数据库原理及应用课程设计

题目：物业管理系统

院 系： 软件工程

班级学号： 软件16-4（29，14，16）

姓 名： 杨赫（组长）

同组成员： 李新茹、刘沣慧

指导教师： 郑惠

2018 年 6 月 22 日

**数据库原理及应用课程设计任务书**

**一、题目：物业管理系统**

**二、设计要求**

（1）杨赫负责设计与实现信息管理（部门信息管理，员工信息管理，业主信息管理）；李新茹负责设计与实现缴费管理（缴费类型管理，缴费信息管理）；刘沣慧负责设计与实现投诉管理和车位管理。

（2）查阅相关资料，自学具体课题中涉及到的新知识。

（3）采用模块化程序设计方法，功能要完善，应有最基本的添加、删除、修改和查询功能。具有一定的创新。

（4）按要求写出课程设计报告，于设计结束后2天内提交。其主要内容包括：封皮、课程设计任务书，指导教师评语与成绩、目录、概述、需求分析、概要设计、逻辑结构、详细设计、软件的测试、总结、附录(带中文注释的程序清单)、参考文献。

**三、课程设计工作量**

一般每人的功能模块均具有最基本的添加、删除、修改和查询功能，不得抄袭。

**四、课程设计工作计划**

2018年6月18日，指导教师讲解布置题目，学生根据题目准备资料；

2018年6月18日，开始进行总体方案设计；

2018年6月19日～2018年6月21日，完成程序模块并通过独立编译；

2018年6月21日～2018年6月22日，将各模块集成为一完整的系统，并录入足够数据进行调试运行；

2018年6月22日，验收、撰写课程设计报告。

指导教师签章：

专业主任签章：

数据库原理及应用课程设计指导教师评语与成绩

|  |
| --- |
| 指导教师评语：  课程设计表现成绩：  课程设计验收成绩：  课程设计报告成绩：  课程设计 总成绩：  指导教师签章  2018年 月 日 |

目录

[第一章 概述](#_Toc29876_WPSOffice_Level1) [4](#_Toc29876_WPSOffice_Level1)

[1.1目的](#_Toc28125_WPSOffice_Level2) [4](#_Toc28125_WPSOffice_Level2)

[1.2系统开发概述](#_Toc7557_WPSOffice_Level2) [4](#_Toc7557_WPSOffice_Level2)

[第二章 需求分析](#_Toc28125_WPSOffice_Level1) [5](#_Toc28125_WPSOffice_Level1)

[2.1调查用户需求](#_Toc14259_WPSOffice_Level2) [5](#_Toc14259_WPSOffice_Level2)

[一、 小区的基本情况](#_Toc28125_WPSOffice_Level3) [5](#_Toc28125_WPSOffice_Level3)

[二、 用户对系统的要求](#_Toc7557_WPSOffice_Level3) [5](#_Toc7557_WPSOffice_Level3)

[1.2系统功能设计与划分](#_Toc32175_WPSOffice_Level2) [6](#_Toc32175_WPSOffice_Level2)

[一、管理员管理部分](#_Toc14259_WPSOffice_Level3) [6](#_Toc14259_WPSOffice_Level3)

[第三章 概要设计](#_Toc7557_WPSOffice_Level1) [10](#_Toc7557_WPSOffice_Level1)

[3.1软件结构图](#_Toc32343_WPSOffice_Level2) [11](#_Toc32343_WPSOffice_Level2)

[3.1.1 人员交互图](#_Toc32175_WPSOffice_Level3) [11](#_Toc32175_WPSOffice_Level3)

[3.1.2 PC端管理部分分布图](#_Toc32343_WPSOffice_Level3) [11](#_Toc32343_WPSOffice_Level3)

[3.2模块功能描述](#_Toc5112_WPSOffice_Level2) [12](#_Toc5112_WPSOffice_Level2)

[3.2.1 运行模块组合运行名称](#_Toc5112_WPSOffice_Level3) [12](#_Toc5112_WPSOffice_Level3)

[3.2.2 运行控制操作](#_Toc7674_WPSOffice_Level3) [12](#_Toc7674_WPSOffice_Level3)

[3.2.3 运行时间](#_Toc10801_WPSOffice_Level3) [12](#_Toc10801_WPSOffice_Level3)

[3.2.4 功能需求与模块关系](#_Toc9731_WPSOffice_Level3) [12](#_Toc9731_WPSOffice_Level3)

[3.2.5 数据结构与模块关系](#_Toc13211_WPSOffice_Level3) [14](#_Toc13211_WPSOffice_Level3)

[3.3 E-R图](#_Toc7674_WPSOffice_Level2) [14](#_Toc7674_WPSOffice_Level2)

[3.3.1 整体E-R图](#_Toc8280_WPSOffice_Level3) [14](#_Toc8280_WPSOffice_Level3)

[3.3.2 业主信息管理子系统](#_Toc3970_WPSOffice_Level3) [15](#_Toc3970_WPSOffice_Level3)

[3.3.3 投诉子系统](#_Toc16_WPSOffice_Level3) [15](#_Toc16_WPSOffice_Level3)

[3.3.4部门管理子系统](#_Toc6581_WPSOffice_Level3) [15](#_Toc6581_WPSOffice_Level3)

[3.3.5 员工管理子系统](#_Toc23174_WPSOffice_Level3) [16](#_Toc23174_WPSOffice_Level3)

[3.3.6 费用类型管理系统](#_Toc16326_WPSOffice_Level3) [16](#_Toc16326_WPSOffice_Level3)

[3.3.7 费用信息管理系统](#_Toc26143_WPSOffice_Level3) [17](#_Toc26143_WPSOffice_Level3)

[第四章 逻辑结构设计](#_Toc14259_WPSOffice_Level1) [17](#_Toc14259_WPSOffice_Level1)

[4.1转换为关系模型及优](#_Toc10801_WPSOffice_Level2) [17](#_Toc10801_WPSOffice_Level2)

[4.1.1 关系](#_Toc18738_WPSOffice_Level3) [17](#_Toc18738_WPSOffice_Level3)

[4.1.2 关系优化](#_Toc21585_WPSOffice_Level3) [17](#_Toc21585_WPSOffice_Level3)

[4.2 设计用户子模式](#_Toc9731_WPSOffice_Level2) [18](#_Toc9731_WPSOffice_Level2)

[1.管理员添加信息](#_Toc12768_WPSOffice_Level3) [18](#_Toc12768_WPSOffice_Level3)

[2.管理员的添加](#_Toc2203_WPSOffice_Level3) [18](#_Toc2203_WPSOffice_Level3)

[3.级别限制](#_Toc13211_WPSOffice_Level2) [18](#_Toc13211_WPSOffice_Level2)

[第五章 详细设计](#_Toc32175_WPSOffice_Level1) [18](#_Toc32175_WPSOffice_Level1)

[5.1建表](#_Toc8280_WPSOffice_Level2) [18](#_Toc8280_WPSOffice_Level2)

[5.1系统模块设计](#_Toc3970_WPSOffice_Level2) [19](#_Toc3970_WPSOffice_Level2)

[5.1.1 登录模块](#_Toc12707_WPSOffice_Level3) [19](#_Toc12707_WPSOffice_Level3)

[5.1.2 部门管理模块](#_Toc31893_WPSOffice_Level3) [20](#_Toc31893_WPSOffice_Level3)

[5.1.3 员工管理模块](#_Toc8339_WPSOffice_Level3) [22](#_Toc8339_WPSOffice_Level3)

[5.1.12 PC端员工信息管理添加页面](#_Toc24905_WPSOffice_Level3) [24](#_Toc24905_WPSOffice_Level3)

[5.1.4 业主管理模块](#_Toc10117_WPSOffice_Level3) [25](#_Toc10117_WPSOffice_Level3)

[第六章 测试](#_Toc32343_WPSOffice_Level1) [28](#_Toc32343_WPSOffice_Level1)

[6.1登录测试](#_Toc16_WPSOffice_Level2) [28](#_Toc16_WPSOffice_Level2)

[6.2添加测试](#_Toc6581_WPSOffice_Level2) [28](#_Toc6581_WPSOffice_Level2)

[6.3修改测试](#_Toc23174_WPSOffice_Level2) [29](#_Toc23174_WPSOffice_Level2)

[6.4删除测试](#_Toc16326_WPSOffice_Level2) [29](#_Toc16326_WPSOffice_Level2)

[6.5验证规则测试](#_Toc26143_WPSOffice_Level2) [29](#_Toc26143_WPSOffice_Level2)

[6.6退出测试](#_Toc18738_WPSOffice_Level2) [30](#_Toc18738_WPSOffice_Level2)

[第七章 总结](#_Toc5112_WPSOffice_Level1) [31](#_Toc5112_WPSOffice_Level1)

[第八章 附录](#_Toc7674_WPSOffice_Level1) [32](#_Toc7674_WPSOffice_Level1)

[8.1 部门](#_Toc21585_WPSOffice_Level2) [32](#_Toc21585_WPSOffice_Level2)

[8.1.1 部门控制器类](#_Toc20584_WPSOffice_Level3) [32](#_Toc20584_WPSOffice_Level3)

[8.1.2 部门验证器类](#_Toc25831_WPSOffice_Level3) [35](#_Toc25831_WPSOffice_Level3)

[8.1.2 部门视图](#_Toc13344_WPSOffice_Level3) [35](#_Toc13344_WPSOffice_Level3)

[8.2 员工](#_Toc12768_WPSOffice_Level2) [35](#_Toc12768_WPSOffice_Level2)

[8.2.1 员工控制器类](#_Toc12742_WPSOffice_Level3) [35](#_Toc12742_WPSOffice_Level3)

[8.2.2 员工验证器类](#_Toc662_WPSOffice_Level3) [42](#_Toc662_WPSOffice_Level3)

[8.2.3 员工视图](#_Toc20431_WPSOffice_Level3) [43](#_Toc20431_WPSOffice_Level3)

[8.2.4 登录模版类](#_Toc4698_WPSOffice_Level3) [43](#_Toc4698_WPSOffice_Level3)

[8.3 业主](#_Toc2203_WPSOffice_Level2) [44](#_Toc2203_WPSOffice_Level2)

[8.3.1 业主控制器类](#_Toc20613_WPSOffice_Level3) [44](#_Toc20613_WPSOffice_Level3)

[8.3.2 业主验证器类](#_Toc14158_WPSOffice_Level3) [48](#_Toc14158_WPSOffice_Level3)

[8.3.2 业主视图](#_Toc18184_WPSOffice_Level3) [49](#_Toc18184_WPSOffice_Level3)

# 第一章 概述

## 1.1目的

随着人们生活水平的提高和房地产业的不断的完善,大型的房地产企业结构中,物业管理的部门结构成为其中的一个细化,目前国家建设部房地产业司已将是否使用计算机进行管理列入了物业管理公司评定级别的项目之中。此举大大推动了“物业管理信息系统”的发展。而物业管理部门成为与业户零距离接触的终端部门,物业管理系统的作用不但使得繁重的手工工作变的简单化,并且能够使得业户对物业的了解更加透明化。

本系统针对现代物业管理的最基本的要求而开发，通过对人员、设备、服务、信息及各项活动实施全面管理，使各个层面的人能够通力合作，节省物业管理的人力，物力，财力的浪费。

## 1.2系统开发概述

本系统针对现代物业管理的最基本的要求而开发，通过对人员、设备、服务、信息及各项活动实施全面管理，使各个层面的人能够通力合作，节省物业管理的人力，物力，财力的浪费。

运行环境：

操作系统：Windows 2012 server

支持环境：php7.012+Apache

数 据 库：Microsoft SQL Server 2008

本系统采用C/S体系结构，由一台服务器与多个客户机组成

配置如下：

服务器：

硬件：阿里云通用型g5

CPU：2VCPu

内存：8Gib

处理器主频：2.5HZ

内网带宽：1G

内网收发包：30万PPS

硬盘空间：40G以上

软件：wamp,sqlserver2008等

输入输出设备：扫描仪、打印机等。

网络设备：网卡、网线等。

系统可运行于Linux、Windows等多种操作系统平台。

# 需求分析

## 2.1调查用户需求

本系统的最终用户为小区业主，小区物业管理人员。根据我们日常生活中的经验，结合对周边小区物业管理人员的咨询和对周边小区业主的调查，以及网上资料查询，得出用户的下列实际要求：

**一、 小区的基本情况**

业主生活在小区里，每个小区都会有若干名工作人员负责本小区的日常管理。

1. **业主的基本信息：**

入住时，每位业主都有唯一的编号，并生成一个小区物业管理系统帐号和密码。同时记录下客户的基本资料（业主资料和住房资料）。

1. **管理员的基本信息：**

每个管理员都有唯一的管理员编号，管理员登陆物业管理系统也需要相应的账号和密码，并记录管理员的基本信息；同时便于小区内业主对管理员的管理工作进行评价，如投诉某位管理员的服务不当或者消极怠工。

1. **投诉的基本信息：**  
   在日常生活当中，业主和物业管理公司必然存在一定的分歧。为满足业主需求，提高物业管理公司办公水平。业主可以向物业管理公司提出建议和投诉。业主一旦投诉，物业管理人员必须马上对投诉进行辨别与确认，继而对合理投诉进行加强改善，同时通知业主。
2. **费用管理的基本信息：**

通过对缴费类型的分类并且，管理业主是否缴纳物业管理费用信息，以及通知业主进行缴水费，电费，煤气费等费用及相关使用情况；使用户的生活更加方便快捷。

**二、 用户对系统的要求**

**1、小区物业管理员：**  **a.信息要求：**

小区物业管理员能查询上面提到的业主、单元房、公共财产管理等所有基本信息，从而实现对于小区的管理。

**b.处理要求：**

管理员可以在原有个人资料上添加、修改、删除个人信息。

当业主基本信息发生变化时，小区物业管理员能对其进行修改。比如，某些业主搬离小区，他们在本小区中相应的记录就应该删去；或者业主将单元房转售他人，他们记录中业主基本信息也要作相应的修改等等。

当业主投诉时，物业管理人员必须马上对投诉进行辨别与确认，继而对合理投诉进行加强改善，同时通知业主。

核对每月资源使用情况，计算后通知业主缴纳相应费用。

**c．安全性与完整性要求：**

**安全性要求：**

1.系统应设置访问用户的标识以鉴别是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用；

2.系统应对不同的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容；

3.系统应对不同用户设置不同的权限，区分不同的用户，如区分普通用户（学生），管理员。

**完整性要求：**

1.各种信息记录的完整性，信息记录内容不能为空；

2.各种数据间相互的联系的正确性；

3.相同的数据在不同记录中的一致性。

**2、小区业主：**

**a.信息要求：**

小区业主们能查询其所在的单元房的所有信息。能查询自己的快件信息。能查询自己的报修记录、投诉记录和缴费记录。

**b.处理要求：**

小区业主可以在个人信息中进行添加、修改、删除家庭人员信息。

小区业主可以在投诉记录中插入投诉信息，表示对于物业管理公司的建议和投诉。

月末时，小区业主能查看到自己的费用情况，并按时缴纳费用，确保生活有序进行。

**1.2系统功能设计与划分**

根据如上得到的用户需求，我们将本系统按照所完成的功能分成以下几部分：

**一、管理员管理部分**

1、处理管理员登录

2、管理员可以查询业主信息。

3、管理员可以添加，修改，删除，费用类型

4、管理员可以添加，修改，删除，车位信息

5、管理员可以插入、修改与查询业主投诉信息。

6、管理员可以发布、处理费用信息。

7、管理员可以修改管理员密码。

## 2.2用户特点

随着社会经济的快速发展人力资源物理资源日趋紧张，城市小区物业管理更加的困难，因此为了改善这种社会坏境与状况，次系统主要运用与物业与业主之间，减少人力物力的消耗，让物业人员更加方便的去了解关注小区业主的信息，同时也方便了业主更加便利的反馈信息于物业，共同加强小区建设，减少双方的财力物力消耗，更方便的服务于社会。

### 2.2.1 一般约束

无卡顿，无约束。

### 2.2.2假设和依据

假设：如果没有合理化、经济化的物业管理系统，物业与业主交流方面就会需要大量的人力、财力和物力，当业主或者物业任意一方面出现问题时不能够及时的沟通，就可能会使物业与业主矛盾激烈化，造成小区服务业与消费者之间双方利益的损失，让业主与物业方面都得不偿失。

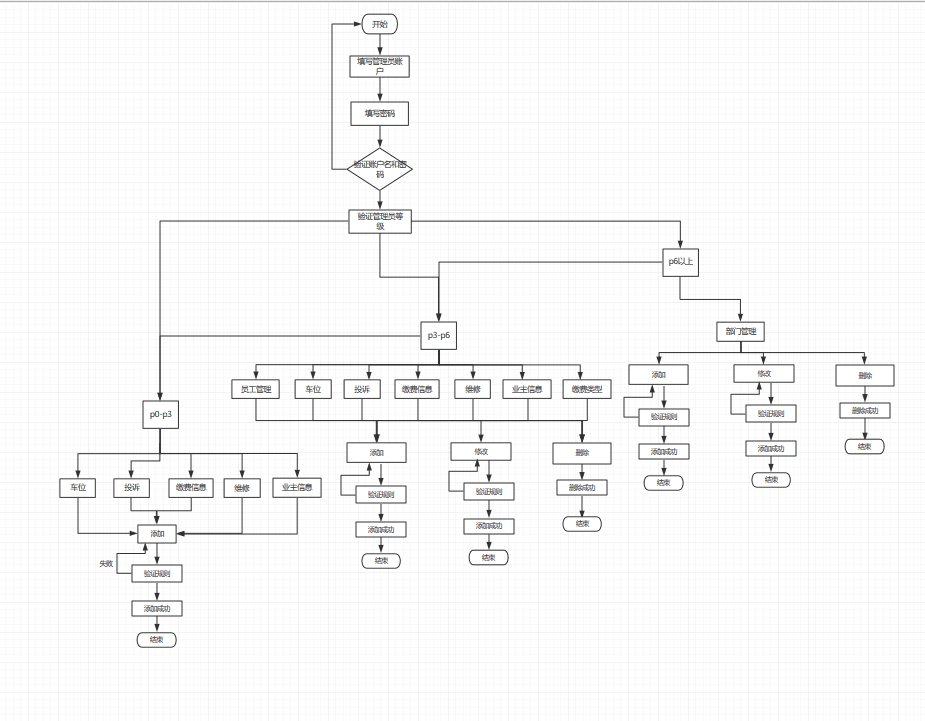
依据：物业管理行业的快速发展、经济地位的持续上涨，使其 物业管理行业的快速发展、经济地位的持续上涨使其成为了与我国经济发展与广大人民群众生活、工作息息相关的一个相对独立的新型行业，它以经营为手段，以为业主提供优质服务为宗旨，以经济效益社会效益的综合统一为目的，成为我国城市经济管理的经济重要组成部分。但是作为新生事物的物业管理行业在发展过程中遇到诸多问题和矛盾，例如房屋以及附属设备、设施的维修和养护，物业管理区域的的清洁、治安保卫、交通以及车辆管理、环境绿化以及业主或使用人生活、工作、生产方面的服务，还有代表业主或使用人就有关事宜与政府各部门或公共事业单位交涉以及房屋的使用出租管理等。因此物业管理企业要想在激烈的市场中脱颖而出就必须改变思想转变观念，用科学、系统的管理理念创造企业个性品牌、创新品牌的知名度、美誉度这是当前物业管理企业迎接市场挑战的战略思维和现实选择。我们的产品就是针对当下这些现象能够更好的解决，户主与小区物业能够更好的沟通所研发出的物业管理系统，它能更快捷，方便的处理住户于物业之间的大小事务。解决户主与物业之间矛盾，让户主与物业之间沟通更便捷，节省物业的人力，财力，物力， 让户主享受更优质的物业服务。

## 2.3具体需求

### 2.3.1 管理员添加信息限制功能

管理员添加信息功能 ：

管理员添加信息修改时需要P6以上的级别，P3以下级别的管理员只能修改部分信息。



### 2.3.2 信息处理功能

信息处理功能：

员工信息：用户名，密码，管理等级，管理员姓名，所属部门，证照图，上岗日期，电话，家庭住址，备注；

部门信息：ID部门名称主管部门电话备注修改日期；

业主信息：ID、房号、房屋类型、业主、合同资料编号、购房日期 电话 房屋状态 备注 附件；

### 2.3.3 基本服务功能

缴费管理功能：

缴费类型ID、费用名称 、费用单价、收费公司、管理员、修改日期和缴费信息ID 、房号、业主、收费类型、用量、收费金额管理员开始日期、结束日期、修改日期。

住户投诉基本信息：投诉编号、投诉日期、投诉住户、物业地址、电话、处理时间、处理人员、投诉内容；

车位管理：记录车位信息（车位ID,房号，业主姓名，车位类型，车位地址，车位时间，业 主车牌，管理ID,时间）

## 2.4业务分析

业务范围和工作流程

物业管理系统主要考虑物业管理的普遍要求,物业管理系统主要包括:

(1) 对所有房屋资料的录入和增、删、改、查询等功能实现,在基于这

些的房产资源对进行管理。

(2) 对内住户的详细资料的管理,包括增、删、改、查询等功能的实现,

这些也是一个的基本资源,毕竟物业管理最后是针对的所有住户而言

的。

(3) 在具有了所有的基本资料信息后,需要实现实质性的物业管理。主要的

管理业务包括:物业设备管理、仪表(水、电、宽带)数据管理、收费管理、住

户投诉管理、故障管理等。这些成为物业管理的主体。

这样就可以便于物业公司对进行全面的了解和管理了。

房产证 房屋资料信息

信息表 住户资料信息

数据单 其它信息

## 2.5 功能分析

（1）系统用户管理功能

系统用户的添加,包括用户名、密码信息。

（2） 住户信息管理功能

住户基本信息的录入,包括住户电话、住户姓名、物业地址、身份证号、入

住时间等信息。

住户基本信息的修改。

住户基本信息的删除。

住户基本信息的查询。

房产信息管理功能

房产基本信息的录入,包括物业地址、使用面积、房屋结构、设备、出售信

住户基本信息实体

住户姓名 住户电话 物业地址 入住时间 房屋信息查询 住户 客服部门

（3）房产基本信息管理功能

资料管理系统 数据查询 管理部门

息等信息。

房产基本信息的修改。 房产基本信息的删除。

房产基本信息的查询。

(4) 收费管理功能

其中包括物业收费和仪表收费两大类信息的录入,包括收费住址、水费、电费、宽带费、物业管理费、收费时间等信息。

收费基本信息的修改。

收费基本信息的删除。

收费基本信息的查询。

住户故障处理管理功能

（5）故障基本信息管理功能

住户报修基本信息的录入,包括住址、报修故障、经办人、处理时间、查询等信息。

住户故障基本信息的修改。

住户故障基本信息的删除。

住户故障基本信息的查询。

(6) 退出

退出物业管理系统。

## 2.6 数据收集

良好的数据系统设计:所物业管理系统应充分具备数据库的维护功能,可以及时根据用户的需求对数据库中的信息进行添加、删除、修改等操作。

完备的信息查询统计:为满足管理和决策工作的需要,在查询基础上应具备一定的统计功能。

后期维护:所开发的物业管理系统应采用模块化程序设计方法,收到既便于系统功能的各种组合和修改,又便于未参与开发的技术维护人员补充、维护的效果。

## 2.7性能需求

无卡顿，无闪退。

以上是对本信息管理系统进行的基本的需求分析,也就是本系统所要实现的基本功能。但是,这并不是整个系统的全部需求,对系统不断出现的新需求,需要系统在日后的使用中通过系统的更新与维护来实现。这样,才能使系统得到不断的完善,功能更加强大,与用户的交互性更强。

# 概要设计

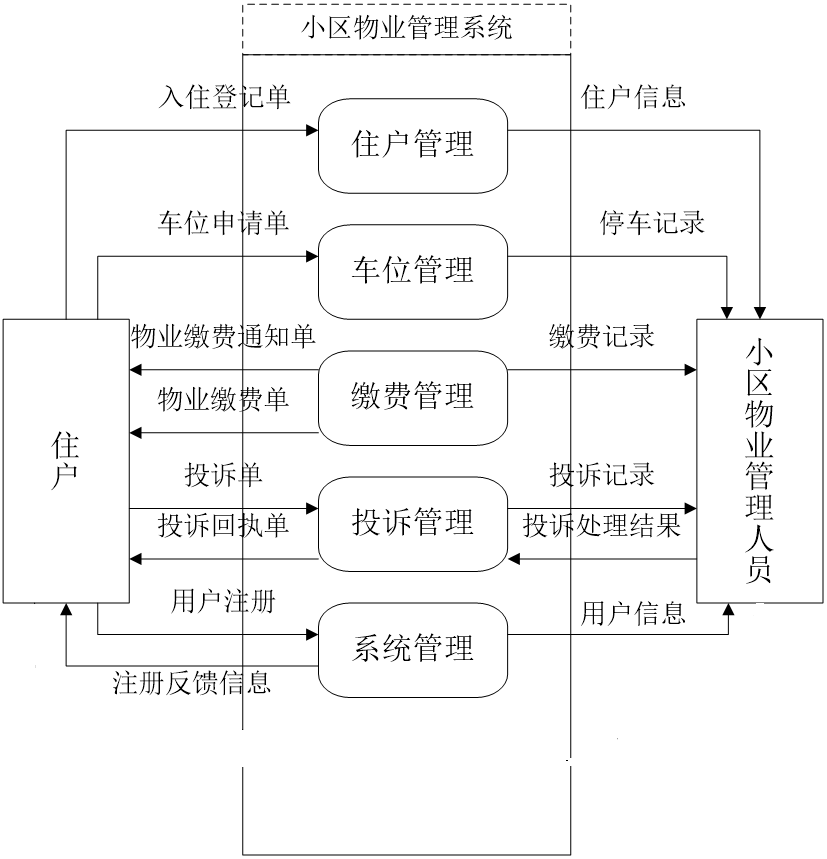
本数据库将反映的现实世界中的实体、属性和它们之间的关系等的原始数据形式，包括各数据项、记录、系、文卷的标识符、定义、类型、度量单位和值域，建立本数据库的每一幅用户视图

## 3.1软件结构图

### 3.1.1 人员交互图



### 3.1.2 PC端管理部分分布图



## 3.2模块功能描述

|  |  |  |
| --- | --- | --- |
| 输 入 | 处 理 | 输 出 |
| 信息 | 验证 | 成功或报错 |

### 3.2.1 运行模块组合运行名称

|  |  |
| --- | --- |
| 运行条件 | 支持软件 |
| 管理员登录并且有足够的权限管理 | 火狐，ie，谷歌，360浏览器等 |
| 管理员登录并且有足够的权限管理 | 火狐，ie，谷歌，360浏览器等 |
| 管理员登录并且有足够的权限管理 | 火狐，ie，谷歌，360浏览器等 |
| 管理员登录并且有足够的权限管理 | 火狐，ie，谷歌，360浏览器等 |

### 3.2.2 运行控制操作

|  |  |
| --- | --- |
| 控制方法 | 操作步骤 |
| 鼠标点击和键盘输入 | 鼠标点击添加按钮，手动输入信息完成添加 |
| 鼠标点击和键盘输入 | 鼠标点击修改按钮，手动输入信息完成修改 |
| 鼠标点击和键盘输入 | 鼠标点击删除按钮，完成删除 |
| 鼠标点击和键盘输入 |  |

### 3.2.3 运行时间

|  |  |
| --- | --- |
| 所占资源 | 时间 |
| 很少 | <0.2s |
| 很少 | <0.2s |
| 很少 | <0.2s |
| 很少 | <0.2s |

### 3.2.4 功能需求与模块关系

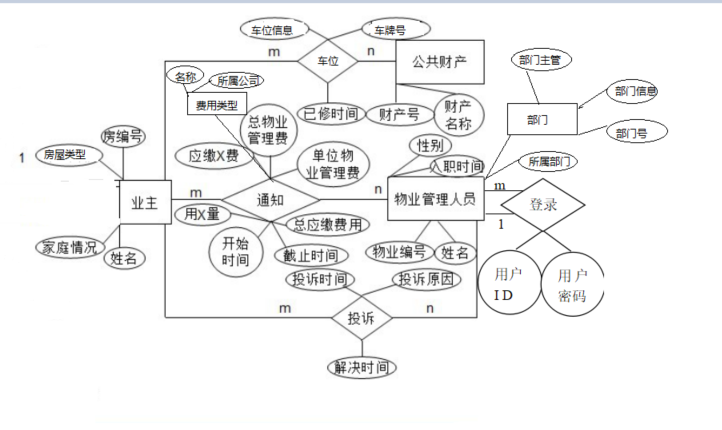
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 创建模块 | 查找模块 | 修改模块 | 删除模块 |
| 管理员添加部门信息 | √ |  |  |  |
| 管理员修改部门信息 |  | √ | √ |  |
| 管理员删除部门信息 |  |  |  | √ |
| 管理员查看部门信息 |  | √ |  |  |
| 管理员添加员工信息 | √ |  |  |  |
| 管理员修改员工信息 |  | √ | √ |  |
| 管理员删除员工信息 |  |  |  | √ |
| 管理员查看员工信息 |  | √ |  |  |
| 管理员添加业主信息 | √ |  |  |  |
| 管理员修改业主信息 |  | √ | √ |  |
| 管理员删除业主信息 |  |  |  | √ |
| 管理员查看业主信息 |  | √ |  |  |
| 管理员添加缴费类型信息 | √ |  |  |  |
| 管理员修改缴费类型信息 |  | √ | √ |  |
| 管理员删除缴费类型信息 |  |  |  | √ |
| 管理员查看缴费类型信息 |  | √ |  |  |
| 管理员添加缴费信息 | √ |  |  |  |
| 管理员修改缴费信息 |  | √ | √ |  |
| 管理员删除缴费信息 |  |  |  | √ |
| 管理员查看缴费信息 |  | √ |  |  |
| 管理员添加车位信息 | √ |  |  |  |
| 管理员修改车位信息 |  | √ | √ |  |
| 管理员删除车位信息 |  |  |  | √ |
| 管理员查看车位信息 |  | √ |  |  |
| 管理员查看投诉信息 |  | √ |  |  |
| 管理员删除投诉信息 |  |  |  | √ |
| 用户添加投诉信息 | √ |  |  |  |
| 用户修改投诉信息 |  | √ | √ |  |

### 3.2.5 数据结构与模块关系

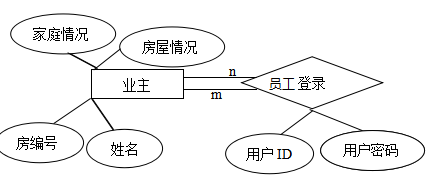
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 数据 | 增加数据模块 | 删除数据模块 | 修改数据模块 | 查询数据模块 |
| gl | √ | √ | √ | √ |
| bn | √ | √ | √ | √ |
| yz | √ | √ | √ | √ |
| ts | √ | √ | √ | √ |
| sf | √ | √ | √ | √ |
| fy | √ | √ | √ | √ |
| cl | √ | √ | √ | √ |
| ba | √ | √ | √ | √ |
| bx | √ | √ | √ | √ |

## 3.3 E-R图

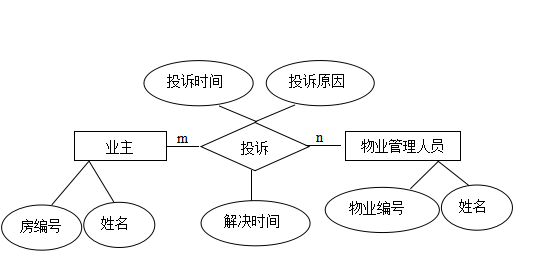
### 3.3.1 整体E-R图



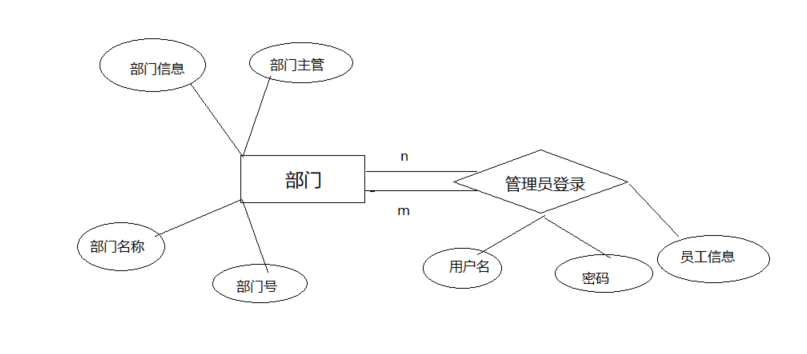
### 3.3.2 业主信息管理子系统



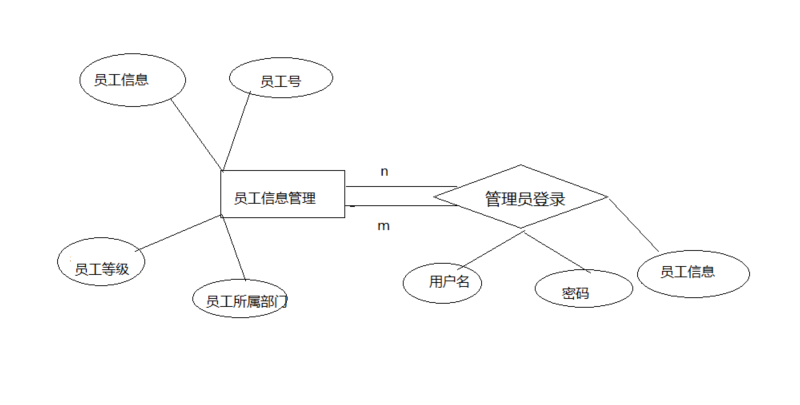
### 3.3.3 投诉子系统



### 3.3.4部门管理子系统



### 3.3.5 员工管理子系统



### 3.3.6 车位管理系统

接收业主申请申请

申请车位

申请车位

停车位

业主

房编号

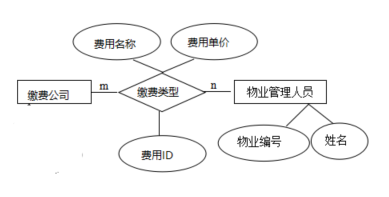
姓名

物业管理人员

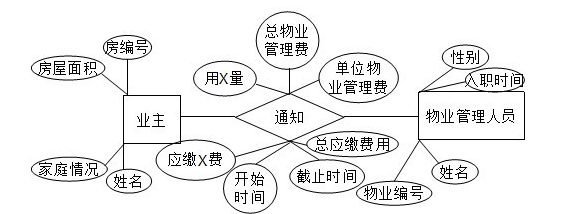
受理

车位空闲

### 3.3.6 费用类型管理系统



### 3.3.7 费用信息管理系统



# 逻辑结构设计

## 4.1转换为关系模型及优

### 4.1.1 关系

1.员工信息：（用户名，密码，管理等级，管理员姓名，所属部门，证照图，上岗日期，电话，家庭住址，备注）；

2.部门信息处理(ID部门名称主管部门电话备注修改日期)；

3.业主信息处理：（ID、房号、房屋类型、业主、合同资料编号、购房日期 电话 房屋状态备注 附件）；

4.住户投诉的基本信息处理：（投诉编号、投诉日期、投诉住户、物业地址、电话、处理时间、处理人员、投诉内容）；

5.缴费管理。记录业主所缴物业费等数据。：（缴费类型（ID 费用名称 费用单价 收费公司 管理员 修改日期）和缴费信息（ID ，房号 业主 收费类型 用量 收费金额 管理员 开始日期 结束日期 修改日期））；

6.车位管理，记录车位信息（车位ID,房号，业主姓名，车位类型，车位地址，车位时间，业主车牌，管理ID,时间）

### 4.1.2 关系优化

以上表是优化后的关系模型，优化就是要消除非主属性对主属性的部分依赖以及传递依赖。在设计关系模型用户（物业管理人员）网页登陆按规则是要写入业主表（物业管理人员表）的，但是存在了部分依赖和传递依赖，所以优化后就给独立出来。自此所有的都是符合规则的3NF。

## 4.2 设计用户子模式

1.管理员添加信息

管理员添加信息再次修改时需要p6以上级别的管理员删除，p3以上级别管理员可以修改部分信息。

2.管理员的添加

管理员添加只能由更高级别的管理员进行添加。

3.级别限制

部分信息和程序有级别限制进行观看。

4.基础功能

完成物业管理的基本服务功能

5.特色服务

适当增加与众不同的特色服务

6.员工信息：（用户名，密码，管理等级，管理员姓名，所属部门，证照图，上岗日期，电话，家庭住址，备注）；

7.部门信息处理(ID部门名称主管部门电话备注修改日期)；

8.业主信息处理：（ID、房号、房屋类型、业主、合同资料编号、购房日期 电话 房屋状态 备注 附件）；

9.住户投诉的基本信息处理：（投诉编号、投诉日期、投诉住户、物业地址、电话、处理时间、处理人员、投诉内容）；

10.缴费管理。记录业主所缴物业费等数据。：（缴费类型（ID 费用名称 费用单价 收费公司 管理员 修改日期）和缴费信息（ID ，房号 业主 收费类型 用量 收费金额 管理员 开始日期 结束日期 修改日期））；

11.车位管理视图，记录车位信息（车位ID,房号，业主姓名，车位类型，车位地址，车位时间，业主车牌，管理ID,时间）

# 详细设计

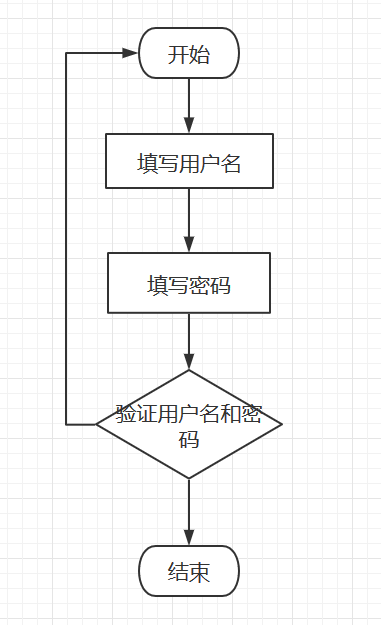
## 5.1建表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据结构 | 注释 | 数据名 | 类型 | 长度 | 备注 |
| 部门信息 | 房屋id | bn\_id | INT | 20 | 主键 |
| 部门名称 | bn\_name | VARCHAR | 20 |  |
| 管理id | gl\_id | INT | 10 |  |
| 管理姓名 | gl\_name | VARCHAR | 50 |  |
| 管理员等级 | gl\_p | CHAR | 10 |  |
| 部门电话 | bn\_phone | VARCHAR | 50 |  |
| 备注 | bn\_note | VARCHAR | 255 |  |
| 时间 | time | DATETIME |  |  |
| 员工信息 | 管理id | gl\_id | INT | 20 | 主键 |
| 用户名 | gl\_user | VARCHAR | 20 |  |
| 密码 | gl\_pwd | VARCHAR | 50 |  |
| 管理员等级 | gl\_p | VARCHAR | 20 |  |
| 管理姓名 | gl\_name | VARCHAR | 20 |  |
| 部门id | bn\_id | INT | 20 |  |
| 证照图 | gl\_picture | VARCHAR | 100 |  |
| 上岗日期 | gl\_date | DATETIME |  |  |
| 电话 | gl\_call | VARCHAR | 25 |  |
| 地址 | gl\_adress | VARCHAR | 150 |  |
| 备注 | gl\_note | VARCHAR | 255 |  |
| 时间 | time | DATETIME |  |  |
| 业主信息 | 业主id | ww\_id | INT | 20 | 主键 |
| 房号 | fw\_number | VARCHAR | 20 |  |
| 房屋类型 | fw\_style | VARCHAR | 20 |  |
| 业主姓名 | yz\_name | VARCHAR | 50 |  |
| 资料编号 | fw\_detailed | VARCHAR | 10 |  |
| 购房日期 | gf\_date | DATETIME | 10 |  |
| 业主电话 | yz\_phone | VARCHAR | 20 |  |
| 房屋状态 | fw\_state | VARCHAR | 20 |  |
| 备注 | fw\_note | VARCHAR | 255 |  |
| 时间 | time | DATETIME | 14 |  |
| 附件 | yz\_fjs | VARCHAR | 100 |  |

## 5.1系统模块设计

### 5.1.1 登录模块

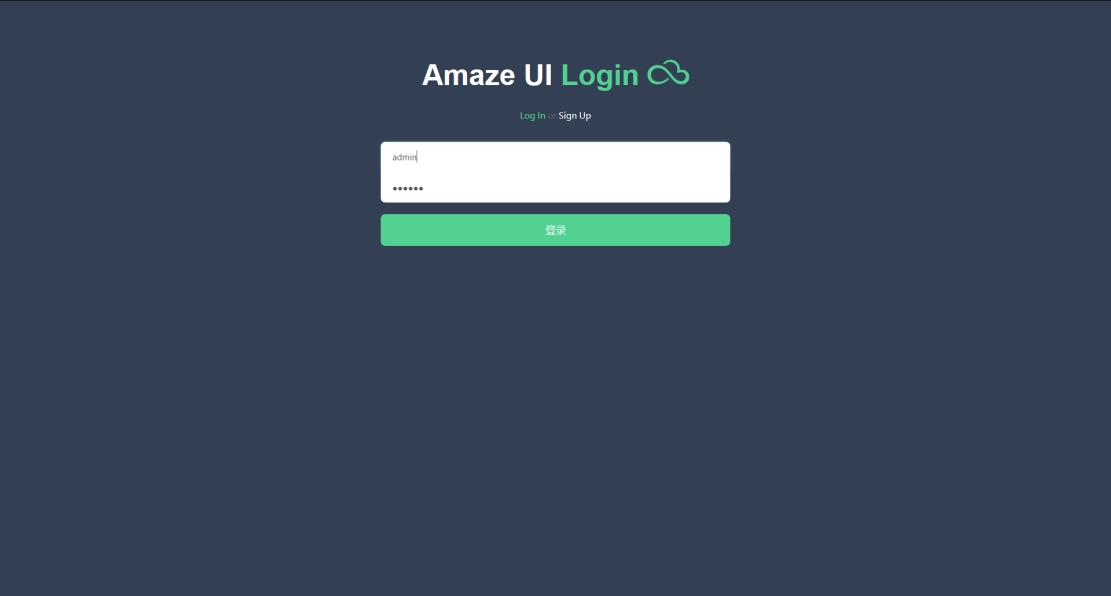
登录流程图



主界面

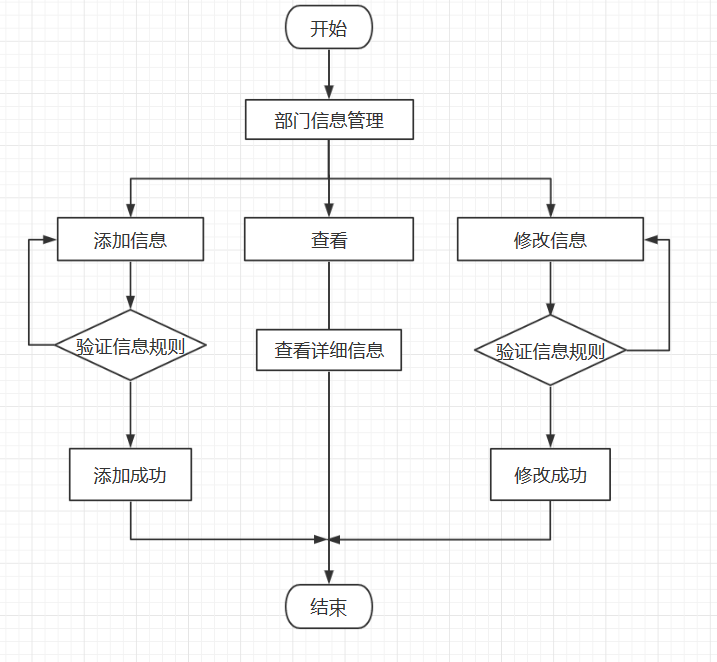
****

登录界面

****

### 5.1.2 部门管理模块

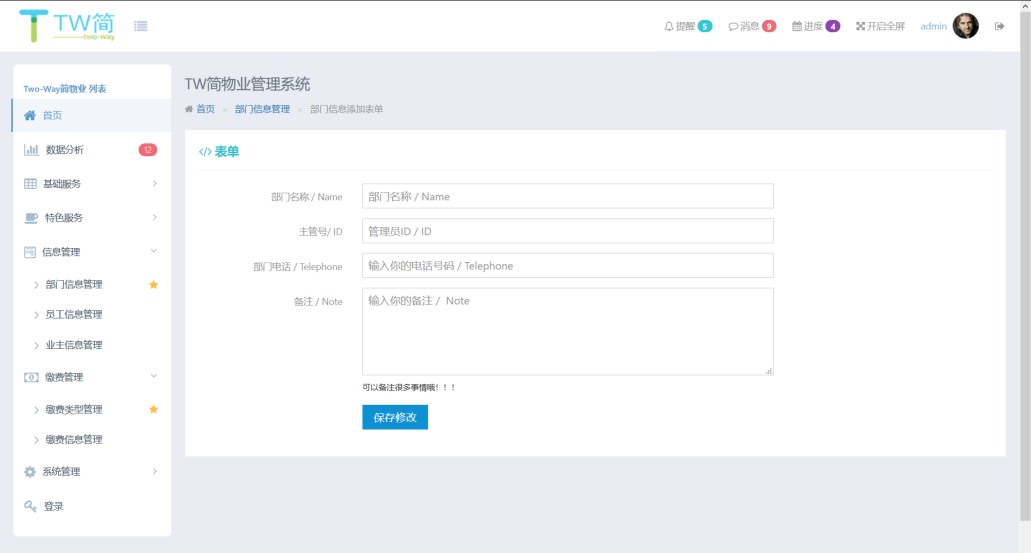
部门管理流程图



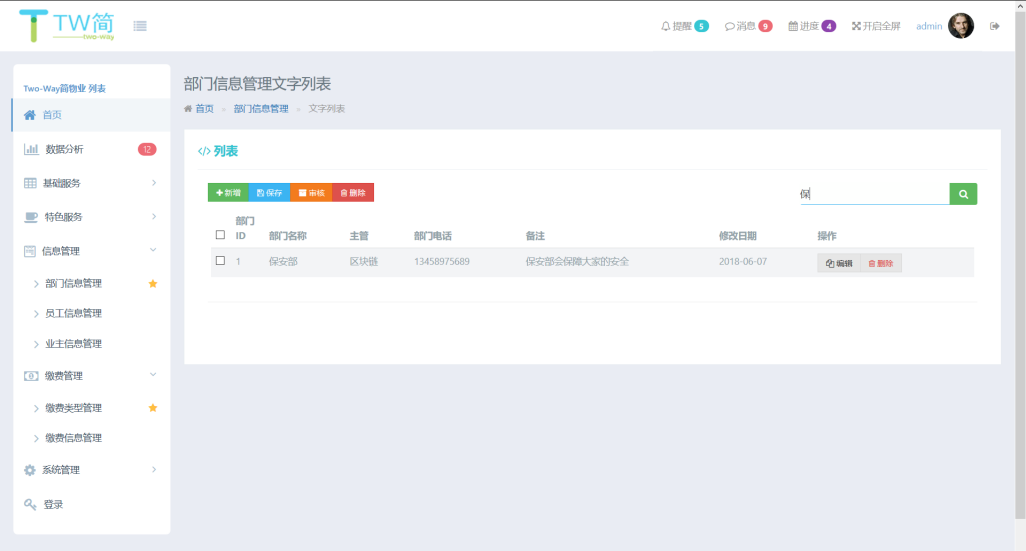
PC端部门信息管理主页面

****

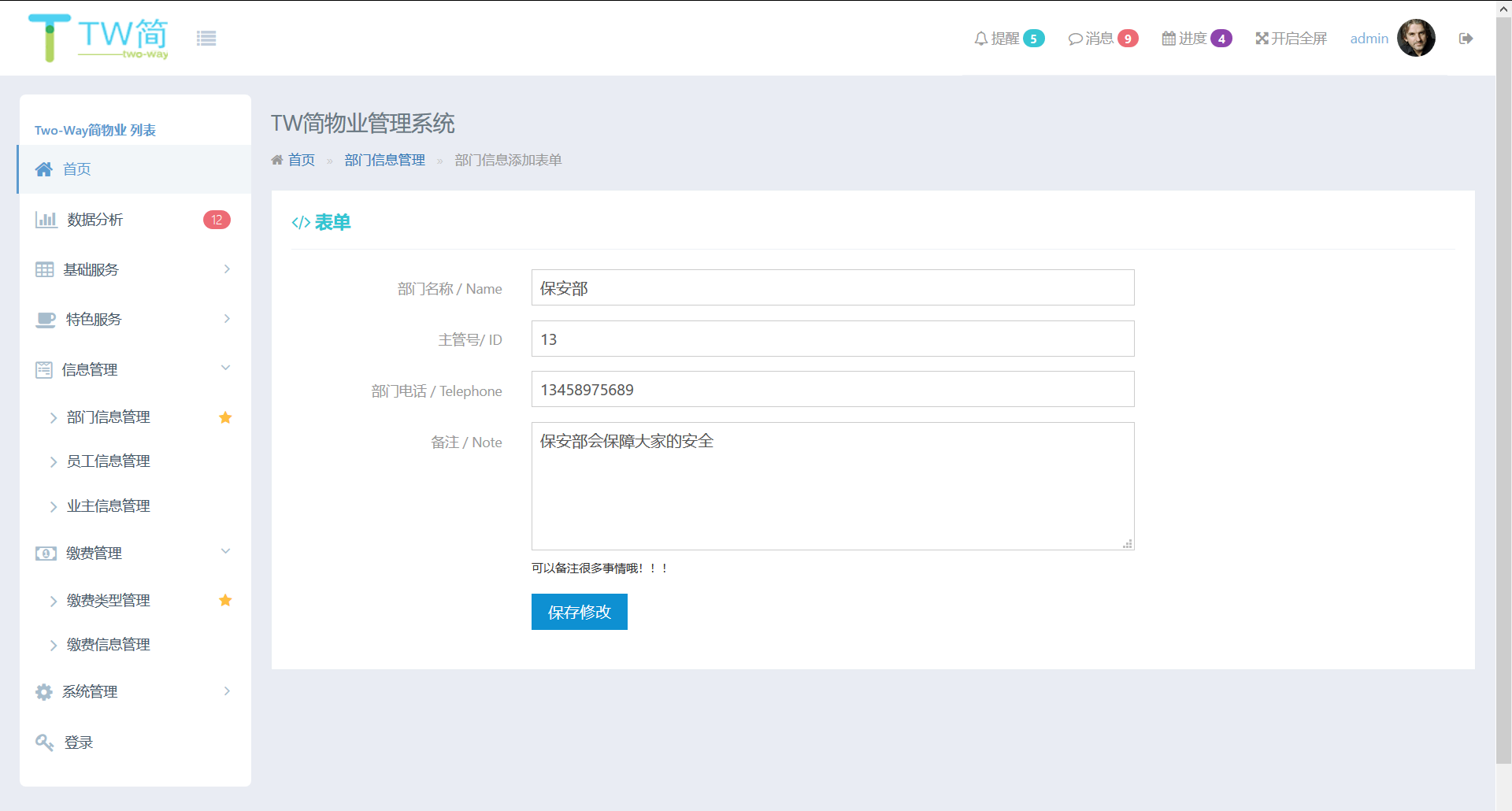
PC端部门信息管理添加页面

****

PC端部门信息管理搜索页面

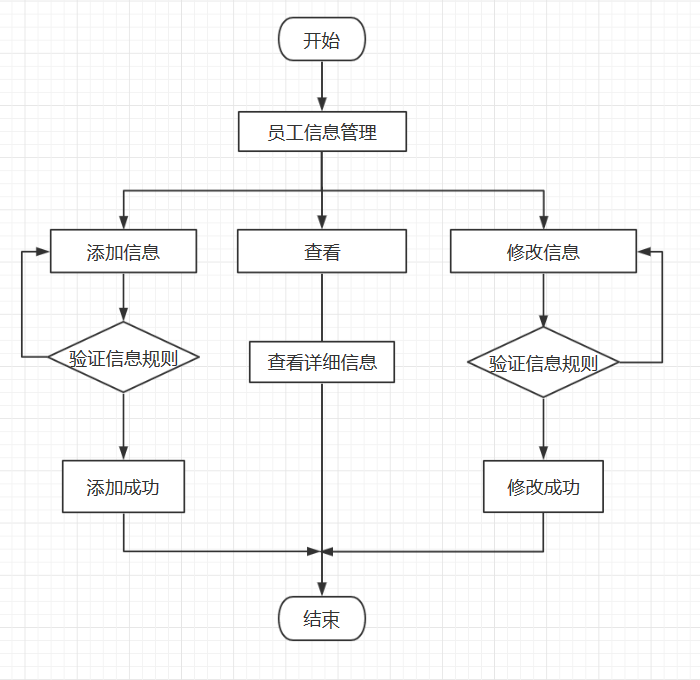
****

PC端部门信息管理修改页面

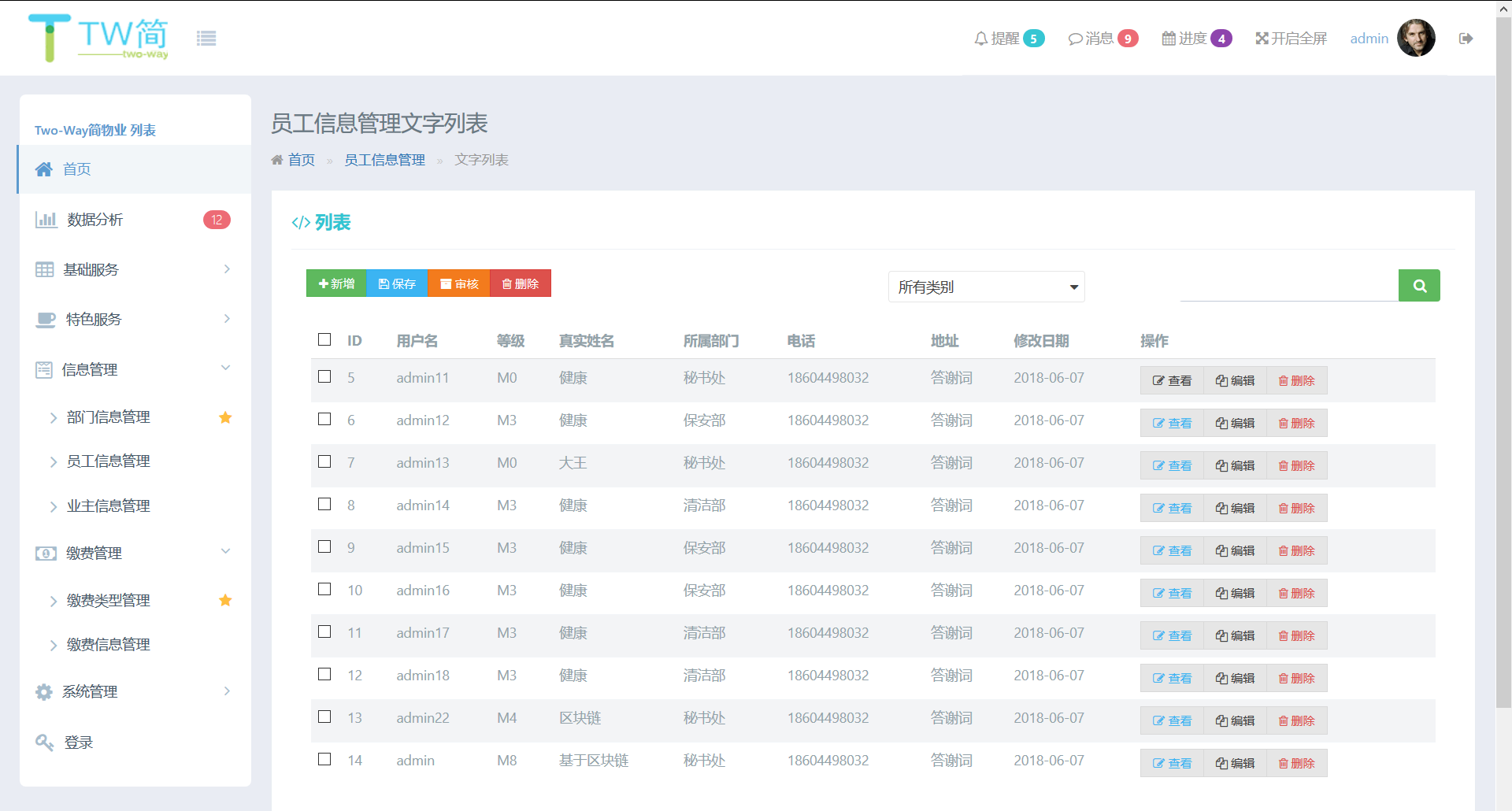


### 5.1.3 员工管理模块

员工管理流程图



PC端员工信息管理主页面



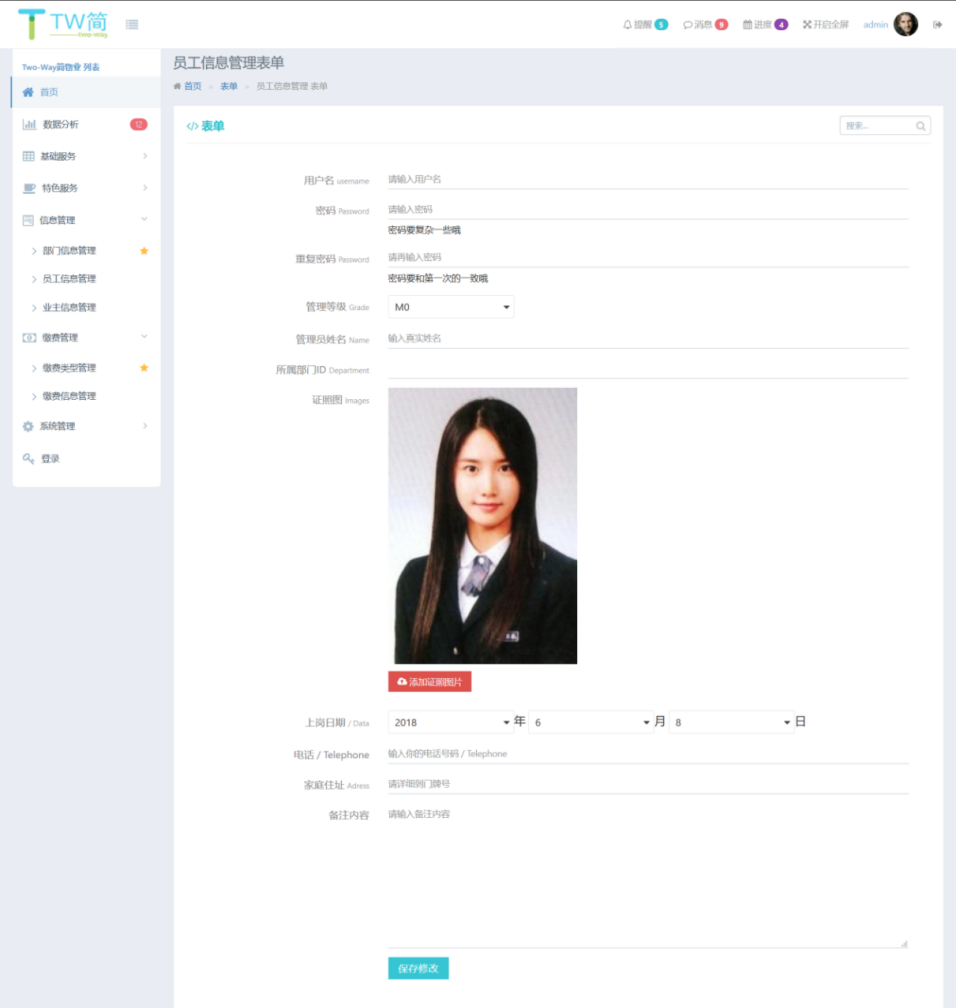
PC端员工信息管理分类页面



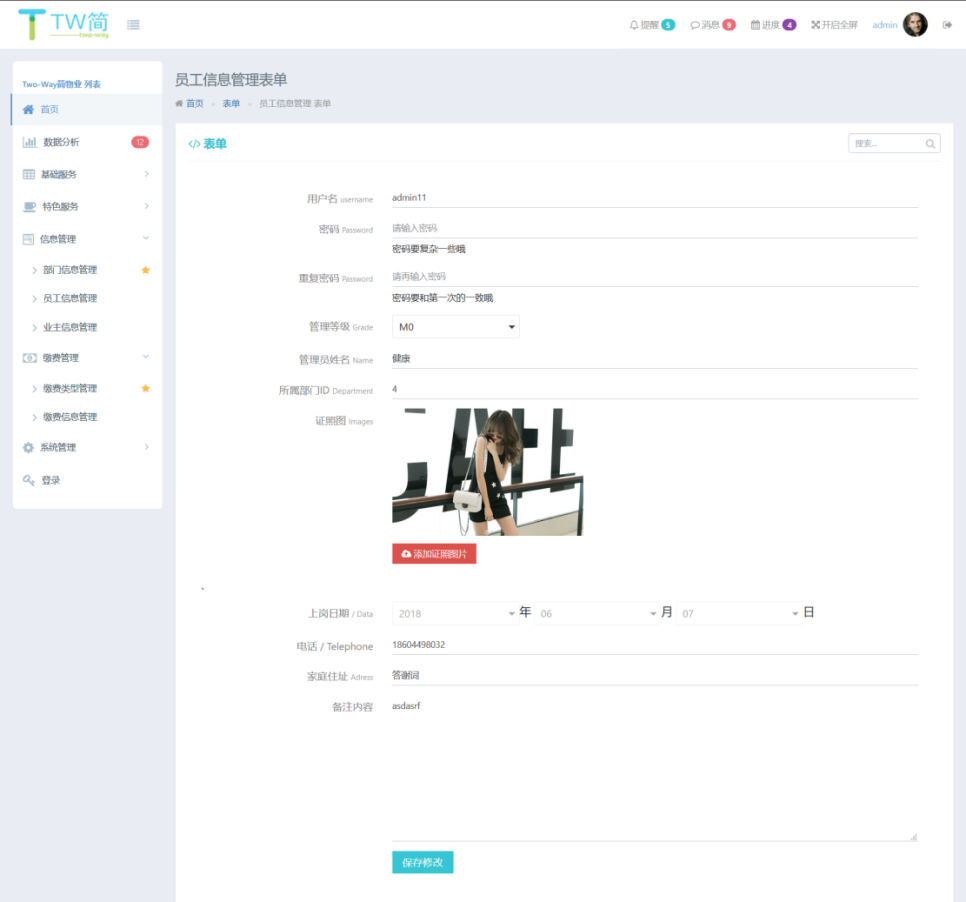
PC端员工信息管理分类+搜索页面



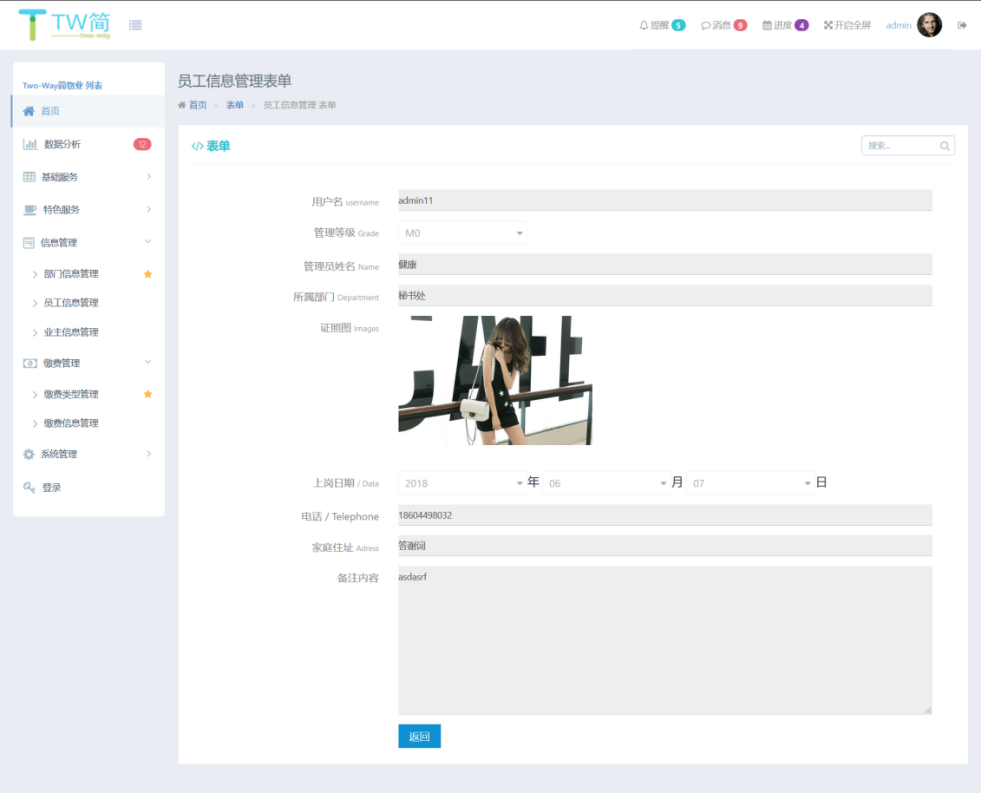
5.1.12 PC端员工信息管理添加页面



PC端员工信息管理修改页面

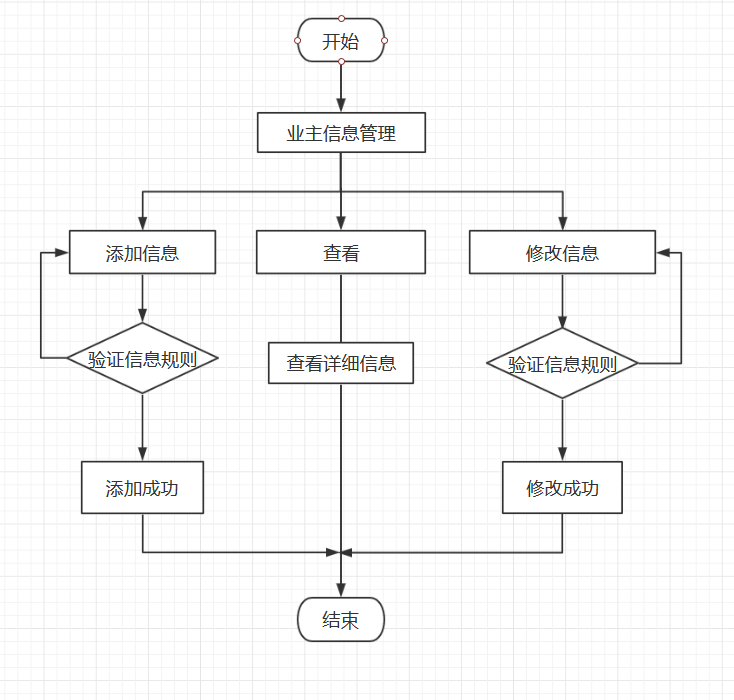


PC端员工信息管理查看页面



### 5.1.4 业主管理模块

业主管理流程图



PC端业主信息管理主页面



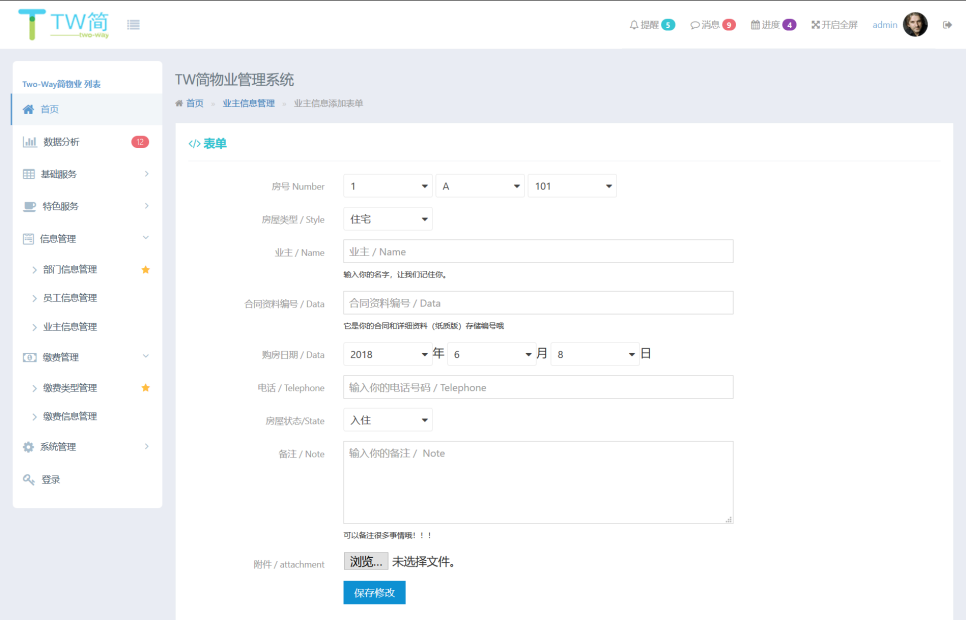
PC端业主信息管理分类页面



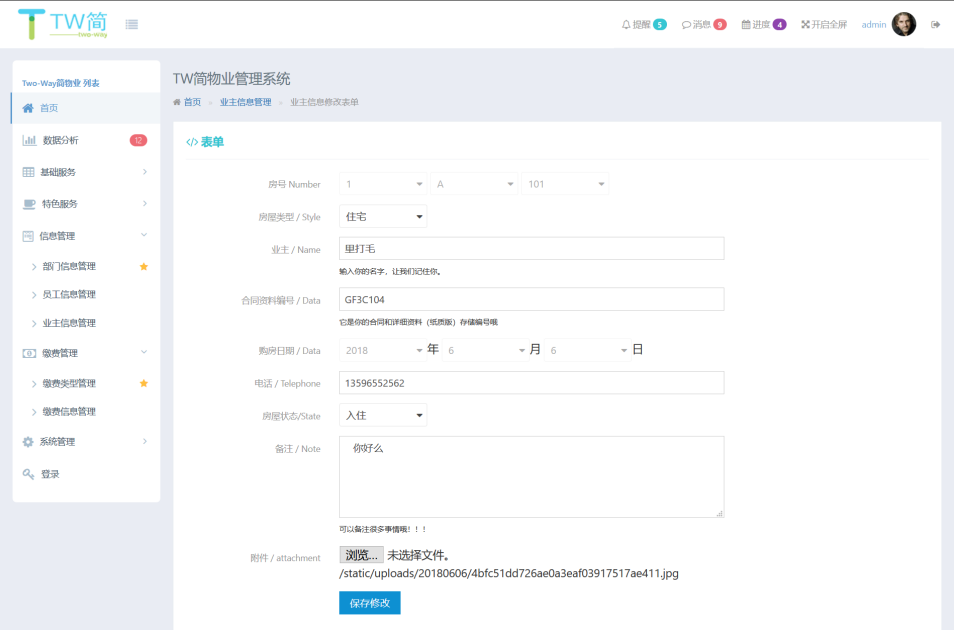
PC端业主信息管理分类+搜索页面



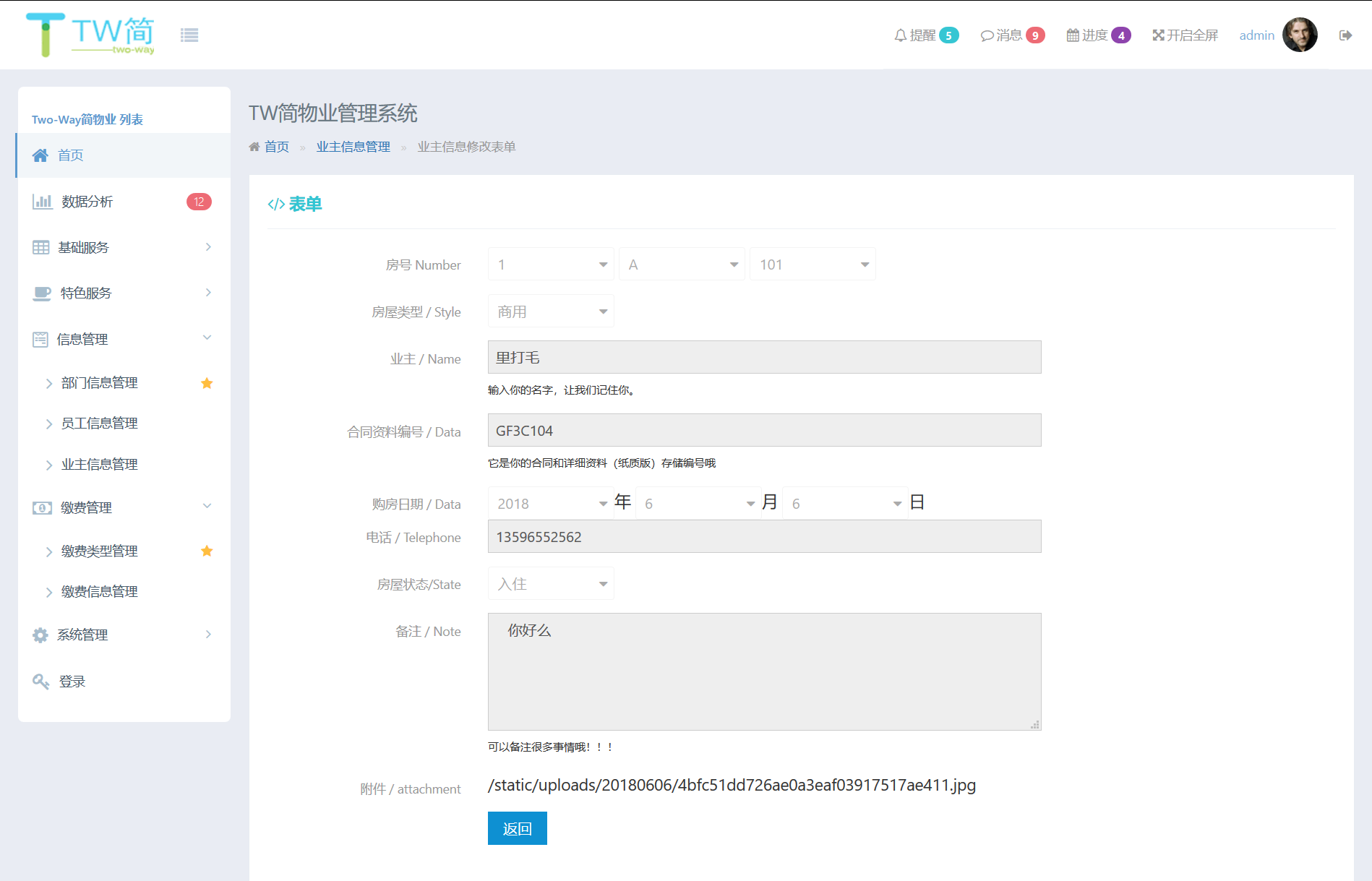
PC端业主信息管理添加页面



PC端业主信息管理修改页面



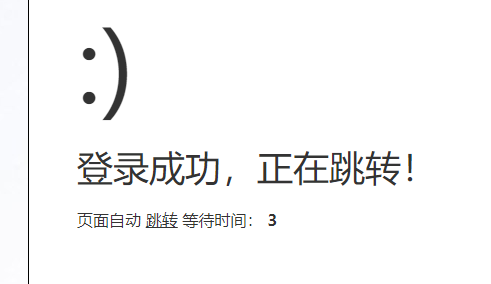
PC端业主信息管理查看页面



# 测试

## 6.1登录测试

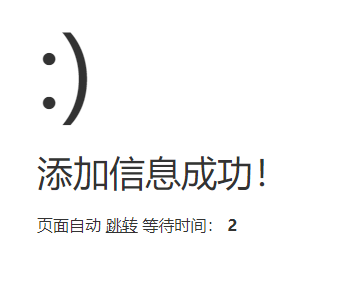
成功



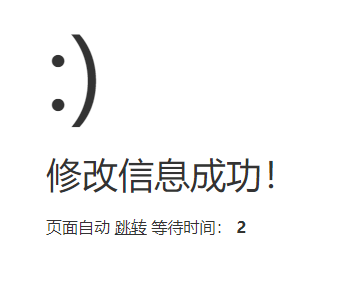
失败



## 6.2添加测试



## 6.3修改测试



## 6.4删除测试



## 6.5验证规则测试

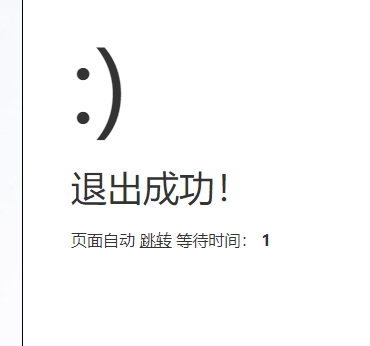






等...

## 6.6退出测试



# 总结

在为期两周的大作业设计实现过程中，组内成员均表现很不错。大家在繁忙的课程和复习之余为小组共同作业付出了许多时间与精力。做大作业的过程既是一份宝贵的学习简单数据库设计的经验，又是大家对于数据库课程查漏补缺的好机会。大家在认真踏实做好自己分内任务的同时，积极参与到别的模块当中，一起合作，讨论，分享自己的资料和心得，彼此监督，均出色完成了此次作业任务。

小组成员一直在积极做着最后的修改。虽然临近期末，但每天晚上项目小组的群里大家总在热烈讨论。短短的时间，大家从毫无头绪到最后做出一个简单雏形，对每一个小组成员都是巨大的收获和成就。这一段过程，也是每一个人的宝贵记忆。

# 附录

## 8.1 部门

### 8.1.1 部门控制器类

<?php

namespace app\admin\controller;

use think\Db;

class Department extends Base

{

public function Index()

{

if (request()->isPost()) {

$find=input('find');

$adminres = \think\Db::name('bn')->where('bn\_name','like',"%".$find."%")->paginate(10);

//相似查询

}else{

$adminres = \think\Db::name('bn')->paginate(10);

}

$this->assign('adminres',$adminres);//模版赋值

return $this->fetch('lst');//输出视图

}

//添加数据方法

public function Add()

{

if (request()->isPost()) {

$list=Db::name('gl')->where('gl\_id',input('gl\_id'))->find();

if(!$list){

return $this->error('不存在此管理人员');

}

$data=[

'bn\_name'=>input('bn\_name'),

'gl\_id'=>input('gl\_id'),

'gl\_name'=>$list['gl\_name'],

'gl\_p'=>$list['gl\_p'],

'bn\_phone'=>input('bn\_phone'),

'bn\_note'=>input('bn\_note'),

'time'=>time(),

]; //打包成数据数组

//验证器

$validate = \think\Loader::validate('department');//连接validate中的information.php

if ($validate ->check($data)) {

$db= \think\Db::name('bn')->insert($data);//将数组添加到数据库中

/\*

使用sql语句写法

$db=insert into (field1,field2) values(value1,value2) ;

\*/

//添加数据验证模块

if ($db) {

return $this->success('添加信息成功！','index');

}else{

return $this->error('添加信息失败！');

}

}else{

return $this->error( $validate->getError());//报错输入与验证规则不符

}

return;

}

return view('add');//添加视图模块

}

//修改数据方法

public function Edit()

{

$id = input('id');

$lst=Db::name('bn')->where('bn\_id',$id)->find();

$this->assign('lst',$lst);

if (request()->isPost()) {

$list=Db::name('gl')->where('gl\_id',input('gl\_id'))->find();

if(!$list){

return $this->error('不存在此管理人员');

}

$data=[

'bn\_name'=>input('bn\_name'),

'gl\_id'=>input('gl\_id'),

'gl\_name'=>$list['gl\_name'],

'gl\_p'=>$list['gl\_p'],

'bn\_phone'=>input('bn\_phone'),

'bn\_note'=>input('bn\_note'),

'time'=>time(),

]; //打包成数据数组

//添加附件模块

//验证器

$validate = \think\Loader::validate('department');

if ($validate ->check($data)) {

//更新数据语句

$db= \think\Db::table('bn')->where('bn\_id', $id)->update($data);

/\*

使用sql语句写法

$list=update table1 set $data where id=$id;

\*/

if ($db) {

return $this->success('修改信息成功！','index');

}else{

return $this->error('修改信息失败！');

}

}else{

return $this->error( $validate->getError());

}

return;

}

return view('edit');

}

//删除数据方法

public function del()

{

$id = input('id');

//删除数据语句

$db=Db::table('bn')->delete($id);

/\*

使用sql语句写法

$db=delete from bn where id=$id;

\*/

if ($db) {

$this->success('删除成功');

}else{

$this->error('删除失败');

}

}

}

### 8.1.2 部门验证器类

<?php

namespace app\admin\validate;

use think\Validate;

class Department extends Validate

{ //验证规则

protected $rule = [

'bn\_name' => 'require|max:100',

'gl\_id' => 'require|max:10',

'bn\_phone' => 'require',

];

//验证提示信息

protected $message = [

'bn\_name.require' => '部门名称不能为空',

'bn\_name.max' => '部门名称不能多于十位',

'gl\_id.require' => '主管ID不能为空',

'gl\_id.max' => '主管ID不能多于十位',

'bn\_phone.require' => '部门电话号码不能为空',

];

}

### 8.1.2 部门视图

略

## 8.2 员工

### 8.2.1 员工控制器类

<?php

namespace app\admin\controller;

use think\Db;

class Worker extends Base

{

public function Index()

{

if (request()->isPost()) {

$find=input('find');

$style=input('style');

//搜索模块

if($find&&$style==1){

$gl=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_user,a.gl\_id,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->where('a.gl\_name','like',"%".$find."%")->paginate(10);

}elseif (!$find&&$style&&$style!=1) {

$gl=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_user,a.gl\_id,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->where('a.gl\_p',$style)->paginate(10);

}elseif ($find&&$style) {

$gl=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_user,a.gl\_id,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->where('a.gl\_name','like',"%".$find."%")->where('a.gl\_p',$style)->paginate(10);

}else{

$gl=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_user,a.gl\_id,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->paginate(10);

//联表查询

}

//从gl表中获取数据并分页每页10条数据

// $gl = \think\Db::name('gl')->paginate(10);

/\*

使用sql语句写法

$gl=select \* from gl ;

联表查询

$gl=select a.title,b.type\_name

from news\_table as a,name\_table as b

where a.type\_id=b.type\_id ;

\*/

}else{

$adminres=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_user,a.gl\_id,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->paginate(10);

}

$this->assign('adminres',$adminres);//模版赋值

return $this->fetch('lst');//输出视图

}

//添加数据方法

public function Add()

{

if (request()->isPost()) {

$list=Db::name('bn')->where('bn\_id',input('bn\_id'))->find();

if (!$list) {

return $this->error('部门号不存在');

}

$data=[

'gl\_user'=>input('username'),

'gl\_pwd'=>MD5(input('password')),

'gl\_p'=>input('grade'),

'gl\_name'=>input('name'),

'bn\_id'=>input('bn\_id'),

'gl\_date'=>(input('gl\_date\_y')."-".input('gl\_date\_m')."-".input('gl\_date\_d')),

'gl\_call'=>input('phone'),

'gl\_adress'=>input('adress'),

'gl\_note'=>input('note'),

'time'=>time(),

]; //打包成数据数组

if ($\_FILES['file']['tmp\_name']){

$file = request()->file('file');

// 移动到框架应用根目录/public/uploads/ 目录下

$info = $file->move(ROOT\_PATH . 'public' . DS . '/static/uploads/');

if($info){

// 成功上传后 获取上传信息

// 输出 jpg

$data['gl\_picture']='/static/uploads/'.date('Ymd').'/'.$info->getFilename();

}else{

// 上传失败获取错误信息

return $this->error($file->getError());

}

}else{

$data['gl\_picture']='/static/uploads/14.jpg';

}

//验证器

if (input('password')!=input('password2')) {

return $this->error('两次密码不一致！');

}

$validate = \think\Loader::validate('Worker');//连接validate中的Worker.php

if ($validate ->check($data)) {

$db= \think\Db::name('gl')->insert($data);//将数组添加到数据库中

/\*

使用sql语句写法

$db=insert into (field1,field2) values(value1,value2) ;

\*/

//添加数据验证模块

if ($db) {

return $this->success('添加信息成功！','index');

}else{

return $this->error('添加信息失败！');

}

}else{

return $this->error( $validate->getError());//报错输入与验证规则不符

}

return;

}

return view('add');//添加视图模块

}

public function Edit()

{

$id = input('id');

$tab=Db::name('gl')->where('gl\_id',$id)->find();

/\*

使用sql语句写法

$tab=select \* from gl where id=$id;

\*/$tab['gl\_date']= explode("-", $tab['gl\_date']);

$this->assign('tab',$tab);

if (request()->isPost()) {

$list=Db::name('bn')->where('bn\_id',input('bn\_id'))->find();

if (!$list) {

return $this->error('部门号不存在');

}

$data=[

'gl\_user'=>input('username'),

'gl\_pwd'=>MD5(input('password')),

'gl\_p'=>input('grade'),

'gl\_name'=>input('name'),

'bn\_id'=>input('bn\_id'),

'gl\_call'=>input('phone'),

'gl\_adress'=>input('adress'),

'gl\_note'=>input('note'),

'time'=>time(),

]; //打包成数据数组

if ($\_FILES['file']['tmp\_name']){

$file = request()->file('file');

// 移动到框架应用根目录/public/uploads/ 目录下

$info = $file->move(ROOT\_PATH . 'public' . DS . '/static/uploads/');

if($info){

// 成功上传后 获取上传信息

// 输出 jpg

$data['gl\_picture']='/static/uploads/'.date('Ymd').'/'.$info->getFilename();

}else{

// 上传失败获取错误信息

return $this->error($file->getError());

}

}

//验证器

if (input('password')!=input('password2')) {

return $this->error('两次密码不一致！');

}

$validate = \think\Loader::validate('Worker');//连接validate中的Worker.php

if ($validate ->scene('edit')->check($data)) {

$db= \think\Db::name('gl')->where('gl\_id', $id)->update($data);//将数组添加到数据库中

/\*

使用sql语句写法

$db=insert into (field1,field2) values(value1,value2) ;

\*/

//添加数据验证模块

if ($db) {

return $this->success('修改信息成功！','index');

}else{

return $this->error('修改信息失败！');

}

}else{

return $this->error( $validate->getError());//报错输入与验证规则不符

}

return;

}

return view('edit');

}

//查看数据方法

public function See()

{

$id = input('id');

//查找数据语句

$line=\think\Db::name('gl')->alias('a')->join('bn c','c.bn\_id =a.bn\_id','LEFT')->field('a.gl\_id,a.gl\_user,a.gl\_pwd,a.gl\_p,a.gl\_name,a.gl\_date,a.gl\_call,a.gl\_picture,a.gl\_adress,a.gl\_note,a.time,bn.bn\_name')->where('a.gl\_id',$id)->find();

/\*

使用sql语句写法

$line=select \* from gl where id=$id;

\*/$line['gl\_date']= explode("-", $line['gl\_date']);

$this->assign('line',$line);

return view('see');

}

// 删除语句方法

public function del()

{

$id = input('id');

//删除数据语句

$db=Db::table('gl')->delete($id);

/\*

使用sql语句写法

$db=delete from yz where id=$id;

\*/

if ($db) {

$this->success('删除成功');

}else{

$this->error('删除失败');

}

}

}

### 8.2.2 员工验证器类

<?php

namespace app\admin\validate;

use think\Validate;

class Worker extends Validate

{ //验证规则

protected $rule = [

'gl\_user' => 'require|max:100|unique:gl',

'gl\_pwd' => 'require|min:5',

'gl\_name' => 'require',

];

//验证提示信息

protected $message = [

'gl\_user.require' => '用户名不能为空',

'gl\_user.max' => '用户名不能多于十位',

'gl\_user.unique' => '用户名不能重复',

'gl\_name.require' => '管理员姓名不能为空',

'gl\_pwd.min' => '密码不能少于5位',

];

protected $scene = [

'edit' => ['gl\_pwd','gl\_name'],

];

}

### 8.2.3 员工视图

略

### 8.2.4 登录模版类

<?php

namespace app\admin\model;

use think\Model;

use think\Db;

class Login extends Model//打造验证继承类

{

public function login($username,$password){//打造登录验证函数

$user=Db::table('gl')->where('gl\_user','=',$username)->find();//判断用户名是否存在

if ($user) {

if ($user['gl\_pwd']==MD5($password)) {//判断密码是否正确

\think\Session::set('id',$user['gl\_id']);//将ID存储到session中

\think\Session::set('username',$user['gl\_user']);//将用户名存储到session中

return 1;//成功登录

}

else{

return -2;//返回-2表示密码不正确

}

}else{

return -1;//返回-1表示用户名不存在

}

}

}

## 8.3 业主

### 8.3.1 业主控制器类

<?php

namespace app\admin\controller;

use think\Db;

class Information extends Base

{

public function Index()

{

if (request()->isPost()) {

$find=input('find');

$style=input('style');

//搜索模块

if($find&&$style==1){

$adminres = \think\Db::name('yz')->where('yz\_name','like',"%".$find."%")->paginate(10);

}elseif (!$find&&$style&&$style!=1) {

$adminres = \think\Db::name('yz')->where('fw\_style',$style)->paginate(10);

}elseif ($find&&$style) {

$adminres = \think\Db::name('yz')->where('yz\_name','like',"%".$find."%")->where('fw\_style',$style)->paginate(10);

}else{

$adminres = \think\Db::name('yz')->paginate(10);

}

//从yz表中获取数据并分页每页10条数据

// $adminres = \think\Db::name('yz')->paginate(10);

/\*

使用sql语句写法

$adminres=select \* from yz ;

\*/

}else{

$adminres = \think\Db::name('yz')->paginate(10);

}

$this->assign('adminres',$adminres);//模版赋值

return $this->fetch('lst');//输出视图

}

//添加数据方法

public function Add()

{

if (request()->isPost()) {

$data=[

'fw\_number'=>(input('home\_number\_1')."-".input('home\_number\_2')."-".input('home\_number\_3')),

'fw\_style'=>input('home\_style'),

'yz\_name'=>input('home\_name'),

'fw\_detailed'=>input('home\_data'),

'gf\_date'=>(input('home\_date\_y')."-".input('home\_date\_m')."-".input('home\_date\_d')),

'yz\_phone'=>input('home\_phone'),

'fw\_state'=>input('home\_state'),

'fw\_note'=>input('home\_note'),

'time'=>time(),

]; //打包成数据数组

//添加附件模块

if ($\_FILES['file']['tmp\_name']){

$file = request()->file('file');

// 移动到框架应用根目录/public/uploads/ 目录下

$info = $file->move(ROOT\_PATH . 'public' . DS . '/static/uploads/');

if($info){

// 成功上传后 获取上传信息

// 输出 jpg

$data['yz\_fjs']='/static/uploads/'.date('Ymd').'/'.$info->getFilename();

}else{

// 上传失败获取错误信息

return $this->error($file->getError());

}

}

//验证器

$validate = \think\Loader::validate('information');//连接validate中的information.php

if ($validate ->check($data)) {

$db= \think\Db::name('yz')->insert($data);//将数组添加到数据库中

/\*

使用sql语句写法

$db=insert into (field1,field2) values(value1,value2) ;

\*/

//添加数据验证模块

if ($db) {

return $this->success('添加信息成功！','index');

}else{

return $this->error('添加信息失败！');

}

}else{

return $this->error( $validate->getError());//报错输入与验证规则不符

}

return;

}

return view('add');//添加视图模块

}

//修改数据方法

public function Edit()

{

$id = input('id');

$lst=Db::name('yz')->where('ww\_id',$id)->find();

$lst['fw\_number']= explode("-", $lst['fw\_number']);//字符串分割（房号）

$lst['gf\_date']= explode("-", $lst['gf\_date']);//字符串分割（购房日期）

$this->assign('lst',$lst);

if (request()->isPost()) {

$data=[

'fw\_style'=>input('home\_style'),

'fw\_detailed'=>input('home\_data'),

'yz\_name'=>input('home\_name'),

'yz\_phone'=>input('home\_phone'),

'fw\_state'=>input('home\_state'),

'fw\_note'=>input('home\_note'),

'time'=>time(),

]; //打包成数据数组

//添加附件模块

if ($\_FILES['file']['tmp\_name']){

$file = request()->file('file');

// 移动到框架应用根目录/public/uploads/ 目录下

$info = $file->move(ROOT\_PATH . 'public' . DS . '/static/uploads/');

if($info){

// 成功上传后 获取上传信息

// 输出 jpg

$data['yz\_fjs']='/static/uploads/'.date('Ymd').'/'.$info->getFilename();

}else{

// 上传失败获取错误信息

return $this->error($file->getError());

}

}

//验证器

$validate = \think\Loader::validate('information');

if ($validate ->check($data)) {

//更新数据语句

$db= \think\Db::table('yz')->where('ww\_id', $id)->update($data);

/\*

使用sql语句写法

$list=update table1 set $data where id=$id;

\*/

if ($db) {

return $this->success('修改信息成功！','index');

}else{

return $this->error('修改信息失败！');

}

}else{

return $this->error( $validate->getError());

}

return;

}

return view('edit');

}

//查看数据方法

public function See()

{

$id = input('id');

//查找数据语句

$list=Db::name('yz')->where('ww\_id',$id)->find();

/\*

使用sql语句写法

$list=select \* from yz where id=$id;

\*/

$list['fw\_number']= explode("-", $list['fw\_number']);

$list['gf\_date']= explode("-", $list['gf\_date']);

$this->assign('list',$list);

return view('see');

}

//删除数据方法

public function del()

{

$id = input('id');

//删除数据语句

$db=Db::table('yz')->delete($id);

/\*

使用sql语句写法

$db=delete from yz where id=$id;

\*/

if ($db) {

$this->success('删除成功');

}else{

$this->error('删除失败');

}

}

}

### 8.3.2 业主验证器类

<?php

namespace app\admin\validate;

use think\Validate;

class Information extends Validate

{ //验证规则

protected $rule = [

'yz\_name' => 'require|max:100',

'fw\_detailed' => 'require|max:10',

'yz\_phone' => 'require',

];

//验证提示信息

protected $message = [

'yz\_name.require' => '业主姓名不能为空',

'yz\_name.max' => '业主姓名不能多于十位',

'fw\_detailed.require' => '资料存储编号不能为空',

'fw\_detailed.max' => '资料存储编号不能多于十位',

'yz\_phone.require' => '电话号码不能为空',

];

}

### 8.3.2 业主视图

略

参考文献：

1. 王珊、萨师煊 《数据库系统概论》（第5版）高等教育出版社 2014年
2. 严蔚敏 《数据结构》 清华大学出版社 2007年
3. 钱雪忠 《数据库原理及应用》（第2版）北京邮电大学出版社 2007年
4. 潘凯华 Php开发1200例清华大学出版社 2016年
5. Thinkphp5完全开发手册 手册网 2017年