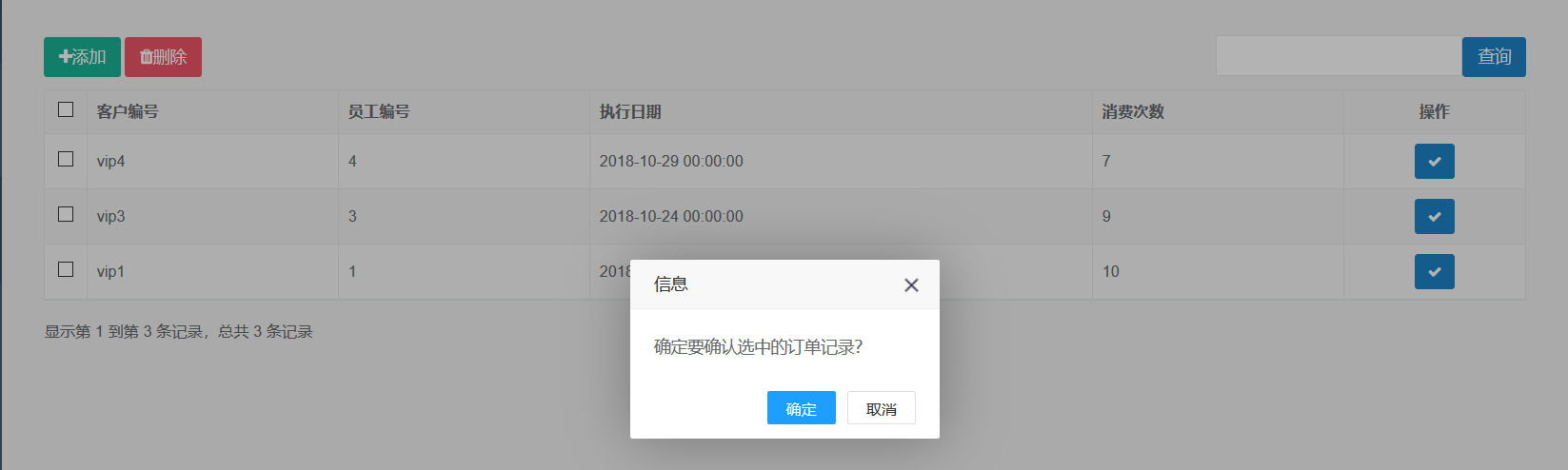
1. 选中某一客户，点击删除按钮可删除该客户信息





1. 客户消费一次，可点击操作按钮使客户消费次数信息自动减一







# 实验感悟

Bootdo是一个后台管理系统的高效开发框架，Springboot作为基础框架，使用mybatis作为持久层框架，与数据库交互。使用官方推荐的thymeleaf做为模板引擎，shiro作为安全框架,主流技术，“一网打尽”。通过本次大作业的学习和完成，我们深入理解和体会了SpringBoot的奇妙之处，还有mybatis数据库交互的方便之处。为今后的web应用开发和进一步学习奠定了基础和不错的开端。