|  |  |
| --- | --- |
| 231 | 123 |
| **PINGDINGSHAN UNIVERSITY** |

**毕业论文(设计)**

**题 目:** 基于SMM的区域拼车系统的

设计与实现

**院(系):** 软件学院

**专业年级:** 软件工程 2017级

**姓 名:** 凌文杰

**学 号:** 171530240

**指导教师:**  张国平 副教授

**2021年3月30日**

原 创 性 声 明

本人郑重声明：本人所呈交的毕业论文，是在指导老师的指导下独立进行研究所取得的成果。毕业论文中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

论文作者签名： 日 期：

关于毕业论文使用授权的声明

本人在指导老师指导下所完成的论文及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属平顶山学院。本人完全了解平顶山学院有关保存、使用毕业论文的规定，同意学校保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅；本人授权平顶山学院可以将本毕业论文的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业论文。如果发表相关成果，一定征得指导教师同意，且第一署名单位为平顶山学院。本人离校后使用毕业论文或与该论文直接相关的学术论文或成果时，第一署名单位仍然为平顶山学院。

论文作者签名： 日 期：

指导老师签名： 日 期：

基于SSM的区域拼车系统的设计与实现

摘 要

在当今社会，人类科技技术正在飞速发展，互联网已经进入人类生活的方方面面，社会进入网络时代，计算机已经成为社会发展的强大动力。传统的乘车模式不断更新完善，但很多乘车模式并不适合城市边缘区域使用。城市边缘远离市中心，相对于市中心来说位于偏远区域，很少车主会在城市边缘，导致城市边缘人们经常打不到车、车价昂贵、出行困难等问题。所以区域拼车系统应运而生，为简化城市边缘人们出行而搭建的一个服务性平台。

当前拼车系统有：滴滴出行、顺风车、高德打车、百度顺风车、同城拼车等，当中有大量注册用户和车主，但是车主和用户集中在市区范围。城市边缘在偏离市区的地方车主和用户都较少，从而车主不愿来，用户拼不到同行人等情况。而针对城市边缘用户的软件更是没有，城市边缘经常会出现高价车拉客。区域拼车系统主要针对城市边缘用户开发，引入信用车主载客，实现车主与乘客共赢。

为满足当代城市边缘人们出行，开发并设计一个区域拼车系统，可以实现城市边缘人们外出时可以与有相同出行意愿的同行者合乘一辆出租车，降低出行费用的同时减少了车辆的空载率，又可以增加车主的收益。符合国家低碳环保的新理念。本系统拥有普通拼车功能的同时。车主可以根据实际情况进行报价，不同车主之间也可以竞价。将高额的出行费用合理的降低的同时也优惠城市边缘人们出行。该系统针对城市边缘人们也有相应的约束，在人与人不信任的当下，每个人都可能放他人鸽子，也可能被放鸽子。那么每人应当缴纳一定的押金，来保障自己的行动力，当其他人无端违规时平台会扣除其押金给同乘的其他人作为补偿，当司机或者其他人违规时采用相同处理模式。这样即保证平台信息的真实有效性，也保证了每个人的诚实守信能力。同时引入信誉模块，对经常违规的人会有相应提示并提示同乘人可能会出现的风险问题。对于不愿与之同乘的人可以发起请出或者自己退出拼车。保证用户使用的流畅性与舒适度。

**关键词**：拼车系统，SSM，J2EE，RestFul

Design and Implementation of Answer Software Based on Android and J2EE Technology

Abstract

In today's society, the Internet has entered all aspects of human life, the society has entered the network era, the computer has become a powerful driving force for social development. Traditional modes of riding are constantly updated and improved, but many modes of riding are not suitable for urban fringe areas. Compared with the city center, they are located in remote areas, and few car owners will be at the edge of the city. As a result, people at the edge of the city often can't get a car, the car price is expensive, travel is difficult and other problems. So the regional ride-sharing system came into being, a service platform to simplify people's travel on the edge of the city.

Currently, ride-sharing systems such as Didi Chuxing, Hitch Che and Gaode Dache have a large number of registered users and owners, but they are concentrated in urban areas. At the edge of the city, there are fewer car owners and users in places that deviate from the urban area, so the car owners are reluctant to come, and the users can not join their peers. There is even less software for users on the urban fringe, where expensive cars are often available. The regional car-sharing system is mainly developed for users at the edge of the city. Credit owners are introduced to carry passengers to achieve a win-win situation between owners and passengers.

The rise of carpooling reduces the travel cost and the empty load rate of vehicles, which can also increase the income of owners. This system has the common carpooling function at the same time. Owners can quote according to the actual situation, and different owners can also bid. The high travel costs can be reasonably reduced, but also preferential urban fringe people travel. The system also has corresponding constraints for people on the edge of the city. In the moment of distrust among people, everyone may stand up others or be stood up by others. In this case, everyone should pay a certain amount of deposit to guarantee their own driving power. When others violate the rules for no reason, the platform will deduct their deposit to others as compensation. When drivers or others violate the rules, the platform will adopt the same treatment mode. In this way, the platform information is guaranteed to be true and effective, and everyone's ability to be honest and trustworthy is also guaranteed. At the same time, the credit module will be introduced, which will prompt the people who often violate the rules and prompt the risks that may occur to the fellow passengers. For those who do not want to ride with them, they can ask out or opt out of the carpool by themselves. Ensure the fluency and comfort of users.

**Keywords:** Car-sharing system , SSM , J2EE , RestFul

1. 绪论

随着计算机技术的推进和发展，J2EE企业级框架的应用也越来越多。针对部分区域拼车难，打车难，打车贵的情况。在网页端实现在线拼车服务，为部分区域人们提供便捷舒适随用随走的服务，后端使用SSM进行搭建并采用RestFul风格，服务器选用MySQL服务器进行数据的存储，数据的传输主要使用Json格式，能快速的适应市场和需求，让开发出软件的生命周期延长。本章主要介绍了项目开发的背景和意义，以及开发拼车软件的主要研究工作，最后对论文的结构进行了介绍。

## 1.1 项目开发背景及意义

在当今社会，人类科技技术正在飞速发展，互联网已经进入人类生活的方方面面，社会进入网络时代，计算机已经成为社会发展的强大动力。传统的乘车模式不断更新完善，但很多乘车模式并不适合城市边缘区域使用。城市边缘远离市中心，相对于市中心来说位于偏远区域，很少车主会在城市边缘，导致城市边缘人们经常打不到车、车价昂贵、出行困难等问题。所以区域拼车系统应运而生，为简化城市边缘人们出行而搭建的一个服务性平台。

为满足当代城市边缘人们出行，开发并设计一个区域拼车系统，可以实现城市边缘人们外出时可以与有相同出行意愿的同行者合乘一辆出租车，降低出行费用的同时减少了车辆的空载率，又可以增加车主的收益。符合国家低碳环保的新理念。本系统拥有普通拼车功能的同时。车主可以根据实际情况进行报价，不同车主之间也可以竞价。将高额的出行费用合理的降低的同时也优惠城市边缘人们出行。该系统针对城市边缘人们也有相应的约束，在人与人不信任的当下，每个人都可能放他人鸽子，也可能被放鸽子。那么每人应当缴纳一定的押金，来保障自己的行动力，当其他人无端违规时平台会扣除其押金给同乘的其他人作为补偿，当司机或者其他人违规时采用相同处理模式。这样即保证平台信息的真实有效性，也保证了每个人的诚实守信能力。同时引入信誉模块，对经常违规的人会有相应提示并提示同乘人可能会出现的风险问题。对于不愿与之同乘的人可以发起请出或者自己退出拼车。保证用户使用的流畅性与舒适度。

## 1.2 项目开发现状

当前拼车系统有：滴滴出行、顺风车、高德打车、百度顺风车、同城拼车等，当中有大量注册用户和车主，但是车主和用户集中在市区范围。城市边缘在偏离市区的地方车主和用户都较少，从而车主不愿来，用户拼不到同行人等情况。而针对城市边缘用户的软件更是没有，城市边缘经常会出现高价车拉客。区域拼车系统主要针对城市边缘用户开发，引入信用车主载客，实现车主与乘客共赢

## 1.3 论文主要工作和结构安排

本论文主要介绍了拼车系统的开发背景、开发与实现的过程，项目实现中遇到的一些技术点和难点如何解决的思路和实现，数据传输和数据存储，系统测试。全文共七章，具体如下安排：

第一章 介绍项目开发背景、开发意义和现阶段开发现状，以及论文的结构安排等。

第二章 介绍该系统使用的相关技术和开发工具还有项目运行环境等。

第三章 对项目功能的需求分析，和项目的性能的需求分析。

第四章 项目的总体设计和各个模块的设计与实现，和数据传输和存储设计与实现。

第五章 详细介绍项目中具体的类如何实现该功能。

第六章 介绍本项目的各种测试及分析测试结果。

第七章 总结本次项目的不足之处和对项目未来的分析与展望。

## 1.4 本章小结

本章首先介绍了本项目开发的背景和意义接着分析了现阶段的开发现状，并且介绍了本次论文的整体结构，以便能按计划顺利完成本次毕业设计。

1. 相关技术及工具介绍

系统整体分两大块，web前端和Web后端。使用MySQL数据库进行数据存储，使用Tomcat作为后台Web服务器。后端开发工具使用IDEA，项目部署在阿里云服务器。

2.1 WEB后端使用的技术点

后端整体采用SSM框架进行搭建，使用Maven工具进行相关类的引用加载维护和版本控制和对Java项目的构建依赖管理，Spring负责数据的维护验证逻辑处理以及数据的传递，实现项目的简单性、可测试性和松耦合性，从而解决应用开发的复杂度。SpringMVC分离了控制器，模型对象，过滤器以及处理程序对象的角色，这种分离让它们更加容易进行控制。 Mybatis 是一款支持普通SQL查询，存储过程和高级映射的优秀持久层框架。

### 2.1.1 Maven

Maven项目管理对象工具，它包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的逻辑。

### 2.1.2 Spring

Spring是Java EE编程领域的一个轻量级开源框架。 Spring框架主要由七部分组成，分别是 Spring Core、 Spring AOP、 Spring ORM、 Spring DAO、Spring Context、 Spring Web和 Spring Web MVC。Spring是一个开源容器框架，它集成各类型的工具，通过核心的Bean factory实现了底层的类的实例化和生命周期的管理。在整个框架中，各类型的功能被抽象成一个个的 Bean，这样就可以实现各种功能的管理，包括动态加载和切面编程。

### 2.1.3 SpringMVC

SpringMVC分离了控制器、模型对象、过滤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。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。使用 Spring 可插入的 MVC 架构，从而在使用Spring进行WEB开发时，可以选择使用Spring的Spring MVC框架或集成其他MVC开发框架，如Struts1(现在一般不用)，Struts 2(一般老项目使用)等等。

### 2.1.3 SpringMVC

Mybatis 是一款支持普通SQL查询，存储过程和高级映射的优秀持久层框架。其消除了几乎所有的JDBC代码和参数的手工设置，以及结果集的检索。Mybatis使用简单的XML或注解用于配置和原始映射，将接口和java的POJOS(Plain Ordinary Java Objects，普通的java对象)映射成数据库中的记录。

2.2 Web前端使用的技术点

Web前端使用React是一个用于构建用户界面JavaScript库，其拥有较高的性能，代码逻辑非常简单,性能出众。TypeScript是微软开发的一个开源的编译语言，通过在JavaScript的基础上添加静态类型定义构建而成。，Umi是一款插件化的企业级前端应用框架。其拥有良好的可扩展性、开箱即用、完备路由等，Ant Design是蚂蚁集团体验技术部经过大量项目实践和总结，逐步打磨出一个服务于企业级产品的设计体系。基于「自然」、「确定性」、「意义感」、「生长性」四大[设计价值观](https://ant.design/docs/spec/values-cn)，通过模块化解决方案，降低冗余的生产成本，让设计者专注于更好的用户体验。

个别技术解释如下：

### 2.2.1 React

React 起源于 Facebook 的内部项目，因为该公司对市场上所有 JavaScript MVC 框架，都不满意，就决定自己写一套，用来架设 Instagram 的网站。由于 React 的设计思想极其独特，属于革命性创新，性能出众，代码逻辑却非常简单。ReactJS是一套JavaScript Web库，由Facebook打造而成且主要用于构建高性能及响应式用户界面。React负责解决其它javascript框架所面对的一大常见难题，即对大规模数据集的处理。能够使用虚拟DOM并在发生变更时利用补丁安装机制只对DOM中的dirty部分进行重新渲染，React得以实现远超其它框架的速度表现。

1.速度很快，它并不直接对DOM进行操作，引入了一个叫做虚拟DOM的概念，安插在javascript逻辑和实际的DOM之间，性能好。

2.跨浏览器兼容。

3.单向数据流，其实reactjs的核心内容就是数据绑定，所谓数据绑定指的是只要将一些服务端的数据和前端页面绑定好，使开发者只关注实现业务。

一切都是component，代码更加模块化，重用代码更容易，可维护性高。

### 2.2.2 TypeScript

TypeScript是微软开发的一个开源的编程语言，通过在JavaScript的基础上添加静态类型定义构建而成。TypeScript通过TypeScript编译器或Babel转译为JavaScript代码，可运行在任何浏览器，任何操作系统。

TypeScript扩展了JavaScript的语法，所以任何现有的JavaScript程序可以运行在TypeScript环境中。TypeScript是为大型应用的开发而设计，并且可以编译为JavaScript。

TypeScript 支持为已存在的 JavaScript 库添加类型信息的头文件，扩展了它对于流行库的支持，如 jQuery，MongoDB，Node.js 和 D3.js 等。这些第三方库的类型定义本身也是开源的，所有开发者都能参与贡献。

TypeScript 增加了代码的可读性和可维护性

1.类型系统实际上是最好的文档，大部分的函数看看类型的定义就可以知道如何使用了。

2.可以在编译阶段就发现大部分错误。

3.增强了编辑器和 IDE 的功能，包括代码补全、接口提示、跳转到定义、重构等。

### 2.2.3 Umi

UmiJS是一款插件化的企业级前端应用框架。基于路由，支持类似next.js的常规路由和各种高级路由功能，例如路由级别的按需加载。

它具有完整的插件系统，涵盖了从源代码到构建的每个生命周期。 Umi能够满足各种功能扩展和业务需求，目前Umi在社区和公司内部拥有近50多个插件。 Umi是蚂蚁金服的基本前端框架，已直接或间接为3000多个应用程序提供服务，包括Java、节点、移动应用程序、混合应用程序、纯前端资产应用程序、CMS应用程序等等。

1.可扩展性，Umi 实现了完整的生命周期，并使其插件化，Umi 内部功能也全由插件完成。此外还支持插件和插件集，以满足功能和垂直域的分层需求。

2.开箱即用，Umi 内置了路由、构建、部署、测试等，仅需一个依赖即可上手开发。并且还提供针对 React 的集成插件集，内涵丰富的功能，可满足日常 80% 的开发需求。

3.企业级，经蚂蚁内部 3000+ 项目以及阿里、优酷、网易、飞猪、口碑等公司项目的验证，值得信赖。

4.大量自研，包含微前端、组件打包、文档工具、请求库、hooks 库、数据流等，满足日常项目的周边需求。

5.完备路由，同时支持配置式路由和约定式路由，同时保持功能的完备性，比如动态路由、嵌套路由、权限路由等等。

6.面向未来，在满足需求的同时，我们也不会停止对新技术的探索。比如 modern mode、webpack@5、自动化 external、bundler less 等等。

### 2.2.4 Ant Design

Ant Design是蚂蚁集团体验技术部经过大量项目实践和总结，逐步打磨出一个服务于企业级产品的设计体系。基于「自然」、「确定性」、「意义感」、「生长性」四大设计价值观，通过模块化解决方案，降低冗余的生产成本，让设计者专注于更好的用户体验。

1.原则，基于价值观衍生出的一般设计技巧，包括如何高效组织信息、帮助建立用户与界面的互动等原则性要求。

2.全局规则，通过规范常见的互动行为，包括导航、数据录入、数据展示、反馈、操作、帮助等，呈现组件元素的标准用法和弹性空间，并了解如何将他们组织在一起来创建良好的体验。

3.模板文档，基于蚂蚁中台的设计经验，我们总结了表单、列表、数据可视化、详情页、工作台、异常、结果、编辑器几类页面的典型解决方案，介绍每类页面的设计目标、设计技巧以及典型页面内容等，帮助广大设计者可以快速开始构建界面。

2.3 本章小结

本章主要介绍了此项目中所用到的相关技术，包括Web后端采用的版本控制工具和项目所使用的框架，以及Web前端使用的React框架和Umi项目中使用的框架。

1. 需求分析

需求分析主要描述了用户对软件应用的期望和使用需求，了解用户的需求后才能使软件开发有一个明确的方向，根据用户需求设计应用的功能。

3.1 业务操作流程

用户使用手机、电脑等上网设备打开浏览器进入拼车网页，可查询的拼车信息，不仅能查看全部的拼车信息，也能进行不同条件的搜索。用户可以单独搜索始发地，或者目的地，也可以综合进行搜索，搜索还分为模糊搜索和精确搜索两种模式，对记不清或者不能准确描述始发地或者目的地的用户提供极大的遍历，使用户更方便快速的找到自己需要的拼车信息。如果看到与自己符合的拼车信息，可以查看详细信息，如已经拼车的用户进行的留言，信誉值、性别等相关信息，不过必要的信息进行隐藏处理。用户想加入拼车，如果用户没有登录，会提示用户进行登录，如果已经登录可以加入拼车，可以查看队友联系方式，并进行自己的留言，比如：这次出行会随身携带一个皮箱，可能会占后备箱位置。防止多人携带大量物品导致车辆无法容纳。

用户中途也可以退出拼车，拼车成功后不可退出拼车，防止用户有意外情况导致拼车期间临时改道的情况，以应对多种不同的环境下导致的问题。给予不同用户拼车的良好的体验。

后台是管理员使用的，主要维护平台的稳定性，与不同用户之间的意见，平台搭建有意见反馈模块，可根据不同的情况进行相应的调整，使得拼车系统更加的人性化，更加贴切人们日常需求。

拼车系统中也有司机模块，司机可以根据不同的路线进行选择不同的拼车，进行竞价，司机可查看所有的拼车信息的同时，可以查看已经拼车成功的信息，并进行竞价拉客，竞价过程在平台中竞价，只有竞价成功后才可以查看拼车人的联系方式，保证来了用户不会被恶意或者竞价原因骚扰，司机参与竞价后，拼车用户可以根据自己的情况选择是否同意司机载客，选择权依然在用户手中，给予用户多种出行方式。

同时平台视情况而定，可以决定是否开启保证金模式，加入拼车前平台收取一定金额的保证金，以防止他人拼车成功后联系不上或者其他原因爽约。

3.2 系统设计目标

本系统基于React、TypeScript、UmiJS、Ant Design、Spring、SpringMvc、Mybatis技术所开发的区域拼车系统，采用这些框架主要是为了快速开发出系统稳定、设计合理规范、扩展性强并具有兼容性的软件。在此基础上力求软件设计的优美，尽可能的提高系统运行时的稳定性、高效性、异常处理性给用户更好的使用体验，帮助用户更快的获取需要的拼车信息。

系统的具体功能实现的目标如下：

1. 考虑到出行的安全性，登录模式采用QQ、微信、支付宝登录，确保每一个用户都是真实存在并可追踪的。同时系统可采用保证金模式，使每一个拼车用户使用支付宝支付相应的金额，利用支付宝实名信息的影响力来为用户的拼车增加安全性。保证每一位拼车的用户都是真实存在并有迹可循。
2. 由于借助第三方权威安全性措施，取消管理员为用户创建账号的权限，管理员可根据用户的行为对用户进行信誉值的处理，并对于严重违反使用规则的用户可以封禁账号操作。如果开启保证金模式，可以通知本人保证金使用到其他人的补偿事宜。
3. 服务器开启用户登录检测，详细的记录每个用户的登录操作痕迹，包括用户每次访问的Ip地址，使用文件存储用户操作，并可实时监控。
4. 对于项目使用扩展性良好的框架进行编写，专注一点的同时，兼容更多的功能，使用户随用随走的同时，也会经常使用此网站。以增加用户与网站的亲密度。

3.3 性能需求

对于项目未来的发展虽然有信心，但是对于初期时项目而言，较高的硬件性能带来的可能是性能的浪费。对于选取性能合适的服务器而言，基础服务器基本满足前期用户需求，因此选取1核2G的服务器运行此程序。

对于软件而言，性能主要体现在用户体验，也就是用户与系统每次交互所使用的时间，我们决定选用懒加载方式，加载用户需要的数据，减少数据传输来增加相应速度。同时要解决页面加载过程中带来的抖动问题。

随着系统影响力的扩大，使用的人数越多。会使数据执行的效率降低，我们应该在后端简化业务逻辑，前端控制资源文件加载的优先级，减少重排（Reflow），并且利用浏览器缓存将网络资源存储在本地，提高软件的效率。

3.4 用户需求描述

区域拼车系统提供三种管理权限：①乘客用户：乘客用户进行查询拼车、加入拼车、发布拼车、退出拼车、请出成员、举报成员、举报车主、选择车主等功能。实现城市边缘人们平常拼车行为，达到陌生人之间公乘一辆车，平均车费的同时也降低车辆的空载率，实名制的情况下可以更安全的出行，也可以通过此平台结交更多目的、爱好相同的小伙伴。②车主用户：车主可以进行查询拼车信息、对发布的拼车信息进行竞价拉客，退出竞价、联系乘客、举报乘客等拉客目的地相对集中更容易拉到更多人。③管理员审核维护：该部分主要是对网站参与的人员信息进行管理，包括账户的增删改查，同时也有对用户的举报进行核实，处理。管理员在系统中有全部权限，具有监管车主和用户的使用情况。

系统管理员：管理员维护着平台的稳定与和谐，对于一切不和谐的事宜，管理员有义务处理，对于不同类型的账号，管理员拥有账号信誉值的提升和降低甚至于账号的封禁，不同用户之间的权限，管理员可以将用户权限进行相应的修改，对于司机用户，首先登陆权限是普通用户，可以申请为车主用户，通过车主认证后，不仅拥有普通用户权限，还拥有车主权限。同时对车主的信息进行维护。对于车主和用户之间、用户和用户之间的信息进行维护，对于恶意用户进行相应处理。致力于维护和谐友善的拼车环境，让人与人之间的信任重建、同时增强每个人的责任感。

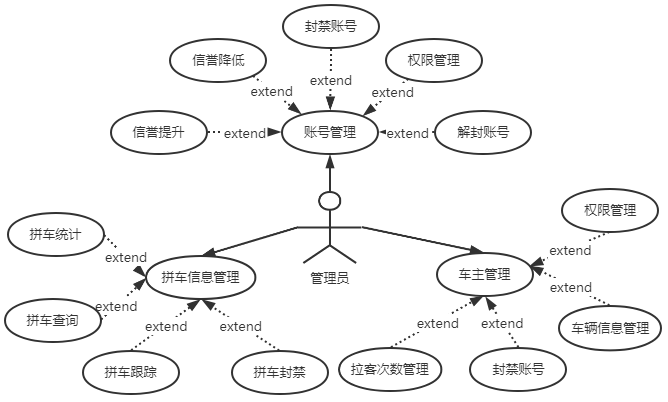


图3-1 系统管理员用例图

系统用户：用户登录后，可以对自己的信息进行相应的修改，实时更新自己的最新信息，以保证拼车后与同乘人之间可以相互联系。用户选择暴露的信息也是多样性的，手机、QQ或者微信以至于邮箱账号。对于人们经常用的社交软件都有涉及。能充分的照顾到每一个人。

用户对自己的历史拼车信息也可以进行查看，对于已经拼车完成的订单，用户仅可以查看拼车情况，对于同行人不再有查看联系方式的权限，也是对每个人隐私的一种保护，保护每一个人的隐私，让每一次拼车都没有后顾之忧。

用户可以查看当前拼车的信息，以及拼车中每个人的联系方式，方便拼车人之间相互交流。没有参与拼车的用户可以进行搜索拼车信息，找寻合适自己的拼车加入。

对于拼车的需求，用户没有找到合适的拼车信息，也可以自己创建拼车信息，发起拼车等待有相同意愿的人加入拼车

如果拼车有临时情况，用户可以申请发起退出拼车，用户退出拼车后，拼车信息再次更新，方便其他有意愿的人加入拼车。

拼车过程中和拼车时间到或者是拼车完成时，用户可以查看与自己拼车人的联系方式。进行相应的联系，可以一起探讨出行时间和出行路线规划，用户们自己商定出行，平台保证用户的真实性。让每一次出行都有迹可循。

选择车主，对于发起的拼车订单，车主可以查看其出发地和目的地，选择自己心仪的路线进行竞价，报出自己的价格让用户进行选择是否同意车主承接此单，如果没有车主愿意接单或者用户不选择车主，用户可自行联系车辆。选择自己愿意的出行方式进行出行需要。

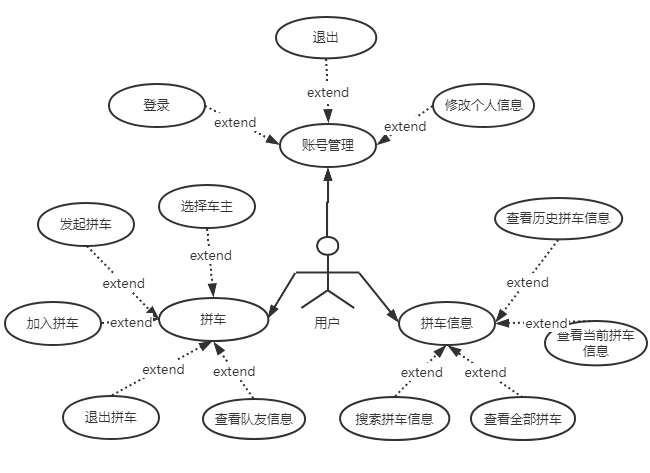


图3-2 系统用户用例图

车主：车主可以查询拼车信息，并对于心仪的路线进行竞价，报出自己的价格让用户选择是否愿意让自己接单，司机参与竞价后用户能查看到司机的报价，以及司机在竞价中的留言信息，以供用户进行选择。

退出竞价：车主竞价后，在用户没有选择自己承接此次出行之前，因为一些原因不能承接此次出行，司机可以选择退出竞价，退出后用户不可选择司机进行承接，用户仅仅可看到车主报价情况。

竞价成功后拉客：车主参与竞价后用户选择此次出行任务车主进行承接，此时车主可以看到用户的联系信息，跟用户进行交流并确定相关事宜进行此次行程。用户选择用户后车主可以接收到通知，同时，为避免车主因时间原因违约，车主在此期间参与的竞价全部取消，车主可再次发起申请。

竞价失败后：可以查看到自己竞价失败的信息，根据竞价失败可以总结失败原因。

车辆信息更新：车主刚刚认证时会有车主的车辆信息，以及车主的部分信息，车主可以及时的更新自己的信息，防止车主因其他原因相应的信息没有及时更新导致损失，车主更新自己的信息也是对乘客的负责。避免车辆信息不一致导致出现误会。

车辆信息维护： 车主车辆信息发生更变时车辆信息要进行相应的维护，以便用户乘车时可以方便的联系并确定车辆，以免车辆信息不一致导致用户与车主之间产生怀疑，同时平台也不允许车主的相关信息不一致，如果发现不一致并查实后会对车主进行相应的处罚，并责令车主进行更新相应的信息。

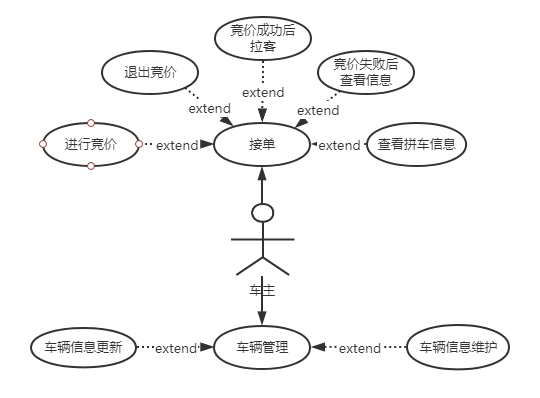


图3-3 系统车主用例图

3.5 用例描述

用户登录用例描述如表3-4

表3-4登录用例描述

|  |  |
| --- | --- |
| **用例名** | 登录 |
| **用例描述** | 系统管理员、一般用户利用该用例登录系统，通过身份认证后展示全部的操作界面 |
| **参与者** | 系统管理员、一般用户、车主用户 |
| **涉众** | 系统管理员、一般用户、车主用户：通过身份验证，展示相应的页面与操作 |
| **相关用例** | 无 |
| **前置条件** | 无 |
| **后置条件** | 如果登录成功，则展示相应的页面与操作 |
| **基本事件流**  （1）用例起始于用户进入系统首页  （2）系统拉起QQ、微信或支付宝的授权登录接口  （3）用户授权（D-1）  （4）系统验证用户名信息，同意用户登录系统（A-1），并返回登陆code  （5）系统验证code的真实性，验证通过后即表示登录成功，展示全部的操作界面，不通过表示登录失败，进入第（1）步 | |
| **备选事件流**  A-\*用户在进入登录界面随时都可以终止用例  （1）系统提示关闭信息  （2）用户可以结束该用例或执行其他的操作  （3）用户授权阶段取消授权，或者不授权  （4）用户关闭系统页面  A-1 用户拒绝授权或者验证码超时  （1）系统提示用户已拒绝授权或者验证码超时，并进入第（1）步  （2）用户可以重新验证（B-1），也可以选择结束该用例 | |
| **补充约束-业务规则**  B-1 验证二维码每10s刷新一次  **补充约束-数据需求**  D-1用户要求：必须使用QQ、微信或支付宝已注册的账号  **补充约束-非功能需求**  安全性：使用QQ、微信或支付宝官方认证 | |

普通用户创建拼车信息的用例描述如表3-2所示。

表3-2创建拼车信息用例描述

|  |  |
| --- | --- |
| **用例名** | 创建拼车信息 |
| **用例描述** | 一般用户可以利用该用例创建拼车信息 |
| **参与者** | 一般用户 |
| **涉众** | 一般用户：需要拼车时发布新的拼车信息 |
| **包含的用例** | 无 |
| **前置条件** | 一般用户登录系统 |
| **后置条件** | 发布的拼车信息正确的保存到系统中 |
| **基本事件流**  （1）用例起始于一般用户进入拼车页面，准备发布拼车信息  （2）包含“查询拼车信息”来获取全部拼车信息的列表（A-1）（D-1）  （3）显示查询拼车的所有拼车信息（A-3）（D-2）  （4）系统显示填写用户的拼车信息表单页（B-1）（D-3）  （5）一般用户填写拼车信息  （6）系统生成完整的拼车信息并返回保存后的拼车信息（A-2），用例结束 | |
| **备选事件流**  A-\*一般用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）一般用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 拼车信息列表获取失败  （1）系统查询拼车列表获取失败  （2）一般用户可以选择再次查询，也可以结束该用例  A-2 发布拼车信息失败  （1）系统显示保存用户发布拼车信息失败  （2）一般用户可以选择查看原因，并更改后再次发布重试，也可以结束该用例，不进行拼车信息的发布。  A-3 拼车信息获取失败  （1）系统查询拼车信息失败  （2）一般用户可以选择再次获取，也可以结束该用例 | |
| **补充约束-业务规则**  B-1每次只允许发布一条拼车数据  **补充约束-数据需求**  D-1显示的拼车信息包括：拼车ID、拼车始发地、拼车目的地、拼最早车出发时间、拼车最晚出发时间、此次拼车人数、此次拼车差多少人、拼车用户的留言、拼车用户的信誉值，拼车发布时间等。  D-2 显示的拼车信息包括用户头像、用户名、拼车留言、信誉值  D-3 表单信息包括：拼车具体内容 | |

普通用户加入拼车信息的用例描述如表3-2所示。

表3-2加入拼车信息用例描述

|  |  |
| --- | --- |
| **用例名** | 加入拼车信息 |
| **用例描述** | 一般用户可以利用该用例加入拼车信息 |
| **参与者** | 一般用户 |
| **涉众** | 一般用户：需要拼车时加入同行的拼车信息 |
| **包含的用例** | 用户查询拼车信息，对拼车信息进行筛选，并查看拼车的详细信息。 |
| **前置条件** | 一般用户登录系统 |
| **后置条件** | 加入的拼车信息正确的保存到系统中 |
| **基本事件流**  （1）用例起始于一般用户进入拼车页面，准备拼车  （2）包含“查询拼车信息”来获取全部拼车信息的列表（A-1）（D-1）  （3）显示查询拼车的所有拼车信息（A-3）（D-2）  （4）系统显示填写用户的加入拼车信息表单页（B-1）（D-3）  （5）一般用户填写加入拼车信息  （6）系统生成完整的拼车信息并返回更新后的拼车信息（A-2），用例结束 | |
| **备选事件流**  A-\*一般用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）一般用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 拼车信息列表获取失败  （1）系统查询拼车列表获取失败  （2）一般用户可以选择再次查询，也可以结束该用例  A-2 加入拼车信息失败  （1）系统显示保存用户加入拼车信息失败  （2）一般用户可以选择查看原因，并更改后再次发布重试，也可以结束该用例，不进行加入拼车信息的发布。  A-3 拼车信息获取失败  （1）系统查询拼车信息失败  （2）一般用户可以选择再次获取，也可以结束该用例 | |
| **补充约束-业务规则**  B-1每次只允许加入一条拼车数据  **补充约束-数据需求**  D-1显示的拼车信息包括：拼车ID、拼车始发地、拼车目的地、拼最早车出发时间、拼车最晚出发时间、此次拼车人数、此次拼车差多少人、拼车用户的留言、拼车用户的信誉值，拼车发布时间等。  D-2 显示的拼车信息包括用户头像、用户名、拼车留言、信誉值  D-3 表单信息包括：拼车具体内容 | |

普通用户查询拼车信息的用例描述如表3-2所示。

表3-2查询拼车信息用例描述

|  |  |
| --- | --- |
| **用例名** | 查询拼车信息 |
| **用例描述** | 一般用户可以利用该用例查询拼车信息 |
| **参与者** | 全部用户 |
| **涉众** | 全部用户：需要拼车时查询同行的拼车信息 |
| **包含的用例** | 无 |
| **前置条件** | 无 |
| **后置条件** | 查询的拼车信息正确的返回给用户 |
| **基本事件流**  （1）用例起始于全部用户进入拼车页面，准备查询拼车  （2）包含“查询拼车信息”来获取全部拼车信息的列表（A-1）（D-1）  （3）显示查询拼车的所有拼车信息（A-3）（D-2）  （4）系统显示用户的加入拼车信息表单页（B-1）（D-3）  （5）全部用户查看拼车信息详细信息  （6）系统查询完整的拼车信息并返回查询到的拼车信息（A-2），用例结束 | |
| **备选事件流**  A-\*全部用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）全部用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 拼车信息列表获取失败  （1）系统查询拼车列表获取失败  （2）全部用户可以选择再次查询，也可以结束该用例  A-2 查看拼车详细信息失败  （1）系统显示查询用户拼车信息的详细信息失败  （2）全部用户可以选择查看原因，并更改后再次查询重试，也可以结束该用例，不进行查看拼车详细信息的发布。  A-3 拼车信息获取失败  （1）系统查询拼车信息失败  （2）全部用户可以选择再次获取，也可以结束该用例 | |
| **补充约束-业务规则**  B-1用户只能查看当前加入拼车队友的联系方式，未加入的联系方式进行打码  **补充约束-数据需求**  D-1显示的拼车信息包括：拼车ID、拼车始发地、拼车目的地、拼最早车出发时间、拼车最晚出发时间、此次拼车人数、此次拼车差多少人、拼车用户的留言、拼车用户的信誉值，拼车发布时间等。  D-2 显示的拼车信息包括用户头像、用户名、拼车留言、信誉值  D-3 表单信息包括：拼车具体内容 | |

普通用户退出拼车信息的用例描述如表3-2所示。

表3-2退出拼车信息用例描述

|  |  |
| --- | --- |
| **用例名** | 退出拼车信息 |
| **用例描述** | 一般用户可以利用该用例退出拼车信息 |
| **参与者** | 一般用户 |
| **涉众** | 一般用户：拼车后有事情退出拼车 |
| **包含的用例** | 无 |
| **前置条件** | 用户登录并加入拼车 |
| **后置条件** | 返回给用户退出拼车后更新的拼车信息 |
| **基本事件流**  （1）用例起始于一般用户进入自己拼车页面，准备退出拼车  （2）包含“查询当前加入拼车信息”来获取当前加入的拼车信息（A-1）（D-1）  （3）显示当前加入的拼车信息（A-3）（D-2）  （4）系统显示用户的加入拼车信息表单页（B-1）（D-3）  （5）一般用户发起退出拼车  （6）系统更新拼车信息并返回更新后的拼车信息（A-2），用例结束 | |
| **备选事件流**  A-\*一般用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）一般用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 拼车信息列表获取失败  （1）系统查询当前拼车列表获取失败  （2）一般用户可以选择再次查询当前加入的拼车信息，也可以结束该用例  A-2 查看拼车详细信息失败  （1）系统显示查询用户拼车信息的详细信息失败  （2）一般用户可以选择查看原因，并更改后再次查询重试，也可以结束该用例，不进行查看拼车详细信息的发布。  A-3 退出拼车信息失败  （1）系统更新用户提出拼车信息失败  （2）一般用户可以选择再次发起退出拼车，也可以结束该用例 | |
| **补充约束-业务规则**  B-1用户退出拼车后不能重复加入。  **补充约束-数据需求**  D-1显示的拼车信息包括：拼车ID、拼车始发地、拼车目的地、拼最早车出发时间、拼车最晚出发时间、此次拼车人数、此次拼车差多少人、拼车用户的留言、拼车用户的信誉值，拼车发布时间等。  D-2 显示的拼车信息包括用户头像、用户名、拼车留言、信誉值  D-3 表单信息包括：拼车具体内容 | |

车主更新车辆信息的用例描述如表3-2所示。

表3-2车主更新车辆信息用例描述

|  |  |
| --- | --- |
| **用例名** | 更新车辆信息 |
| **用例描述** | 车主用户可以利用该用例更新车辆信息 |
| **参与者** | 车主用户 |
| **涉众** | 车主用户：更新车辆信息 |
| **包含的用例** | 无 |
| **前置条件** | 车主用户登录 |
| **后置条件** | 返回给车主用户车辆信息更新后的车辆信息 |
| **基本事件流**  （1）用例起始于车主用户进入信息管理页面，准备更新车辆信息  （2）包含“查询当前车辆信息”来获取当前的车辆信息（A-1）（D-1）  （3）显示当前的车辆信息（A-3）（D-2）  （4）系统显示车主用户更新车辆信息的表单页（B-1）（D-3）  （5）车主用户发起更新车辆信息  （6）系统更新车辆信息并返回更新后的车辆信息（A-2），用例结束 | |
| **备选事件流**  A-\*车主用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）车主用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 车辆信息获取失败  （1）系统查询当前车主的车辆查询获取失败  （2）车主用户可以选择再次查询当前车辆信息，也可以结束该用例  A-2 更新车辆信息失败  （1）系统更新车辆信息失败  （2）车主用户可以选择查看原因，并更改后再次更新，也可以结束该用例，到意见反馈处反馈原因。  A-3 显示车辆信息失败  （1）系统显示车主用户更新的拼车信息失败  （2）车主用户可以选择再次发起更新拼车信息，也可以结束该用例 | |
| **补充约束-业务规则**  B-1车主用户更新车辆信息后不可恢复。  **补充约束-数据需求**  D-1显示的车辆信息包括：车主ID、接单数量、车牌号码车辆颜色、车的座位数等车辆相关信息。  D-2 显示的车辆信息包括用户头像、用户名  D-3 表单信息包括：车辆详细信息 | |

车主抢单信息的用例描述如表3-2所示。

表3-2车主抢单信息用例描述

|  |  |
| --- | --- |
| **用例名** | 抢单信息 |
| **用例描述** | 车主用户可以利用该用例进行抢单 |
| **参与者** | 车主用户 |
| **涉众** | 车主用户：发起抢单。一般用户：用户接收到抢单信息 |
| **包含的用例** | 查询拼车信息 |
| **前置条件** | 车主用户登录 |
| **后置条件** | 返回给车主用户发起抢单后生成的抢单信息 |
| **基本事件流**  （1）用例起始于车主用户进入查询拼车信息页面，准备发起抢单申请  （2）包含“查询当前拼车信息”来获取当前的拼车信息（A-1）（D-1）  （3）显示当前的拼车信息（A-3）（D-2）  （4）系统显示车主用户申请抢单后生成信息的表单页（B-1）（D-3）  （5）车主用户发起抢单信息  （6）系统生成抢单信息并返回生成抢单的信息（A-2），用例结束 | |
| **备选事件流**  A-\*车主用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）车主用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 查询拼车信息获取失败  （1）系统查询当前拼车的信息列表获取失败  （2）车主用户可以选择再次查询当前拼车信息，也可以结束该用例  A-2生成抢单信息失败  （1）系统生成抢单信息失败  （2）车主用户可以选择查看原因，并更改后再次发起申请，也可以结束该用例，到意见反馈处反馈原因。  A-3 显示当前拼车信息失败  （1）系统显示车主用户查看拼车详细信息失败  （2）车主用户可以选择再次发起请求获取拼车详细信息，也可以结束该用例 | |
| **补充约束-业务规则**  B-1车主用户同一时间段只能申请一个拼车。  **补充约束-数据需求**  D-1申请抢单后拼车中显示车辆信息包括：车主ID、接单数量、车牌号码车辆颜色、车的座位数等车辆相关信息。  D-2 显示的用户信息包括用户头像、用户名  D-3 表单信息包括：拼车以及抢单详细信息 | |

车主退出抢单信息的用例描述如表3-2所示。

表3-2车主退出抢单信息用例描述

|  |  |
| --- | --- |
| **用例名** | 退出抢单信息 |
| **用例描述** | 车主用户可以利用该用例退出抢单 |
| **参与者** | 车主用户 |
| **涉众** | 车主用户：发起退出抢单。一般用户：用户接收到退出抢单信息 |
| **包含的用例** | 查询拼车信息，查询自己发起的抢单信息 |
| **前置条件** | 车主用户登录并已经申请抢单 |
| **后置条件** | 返回给车主用户发起退出抢单后生成的退出抢单信息 |
| **基本事件流**  （1）用例起始于车主用户进入查询自己抢单的信息页面，准备发起退出抢单申请  （2）包含“查询当前抢单信息”来获取当前发起的抢单信息（A-1）（D-1）  （3）显示当前已经申请抢单的信息（A-3）（D-2）  （4）系统显示车主用户申请退出抢单后生成信息的表单页（B-1）（D-3）  （5）车主用户发起退出抢单申请  （6）系统更新抢单信息并返回更新后退出抢单的信息（A-2），用例结束 | |
| **备选事件流**  A-\*车主用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）车主用户可以结束该用例或执行其他的操作  （3）用户关闭浏览页面或者退出登录  A-1 当前抢单信息获取失败  （1）系统查询当前申请抢单的信息获取失败  （2）车主用户可以选择再次查询当前发起的抢单信息，也可以结束该用例  A-2更新退出抢单信息失败  （1）系统更新退出抢单信息失败  （2）车主用户可以选择查看原因，并更改后再次发起申请，也可以结束该用例，到意见反馈处反馈原因。  A-3 显示当前退出抢单申请失败  （1）系统显示车主用户申请退出抢单信息失败  （2）车主用户可以选择再次发起请求退出当前抢单，也可以结束该用例 | |
| **补充约束-业务规则**  B-1车主用户只能在还在抢单中发起退出抢单，抢单成功后不可发起退出。  **补充约束-数据需求**  D-1退出申请抢单后拼车中显示车辆信息包括：车主ID、接单数量、车牌号码车辆颜色、车的座位数等车辆相关信息。  D-2 显示的用户信息包括用户头像、用户名  D-3 表单信息包括：拼车以及抢单详细信息 | |

4 总体设计

在需求分析阶段我们对拼车系统的开发做了详细的分析工作，确立了整体架构与功能。本章主要进行系统的总体设计，包括系统结构设计、软件功能设计、数据库设计和页面设计。

## 4.1 系统结构设计

系统后台主体框架采用Spring + SpringMVC + Mybatis框架搭建。系统使用SpringMVC作为构建WEB应用程序的全功能MVC模块，该框架通过策略接口，Spring框架的高度可配置性、分离了控制器、模型对象、过滤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。MVC作为WEB项目开发的核心环节，控制器、视图、JavaBean数据封装共同组成了MVC。系统使用Spring框架，Spring的框架优点在于轻量、控制反转、面向切面、容器、框架、MVC等特征。能够编写出干净、可管理、易于测试的代码。它们同时也为Spring中的各个模块提供了基础支持。持久层选择使用Mybatis框架开发，Mybatis不仅支持普通SQL查询，也是存储过程和高级映射的优秀持久层框架，Mybatis使用简单的XML或注解用于配置和原始映射，将接口和Java的POJOS（Plain Ordinary Java Object，普通Java对象）映射成数据库中的记录。

系统架构图，如4-1所示：

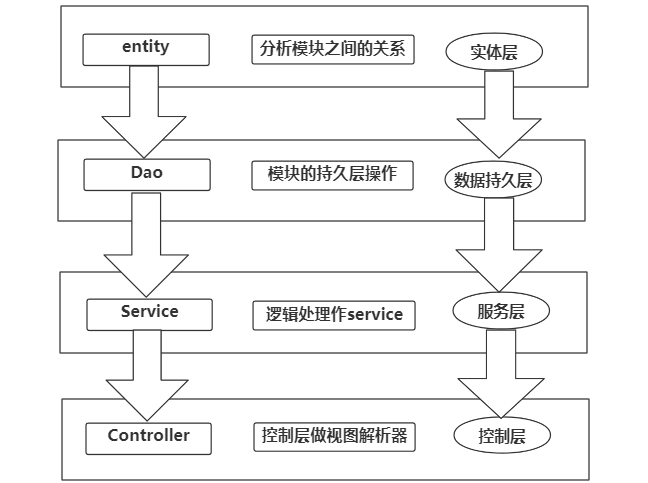


图4-1 系统架构图

系统前端主要使用的是React+Ant Design+TypeScript+Umi等多个前端技术搭建，React主要用于构建高性能及响应式用户界面。能够使用虚拟DOM并在发生变更时利用补丁安装机制只对DOM中的dirty部分进行重新渲染。用户基于页面进行的所有向服务器发送的请求，都是Axios异步请求的模式，不仅可以提高整个系统的通用性，而且也能大大降低服务器端的压力。

前台通过Axios进行数据访问，后台以Json的数据格式进行响应，数据交互示意图，如图4-2所示。

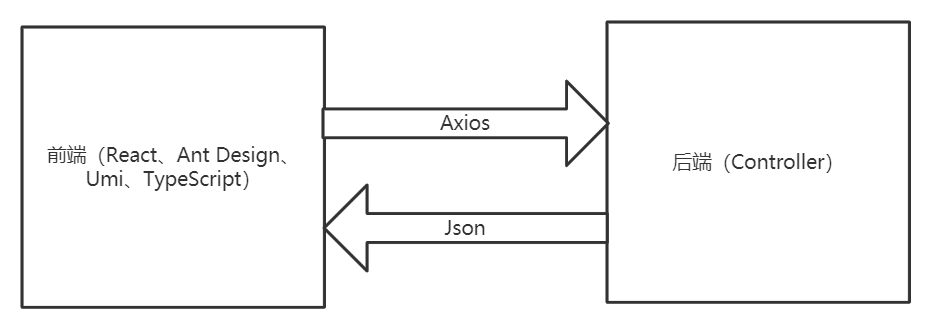


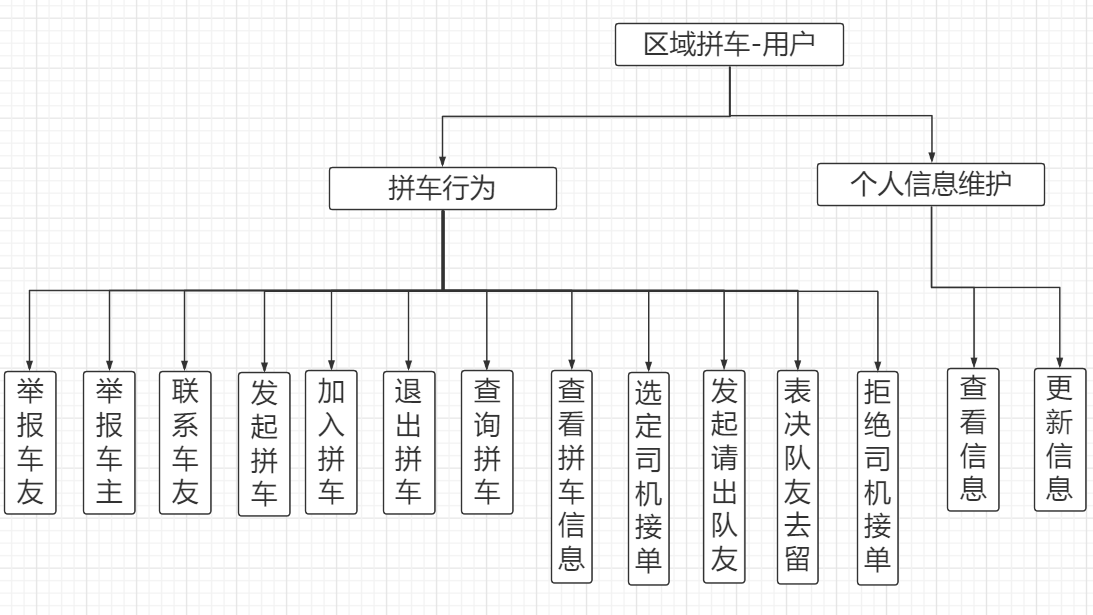
图4-2 数据交互示意图

## 4.2 软件功能设计

设计并开发一个区域拼车系统，该系统主要包括三大角色：①乘客用户：乘客用户进行查询拼车、加入拼车、发布拼车、退出拼车、请出成员、举报成员、举报车主、选择车主等功能。实现城市边缘人们平常拼车行为，达到陌生人之间公乘一辆车，平均车费的同时也降低车辆的空载率，实名制的情况下可以更安全的出行，也可以通过此平台结交更多目的、爱好相同的小伙伴。②车主用户：车主可以进行查询拼车信息、对发布的拼车信息进行竞价拉客，退出竞价、联系乘客、举报乘客等拉客目的地相对集中更容易拉到更多人。③管理员审核维护：该部分主要是对网站参与的人员信息进行管理，包括账户信息的变动，权限的更改，信誉值包括封禁和解封用户账号，同时也有对用户的举报进行核实，处理。管理员在系统中有全部权限，具有监管车主和用户的使用情况。

接下来我们分别阐述系统中各个用户角色中每个管理模块的具体功能。

乘客用户：乘客用户进行查询拼车、加入拼车、发布拼车、退出拼车、请出成员、举报成员、举报车主、选择车主等功能。实现城市边缘人们平常拼车行为，达到陌生人之间公乘一辆车，平均车费的同时也降低车辆的空载率，实名制的情况下可以更安全的出行，也可以通过此平台结交更多目的、爱好相同的小伙伴。

图4-3 用户软件功能结构图

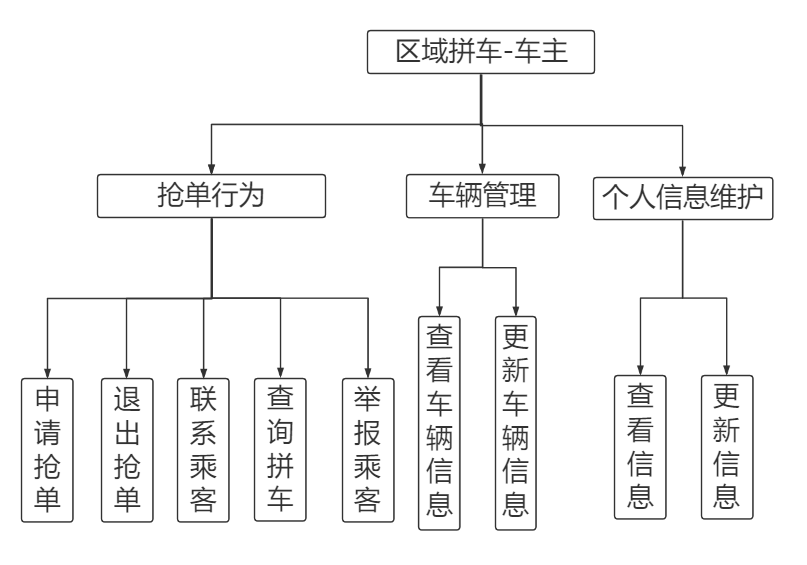
车主用户：车主可以进行查询拼车信息、对发布的拼车信息进行竞价拉客，退出竞价、联系乘客、举报乘客等拉客目的地相对集中更容易拉到更多人。

图4-4车主软件功能结构图

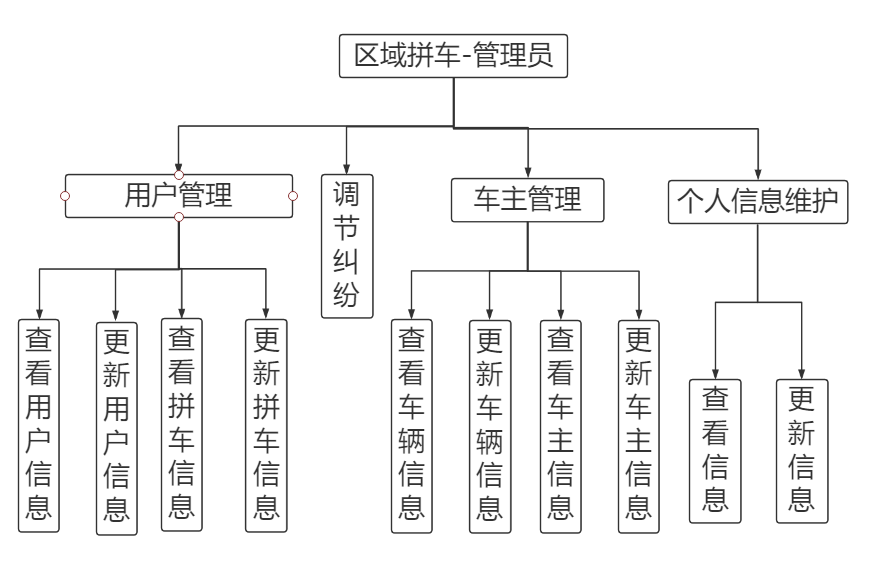
管理员审核维护：该部分主要是对网站参与的人员信息进行管理，包括账户信息的变动，权限的更改，信誉值包括封禁和解封用户账号，同时也有对用户的举报进行核实，处理。管理员在系统中有全部权限，具有监管车主和用户的使用情况。

图4-5管理员软件功能结构图

## 4.3 数据库设计

数据库设计在整个软件开发的起到的举足轻重的作用，数据库与需求是相辅相成的，我经常把软件开发比作汽车制造。汽车制造会经过图纸设计，模型制作，样车制造，小批量试生产，最后是批量生产等步骤。整个过程环环相扣，后一过程是建立在前一过程正确的前提基础之上的。如果在图纸设计阶段发现了一个纰漏，我们可以重新进行图纸设计，如果到了样车制造阶段发现这个错误，那么我们就要把从图纸设计到样车制造的阶段重来，越到后面发现设计上的问题，所付出的代价越大，修改的难度也越大。通常开发人员在设计数据库时有两个重要方面：分析数据库、设计和实现数据库。

### 4.3.1 数据库概念结构设计

在数据库的开发之前，我们需要思考使用系统的角色都具有什么属性（即概念模型），只有明确了用户的真正需求，我们才能真正的设计推导出概念模型。

概念模型的主要特点是：

1.对现实世界的一个真实模型，能真实、充分地反映现实世界中事物和事物之间的联系，以此来满足用户对数据的处理要求。

2.易于理解，哪怕是不熟悉计算机的用户，也能简单的理解并交换意见，沟通十分方便。数据库设计成功的关键与否很大程度上取决于用户是否积极参与。

3.易于更改，当应用的场景和用户的需求发生改变的时候，概念模型能够更加容易的修改或扩充。

4.易于向关系、网状、层次等各种数据模型转换。

相比于数据模型，概念模型更加独立、稳定、和抽象。我们将使用E-R图来

描述概念模型。

根据区域拼车系统需求分析，可以确定开发此系统平台被用户分为三大类：管理员用户、车主用户、一般用户。每一种用户所属权限均有不同，涉及的事件也有很大的不同。通过每种用户的需求我们最终确定此系统包括以下实体：用户登录验证实体类、用户信息表实体类、用户在拼车中信息实体类、拼车信息实体类、被请出人实体类、车主抢单实体类、车主车辆信息实体类、交易信息实体类、信誉变更实体类、路况均价实体类。下面对各个实体类进行具体介绍：

1、用户登录验证实体类

用户登录验证实体类中属性包括:用户ID 、QQ登录标识、微信登录标识、支付宝登录标识、淘宝登录标识、本地账号、本地密码、账号是否可用。如下图所示：

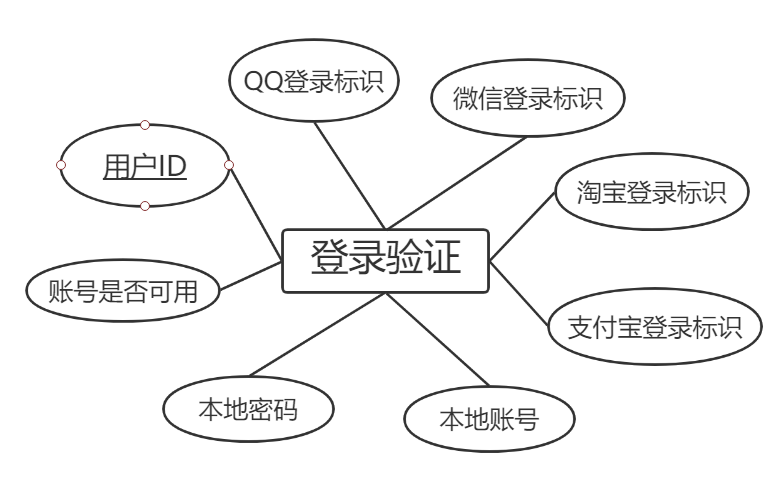


图4-6用户登录验证实体类

2、用户信息表实体类

用户信息表实体类中属性包括:用户ID、QQ号、微信号、手机号、邮箱、真实名称、年龄、性别、昵称、个人签名、钱包、信誉值。如下图所示：

3、用户在拼车中信息实体类

用户在拼车中信息实体类中属性包括:用户ID、拼车ID、昵称、用户头像、性别、拼车中留言、加入拼车时间、拼车完结时间、拼车状态、暴露的信息、QQ号、微信号、手机号、邮箱。如下图所示：

4、拼车信息实体类

拼车信息实体类中属性包括:拼车ID、发起人ID、加入拼车的IDs、大概路费、共需要人数、发起拼车时间、结束拼车时间、最早出发时间、最晚出发时间、出发地、目的地、拼车限制、拼车状态、退出/被请出拼车的IDs。如下图所示：

5、被请出人实体类

被请出人实体类中属性包括:被请出ID、发起请出人的ID、拼车的ID、被请出人的ID、同意请出的ID、不同意请出的ID、请出的原因、发起请出的时间、结束时间、最终结果。如下图所示：

6、车主抢单实体类

车主抢单实体类中属性包括:抢单的ID、司机的ID、拼车的ID、司机报价、抢单时间、抢单留言、抢单状态。如下图所示：

7、车主车辆信息实体类

车主车辆信息实体类中属性包括:车主ID、接单数量、车牌号码、车的颜色、车的座位数。如下图所示：

8、交易信息实体类

交易信息实体类中属性包括:交易ID、阿里支付编码、交易标识符、交易对象ID、交易相关IDs、交易状态、交易金额、交易原因、交易时间。如下图所示：

9、信誉变更实体类

信誉变更实体类中属性包括:信誉变更ID、变更用户ID、变更状态、变更分数、变更原因、变更相关IDs、变更时间。如下图所示：

10、路况均价实体类

路况均价实体类中属性包括:出发地、目的地、参考车费