

毕业设计（论文）

题目名称：同城独生子女联盟网站建设——系统管理

院系名称：计算机学院

班级：软件162

学号：201608040235

学生姓名：王斌

指导教师：张西广

2020年5月

论文编号：201608040235

同城独生子女联盟网站建设

—系统管理

**Website construction of one child alliance in the same city**

**—System Management**

院系名称：计算机学院

班级：软件162班

学号：201608040235

学生姓名：王斌

指导教师：张西广

2020年5月

摘要

目前我国正逐渐步入老龄化社会。据报道，2019年全球人口老龄化问题加剧，中国人口老龄化增速世界第一。所谓人口老龄化是指人口生育率降低和人均寿命延长导致的总人口中因年轻人口数量减少、年长人口数量增加而导致的老年人口比例相应增长的动态。当一个国家或地区60岁以上老年人口占人口总数的10%，或65岁以上老年人口占人口总数的7%，即意味着这个国家或地区的人口处于老龄化社会。预计2030年，中国将成为全球人口老龄化程度最高国家，2050年，社会将进入深度老龄化阶段，60岁以上人口占比超30%。

面对人口老龄化的难题，我们应该深思，特别是随着我国第一代独生子女大多已进入而立之年，421家庭模式（即四个老人、一对夫妻、一个孩子）开始呈现出主流倾向，且随着国家对二胎政策的全面放开，422家庭模式也会逐渐多起来，四个老人两个孩子的压力是对中间层的独生子女们提出更大的挑战。而这种“倒金字塔”的家庭结构，也衍生出一些现实问题：身处“上有老、下有小”的中间层的独生子女们在承受巨大压力的情况下，如何借助一种有效的机制来更好的为家庭和社会服务。如果能提供一种途径，让独生子女之间通过志愿者联盟互帮互助形式的集体力量来减少面临的家庭现实问题是具有重要意义的。基于此，一个网络化、区域化、智能化和具有良好用户交互性能的独生子女联盟平台是有价值的，借助这个平台，处在同一区域的独生子女们之间互帮补助，共同面对和解决他们所面临的问题，是十分理想和极具社会人文关怀的事情。

该系统实现的功能主要分为三大模块，分别是系统管理、家庭乐园、爱心公益三部分；系统还应开发移动端，该功能可根据具体情况分解到三个模块。系统任务可由3位技术人员完成，其中系统需求分析、系统架构设计、数据库设计等任务由大家共同协商完成，系统实现阶段任务大致分工如下：系统管理部分由1人负责，家庭乐园部分由1人负责，爱心公益部分由1人负责。系统开发过程采用项目组长负责制，系统中其它未明确分工的模块功能由项目组长根据项目开发过程中组员的工作量情况给予动态分配和调整。

按照软件工程规范描述系统管理部分的需求，细化用例规约，合理设计数据库，实现系统管理部分以下主要功能：

1、登录注册：凡是网站注册用户登陆后方可使用系统相关功能，登陆联盟平台需要通过输入用户名、密码、验证码等环节要求。

2、用户管理：系统用户可大致分为浏览用户、独生子女用户、系统管理员等，用户种类的扩展可以在项目推进过程中根据需要完善，该部分涉及到用户的权限管理和用户信息的维护等。

3、新闻管理：管理员负责对网站新闻进行管理，可以对新闻稿进行常规的的增删改查以及置顶等管理操作。

4：通告管理：通知通告是一般网站系统都具备的功能模块，和新闻管理一样，管理员负责对网站通知信息进行管理，主要是对网站通知通告进行常规的的增删改查以及置顶等管理操作。

5：交流园地：该部分类似于一个联盟用户的论坛，联盟用户可以在交流园地模块进行交流沟通。

6：统计分析：该模块可以对系统中相关数据进行一些常规统计，比如说帮扶活动按年月统计、联盟用户参与活动率统计等，具体统计细节，可在系统设计分析时扩展。

7：城市管理：针对有独生子女联盟的城市进行管理。

以上仅列出该模块的主要功能，其它需求可随项目组完成情况进行相应动态调整补充。

关键词：MVC；Spring Boot；Element-ui; Jpa;Vue；Mysql

Abstract

At present, China is gradually stepping into an aging society. It is reported that in 2019, the global population aging problem will be intensified, and China's population aging growth rate will be the first in the world. The so-called population aging refers to the dynamic growth of the proportion of the elderly population in the total population due to the decrease of the number of young population and the increase of the number of old population caused by the decrease of population fertility and the increase of life expectancy. When a country or region's population aged over 60 accounts for 10% of the total population, or the population aged over 65 accounts for 7% of the total population, it means that the population of this country or region is in an aging society. It is predicted that in 2030, China will become the country with the highest degree of population aging in the world. In 2050, the society will enter the stage of deep aging, and the proportion of people over 60 years old will exceed 30%.

In the face of the problem of aging population, we should think deeply, especially with the first generation of only children in our country, 421 family model (i.e. four old people, one couple, one child) has begun to show a mainstream trend, and with the overall liberalization of the national policy on second child, 422 family model will gradually increase, the pressure of four old people and two children is to The only children in the middle put forward greater challenges. The family structure of the "inverted pyramid" also gives rise to some practical problems: how to better serve the family and society with the help of an effective mechanism when the only children in the middle tier who are "old at the top and small at the bottom" are under great pressure. If we can provide a way, it is of great significance to reduce the family problems faced by the only child through the collective strength of volunteer alliance in the form of mutual help. Based on this, a networked, regional, intelligent and user interactive one-child alliance platform is valuable. With the help of this platform, only children in the same region can help each other and solve the problems they face together, which is a very ideal and social and humanistic concern.

The functions of the system are mainly divided into three modules, which are system management, family paradise, love and public welfare. The system should also develop a mobile terminal, which can be divided into three modules according to the specific situation. The system tasks can be completed by 3 technicians, among which the tasks of system demand analysis, system architecture design, database design, etc. are completed by everyone through consultation. The tasks in the system realization stage are roughly divided as follows: 1 person is responsible for the system management part, 1 person is responsible for the family Paradise part, and 1 person is responsible for the love and public welfare part. The project team leader responsibility system is adopted in the system development process. Other module functions in the system without clear division of labor are dynamically allocated and adjusted by the project team leader according to the workload of team members in the project development process.

According to the software engineering specification, describe the requirements of the system management part, refine the use case specification, reasonably design the database, and realize the following main functions of the system management part:

1. Login and registration: all registered users of the website can use the relevant functions of the system after login. To log in to the alliance platform, you need to enter the user name, password, verification code and other link requirements.

2. User management: system users can be roughly divided into browsing users, only child users, system administrators, etc. the expansion of user types can be improved as needed in the process of project promotion. This part involves user rights management and user information maintenance, etc.

3. News management: the administrator is responsible for the management of the website news, and can carry out regular addition, deletion, modification and top management operations for the press release.

4: Notice management: notice notice management is a functional module of general website system. Like news management, administrators are responsible for the management of website notice information, mainly for the regular addition, deletion, modification and top management of website notice.

5: Exchange Garden: this part is similar to a forum for alliance users. Alliance users can communicate in the exchange garden module.

6: Statistical analysis: This module can carry out some routine statistics on the relevant data in the system, such as the statistics of assistance activities by month and year, the statistics of the participation rate of alliance users, etc. the specific statistical details can be expanded during the system design and analysis.

7: Urban management: for cities with only child alliance.

The above only lists the main functions of the module. Other requirements can be adjusted and supplemented dynamically according to the completion of the project team.

**Keywords:** MVC；Spring Boot；Element-ui; Jpa;Vue；Mysql

目 录

[第1章引言 1](#_Toc453539339)

[1.1 背景 1](#_Toc453539340)

[1.2 现状及发展趋势 1](#_Toc453539341)

[1.2.1 国外现状 1](#_Toc453539342)

[1.2.2 国内现状 1](#_Toc453539343)

[1.3 研究内容 2](#_Toc453539344)

[1.4 课题意义和目的 2](#_Toc453539345)

[1.5 相关技术 3](#_Toc453539346)

[第2章需求分析 5](#_Toc453539347)

[2.1 总体需求 5](#_Toc453539348)

[2.2 功能需求 6](#_Toc453539349)

[2.2.1 系统角色 6](#_Toc453539350)

[2.2.2 需求描述 7](#_Toc453539351)

[2.3非功能需求 9](#_Toc453539352)

[2.3.1 性能要求 9](#_Toc453539353)

[2.3.2 可扩展性 9](#_Toc453539354)

[2.3.3 可用性 10](#_Toc453539355)

[2.3.4 安全性 10](#_Toc453539356)

[2.3.5 易用性 10](#_Toc453539357)

[2.3.6 开发环境 10](#_Toc453539358)

[2.4 用例规约描述 10](#_Toc453539359)

[2.4.1 用例目的 11](#_Toc453539360)

[2.4.2 用例描述 11](#_Toc453539361)

[第3章系统分析 25](#_Toc453539362)

[3.1 静态模型 25](#_Toc453539363)

[3.1.1 实体类 25](#_Toc453539364)

[3.1.2 边界类 26](#_Toc453539365)

[3.1.3 控制类 27](#_Toc453539366)

[3.2 时序图的分析 28](#_Toc453539367)

[第4章系统设计 35](#_Toc453539368)

[4.1 架构设计 35](#_Toc453539369)

[4.2 类设计 35](#_Toc453539370)

[4.3 数据库表设计 39](#_Toc453539371)

[4.3.1 概念模型 39](#_Toc453539372)

[4.3.2 物理模型 44](#_Toc453539373)

[第5章系统实现与测试 51](#_Toc453539374)

[5.1 通用模块 51](#_Toc453539375)

[5.1.1 工具类 51](#_Toc453539376)

[5.1.2 数据源配置 51](#_Toc453539377)

[5.2 功能模块 52](#_Toc453539378)

[5.2.1 后台登录界面 52](#_Toc453539379)

[5.2.2 后台主页 53](#_Toc453539380)

[5.2.3 个人中心 53](#_Toc453539381)

[5.2.4 系统管理 54](#_Toc453539382)

[5.2.5 会员管理 54](#_Toc453539383)

[5.2.6 车辆管理 55](#_Toc453539384)

[5.2.7 配车管理 55](#_Toc453539385)

[5.2.8 订单管理 56](#_Toc453539386)

[5.2.9 网点管理 56](#_Toc453539387)

[5.2.10 财务管理 57](#_Toc453539388)

[5.2.11 统计管理 57](#_Toc453539389)

[5.3系统运行及测试 58](#_Toc453539390)

[5.3.1 登录测试用例 58](#_Toc453539391)

[5.3.2 添加用户测试用例 58](#_Toc453539392)

[5.3.3 修改用户测试用例 59](#_Toc453539393)

[5.3.4 查询用户测试用例 60](#_Toc453539394)

[5.3.5 删除用户测试用例 61](#_Toc453539395)

[第6章结束语 62](#_Toc453539396)

[致谢 63](#_Toc453539397)

[参考文献 64](#_Toc453539398)

[附录 65](#_Toc453539399)

[附录A: 光盘 65](#_Toc453539400)

第1章 引 言

## 1.1背景

随着时间的推移多年前的计划生育政策所限制的只能生一个孩子的家庭现在的家庭主力已经逐渐转换。在很多90后当中有大批的孩子是独生子或者独生女，而在当下90后已经逐渐成为了社会的主力军。因此大批的人们都涌向了工作岗位。在快节奏的生活中人们都忙于工作，忙于应酬，在好不容易的假期中很多人都选择去放松一下自己，有兄弟姐妹的人自然觉得大家庭其乐融融，而独生子女则显得相对于孤独。随着父辈人的退休赡养老人的重担也搭在了独生子女的肩膀上。如果一个独生子和一个独生女联姻，生一个孩子。这就是典型的421家庭。对两个人的压力十分的大。

本网站为同城的独生子女提供一个交流的平台希望，让彼此可以互相帮助传递社会爱心弘扬社会正能量。

## 1.2现状及发展趋势

### 1.2.1国外现状

对于现阶段独生子女的生活现状首先要从独生子女的出生时说起。

首先是上世纪40年代中到60年代中美国人口出生高峰期的“婴儿潮”时代之后，美国人口变化规律的自然演变。也就是説，“婴儿潮”时代大量出生的人口导致教育和就业的竞争加剧，令许多美国人自觉选择一个孩子。美国人口普查局统计学家汤姆斯说，出生在婴儿潮时代的为人父母者，“自己为了教育和就业机会而推迟生育”也是一个原因。

其次是美国不断上涨的养儿育女经济压力，迫使父母不敢多要孩子。皮尤研究中心调查显示，其实美国只有3%的人认为生一个孩子是理想的选择，认为生两个或三个孩子是理想选择的人占7到8成，但是现实中有更多的人只生一个孩子，主要是觉得养儿育女的经济压力很大。

美国农业部2011年做过估算，在美国养育一个孩子到18岁，平均需要花23.5万美元。在美国，从幼儿园到高中上公立学校是免费的，但孩子5岁之前上托儿所的费用很贵，基本上每月要1000美元以上；到了孩子18岁以后，当父母的又要为孩子上大学的学费而发愁，一个孩子一年上私立大学的学费和生活费用至少要7、8万美元，上本州的公立大学也要4万美元左右，试想如果一个年收入5、6万美元的中产阶级家庭有两三个孩子要连续上大学的话，当父母的压力会有多大？

有研究发现，美国养儿育女最多的家庭在两极，即最穷的人和最富的人，贫困线下的家庭孩子最多，收入在全美“金字塔尖”5%的家庭孩子也不少，但中产阶级实在是不敢多生孩子。关注美国中产阶级负担的“美国养儿意识协会”高级顾问麦克雷迪说：“这不仅是低收入的问题，也是中产阶级的问题，如果你不是富人，这绝对会影响你。“

因此在以美国为首的发达国家中独生子女人数也在猛烈增加。

### 1.2.2国内现状

目前，我国已经开放了二胎政策，很多家庭会选择生两个孩子。而当年的因为计划生育导致一个家庭一个孩子，也会给相关家庭独生子女证书，在某些情况下会有一定的帮扶。

有相关的研究独生子女的生活方式问题。风笑天分析了这种新的社会细胞具有一些特征。作者指出，三口之家是独身子女家庭在规模上的主要特征，这种“三个人的世界“包含了绝大部分独生子女家庭在生活方式上的个只能发特征的内涵，也是形成和决定对其他各种生活方式特征的根本因素和内在条件。而与祖辈分而不离、子女成为家庭中心、亲自关系日趋平等，则是独生子女家庭在家庭关系方面的几大特征。此外，集体化的闲暇和娱乐、潮流话、智力化的子女消费也是独生子女家庭在生活方式上的明显特征。

## 1.3研究内容

在互联网的快速发展下使用已经十分成熟的框架和架构对独生子女打造一个交流互助平台，本文主要是应用现有的框架spring vue等，以及前后分离的方式对系统进行管理。

主要内容包括：

1、登录注册：凡是网站注册用户登陆后方可使用系统相关功能，登陆联盟平台需要通过输入用户名、密码、验证码等环节要求。

2、用户管理：系统用户可大致分为浏览用户、独生子女用户、系统管理员等，用户种类的扩展可以在项目推进过程中根据需要完善，该部分涉及到用户的权限管理和用户信息的维护等。

3、新闻管理：管理员负责对网站新闻进行管理，可以对新闻稿进行常规的的增删改查以及置顶等管理操作。

4：通告管理：通知通告是一般网站系统都具备的功能模块，和新闻管理一样，管理员负责对网站通知信息进行管理，主要是对网站通知通告进行常规的的增删改查以及置顶等管理操作。

5：交流园地：该部分类似于一个联盟用户的论坛，联盟用户可以在交流园地模块进行交流沟通。

6：统计分析：该模块可以对系统中相关数据进行一些常规统计，比如说帮扶活动按年月统计、联盟用户参与活动率统计等，具体统计细节，可在系统设计分析时扩展。

7：城市管理：针对有独生子女联盟的城市进行管理。

## 1.4课题意义和目的

随着目前我国正逐渐步入老龄化社会。据报道，2019年全球人口老龄化问题加剧，中国人口老龄化增速世界第一。所谓人口老龄化是指人口生育率降低和人均寿命延长导致的总人口中因年轻人口数量减少、年长人口数量增加而导致的老年人口比例相应增长的动态。当一个国家或地区60岁以上老年人口占人口总数的10%，或65岁以上老年人口占人口总数的7%，即意味着这个国家或地区的人口处于老龄化社会。预计2030年，中国将成为全球人口老龄化程度最高国家，2050年，社会将进入深度老龄化阶段，60岁以上人口占比超30%。

面对人口老龄化的难题，我们应该深思，特别是随着我国第一代独生子女大多已进入而立之年，421家庭模式（即四个老人、一对夫妻、一个孩子）开始呈现出主流倾向，且随着国家对二胎政策的全面放开，422家庭模式也会逐渐多起来，四个老人两个孩子的压力是对中间层的独生子女们提出更大的挑战。而这种“倒金字塔”的家庭结构，也衍生出一些现实问题：身处“上有老、下有小”的中间层的独生子女们在承受巨大压力的情况下，如何借助一种有效的机制来更好的为家庭和社会服务。如果能提供一种途径，让独生子女之间通过志愿者联盟互帮互助形式的集体力量来减少面临的家庭现实问题是具有重要意义的。基于此，一个网络化、区域化、智能化和具有良好用户交互性能的独生子女联盟平台是有价值的，借助这个平台，处在同一区域的独生子女们之间互帮补助，共同面对和解决他们所面临的问题，是十分理想和极具社会人文关怀的事情。

因此建立该系统的意义有：

1. 为独生子女们打造一个交流空间

因为在众多的交流平台中，什么样的人都有，什么样年龄阶段的人都有。而独生子女这个特殊群体有时候面临的一些问题其他们并没有遇到。很多人并不了解他们的处境，更加不可能感同身受。所以只有独生子女对独生子女的群体才能了解。为他们提供一个良好的交流环境是一件十分温馨的事情。

1. 同城的独生子女互相帮助

在兄弟姐妹多的家庭中照顾老人往往比较都各自有分担，或者在某些情况或者突发情况总会有人能到老人跟前。而独生子女就没有这一优势，因此希望同城独生子女，无论是在照顾老人还是独自生活上能互相帮助。

1. 弘扬社会正能量传递爱心

在这个快节奏的时代，互联网的便捷和快速的特性无疑给社会带来了很多的便利。而在我国这个拥有千年文化的大国，孝敬父母，朋友互助，达则接济天下。这种充满爱心和暖意的事情从来否是值得被尊重的发扬的。正好这个平台就是一个以互助为核心的平台。

## 1.5相关技术

该系统文档描述中所用到的相关技术及其术语说明如表1-1所示：

表格 1-1 系统术语表

|  |  |
| --- | --- |
| 术语 | 说明 |
| 用例图 | 指由参与者（Actor）、[用例](http://baike.baidu.com/view/706238.htm)（Use Case）以及它们之间的关系构成的，用于描述系统功能的静态视图。 |
| 时序图 | 描述对象之间发送消息的时间顺序显示多个对象之间的动态协作，是一种UML交互图.。 |
| 类图 | 显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性信息。 |
| Spring | Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架. |
| jpa | JPA是Java Persistence API的简称，中文名为Java持久层API，是JDK 5.0注解或XML描述对象－关系表的映射关系，并将运行期的实体对象持久化到数据库中。  可以理解为JPA规范的再次封装抽象，底层还是使用了Hibernate的JPA技术实现，引用JPQL（Java Persistence Query Language）查询语言，属于Spring整个生态体系的一部分。随着Spring Boot和Spring Cloud在市场上的流行，Spring Data JPA也逐渐进入大家的视野，它们组成有机的整体，使用起来比较方便，加快了开发的效率，使开发者不需要关心和配置更多的东西，完全可以沉浸在Spring的完整生态标准实现下。JPA上手简单，开发效率高，对对象的支持比较好，又有很大的灵活性，市场的认可度越来越高。 |
| axios技术 | axios 是一个基于Promise 用于浏览器和 nodejs 的 HTTP 客户端，它本身具有以下特征：  从浏览器中创建 XMLHttpRequest  从 node.js 发出 http 请求  支持 Promise API  拦截请求和响应  转换请求和响应数据  取消请求  自动转换JSON数据  客户端支持防止 CSRF/XSRF  是一种异步局部刷新技术 |
| MYSQL | MYSQL是一个关系型数据库管理系统。 |
| JDBC | JDBC（Java Data Base Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。 |
| vue | 是一个构建用户界面的框架  是一个轻量级的MVVM（Model-View-ViewModel），其实就是所谓的数据的双向绑定  数据驱动和组件化的前端开发  通过简单的API就能实现**响应式的数据绑定**和**组合的视图组件** 指令： 用来扩展html标签的功能  vue与其他框架的对比  简单、易学、更轻量  指令v-xxx开头  HTML + JSON数据  开发者 尤雨溪 华人 |
| Spring | Spring是一个开源框架，Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。简单来说，Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。 |
| SringMVC | Spring MVC属于SpringFrameWork的后续产品，Spring MVC 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。 |
| Element ui | Element是（饿了么团队）基于MVVM框架Vue开源出来的一套前端ui组件 |
| npm | npm是node的包管理工具，定义明确就是用来管理node的包，包括安装，卸载，更新，发布等 |
| Node.js | Node.js 是一个基于 Chrome V8 引擎的 [JavaScript](https://baike.baidu.com/item/JavaScript/321142) 运行环境。 Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型。[1]  Node 是一个让 JavaScript 运行在[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF/6492316)的开发平台，它让 JavaScript 成为与[PHP](https://baike.baidu.com/item/PHP/9337)、[Python](https://baike.baidu.com/item/Python/407313)、[Perl](https://baike.baidu.com/item/Perl/851577)、[Ruby](https://baike.baidu.com/item/Ruby/11419) 等服务端语言平起平坐的[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80/1379708)。[2]  发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。  Node对一些特殊用例进行优化，提供替代的[API](https://baike.baidu.com/item/API/10154)，使得V8在非浏览器环境下运行得更好。V8引擎执行Javascript的速度非常快，性能非常好。Node是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node 使用[事件驱动](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8/9597519)， 非阻塞[I/O](https://baike.baidu.com/item/I%2FO/84718) 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。 |
| Es6 | ECMAScript 6（亦称 ES2015，以下简称 ES6）是2015年6月份定稿的第六版 JavaScript 语言规范。ES6 包含了很多新的让 JavaScript 更为强大、生动的语言特性。 |

第2章 需求分析

独生子女联盟网站是一个社会服务型网站。它包含三个较大的子模块，即：系统管理、家庭乐园和爱心公益。本章节将详细讲同城独生子女联盟网站-系统管理的需求分析，采用面向对象的分析方法，使用UML建立系统用例模型，创建系统功能模型。

## 2.1总体需求

同城独生子女联盟网站系统的用户分为两类用户：系统管理员和用户。系统管理员可以登录后台系统在后台管理网站的用户和内容。用户可以登录前端网站，在网站进行交流，浏览；

1. 用户管理：负责查看用户的信息，对用户进行删除和查询管理；
2. 新闻管理：对新闻也信息进行管理，管理包括首页的视频上传，对新闻进行编辑发布对新闻进行删除，首页视频进行上传后可以进行置顶操作，对内容新闻发布后也可以进行置顶操作，将置顶新闻停留在图片轮播图上；
3. 通告管理：对通告进行编辑，对通告进行发布；
4. 统计分析：统计分析对用户人数进行统计，每天的注册人数用条形图统计显示，用折线图显示注册人数的变化趋势；
5. 城市管理：对每个城市的人员进行管理，对每个人员的发表内容进行管理包括删除；
6. **系统管理员**
7. 用户管理：可以在后台查看用户的个人信息，可以删除其账号；
8. 新闻管理：管理员在后台编辑发布新闻，对新闻置顶操作，删除新闻等功能；
9. 通告管理：在后台编辑通告发布通告，
10. 城市管理：查看每个城市的人数分布，每个人发表的内容同时对内容进行管理；
11. 统计分析：对用户人数进行统计，对城市进行统计；
12. **用户**
13. 个人中心：维护自己的个人资料、修改个人密码；
14. 个人空间：完善自己在交流论坛的个人信息，设置论坛的个人头像，发表动态，查看动态以及回复评论；
15. 注册登录：填写自己的手机号回收到验证码进行注册登录；

## 2.2功能需求

对于独生子女联盟网站平台系统的功能需求用例描述如图2-1，2-2所示：

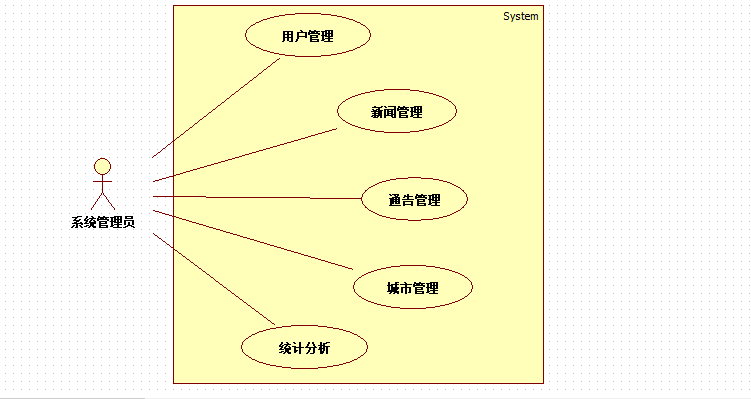


图2-1 系统用例

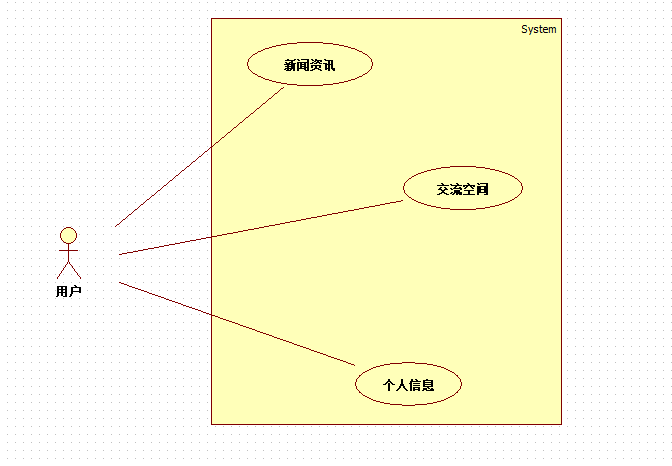


图 2-2 用户用例

### 2.2.1系统角色

系统管理员的功能有：用户管理、新闻管理、浏览用户信息、用户内容查看、城市管理、通告管理、查看统计信息。如图2-3示：

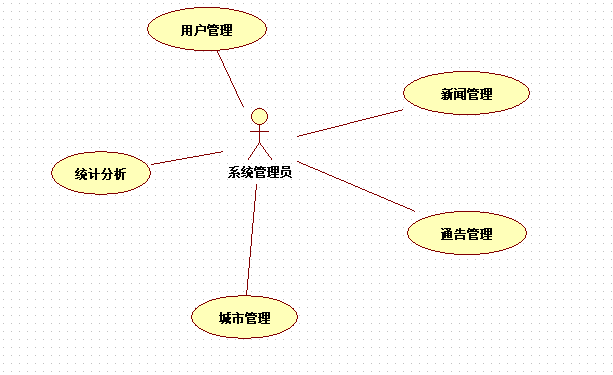


图2-3 管理员功能

系统用户的功能有，个人空间，新闻资讯，个人信息，如图2-4所示

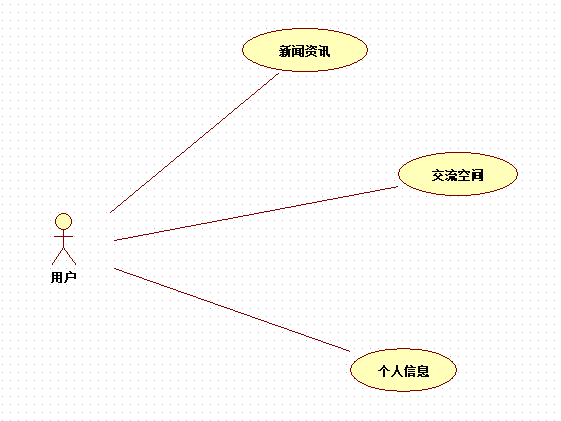


图 2-4 用户部分功能

### 2.2.2 需求描述

1. **登录**

关联角色：系统管理员、用户

需求编号：001

需求描述：系统数据库用户表中的所有用户都可以登录客户端界面，管理员表中的用户都有可以登录系统管理界面。

备注说明：该系统数据库中必须存在用户的登录账号和密码，并且用户的状态是“启用”时，才可进行登录操作。

1. **用户管理**

关联角色：系统管理员

需求编号：002

需求描述：管理员登录后，可以查看用户的信息（性名，年龄，账号，密码等），可以删除用户的账号。

备注说明：登录后，才可进行相关的操作。

1. **新闻管理**

关联角色：系统管理员

需求编号：003

需求描述：管理员登录后，可以对新闻模块的新闻进行编辑，编辑新闻包括图片，可以发布新闻。发不完新闻后对重要的新闻可以进行置顶操作。

备注说明：登录后，才可进行相关的操作。

1. **通告管理**

关联角色：系统管理员

需求编号：004

需求描述：管理员登录后，可以对需要发布的通告进行编辑，然后发布。

备注说明：登录后，才可进行相关的操作。

1. **城市管理**

关联角色：系统管理员

需求编号：005

需求描述：管理员登录后，可以对这个城市二的人进行管理，可以管理这些人发布的内容。可以查看内容，可以删除内容。

备注说明：登录后，才可进行相关的操作。

1. **统计分析**

关联角色：系统管理员

需求编号：006

需求描述：对数据库已有的数据经进行统计显示（图表的形式）。

备注说明：登录后，才可进行相关的操作。

1. **新闻栏目**

关联角色：用户、散客

需求编号：007

需求描述：登录用户和散客都可以对新闻栏目进行浏览，可以观看新闻栏目的视频信息。

备注说明：访问网站的所有人都可以操作。

1. **交流空间**

关联角色：用户

需求编号：008

需求描述：登录的用户可以在交流空间继续进行交流。可以发表自己的内容，可以评论别人发表的内容，可以点赞别人发表的内容，可以diss别人发表的内容，同时可以回复别人回复你的内容。

备注说明：登录后，才可进行相关的操作。

## 2.3非功能需求

### 2.3.1性能要求

1、系统支持多用户并发操作。

2、系统处理操作响应时间不超过10秒。

### 2.3.2可扩展性

系统当前使用的人数不多且功能简单，对于后期使用人数的增多以及功能的修改和增删，都能够做出相应的扩展。

### 2.3.3可用性

保证正常情况下系统可用，可允许系统每月停机一段时间，来进行相关的系统维护。维护时间可安排在恰当的非工作时间进行。

### 2.3.4安全性

1. **操作跟踪**

针对系统运行出现的异常，跟踪调查出现异常的情况，有针对性的解决问题。

系统日志，便于查看系统的运行情况。

操作日志, 提供用户在系统中增加、修改系统数据信息时日志记录，用于跟踪用户的相关操作。

1. **访问控制**

页面不可以直接访问，防止他人对页面进行篡改。页面访问通过连接动作驱动有效防止用户通过地址栏输入地址而对信息非法访问作了权限检查相关操作。

### 2.3.5 易用性

系统简单、易用，具有清晰的导航功能，使用户能快速找到自己想要执行的操作页面。同时，也能保证用户通阅读用户手册，可以使用此系统。

### 2.3.6开发环境

**1. 软件约束**

加盟中心系统采用IDEA和MySQL为工具进行开发。

开发及运行的软件环境为：win10/centos;

开发系统平台：IDEA +Tomcat8.0 + JDK1.8；

浏览器：chrome浏览器或者以上版本、Googole、Firefox等；

数据库：MySQL;  
开发框架：spring boot框架，vue,element-ui。

**2. 硬件约束**

操作系统：32/64位的 Windows 10系统

## 2.4用例规约描述

本章节主要根据《软件需求规格说明书》，来进行的用例建模，并对用例模型进行具体描述。用例规约是面向对象分析和设计的重要步骤。

### 2.4.1 用例目的

用例是从外部角度描述用户和系统之间相互作用的行为，表达系统应该做什么。本文档通过用例规约，来进一步描述该系统需求，它是系统设计的基础，也是测试用例编写的重要依据。

### 2.4.2用例描述

1. **管理员登录**

管理员登录系统后台的用例规约如表2-1所示：

表2-1管理员登录用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员登录 |
| 用例ID | CR-UC-001 |
| 用例说明 | 管理员登录系统 |
| 执行者 | 管理员 |
| 前置条件 | 无 |
| 主成功场景 | 1. 用户访问系统 2. 系统显示登录页面 3. 用户输入用户名和密码，然后单击“登录”按钮 4. 系统验证用户名和密码、身份及用户的状态 5. 系统登录成功显示管理员界面 |
| 扩展1 | （4a）用户名和密码为空  (4a1) 系统弹出提示框“登录失败”，显示当前的登录页。  (4a2)返回主成功场景步骤(3) |
| 后置条件： | 登录成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 只有系统数据库中存在的用户，并且用户状态为“启用”时才可按身份登录该系统。 |

1. **管理员添加新管理员**

管理员登录系统后台进行添加用户操作的用例规约如表2-2所示：

表2-2管理员添加用户用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员添加新管理员 |
| 用例ID | CR-UC-002 |
| 用例说明 | 管理员操作系统管理模块的用户信息---添加管理员 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台添加管理员栏目 2. 输入需要添加管理员的账号和密码 3. 点击“用户信息”界面的“添加”按钮，弹出一个添加用户信息的对话框 4. 如果添加成功会弹出添加成功的提示 5. 系统将信息添加到数据库中的用户表里 |
| 扩展 | （5a）填写用户信息为空  (5a1) 系统无法提交信息更新数据库的用户表，“添加”提示填写完成信息  (5a2)添加成功停留在当前界面有添加成功提示 |
| 后置条件： | 添加用户成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 只有输入的信息不为空，操作才可顺利进行 |

1. **管理员删除用户**

管理员登录系统后台进行修改用户操作的用例规约如表2-3所示：

表2-3管理员修改用户用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员修改用户 |
| 用例ID | CR-UC-003 |
| 用例说明 | 管理员操作系统管理模块的用户管理---用户管理 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“系统管理”，点击“系统管理”，展开其子菜单，点击“用户信息”，进入用户信息界面 3. 选择“用户信息”界面中用户信息列表中的某一条记录，点击“删除”按钮，弹出一个被选中用户信息的删除对话框 4. 删除成功则提示删除成功 |
| 后置条件： | 删除用户成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 只有登录，操作才可顺利进行 |

1. **管理员查询用户**

管理员登录系统后台进行查询用户操作的用例规约如表2-4所示：

表2-4管理员查询用户用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员查询用户 |
| 用例ID | CR-UC-004 |
| 用例说明 | 管理员操作系统管理模块的用户管理---查询用户 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“用户管理”，点击“用户管理”，展开其子菜单，点击“用户信息”，进入用户信息界面 3. 在用户管理子栏目下点击可以看到所有用户的信息 4. 在具体的某一个用户上点击查看可以查看其所有的数据库信息 |
| 扩展 |  |
| 后置条件： | 查询用户成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 只有登录成功件，操作才可顺利进行 |

1. **管理员管理新闻添加**

管理员登录系统后台对新闻信息进行管理进行类似对用户操作的用户角色的增删改查用例规约如表2-6所示：

表2-6管理员管理角色用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员管理新闻添加 |
| 用例ID | CR-UC-006 |
| 用例说明 | 管理员登录后台管理点击新闻管理进行增删改查 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 左边菜单栏点击新闻管理然后会有子兰目显示 3. 在子栏目中选择新闻编辑出现新闻编辑页面 4. 在新闻编辑页面有富文本编辑器可以编写html网页进行新闻编辑 5. 编辑成功可以保存到数据库 |
| 扩展 | (4a)新闻编辑页面可以添加图片但是数量不能多余4张  (4a1)可以编写html网页但是不能出错。 |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 添加新闻如要编写html网页要注意语法 |
|  |  |

1. **管理员新闻管理删除**

管理员登录系统后台进行类似对用户操作的用户权限的增删改查用例规约如表2-7所示：

表2-7管理员管理权限用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员新闻管理删除 |
| 用例ID | CR-UC-007 |
| 用例说明 | 管理员登录后台系统可以删除已经有的新闻 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“新闻管理”，点击“新闻管理”，展开其子菜单，点击“新闻管理管理”，进入权限管理界面 3. 管理员可以查看所有的新闻并且进行删除 |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 登录后才可以删除 |

1. **管理员管理城市**

管理员登录系统后台进行类似对用户操作的网点信息的增删改查用例规约如表2-8所示：

表2-8管理员管理网点信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员管理城市内容 |
| 用例ID | CR-UC-008 |
| 用例说明 | 管理员操作城市管理模块 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“城市管理”，点击“城市管理”，进入城市信息界面 3. 管理员按照对用户的相关操作，对城市人员内容进行操作 4. 操作成功提示成功 |
| 扩展 |  |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 登录成功后才可以操作 |

1. **管理员查看统计分析**

管理员登录系统后台进行统计分析浏览的用例规约如表2-9所示：

表2-9管理员浏览网点章程用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员浏览统计分析 |
| 用例ID | CR-UC-009 |
| 用例说明 | 管理员查看统计信息模块的 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“统计分析”，点击“统计分析”，进入统计分析界面 3. 会显示图表信息 |
| 扩展 |  |
| 后置条件： |  |
| 备注说明： | 登录成功才可以查看 |

1. **管理员新闻首页管理**

管理员登录系统后台进行新闻界面首页的管理规约如表2-10所示：

表2-10管理员查询会员信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员新闻首页管理 |
| 用例ID | CR-UC-010 |
| 用例说明 | 管理员登录后台可以添加置顶首页的视频 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“新闻管理”，点击“新闻管理”，展开其子菜单，点击“首页管理”，进入会员信息界面 3. 可以看到加号点击加号上传视频到服务器 4. 在将视频进行置顶在首页就可以看到 |
| 扩展 |  |
| 后置条件： | 需要成功登录系统后台才可以操作 |
| 备注说明： |  |

1. **管理员通告管理编辑**

管理员登录系统后台进行通告管理章程用例规约表：

表2-11管理员浏览会员章程用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员通告管理编辑 |
| 用例ID | CR-UC-0011 |
| 用例说明 | 管理员操作后太进行通告编辑 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“通告管理“会员管理”，展开其子菜单，点击“通告编辑” 3. 进入同通告编辑页后进行编辑 |
| 扩展 |  |
| 后置条件： |  |
| 备注说明： | 需要登录成功才可以操作 |

1. **管理员通告管理**

管理员登录系统后台进行订单信息查询的用例规约如表2-12所示：

表2-12管理员查询订单信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员通告管理 |
| 用例ID | CR-UC-012 |
| 用例说明 | 管理员后台对通告进行管理查看删除置顶 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“通告管理”击“通告管理”其子菜单，点击“通告管理” 3. 进入通告管理界面 4. 会显示每条通告可以对通告进行操作 |
| 扩展 |  |
| 后置条件： | 操作成功会有提示 |
| 备注说明： | 需要成功登录后可操作 |

1. **用户登录**

用户登录用例规约表如下 2-13

表2-13管理员管理车辆信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户登录 |
| 用例ID | CR-UC-013 |
| 用例说明 | 用户可以登录客户端网站 |
| 执行者 | 用户 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 访问网站首页，点击登录。 2. 条转到登录界面，输入账号密码，验证码。点击”登录” 3. 登录成功界面右上方显示一个小人图标 |
| 扩展 | (4a)输入的信息格式不符合  (4a1)账号密码不正确系统会提示  (4a2)验证码不正确系统也会提示 |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 用户必须有已经注册的账号 |

1. **用户注册**

用户注册账号用例规约如表2-14所示：

表2-14用户注册用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户注册 |
| 用例ID | CR-UC-014 |
| 用例说明 | 用户在客户端界面注册账号 |
| 执行者 | 用户 |
| 前置条件 | 进入客户端界面 |
| 主成功场景 | 1. 用户进入客户端界面 2. 选择注册并且点击注册 3. 在注册界面首先输入手机号 4. 点击”获取验证码” 5. 用户手机会接收到短信验证码 6. 输入验证码 7. 验证码正确会出现输入密码提示 8. 输入密码后点击注册 9. 注册成功会跳转到首页 |
| 扩展 | (4a)输入的信息格式不符合  (4a1)必须输入符合中国大陆手机号  (4a2)不能接收到验证码 |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 手机号必须是中国大陆手机号 |

1. **新闻资讯**

新闻资讯用例规约如表2-15所示：

表2-15管理员管理车辆配送用例规约

|  |  |
| --- | --- |
| 用例名称 | 新闻资讯 |
| 用例ID | CR-UC-015 |
| 用例说明 | 所有访问客户端界面的用户都可以查看新闻 |
| 执行者 | 对所有的用户都敞开 |
| 前置条件 | 访问客户端 |
| 主成功场景 | 1. 访问网站客户端首页 2. 点击新闻资讯 3. 跳转到新闻界面 4. 界面首页轮播图是管理员置顶的内容 5. 其他新闻都会出现在下面 6. 右边会出现点击率最高的新闻 7. 可以点击新闻进入新闻浏览界面查看新闻 |
| 扩展 |  |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 必须成功访问客户端 |

1. **用户个人信息完善**

用户个人信息完善用例规约如表2-16所示：

表2-16 用户个人信息完善用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户个人信息完善 |
| 用例ID | CR-UC-016 |
| 用例说明 | 登录后完善个人信息，年龄、性别、所在国家、省份、城市 |
| 执行者 | 用户 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 用户登录成功 2. 点击人形图像出现个人信息和推出登录 3. 点击个人信息后会出现信息完善页面 4. 输入年龄、选择性别 5. 点击确认 6. 年龄只能输入阿拉伯数字 |
| 扩展 | (4a)输入的信息格式不符  (4a1)格式不正确会提示 |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 用户必须成功登录客户端 |

1. **用户查看通告**

用户通告查看用例规约如表2-17所示：

表2-17 管理员管理收支信息用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户查看通告 |
| 用例ID | CR-UC-017 |
| 用例说明 | 查看平台发出的通告 |
| 执行者 | 用户 |
| 前置条件 | 任何访问客户端界面的人 |
| 主成功场景 | 1. 访问客户端首页 2. 点击最新通告 |
| 扩展 |  |
| 后置条件： | 相关操作成功，可以进行权限范围内的其他操作。 |
| 备注说明： | 必须登录客户端 |

1. **用户交流空间**

用户交流空间的用例规约如表2-19所示：

表2-19管理员查看个人资料用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户交流空间 |
| 用例ID | CR-UC-019 |
| 用例说明 | 用户可在交流空间进行交流互动 |
| 执行者 | 用户 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 用户访问客户端 2. 会看到交流空间栏目 3. 点击跳转到交流空间首页 |
| 扩展 |  |
| 后置条件： | 可以修改个人信息、发表内容、评论、回复、查看 |
| 备注说明： |  |

1. **管理员修改密码**

管理员登录系统后台进行密码修改的用例规约如表2-20所示：

表2-20管理员修改密码用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员修改密码 |
| 用例ID | CR-UC-020 |
| 用例说明 | 管理员操作个人中心模块的修改密码---修改密码 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“个人中心”，点击“个人中心”，展开其子菜单，点击“修改密码”，进入修改密码界面 3. 系统直接获取当前登录用户的原密码 4. 输入“新密码”和“确定新密码”，单击“确定”按钮 5. 系统对输入信息进行验证 6. 系统更新用户表中密码字段 |
| 扩展1 | (5a)“新密码”和“确定新密码”输入不一致  (5a1)系统无法提交信息更新数据库的用户表。  (5a2)返回主成功场景步骤(4) |
| 扩展2 | (5b)“新密码”和“原密码”一样  (5b1)系统无法提交信息更新数据库的用户表。  (5b2)返回主成功场景步骤(4) |
| 后置条件： | 修改密码成功，可以进行权限范围内的其他操作。 |
| 备注说明： |  |

1. **管理员注销登录**

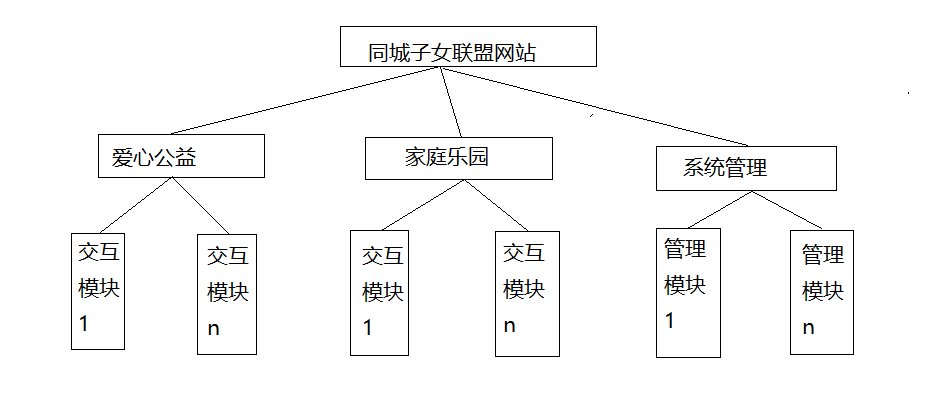
管理员登录系统后台进行注销登录的用例规约如表2-21所示：

表2-21管理员注销登录用例规约

|  |  |
| --- | --- |
| 用例名称 | 管理员注销登录 |
| 用例ID | CR-UC-021 |
| 用例说明 | 管理员操作个人中心模块的注销登录---注销登录 |
| 执行者 | 管理员 |
| 前置条件 | 登录成功 |
| 主成功场景 | 1. 管理员进入系统后台主页 2. 选择后台主页左侧的菜单导航栏的“个人中心”，点击“个人中心”，展开其子菜单，点击“注销登录”，注销当前登录用户 3. 系统返回到登录页面 |
| 扩展 |  |
| 后置条件： | 注销登录成功，可以进行权限范围内的其他操作。 |
| 备注说明： |  |

第3章 系统分析

同城子女联盟网站实现信息户动，主要分为系统管理、家庭乐园、爱心公益三部分。系统管理主要有两层结构，用户管理，用户发表内容管理结构图如下所示



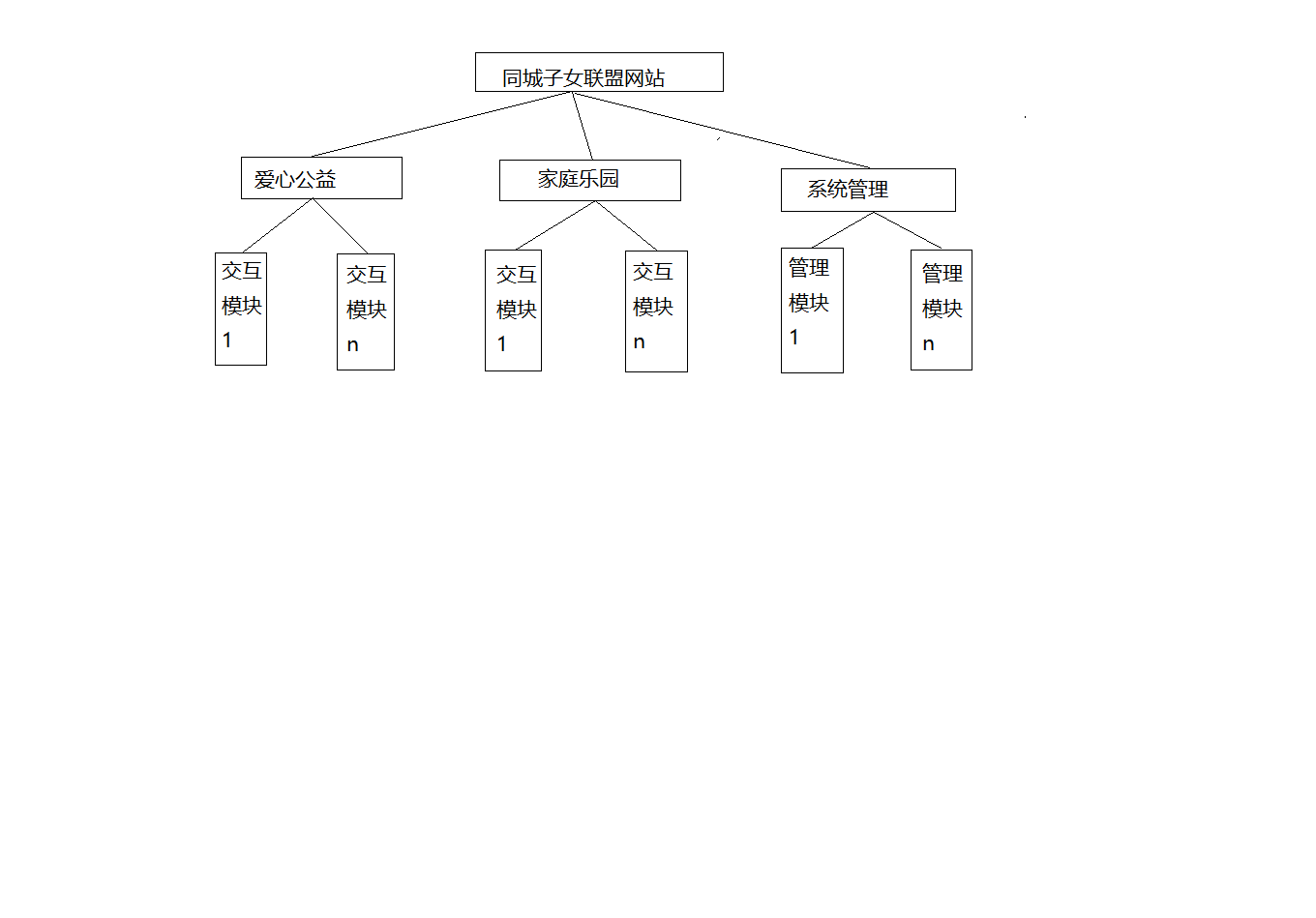


图3-1 系统总体结构

## 3.1静态模型

### 3.1.1 实体类

1. 有属性
2. 有操作
3. 与其他类有关联

类描述了一组对象的公共属性和行为，类为对象规定了它们的属性和所能提供的操作。下面是该系统中系统管理员权限内各个类之间的关联模型图。这些类包括：管理员、用户、新闻、图片、地址、消息、评论、回复、通告、新闻视频。如图3-1所示：

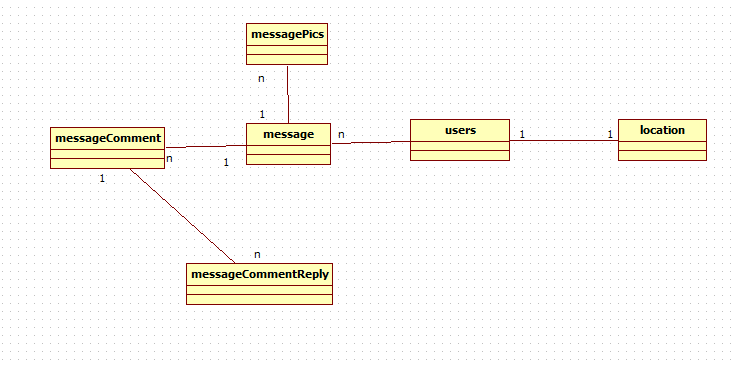


图3-2 分析实体类

### 3.1.2边界类

系统的用户通过边界类与系统打交道，用户向边界类发起访问，边界类把用户的申请提交给控制类，控制类最终访问实体类。一般在进行设计时可能一个用例对应一个边界类，也可能是一个用户对应一个边界类，有时一个边界类可能还要调用另一个边界类才能完成用例。系统主要的边界类如下表所示，边界类的命名以具体的英文描述，系统的边界类说明如表3-1所示：

表3-1系统边界类说明表

|  |  |
| --- | --- |
| **边界类名称** | **说明** |
| LoginSystem.vue | 后台登录边界类 |
| Admin.vue | 后台登录成功主页面边界类 |
| NewsEdit.vue | 后台新闻编辑边界类 |
| Newsmanager.vue | 新闻管理边界类 |
| NewsFirstpageManager.vue | 新闻首页边界类 |
| UserManager.vue | 用户管理边界类 |
| AddAdministrater.vue | 管理员添加边界类 |
| Notice.vue | 通告编辑边界类 |
| NoticeManager.vue | 通告管理边界类 |
| staticeManager.vue | 信息统计边界类 |
| CityManager.vue | 城市管理边界类 |
| Login.vue | 用户登录边界类 |
| FinishedInfo.vue | 用户完善信息边界类 |
| Register.vue | 用户注册边界类 |
| News.vue | 新闻信息边界类 |
| NewsNextPage.vue | 新闻查看边界类 |
| Home.vue | 首页边界类 |
| HomePage.vue | 交流空间边界类 |
| PersonSpace.vue | 个人空间边界类 |
| MessageDisplay.vue | 消息查看边界类 |

### 3.1.3控制类

边界类通过控制类访问实体类，这是三层架构设计体系结构，本系统一个实体类基本有一个控制类做管理。系统主要的控制类如表3-2所示：

表3-2 系统控制类说明表

|  |  |
| --- | --- |
| **控制类名称** | **说明** |
| administratorController | 管理员添加控制类 |
| userController | 用户管理控制类 |
| register | 用户注册控制类 |
| login | 用户登录控制类 |
| locationControlller | 用户地址控制类 |
| noticeOperate | 通告控制类 |
| httpVideos | 首页视频控制类 |
| newsMessageController | 新闻信息控制类 |
| NewsPageOperate | 新闻推送控制类 |
| newsVideo | 视频推送控制类 |
| messageCommentController | 消息评论控制类 |
| messageCommentReplyController | 消息回复控制类 |
| messageController | 消息控制类 |
| messagePicsController | 消息图片控制类 |

## 3.2时序图的分析

时序图主要描述了不同对象之间发送消息的时间顺序显示多个对象之间的动态协作。本系统基本上对每个用例建立了一个时序图，建立时序图是为了进一步描述用例。

本项目的体系结构采用三层架构规范。其中:

表示层，用于与用户进行交互并显示结果，在时序图中由边界类来实现这一层。

控制层，主要完成三项任务，一是调用实体类，二是业务逻辑验证，三是决定将合适的视图组件返回给用户。

模型层，主要由实体类组成，包括一些基本的逻辑。

1. **登录时序图**

管理员在登录边界类中输入登录验证信息：用户名、密码、用户类型，通过用户控制类调用用户实体类完成用例，最终验证成功时进入系统主界面。管理员登录时序图如图3-3所示：

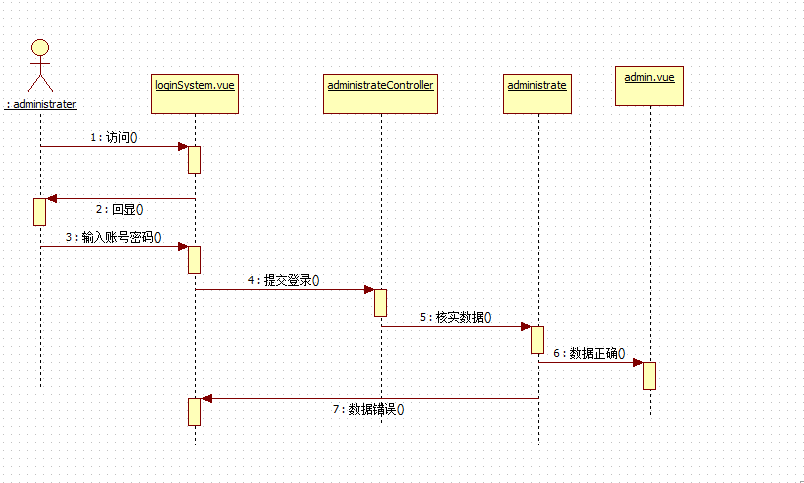


图3-3管理员登录时序图

1. **用户登录时序图**

管理员在修改密码边界类中用户输入：原密码、新密码、确认新密码，通过用户控制类调用用户实体类完成用例，最终返回到系统主界面。管理员管理个人中心-修改密码时序图如图3-4所示：

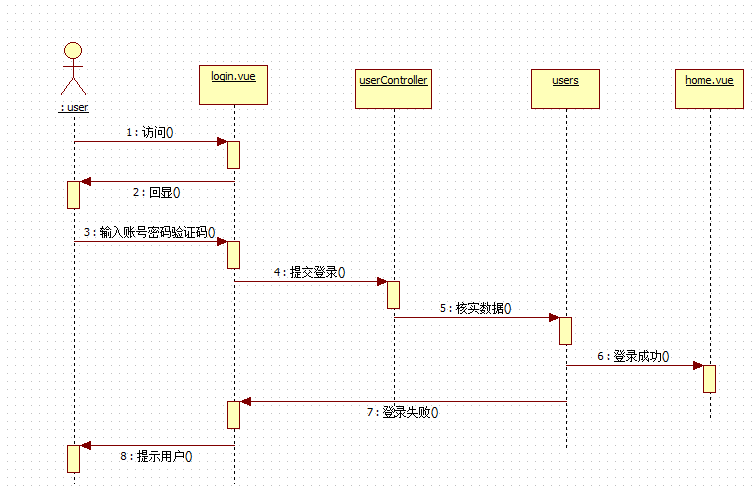


图3-4 修改密码时序图

1. **用户完善信息时序图**

管理员登录系统后台后进入系统用户信息边界类，选择查看、删除、添加、修改用户，通过用户控制类调用用户实体类完成用例，最终系统提示操作成功。管理员管理用户时序图如图3-5所示：

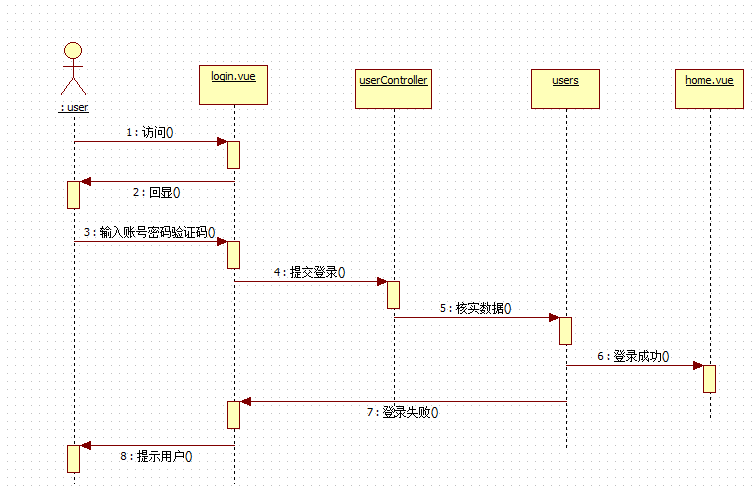


图3-5用户信息完善时序图

1. **用户修改密码时序图**

管理员登录系统后台后进入系统会员信息边界类，选择查看、查询会员，通过会员控制类调用会员实体类完成用例，最终系统提示操作成功。管理员管理会员时序图如图3-6所示：

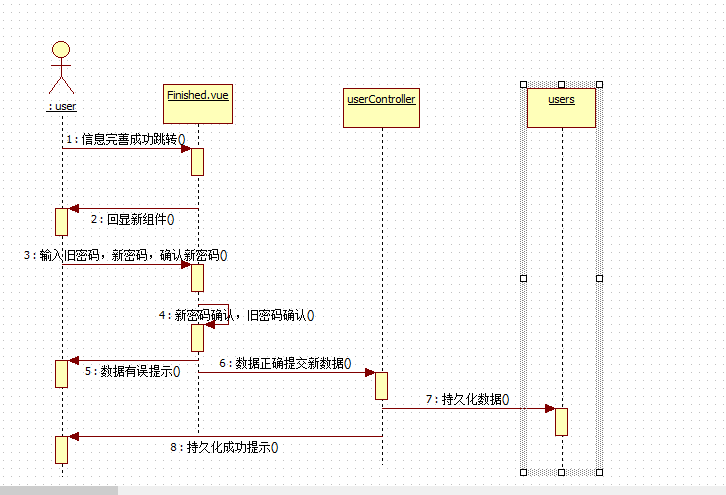


图3-6会员信息管理时序图

1. **用户浏览新闻时序图**

管理员登录系统后台后进入系统车辆信息边界类，选择查看、查询、添加、修改、删除车辆，通过车辆控制类调用车辆实体类完成用例，最终系统提示操作成功。管理员管理车辆时序图如图3-7所示：

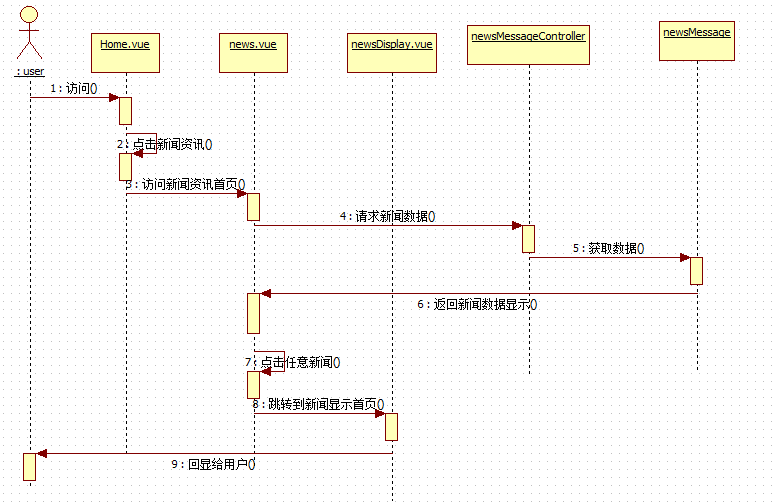


图3-7 用户浏览新闻时序图

1. **用户注册时序图**

用户登录首页点击注册跳转至注册页面通过输入手机号获取验证码来注册账号最终系统提示操作成功。管理员管理车主时序图如图3-3所示：

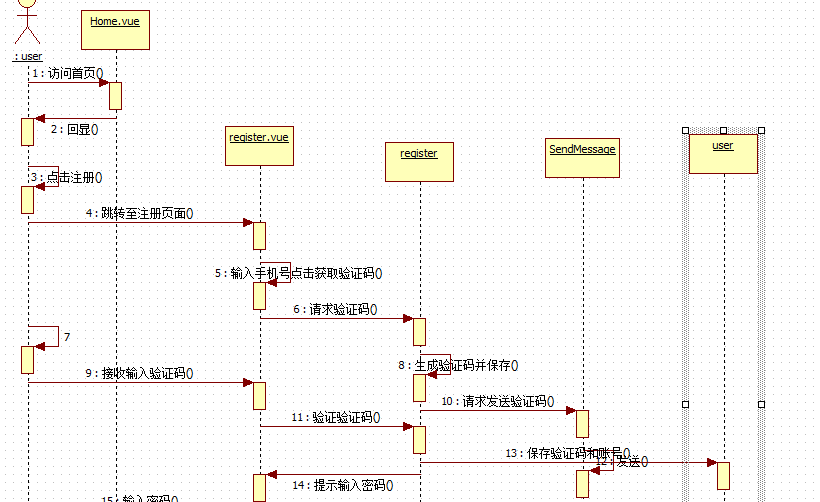


图3-8用户注册时序图

1. **用户交流空间时序图**

用户登陆后可进入交流空间然后可以发表自己的内容参与交流，最终系统提示操作成功。用户交流空间时序图如图3-9所示：

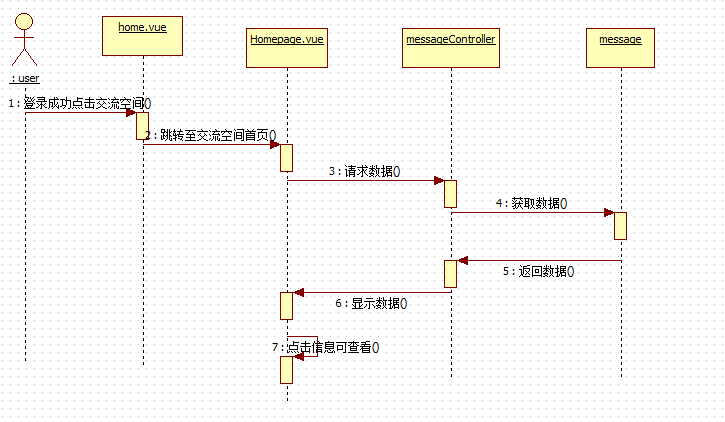


图3-9交流空间时序图

1. **用户个人空间时序图**

用户登录客户端进入交流空间首页点击个人空间进行设置和交互，最终系统提示操作成功。用户个人空间时序图如图3-10所示：

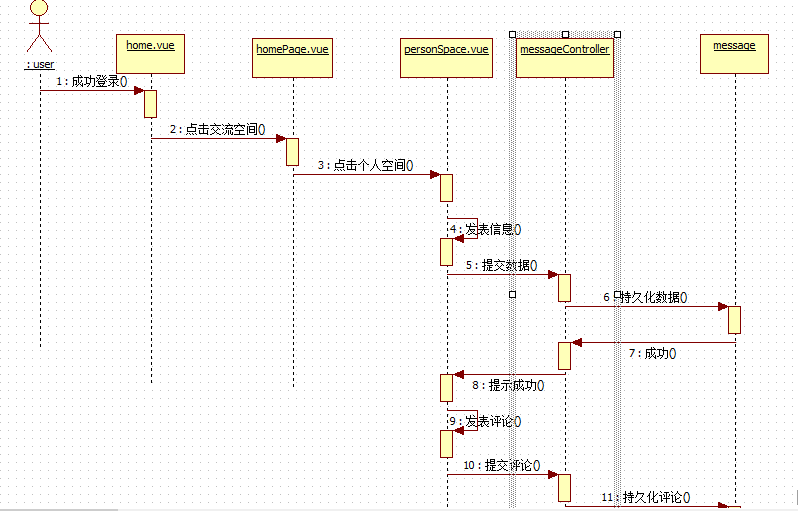


图3-10个人空间时序图

1. **管理员用户管理时序图**

管理员登录系统后台后进入系统网点对用户信息进行查看删除。通过用户管理控制类调用用户实体类完成用例，最终系统提示操作成功。管理员管理用户时序图如图3-11所示：

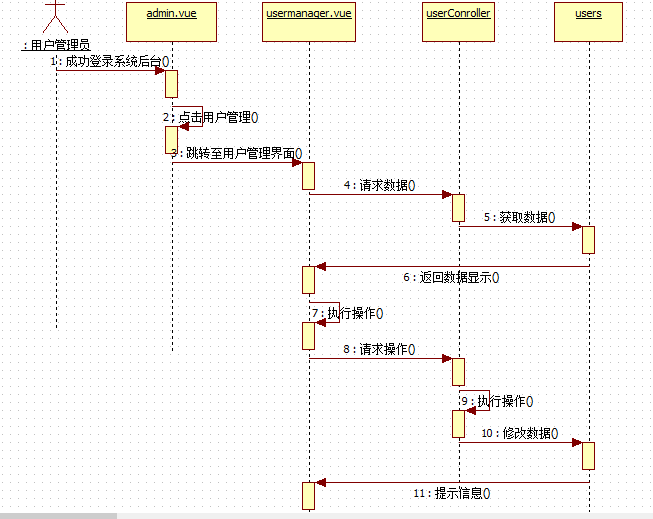


图3-11用户管理时序图

1. **管理员登录时序图**

管理员访问后台登录边界类，输入数据，通过调用管理员登录控制类来核实数据实体类完成用例，最终系统提示操作成功。管理员登录时序图如图3-12所示：

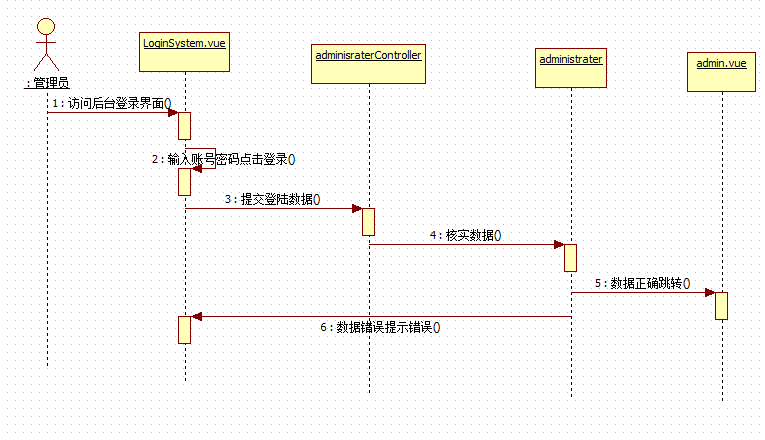


图3-12管理员登录时序图

1. **统计管理时序图**

管理员登录系统后台后进入系统统计边界类，选择查看、查询网点数据统计，通过统计控制类调用统计实体类完成用例，最终系统提示操作成功。管理员管理统计时序图如图3-12所示：

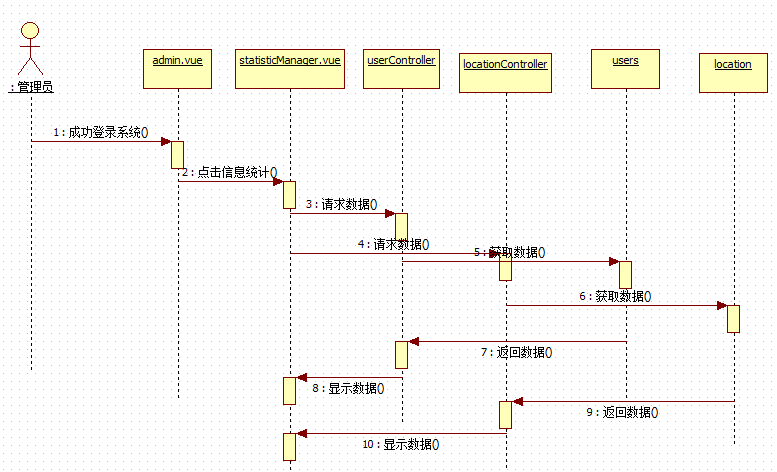
****

图3-12统计管理时序图

1. **新闻管理操作时序图**

新闻管理员访问边界类，通过控制类登录后台系统，在边界类中执行操作，调用控制类对实体数据进行修改。最终提示操作成功信息。管理员管理新闻时序图如图3-13所示：

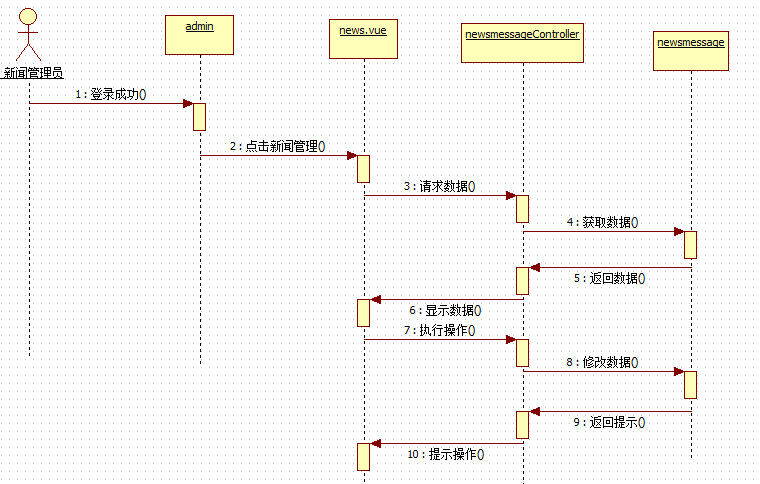
****

图3-13管理员新闻管理时序图

1. **通告管理操作时序图**

通告管理员访问边界类，通过控制类登录后台系统，在边界类中执行操作，调用控制类对实体数据进行修改。最终提示操作成功信息。管理员管理通告时序图如图3-14所示：

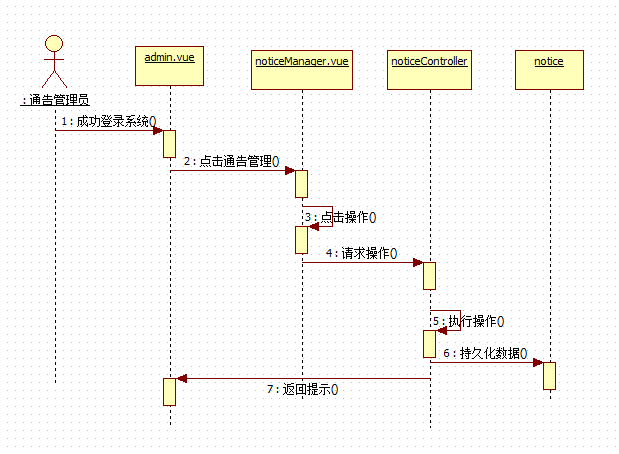


图3-14管理员通告管理时序图

1. **城市管理操作时序图**

城市管理员访问边界类，通过控制类登录后台系统，在边界类中执行操作，调用控制类对实体数据进行修改。最终提示操作成功信息。管理员管理城市时序图如图3-15所示：

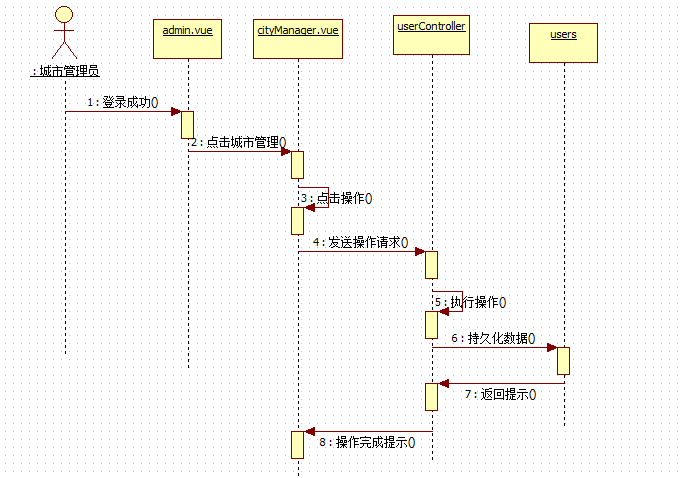


图3-15管理员城市管理时序图

**第4章 系统设计**

## 4.1架构设计

本系统采用基于Web服务的B/S结构（即浏览器和服务器结构）的架构。在B/S结构下，用户操作界面是通过Web页面来展示实现，极少部分的事务逻辑需要在前端（Browser）实现，但是主要事务逻辑操作在服务器端实现，形成

所谓三层Three-tier结构。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。B/S架构管理软件更方便、速度快、效果优。

系统使用基于MVC模式的三层架构，分为View表示层、Controller控制层、JavaBean数据访问层以及数据库，如图4-1所示：



图4-1 MVC模式的三层架构图

## 4.2类设计

1. **用户信息Users类**

用户信息类包含属性用户id、姓名、性别、年龄、用户类型、账号、登录标志、注册验证码、密码、图像、昵称、用户权限、记住密码标志、完成信息标志、用户图像、地址、用户消息，用户类的方法有登录、增加、删除、修改和查询等，设计用户类图如图4-2所示：

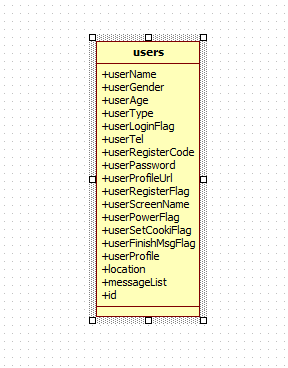


图4-2 用户类图

1. **通告notice类**

权限类包含属性权限id、通告标题、通告内容，置顶标志，通告类的操作有增加、删除、查询等，通告权限类图如图4-3所示：

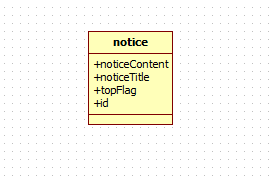


图4-3通告类图

1. **新闻信息newsMessage类**

角色类包含属性角色id、标题、内容、置顶标志、图片、点击量、视频、类型、作者，新闻信息类的方法有增加、删除、查询等，新闻信息类图如图4-4所示：

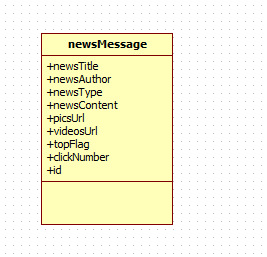


图4-4 新闻信息类图

1. **地址location类**

车辆信息类包含属性车辆id、省份、城市、国家，地址类的方法有增加，设计地址类图如图4-5所示：

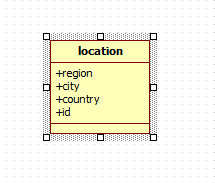


图4-5 地址类图

1. **管理员administrater类**

车主信息类包含属性车主id、账号、密码、类型、登录标志，管理员类图如4-6所示：

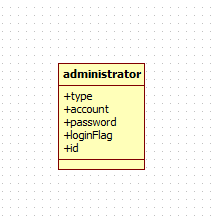


图4-6管理员类图

1. **消息message类**

交流空间的消息类包含的属性有id、标题、内容、种类、点赞数、批评数、评论、国家、城市、省份。消息类图如图4-7所示：

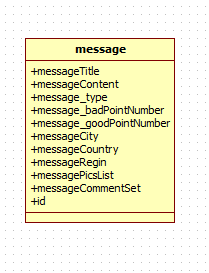


图4-7 消息类图

1. **评论messageComment类**

评论类包含属性id、评论内容、评论人昵称、评论人id、消息id、回复、评论类图如图4-8所示：



图4-8评论类图

1. **评论回复messageCommentReply类**

评论回复类包含的属性有id、评论人昵称、回复人昵称、回复内容、评论人id、回复人id、评论id、消息id评论回复类图如图4-9所示：

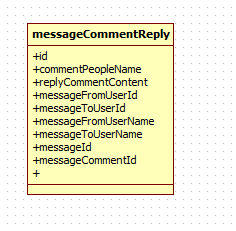


图4-9 评论回复类图

1. **消息图片messagePics类**

网点信息类包含属性网点id、图片信息、消息图片类图如图4-10所示：

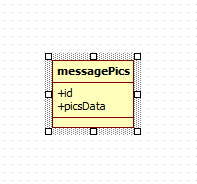


图4-10 消息图片类图

## 4.3物理模型

Jpa根据实体类上的注解以及字段上的注解的属性会在数据库映射相对应的表和字段。在Jpa映射规则中String 类型在数据库默认映射成varchar类型。下面将一一展示各个实体类中在mysql数据库表中的具体类型。

1. **用户信息user\_info表**

用户表（user\_info）用于存放用户的基本信息，如表4-1所示：

表4-1用户表(join\_user\_tb)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 系统用户id（用户表的主键，不为空） |
| datecreated | datetime | 创建日期 （不为空） |
| datemodified | datetime | 修改日期（不为空） |
| deleted | bit(1) | 删除标志 |
| entity\_name | varchar | 用户头像 |
| user\_age | varchar | 用户年龄 |
| user\_finish\_msg\_flag | varchar | 信息完成标志 |
| user\_gender | varchar | 用户性别 |
| user\_login\_flag | varchar | 用户登录标志 |
| user\_name | varchar | 用户名子 |
| user\_password | varchar | 用户密码 |
| user\_power\_flag | varchar | 用户权限 |
| user\_profile\_url | mediumblob | 用户图像地址 |
| user\_profile | Longtext | 用户图像 |
| user\_register\_flag | Varchar | 注册标志 |
| user\_register\_code | Varchar | 注册验证码 |
| user\_screen\_name | Varchar | 昵称 |
| user\_set\_cooki\_flag | Varchar | 设置cooki标志 |
| user\_type | Varchar | 类型 |
| location\_id | bigint(20) | 地址外键 |
| user\_tel | Varchar | 账号/电话 |

1. **通告notice表**

权限表（notice）用于存放系统通告信息，如表4-2所示：

表4-2权限表(notice)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 权限id（权限表的主键，不为空） |
| datecreated | datetime | 创建日期 （不为空） |
| entity\_name | Varchar | 修改日期（不为空） |
| deleted | bit(1) | 删除标志 |
| top\_flag | Varchar | 置顶标志 |
| notice\_title | Varchar | 通告标题 |
| notice\_content | Varchar | 通告内容 |
| notice\_content | Varchar | 通告内容 |
| datemodified | datetime | 修改日期（不为空） |

1. **地址location表**

角色表（location）用于存放系统用户的地址信息，如表4-3所示：

表4-3地址表(location)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | datetime | 创建日期 （不为空） |
| datemodified | datetime | 修改日期（不为空） |
| deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名称 |
| region | Varchar | 省 |
| country | Varchar | 国 |
| city | Varchar | 市 |

1. **新闻信息newsMessage表**

车辆表（news\_message）用于存放系统新闻资讯的新闻内容基本信，如表4-4所示：

表4-4新闻信息表(news\_message )

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | datetime | 创建日期 （不为空） |
| datemodified | datetime | 修改日期（不为空） |
| deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名称 |
| click\_number | Varchar | 点击量 |
| news\_author | Varchar | 作者 |
| news\_content | longtext | 内容 |
| news\_title | Varchar | 标题 |
| news\_type | Varchar | 类型 |
| pics\_url | mediumblob | 图片地址 |
| top\_flag | Varchar | 置顶标志 |
| videos\_url | Varchar | 视频地址 |
| news\_page\_id | bigint(20) | 新闻id |
|  |  |  |

1. **管理员信息表**

车主表（administrator）用于存放系统管理员信息，如表4-5所示：

表4-5车主表(administrator)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | Datetime | 创建日期 （不为空） |
| datemodified | Datetime | 修改日期（不为空） |
| deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名 |
| login\_flag | bit(1) | 登录标志 |
| account | varchar(255) | 账号 |
| password | varchar(255) | 密码 |
| type | varchar(25) | 类型 |

1. **交流信息表**

交流信息表（communication\_space\_message）用于交流空间信息，如表4-6所示：

表4-6交流信息表(communication\_space\_message )

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | Datetime | 创建日期（不为空） |
| datemodified | Datetime | 修改日期（不为空） |
| deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名 |
| message\_content | Longtext | 内容 |
| message\_title | Varchar | 标题 |
| message\_bad\_point\_number | int(11) | 差点击 |
| message\_good\_point\_number | int(11) | 好点击 |
| message\_type | Varchar | 类型 |
| message\_city | Varchar | 城市 |
| message\_country | Varchar | 国家 |
| message\_regin | Varchar | 省份 |

1. **信息评论表**

信息评论表（communication\_space\_message\_comment）用于存放配发车辆的基本信息，如表4-7所示：

表4-7信息评论表(communication\_space\_message\_comment)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| Datecreated | Datetime | 创建日期 （不为空） |
| datemodified | Datetime | 修改日期（不为空） |
| Deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名 |
| comment\_content | Varchar | 内容 |
| comment\_people\_name | Varchar | 昵称 |
| message\_id | bigint(20) | Id |
| user\_id | bigint(20) | 用户id |

1. **信息评论回复表**

信息评论回复表（communication\_space\_message\_comment\_reply）用于存放派送车辆的基本信息，如表4-8所示：

表4-8信息评论回复表(communication\_space\_message\_comment\_reply)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | Datetime | 创建日期 （不为空） |
| datemodified | Datetime | 修改日期（不为空） |
| deleted | Varchar | 删除标志 |
| entity\_name | Varchar | 实体名 |
| message\_comment\_id | bigint(20) | 评论id |
| message\_from\_user\_id | bigint(20) | 发表人id |
| message\_from\_user\_name | Varchar | 发表人昵称 |
| message\_id | bigint(20) | 消息id |
| message\_to\_user\_id | bigint(20) | 发消息人id |
| reply\_comment\_content | Varchar | 回复评论内容 |
| message\_to\_user\_name | Varchar | 发消息人昵称 |
| comment\_people\_name | Varchar | 评论人昵称 |

1. **信息图片表**

网点信息表（message\_pics）用于交流信息图片信息，如表4-9所示：

表4-9信息图片表(message\_pics)

|  |  |  |
| --- | --- | --- |
| **字段名** | **字段类型** | **字段描述及说明** |
| Id | bigint(20) | 角色id（角色表的主键，不为空） |
| datecreated | Datetime | 创建日期（不为空） |
| datemodified | Datetime | 修改日期（不为空） |
| pic\_data | mediumblob | 图片Base64编码信息 |

第5章 系统实现与测试

本系统应用的是前后端分离实现人机交互。前端使用vue框架和element –ui组件以及node.js其他的相关插件。后端使用以spring boot为核心的配置框架。在数据持久化方面使用mysql来存储一些结构化数据，而对于类似于视频大文件则采用文件存储来实现持久化。在前后端交互方面主要采用以http协议为核心的axios插件来实现前后端的交互。后端的控制类则采用REST 风格的接口来实现对于http请求的相应。本系统的权限管理在前端实现，以对路由管理的方式来实现用户是否有权限进行跳转来实现权限管理。对于部分模块采用vuex的结构的方式来实现状态管理。

## 5.1 通用模块

### 5.1.1工具类

1. IP查询操作类

package gp.lcw.sd.by.g3p.extension.aLiYunIpCheckService;  
  
import java.io.UnsupportedEncodingException;  
import java.net.URLEncoder;  
import java.security.KeyManagementException;  
import java.security.NoSuchAlgorithmException;  
import java.security.cert.X509Certificate;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Map;  
  
import javax.net.ssl.SSLContext;  
import javax.net.ssl.TrustManager;  
import javax.net.ssl.X509TrustManager;  
  
import org.apache.commons.lang.StringUtils;  
import org.apache.http.HttpResponse;  
import org.apache.http.NameValuePair;  
import org.apache.http.client.HttpClient;  
import org.apache.http.client.entity.UrlEncodedFormEntity;  
import org.apache.http.client.methods.HttpDelete;  
import org.apache.http.client.methods.HttpGet;  
import org.apache.http.client.methods.HttpPost;  
import org.apache.http.client.methods.HttpPut;  
import org.apache.http.conn.ClientConnectionManager;  
import org.apache.http.conn.scheme.Scheme;  
import org.apache.http.conn.scheme.SchemeRegistry;  
import org.apache.http.conn.ssl.SSLSocketFactory;  
import org.apache.http.entity.ByteArrayEntity;  
import org.apache.http.entity.StringEntity;  
import org.apache.http.impl.client.DefaultHttpClient;  
import org.apache.http.message.BasicNameValuePair;  
  
public class HttpUtils {  
  
 */\*\*  
 \* get  
 \*  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doGet(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpGet request = new HttpGet(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* post form  
 \*  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@param*** *bodys  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doPost(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys,  
 Map<String, String> bodys)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpPost request = new HttpPost(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 if (bodys != null) {  
 List<NameValuePair> nameValuePairList = new ArrayList<NameValuePair>();  
  
 for (String key : bodys.keySet()) {  
 nameValuePairList.add(new BasicNameValuePair(key, bodys.get(key)));  
 }  
 UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(nameValuePairList, "utf-8");  
 formEntity.setContentType("application/x-www-form-urlencoded; charset=UTF-8");  
 request.setEntity(formEntity);  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* Post String  
 \*  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@param*** *body  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doPost(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys,  
 String body)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpPost request = new HttpPost(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 if (StringUtils.*isNotBlank*(body)) {  
 request.setEntity(new StringEntity(body, "utf-8"));  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* Post stream  
 \*  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@param*** *body  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doPost(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys,  
 byte[] body)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpPost request = new HttpPost(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 if (body != null) {  
 request.setEntity(new ByteArrayEntity(body));  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* Put String  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@param*** *body  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doPut(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys,  
 String body)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpPut request = new HttpPut(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 if (StringUtils.*isNotBlank*(body)) {  
 request.setEntity(new StringEntity(body, "utf-8"));  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* Put stream  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@param*** *body  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doPut(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys,  
 byte[] body)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpPut request = new HttpPut(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 if (body != null) {  
 request.setEntity(new ByteArrayEntity(body));  
 }  
  
 return httpClient.execute(request);  
 }  
  
 */\*\*  
 \* Delete  
 \*  
 \** ***@param*** *host  
 \** ***@param*** *path  
 \** ***@param*** *method  
 \** ***@param*** *headers  
 \** ***@param*** *querys  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* public static HttpResponse doDelete(String host, String path, String method,  
 Map<String, String> headers,  
 Map<String, String> querys)  
 throws Exception {  
 HttpClient httpClient = *wrapClient*(host);  
  
 HttpDelete request = new HttpDelete(*buildUrl*(host, path, querys));  
 for (Map.Entry<String, String> e : headers.entrySet()) {  
 request.addHeader(e.getKey(), e.getValue());  
 }  
  
 return httpClient.execute(request);  
 }  
  
 private static String buildUrl(String host, String path, Map<String, String> querys) throws UnsupportedEncodingException {  
 StringBuilder sbUrl = new StringBuilder();  
 sbUrl.append(host);  
 if (!StringUtils.*isBlank*(path)) {  
 sbUrl.append(path);  
 }  
 if (null != querys) {  
 StringBuilder sbQuery = new StringBuilder();  
 for (Map.Entry<String, String> query : querys.entrySet()) {  
 if (0 < sbQuery.length()) {  
 sbQuery.append("&");  
 }  
 if (StringUtils.*isBlank*(query.getKey()) && !StringUtils.*isBlank*(query.getValue())) {  
 sbQuery.append(query.getValue());  
 }  
 if (!StringUtils.*isBlank*(query.getKey())) {  
 sbQuery.append(query.getKey());  
 if (!StringUtils.*isBlank*(query.getValue())) {  
 sbQuery.append("=");  
 sbQuery.append(URLEncoder.*encode*(query.getValue(), "utf-8"));  
 }  
 }  
 }  
 if (0 < sbQuery.length()) {  
 sbUrl.append("?").append(sbQuery);  
 }  
 }  
  
 return sbUrl.toString();  
 }  
  
 private static HttpClient wrapClient(String host) {  
 HttpClient httpClient = new DefaultHttpClient();  
 if (host.startsWith("https://")) {  
 *sslClient*(httpClient);  
 }  
  
 return httpClient;  
 }  
  
 private static void sslClient(HttpClient httpClient) {  
 try {  
 SSLContext ctx = SSLContext.*getInstance*("TLS");  
 X509TrustManager tm = new X509TrustManager() {  
 public X509Certificate[] getAcceptedIssuers() {  
 return null;  
 }  
 public void checkClientTrusted(X509Certificate[] xcs, String str) {  
  
 }  
 public void checkServerTrusted(X509Certificate[] xcs, String str) {  
  
 }  
 };  
 ctx.init(null, new TrustManager[] { tm }, null);  
 SSLSocketFactory ssf = new SSLSocketFactory(ctx);  
 ssf.setHostnameVerifier(SSLSocketFactory.*ALLOW\_ALL\_HOSTNAME\_VERIFIER*);  
 ClientConnectionManager ccm = httpClient.getConnectionManager();  
 SchemeRegistry registry = ccm.getSchemeRegistry();  
 registry.register(new Scheme("https", 443, ssf));  
 } catch (KeyManagementException ex) {  
 throw new RuntimeException(ex);  
 } catch (NoSuchAlgorithmException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
}

1. 短信发送类

@lombok.Data  
public class Data {  
  
 // 短信应用 SDK AppID，SDK AppID 以1400开头  
 private int appId = 1400353262;  
 // 短信应用SDK AppKey  
 private String appKey = "6437a505ca2181bb30482b12b12422ae";  
 // 需要发送短信的手机号码，可以定义多个手机号码  
 private String phoneNumbers;  
 // 短信模板ID，需要在短信控制台中申请，我们查看自己的短信模板ID即可  
 private int templateId = 582647;  
 // 签名，签名参数使用的是`签名内容`，而不是`签名ID`，真实的签名需要在短信控制台申请，这里按自己的来修改就好  
 private String smsSign = "少儿编程教学训练营";  
}

private Data smsParameter = new Data();  
/\*https://www.jianshu.com/p/4c0d473d743e\*/  
   
 public void testSendMessage(String Contentofshortmessage){  
 try {  
 System.*out*.println("短信测试");  
 //短信模板中的参数列表  
 String[] params = {Contentofshortmessage,"5"};//设置短信内容Contentofshortmessage  
 SmsSingleSender sender = new SmsSingleSender(smsParameter.getAppId(), smsParameter.getAppKey());  
 SmsSingleSenderResult result = sender.sendWithParam("86", smsParameter.getPhoneNumbers(), smsParameter.getTemplateId(), params, smsParameter.getSmsSign(), "", "");  
 } catch (HTTPException e) {  
 System.*out*.println(" HTTP 响应码错误");  
 // HTTP 响应码错误  
 e.printStackTrace();  
 } catch (JSONException e) {  
 System.*out*.println(" JSON 解析错误");  
 // JSON 解析错误  
 e.printStackTrace();  
 } catch (IOException e) {  
 System.*out*.println("网络 IO 错误");  
 // 网络 IO 错误  
 e.printStackTrace();  
 }  
 }

### 5.1.2数据源配置

1. 数据源配置

spring.datasource.url=jdbc:mysql://localhost:3306/qianfuzhe?serverTimezone=UTC&useSSL=true  
useUnicode=true&characterEncoding=UTF-8  
spring.datasource.username=root  
spring.datasource.password=root  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.jpa.database = *MYSQL*spring.jpa.show-sql = **true**spring.jpa.hibernate.ddl-auto = update

1. http头相关配置

spring.servlet.multipart.enabled=**true**spring.servlet.multipart.file-size-threshold=0  
spring.servlet.multipart.location="E:\\newsMessage\\test.txt"  
spring.servlet.multipart.max-file-size=100MB  
spring.servlet.multipart.max-request-size=100MB  
spring.servlet.multipart.resolve-lazily=**false**#设置请求头的长度  
server.max-http-header-size=10000000

## 5.2 功能模块

### 5.2.1 后台登录界面

系统后台运行之后，进入的就是一个后台登录的界面，在这个页面中你需要输入用户名和密码，同时要保证数据库中该用户的信息是存在，才登录后台界面。如图5-1所示：



图5-1后台登录界面

### 5.2.2 后台主页

以管理员身份登录后，进入系统后台主页，这是用EasyUi实现的框架。管理员可以操作界面左侧导航菜单的相应管理模块。如：个人中心、系统管理、会员管理、车辆管理、配车管理、订单管理、网点管理、财务管理、统计管理。如图5-2所示：



图5-2后台主页面（个人中心页）

### 5.2.3 个人中心

个人中心包含个人资料、修改密码和注销登录，个人资料可以查看和修改、密码进行修改之后可以更新到数据库的用户表里，注销登录操作可以退出系统跳转到登录页。该界面和上图中的效果一样。

### 5.2.4 系统管理

系统管理模块包含用户信息、角色管理和权限管理三个子模块，每个子模块都是对应数据库表里信息的展示，可以进行浏览、查询、添加、修改和删除操作。如图5-3所示：

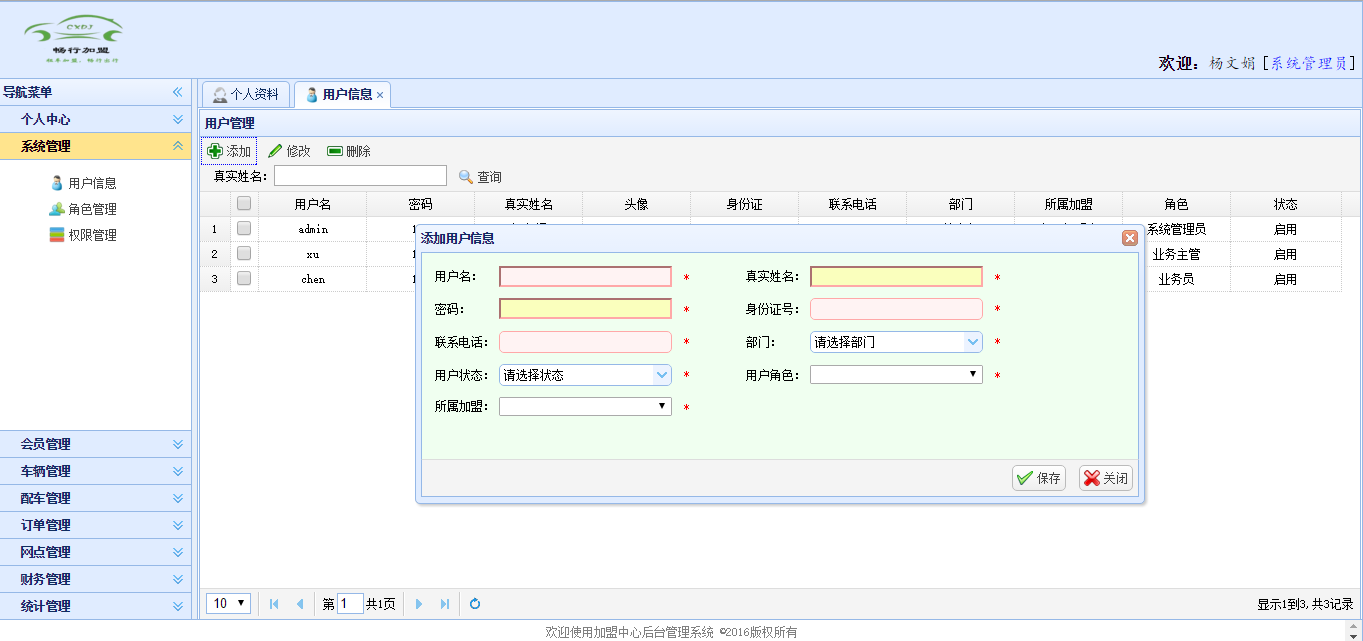


图5-3后台系统管理---用户添加页面

### 5.2.5 会员管理

会员管理模块包含会员信息、会员等级和会员权限三个子模块，每个子模块都是对应数据库表里信息的展示，可以进行浏览、查询操作。如图5-4所示：

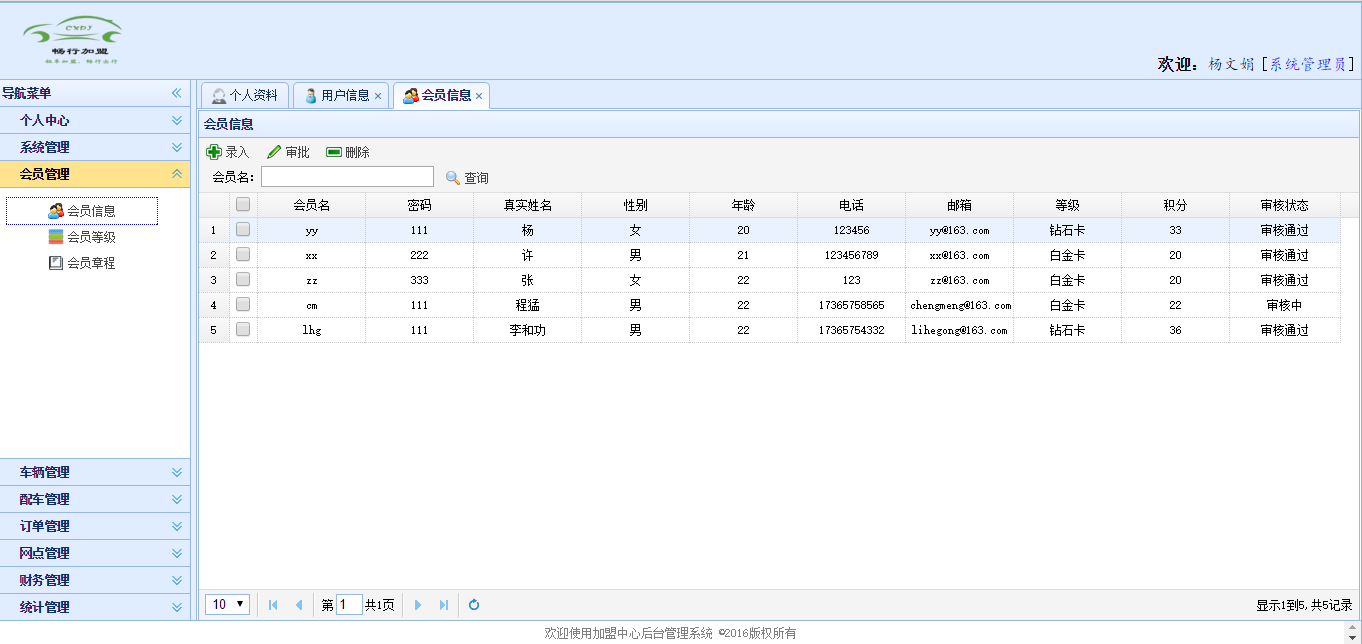
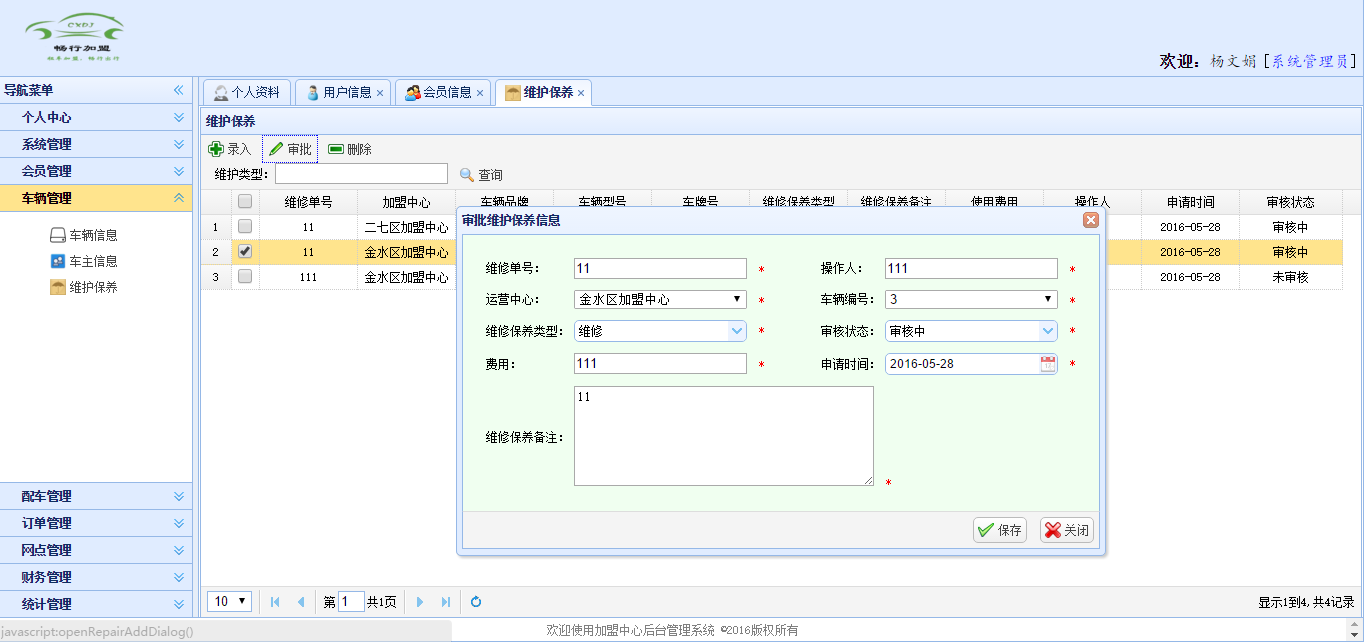


图5-4后台会员管理---会员信息列表页

### 5.2.6 车辆管理

车辆管理模块包含车辆信息、车主信息和维护保养三个子模块，每个子模块都是对应数据库表里信息的展示，车主信息对应的是车辆信息中属于挂靠车辆所属的信息，都可以进行浏览、查询、添加、修改和删除操作。如图5-5所示：

图5-5后台车辆管理---维护保养添加页面

### 5.2.7 配车管理

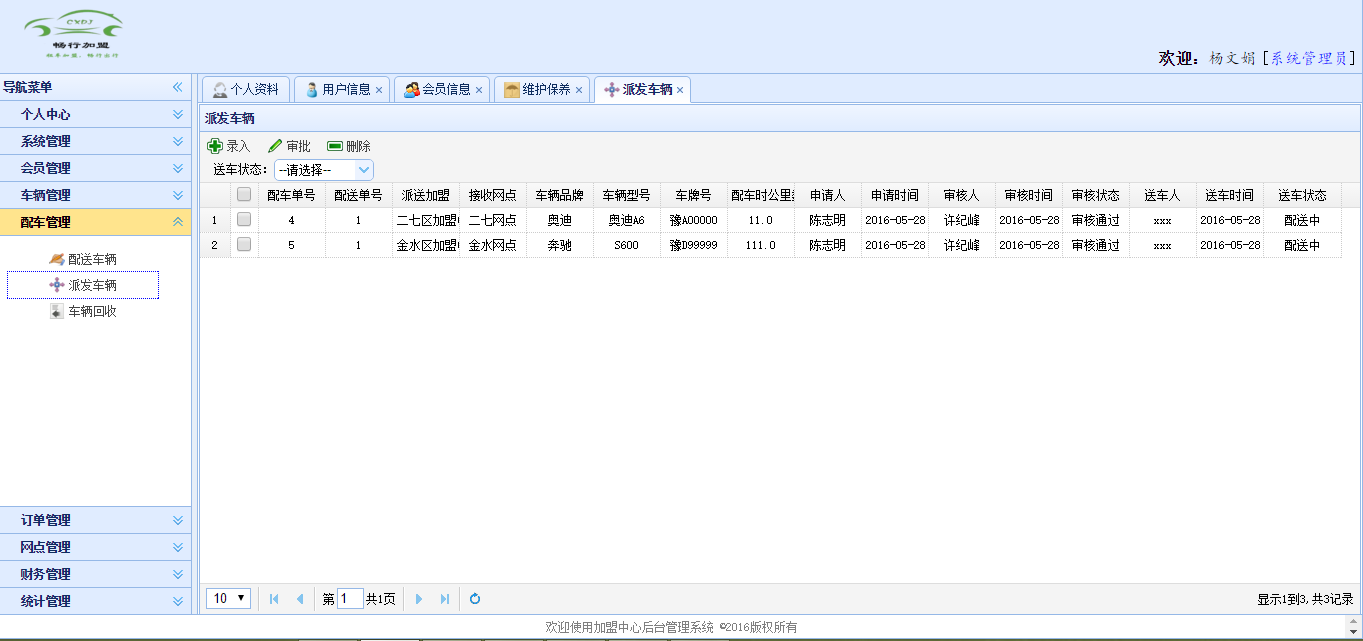
配车模块包含配送车辆、派送车辆和车辆回收三个子模块，前两个子模块都是对应数据库表里信息的展示，可以进行浏览、查询、添加、修改和删除操作。回收车辆只能回收自持车辆，但自持车辆是运营总部 的，所以只有运营总部可以操作，这里仅是留了一个接口。如图5-6所示：

图5-6后台配车管理---派送信息列表页面

### 5.2.8 订单管理

订单管理模块包含订单信息，可以进行浏览、查询操作。

### 5.2.9 网点管理

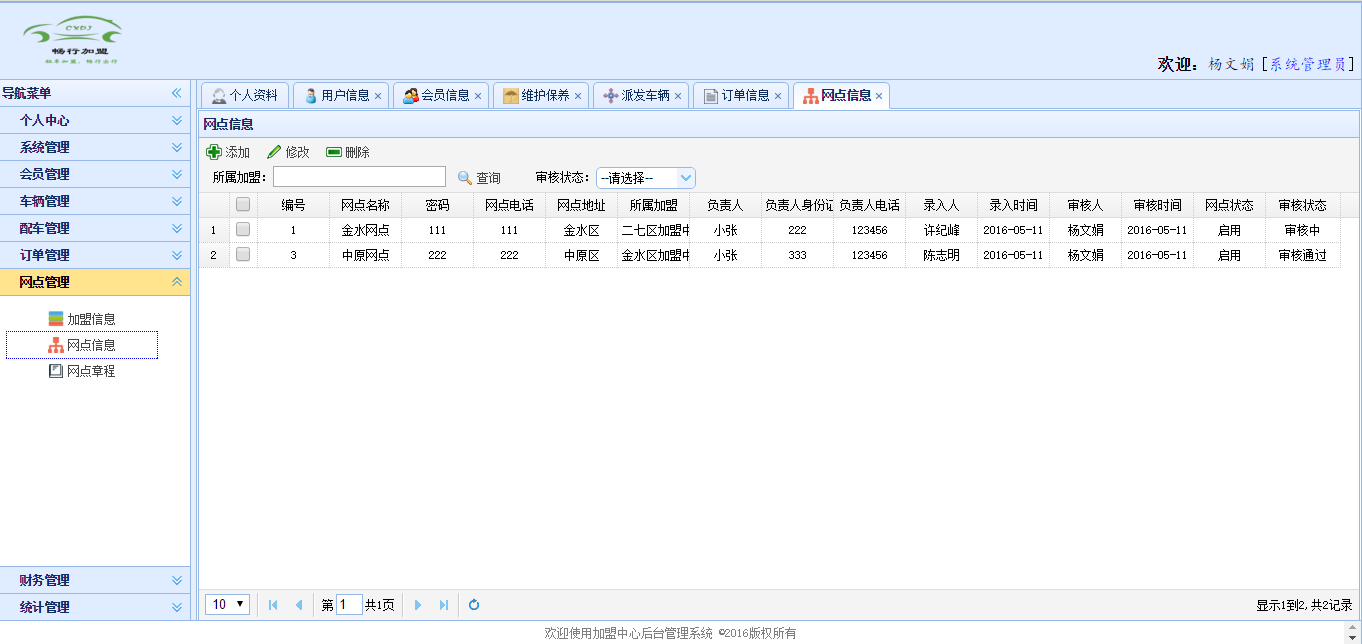
网点管理模块包含加盟信息、网点信息和网点章程三个子模块，每个子模块都是对应数据库表里信息的展示，加盟信息对应的本来是运营总部的内容，这是为了操作提供的一个接口，进行浏览、查询、添加、修改和删除操作。如图5-7所示：

图5-7后台网点管理---网点信息页面

### 5.2.10 财务管理

财务管理模块包含员工工资、收支管理两个个子模块，每个子模块都是对应数据库表里信息的展示都可以进行浏览、查询、添加、修改和删除操作。如图5-8所示：

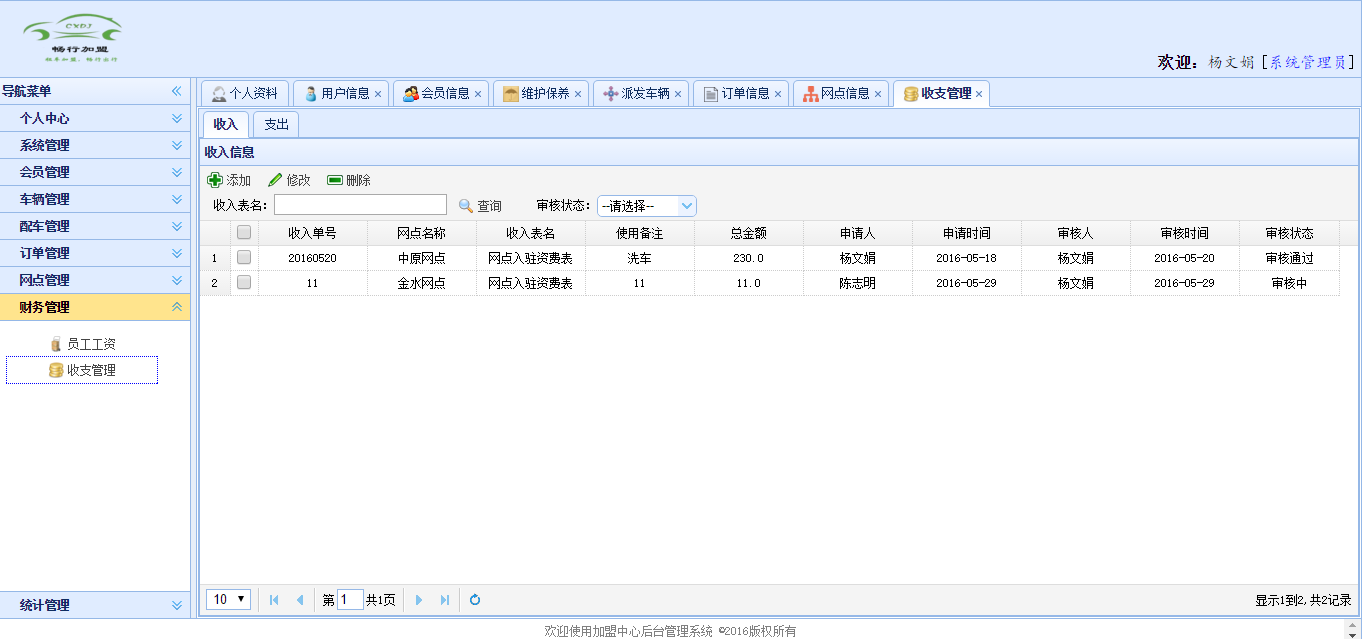


图5-8后台财务管理---收支管理页面

### 5.2.11 统计管理

统计管理模块主要是对网点数据的统计和分析。如图5-9所示：

图5-9后台统计管理---网点分布页面

## 5.3系统运行及测试

测试贯穿于系统的整个开发过程，这里仅列出主要的界面测试用例。

### 5.3.1 登录测试用例

下图的是登录界面的测试用例，测试登录界面能够正确响应鼠标的单击事件。如表5-1所示：

表5-1登录模块测试用例

|  |  |
| --- | --- |
| 项目名称 | 汽车租赁联盟平台加盟运营中心系统设计与实现---系统管理 |
| 测试ID | CR-TEST-001 |
| 测试说明 | 登录界面各组件均能正确响应鼠标键盘单击事件，登录成功后能正确进入系统后台主界面 |
| 测试模块 | 登录模块 |
| 测试目的 | 验证：登录界面各组件均能正确响应鼠标键盘单击事件，进入后台 |
| 初始条件 | 运行系统，显示登录界面 |
| 步骤 | 1. 鼠标单击登录界面中用户名框，在框内输入已存在、正确的用户名。 2. 鼠标单击登录界面中密码框，输入和管理员用户名对应的正确密码。 3. 鼠标点击登录界面中的用户类型下拉框，选择“系统管理员” 4. 鼠标键盘单击“登录”按钮 |
| 期望输出： | 登录操作成功，进入后台管理的主页面 |
| 实际输出： | 正确输出，无误 |
| 测试状态 | 成功通过测试 |
| 测试人： | 杨文娟 |
| 测试时间 | 2016年05月15日 |
| 备注 | 其他窗口界面中各组件对鼠标键盘相应测试参照本测试用例 |

### 5.3.2 添加用户测试用例

表5-2是添加用户功能测试用例。测试添加用户界面能正确响应事件。如表5-2所示：

表5-2添加用户模块测试用例

|  |  |
| --- | --- |
| 项目名称 | 汽车租赁联盟平台加盟运营中心系统设计与实现---系统管理 |
| 测试ID | CR-TEST-002 |
| 测试说明 | 添加用户界面各组件均能正确响应鼠标单击事件，填写信息后可以正确提交数据库中。 |
| 测试模块 | 添加用户模块 |
| 测试目的 | 验证：添加用户界面各组件均能正确响应时间，完成添加功能 |
| 初始条件 | 运行系统，显示添加用户界面 |
| 步骤 | 1. 鼠标单击用户界面中用户编号、用户名、密码、真实名、头像、身份证号、联系电话、部门、所属加盟、角色、状态，在输入信息，系统提示相应的提示信息，输入值符合规范要求。 2. 鼠标键盘单击“确定”按钮 |
| 期望输出： | 系统提示添加成功的提示信息 |
| 实际输出： | 正确输出，无误 |
| 测试状态 | 成功通过测试 |
| 测试人： | 杨文娟 |
| 测试时间 | 2016年05月15日 |
| 备注 | 其他模块中的添加界面中各组件对鼠标键盘相应测试参照本测试用例 |

### 5.3.3 修改用户测试用例

表5-3是修改用户功能测试用例。测试修改用户界面能正确响应事件。如表5-3所示：

表5-3修改用户模块测试用例

|  |  |
| --- | --- |
| 项目名称 | 汽车租赁联盟平台加盟运营中心系统设计与实现---系统管理 |
| 测试ID | CR-TEST-003 |
| 测试说明 | 修改用户界面各组件均能正确响应鼠标单击事件，填写信息后可以正确提交数据库中。 |
| 测试模块 | 修改用户模块 |
| 测试目的 | 验证：修改用户界面各组件均能正确响应时间，完成修改功能 |
| 初始条件 | 运行系统，选择一条用户信息，显示修改用户界面 |
| 步骤 | 1. 鼠标单击修改用户界面中用户编号、用户名、密码、真实名、头像、身份证号、联系电话、部门、所属加盟、角色、状态，在输入信息，系统提示相应的提示信息，输入值符合规范要求。 2. 鼠标键盘单击“确定”按钮 |
| 期望输出： | 系统提示修改成功的提示信息 |
| 实际输出： | 正确输出，无误 |
| 测试状态 | 成功通过测试 |
| 测试人： | 杨文娟 |
| 测试时间 | 2016年05月15日 |
| 备注 | 其他模块中的修改界面中各组件对鼠标键盘相应测试参照本测试用例 |

### 5.3.4 查询用户测试用例

表5-4是查询用户功能测试用例。测试查询用户界面能正确响应事件。如表5-4所示：

表5-4查询用户模块测试用例

|  |  |
| --- | --- |
| 项目名称 | 汽车租赁联盟平台加盟运营中心系统设计与实现---系统管理 |
| 测试ID | CR-TEST-004 |
| 测试说明 | 查询用户界面各组件均能正确响应鼠标单击事件，输入查询条件后可以正确从数据库中读取出相应的查询信息。 |
| 测试模块 | 查询用户模块 |
| 测试目的 | 验证：查询用户界面各组件均能正确响应时间，完成查询功能 |
| 初始条件 | 运行系统，显示用户界面 |
| 步骤 | 1. 鼠标单击用户界面中输入框，输入查询的用户名 2. 鼠标键盘单击“查询”按钮 |
| 期望输出： | 系统显示查询的结果信息 |
| 实际输出： | 正确输出，无误 |
| 测试状态 | 成功通过测试 |
| 测试人： | 杨文娟 |
| 测试时间 | 2016年05月15日 |
| 备注 | 其他模块中的查询界面中根据不同的查询条件，各组件对鼠标键盘相应测试参照本测试用例 |

### 5.3.5 删除用户测试用例

表5-5是删除用户功能测试用例。测试删除用户界面能正确响应事件。如表5-5所示：

表5-5删除用户模块测试用例

|  |  |
| --- | --- |
| 项目名称 | 汽车租赁联盟平台加盟运营中心系统设计与实现---系统管理 |
| 测试ID | CR-TEST-005 |
| 测试说明 | 删除用户界面中的某一条用户信息，各组件均能正确响应鼠标单击事件，删除之后系统更新数据库。 |
| 测试模块 | 删除用户模块 |
| 测试目的 | 验证：删除用户界面各组件均能正确响应时间，完成删除功能 |
| 初始条件 | 运行系统，显示用户界面，选择某一条用户信息 |
| 步骤 | 1. 鼠标单击用户界面中”删除”按钮，弹出删除的提示框。 2. 点击提示框的”确定”按钮。 |
| 期望输出： | 系统删除选中的用户信息，并提示删除操作成功的信息 |
| 实际输出： | 正确输出，无误 |
| 测试状态 | 成功通过测试 |
| 测试人： | 杨文娟 |
| 测试时间 | 2016年05月15日 |
| 备注 | 其他模块中的删除界面中各组件对鼠标键盘相应测试参照本测试用例 |

第6章 结 束 语

通过汽车租赁联盟平台加盟运营中心系统的开发，我从一个只会前端，而Javaweb不会变成了对前端和后台都有很大兴趣的状态，同时使我更加了解项目的开发流程，首先是功能需求、需求分析，深入的了解了实体类、边界类、控制类的作用。同时也学会了画用例图、时序图、概念模型E-R图，然后是功能设计、概要设计、详细设计、数据库设计等，并较深的理解对软件测试的方法、手段，同时认识了一个javaweb项目开发的流程。

通过本次系统的设计开发，不断的进行调试，使系统运行相当良好。但由于时间有限，及个人经验和能力的原因，所以在功能方面还有很多需要完善和添加的地方，在以后的学习中还需要更细更认真的去学习与实践，需要更加深入得了解该技术，能够很好的理解并运用它，开发更好的系统。

总结起来本次毕业设计系统开发主要有以下几个方面的宝贵经验和深刻体会：一是：综合查阅资料是一项重要的学习能力。本次毕业设计中遇到不少的难题，通过网络资源、图书馆资源及咨询专业人员等各种途径解决问题，提高了个人应变能力。在互联网迅速普及的今天，大量的信息资源都是公开的，有效利用这些网络资源成为工作学习的重要方法。二是：软件开发是一种逻辑分析思维要求很高的设计开发工作，在设计开发一个项目前一定要将需求分析完成。一个软件的开发总是要对软件所用到的行业做好相应的需求分析，并且在使用过程中不断地完善其中的功能，去满足人们日益增长的要求。三是：经过毕业设计系统开发的锻炼，团队合作精神日趋更为重要，这对将来的工作和发展都产生不错的影响。这次毕业设计是一次重要的经历，不仅是对大学四年学习的理论基础的实践锻炼，而且应用四年学习的知识，发挥在专业方面的综合素质，也磨砺了一个人的为人处事的正确态度。

汲取经验的同时要明确今后的努力方向，解决存在的问题。本次毕业设计也有不少的不足，主要表现为：一是拓展在Java语言方面的知识面，规范代码的书写风格。这一点其实十分重要，因为编写的程序代码是需要所有能写程序的人读懂的；二是加强对数据库的学习，掌握其中的核心关键技术。三是不断提高编程能力，包括扩大编程语言、开发工具等。

致谢

# 参考文献

[1] 孙卫琴.Tomcat与Java Web 开发技术详解(第二版)[M]. 北京：电子工业出版社,2009.1

[2]刘新．JAVA开发技术大全[M]．清华大学出版社，2009.2

[3] 杨俊生，唐琳，张坤. Java Web 开发实践教程[M]. 北京：清华大学出版社，2010.9

[4]关丽荣,张晓坤(译).JAVA经典实例(第二版)[M]．中国电力出版社,2009.2

[5] 程杰.大话设计模式[M]. 北京：清华大学出版社，2007.12.48-49

[6] 孙卫琴.Java面向对象编程[M]. 北京：电子工业出版社,2006.7.168-172

[7]苗春义．JAVA项目开发全程实录[M]．清华大学出版社,2008.6

[8]Li,M.and H.Wang.A device management system based on JAVAEE Web[M].2009.Wuhan,China.

[9] 李刚.Struts权威指南[M].北京：电子工业出版社,2007.9.156-168

[10] 高洪岩. 至简SSH:精通JavaWeb实用开发技术[M].北京：电子工业出版社,2009.11.99-103

[11] 李刚.轻量级J2EE企业应用实战Struts+Spring+Hibernate整合开发[M]. 北京：北京电子工业出版社, 2007.4.158-159

[12]GaryMak,DanielRubio,JoshLong.SpringRecipes:AProblem-SolutionApproach[M]. Apress,2008.71-78

[13]李钟尉,马文强,陈丹丹.JAVA从入门到精通[M]．清华大学出版社,2008.9

[14] 黄平主编.数据库技术[M].北京：机械工业出版社，2004

[15] Joshua Bloch. Effective Java[M]. Piscataway, N.J: IEEE Press, 2009

# 附录

## 附录A:光盘

光盘说明

1目录结构

根目录名字为：201200834202-杨文娟，根目录下放两个文件夹：分别命名为源码和相关文档。

2文件说明

源码目录文件夹下放置Car-Rent-Join文件（源代码）和数据库文件crj.sql。相关文档目录下放置文件：审批表、任务书、开题报告、中期报告、实习报告、中英文献翻译、论文。