融资公司客户管理系统

──系统管理与业务处理

学 院： 计算机工程学院

专 业： 软件服务外包

年 级： 2013

姓 名： 张 伟

指导教师： 范会联、戴茂华(安博)

**独　创　性　声　明**

本人声明所呈交的学位论文是本人在老师指导下进行的研究工作及取得的研究成果，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：　　　　　 日期：

**学位论文版权使用授权书**

本学位论文作者授权长江师范学院可以将学位论文的全部或部分内容编入有关数据库进行检索，允许论文被查阅和借阅，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：　　 　 指导教师签名：

日　 　期：　　　　 日　　　　期：

目 录

[摘 要 I](#_Toc470015547)

[第1章 概述 1](#_Toc470015548)

[1.1 论文研究背景与意义 1](#_Toc470015549)

[1.2 论文组织 1](#_Toc470015550)

[1.3 小结 2](#_Toc470015551)

[第2章 需求分析 3](#_Toc470015552)

[2.1 系统功能需求 3](#_Toc470015553)

[2.2 系统管理 3](#_Toc470015554)

[2.2.1 角色管理 3](#_Toc470015555)

[2.2.2 用户管理 4](#_Toc470015556)

[2.2.3 资源管理 4](#_Toc470015557)

[2.2.4 产品管理 4](#_Toc470015558)

[2.2.5 客户管理 5](#_Toc470015559)

[2.3 业务流程 5](#_Toc470015560)

[2.3.1 产品购买 5](#_Toc470015561)

[2.3.2 更新状态 5](#_Toc470015562)

[2.4 微信端 6](#_Toc470015563)

[2.5 非功能需求 6](#_Toc470015564)

[2.5.1 可靠性 6](#_Toc470015565)

[2.5.2 安全性 7](#_Toc470015566)

[2.6 小结 7](#_Toc470015567)

[第3章 系统设计 8](#_Toc470015568)

[3.1 框架概述 8](#_Toc470015569)

[3.2 设计原则 8](#_Toc470015570)

[3.3 数据库设计 8](#_Toc470015571)

[3.3.1数据库结构设计 9](#_Toc470015572)

[3.3.2数据库表详细描述 9](#_Toc470015573)

[第4章 系统实现 17](#_Toc470015574)

[4.1 用户登录 17](#_Toc470015575)

[4.2 系统管理 19](#_Toc470015576)

[4.2.1 用户管理 19](#_Toc470015577)

[4.2.2 角色管理 20](#_Toc470015578)

[4.2.3 资源管理 21](#_Toc470015579)

[4.2.4 客户管理 22](#_Toc470015580)

[4.2.5 产品管理 23](#_Toc470015581)

[4.3 产品购买 24](#_Toc470015582)

[4.4 订单 24](#_Toc470015583)

[4.5 微信端 26](#_Toc470015584)

[4.6 小结 28](#_Toc470015585)

[第5章 结束语 29](#_Toc470015586)

[参考文献 30](#_Toc470015587)

# 摘 要

客户管理是企业信息化的重要组成部分，办公自动化程度、员工计算机应用能力、企业信息化水平、企业管理水平的提高都有利于客户关系管理的实现。我们很难想象，在一个管理水平低下、员工意识落后、信息化水平很低的企业从技术上实现客户关系管理。有一种说法很有道理：客户关系管理的作用是锦上添花。进入二十一世界，信息化、网络化的理念在我国很多企业已经深入人心，很多企业有了相当的信息化基础。

电子商务在全球范围内正开展的如火如荼，正在改变着企业做生意的方式。通过Internet，可开展营销活动，向客户销售产品，提供售后服务，收集客户信息。重要的是，这一切的成本是那么低。

客户信息是客户关系管理的基础。数据仓库、商业智能、知识发现等技术的发展，使得收集、整理、加工和利用客户信息的质量大大提高。

本文研究的融资公司客户管理系统是基于Internet环境下的融资公司客户管理系统，它采用了Internet技术与信息管理系统相结合的开发方法，系统的开发实现采用Java语言，MySQL5.6作为后台数据库。该客户管理系统包括前台页面、微信端系统和后台管理系统。该客户管理系统采用Web技术及目前常用的Browser／Server的模式，使用spring+hibernate+springmvc作为框架开发，并使用shiro作为权限管理。本系统只需要通过电脑联网即可使用。论文首先对客户管理系统的开发背景进行分析阐述，接下来针对国内外客户管理的现状对系统的功能需求进行了分析，再对非功能性的需求做了相应的描述，针对系统框架结构、功能设计、数据库等内容给出了相应的设计方案，接下来是详细介绍了系统主要功能的实现及编写的代码，同时在测试阶段给出相应的依据和结果，最后对整个系统进行了总结说明及展望。

关键词： MVC结构、融资公司、客户管理

# 第1章 概述

## 1.1 论文研究背景与意义

客户是公司最宝贵的资源，为了更好的发掘老客户的价值，并开发更多新客户，XX融资公司决定实施客户关系管理系统。希望通过这个系统完成对客户基本信息、产品信息、业务员处理业务信息的充分共享和规范化管理；希望通过对产品销售机会、客户在购买产品过程的追踪和记录，提高对新客户的开发能力。并提供相关报表，以便公司高层随时了解公司客户情况。

作为计算机应用的一部分，使用计算机对客户信息进行管理，具有很多优点和好处。例如：查找快捷、咨询方便、可靠性高、存储量大、保密性好等。因此客户信息管理的信息化、网络化已成为未来的必然趋势，开发融资公司客户管理系统已成为必要。

现在很多融资公司管理客户信息方式为手工管理，效率低下、易出错、手续繁琐，而且耗费大量的人力，物力，财力。一般来说，融资公司的客户量都是很大的，客户对产品的了解也只能通过纸质的形式来了解，这样是很不方便的。现在急需要一套融资客户管理系统来完成这些操作。

我所研究的融资公司客户管理系统是专门融资公司的管理而开发的，融资公司客户管理越来越趋向于信息化管理，客户信息管理也由传统报表式管理过渡到信息化管理.融资公司客户管理系统是一套先进的管理模式，其实要取得成功，必须有强大的技术和工具支持，是实施融资公司客户管理必不可少的一套技术和工具集成支持平台。融资公司客户管理系统基于网络、通讯、计算机等信息技术，能摆脱当前融资公司用文本的方式进行管理，提高公司管理效率，它能快速适应现代化管理水平和减轻工作人员繁重而冗长的工作量，使得公司员工在管理客户时达到事半功倍的效果，减轻人力的负担，方便数据的存储。

## 1.2 论文组织

论文将从下面几个部分进行论述：

第l章为概述部分，首先对课题的研究背景与研究意义进行了全面的介绍，其次对各个行业及各个领域客户管理系统的现状以及本文的主要工作和论文结构进行了概括。

第2章为需求分析。先以客户管理应用为基础，对客户管理系统的业务进行了需求分析，再从客户管理系统中功能需求和非功能需求两个方面对系统需求分析进行了描述。

第3章为系统架构设计。主要是介绍客户管理系统的设计原则和架构设计方法。包括系统的各个模块的详细设计、系统的部署架构、架构以及安全性等。

第4章为系统的实现，主要介绍客户管理系统的实现过程以及对系统的测试，包括代码实现和主要功能页面展示等等。

第5章为本文的结束语。论文对系统整个开发过程的一个总结与展望，主要就系统的一些不足以及解决方案提出自己的意见。

## 1.3 小结

本章首先阐述了本次论文的研究背景与研究意义，然后简短阐述了国内外客户管理系统的研究现状。最后，针对客户管理系统的框架结构与论文的组织结构进行了描述

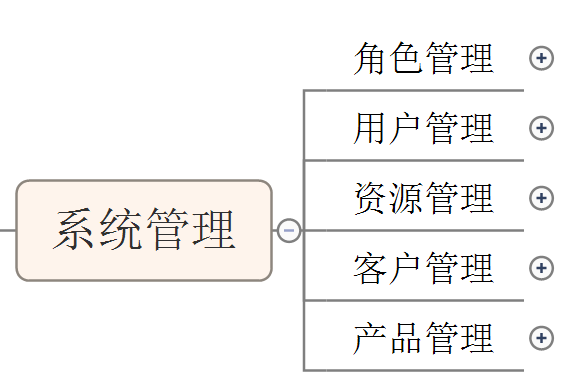
# 第2章 需求分析

## 2.1 系统功能需求

融资公司客户管理越来越趋向于信息化管理，客户信息管理也由传统报表式管理过渡到信息化管理.融资公司客户管理系统是一套先进的管理模式，其实要取得成功，必须有强大的技术和工具支持，是实施融资公司客户管理必不可少的一套技术和工具集成支持平台。融资公司客户管理系统基于网络、通讯、计算机等信息技术，能摆脱当前融资公司用文本的方式进行管理，提高公司管理效率，它能快速适应现代化管理水平和减轻工作人员繁重而冗长的工作量，使得公司员工在管理客户时达到事半功倍的效果，减轻人力的负担，方便数据的存储。

## 2.2 系统管理

系统管理主要分为：用户管理功能、角色管理功能、资源管理功能、产品管理功能、客户管理功能；如下图所示：



## 2.2.1 角色管理

进入本系统须要登录。登录成功之后，不同的角色我们只需要让他看见特定的界面，所以为用户分配角色是非常重要的。该功能权限只能管理员所拥有。该模块主要提供角色基本信息管理。

系统用途：

角色基本信息管理功能：角色基本维护。角色基本维护是对角色信息的添加、修改、删除。

## 2.2.2 用户管理

进入本系统须要登录。所以用户管理功能是非常重要的。该功能权限只能管理员所拥有。该模块主要提供用户基本信息管理。

系统用途：

用户基本信息管理功能：用户信息查询和用户基本维护。用户信息查询对用户基本信息及特定信息进行查询操作，找到相应的用户；用户基本维护是对用户信息的添加、修改、删除。

## 2.2.3 资源管理

进入本系统须要登录。登录成功之后用户客户访问的界面是有固定，不同的用户拥有的资源是不一样的,所以用户管理功能是非常重要的。该功能权限只能管理员所拥有。该模块主要提供资源基本信息管理。

系统用途：

资源基本信息管理功能：资源信息查询和资源基本维护。资源信息查询对用户基本信息及特定信息进行查询操作，找到相应的资源；资源基本维护是对资源信息的添加、修改、删除。

## 2.2.4 产品管理

一个融资公司对应的有自己的产品，客户才能购买产品，对产品的管理是该系统的重要组成部分。该模块主要提供产品基本信息管理和产品数据导出。

系统用途：

产品基本信息管理功能：产品信息查询和产品基本信息维护。产品信息查询是对产品基本信息进行特定的查询，找到相应的产品；产品基本维护是对产品信息的添加、修改、删除。添加产品也可以从excel导入，给用户避免了过多的文字录入效果，方便快捷。

产品数据导出：直接导出到excel，方便游览和统计

功能特点：

可以快速查询出特定条件的产品。方便游览和操作。导入、导出excel功能，为用户带来更好地效果。

## 2.2.5 客户管理

客户信息管理功能为核心功能，用于管理客户各种信息。该模块主要提供以下功能：客户基本信息管理。

系统用途：

客户基本信息管理功能：客户信息查询和客户基本维护。客户信息查询是对客户基本信息及特定信息进行查询操作，找到相应的客户；客户基本维护是对客户信息的添加、删除、修改。添加客户信息可以通过excel形式导入，很多时候，才开始客户的信息。我们都是在excel里面的。减少了业务员的录入量。

系统特点：

在查询客户时，可以根据客户的各种分类快速查询；添加客户可以通过excel形式导入，方便快捷；也可以通过excel形式导出客户信息，方便业务员统计和查看。

## 2.3 业务流程

### 2.3.1 产品购买

产品购买时本系统的核心功能，也是本系统存在的重要价值，公司推出了产品，产品的购买流程和操作，就成了关键的一步。该模块主要提供以下功能：购买产品。

系统用途：

公司推出了产品，客户要购买产品。产品购买就有了存在的必要。产品是留住客户的主要原因，客户在购买产品时，也是想要一个方便快捷的处理过程，业务员在操作时也是想要一个方便快捷的操作。客户在购买产品时，需满足该产品的条件。系统能自动匹配这些信息，不用业务员手动去查看，提高了效率。

系统特点：

客户在购买产品成功之后，会收到手机短信提示，如果客户关注了本融资公司的微信公众号，在微信上,同时也能收到发送过来的通知。为客户提供多种提示方式。

### 2.3.2 更新状态

客户购买产品成功之后，是要走一定的业务流程的，比如：签订合同—收齐资料—提交渠道处—收费—完成服务；每当进行到一定的业务流程时，就会需要更新相应的状态。本模块提供的功能是：更新状态、通知客户状态改变。

系统用途：

更新订单状态，业务员只需要在操作界面，根据订单号快速定位到需要更新的订单，直接更新订单，订状态就更新成功了。

系统特点：

订单状态更新成功的同时，客户会收到短信提示，如果客户关注了本融资公司的微信公众号，在微信上,同时也能收到发送过来的通知。为客户提供多种提示方式。

## 2.4 微信端

微信端，融资公司提供一个微信公众号，用户关注。本模块提供以下功能：查看产品信息；查看自己的订单信息；微信模板消息通知；

系统用途：

查看产品信息模块：客户通过微信，随时查看最新的产品的信息，更加的方便。订单信息模块：客户通过微信公众号，绑定自己的信息，经过系统核实成功之后，查看到自己所购买的产品。微信模板消息通知：当订单状态改变时，微信自动给客户发送信息，让客户随时知道自己的订单状态。

## 2.5 非功能需求

软件的易用性、可靠性、安全性等方面可称为软件的非功能性需求。非功能性需求是软件需求的重要方面，也是对功能性需求的重要支撑。非功能需求规定了系统必须满足的服务水平、系统非运行时间的属性以及系统必须遵守的约束。非功能需求可分为多个层面，有系统、模块的或者某个特例的非功能行需求。它虽不会直接影响系统功能，但用户的可操作性、用户体验等方面有很大的影响。会直接影响到用户对系统的满意度。

### 2.5.1 可靠性

故障率、可恢复性和可维护性都是系统可靠性的体现。依据对客户管理工作的需求分析，有如下的技术指标要求：

1．低的平均故障率

首先无不可恢复性故障发生，然后使得故障发生频率不得高于每个月一次。

2．故障可恢复性高

要提供方便的系统备份恢复工具，这就要有备份机制。凡是重要数据都要进行有效性检查，确保发生故障的时候不会产生无效数据。

3．可维护性高

为方便管理员进行系统设置和管理，对系统进行定期备份和在发生故障时恢复系统，必须提供方便地系统管理工具和系统备份／恢复工具。

### 2.5.2 安全性

下面从管理安全、系统安全、数据安全三个方面来对系统的安全性进行阐述。

1．客户管理系统的管理安全管理安全主要在表现在以下几个方面：

(1)系统登陆使用严格的身份认证机制，确保只有注册了的用户才能登陆，防止伪造身份人员登录系统。

(2)提供系统功能权限控制机制，禁止不合法操作，避免一个用户对其他用户数据的误操作。

(3)系统还要提供日志管理，记录用户的登陆和操作记录。

2．整个系统安全

系统安全主要体现在系统级和网络层，主要的要求是系统能够不受外来入侵和破坏和防止外来攻击。对于操作系统、数据库系统要求符合相关的安全标准，提供管理、监控、审计和故障处理等相关功能。

3．用户数据安全

不光要确保用户存储在系统中的数据不丢失、不泄露。还要保证数据的完整性、一致性和有效性，数据访问的安全性。

## 2.6 小结

本章对融资客户管理系统的需求分析作了详细描述，在进行需求分析描述时首先对系统的总体要求进行了介绍，并对系统的功能性需求以及系统的安全性、系统性以及可扩展性需求做了详细说明。一个好的软件需求分析可以为后面系统的设计与实现奠定扎实的基础

# 第3章 系统设计

## 3.1 框架概述

本系统是基于游览器/服务器体系结构(B/S),采用的是Java语言作为基础，使用Spring + hibernate +SpringMVC框架用来表示一种分层结构，从而用来实现本系统，该框架模式可以提高开发效率，并使用shiro框架作为系统的权限管理，使用MySQL5.6作为数据库存储。

## 3.2 设计原则

经过对系统的业务场景及相关的需求分析，在本系统设计中应特别注意以下原则：

(1)安全性

本系统存放有大量有关企业客户的私密性资料，这些资料涉客户的切身利益，合同等。因此设计方案必须将安全性和稳定性优先考虑，尽最大努力确保信息安全。

(2)易用性和可管理性

必须保证担保机构内具有不同的计算机应用水平的员工都能够对本系统快速地掌握并进行方便地使用，因此要求系统具有良好的人际交互界面和简单易学的操作方法。另外系统在设计时必须考虑可管理性，尽量降低系统使用中的人力财力成本。

(3)开放性和可扩展性

体系结构上要保证整体结构具有很好的模块化设计，模块之间有明确的接口，系统之间有明确的规范；系统设计要从技术路线选择、软件产品选择、软件结构等方面保证系统的开放性口。

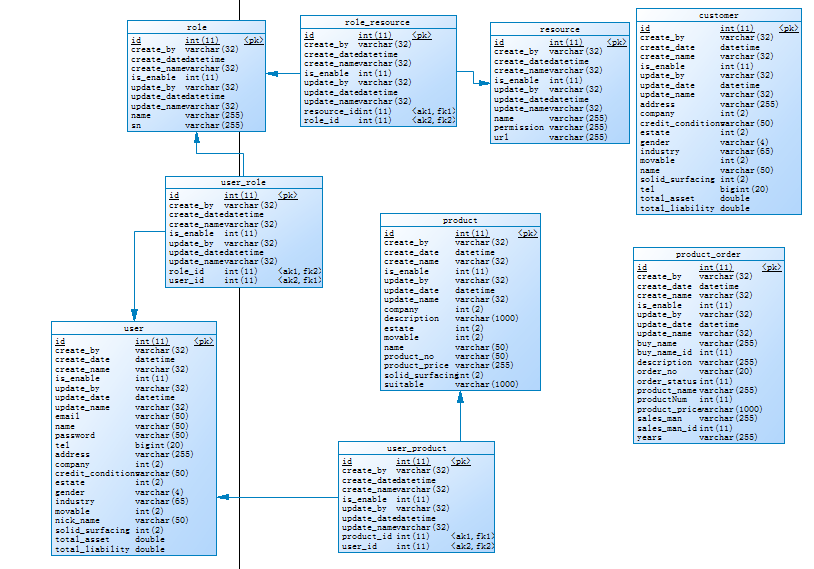
## 3.3 数据库设计

数据库在一个系统中占有非常重要的地位，优秀的数据库设计可以保证数据的完整性和一致性，减少数据的存储空问，缩小冗余。同时优秀的数据库结构的设计也有利于程序的实现。数据库设计的好坏直接影响应用系统的效率以及实现效果。

数据库是存放在一起的相关数据记录的集合，是系统的核心内容，数据库设计质量的好坏、数据结构的优劣，会影响到客户管理系统成功与否。整个系统内共享数据服务由数据服务器提供。客户管理系统的数据库设计的原则为：建立一个逻辑上统一、物理相对独立的数据库管理系统，保证系统的信息交换。在本系统的数据库设计中，尽量避免了使用外键、满足范式要求、安全等原则。

### 3.3.1数据库结构设计

本节描述平台中主要数据库实体关系（ER）模型，来表达平台业务中的关键关系，不涉及其他辅助表



### 3.3.2数据库表详细描述

客户信息表customer

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| address | address | varchar(255) | 地址 |
| company | company | int(2) | 公司 1:有,0：无 |
| credit\_conditions | credit\_conditions | varchar(50) | 征信情况 |
| estate | estate | int(2) | 房产 1:有,0：无 |
| gender | gender | varchar(4) | 性别 |
| industry | industry | varchar(65) | 行业 |
| movable | movable | int(2) | 动产 1:有,0：无 |
| name | name | varchar(50) | 姓名 |
| solid\_surfacing | solid\_surfacing | int(2) | 实体铺面 1:有,0：无 |
| tel | tel | bigint(20) | 电话 |
| total\_asset | total\_asset | double | 总资产 |
| total\_liability | total\_liability | double | 总负债 |

产品信息表 product

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| company | company | int(2) | 公司 1:有,0：无 |
| description | description | varchar(1000) |  |
| estate | estate | int(2) | 房产 1:有,0：无 |
| movable | movable | int(2) | 动产 1:有,0：无 |
| name | name | varchar(50) | 产品名称 |
| product\_no | product\_no | varchar(50) | 产品编号 |
| product\_price | product\_price | varchar(255) |  |
| solid\_surfacing | solid\_surfacing | int(2) | 实体铺面 1:有,0：无 |
| suitable | suitable | varchar(1000) |  |

订单表product\_order

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| buy\_name | buy\_name | varchar(255) |  |
| buy\_name\_id | buy\_name\_id | int(11) |  |
| description | description | varchar(255) |  |
| order\_no | order\_no | varchar(20) |  |
| order\_status | order\_status | int(11) | 1与客户签订合同---2收齐资料---3递交渠道处----4审核阶段---5下款，6收费，7完成服务 |
| product\_name | product\_name | varchar(255) |  |
| productNum | productNum | int(11) |  |
| product\_price | product\_price | varchar(1000) | 产品价格 |
| sales\_man | sales\_man | varchar(255) |  |
| sales\_man\_id | sales\_man\_id | int(11) |  |
| years | years | varchar(255) |  |

操作记录表record

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| ipv4 | ipv4 | varchar(32) | 操作 |
| record | record | varchar(5000) | 操作 |
| time | time | varchar(50) | 操作 |
| userid | userid | varchar(50) | 真实姓名 |

资源表resource

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| name | name | varchar(255) |  |
| permission | permission | varchar(255) |  |
| url | url | varchar(255) |  |

角色表role

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| name | name | varchar(255) |  |
| sn | sn | varchar(255) |  |

角色资源表role\_resource

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| resource\_id | resource\_id | int(11) |  |
| role\_id | role\_id | int(11) |  |

短信通知记录表send\_sms

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| sms\_mob | sms\_mob | bigint(20) | 短信接收手机 |
| sms\_text | sms\_text | varchar(255) | 发送内容 |
| status | status | int(2) | 状态 0：未发送，1：已发送 |

用户表user

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| email | email | varchar(50) | 邮箱 |
| name | name | varchar(50) | 登录名 |
| password | password | varchar(50) | 密码 |
| tel | tel | bigint(20) | 电话 |
| address | address | varchar(255) | 地址 |
| company | company | int(2) | 公司 1:有,0：无 |
| credit\_conditions | credit\_conditions | varchar(50) | 征信情况 |
| estate | estate | int(2) | 房产 1:有,0：无 |
| gender | gender | varchar(4) | 性别 |
| industry | industry | varchar(65) | 行业 |
| movable | movable | int(2) | 动产 1:有,0：无 |
| nick\_name | nick\_name | varchar(50) | 昵称 |
| solid\_surfacing | solid\_surfacing | int(2) | 实体铺面 1:有,0：无 |
| total\_asset | total\_asset | double | 总资产 |
| total\_liability | total\_liability | double | 总负债 |

用户角色表user\_role

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 注释 |
| id | id | int(11) |  |
| create\_by | create\_by | varchar(32) | 创建人id |
| create\_date | create\_date | datetime | 创建人时间 |
| create\_name | create\_name | varchar(32) | 创建人名称 |
| is\_enable | is\_enable | int(11) | 是否逻辑删除 0:已删除 1：未删除 |
| update\_by | update\_by | varchar(32) | 修改人 |
| update\_date | update\_date | datetime | 修改时间 |
| update\_name | update\_name | varchar(32) | 修改人名称 |
| role\_id | role\_id | int(11) |  |
| user\_id | user\_id | int(11) |  |

# 第4章 系统实现

## 4.1 用户登录

本模块实现的功能是用户登录完成验证，用户登录界面（图4.1.1）。



图4.1.1

用户输入账户、密码、验证码，点击提交，系统会验证输入信息的正确性。相关代码如下所示：

|  |
| --- |
| $("#loginbtn").click(**function**() {  **var** captcha = $("#captcha").val();  **var** loginname = $("#name").val();  **var** pwd = $("#password").val();  **var** sendInfo = {  name : loginname,  password : pwd,  captcha : captcha  };  $.ajax({  type : "POST",  url : "login",  dataType : "json",  contentType : 'application/json',  data : JSON.stringify(sendInfo),  success : **function**(data) {  **if** (data.result == "success") {  //alert("登录成功");  window.location.href = "admin/index";  } **else** **if** (data.result == "usererror") {  alert("用户名或密码有误");  } **else** **if** (data.result == "验证码错误") {  alert("验证码错误");  } **else** {  }  },  error : **function**() {  alert("没有返回值");  }  });  }); |

然后去后台验证数据的正确性，若正确，则跳转到主页（main.jsp），如果不正确，给出提示信息。后台相关验证代码如下所示：

|  |
| --- |
| @RequestMapping(value = "/login", method = RequestMethod.***POST***)  **public** @ResponseBody Object login(@RequestBody JSONObject jsonObj, HttpServletRequest request) {  log.info("开始执行登录操作====");  String name = jsonObj.getString("name");  String pwd = jsonObj.getString("password");  String captcha = jsonObj.getString("captcha");  String errInfo = "";  log.info("获取到的用户名：" + name + ",密码：" + pwd + ",验证码：" + captcha);  String randomString = (String) request.getSession(**true**).getAttribute("randomString");  log.info("缓存中的验证码为：" + randomString);  **if** (!captcha.toUpperCase().equals(randomString)) {  errInfo = "验证码错误";  } **else** {  User user = userService.getUserByNameAndPassword(name, pwd);  **if** (user != **null**) {  // shiro加入身份验证  Subject subject = SecurityUtils.*getSubject*();  UsernamePasswordToken token = **new** UsernamePasswordToken(name, pwd);  **try** {  subject.login(token);  errInfo = "success"; // 验证成功  } **catch** (AuthenticationException e) {  errInfo = "身份验证失败！";  }  } **else** {  errInfo = "usererror"; // 用户名或密码有误  }  **if** (errInfo.equals("success")) {  Record record = **new** Record();  record.setUserid(user.getId());  record.setIpv4(getIpAddr(request));  record.setRecord("登录");  record.setTime(getTime());  iRecordService.add(record);  }  }  Map map = **new** HashMap<String, String>();  map.put("result", errInfo);  **return** map;  } |

## 4.2 系统管理

### 4.2.1 用户管理

用户管理主界面（图4.2.1）所示：

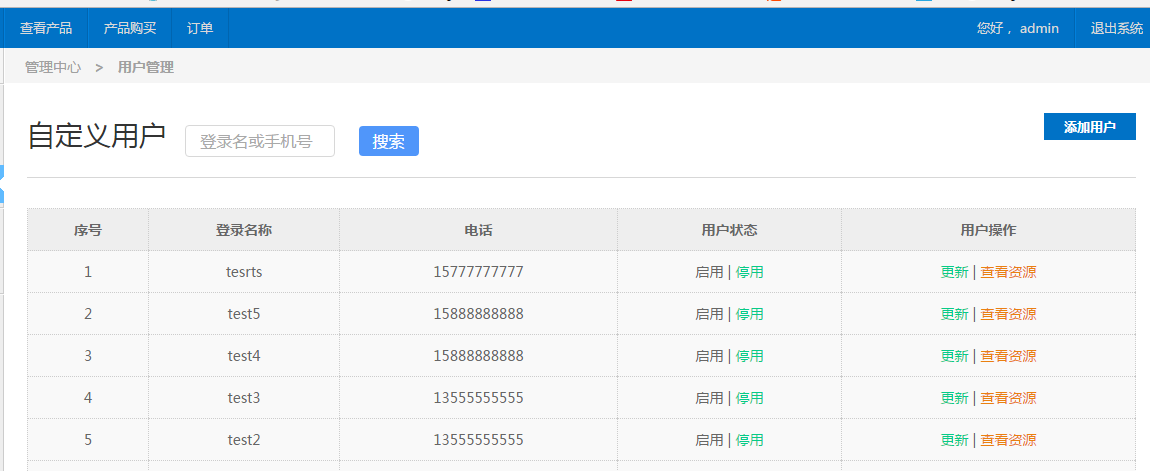


图4.2.1

管理员登录系统，可以进行用户管理全部操作。在输入框中根据提示搜索，可以快速搜索出信息。

点击“更新”可以修改用户信息，也可以查看这个用户所拥有的资源权限。

添加“添加用户”，如图4.2.2所示。



图4.2.2

录入信息，点击“确认提交”，提交到系统。相关代码如下所示：

|  |
| --- |
| **function** check(){  **var** name = $("#addName").val();  **var** pwd = $("#addPassword").val();  **var** pwd2 = $("#addPassword2").val();  **var** tel = $("#addTel").val();  **var** isEnable = $("#addisEnable").val();  }  **function** checkName(){  **var** name = $("#addName").val();  **var** patrn = /^(?!\_+$)[a-z\d\_]{2,25}$/;  **if** (!patrn.test(name))  {  divname.innerHTML='<font id="tt" class="tips\_false">2-25位长度</font>';  **return** **false**;  }**else**{  $.get("checkName?name="+name+"", **function**(msg){  **if**(msg=="success"){  divname.innerHTML='<font id="tt" class="tips\_false">角色已存在</font>';  }**else**{  divname.innerHTML='<font id="tt" class="tips\_true">正确</font>';  }  });  **if**($("#tt").hasClass("tips\_true")){  **return** **true**;  }**else**{  **return** **false**;  }    }  }  **function** checkPwd() {  **var** pwd = $("#addPassword").val();  **var** patrn=/^(\w){6,20}$/;  **if** (!patrn.test(pwd)) {  divpwd.innerHTML='<font class="tips\_false">字母和数字6-20位组成</font>';  **return** **false**;  }**else**{  divpwd.innerHTML='<font class="tips\_true">输入正确</font>';  **return** **true**;  }    } |

输入信息，全部验证成功之后，会给出提示，添加用户成功。

### 4.2.2 角色管理

角色管理界面如图4.2.3所示：

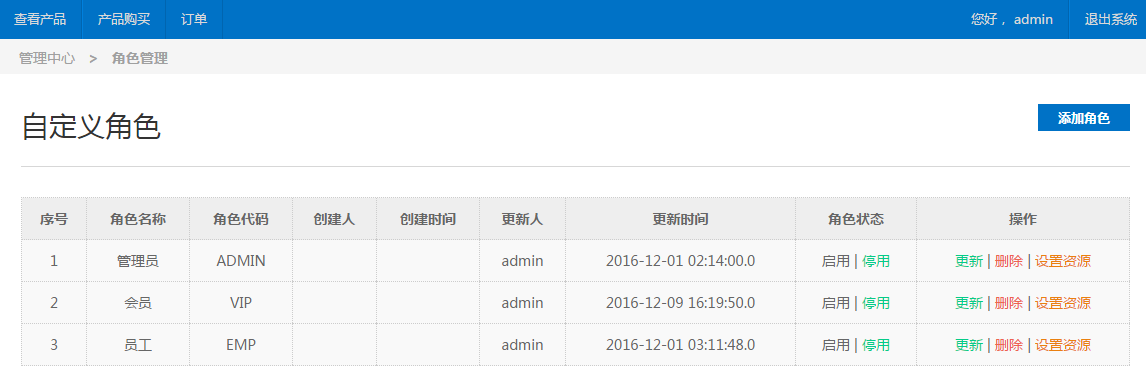


图4.2.3

点击“添加角色”，弹出信息录入框，录入信息，点击“提交”，传给后台验证，如果录入信息格式要求正确，录入成功，给出提示信息，表明添加角色成功。



图4.2.4

点击“更新”，弹出信息录入框，录入信息，点击提交“提交”，传给后台验证，如果录入信息格式要求正确，符合系统要求，给出提示，表明修改成功。  


图4.2.5

点击“删除”，直接删除该角色，添加用户时，不能选择该角色。

点击“设置资源”，为角色分配资源权限。

### 4.2.3 资源管理

资源管理是为角色设定访问系统权限。界面如图4.2.6所示：

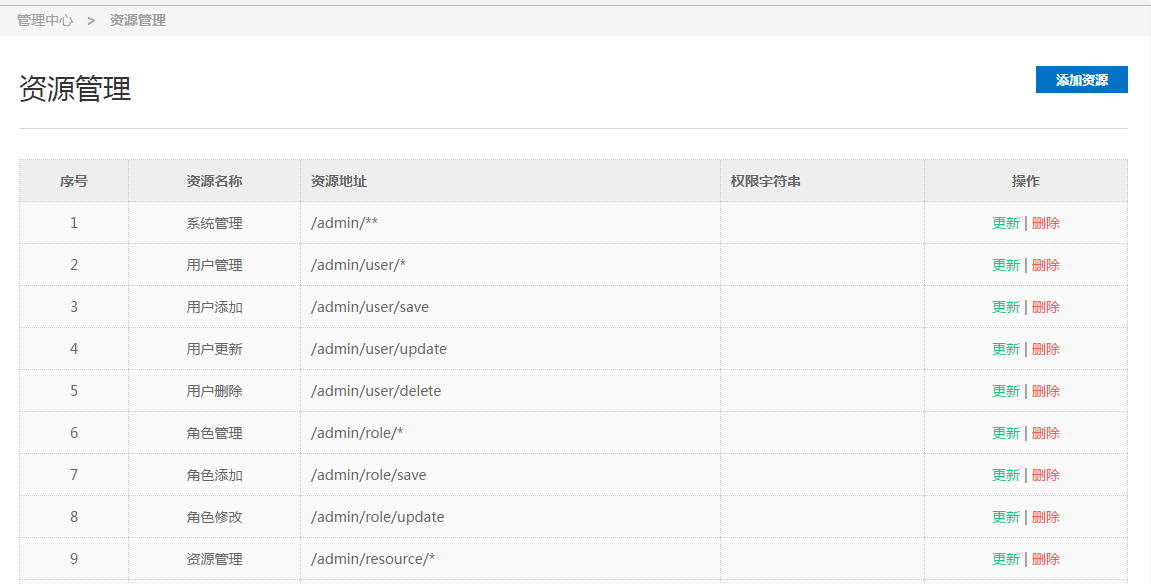


图4.2.6

该界面包括：添加新的资源权限、更新资源权限、删除资源权限。要想删除某个资源，只需要点击“删除”按钮，就可以做到了，方便快捷。如果想要添加新的资源，点击“添加资源”，弹出如图4.2.7所示界面，录入信息点击“提交”即可。更新操作也是如此。

图4.2.7

### 4.2.4 客户管理

该功能是负责客户信息的增加、编辑、查找。主界面如图4.2.8所示：

****

图4.2.8

查询功能：录入信息，点击“搜索”，信息通过ajax提交给后台，后台返回json格式的数据，前台界面显示出数据，相关代码如下图所示：

客户信息添加功能，点击“添加客户”，弹出录入信息框，录入信息点击保存按钮，提交到后台进行处理，如果录入信息符合要求，给出提示录入成功，反正给出提示，信息录入有误。客户信息添加还可以通过excel格式导入，点击“添加文件”，接着点击“导入”即可。excel格式导入信息相关代码如下所示：

客户信息修改功能，点击“更新”，弹出录入信息框，录入信息点击保存按钮，提交到后台进行处理，如果录入信息符合要求，给出提示信息，信息更新成功，反之，不成功。

客户信息导出功能，点击“导出客户信息”，即可导出客户的信息，导出信息格式为excel。

### 4.2.5 产品管理

该功能是负责产品信息的增加、编辑、查找。主界面如图4.2.9所示：



图4.2.9

查询功能：录入信息，点击“搜索”，信息通过ajax提交给后台，后台返回json格式的数据，前台界面显示出数据。

产品信息添加功能，点击“添加产品”，弹出录入信息框，录入信息点击保存按钮，提交到后台进行处理，如果录入信息符合要求，给出提示录入成功，反正给出提示，信息录入有误。产品信息添加还可以通过excel格式导入，点击“添加文件”，接着点击“导入”即可。

产品信息修改功能，点击“更新”，弹出录入信息框，录入信息点击保存按钮，提交到后台进行处理，如果录入信息符合要求，给出提示信息，信息更新成功，反之，不成功。

产品信息导出功能，点击“导出产品”，即可导出客户的信息，导出信息格式为excel。

## 4.3 产品购买

该功能是负责为客户购买产品。界面如图4.3.1所示：



图4.3.1

点击“购买”,弹出产品购买界面（图4.3.2），录入客户姓名，点击“提交”，信息提交到后台，如果客户符合购买产品的要求，就会提示购买成功，反之，不成功。



图4.3.2

## 4.4 订单

该功能是查看业务员的所做的订单信息，分为全部订单（图4.4.1）和个人订单(图4.4.2)，管理员可以看见全部订单，业务员只能看见个人订单。图4.4.1

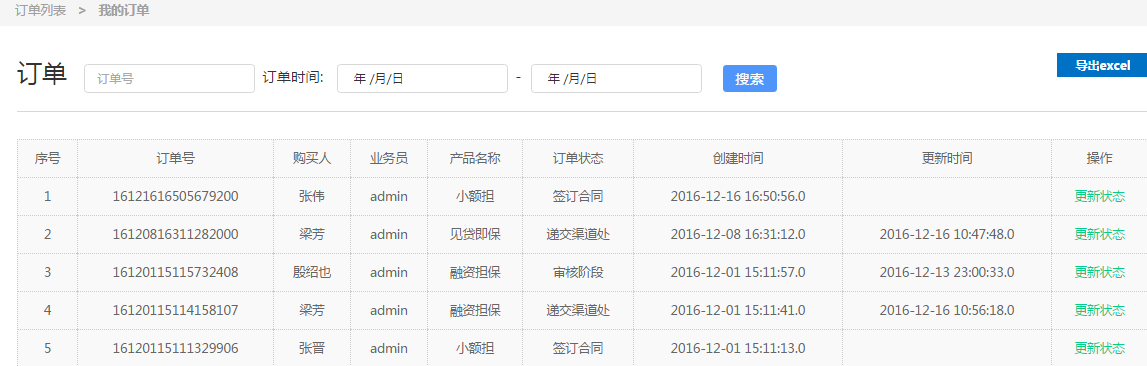


图4.4.2

个人订单和全部订单界面，都可以进行搜索，录入查询条件，点击“搜索”即可查询出指定的订单信息。

点击“更新状态”，选择状态，点击保存即可更新状态成功。

点击“导出excel”，即可导出页面显示的订单信息到excel。方便业务员统计。

在全部订单页面，点击“分析”按钮显示效果如图4.4.3所示。这个功能可以快速反应出业务员最近6个月的完成订单量，便于绩效管理。

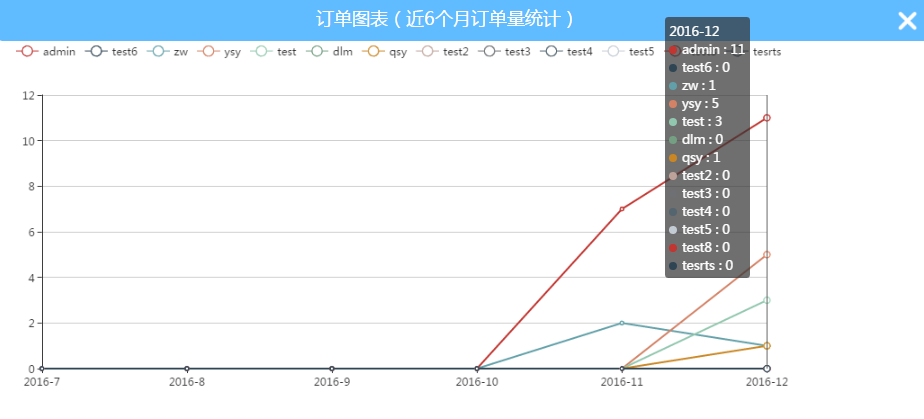


图4.4.3

## 4.5 微信端

该功能是微信公众号，用户关注公司的公众号，通过个人信息绑定账户，查看自己的订单和产品信息。也可以用来接收订单状态改变的通知。

用户关注：用户关注公众号之界面如图4.5.1所示。



图4.5.1

用户关注公众号，即可查看融资公司产品介绍（图4.5.2）。要查看个人订单信息，需要先绑定个人账户，验证成功之后，即可查看自己的订单信息（图4.5.3）。当订单状态发生改变，可以通过微信模板消息给用户发送微信消息（图4.5.4），让客户随时掌握订单情况。



**图4.5.2 图4.5.3**



**图4.5.4**

## 4.6 小结

本章主要介绍了本系统的实现，对本系统的实现状况进行了展示。对系统的用户登录实现、系统管理的实现、产品购买的实现和订单界面的实现、微信端界面的实现做了展示和详细说明。一个软件系统实现的效果好坏是该软件最终能否被需求客户认可及应用的重要标准

# 第5章 结束语

该系统目前正处在用户测试阶段还没有投入使用。由于我们系统在设计中采用了最优化的办公流程，因此在数据统计、资料报送、实时监控等方面很大程度地提高了各方用户的工作效率。

但系统还有许多不尽人意的地方，这些功能我们会在后继的开发过程中进行修改完善，根据用户的需求来加以改变。

# 参考文献

[1] (美)Bruce Eckel著 陈昊鹏译．Java编程思想．北京：机械工业出版社.2007

[2] 王珺．客户管理系统的设计与实现[学位论文]．山东：山东大学，2013

[3] 苏醒．融资担保系统的设计与开发[学位论文]．天津：南开大学，2012.5

[4] 张开涛.Shiro权限管理框架 . http://jinnianshilongnian.iteye.com/blog/2018936