柠檬手机销售管理系统开发文档

1. **需求分析**

1.1功能需求

1.1.1财务管理

系统应有财务管理模块，维护公司的盈利以及员工工资。

工资管理：系统应在每个月的1号进行员工工资的维护，超过该日期不可维护员工工资。本月维护上个月员工工资以及维护实时的税率。

盈利管理：系统应在每个月的1号进行盈利的维护，本月维护的是上个月的盈利情况，只有在完成维护上个月员工工资的情况下，才能维护盈利。系统应有维护其他杂费的功能。

1.1.2统计分析

系统应有对本月以前的销售额，客户注册人数，每个月的进货数以及杂费的 数据对比。

1.1.3销售订单

系统应有生成订单的维护。

该维护包括订单的生成，订单的编辑，订单的审批以及订单的删除等功能， 该功能只对管理员，销售人员以及销售经理开放。每一个订单只有在审批通过后 才会形成真正的订单。

1.1.4部门管理

系统应有部门管理分模块。

该模块管理公司的所有部门。功能包括:部门的新增，修改以及查看每个部

门下员工信息等功能。

1.2浏览器兼容需求

表1.1浏览器需求表

|  |  |  |
| --- | --- | --- |
| **序号** | **浏览器** | **支持版本** |
| 1 | Firefox | 当前主流版本 |
| 2 | Chrome | 当前主流版本 |
| 3 | Internet Explorer | 10.0以上版本 |

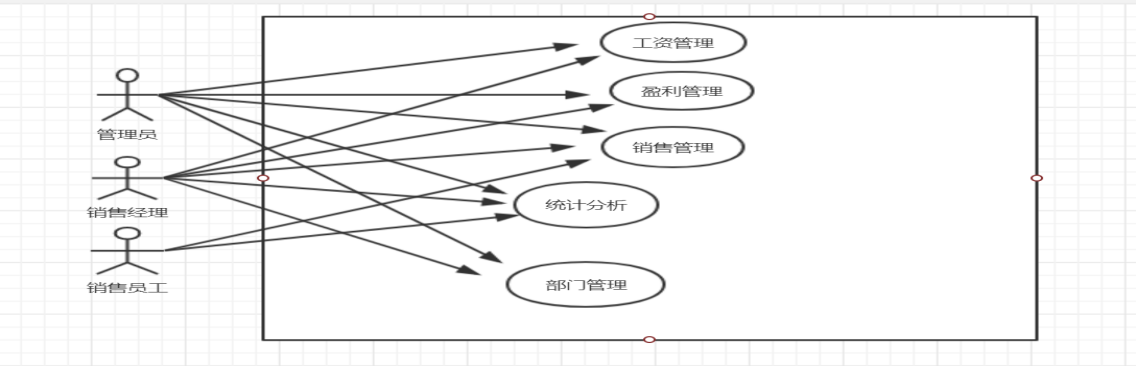


图 1.1系统用例图

1. **概要设计**

2.1.总体设计

2.1.1总体结构

用户层

View(视图）层)

javascript

JQuery(ajax)

Controller(控制层)

ThinkPHP

Controller

Modal(模型层)

ThinkPHP

Modal

数据库

图2.1系统总体结构图

2.2.模块设计

2.2.1模块设计

我负责的模块包括财务管理(其中包括子模块工资管理和盈利管理)，销售管理，统计分析以及部门管理。

2.2.2模块之间关系

财务模块与销售模块。每个月盈利与每个月销售额有着密切的关系，成正比系，

并且与成本成反比关系。

统计分析中成本与利润的对比，是根据每个月进货成本以及每个月的开销(如水

电费等)和每个月净销售额通过柱状图的形式展现出来。

部门模块对于其他三个模块是相对独立的一个模块。

销售模块，可直接形成销售订单，得出每一个的销售利润，直接影响着财务模

块与统计分析中成本与利润的对比关系。

2.3各模块详细设计

2.3.1财务模块

管理每个月员工工资以及每个月盈利状况。

员工工资的管理，输入员工信息通过接口形成一条员工工资的详情，展现在用

户面前，通过接口删除该员工工资信息，通过接口修改工资信息。

盈利管理，维护上个月的水电等杂费，通过接口系统自动得出上个月盈利详细信息。

2.3.2销售模块

形成销售订单。每一单的销售订单通过接口传输到控制层，在控制层进行数据的验证，以及处理，在通过modal层存到数据库中。

2.3.3统计分析模块

通过接口获取相应数据进行处理转化传到前端，前端通过框架来处理显示得到的数据，并通过计时器来规定显示周期。

2.3.4部门模块

该模块的作用为管理部门的基本信息。部门的新增，通过接口传输json数据到后端，经过处理存到数据库中。以用来添加员工使用。

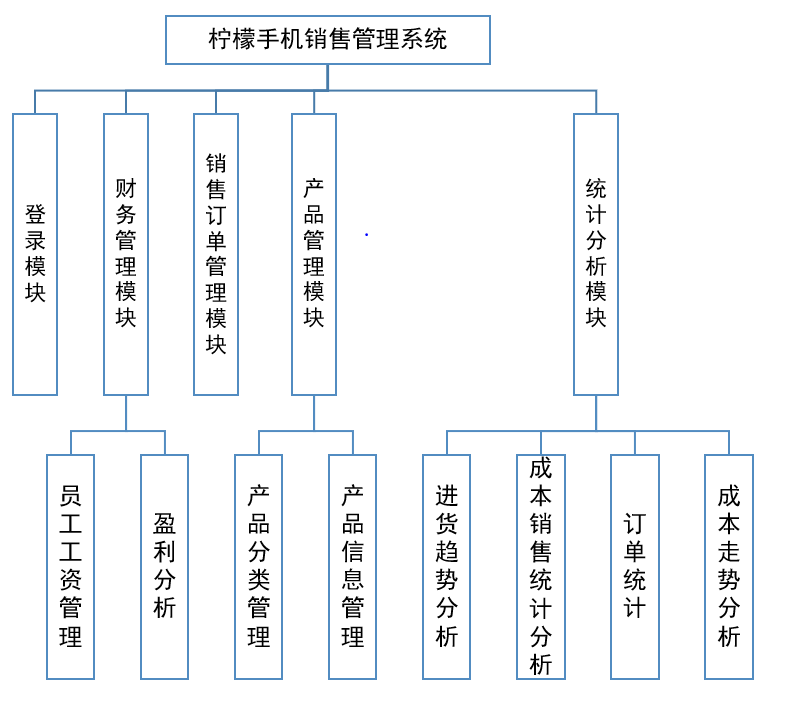


图 2.2系统功能模块

**三、详细设计**

3.1.财务管理

3.1.1工资管理

通过列表显示本月以前的所有员工工资信息，列表信息把包括员工姓名，所属 部门，销售提成，个人所得税，所得工资，状态，创建时间，操作包括审批和删除，

已审批的工资单不能再审批。员工工资单的维护应在本月的1号进行维护，超过该日将不能进行工资维护。该页面包括税率的维护，维护5001-8000，8001-17000以及销售提成。该页面还包括工资维护通过维护的税率和销售提成以及本月总工资系统自动生成所得工资。

3.1.2盈利管理

通过列表显示本月以前的盈利情况，列表包括月份，销售额，员工工资，产品花销，房租，电费，水费，总利润，操作包括删除。该页面应有维护电费等杂费的功能，盈利维护有在员工工资维护的前提下，以及本月的1月才能形成盈利账单。

3.1.3涉及到接口：

接口名： deleteWage

功能： 删除工资详情

参数： id

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： approveWage

功能： 审批员工工资

参数： id

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： saveWage

功能： 形成员工工资详情

参数： manger\_id(员工id)，department(部门id)，allWage(总工资),surplusWage(税后工资)，bonus(应发工资)

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： deleteProfit

功能： 删除盈利详情

参数： id

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： saveProfit

功能： 新增盈利详情

参数： rent(房租),electricity(电费),water(水费)

返回数据： code: 1,msg:’操作成功’，href:’’

3.2销售管理

3.2.1销售订单

通过列表显示每单信息情况，列包括订单编号，产品名称，订单客户，预定数量，预定价格，状态，创建人，操作包括删除，审批。每一单订单的生成，必须审批通过以后才能进行发货，库存数的操作，再次之前将只是一单无效订单。该页面应包括形成新单订的功能。新订单的形成，通过选择产品系统提取出该产品的相关信息如库存数量，售价等信息。提交订单的时候要判断预定数量与库存之间的关系。审批成功的订单不能再次审批。

3.2.2涉及到接口：

接口名： deleteOrderInfo

功能： 新增订单详情

参数： id

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： approve

功能： 审批订单

参数： id

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： getProductInfo

功能： 获取产品信息

参数： id(产品编号)

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： saveOrderInfo

功能： 生成订单信息

参数： no(产品编号),client(客户id),product(产品id),num(预定数量),price(订单价格)

返回数据： code: 1,msg:’操作成功’，href:’’

3.3.统计分析

3.3.1销售分析

通过视图的形式显示每月订单走势，每月产品进货数量，成本与销售额之间的关系以及每月杂费的情况。控制层调用modal层接口获取相应的数据信息，并将获得的数据处理提取出前端需要的数据，并封装成json格式数据，前端通过接口获取该数据。推荐使用百度的Echart框架。

3.3.2涉及到接口：

接口名： incidental

功能： 获取每个月水电等杂费信息

参数：

返回数据： data(json)

接口名： costTrend

功能： 获取每个月成本与销售利润信息

参数：

返回数据： data(json)

接口名： productTrend

功能： 获取每个月进货数量

参数：

返回数据： data(json)

接口名： orderTrend

功能： 获取每个月订单数

参数：

返回数据： data(json)

3.4.部门管理

3.4.1部门设置

通过列表显示每个部门的基本信息。列包括部门编号，部门名称，部门描述，操作包括修改，详情。修改只能修改部门的名称以及描述，编号一旦生成则不能修改。点击详情显示该部门下的员工信息情况。该页面应有新增部门的功能，维护部门编号，名称以及描述，描述主要讲述该部门的性质。

3.4.2涉及到接口：

接口名： addDepart

功能： 新增部门信息

参数： no(部门编号),name(部门名称),desc(部门描述)

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： department

功能： 获取部门信息

参数： id

返回数据： data(json)

接口名： getModify

功能： 修改部门信息

参数： id(部门id),name(部门名称),desc(部门描述)

返回数据： code: 1,msg:’操作成功’，href:’’

接口名： getDepInfo

功能： 获取对应部门下员工信息

参数： id(部门id)

返回数据： data(json)

四**、系统设计**

**4.**1.各实体之间的E-R图

订单，部门以及工资部门属于我的。

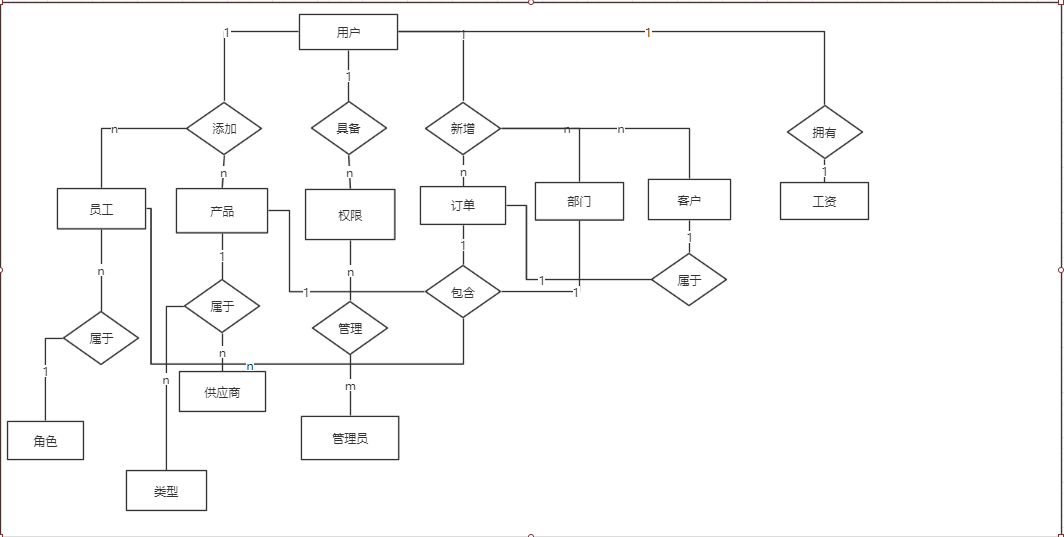


图4.1柠檬手机管理系统的E-R图

**4.**2.数据库逻辑结构设计

我们将概念结构设计阶段的E-R图转化为MySQL数据库中支持的数据模型相

符合的逻辑结构，也就是E-R图向关系模型转换。

（1）部门表的具体信息如表4.1所示：

表4.1 部门表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Department（部门） | | | | | |
| 字段名 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| Id | int(11) |  | N | Y | id |
| department\_id | varchar(10) |  | N |  | 部门编号 |
| department\_name | varchar(16) |  | N |  | 部门名称 |
| department\_desc | varchar(16) |  |  |  | 部门描述 |

1. 工资表的具体信息如表4.2所示：

表4.2 工资表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Wage(工资表) | | | | | |
| 字段名 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| Id | int |  | N | Y | Id |
| manger\_id | int |  | N |  | 员工 |
| dep\_id | int |  | N |  | 员工部门 |
| allWage | varchar |  | N |  | 总工资 |
| surplusWage | int |  | N |  | 扣税工资 |
| bonus | Int |  | N |  | 提成 |
| months | int |  | N |  | 月份 |

1. 订单表具体信息如表4.3所示：

表4.3 订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order(订单表) | | | | | |
| 字段名 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| Id | int |  | N | Y | id |
| order\_no | varchar |  | N |  | 订单编号 |
| pro\_id | int |  | N |  | 订购产品 |
| manger\_id | int |  | N |  | 形成订单员工 |
| client\_id | int |  | N |  | 订单客户 |
| pro\_num | int |  | N |  | 订购数量 |
| price | int |  | N |  | 订购价格 |
| state | int |  | N |  | 是否审批(0-未审批 1-已审批) |
| months | int |  | N |  | 添加月份 |

1. 利润表具体信息如表4.4所示：

表4.4 利润表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Profit(利润表) | | | | | |
| 字段名 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| Id | int |  | N | Y | Id |
| months | int |  | N |  | 月份 |
| manger\_wage | int |  | N |  | 员工总工资 |
| product\_wage | int |  | N |  | 成本总数 |
| order\_wage | int |  | N |  | 销售总额 |
| rent | Int |  | N |  | 房租 |
| electricity | int |  | N |  | 电费 |
| charge\_for\_water | Int |  | N |  | 水费 |

**五、系统实现**

5.1.财务管理

5.1.1工资管理



图5.1工资管理

实现工资管理，需要维护税率维护以及工资维护

5.1.1.1实现维护税率

$('#save').click(function (e) {

e.preventDefault();

var tax5 = $('#tax5').val();

var tax8 = $('#tax8').val();

var royalty = $('#sale').val();

if (tax5 == '' || tax8 == '') {

msg("错误","请填写正确的税率");

return;

}

if (royalty == '') {

msg("错误","请填写每单相应的提成");

return;

}

var info = {

id: $('#id').val(),

tax5: tax5,

tax8: tax8,

royalty: royalty

};

$.ajax({

type: "post",

url: "{:url('index/index/updateTaxRate')}",

data: info,

dataType: "json",

success: function (data) {

if (data.code == 1) {

layer.msg(data.msg, {

icon: 6,

time: 2000,

}, function () {

location.href = data.url;

});

} else {

layer.open({

title: '操作失败',

icon: 2,

content: data.msg,

});

}

}

})

});

5.1.1.2实现维护工资

if ($(this).val() == 0) {

return;

}

var name = {

id: $(this).val()

};

$.ajax({

type: "post",

url: "{:url('index/index/getMangerInfo')}",

data: name,

dataType: "json",

success: function (data) {

if (data) {

depart = data.department.id;

tax1 = data.tax.tax5;

tax2 = data.tax.tax8;

sale = data.tax.royalty;

$('#saleNum').val(data.order);

$('#shouldGet').val(data.order \* sale);

saleNum = data.order;

shouldGet = data.order \* sale;

} else {

msg("错误","获取数据错误，请联系管理员");

}

5.1.2盈利维护

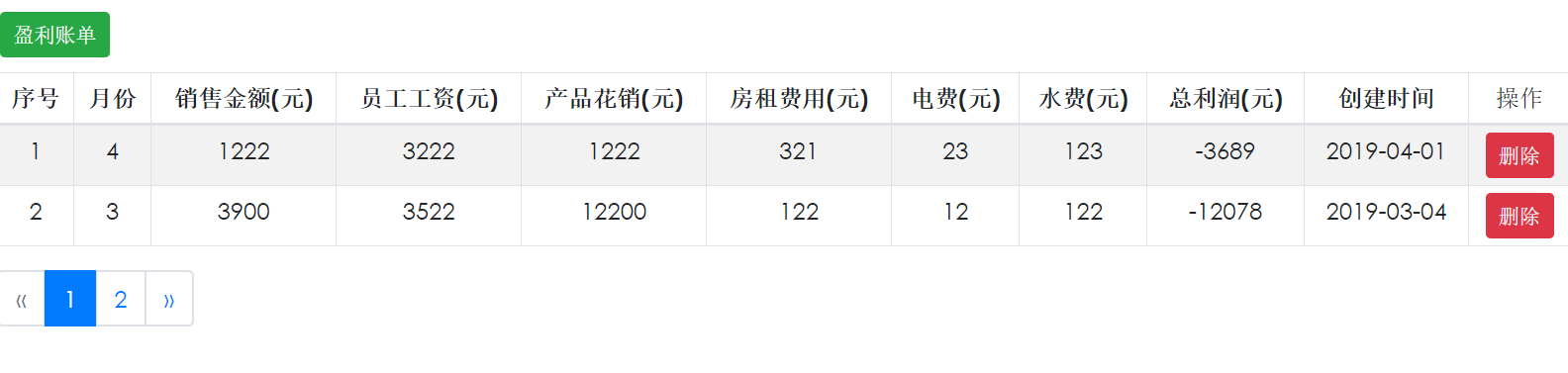


图5.2盈利维护

5.1.2.1实现维护账单

$('#save').click(function (e) {

e.preventDefault();

var myDate = new Date();

var date = myDate.getDate();

if (date != 1) {

msg("错误","未到结算日，不能形成上月利润账单");

return;

}

var info = {

rent: $('#rent').val(),

electricity: $('#electricity').val(),

water: $('#water').val(),

};

$.ajax({

type: "post",

url: "{:url('index/index/saveProfit')}",

data: info,

dataType: "json",

success: function (data) {

if (data.code == 1) {

layer.msg(data.msg, {

icon: 6,

time: 2000,

}, function () {

location.href = data.url;

});

} else {

layer.open({

title: '操作失败',

icon: 2,

content: data.msg,

});

}

}

});

});

5.2.销售管理

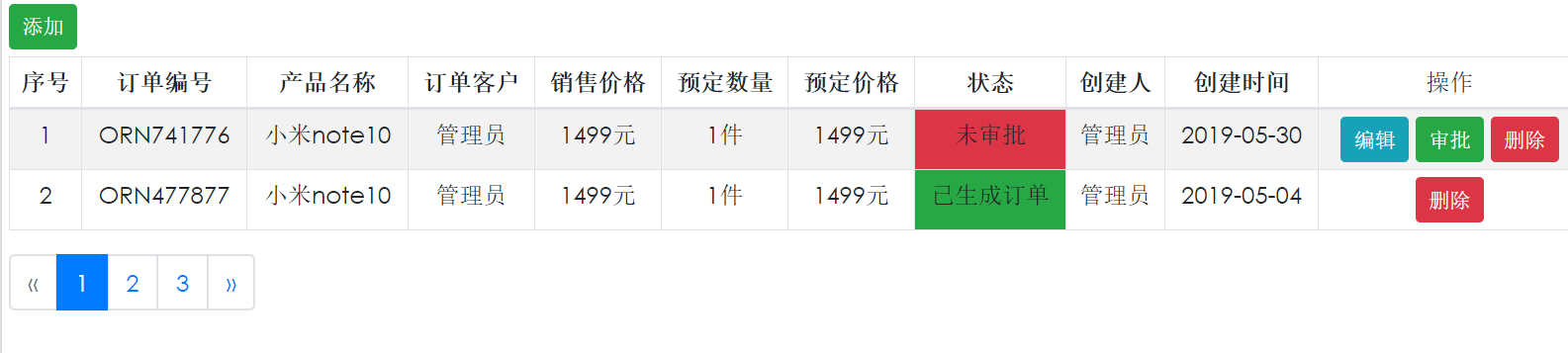


图5.3销售管理

5.2.1实现添加功能

$('#save').click(function (e) {

e.preventDefault();

if (wantNum > surplusNum) {

msg('警告','预定数量大于库存数量，请重新选择！');

return;

}

var orderInfo = {

no: $('#manger\_no').val(),

client: $('#client').val(),

product: $('#product').val(),

num: wantNum,

price: wantPrice

};

$.ajax({

type: "post",

url: "{:url('index/index/saveOrderInfo')}",

data: orderInfo,

dataType: "json",

success: function (data) {

if (data.code == 1) {

layer.msg(data.msg, {

icon: 6,

time: 2000,

}, function () {

location.href = data.url;

});

} else {

layer.open({

title: '删除失败',

icon: 2,

content: data.msg,

});

}

}

});

});

5.2.2实现审批功能

$('.approve').click(function (e) {

e.preventDefault();

var approve = {

id: $(this).attr('data-id')

};

$.ajax({

type: "post",

url: "{:url('index/index/approve')}",

data: approve,

dataType: "json",

success: function (data) {

if (data.code == 1) {

layer.msg(data.msg, {

icon: 6,

time: 2000,

}, function () {

location.href = data.url;

});

} else {

layer.open({

title: '操作失败',

icon: 2,

content: data.msg,

});

}

}

});

});

5.3.统计分析

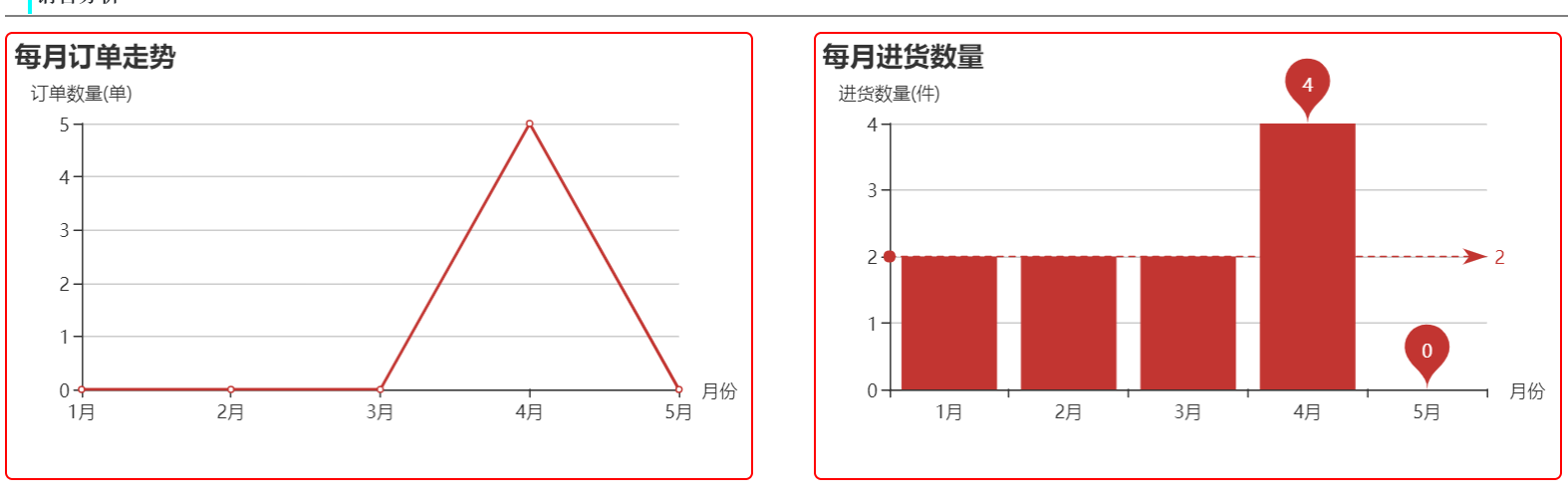


图5.4走势图

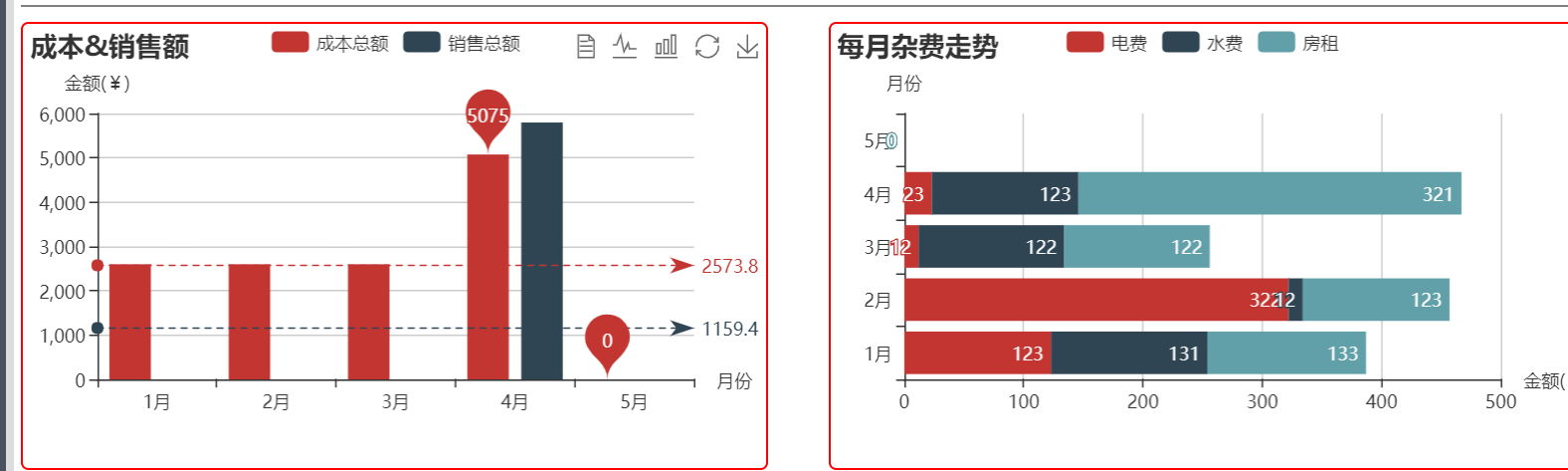


图5.5走势图

5.3.1订单走势实现

$.ajax({

url: "{:url('index/index/orderTrend')}",

type: 'post',

dataType: 'json',

success: function (data) {

var cateDate = [];

var cateCbm = [];

for (var i = 0; i < data.length; i++) {

var date = data[i].month + '月';

var num = data[i].num;

cateDate.push(date);

cateCbm.push(num);

}

option = {

title : {

text: '每月订单走势',

subtext: ''

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

}

},

calculable : true,

xAxis : [

{

type : 'category',

name: '月份',

boundaryGap : false,

data :cateDate

}

],

yAxis : [

{

type : 'value',

name: '订单数量(单)'

}

],

series : [

{

name:'订单数量(单)',

type:'line',

stack: '单数',

data:cateCbm

}

]

};

var myChart = echarts.init(document.getElementById('throughput3'));

myChart.setOption(option);

}

5.3.2杂费走势实现

for (var o = 0; o < data.electricity.length; o++) {

var date = data.electricity[o].month + '月';

var num = data.electricity[o].num;

inDate.push(date);

inCbm.push(num);

}

// 水费

for (var p = 0; p < data.water.length; p++) {

water.push(data.water[p].num);

}

// 房租

for (var m = 0; m < data.rent.length; m++) {

rent.push(data.rent[m].num);

5.3.3进货数走势

根据返回的信息封装数据

for (var i = 0; i < data.length; i++) {

var date = data[i].month + '月';

var num = data[i].num;

proDate.push(date);

proCbm.push(num);

}

**5.**3.4成本与利润对比

option = {

title : {

text: '每月订单走势',

subtext: ''

},

tooltip : {

trigger: 'axis'

},

toolbox: {

show : true,

feature : {

}

},

calculable : true,

xAxis : [

{

type : 'category',

name: '月份',

boundaryGap : false,

data :cateDate

}

],

yAxis : [

{

type : 'value',

name: '订单数量(单)'

}

],

series : [

{

name:'订单数量(单)',

type:'line',

stack: '单数',

data:cateCbm

}

]

5.4.部门管理

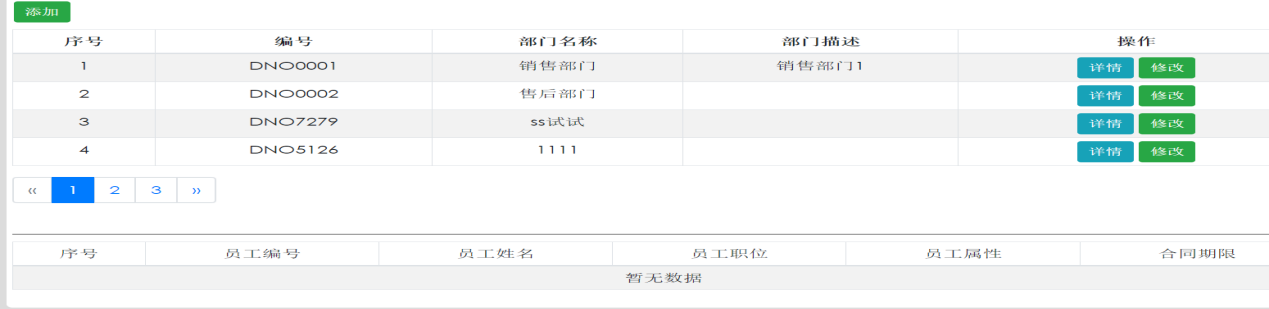


图5.6部门管理

5.4.1部门添加功能

if (data.code == 1) {

layer.msg(data.msg, {

icon: 6,

time: 2000,

}, function () {

location.href = data.url;

});

5.4.2显示详情功能实现

if (data.length > 0) {

for (var i = 0; i < data.length; i++) {

if (data[i].incumbency == '0') {

incumbency = '合同员工';

} else {

incumbency = '在职员工';

}

if (data[i].jobyear == null) {

jobyear = '无';

} else {

jobyear = data[i].jobyear;

}

depHtml += '<tr><td>'+(i+1)+'</td><td>'+data[i].manger\_no+'</td><td>'+data[i].manger\_name+'</td><td>'+data[i].role.role\_name+'</td><td>'+incumbency+'</td><td>'+jobyear+'</td></tr>'

}

**六、软件测试**

通过postman软件进行测试主要接口

1. 生成销售订单接口测试如表6.1所示

表6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | saveOrderInfo | | | |
| **功能** | 生成销售订单 | | | |
| **请求参数** | **参数名** | **数据类型** | **是否必填** | **备注** |
| No | String | Y | 订单编号 |
| Client | Int | Y | 客户 |
| Product | Int | Y | 产品 |
| Num | Int | Y | 产品数量 |
| Price | Int | Y | 订单价格 |
| **应答** | Code | Int | | 返回码 |
| Msg | String | | 返回信息 |
| Href | String | | 返回路径 |
| **实例** | **请求** | Post：[http://www.phone.com/phone/](http://www.phone.com/phone/login)index/index/saveOrderInfo  {  no:’’,  client:’’,  product: ,  num: ,  price:  } | | |
| **应答** | {  code: 1,  msg:’操作成功’  href:’’  } | | |

（2）工资接口测试如表6.2所示

表6.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | saveWage | | | |
| **功能** | 维护员工工资 | | | |
| **请求参数** | **参数名** | **数据类型** | **是否必填** | **备注** |
| manger\_id | int | Y | 员工id |
| department | int | Y | 部门 |
| allwage | int | Y | 总工资 |
| surplusWage | int | Y | 扣税工资 |
| bouns | int | Y | 销售提成 |
| **应答** | code | int | | 返回码 |
| msg | string | | 返回信息 |
| href | string | | 返回路径 |
| **实例** | **请求** | Post [http://www.phone.com/phone/](http://www.phone.com/phone/login)index/index/saveWage  {  manger\_id: ‘’,  department:’’,  allWage:’’,  surplusWage: ‘’,  bouns: ‘’  } | | |
| **应答** | {  code: 1,  msg:’操作成功’  href:’’  } | | |

（3）维护销售利润接口测试如表6.3所示

表6.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名** | saveProfit | | | |
| **功能** | 维护公司利润 | | | |
| **请求参数** | **参数名** | **数据类型** | **是否必填** | **备注** |
| rent | int | Y | 房租 |
| electricity | int | Y | 电费 |
| water | int | Y | 水费 |
| **应答** | code | int | | 返回码 |
| msg | string | | 返回信息 |
| href | string | | 返回路径 |
| **实例** | **请求** | Post [http://www.phone.com/phone/](http://www.phone.com/phone/login)index/index/saveProfit  {  rent: ‘’,  electricity:’’,  water: ‘’  } | | |
| **应答** | {  code: 1,  msg:’操作成功’  href:’’  } | | |

**七、软件完成后的总结汇报**

以下为设计和编程过程中遇到的问题：

（1）前期项目管理不好，每个人的程序合并麻烦，中后期使用git管理。

（2）数据库搭建表的时候没有得到很好的设计，在编程的过程中修改了好几次。

（3）前端布局css样式过于繁琐。

（4）编程过程中数据处理错误，如在计算本月以前的公司利润的时候，开始时没

有考虑杂费等费用的影响，修改了好几次。

（5）编程中出现好多不应该不出现的失误，如往后端传值类型的错误或者根本没有传过去值。

（6）在处理统计分析的时候，由于数据来源好几个表，没有做到系统的数据管理，所以只能单表查询，整合起来传到前端。

综上为开发过程中所有遇到的问题。