|  |  |
| --- | --- |
| 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前后端使用的技术点

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端使用的技术点

Web端使用SpringBoot快速搭建，使用Maven对项目中的Jar包进行管理控制。集成了J2EE开发框架：Spring + Spring MVC + Spring Data JPA。后台对数据的处理和提供相应的访问接口。

数据解析工具使用Apache POI对题库Word文档进行解析，使用正则表达式对文档中的每行数据进行分析，然后将数据进行规整存储。个别技术解释如下：

### 2.2.1 SpringBoot

SpringBoot是将Spring应用更加轻量化，所有的Spring的应用开箱即用，提供很多默认配置来简化项目配置[3]。有内嵌式Tomcat服务器，没有大量的XML的冗余配置。本项目使用SpringBoot集成了SSSP(Spring,SpringMVC,SpringData,

JPA)框架。项目中多使用注解开发。

### 2.2.2 Apache POI

Apache POI是对Microsoft Office格式的文档进行读写的框架。主要用于对题库的解析，配合正则表达式完成数据的规整化。

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Android端采用的XUtils3框架和Gson解析数据工具，以及Web端使用的SpringBoot集成框架的使用，Apache POI文档解析的使用。介绍了开发工具Eclipse和Android Developer Tools的使用。

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

用户使用手机端打开答题软件App，软件进入引导页面，用户可查看当前公告和检测版本是否需要升级。如果需要升级则自动下载最新安装包进行覆盖安装。再次打开。用户可选择进行答题或者复习错题，在答题过程中如果打错则系统自动将本题加入到错题集中，可在进入页面选择复习错题来打开错题库。在错题库中如果回答正确则将本题目移出错题库。在答题的过程中可选择模拟测试和查看所有答案的背题模式。如果开启背题模式，所有题目显示出答案，用户可以进行背题、学习。

后台是管理员使用的，需要将题库放到指定的文件夹中，系统启动的时候自动扫描改文件夹下的Word文档，将该文档中的所有题库存入到数据库中。以便手机端App访问，后台使用J2EE技术提供数据访问接口，将数据库中的数据通过转化传递到手机端。然后手机端进行数据解析就可以获取到所有的题目了。根据业务操作流程的描述，可以画出系统流程图。系统流程图如下图3-1所示。



图3-1 系统流程图

3.2 系统设计目标