

**毕 业 论 文（设 计）**

题目：房地产运营支撑系统

(英文)：Real estate operation support system

院别：电子与信息学院

专业：电子信息工程

姓名：梁锦银

学号：2014044243013

指导教师：王春安

日期：2018年5月

# 房地产运营支撑系统

**摘要**

如今公司如雨后春笋搬地涌现，每一家公司有具有着自己的独特，独特的运营模式，还有各种机构等等。随着房地产经纪行业网络信息化的发展，网上找房已经成为房产成交前的必不可少的环节，房产经纪公司也在加大在搜房平台上对房源信息的网络投放。但是由于网络的特殊性，房产经纪公司并不知道网络投放给成交带来多少效果，也很难判断该如何改进投放策略和效果[2]。公司内部的人员，特别是高层，他们需要一个系统来管理，和查看公司的运营情况。本文将结合现在的房地产经营情况开发一套房地产运营支撑系统。系统运用的Java语言编写，采用SSM（SpringMVC+spring+mybatis）三层架构框架进行开发,前端使用JQuery+CSS+JSP+bootstrap技术对视图渲染，后台使用mysql对数据进行处理和存储，运用shiro进行功能权限的控制。系统实现了客户管理，案场管理，会议纪要，系统基本模块；可以清晰快捷的查询案场销售情况和销售人的具体销售。系统简单易懂，运行稳定，适用于中小型房地产公司。□□

**关键词：**springMVC；Spring；mybatis；shiro；MySQL等技术

Real estate operation support system

Abstract

Now companies are springing up, and each company has its own unique, unique operating model, a variety of institutions and so on. The people inside the company, especially at the top, need a system to manage it. This paper will develop a set of real estate operation support system combined with the current real estate management situation. The system will be written in Java language and developed with a three-tier framework of SSM(SpringMVC spring mybatis. The front end uses JQuery CSS JSP bootstrap technology to render the view, the background uses mysql to process and store the data, uses shiro to carry on the function authority control. The system has realized the customer management, the case field management, the meeting minutes, the system basic module; Can be clear and quick inquiry case field sales and specific sales of the salesperson. The system is easy to understand, stable operation, suitable for small and medium-sized real estate companies. --

Key words: springMVC；Spring；mybatis；shiro

# 1绪论

## 1.1研究背景与意义

在当今的数据社会中，运营支撑系统是企业持续发展的关键性因素。旧时代的那种文档式追踪项目已经不适合现代人的追求了，随着数据信息的发展，大数据的运用，如今的数据模式已变为数据可视化才能满足现在人们对对于房地产行业来说，采用先进的管理运营模式非常紧迫，运营支撑系统重在公司的运营管理，运用数据可靠的手段对公司项目的进度管理和监督。根据客户的价值对客户进行管理，通过跟踪客户需求，提升客户的满意度和服务体验度，以谋求企业的持续发展和利益的最大化。

## 1.2研究现状

随着互联网的发展,传统的数据处理技术已经不能满足人们的需求。在计算机以及有线、无线网络飞速发展的今天,一方面人们对于信息能够及时传输、方便携带等方面的要求越来越突出；另一方面对于数字化进程中的人类社会来说,日益突出的海量数据存储问题逐渐被人们所意识到[3]。如今的中小型房地产企业对于业务的管理都是十分离散的，部门之间的项目都是各自记录、管理、和跟进。没有一个汇总的运营，同类项目之间也没有对比，相互激进。为了帮助房地产企业降低建设成本,增加资金的把控能力,及时获取相关条例执行和项目的进展情况,从而设计出房地产动态成本管理模型,同时开发与之配套的系统,即为房地产动态成本管理系统[11]。之前的房地产的管理侧重点在于销售模块，而忽视了公司的运营方面。公司的高层也无法总览全局，根据公司的运营情况做出进一步有效的指示。这无疑是公司运营效率低的体现。还有客户资料的浪费，在销售的过程中会手动登记有很多的客户信息，楼盘销售结束之后大部分的数据都会流失。而房地产支撑系统是管理和记录案场的销售进度，客户的信息、购房意愿度。公司高层的决策者可以监控所有的案场的各个环节，及时了解各个环节的实际情况，可根据销售进度做出相对的销售决策，亦可以相比较不同的案场，对效率低的案场进行调整。随着竞争环境的变化，企业的运营管理作为一种经营战略越来越受到房地产行业的重视。

# 2系统框架技术

本系统主要是由SSM（springMVC+spring+mybatis）+shiro框架构建整合框架，并整合前端框架bootstrap实现视图渲染的web系统。在Web应用的开发过程中,开发框架的选择非常重要。一个好的开发框架能够加速Web应用的开发速度,降低开发成本,减少开发人员的工作量[1]。

## 2.1 springMVC框架



图2.1SpringMVC架构图

SpringMVC其中的MVC （Model-View-Control）模型-视图-控制；其性质的C 层要完成的主要工作：封装web 请求为一个数据对象、调用业务逻辑层来处理数据对象、返回处理数据结果及相应的视图给用户。Spring C 层框架的核心是 DispatcherServlet，它的作用是将请求分发给不同的后端处理器，也即使用了一种被称为Front Controller 的模式（后面对此模式有简要说明）。 Spring 的C 层框架使用了后端控制器来、映射处理器和视图解析器来共同完成C 层框架的主要工作。并且spring 的C 层框架还真正地把业务层处理的数据结果和相应的视图拼成一个对象，即ModelAndView 对象。

## 2.2 spring框架

## Spring框架一个开源的轻量级框架，它不仅可以解决许多J2EE开发中一些常见的问题，同时也可替代EJB技术，Spring的产生很大部分是企业开发应用的复杂性促使的。Spring提供了组件核心底层机制的一些运作情况，例如组件的创立、获得、结束、资源管理、状态管理等生命周期的相关支持功能，是一个组件技术框架。Spring作为一种面向切面（AOP）的容器以及轻量级的控制反转（IOC）框架，可以很方便快捷地实现简单的组合或者一些复杂的应用以及组件的配置等，对于开发人员来说，这可以帮助他们编写出方便管理并且易于测试的代码。

## 2.3 mybatis框架

MyBatis 本是[apache](http://baike.baidu.com/view/28283.htm)的一个开源项目[iBatis](http://baike.baidu.com/view/628102.htm), 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis，实质上Mybatis对ibatis进行一些改进。 目前mybatis在github上托管。git（分布式版本控制，当前比较流程）MyBatis是一个优秀的持久层框架，它对jdbc的操作数据库的过程进行封装，使开发者只需要关注 SQL 本身，而不需要花费精力去处理例如注册驱动、创建connection、创建statement、手动设置参数、结果集检索等jdbc繁杂的过程代码。Mybatis通过xml或注解的方式将要执行的各种statement（statement、preparedStatemnt、CallableStatement）配置起来，并通过java对象和statement中的sql进行映射生成最终执行的sql语句，最后由mybatis框架执行sql并将结果映射成java对象并返回。本系统通过Spring与MyBatis 的集成，探讨如何有效地构建符合数据持久层的系统架构，实现持久层之间的松散耦合，使整个系统具备了良好的稳定性、可伸缩性和可扩展性[9]。

## 2.4 shiro框架

权限管理系统是现代软件项目中必不可少的子系统,用户通常要求权限管理系统是细粒度和松耦合,尽量减少在权限系统开发中花费更多的精力[8]。apache shiro 是一个安全认证框架，和 spring security 相比，在于他使用了比较简洁易懂的认证和授权方式。其提供的 native-session（即把用户认证后的授权信息保存在其自身提供Session 中）机制，这样就可以和 HttpSession、EJB Session Bean 的基于容器的 Session脱耦，到和客户端应用、Flex 应用、远程方法调用等都可以使用它来配置权限认证。 在exit-web-framework 里的 vcs-admin 例子用到该框架，具体使用说明可以参考官方帮助文档。在这里主要讲解如何与 spring 结合、动态创建 filterchaindefinitions、以及认证、授权、和缓存处理。

## 2.5 SSM整合

SSM（Spring+SpringMVC+MyBatis）框架集由Spring、SpringMVC、MyBatis三个开源框架整合而成，常作为数据源较简单的web项目的框架。

其中spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。

SpringMVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。

MyBatis是一个支持普通SQL查询，存储过程和高级映射的优秀持久层框架。

# 3系统需求分析与设计

## 3.1 系统需求

系统实现应该具备以下功能：

一、客户管理：客户管理包含客户信息、合作伙伴。

1、 客户信息，客户信息模块应关联楼盘的目标户型，楼盘的改变会改 变客户的目标户型，还要关联置业顾问；客户信息基本的增删改查 功能；客户信息的导入、导出（Excel）功能。

2、合作伙伴，应具备查看合作伙伴、添加合作伙伴、修改合作伙伴、删 除合作伙伴功能。

二、商机管理

三、案场管理：案场管理包含案场看板、案场信息、房型信息、销售看板。

1、案场信息，应具备基本增删改查功能，并关联系统内部人员。

2、案场看板，具有查看本月楼盘销售量，剩余数量，销售额。

3、房型信息，具备基本增删改查功能，并关联上案场维度。

4、销售看板，应有查看置业顾问的月销售额，签单数，完成指标率。

四、我的面板：包含个人信息、个人来往信箱、会议纪要。

1、个人信息，可以查看自己的电话、邮箱、职位、所属部门等基本信息。

2、信箱，具有同公司人员发送邮箱功能，可以发送工作汇报、请假事宜。

3、会议纪要，可以记录会议主题，会议内容，解决的问题。应具有在会 议纪要中发送邮箱到公司其他人员。

五、系统管理：应包含用户管理、机构管理、角色管理、字典管理

1、用户管理，应具有增删改查功能，记录用户的基本信息。

2、机构管理，应拥有增删改查功能，用树状图展示机构的层级关系。

3、角色管理，应具有增删改查功能，具有分配权限和分配用户功能。

六、系统监控：首页（mysql驱动信息）、SQL监控

1、首页应展示mysql驱动信息。

2、记录系统运作中的sql运行次数、返回的结果等信息。

七、数据权限和功能权限

1、数据权限，系统用户应具有职级，职级不同看到的数据不同。职级越 高查看的数据就越多。

2、功能权限，职级的等级越高界面的功能就越多。

八、负载均衡

1、系统使用Nginx代理，实现负载均衡处理高并发，并发量可达3000 访问量。需在虚拟机用Nginx代理两个不同端口的服务器（Tomcat）。

## 3.2各功能需求分析

### 3.2.1用户权限控制

用户可分为三种，普通置业顾问、案场负责人（项目负责人）、总主管（admin）。

普通的置业顾问只能操作所属自己的客户数据，案场负责人可以操作所属案场的所有数据，admin可以操作所有数据。一个用户可以具有多角色，角色与用户之间的关系是多对多。权限的控制包括数据权限和功能权限，根据角色的不同查看的数据数目不同，职级越高查看的数据就越多，职级的等级越高界面的功能就越多。权限可由admin管理员配置。

### 3.2.2客户管理模块

客户管理分为客户信息和合作伙伴两部分，其中客户信息具有增删改查功能、导入导出功能。如图3.2所示

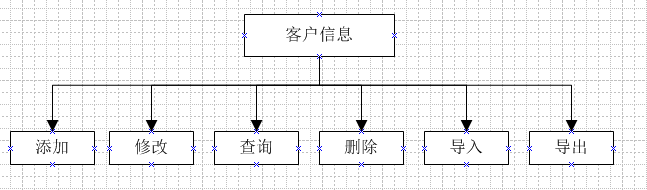


图3.2客户管理

客户信息

1、添加新客户

添加客户个人信息，客户姓名、联系电话、客户状态、购房用途、认知途径、关联案场、意向户型、案场负责人。

2、修改客户

修改客户信息。客户的有三种状态：潜在客户、正在跟进客户、已交易客户。潜在客户：经过大数据的匹配的有意图卖房的客户；正在跟进客户：来案场咨询或是来电咨询客户；已交易客户：已进行金钱交易，有合同保障客户。一旦客户的状态改变为已交易客户，该客户的置业顾问的销售额会添加对应客户选中的案场房型的金额值，案场的房型数量也会减少1，本季度的案场销售额亦添加相应的金额。客户信息中包含的楼盘属性与房型和案场联系人有关联，若是楼盘的变更，其目标户型和案场负责人也会随之变化。

3、查询客户信息

具有客户名称检索、客户状态查询。

4、删除客户

删除客户信息。

5、导入导出

客户信息可通过Excel文件导入导出。

合作伙伴具有增删改查功能。如图3.3所示

只有admin管理员才能对其进行增删改功能，其他用户只能查看。

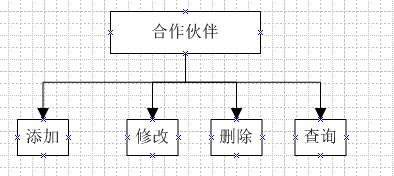


图3.3合作伙伴

### 3.2.3案场管理

案场管理包含案场看板、案场信息、房型信息、销售看板四个节点。

结构图如图3.4所示

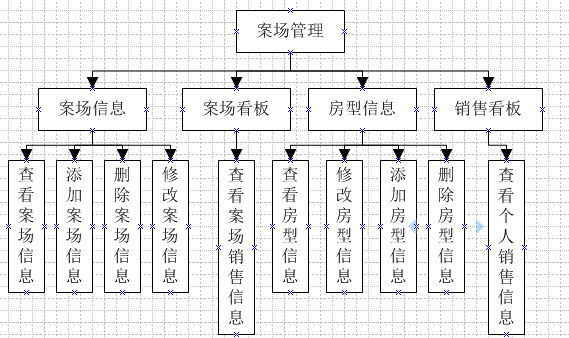


图3.4案场管理

案场信息

1、添加案场信息

添加案场信息，其中维度有案场名称、案场地址、所属集团、案场负责人、负责人电话、规定月销售额、开盘时间、结束时间、关联团队成员、备注信息。

其中案场负责人、团队成员关联用户，案场信息一般由案场负责人添加。

2、查询案场信息

查询案场信息有三种检索方式，案场负责人、开盘时间、所属集团的搜索。其中案场负责人和所属集团运用的是模糊查询。

3、删除案场信息

根据案场id删除案场信息。

4、修改案场信息

案场信息的修改由案场负责人修改。

案场看板

1、案场看板包含了案场名称、已售楼房数量、剩余楼房数量、本月销售额、本月出售楼房数量。看板的统计来自于客户的成交量。当客户信息中的客户状态为已交易状态，那么客户对应的目标户型关联的楼盘（案场）的已售楼房数量会加1，从而剩余楼房数量会减1，本月销售额会增加目标户型对应的价格数目，本月出售数量也是会加1。

房型信息

1、添加房型信息

添加房型信息，其中维度有房子数目、房总价格、房型、面积、剩余数量、所属楼盘、备注。剩余数量=房数目-已售数目。所属楼盘关联案场，房型信息一般由案场负责人添加。

2、查询房型信息

查询房型有两种索方式，房子总价格、套房面积，都是运用模糊查询。

3、删除房型信息

根据案场id删除房型信息。

4、修改房型信息

房型信息的修改由案场负责人修改，套房的数目的改变会改变案场套房总数目。

销售信息

1、销售看板包含了销售负责人（置业顾问）、楼房的销售指标、个人的签单数、本月的销售额、完成率。销售指标是由案场负责人定义的，每个案场的销售指标可能不同；个人签单数就是销售套房的数量；完成率就是本月有没有达到指标数，完成率的公式：完成率=指标数\*100%。销售额前三名的职业顾问会作为公司的榜样在首页展示。

### 3.2.4我的面板

我的面板包括三个节点，个人信息、邮箱、会议纪要。结构图如图3.5所示

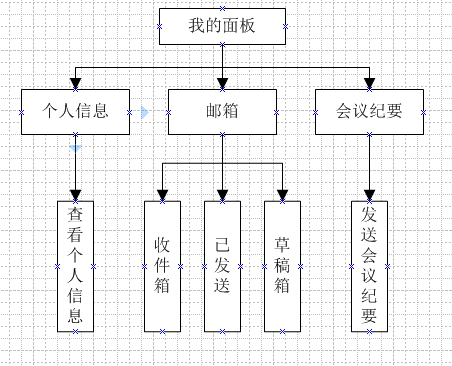


图3.5我的面板

个人信息

1、个人信息，可以查看个人的手机号、邮箱、所属部门、公司、所属角色等。

邮箱

1、收件箱

接收别人发送过来的邮件。邮件有增删查功能，可以对邮件的正文或是标题进行模糊查询；删除可以批量删除或是删除单条数据；发送邮件，可以发送到公司内部人员。

2、已发送

查看已发送的邮件。一是在邮箱发送的邮件，二是在会议纪要发送邮件。

3、草稿箱

一是信箱里编辑的信件草稿，二是会议纪要中保存的草稿。

会议纪要

1、记录会议的主题、出席会议的人员、会议内容、会议地点、会议时间。可以以邮件的方式发送给内部人员。

### 3.2.5系统设置

系统设置包含了用户管理、机构管理、角色管理。结构图如图3.6所示

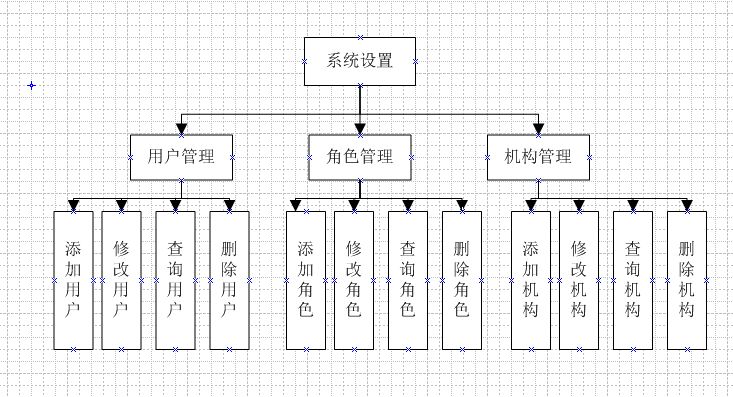


图3.6系统设置

用户管理

1. 添加用户

用户的维度有用户名称、登陆名、邮箱、手机号、工号、用户类型、密码、关联机构、角色。添加用户一般由管理员添加。

1. 修改用户

可以修改用户名称、角色、登陆名、邮箱、手机号、工号、用户类型、密码；一般由管理员修改。

1. 查询用户

可以根据登陆名、姓名进行模糊查询，也可以根据用户所属的机构进行检索。

1. 删除用户

支持批量删除和根据id删除。

角色管理

1、添加角色

用户角色的维度有角色名称（中英文）、是否系统数据（是”代表此数据只有超级管理员能进行修改，“否”则表示拥有角色修改人员的权限都能进行修改）、数据范围（数据范围有所有数据，部门数据，仅本人数据，按明细设置，特殊情况下，设置为“按明细设置”，可进行跨机构授权）、备注

2、修改角色

可以修改角色名称，数据范围。一般由管理员修改。

3、查询角色

可以根据角色名称（中英文）进行模糊查询。

4、删除角色

支持批量删除和根据id删除。删除角色之后，属于其的用户就会解除与之的关联。

5、权限的设置

根据角色的不同，给角色分配可见的菜单，该角色的用户登陆系统只能看到这里只有勾选的菜单。

6、分配用户

将角色分配给用户，一个用户可以有多个角色。用户登陆可以看到其所属角色的所有菜单，并拥有其角色中的最高数据权限的权限。

机构管理

1、添加机构

用户的维度有上级机构（若是没有上级可以不填）、机构名称、机构编码、机构类型（例如 部门，小组）、机构级别、主要负责人、联系地址、电话。添加用户一般由管理员添加。

2、修改机构

修改机构名称、机构编码、机构类型（例如 部门，小组）、机构级别、主要负责人、联系地址、电话。

3、删除机构

删除机构，其下的下级机构也会被删除。

### 3.2.6系统监控

1. 日志查询

日志查询，是将用户的登陆信息记录起来。记录的信息包括了操作菜单、操作用户、所在公司或是部门、请求的url、提交方式、操作ip、操作时间。日志查询提供了操作菜单、用户ID、URL和时间段的检索。日志列表中还有根据日志删除（支持批量删除）和清除的功能。

1. 连接池监控

连接池的监控有两部分组成，首页和SQL监控组成。首页可看到的信息有程序的版本、mysql的驱动、java版本、jvm虚拟机的版本、classpath路径、启动时间。SQL监控记录了sql语句、执行次数、执行时间、最大并发量、sql执行后更新的行数、读取的行数等。

## 3.3数据库表设计

### 3.3.1数据库的需求分析

根据系统的需求，数据库表也是分为四个模块。分别是客户管理、案场管理、我的面板、系统管理。其中客户管理模块包括客户信息表（fdc\_c\_customer）和合作伙伴表（fdc\_c\_partner）；案场管理有案场信息表(fdc\_b\_building)，房型信息表和销售记录表；我的面板包含信箱表、会议纪要表；系统管理包括用户信息表、机构信息表、角色信息表。

表与表之间的关系如下图所示。

客户表与房型信息表、案场信息表是一对一关系，一个客户只能买一套房；案场表与房型信息表是多对多关系，一个案场包含多种房型，一种房型也能对应多个案场；用户表与机构表是多对一的关系，一个员工能属于一个部门，一个部门有多个员工；用户表与角色表是多对多关系，一个用户有多个角色，一个角色也具有多个用户。

### 3.3.2数据库的概念结构设计

以下是数据表的逻辑结构设计，如下表3.1-3.11所示。

**表3.1 客户信息信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 客户名称 |
| way | varchar | 64 | 认知途径 |
| contacts | varchar | 64 | 置业顾问 |
| phone | varchar | 64 | 电话号码 |
| target | varchar | 64 | 目标户型 |
| houses | varchar | 64 | 楼盘 |
| status | varchar | 64 | 客户状态 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |

**表3.2 合作伙伴信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 合作伙伴名称 |
| signing\_time | datetime | 64 | 签约时间 |
| project | varchar | 64 | 合作项目 |
| contacts | varchar | 64 | 联系人 |
| phone | varchar | 64 | 联系人电话 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |

**表3.3 邮件信息表1**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| title | varchar | 64 | 主题 |
| overview | varchar | 128 | 内容概要 |
| content | varchar | 500 | 信件内容 |

**表3.4 邮件信息表2**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| mailid | int | 11 | 邮件外键 |
| senderId | varchar | 64 | 发件人 |
| receiverId | varchar | 64 | 收件人 |
| sendtime | datatime | 64 | 发送时间 |
| readstatus | varchar | 1 | 状态 0 未读 1 已读 |

**表3.5 会议信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| theme | varchar | 64 | 会议主题 |
| caller | varchar | 64 | 召集人 |
| address | varchar | 64 | 会议地址 |
| recorder | varchar | 64 | 记录人 |
| purpose | varchar | 64 | 会议目的 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| attender | varchar | 64 | 出席者 |
| absentee | varchar | 64 | 缺席者 |
| content | varchar | 500 | 会议内容 |
| beginTime | datetime | 64 | 会议开始时间 |
| mainEmail | varchar | 64 | 主送人 |
| addEmail | Varchar | 64 | 抄送人 |
| status | Varchar | 64 | 邮件状态 |
| meet\_time | varchar | 64 | 会议消耗时间 |

**表3.6 用户信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| office\_id | varchar | 64 | 所属部门 |
| login\_name | varchar | 64 | 登陆名称 |
| password | varchar | 64 | 密码 |
| no | varchar | 64 | 工号 |
| name | varchar | 64 | 名称 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| email | varchar | 64 | 邮箱号 |
| phone | varchar | 64 | 电话号码 |
| user\_type | varchar | 8 | 用户类型 |
| login\_ip | varchar | 128 | 最后登陆ip |
| login\_date | datetime | 64 | 登陆时间 |

**表3.7 角色信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 角色名称 |
| enname | varchar | 64 | 英文名称 |
| role\_type | varchar | 64 | 角色类型 |
| data\_scope | varchar | 64 | 数据范围 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |

**表3.8 机构信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 机构名称 |
| parent\_id | varchar | 64 | 机构父级编号 |
| parent\_ids | varchar | 64 | 所有父级编号 |
| sort | varchar | 64 | 排序 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| type | char | 1 | 机构类型 |
| grade | char | 1 | 机构等级 |

**表3.9 菜单信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 菜单名称 |
| parent\_id | varchar | 64 | 菜单父级编号 |
| parent\_ids | varchar | 64 | 所有父级编号 |
| sort | varchar | 64 | 排序 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| href | varchar | 128 | 连接 |
| permission | varchar | 128 | 权限 |

**表3.10 案场信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| name | varchar | 64 | 案场名称 |
| address | varchar | 64 | 案场地址 |
| company | varchar | 64 | 所属集团 |
| charge | varchar | 64 | 案场负责人 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| phone | varchar | 128 | 联系人电话 |
| begin\_time | varchar | 128 | 开盘时间 |
| end\_time | Varchar | 64 | 预计结束时间 |
| team | Varchar | 128 | 销售团队 |
| num | int | 6 | 套房总数 |
| sale | int | 8 | 个人规定月销售额 |

**表3.11 房型信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段宽度 | 备注 |
| id | Int | 11 | 主键 |
| price | varchar | 64 | 房价 |
| type | varchar | 64 | 房型 |
| area | varchar | 64 | 面积 |
| num | varchar | 64 | 数量 |
| create\_by | varchar | 64 | 创建人 |
| create\_date | datetime | 64 | 创建时间 |
| update\_by | varchar | 64 | 更新者 |
| update\_date | datetime | 64 | 更新时间 |
| del\_flag | char | 1 | 删除标识 |
| cases | varchar | 128 | 所属楼盘 |
| spare | varchar | 128 | 剩余数量 |

# 4系统功能实现

## 4.1权限控制

该系统具有数据权限和功能权限，功能权限在用户登录的时候，继承由shiro框架提供的AuthorizingRealm类进行授权和验证。不同的角色的功能权限和数据权限不同，admin（管理员）的权限是最大的，可以访问各个模块的所有权限控制的功能。而普通的角色只能访问客户模块、案场模块、我的面板，系统设置模块是不可见。

主要代码实现：

User user = getSystemService().getUserByLoginName(principal.getLoginName());

if (user != null) {

SimpleAuthorizationInfo info = new SimpleAuthorizationInfo();

List<Menu> list = UserUtils.getMenuList();

for (Menu menu : list){

if (StringUtils.isNotBlank(menu.getPermission())){

// 添加基于Permission的权限信息

for (String permission : StringUtils.split(menu.getPermission(),",")){

info.addStringPermission(permission);}}}

// 添加用户权限

info.addStringPermission("user");

// 添加用户角色信息

for (Role role : user.getRoleList()){

info.addRole(role.getEnname());

}

## 4.2客户管理模块

### 4.2.1客户信息模块

1. 客户信息

由置业顾问或是案场管理人员添加自己接待的客户，界面显示客户名称、置业顾问、置业顾问号码、目标户型、客户状态、更新时间；操作栏可以修改和删除客户，修改案场时，目标户型会动态变化。排序是根据更新时间排序的，页面也可以根据各个字段排序。一页可以浏览10条数据，数据支持导入导出功能。如图4.1所示



图4.1客户信息

主要实现代码：

1. 合作伙伴

合作伙伴，是公司项目的合作人。信息有置业顾问填写，界面列表显示合作伙伴名称、合作公司简介、签约时间、合作项目、联系人、联系人电话，有合作伙伴名称、公司简介、签约时间进行检索。如图4.2所示。



图4.2合作伙伴

主要代码实现：

public String list(Partner partner, HttpServletRequest request, HttpServletResponse response, Model model) {

Page<Partner> page = partnerService.findPage(new Page<Partner>(request, response), partner);

model.addAttribute("page", page);//进行视图渲染

return "modules/partner/partnerList";//放回到页面

}

<!-- 分页代码 -->

<table:page page="${page}"></table:page>

## 4.3案场管理模块

### 4.3.1案场信息

1. 案场信息；记录案场的案场名称、案场位置、所属集团、负责人（关联员工），规定月销售额、开盘时间、销售团队成员；
2. 房型信息；关联案场信息、案场房子数量，记录房子类型、价格。一旦所属楼盘改变，房型信息会随之改变。如图4.3所示



图4.3房型信息

主要代码实现：

public String save(Houses houses, Model model, RedirectAttributes redirectAttributes) throws Exception{

if (!beanValidator(model, houses)){

return form(houses, model);

}

if(!houses.getIsNewRecord()){//编辑表单保存

Houses t = housesService.get(houses.getId());//从数据库取出记录的值

MyBeanUtils.copyBeanNotNull2Bean(houses, t);//将编辑表单中的非NULL值覆盖数据库记录中的值

housesService.save(t);//保存

}else{//新增表单保存

housesService.save(houses);//保存

}

addMessage(redirectAttributes, "保存房型成功");

return "redirect:"+Global.getAdminPath()+"/houses/houses/?repage";

}

### 4.3.2 销售信息

1、案场看板；记录本月的案场销售信息，当客户信息的购买状态变为已完成购买状态。其购买的套房所属的楼盘就会添加一条购买记录，楼盘套房数量会减少1，本月出售楼房会加1，相应的案场销售额会添加相应的金额。

主要实现代码：

public String lookList(Model model,String casesName,Date date1) {

List<Cases> casesList = casesService.getAllCasesList();

Date date = null;

if(!StringUtils.isNotEmpty(DateUtils.formatDateTime(date1))){

date = new Date();}else{date = date1; }

List<Map<String,Object>> record = casesService.getLookList(casesName,TimeUtils.formatDate(date, "yyyy-MM"));

Look look = new Look();

model.addAttribute("look",look);

model.addAttribute("casesName", casesName);

return "modules/cases/lookList";

}

2、销售看板；记录本月的个人销售信息，当客户信息的购买状态变为已完成购买状态。其购买的套房所属的置业顾问就会添加一条购买记录。置业顾问的签单数会加1，本月销售总额会添加相应的金额。完成率会提高（完成率=本月销售总额/销售指标\*100）。

主要代码实现：

public String salelist(Model model,String name,Date date1) {

Date date = new Date();

List<Map<String,Object>> sale = casesService.getSalelist(name,TimeUtils.formatDate(date, "yyyy-MM"));

model.addAttribute("sale", sale);

model.addAttribute("saleTime", date1);

model.addAttribute("casesName", name);

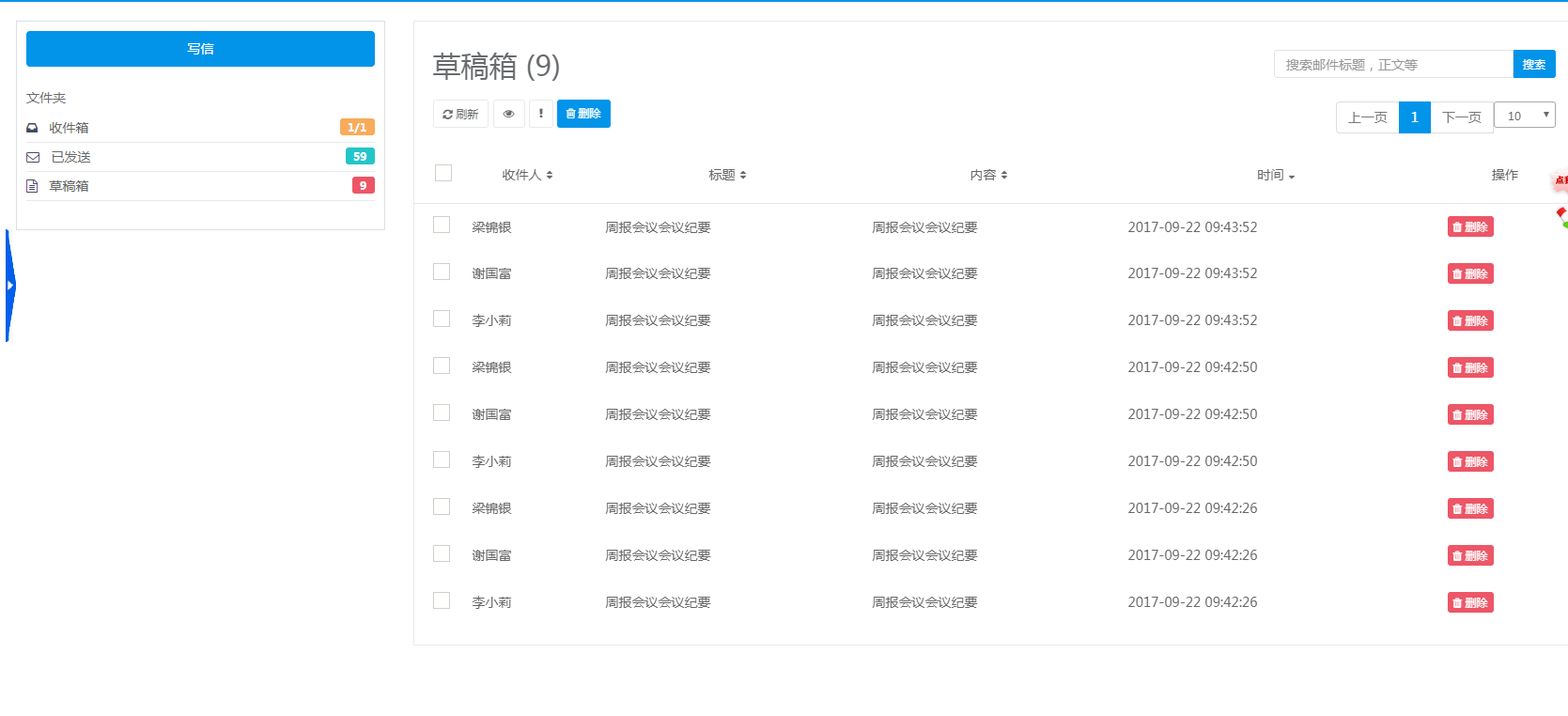
return "modules/cases/saleLook";

}

## 4.4我的面板

### 4.4.1邮箱

邮箱是提供系统内部人员使用的，可以群发邮件。可以用来提交周工作任务报告。随时随地依靠一封E-mail就能立即让相关人员及时收到文件，大大提高工作效率。将员工E-mail作为电子工作证进行管理，一旦关闭邮箱相应人员就不再有资格参与公司的工作和代表公司与客户接洽。确保了企业正常、安全的运作。整体功能界面如图4.4所示。

图4.4邮箱

主要代码实现：

public String list(MailBox mailBox, HttpServletRequest request, HttpServletResponse response, Model model) {

//查询总条数

MailBox serachBox2 = new MailBox();

serachBox2.setReceiver(UserUtils.getUser());

model.addAttribute("mailBoxCount", mailBoxService.getCount(serachBox2));

//查询草稿箱条数

MailCompose serachBox4 = new MailCompose();

serachBox4.setSender(UserUtils.getUser());

serachBox4.setStatus("0");//草稿

model.addAttribute("mailDraftCount", mailComposeService.getCount(serachBox4));

return "modules/iim/mailBoxList";

}

### 4.4.2会议纪要

记录会议的主题、出席会议的人员、会议内容、会议地点、会议时间。可以以邮件的方式发送给内部人员如图4.5所示。

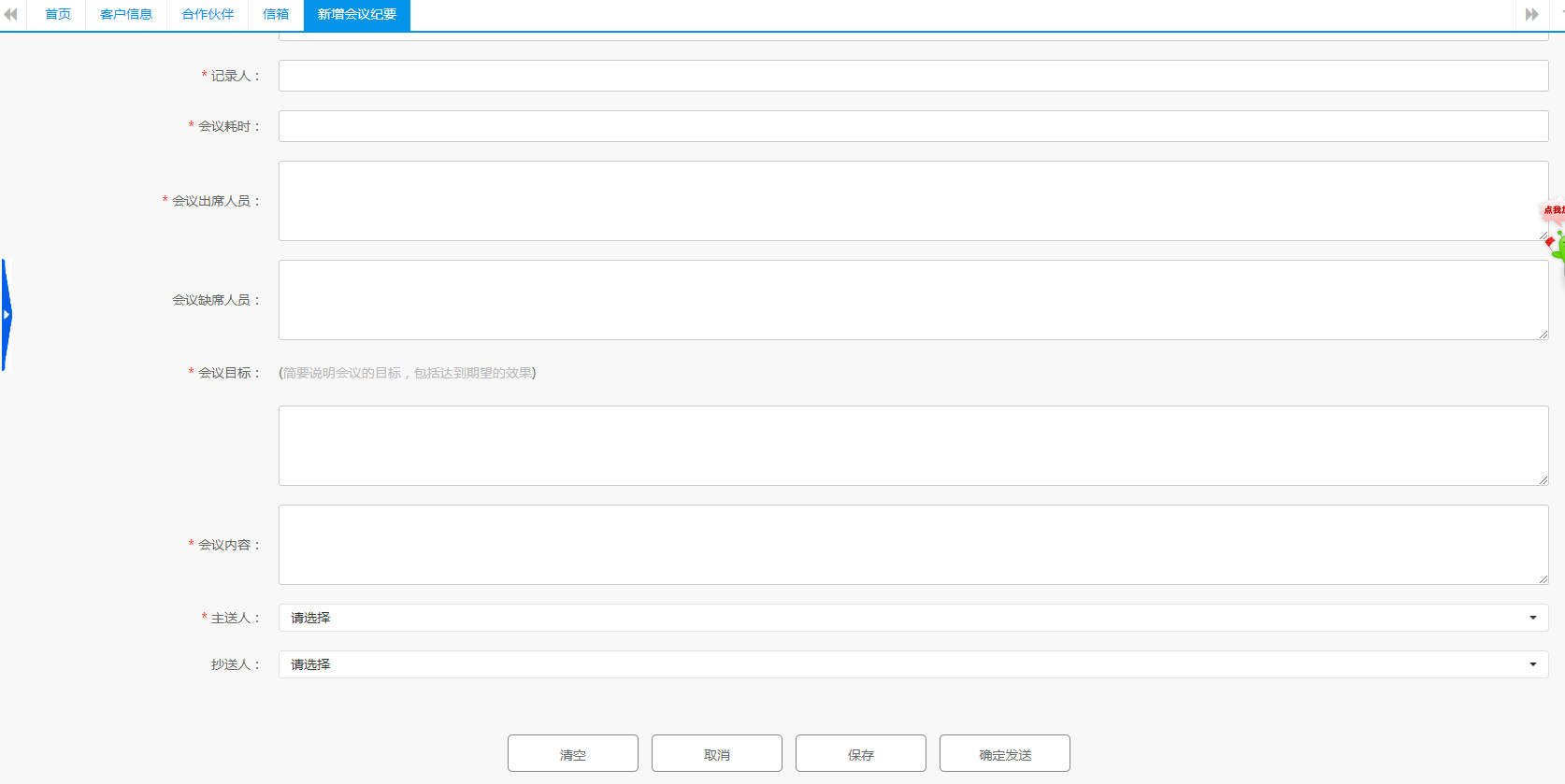


图4.5会议纪要

主要代码实现：

private void sendMassage(Mail mail,String receiverNames,String status){

List<User> receiverList=Lists.newArrayList();

for (User user : DictUtils.fundUsers())

{

if(receiverNames!=null&&receiverNames.contains(user.getName())){

receiverList.add(user);}}

for(User receiver : receiverList){

mailCompose.setReceiver(receiver);

mailCompose.setId(null);//标记为新纪录，每次往发件箱插入一条记录

mailCompose.setIsNewRecord(true);

mailComposeService.save(mailCompose);//0 显示在草稿箱，1 显示在已发送需同时保存到收信人的收件箱。

if(mailCompose.getStatus().equals("1"))//已发送，同时保存到收信人的收件箱{

MailBox mailBox = new MailBox();

mailBox.setReadstatus("0");

mailBox.setReceiver(receiver);

mailBox.setSender(UserUtils.getUser());

mailBox.setMail(mailCompose.getMail());

mailBox.setSendtime(date);

mailBoxService.save(mailBox);}}}

# 5系统测试

## 5.1测试环境

操作系统：VMware虚拟机+CentISO

数据库：5.6 Mysql数据库

服务器：Tomcat7.0+Nginx

JDK版本：JDK7.0

开发工具：eclipse

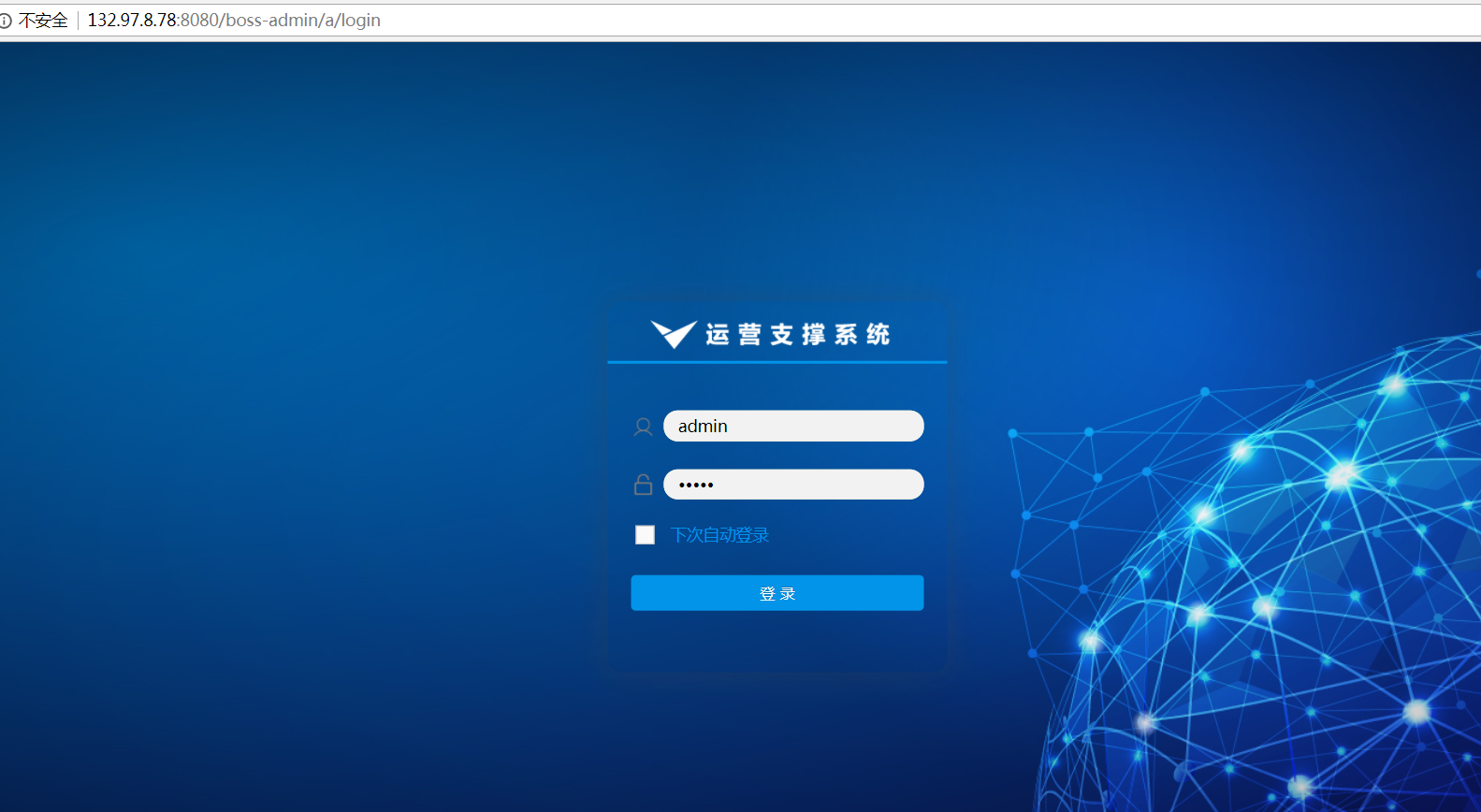
CPU:CORE i7

内存：8G

磁盘空间：20

## 5.2运行步骤

1. 工具准备：VMware虚拟机、CentOS-7-x86\_64-Everything-1708.ios、nginx-0.8.55.tar、jdk7u79linuxx64.tar、apache-tomcat-7.0.42.tar、Xshell。
2. 搭建运行环境：选择CentOS-7-x86\_64-Everything-1708.ios镜像安装好虚拟机，搭建好网络环境，一般网络使用桥式网络连接、代理网络使用net连接；用xshell连接上虚拟机（之前关闭虚拟机防火墙）；在windows下载好以上需要的工具包。进入虚拟机的local目录；如图5.1所示；在local目录下创建（mdkir）Tomcat、Nginx、java目录，将nginx-0.8.55.tar、jdk7u79linuxx64.tar、apache-tomcat-7.0.42.tar通过xshell上传到相应的目录。利用tar -zxvf \*\*.tar将以上的工具包解压。编辑配置环境Vim /etc/profile 配置好JDK环境 。复制一份Tomcat并重命名为Tomcat1和Tomcat2，改变Tomcat2的端口号。进入Nginx的conf目录并配置Nginx的代理（代理两份Tomcat）。在eclipse将首页加入区别标识中导出项目的不同的war包，将两份war包通过xshell上传到两个Tomcat的webapp目录下。启动Nginx、两个Tomcat。查看虚拟机的ip（在xshell中执行ifconfig命令）。在windows的浏览中访问网址 即可。结果如图5.2所示。



5.2登陆页面

# 6总结

至此、本系统基本完成。经分析系统可以实现管理层通过系统数据分析公司的运行情况，根据公司、案场的运营情况对公司或是案场进行方案的调整。相比之前的盲目运营，没有系统对比的运营。好的运营方式可以推荐给别的案场进行实施，不好的运营需要不断改进。现在可以实现根据客户的状态分析客户的意愿，根据每个案场的销售信息分析运营情况的好坏，根据会议纪要可以查阅案场、公司遇到的问题，如何解决，根据合作伙伴可以查看公司运营方向的拓展。

当然系统也有不足之处，系统本身内容不充足，可以再添加OA模块，用来记录员工的考勤等事宜。Nginx的负载均衡处理简单，应发布到多台不同的服务器进行代理。系统没有缓存机制，以后可以添加redis缓存机制。

# 参考文献

[1]李洋. SSM框架在Web应用开发中的设计与实现[J]. 计算机技术与发展, 2016, 26(12):190-194.

[2]徐瑞东. 房地产网站营销支持数据分析系统设计与实现. Diss. 南开大学, 2015.

[3]孙磊. 房地产销售管理系统的设计和实现[D]. 大连理工大学, 2015.

[4]王利萍. 基于Nginx服务器集群负载均衡技术的研究与改进[D]. 山东大学, 2015.

[5]怯肇乾. Tomcat应用服务器高并发优化处理[J]. 电脑编程技巧与维护, 2018(2).

[6]刘京华等.Java Web整合开发王者归来[M].北京:清华大学出版社，2010.

[7]徐涛.深入理解Bootstrap[M].北京:机械工业出版社，2014:7-8.

[8]荣艳冬, 冯建平. 深入探索Shiro权限框架的体系结构和实际应用[J]. 电脑知识与技术.

[9]乔岚. 基于MyBatis和Spring的JavaEE数据持久层的研究与应用[J]. 信息与电脑(理论版), 2017(8):73-76.

[10]温立辉. Spring框架在模型层的应用及原理[J]. 福建电脑, 2017, 33(5):147-148.

［11］林泽炜. 房地产动态成本管理系统探析[J]. 住宅与房地产, 2017(3).

［12］徐荣, 郭娜, 李金鑫,等. 我国房地产价格波动对系统性金融风险影响的动态机制研究——基于有向无环图的分析[J]. 南方经济, 2017, V36(11):1-17.

［13］孙成申. 房地产项目品牌支撑体系研究[J]. 住宅与房地产, 2016(18).

［14］佟延飞, 程士东. 建筑工程模板支撑系统计算的探讨[J]. 中国房地产业, 2016(6).

［15］赵彦辉. 房地产开发风险多因素层次决策支持系统运用探讨[J]. 商品与质量, 2016(43).

# 致谢

经过这段时间的努力，运营支撑系统终于完成了。从刚开始的以为很容易到什么都不容易的过程学会了很多很多。在编码运营支撑系统过程中，问题总是总是来光顾，但是有问题的产生就有它的解决方法和解决它的人。

首先，感谢王春安导师的细心指导，从选题到定题目再到论文的编写，王导师都在指导和给有意义的意见，使我受益匪浅，茅塞顿开。另外我还要感谢微构科技的同事，他们的细心的教导，使刚入java门槛的我向前跨出了一大步，在开发的过程中，他们给予我很多创新性的建议和帮助。我很庆幸遇到他们，一群努力又有乐趣的人。然后还要感谢学校四年的培育，经过四年的教导，无论是技术还是为人处世，我都有一个质的飞越。学校给了我一份完整的面向社会的面试推荐书，感激母校！

最后，我要感谢在我身后一直支持我、鼓励我的亲人！

感谢你们！