**“飞泊通”客服管理系统软件说明书**

目录

[１.引言 2](#_Toc441503787)

[２.概述 3](#_Toc441503788)

[2.1 整个系统原理流程图 3](#_Toc441503789)

[2.2 系统软件流程图 4](#_Toc441503790)

[3. 功能介绍 5](#_Toc441503791)

[3.1 系统主界面 5](#_Toc441503792)

[4. 操作流程说明 6](#_Toc441503793)

[4.1 打开软件登录界面 6](#_Toc441503794)

[4.2 系统首页的订单查询操作 6](#_Toc441503795)

[4.3 订单管理操作 8](#_Toc441503796)

[4.3.1 选择不同状态的订单 9](#_Toc441503797)

[4.3.2 对订单进行操作 11](#_Toc441503798)

[4.4 报表查询操作 13](#_Toc441503799)

[4.4.1 机场月度报表 13](#_Toc441503800)

[4.4.2 工作量统计表 13](#_Toc441503801)

[4.4.3 服务效率统计表 14](#_Toc441503802)

[4.4.4 预定订单统计表 14](#_Toc441503803)

[4.5活动查看操作 15](#_Toc441503804)

[4.6优惠券操作 17](#_Toc441503805)

[附录 18](#_Toc441503806)

# １.引言

在当今铺天盖地的互联网行业的快速发展中，人们的生活水平在不断提高，各种不同的需求也越来越多，为了满足这些需求，对于软件开发人员的技术要求也变得更高。当软件在进行大量的数据操作的过程中，有很多时候为了防止数据出错，需要人工实时的进行检查去判断数据是否出错，不过在进行大批量数据的检查与操作过程中，由于数据的条目可能是成千上万条的，这时的人工的检查与操作就存在了一定的弊端，例如：检查效率低、检查容易遗漏、操作数据容易出错等。

“飞泊通”客服管理系统可以实时全面的对用户的订单信息进行检查和操作，对半个小时内出现的紧急订单能及时报警并提示客服工作人员尽快处理，从而大大提高生产效率，降低出错率，避免了由于订单量过大而导致的错误造成不必要的损失，也进一步降低生产成本。搭建“飞泊通”客服管理系统的目的是为公司完整地认识整个客户生命周期，提供与客户沟通的统一平台，提高员工与客户接触的效率和客户反馈率，同时也为公司日常服务管理提供数据依据，不断优化人员管理和流程管理。

# ２.概述

“飞泊通”客服管理系统软件的构成主要是由：订单查询模块、订单管理模块、报表查询模块、活动查看模块、优惠券模块以及紧急订单报警模块组成。订单操作：包括立即派单、填写返程时间、备注订单、置空返程信息、取消订单。

## 2.1 整个系统原理流程图

系统的运行原理是建立在订单状态变迁上，将不同状态的订单通过查询条件分别展示出来。

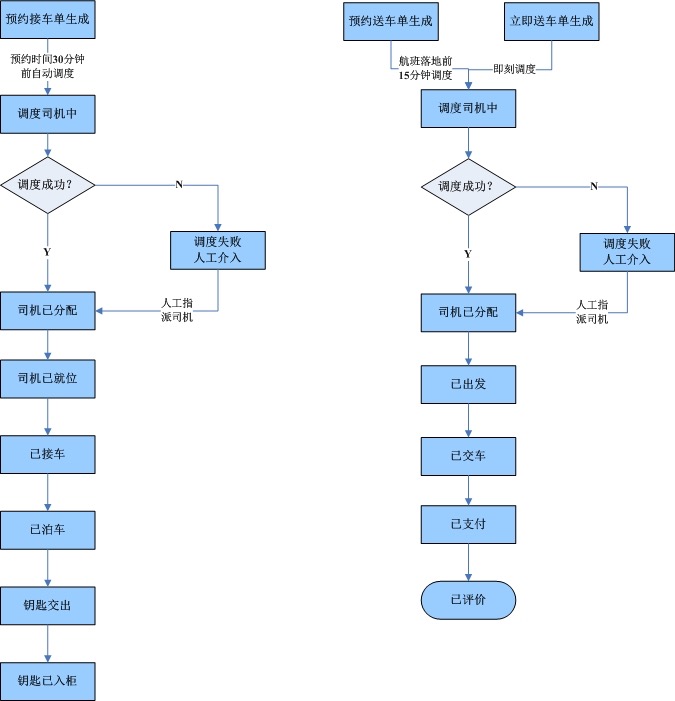


图2.1系统原理流程图

## 2.2 系统软件流程图

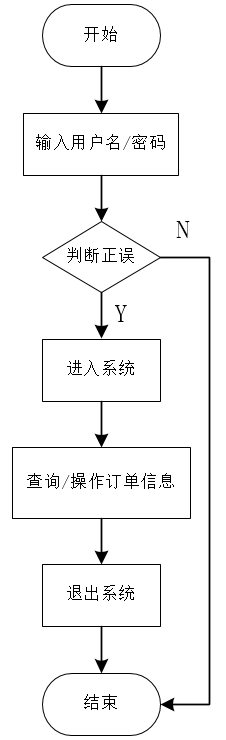


图2.2系统软件流程图

# 3. 功能介绍

## 3.1 系统主界面

“飞泊通”客服管理系统首页主要由订单查询、订单列表和订单详情组成，如图3.1所示。客服人员通过输入与订单相关的信息去搜索订单，点击由“姓名/手机号”组成的“用户”就能够显示该订单的详情，如图3.2所示。

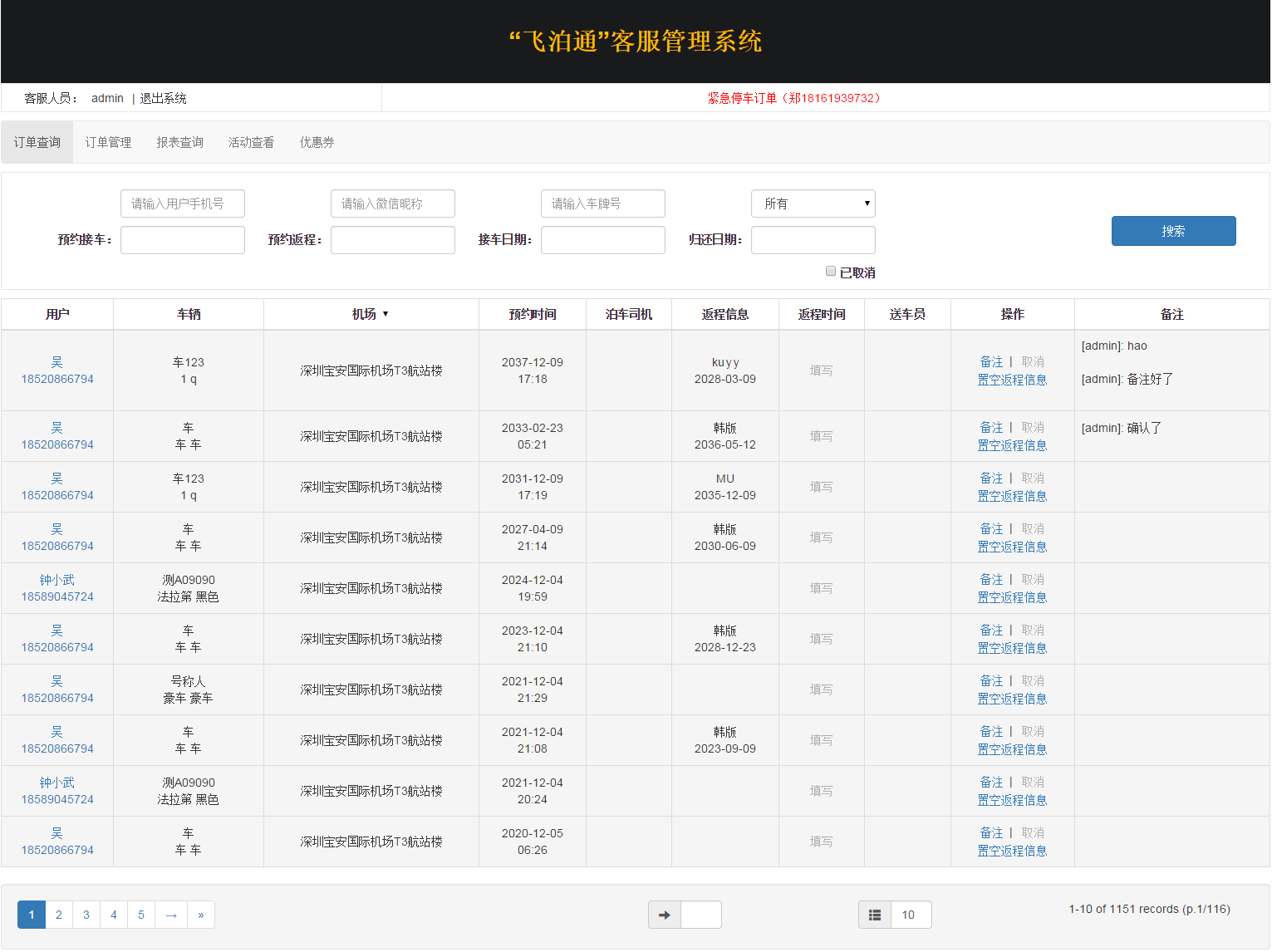


图3.1系统首页



图3.2订单详情

# 4. 操作流程说明

## 4.1 打开软件登录界面

输入用户名、密码进行验证登录，登录界面如图4.1所示。



图4.1系统登录界面

## 4.2 系统首页的订单查询操作

登录进入首页后，30分钟内需要处理的紧急预约订单会以跑马灯滚动（如图4.2所示）、弹出框（如图4.3）所示这两种方式提示客服人员，这里以手机号“18161939732”查询为例，在待处理的订单弹出框提示中选择手机号进行复制。



图4.2滚动的紧急订单信息



图4.3弹出框的紧急订单信息

将复制好的手机号码粘贴到搜索框中，点击“搜索”按钮，查询到该用户订单的信息结果如图4.4所示。



图4.4使用用户手机号搜索订单

如果用户是通过“飞泊通”微信公众号进行预约的订单，还可以通过该用户的微信昵称查询到用户的订单信息，以微信昵称为“ifeng”为例，搜索到的结果如图4.5所示。



图4.5使用用户微信昵称搜索订单

用户的订单中包含了车牌号码，所以也可以通过用户的车牌号查询，以车牌号为“测A88888”查询到用户的订单信息，使用下拉进行筛选到“虹桥国际机场”的订单信息如图4.6所示。



图4.6使用用户车牌号搜索并筛选的订单

此外，还可以通过用户的预约接车时间、预约返程时间、接车日期、归还日期中的一个具体日期来查询到用户的相应订单，如果选择“已取消”的选择框，则可以筛选出已取消的订单，如图4.7是预约接车时间为“2016-01-04”、接车地点为“虹桥国际机场”、“已取消”的订单信息。



图4.7筛选出已取消的订单信息

如果筛选后的订单量显示非常多，系统默认会只显示10条订单信息，这时可以在页脚的分页筛选选择跳转到指定的页面，也可以指定该页面想要显示的记录条数来过滤订单，如图4.8所示为显示2条“已取消”的订单。



图4.8显示2条“已取消”的订单

若查询到的订单是新预约的订单，客服人员可以对订单信息进行确认并备注，以便于分配司机、填写返程时间，当客户有变动也可以置空用户的返程信息或者取消未分配司机的订单。

## 4.3 订单管理操作

订单管理对于减轻客服人员的工作效率有非常大的帮助，在订单管理中可以直接看到每个状态的订单，操作起来非常方便，如图4.9所示。需要查看订单的详情只需点击由“姓名/手机号”组成的“用户”就能够查看该订单的详情，如图4.10所示。



图4.9订单管理页面



图4.10订单详情

### 4.3.1 选择不同状态的订单

点击“待联系订单”，订单对应的订单状态为“预约停车”如图4.11所示。



图4.11订单状态为“预约停车”的订单

点击“待分配接车单”，订单对应的订单状态为“客服已确认”如图4.12所示。



图4.12订单状态为“客服已确认”的订单

点击“在进行接车单”，订单对应的订单状态为“已泊车”如图4.13所示。



图4.13订单状态为“已泊车”的订单

点击“在库车辆”，订单对应的订单状态为“钥匙交出”如图4.14所示。



图4.14订单状态为“钥匙交出”的订单

点击“待分配送车单”，订单对应的订单状态为“立即取车”如图4.15所示。



图4.15订单状态为“立即取车”的订单

点击“在进行送车单”，订单对应的订单状态为“已分配送车司机”如图4.16所示。



图4.16订单状态为“已分配送车司机”的订单

点击“已完成订单”，订单对应的订单状态为“已完成”如图4.17所示。



图4.17订单状态为“已完成”的订单

点击“未付款订单”，订单对应的订单状态为“已送还”如图4.18所示。



图4.18订单状态为“已送还”的订单

### 4.3.2 对订单进行操作

客服人员对“待联系订单”、状态为“预约停车”的订单进行确认后，需要对该订单进行添加备注，然后点击“下一步”，这时订单的状态将自动进入到下一个状态，即变成“待分配接车单”的“客服已确认”的状态，添加备注弹出框如图4.19所示。



图4.19添加备注弹出框

进入“待分配接车单”后，客服人员应该对该订单进行分配泊车司机，只需在“分配司机”的弹出框中选择司机的姓名，然后点击“确定”即可完成操作，这时订单并自动进入到“在进行接车单”的“已分配泊车司机”状态，分配司机弹出框如图4.20所示。



图4.20分配司机弹出框

当司机到达机场出发层，司机需要与客户互相进行确认身份，然后完成接车的流程，并按照预定的轨迹将车辆停放到规定的停车场，上传与车辆相关的照片信息，然后交出钥匙，这时该订单的状态为“在库车辆”的“钥匙交出”状态。

当客户填写返程航班信息，并预约取车后，这时订单的状态则为“待分配送车单”的“立即取车”状态。客服人员需要与客户确认并输入航班落地时间，或者修改航班落地时间，以便提前安排送车员，航班落地时间弹出框如图4.21所示。



图4.21航班落地时间弹出框

客服人员分配了送车员后，订单的状态变为“在进行送车单”的“已分配送车司机”状态，送车员确认开始送车时，状态将自动变成“送车中”。

送车员与客户完成了送车的流程并付款后，订单的状态为“已完成订单”的 “已完成”状态。

如果客户并未付款，则订单的状态为“未付款订单”的“已送还”状态，这时需要客服人员与客户进行沟通，以便完成付款。

## 4.4 报表查询操作

报表的内容包括：机场月度报表、工作量统计表、服务效率统计表和预定订单统计，每个报表都可以查询到某个时间段的相应报表内容。

### 4.4.1 机场月度报表

机场月度报表展示的是一个月内的各个机场每天的接车单和送车单的数量，并统计出本月的总订单量，如图4.22所示。

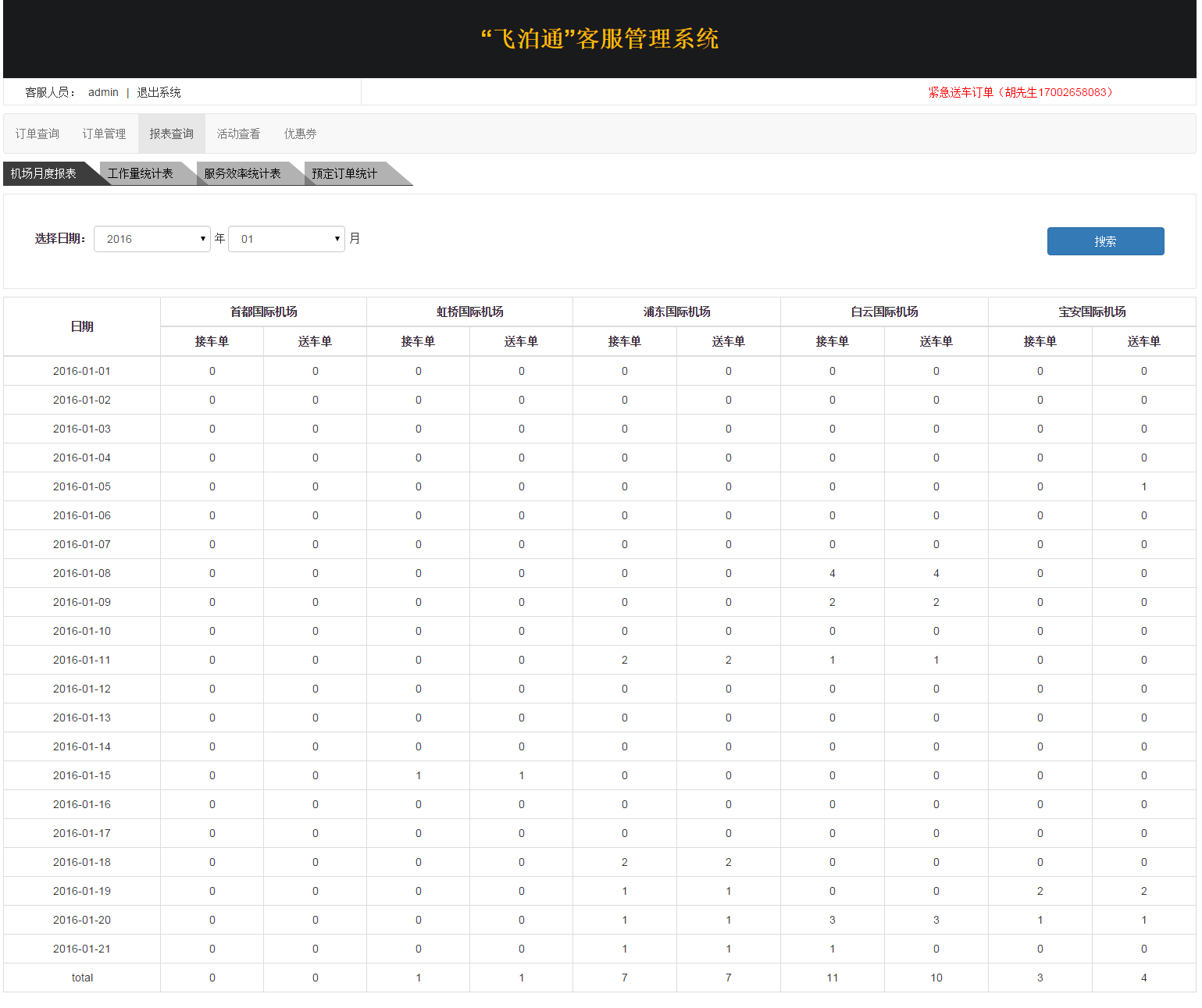


图4.22机场月度报表

### 4.4.2 工作量统计表

工作量统计表展示的是某个时间段内的不同机场的司机的接车数量和送车数量，如图4.23所示。

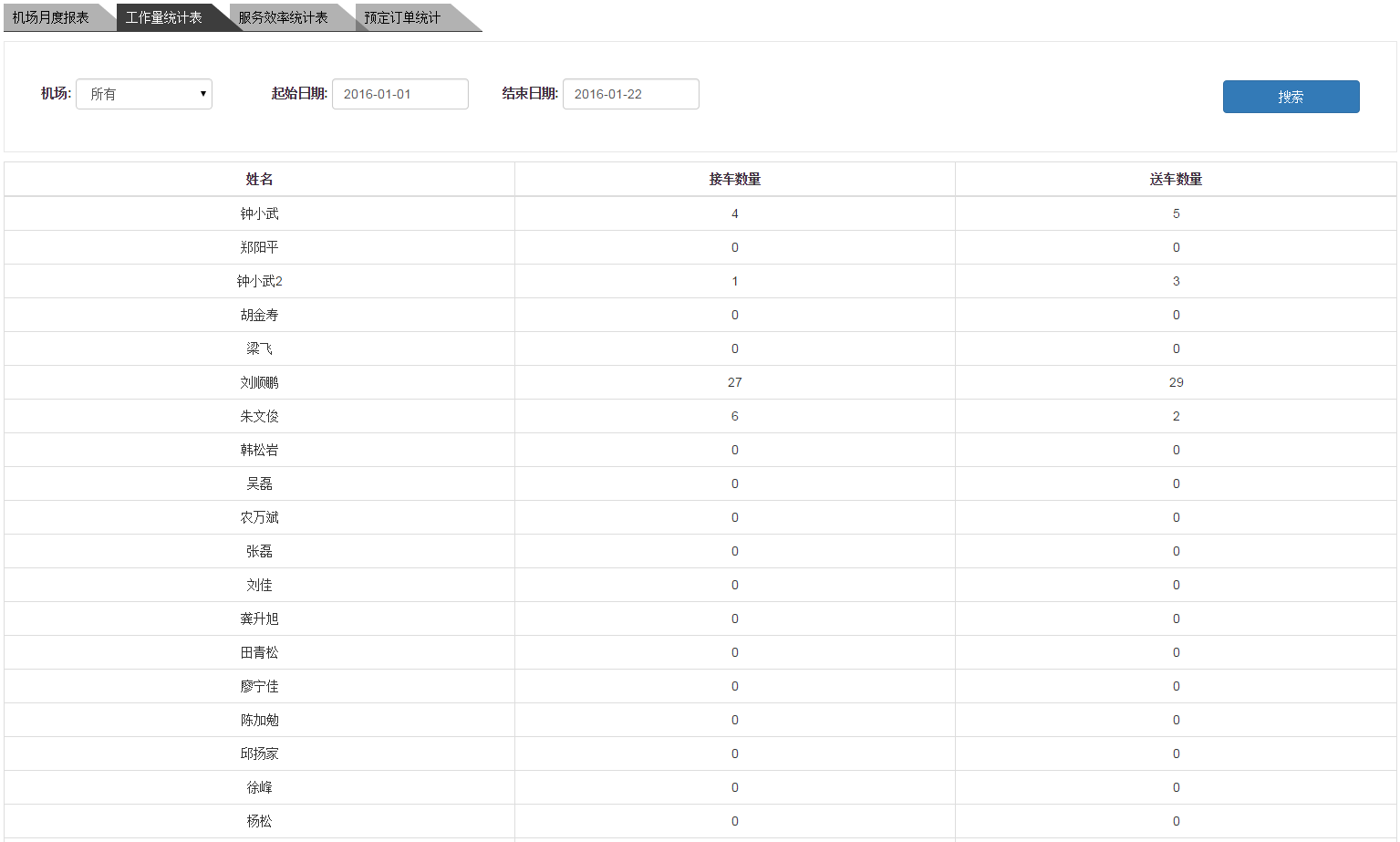


图4.23工作量统计表

### 4.4.3 服务效率统计表

服务效率统计表展示的是某个时间段内不同机场的接车数据和送车数据，如图4.24所示。



图4.24服务效率统计表

### 4.4.4 预定订单统计表

预定订单统计展示的是某个时间段内的不同机场的客户预约的接车单和送车单的数量,并统计出这个时间段的总订单量，如图4.25所示。

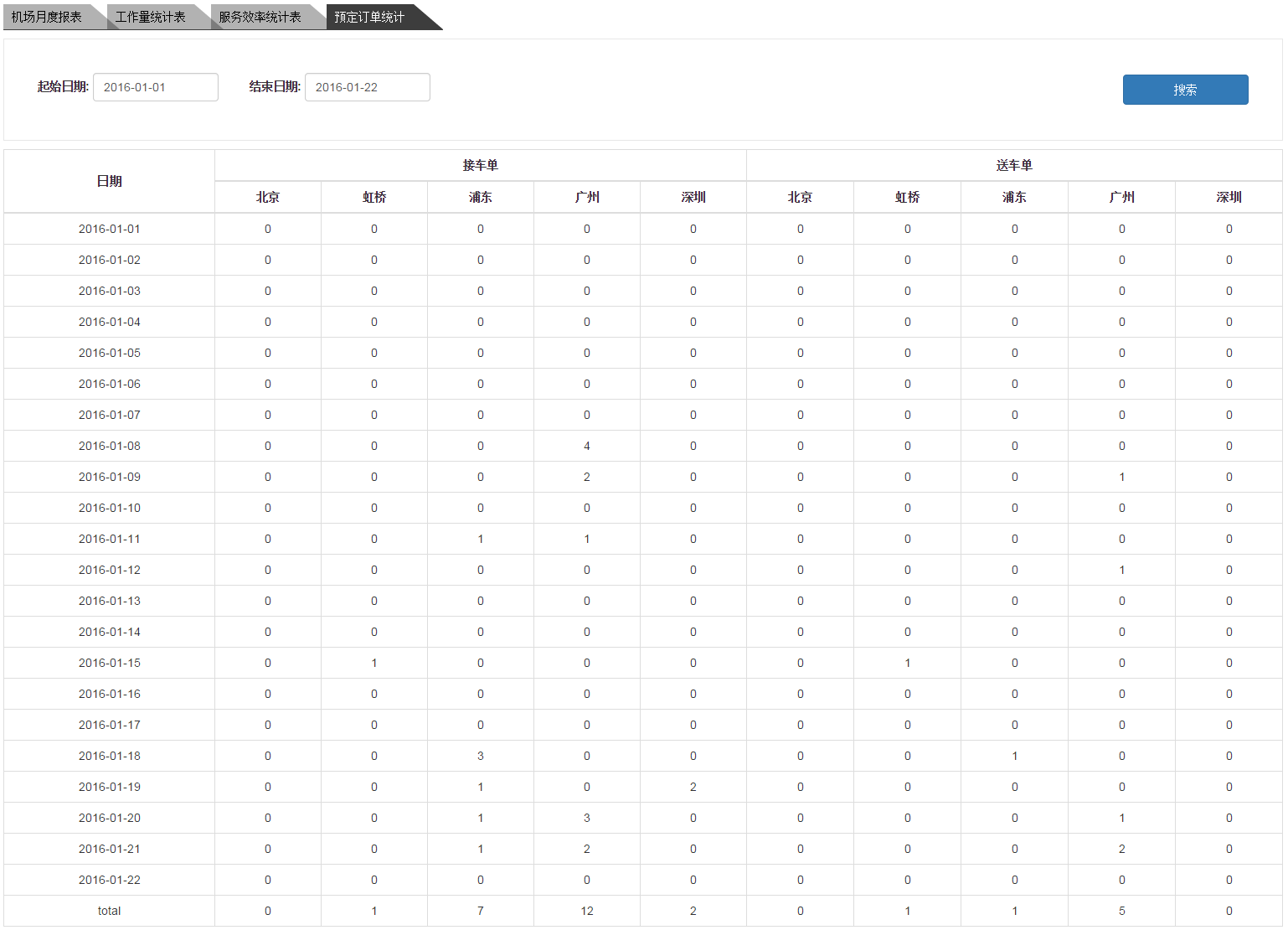


图4.25预定订单统计表

## 4.5活动查看操作

活动查看页面展示的内容为所有的在微信中进行抢停车券一系列活动，客服人员在这个页面可以看到近期在进行的活动和渠道，主要是利于客服人员和客户进行确认，以免产生出错等问题，近期的微信抢券活动信息如图4.26所示。

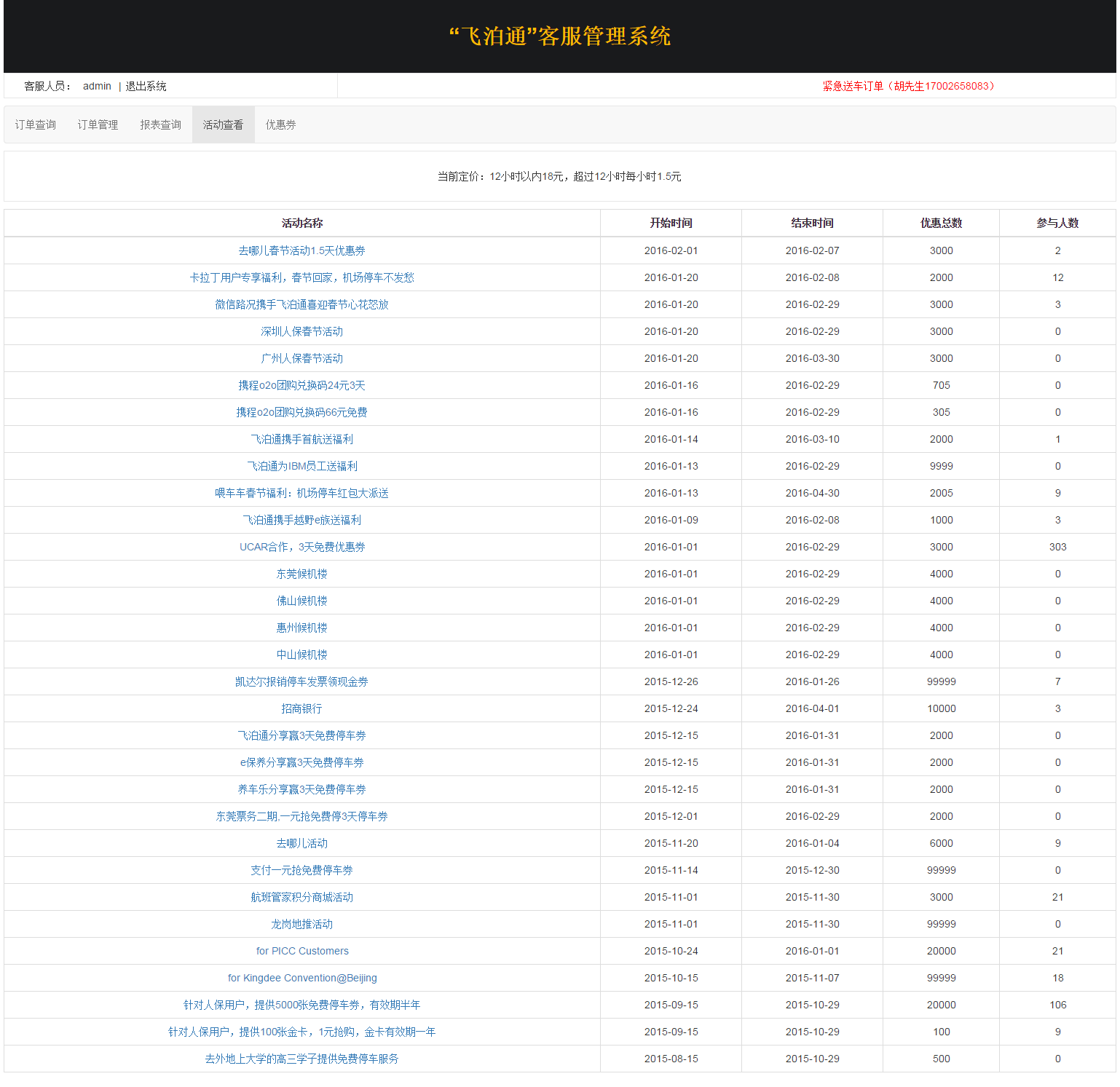


图4.26微信抢券活动信息

点击“活动名称”，可以查看到这个活动的详细内容，如图4.27所示。

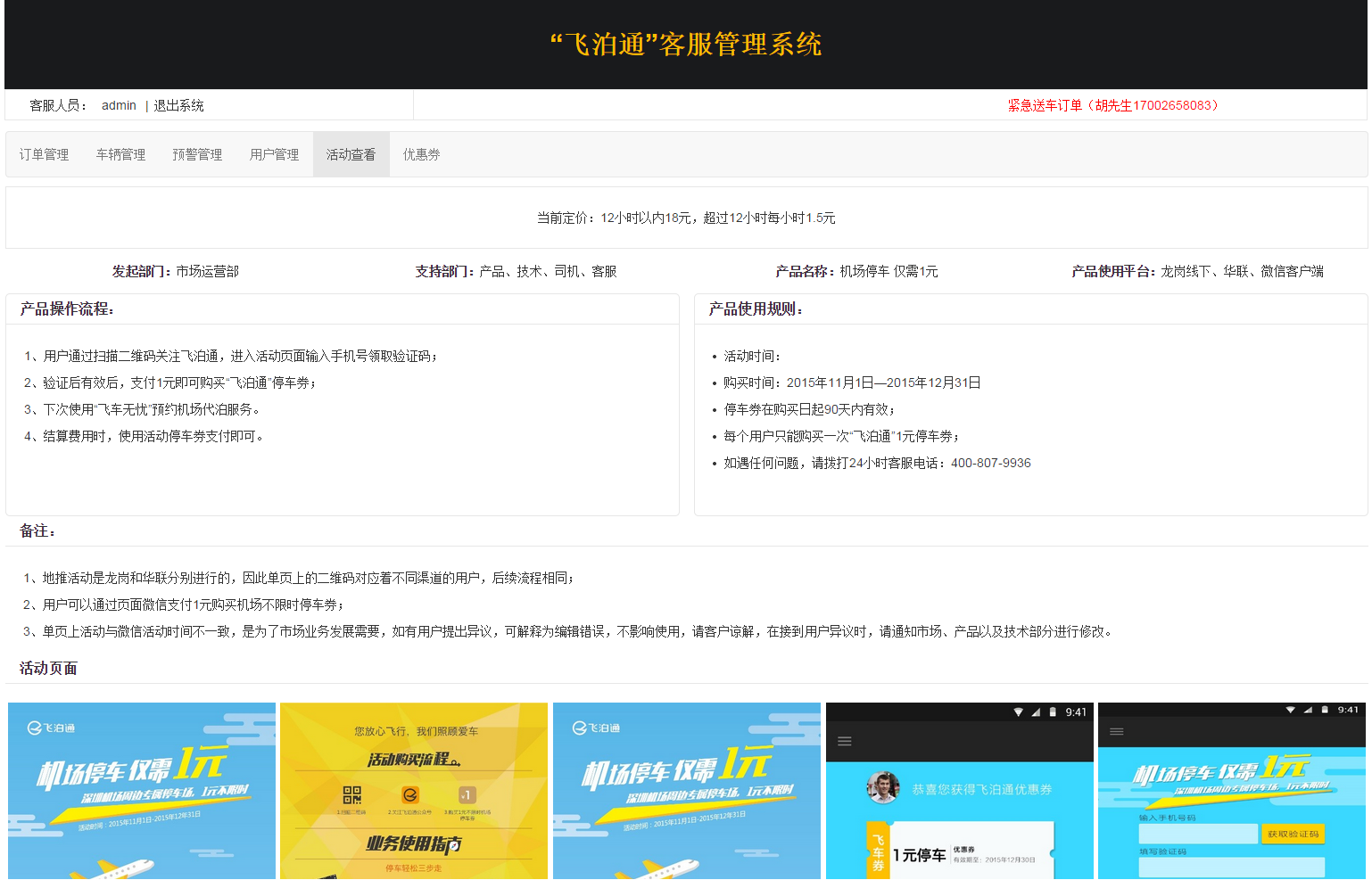


图4.27微信抢券活动详细信息

## 4.6优惠券操作

优惠券界面展示的是在微信抢券活动中成功抢到优惠券的客户信息，包括客户的手机号、优惠券类型、优惠券状态、优惠券金额、截至时间和活动来源，如图4.28所示。

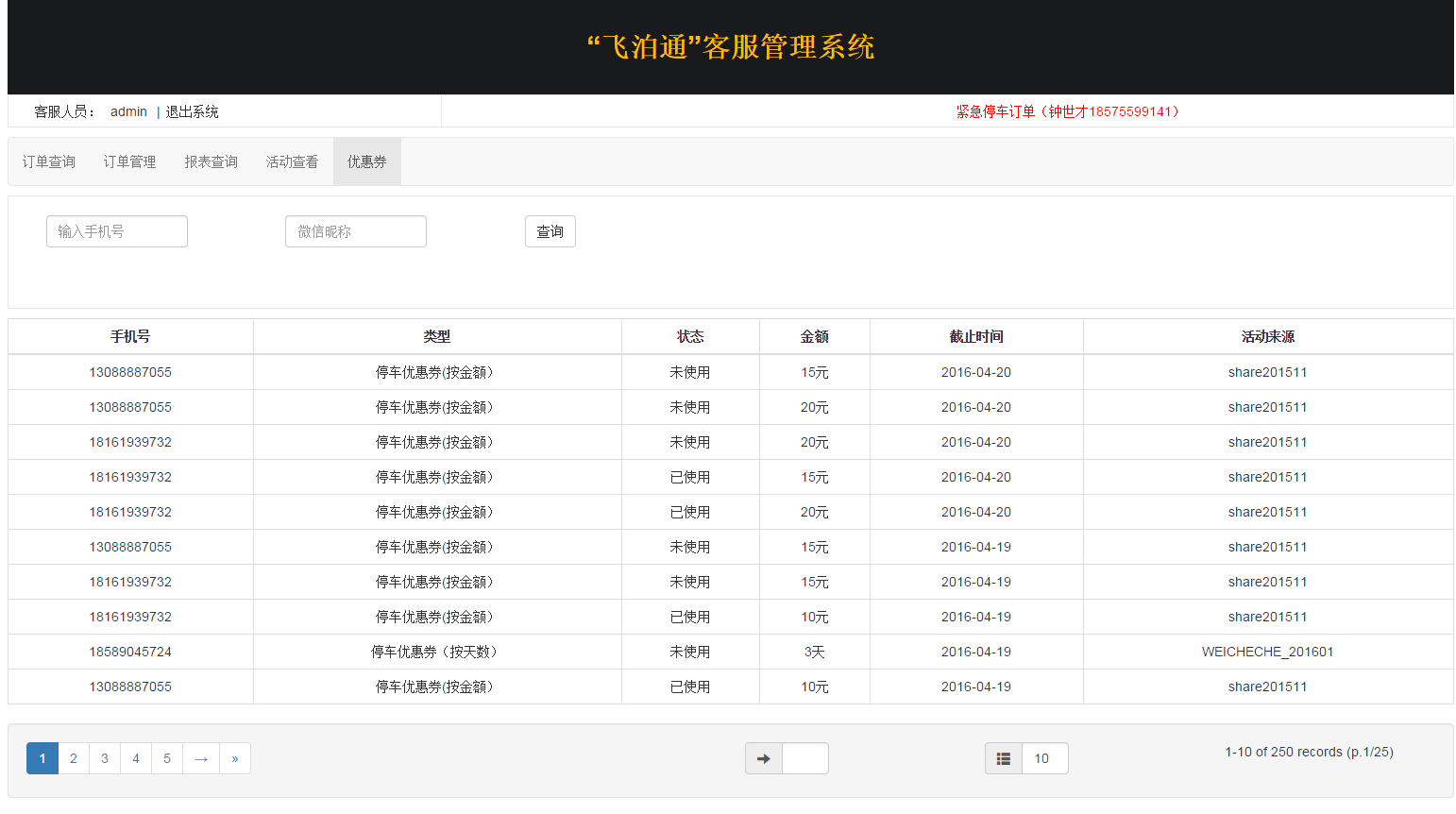


图4.28优惠券界面

客服人员可以输入客户的手机号或者微信昵称查询到该客户的优惠券信息，以手机号为“18161939732”查询的结果如图4.29所示，以微信昵称为“ifeng”查询的结果如图4.30所示。



图4.29使用手机号查询优惠券结果

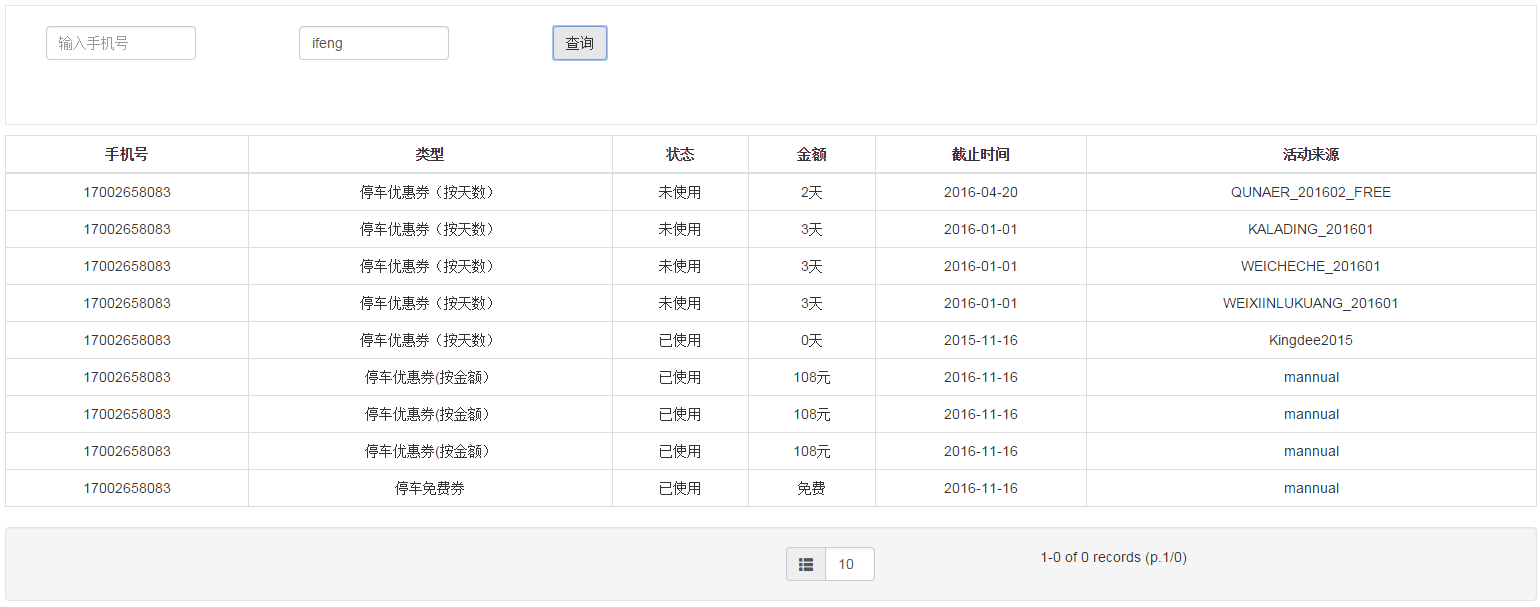


图4.30使用微信昵称查询优惠券结果

# 附录

核心代码orders.js：

template.config('escape', false);

var STATUS\_CANCELED = -1;

var STATUS\_PARKING\_NEW = 0; //预约接车单生成

var STATUS\_PARKING\_CUSTOMER\_CONFIRMED = 1; //客服已确认

var STATUS\_PARKING\_DELIVERING = 3; //调度司机中

var STATUS\_PARKING\_ASSIGNED = 5; //司机已分配

var STATUS\_PARKING\_READY = 8; //司机已就位

var STATUS\_PARKING\_PICKED = 10; //已接车

var STATUS\_PARKING\_PARKED = 15; //已泊车

var STATUS\_PARKING\_KEYOUT = 16; //钥匙交出

var STATUS\_PARKING\_FINISHED = 17; //钥匙已入柜

var STATUS\_RETURNING\_NEW = 20; //立即送车单生成

var STATUS\_RETURNING\_DELIVERING = 23; //调度司机中

var STATUS\_RETURNING\_ASSIGNED = 25; //司机已分配

var STATUS\_RETURNING\_KEYOUT = 27; //司机已就位

var STATUS\_RETURNING\_START = 30; //已出发

var STATUS\_RETURNING\_FINISHED = 35; //已交车

var STATUS\_RETURNING\_CONFIRMED = 36; //用户确认已交车

var STATUS\_RETURNING\_PAID = 38; //已支付

var STATUS\_RETURNING\_COMMENTED = 50; //已评价

var ORDER\_STATUS\_END = 50;

var status\_map = {

"-1" : "已取消",

"0" : "预约停车",

"1" : "客服已确认",

"3" : "泊车调度中",

"5" : "已分配泊车司机",

"8" : "司机已就位",

"10" : "已接车",

"15" : "已泊车",

"16" : "钥匙交出",

"17" : "钥匙已入柜",

"20" : "立即取车",

"23" : "取车调度中",

"25" : "已分配送车司机",

"27" : "送车已就位",

"30" : "送车中",

"35" : "已送还",

"36" : "用户确认收车",

"38" : "已支付",

"50" : "已完成"

}

var status\_color\_map = {

"-1" : "#CCC",//已取消

"0" : "#FF0033", //预约停车

"3" : "#FF9900",//泊车调度中

"5" : "#666633",//已分配泊车司机

"8" : "#CCCC99", //司机已就位

"10" : "#CC6666",//已接车

"15" : "#CC6600",//已泊车

"16" : "#0000FF", //钥匙交出

"17" : "#99CC66",//钥匙已入柜

"20" : "#FF0000",//立即取车

"23" : "#FF9900", //取车调度中

"25" : "#009966", //已分配送车司机

"27" : "#333",//送车已就位

"30" : "#333",//送车中

"35" : "#333",//已送还

"36" : "#333",//用户确认收车

"38" : "#333",//已支付

"50" : "#333"//已完成

}

var returningDate = function(order){

var returningdate = order.returningdate;

if(returningdate != 0){

var \_returningdate = getLocalDate2(returningdate);

}

return \_returningdate;

}

var bookingTime = function(order){

var \_bookingtime = "";

if (order.bookingtime != 0){

\_bookingtime = getLocalDate(order.bookingtime);

}

return \_bookingtime;

}

var paymentTime = function(order){

var paymenttime = order.payment.paymenttime;

if(paymenttime != 0){

var \_paymenttime = getLocalDate(paymenttime);

}

return \_paymenttime;

}

function orderUtils(){

function \_assign\_driver\_btn(order\_id, driver, assign\_type) {

var name = "立即派单";

if (driver != null) {

name = driver.realname;

}

return '<button class="btn btn-link" data-toggle="modal" data-target=".assign-driver-modal" assign\_type="' + assign\_type + '" order\_id="' + order\_id + '">' + name + '</button>';

}

var parkingdriver = function(order) {

if (order.status == -1) {

return "";

}

return \_assign\_driver\_btn(order.serialnumber, order.parking.driver, 'parking');

}

var returningdriver = function(order) {

if (order.status == -1) {

return "";

}

return \_assign\_driver\_btn(order.serialnumber, order.returning.driver, 'returning');

}

var flightLandingTime = function(order){

var landingtime = order.returning.flightlandingtime;

var \_flightlandingtime = {'value': order.returning.flightlandingtime, 'enable': ''};

if (\_flightlandingtime.value == null || 0 == \_flightlandingtime.value){

\_flightlandingtime.value = "填写";

}else if(Object.prototype.toString.call(landingtime) == '[object Number]'){

\_flightlandingtime.value = getLocalDate(landingtime);

}else{

\_flightlandingtime.value = landingtime;

}

if (order.status == -1 || order.status >= STATUS\_RETURNING\_DELIVERING){

\_flightlandingtime.enable = 'disabled="disabled"';

}

return \_flightlandingtime;

}

var renderHTML = function(order) {

var data = jQuery.extend(true, {}, order);

data.parking\_driver\_html = parkingdriver(order);

data.returning\_driver\_html = returningdriver(order);

data.\_flightlandingtime = flightLandingTime(order);

data.\_bookingtime = bookingTime(order);

data.\_returningdate = returningDate(order);

data.status\_desc = status\_map[order.status.toString()];

data.status\_color = status\_color\_map[order.status.toString()];

if (order.status >= STATUS\_PARKING\_PICKED || order.status == STATUS\_CANCELED) {

data.cancel\_enable = 'disabled="disabled"';

}else if (order.status == STATUS\_RETURNING\_COMMENTED || order.status < STATUS\_RETURNING\_NEW){

data.empty\_enable = 'disabled="disabled"';

}else {

data.cancel\_enable = '';

}

data.remark\_enable = '';

try {

if (typeof(order.remark) != 'object'){

order.remark = eval("(" + order.remark + ")");

}

} catch(error) {

order.remark = [];

　　 }

var html = template('order-item-template', data);

return html;

}

function readerOrderDetails(order) {

var data = jQuery.extend(true, {}, order);

data.\_parkingassignedtime = getLocalDate(order.parking.assignedtime);

data.\_parkingstartedtime = getLocalDate(order.parking.startedtime);

data.\_parkingfinishedtime = getLocalDate(order.parking.finishedtime);

data.\_returningassignedtime = getLocalDate(order.returning.assignedtime);

data.\_returningstartedtime = getLocalDate(order.returning.startedtime);

data.\_returningFinishedTime = getLocalDate(order.returning.finishedtime);

data.\_createtime = getLocalDate(order.createtime);

data.\_bookingtime = bookingTime(order);

data.\_paymenttime = paymentTime(order);

data.status\_desc = status\_map[order.status.toString()];

var html = template('order-detail-template', data);

return html;

}

return {

"renderHTML": renderHTML,

"readerOrderDetails" : readerOrderDetails,

}

}

function OrderManager(orders){

var manager = {};

manager.orderUtils = orderUtils();

manager.orders = orders;

manager.\_order\_id = null;

var findOrder = function (orders, order\_id){

for (var i = orders.length - 1; i >= 0; i--) {

var order = orders[i];

if (order.serialnumber == order\_id){

return order;

}

}

return null;

}

var indexOf = function(order\_id){

for (var i = manager.orders.length - 1; i >= 0; i--) {

var order = manager.orders[i];

if (order.serialnumber == order\_id){

return i;

}

}

return -1;

}

manager.renderHTML = function(){

var order\_list\_html = '';

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

var order = manager.orders[i];

var order\_html = '<tr id="order-item-' + order.serialnumber + '" status=' + order.status + '>' + manager.orderUtils.renderHTML(order) + '</tr>';

var order\_details\_html = '<tr class="toggle-orderdetails" id="order-details-' + order.serialnumber + '">' + manager.orderUtils.readerOrderDetails(order) + '</tr>';

order\_list\_html += order\_html;

order\_list\_html += order\_details\_html;

}

return order\_list\_html;

}

manager.render = function(){

var html = manager.renderHTML();

$('#order-list').html(manager.renderHTML());

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

var order = manager.orders[i];

var order\_list\_id ="order-item-" + order.serialnumber;

if($('#' + order\_list\_id).attr('status') == STATUS\_CANCELED){

$('#' + order\_list\_id).attr('class','active');

}else if($('#' + order\_list\_id).attr('status') == STATUS\_PARKING\_NEW || $('#' + order\_list\_id).attr('status') == STATUS\_RETURNING\_NEW){

$('#' + order\_list\_id).attr('class','warning');

}else if($('#' + order\_list\_id).attr('status') == STATUS\_PARKING\_PICKED || $('#' + order\_list\_id).attr('status') == STATUS\_RETURNING\_START){

$('#' + order\_list\_id).attr('class','success');

}

}

}

manager.updateOrderHTML = function(order){

var order\_html = manager.orderUtils.renderHTML(order);

var order\_details\_html = manager.orderUtils.readerOrderDetails(order);

$('#order-item-' + order.serialnumber).html(order\_html);

$("#order-details-" + order.serialnumber).html(order\_details\_html);

}

/\*给订单添加备注\*/

manager.onRemarkModal = function(event){

var jqbtn = $(event.relatedTarget);

var order\_id = jqbtn.attr("order\_id");

manager.\_order\_id = order\_id;

var order = findOrder(manager.orders, order\_id);

var status = $("#order-list>tr:first-child").attr("status");

if(parseInt(status) == STATUS\_PARKING\_NEW){

$("#btn-next").removeAttr("disabled");

}else{

$("#btn-next").attr("disabled","disabled");

}

$("#btn-remark-confirm").attr("order\_id", order\_id);

$("#btn-next").attr("order\_id", order\_id);

var remark\_html = '<tbody><tr><td>'

+ '<div>' + order.user.realname + '</div>'

+ '<div>' + order.user.phoneno + '</div>'

+ '</td>'

+ '<td>'

+ '<span>' + order.car.carno + '</span>'

+ '<span>' + order.car.color + '</span>'

+ '<span>' + order.car.brand + '</span>'

+ '</td>'

+ '</tr><tr><td>备注</td><td>'

+ '<textarea id="id-order-msg" class="form-control" rows="10"> '

+ '</textarea></td></tr>'

+ '</tbody>';

$('.user-mark').html(remark\_html);

}

manager.onRemarkCallback = function(data){

if (data.code != 0){

alert(data.msg);

return ;

}

var order\_id = manager.\_order\_id;

var order = findOrder(manager.orders, order\_id);

order.remark = data.remark;

manager.updateOrderHTML(order);

}

manager.onCancelModal = function(event){

var jqbtn = $(event.relatedTarget);

var order\_id = jqbtn.attr("order\_id");

manager.\_order\_id = order\_id;

var order = findOrder(manager.orders, order\_id);

/\*取消订单\*/

var cancel\_order\_html =

'<table class="table table-bordered">'

+' <tr>'

+ '<td><div>' + order.user.realname + '</div>'

+ '<div>' + order.user.phoneno + '</div></td>'

+ '<td>'

+ '<span>' + order.car.carno + '</span>'

+ '<span>' + order.car.color + '</span>'

+ '<span>' + order.car.brand + '</span>'

+ '</td></tr></table>';

$('#cancel\_order\_tmpl').html(cancel\_order\_html);

$("#btn-yes").attr("order\_id", order\_id);

}

manager.onCancelCallback = function(data){

if (data.code != 0){

alert(data.msg);

return ;

}

var order\_id = manager.\_order\_id;

var order = findOrder(manager.orders, order\_id);

order.status = STATUS\_CANCELED;

manager.updateOrderHTML(order);

}

manager.onAssignModal = function(order\_id, assign\_type){

var order = findOrder(manager.orders, order\_id);

manager.\_order\_id = order\_id;

manager.\_assign\_type = assign\_type;

var html = template('assign-driver-template', order);

$('.assign-driver-modal .modal-body').html(html);

domain\_ajax('/admin/api/drivers/list.js?airportid=' + order.terminal.airportid, {}, 'show\_driver\_list');

}

manager.onAssignCallback = function(data){

if (data.code != 0 ){

alert(data.msg);

return;

}

var order\_id = manager.\_order\_id;

var order = findOrder(manager.orders, order\_id);

order[manager.\_assign\_type].driver = data.driver;

manager.updateOrderHTML(order);

}

manager.onDriverAssignment = function(driver\_id){

var order\_id = manager.\_order\_id;

var assign\_type = manager.\_assign\_type;

var url = '/admin/api/orders/assign\_' + assign\_type + '\_driver.js';

var data = {'order\_id': order\_id, 'assign\_type': assign\_type, 'driver\_id': driver\_id};

domain\_ajax(url, data, 'onAssignCallback');

}

manager.onClearReturningInfoModal = function(order\_id){

manager.\_order\_id = order\_id;

var order = findOrder(manager.orders, order\_id);

order.\_returningdate = returningDate(order);

order.\_order\_id = order\_id;

var html = template('clear-returning-template', order);

$('.empty-info-modal .modal-body').html(html);

}

manager.onClearReturningInfo = function(order\_id){

domain\_ajax('/admin/api/orders/clear\_returning\_info.js', {'order\_id': order\_id}, 'onClearReturningInfoCallback');

}

manager.onClearReturningInfoCallback = function (data) {

var order\_id = manager.\_order\_id;

if (data.code != 0){

alert('操作失败，当前无法置空返程信息！');

return ;

}

var index = indexOf(order\_id);

manager.orders[index] = data.book;

manager.render();

}

manager.onFlightLandingTimeModal = function(order\_id){

manager.\_order\_id = order\_id;

var order = findOrder(manager.orders, order\_id);

var html = template('flightlandingtime-template', order);

$('#flightlandingtime').html(html);

}

manager.onFlightLandingTime = function(flightlandingtime){

var order\_id = manager.\_order\_id;

domain\_ajax('/admin/api/orders/set\_flight\_landing\_time.js', {'order\_id': order\_id, 'flightlandingtime': flightlandingtime},

'onFlightLandingTimeCallback');

}

manager.onFlightLandingTimeCallback = function(data){

if(data.code != 0){

alert(data.msg);

}

var order\_id = data.order\_id;

var order = findOrder(manager.orders, order\_id);

order.returning.flightlandingtime = data.flightlandingtime;

manager.updateOrderHTML(order);

}

return manager;

}

function show\_driver\_list(data){

var online\_drivers = [];

var offline\_drivers = [];

for (var i = data.drivers.length - 1; i >= 0; i--) {

var driver = data.drivers[i];

if (driver.onlinestatus == 1){

online\_drivers.push(driver);

}else{

offline\_drivers.push(driver);

}

}

var html = template('driver-list-template', {'online\_drivers': online\_drivers, 'offline\_drivers': offline\_drivers});

$('#driver-list-wrapper').html(html);

$('.select-driver').change(function(){

var driver\_id = $(this).children('option:selected').val();

var driver\_name = $(this).children('option:selected').html();

$('#current-selected-driver').html(driver\_name);

$('#current-selected-driver').attr('driver\_id', driver\_id);

});

}

var orderManager = OrderManager({});

function search\_orders(page, pagesize){

var params = {page: page, pagesize: pagesize};

if ($('input[name=phoneno]').length > 0){

params.phoneno = $('input[name=phoneno]').val();

}

if ($('input[name=nickname]').length > 0) {

params.nickname = $('input[name=nickname]').val();

}

if ($('input[name=carno]').length > 0){

params.carno = $('input[name=carno]').val();

}

if ($('select[name=airportid]').length > 0){

params.airportid = $('select[name=airportid]').val();

}

if ($('select[name=ordertype]').length > 0){

params.ordertype = $('select[name=ordertype]').val();

}

if ($('select[name=orderstatus]').length > 0){

params.orderstatus = $('select[name=orderstatus]').val();

}

if ($(".tabs .current").length > 0){

params.orderstatus = $(".tabs .current").attr("orderstatus");

}

if ($('input[name=bookingtime]').length > 0){

params.bookingtime = $('input[name=bookingtime]').val();

}

if ($('input[name=returningdate]').length > 0){

params.returningdate = $('input[name=returningdate]').val();

}

if ($('input[name=returningfinishedtime]').length > 0){

params.returningfinishedtime = $('input[name=returningfinishedtime]').val();

}

if ($('input[name=parkingstartedtime]').length > 0){

params.parkingstartedtime = $('input[name=parkingstartedtime]').val();

}

if ($('input[name=canceled]').length > 0){

params.canceled = $('#canceled-checkbox').prop('checked');

}

load\_orders(params);

}

function onClearReturningInfoCallback (data) {

orderManager.onClearReturningInfoCallback(data);

}

function onAssignCallback(data){

console.log(data)

orderManager.onAssignCallback(data);

}

function onRemarkCallback(data){

orderManager.onRemarkCallback(data);

}

function onCancelCallback(data){

orderManager.onCancelCallback(data);

}

function onFlightLandingTimeCallback(data){

orderManager.onFlightLandingTimeCallback(data);

}

function \_onOrdersLoaded(data, onPageChange){

$("#pager").bs\_pagination({

currentPage: data.page,

rowsPerPage: data.pagesize,

maxRowsPerPage: 10,

totalPages: data.pagecount,

totalRows: data.totalcount,

visiblePageLinks: 5,

showGoToPage: true,

showRowsPerPage: true,

showRowsInfo: true,

showRowsDefaultInfo: true,

onChangePage: function(event, data){

var page = data.currentPage;

var pagesize = data.rowsPerPage;

search\_orders(page, pagesize);

}

});

orderManager.orders = data.orders;

orderManager.render();

$(".orderid").click(function() {

var toggleNode = $(this).parent().parent().next(".toggle-orderdetails");

toggleNode.slideToggle("fast");

});

$(".remark\_modal").on("show.bs.modal", function(event) {

orderManager.onRemarkModal(event);

});

$(".cancel-order-modal").on("show.bs.modal", function(event) {

orderManager.onCancelModal(event);

});

/\*置空返程信息\*/

$(".empty-info-modal").on("show.bs.modal", function(event) {

var jqbtn = $(event.relatedTarget);

var order\_id = jqbtn.attr("order\_id");

orderManager.onClearReturningInfoModal(order\_id);

$("#btn-empty").attr("order\_id", order\_id);

});

/\*分配司机\*/

$(".assign-driver-modal").on("show.bs.modal", function(event){

var jqbtn = $(event.relatedTarget);

var order\_id = jqbtn.attr("order\_id");

var assign\_type = jqbtn.attr('assign\_type');

orderManager.onAssignModal(order\_id, assign\_type);

});

$('.flight-landing-time-modal').on('show.bs.modal', function(event){

var jqbtn = $(event.relatedTarget);

var order\_id = jqbtn.attr("order\_id");

orderManager.onFlightLandingTimeModal(order\_id);

});

/\*鼠标经过显示详细备注信息\*/

$(".remarks").popover({placement:'left'});

/\*车辆停放信息\*/

$('.venobox').venobox({

numeratio: true,

border: '5px',

height: 'auto'

});

}

function onOrdersLoaded(data){

function onPageChange(event, data) {

// your code here e.g.

var page = data.currentPage;

var pagesize = data.rowsPerPage;

search\_orders(page, pagesize);

}

\_onOrdersLoaded(data, onPageChange);

}

var checking\_time = 30 \* 1000;

function onPendingOrderCallback(data){

//console.log("待处理订单：");

console.log(data);

var showing = false;

if(data.parkings.length > 0){

showing = true;

for (var i = data.parkings.length - 1; i >= 0; i--) {

var parking = data.parkings[i];

parking.\_bookingtime = getLocalDate(parking.bookingtime);

};

}

if(data.returnings.length > 0){

showing = true;

for (var i = data.returnings.length - 1; i >= 0; i--) {

var returning = data.returnings[i];

returning.\_returningdate = getLocalDate(returning.returningdate);

var flightlandingtime = returning.returning.flightlandingtime;

if(flightlandingtime != 0){

returning.\_flightlandingtime = getLocalDate(flightlandingtime);

}

};

}

if (showing){

checking\_time = 2 \* 60 \* 1000;

var pending\_order\_html = template('pending-order-template', data);

$('.pending-order-modal').modal('show');

$('#pending\_order\_tmpl').html(pending\_order\_html);

}else{

checking\_time = 30 \* 1000;

}

setTimeout('refresh\_pendingorder();', checking\_time);

}

function refresh\_pendingorder() {

domain\_ajax('/admin/api/orders/check\_new.js', {}, 'onPendingOrderCallback');

}

/\*紧急订单\*/

$(function(){

load\_template('#template-container', "/csbms/templates/orders.tpl");

load\_template('#template-warning', "/csbms/templates/warning.tpl");

refresh\_pendingorder();

setTimeout('refresh\_pendingorder();', checking\_time);

$('#btn-assign-driver').click(function(){

$(".assign-driver-modal").modal('toggle');

var driver\_id = $('#current-selected-driver').attr('driver\_id');

orderManager.onDriverAssignment(driver\_id);

});

$('#search-form').submit(function(){

search\_orders(1, 10);

return false;

});

$("#btn-remark-confirm").on('click', function(event) {

$(".remark\_modal").modal('toggle');

var order\_id = $(this).attr("order\_id");

var remark = $('#id-order-msg').val();

load\_remarks({order\_id: order\_id, remark:remark});

});

$("#btn-next").on('click', function(event) {

$(".remark\_modal").modal('toggle');

var order\_id = $(this).attr("order\_id");

var remark = $('#id-order-msg').val();

var next = true;

load\_remarks({order\_id: order\_id, remark:remark, next: next});

});

$("#btn-yes").on('click', function(event) {

$(".cancel-order-modal").modal('toggle');

var order\_id = $(this).attr("order\_id");

domain\_ajax('/admin/api/orders/cancel.js', {order\_id: order\_id}, 'onCancelCallback');

});

$("#btn-empty").on('click', function(event) {

$(".empty-info-modal").modal('toggle');

var order\_id = $(this).attr("order\_id");

orderManager.onClearReturningInfo(order\_id);

});

$('#btn-landing-time').on('click', function(event){

var flightlandingtime = $('input[name=flightlandingtime]').val();

orderManager.onFlightLandingTime(flightlandingtime);

$(".flight-landing-time-modal").modal('toggle');

});

});

function load\_remarks(params){

domain\_ajax('/admin/api/orders/remark.js', params, 'onRemarkCallback');

}

function load\_orders(params){

$("#order-list").html(loading\_html);

domain\_ajax('/admin/api/orders/list.js', params, 'onOrdersLoaded');

}

function update\_html(summary){

$('.tabs a').each(function(){

var orderstatus = $(this).attr('orderstatus');

console.log("set orderstatus: " + orderstatus);

$(this).find('label').html(summary[orderstatus]);

});

}

function onOrdersSummary(data){

var summary = {

'parking\_new': 0,

'delivering': 0,

'parking\_going': 0,

'parking\_parked': 0,

'returning\_new': 0,

'returning\_going': 0,

'returning\_finished': 0,

'not\_paid': 0

}

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

var scount = data.summary[i];

var status = scount[0];

console.log('status:'+ status);

var count = scount[1];

/\*STATUS\_PARKING\_NEW (0)\*/

if (status == STATUS\_PARKING\_NEW){

summary['parking\_new'] += count;

}

else if (status >= STATUS\_PARKING\_CUSTOMER\_CONFIRMED && status < STATUS\_PARKING\_ASSIGNED){

summary['delivering'] += count;

}

else if (status >= STATUS\_PARKING\_ASSIGNED && status < STATUS\_PARKING\_KEYOUT){

summary['parking\_going'] += count;

}

else if(status >= STATUS\_PARKING\_KEYOUT && status < STATUS\_RETURNING\_NEW){

summary['parking\_parked'] += count;

}

else if (status > STATUS\_PARKING\_KEYOUT && status < STATUS\_RETURNING\_ASSIGNED){

summary['returning\_new'] += count;

}

else if(status >= STATUS\_RETURNING\_ASSIGNED && status < STATUS\_RETURNING\_FINISHED){

summary['returning\_going'] += count;

}

if (status >= STATUS\_RETURNING\_FINISHED && status <= ORDER\_STATUS\_END){

summary['returning\_finished'] += count;

}

if(status >= STATUS\_RETURNING\_FINISHED && status < STATUS\_RETURNING\_PAID ){

summary['not\_paid'] += count;

}

}

summary.parking\_parked = data.parking\_parked;

summary.returning\_new = data.returning\_new;

update\_html(summary);

}

function load\_summary(params){

domain\_ajax('/admin/api/orders/summary.js', params, 'onOrdersSummary');

}